ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Αρχιτεκτονική, Διαχείριση και Ασφάλεια Προγραμματιζόμενων Δικτυακών Υποδομών

Διδακτορική Διατριβή

Δημήτριος Α. Γλυνός

ΠΕΙΡΑΙΑΣ 2012

.....

Δημήτριος Α. Γλυνός

BSc (Hons) Computer Science, University of Salford, UK
MSc Computer & Information Networks, University of Essex, UK

Περίληψη

Οι προγραμματιζόμενες δικτυακές υποδομές επιτρέπουν τη δυναμική προσαρμογή των υπηρεσιών ενός δικτύου ως προς τις ανάγκες των χρηστών αυτού. Στόχος αυτής της διατριβής είναι η μελέτη των προγραμματιζόμενων δικτυακών υποδομών και η διερεύνηση μιας σειράς θεμάτων που αφορούν στη λειτουργία τους. Αρχικά μελετάται η αρχιτεκτονική δικτυακών υποδομών που περιλαμβάνουν προγραμματιζόμενους κόμβους στον κορμό και στα άκρα του δικτύου. Χρησιμοποιώντας ως παραδείγματα δύο καινοτόμες υπηρεσίες (δυναμική δρομολόγηση & επεξεργασία μηνυμάτων ηλ. ταχυδρομείου, και κατανεμημένη ανίχνευση πολυμορφικού κακόβουλου λογισμικού σε δικτυακή κίνηση), παρουσιάζονται τα σημαντικά εκείνα οφέλη που μπορούν να προσφέρουν οι υποδομές αυτού του τύπου στους χρήστες και στους διαχειριστές δικτύων πληροφοριών. Το πρώτο μέρος της διατριβής ολοκληρώνεται με την πρόταση αλγορίθμων και λογισμικού για τη βέλτιστη οργάνωση της λειτουργίας μιας προγραμματιζόμενης υποδομής όταν απαιτείται η ελάχιστη δυνατή κατανάλωση ενέργειας.

Το δεύτερο μέρος της διατριβής ασχολείται με θέματα ασφάλειας στις προγραμματιζόμενες δικτυακές υποδομές. Αρχικά εξετάζονται τα μέτρα ασφάλειας που μπορούν να εφαρμοστούν τόσο στην υλοποίηση όσο και στη λειτουργία τέτοιων υποδομών. Στη συνέχεια προτείνεται ένας νέος τύπος δικτύου εμπιστοσύνης που μπορεί να βοηθήσει σημαντικά τους προγραμματιζόμενους κόμβους κατά τη διαδικασία εύρεσης έμπιστων παρόχων υπηρεσιών. Για την αντιμετώπιση των προβλημάτων που προκύπτουν από κόμβους που παρουσιάζουν ψευδή στοιχεία ταυτοποίησης, προτείνεται μία λύση που εκμεταλλεύεται, μεταξύ άλλων, χαρακτηριστικά των κόμβων που μπορούν να επαληθευθούν όταν αυτοί επικοινωνούν μεταξύ τους μέσω ασύρματων δικτύων. Η διατριβή ολοκληρώνεται με την παρουσίαση των συμπερασμάτων της παραπάνω έρευνας καθώς και των μελλοντικών κατευθύνσεων αυτής.

Abstract

Programmable network infrastructures enable network services to dynamically adapt to their users' needs. This thesis discusses the technologies behind programmable network infrastructures and proposes solutions to specific problems that affect their operation. Initially, the architectures of two types of programmable networks are discussed: networks with programmable nodes in their core, and networks with programmable edge nodes. Using two novel services as examples (dynamic routing & filtering of e-mail messages, and distributed detection of polymorphic shellcode in network traffic) the distinct benefits that these architectures bring to users and administrators are shown. The first part of the thesis ends with the proposal of two algorithms for extending the lifetime of programmable networks, when each node is equipped with only a limited amount of energy resources.

The second part of the thesis discusses security issues that arise in programmable network architectures. It starts with an overview of the security measures that have been proposed in the literature to protect programmable nodes and their operation. It then proposes a new type of trust network that allows programmable nodes to quickly locate trustworthy service providers. To protect programmable networks from nodes that operate under fake identities, a multi-factor authentication scheme is proposed that utilizes, among other things, node characteristics that can be verified if the nodes are communicating via a wireless network. Finally, the conclusions of the thesis are presented along with ideas for future areas of research.

Περιεχόμενα

Ευχαριστίες

Η εκπόνηση της παρούσας διατριβής πραγματοποιήθηκε στο εργαστήριο «Διαδικτυακών και Τηλεπικοινωνιακών Συστημάτων, Υπηρεσιών και Ασφάλειας» του τμήματος Πληροφορικής του Πανεπιστημίου Πειραιώς.

Θα ήθελα να ευχαριστήσω θερμά τον επιβλέποντα στη διατριβή μου Καθηγητή κ. Χρήστο Δουληγέρη για τη σημαντική καθοδήγηση και υποστήριξη που μου παρείχε κατά τη διάρκεια εκπόνησης της διατριβής, Η εξαιρετική συνεργασία που είχαμε με γέμισε με εμπειρίες και γνώσεις που με βοήθησαν τόσο σαν επιστήμονα όσο και σαν άνθρωπο, και τον ευγνωμονώ ιδιαίτερα για αυτό.

Θα ήθελα ακόμη να ευχαριστήσω τον Αναπληρωτή Καθηγητή κ. Δημήτριο Γκιζόπουλο και τον Καθηγητή κ. Νικήτα–Μαρίνο Σγούρο, που διετέλεσαν μέλη της συμβουλευτικής μου επιτροπής και με βοήθησαν σημαντικά με την καθοδήγησή τους όλα αυτά τα χρόνια.

Οφείλω να ευχαριστήσω επίσης το Πανεπιστήμιο Πειραιώς για τα οικονομικά και υλικοτεχνικά μέσα που διέθεσε για την υποστήριξη της έρευνας της παρούσας διατριβής.

Τέλος, θα ήθελα να ευχαριστήσω ολόψυχα την οικογένεια μου, τους φίλους μου, καθώς και τους συνεργάτες Πάτροκλο Αργυρούδη και Νικόλαο Τσαγκαράκη, για την αμέριστη συμπαράσταση που μου έδειξαν κατά τη διάρκεια εκπόνησης της διατριβής.

Στους γονείς μου και στη μνήμη του στενού φίλου και συνεργάτη Αντώνη Πετρόπουλου.

Κεφάλαιο 1

Προγραμματιζόμενες Δικτυακές Υποδομές: Βασικές Έννοιες και Προβλήματα

Σε μεγάλα δίκτυα δεδομένων, όπως το Διαδίκτυο, οι κόμβοι που απαρτίζουν τον κορμό του δικτύου ασχολούνται κυρίως με τη δρομολόγηση πακέτων. Η δρομολόγηση ως διαδικασία αποτελεί ένα μικρό μόνο μέρος της συνολικής επεξεργασίας στην οποία υπόκεινται τα πακέτα δύο συστημάτων που επικοινωνούν μέσω ενός δικτύου. Το μεγαλύτερο μέρος της επεξεργασίας συμβαίνει στα ίδια τα επικοινωνούντα συστήματα τα οποία επί το πλείστον βρίσκονται στα άκρα του δικτύου. Η αρχιτεκτονική αυτή, που συνάδει με την αρχή λειτουργίας «από άκρο σε άκρο» (end-to-end principle [1]), επιτρέπει την ελάχιστη δυνατή επιβάρυνση των κόμβων που απαρτίζουν τον κορμό ενός δικτύου σε περιόδους συμφόρησης.

Στα μέσα της δεκαετίας του 1990 υπήρξε μια άνθηση στον ερευνητικό χώρο των κατανεμημένων συστημάτων. Η άνθηση αυτή οφειλόταν μερικώς στην εμφάνιση νέων τεχνολογιών λογισμικού, όπως η Java [2], οι οποίες επέτρεψαν την εκτέλεση του ίδιου λογισμικού σε συστήματα διαφορετικών αρχιτεκτονικών με αποδοτικό τρόπο, χρησιμοποιώντας μια ενδιάμεση μορφή κώδικα (byte code). Οι κατανεμημένες εφαρμογές μπορούσαν πλέον εκτός από δεδομένα να ανταλλάξουν και λογισμικό, το οποίο θα μπορούσε να εκτελεστεί άμεσα στα συστήματα – παραλήπτες αυτού. Παραδείγματα τέτοιων εφαρμογών ήταν οι κινητοί πράκτορες [3] που ενεργούσαν σε απομακρυσμένα συστήματα για λογαριασμό των χρηστών τους. Οι νέες αυτές δυνατότητες σε συνδυασμό με την εμφάνιση συνεπεξεργαστών που μπορούσαν να εκτελέσουν απευθείας την ενδιάμεση μορφή κώδικα [4] αλλά και την άνοδο της επεξεργαστικής ισχύος των υπολογιστικών και δικτυακών συστημάτων, οδήγησαν τόσο τους ερευνητές όσο και τη βιομηχανία στην αναζήτηση νέων πεδίων εφαρμογής αυτών των τεχνολογιών. Μία από τις πιο σημαντικές προτάσεις εκείνης της περιόδου ήταν οι Προγραμματιζόμενες Δικτυακές Υποδομές.

Οι Προγραμματιζόμενες Δικτυακές Υποδομές αποτελούνται από δικτυακούς κόμβους οι οποίοι παρέχουν τη δυνατότητα δυναμικής διαμόρφωσης των υπηρεσιών που προσφέρουν [5]. Η διαμόρφωση αυτή μπορεί να έχει τη μορφή είτε εισαγωγής νέου λογισμικού στον κόμβο, είτε ρύθμισης/ανανέωσης του λογισμικού που υπάρχει ήδη σε αυτόν. Στόχος της δυναμικής διαμόρφωσης είναι η αυτόματη προσαρμογή των λειτουργιών του δικτύου στις τρέχουσες ανάγκες των χρηστών αυτού. Η κεντρική ιδέα των προγραμματιζόμενων δικτυακών υποδομών αποτελεί έναν πειραματισμό πάνω στους τρόπους με τους οποίους μπορούν να επωφεληθούν οι χρήστες ενός δικτύου αν μεταφερθεί ένα μέρος της επεξεργασίας των δεδομένων από τα άκρα του δικτύου στο δικτυακό κορμό. Η δυνατότητα προγραμματισμού

των βασικών συστατικών του δικτύου επιτρέπει, μεταξύ άλλων, την παροχή νέων καινοτόμων υπηρεσιών αλλά και την πειραματική εξέταση νέων πρωτοκόλλων επικοινωνίας σε πραγματικές συνθήκες. Η πιο προηγμένη μορφή Προγραμματιζόμενων Δικτυακών Υποδομών είναι τα Ενεργά Δίκτυα [6], τα οποία επιτρέπουν τη δυναμική σύνθεση νέων υπηρεσιών βάσει πληροφοριών και λογισμικού που εμπεριέχεται στη δικτυακή κίνηση.

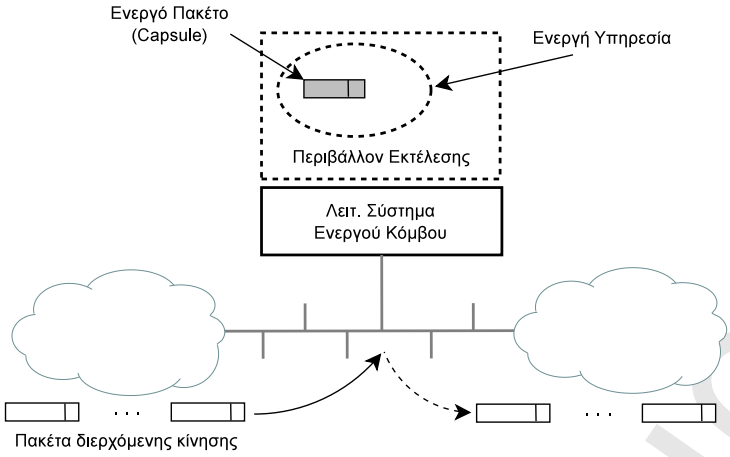
1.1 Ενεργά Δίκτυα

Τα «Ενεργά Δίκτυα» εισάγουν μια νέα αρχιτεκτονική στο χώρο των δικτύων Η/Υ. Μέρος των κόμβων που ανήκαν στον κορμό του δικτύου αντικαθίστανται από προγραμματιζόμενους κόμβους που ονομάζονται «Ενεργοί Κόμβοι». Ένας Ενεργός Κόμβος προτού προωθήσει ένα πακέτο που ανήκει σε κάποια εφαρμογή θα εξετάσει τα περιεχόμενα αυτού και ανάλογα με τις πληροφορίες που θα βρει μπορεί να υποβάλει σε κάποια περαιτέρω επεξεργασία το πακέτο ή να ξεκινήσει κάποια υπηρεσία από την οποία θα επωφεληθούν και τα επόμενα πακέτα της σχετικής συνόδου. Οι διεργασίες αυτές εκτελούνται σε Εικονικά Περιβάλλοντα Εκτέλεσης εντολών (execution environments) που διαχειρίζεται ο Ενεργός Κόμβος. Με τον τρόπο αυτό, το δίκτυο μετατρέπεται σε ένα προγραμματιζόμενο μέσο που προσφέρει κατά τόπους εξειδικευμένες υπηρεσίες σε δικτυακά πρωτόκολλα και εφαρμογές. Με τη δυναμική εγκατάσταση λογισμικού στους Ενεργούς Κόμβους γίνεται δυνατή η εφαρμογή νέων πρωτοκόλλων αλλά και η παροχή νέων υπηρεσιών στις οποίες οι κόμβοι του δικτύου παίζουν πλέον έναν πιο ενεργό ρόλο.

Η έρευνα στο χώρο των Ενεργών Δικτύων ξεκίνησε από προγράμματα που χρηματοδότησε το αμερικανικό στρατιωτικό κέντρο ερευνών (DARPA) [7] με στόχο τη δημιουργία προγραμματιζόμενων δικτύων που προσαρμόζονται δυναμικά στις υπηρεσίες τις οποίες παρέχουν. Θεμέλιο λίθο στην προσπάθεια αυτή αποτέλεσε η εργασία [6], στην οποία περιγράφεται μία μέθοδος διακίνησης δεδομένων και λογισμικού μέσω ειδικά διαμορφωμένων πακέτων τα οποία προγραμματίζουν δικτυακούς κόμβους. Τα πακέτα αυτά ονομάζονται «κάψουλες» (capsules) ή Ενεργά Πακέτα (Active Packets ή Smart Packets).

Η ερευνητική κοινότητα ξεχώρισε από νωρίς [8] δύο διαφορετικές προσεγγίσεις στην υλοποίηση των Ενεργών Δικτύων. Η πρώτη προσέγγιση [9, 10] είναι συνυφασμένη με τις τεχνολογίες τύπου *Push* και ορίζει ότι μαζί με τα πακέτα δεδομένων αποστέλλεται και ο κώδικας που απαιτείται προκειμένου να γίνει κάποια επεξεργασία επί αυτών στους Ενεργούς Κόμβους του δικτύου. Μάλιστα, για τα Ενεργά Πακέτα δημιουργήθηκε ένα νέο πρωτόκολλο, το Active Network Encapsulation Protocol [11]. Η ύπαρξη αυτού του πρωτοκόλλου σε ένα πακέτο σημειώνεται στην επικεφαλίδα IP, ώστε οι «μη ενεργοί» κόμβοι να μπορούν να αγνοήσουν τα περιεχόμενα αυτού. Επίσης, προτάθηκαν υψηλού επιπέδου γλώσσες, όπως η PLAN [12], για την υλοποίηση των μικρών εφαρμογών που θα εμπεριέχονται στα πακέτα. Γλώσσες όπως αυτή που παρουσιάζεται στην εργασία [13] επιτρέπουν επίσης τη σύνθεση νέων υπηρεσιών από υπηρεσίες που παρέχονται ήδη από Ενεργούς Κόμβους.

Τα Ενεργά Δίκτυα που ακολουθούν την παραπάνω προσέγγιση ονομάζονται Ισχυρά Ενεργά Δίκτυα. Ο πιο σημαντικός εκπρόσωπος αυτής της αρχιτεκτονικής είναι η πλατφόρμα ANTS [14] του ιδρύματος MIT, η οποία βοηθά στην έρευνα και την ανάπτυξη νέων δικτυακών πρωτοκόλλων. Στο σχήμα 1.1 παρουσιάζεται γραφικά η λειτουργία ενός Ενεργού Κόμβου σε ένα δίκτυο που ακολουθεί την παραπάνω προσέγγιση. Μπορεί κανείς εύκολα να διαπιστώσει ότι η προσέγγιση αυτή δίνει ιδιαίτερη έμφαση σε μεμονωμένα πακέτα και έτσι συνεργάζεται καλύτερα με υπηρεσίες που αποστέλλουν πληροφορίες σε αυτόνομα πακέτα (π.χ. υπηρεσίες που χρησιμοποιούν το πρωτόκολλο UDP). Η δυναμική επεξεργασία διερχόμενων πακέτων επιτρέπει μεν τη γρήγορη εξάπλωση νέων πρωτοκόλλων και υπηρεσιών,



Σχήμα 1.1: Κόμβος Ισχυρού Ενεργού Δικτύου

δημιουργεί όμως παράλληλα μια πληθώρα θεμάτων ασφάλειας για το περιβάλλον εκτέλεσης του Ενεργού Κόμβου. Μια πιο αναλυτική εξέταση αυτών των προβλημάτων παρουσιάζεται στο κεφάλαιο 5.

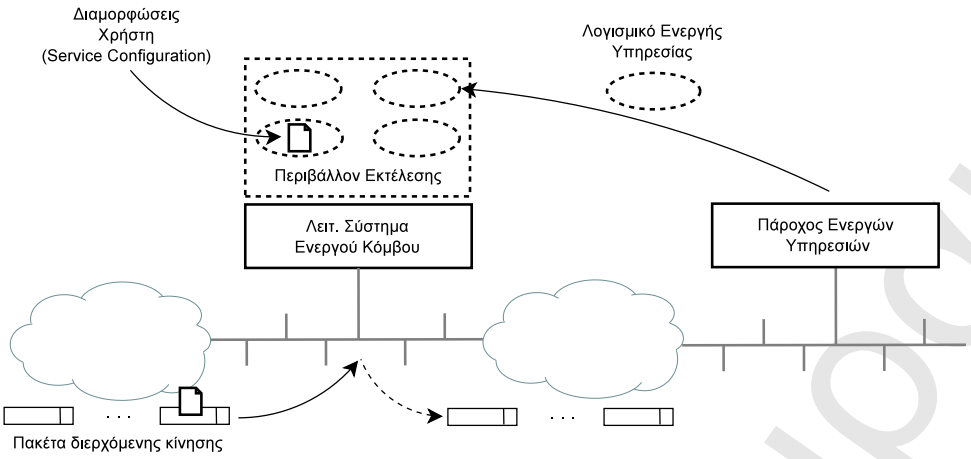
Η δεύτερη προσέγγιση εστιάζει περισσότερο στον ίδιο τον Ενεργό Κόμβο και στις υπηρεσίες που αυτός παρέχει. Στη θέση των δυναμικών υπηρεσιών που καθορίζονταν από τα διερχόμενα πακέτα υπάρχει πλέον ένα σύνολο προκαθορισμένων υπηρεσιών. Ο Ενεργός Κόμβος εφαρμόζει μια σειρά φίλτρων στα διερχόμενα πακέτα, ώστε να εντοπίσει και να επεξεργαστεί μόνο αυτά που αντιστοιχούν στις υπηρεσίες που αυτός υποστηρίζει. Οι υπηρεσίες αυτές αν δεν παρέχονται ήδη μέσω τοπικού λογισμικού, μπορεί να αναζητηθούν σε κάποιο Πάροχο Ενεργών Υπηρεσιών. Τυπικά ο πάροχος αυτός παίζει το ρόλο μιας υπηρεσίας καταλόγου (Directory Server) σχετικής με λογισμικό για Ενεργούς Κόμβους. Μόλις εντοπιστεί το επιθυμητό λογισμικό, αυτό μεταφέρεται και εκτελείται στον Ενεργό Κόμβο (βλ. σχήμα 1.2).

Όταν τα δεδομένα ενός χρήστη φτάσουν σε μια τέτοια υπηρεσία, θα αναζητηθούν (είτε στο μηχάνημα του χρήστη, είτε σε πακέτα αυτού, είτε σε κάποιο Directory Server) οι προτιμήσεις του για αυτή την υπηρεσία. Στη συνέχεια η εφαρμογή που εκτελείται στον Ενεργό Κόμβο θα μεταβάλλει την συμπεριφορά της ως προς την κίνηση του χρήστη ανάλογα με τις προτιμήσεις αυτού. Τα προγραμματιζόμενα δίκτυα αυτού του τύπου ονομάζονται Ενεργά Δίκτυα Επιπέδου Εφαρμογής (Application Layer Active Networking) [15] ή Ασθενή Ενεργά Δίκτυα και αποτελούν μια τεχνολογία τύπου *Pull*.

Η προσέγγιση αυτή επιτρέπει τη δημιουργία ιδεατών ιδιωτικών δικτύων όπου πολλαπλοί χρήστες αξιοποιούν ένα κοινό σύνολο από υπηρεσίες που προσαρμόζονται αυτόματα στις απαιτήσεις κάθε χρήστη. Επίσης παρέχει μεγαλύτερη δυνατότητα ελέγχου επί των προσφερόμενων υπηρεσιών, περιορίζοντας όμως σημαντικά τη δυνατότητα προγραμματισμού του δικτυακού μέσου.

Στη βιβλιογραφία μπορεί κανείς να βρει πλήθος από ερευνητικές πλατφόρμες που αξιοποιούν τα Ενεργά Δίκτυα αυτής της προσέγγισης [8, 16, 5]. Μία τέτοια πλατφόρμα είναι η ConCEPT [17], η οποία παρέχει υπηρεσίες προσωρινής αποθήκευσης και προσαρμογής ιστοσελίδων (web-caching & content-adaptation). Στο ConCEPT οι διακομιστές προσωρινής αποθήκευσης ιστοσελίδων (web-caches) μεταφέρουν μεταξύ τους δεδομένα ιστοσελίδων σε συμπιεσμένη μορφή. Όταν ο χρήστης ζητήσει μια ιστοσελίδα, αυτή διαμορφώνεται και παραδίδεται σε αυτόν σύμφωνα με μια σειρά φίλτρων που εκείνος έχει πρωτύτερα ορίσει.

Το ABONE [18] ήταν ένα πειραματικό δίκτυο από Ενεργούς Κόμβους, όπου οι ερευνητές



Σχήμα 1.2: Κόμβος Ασθενούς Ενεργού Δικτύου και Πάροχος Ενεργών Υπηρεσιών

μπορούσαν να ελέγξουν την λειτουργία των Ενεργών Υπηρεσιών αλλά και των ίδιων των Ενεργών Κόμβων, σε πραγματικές συνθήκες κίνησης. Η υλοποίηση ενός Ενεργού Δικτύου τέτοιας κλίμακας έδωσε περισσότερη ώθηση στην έρευνα σε τομείς όπως η διαχείριση των Ενεργών Υπηρεσιών [19, 20], η ασφάλεια των Ενεργών Κόμβων [21] αλλά και των δεδομένων [22] που αυτοί επεξεργάζονται.

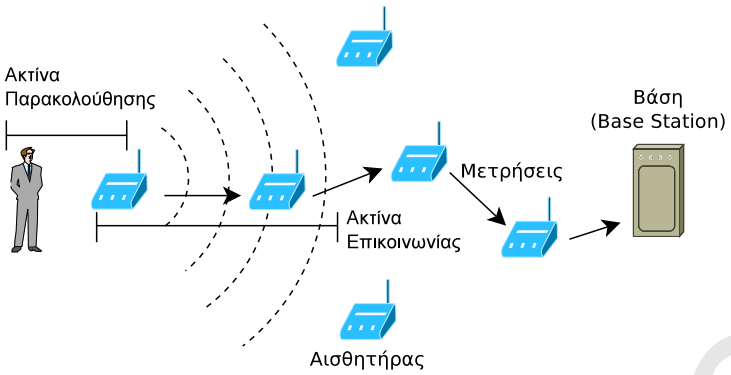
Τα προβλήματα ασφάλειας και απόδοσης καθώς και το περιορισμένο εύρος εφαρμογών των Ενεργών Δικτύων ήταν μερικοί από τους λόγους για τους οποίους δεν υπήρξε ευρεία υιοθέτηση αυτών από εφαρμογές στο Διαδίκτυο [23]. Ωστόσο, πολλές από τις τεχνολογίες που δημιουργήθηκαν για τα Ενεργά Δίκτυα, μπορεί κανείς να τις συναντήσει σήμερα στις τεχνολογίες προσαρμογής περιεχομένου (content adaptation) [24], στις τεχνολογίες πλέγματος (grid computing) [25], στις υποδομές ιδεατών υπολογιστών γενικής χρήσης (virtualization infrastructures) [26] αλλά και στις υποδομές συννέφου (cloud computing) [27].

1.2 Προγραμματιζόμενα Δίκτυα Αισθητήρων

Τα Δίκτυα Αισθητήρων [28] αποτελούνται από συσκευές που έχουν τη δυνατότητα να λαμβάνουν μετρήσεις από το περιβάλλον στο οποίο είναι τοποθετημένες και να επικοινωνούν (επί το πλείστον) ασύρματα μεταξύ τους. Τα δεδομένα που συλλέγουν μεταφέρονται μέσω του δικτύου των συσκευών σε ένα σταθμό *βάσης* (base station), όπου γίνεται η καταγραφή και ανάλυση αυτών (βλ. σχήμα 1.3).

Η επικοινωνία μεταξύ των συσκευών γίνεται μέσω ασυρμάτου δικτύου, το οποίο δημιουργείται ad hoc μέσω των κατάλληλων υποσυστημάτων των συσκευών. Ένας αισθητήρας μπορεί να επικοινωνήσει με οποιονδήποτε άλλο αισθητήρα βρίσκεται σε απόσταση μικρότερη της ακτίνας επικοινωνίας R_c που χαρακτηρίζει το υποσύστημα επικοινωνίας αυτού. Ομοίως, ένας αισθητήρας μπορεί να παρατηρήσει αντικείμενα ή φαινόμενα που βρίσκονται σε απόσταση μικρότερη της ακτίνας παρακολούθησης R_s αυτού. Κάθε αισθητήρας μπορεί να είναι εξοπλισμένος με ένα ή περισσότερα υποσυστήματα επικοινωνίας, όπως και παρακολούθησης.

Στις εφαρμογές των δικτύων αισθητήρων περιλαμβάνονται στρατιωτικές εφαρμογές [28], παρακολούθηση συνθηκών περιβάλλοντος [29], παρακολούθηση συγκεκριμένων στόχων [30], παροχή προειδοποίησης σε περιπτώσεις (φυσικών ή άλλων) καταστροφών [31, 32] κ.α.. Για τις εφαρμογές αυτές οι αισθητήρες απαιτούν κάποια ενεργειακή αυτονομία, η οποία συνήθως παρέχεται μέσω μιας μπαταρίας. Το χρονικό διάστημα για το οποίο ένας



Σχήμα 1.3: Επικοινωνία με τη βάση σε ένα δίκτυο αισθητήρων

αισθητήρας παραμένει λειτουργικός ονομάζεται *διάρκεια ζωής* του αισθητήρα.

Για την καλύτερη αξιοποίηση της ενέργειας που έχει στη διάθεσή της η κάθε συσκευή, λειτουργικά συστήματα για αισθητήρες, όπως το TinyOS [33], ορίζουν τρεις καταστάσεις λειτουργίας:

- την ενεργή κατάσταση (active mode)
- την κατάσταση ύπνου (sleep mode)
- την ανενεργή κατάσταση (off mode)

Στην *ενεργή* κατάσταση ένας αισθητήρας μπορεί να επικοινωνήσει με άλλους αισθητήρες, μπορεί να προβεί σε μετρήσεις και μπορεί, επίσης, να δρομολογήσει δεδομένα άλλων αισθητήρων. Στην κατάσταση *ύπνου* ο αισθητήρας δεν συμμετέχει σε καμία από τις παραπάνω δραστηριότητες, αλλά μπορεί να μεταβεί στην ενεργή κατάσταση μετά τη λήψη ειδικού σήματος (από άλλους αισθητήρες, από τη βάση ή από το λειτουργικό του σύστημα). Τέλος, στην *ανενεργή* κατάσταση ο αισθητήρας είναι πλήρως απενεργοποιημένος. Χρησιμοποιώντας τις παραπάνω τρεις καταστάσεις οι εφαρμογές των Δικτύων Αισθητήρων μπορούν να ρυθμίσουν ποιοι αισθητήρες θα συμμετέχουν σε κάθε φάση λειτουργίας του δικτύου.

Η τοποθέτηση των αισθητήρων στο περιβάλλον παρακολούθησης μπορεί να γίνει με συγκεκριμένο ή ακαθόριστο τρόπο. Π.χ. σε στρατιωτικές εφαρμογές μπορεί να γίνει ρίψη αυτών από αεροπλάνο και έτσι μπορεί να μην είναι προβλέψιμη η τοπολογία του δικτύου που θα σχηματίσουν. Επίσης, υπάρχει περίπτωση, εξαιτίας δυσμενών συνθηκών που επικρατούν στο περιβάλλον εγκατάστασης αυτών, να μην είναι δυνατή η μετακίνηση ή επιδιόρθωση αυτών. Σε τέτοιες συνθήκες συνήθως τοποθετείται μεγαλύτερος αριθμός αισθητήρων ώστε να καλύπτονται οι ανάγκες του δικτύου ακόμη και σε περιπτώσεις καταστροφών ή άλλων δυσλειτουργιών του σχετικού εξοπλισμού.

Η περιορισμένη δυνατότητα αναβάθμισης αλλά και η περιορισμένη διάρκεια ζωής των αισθητήρων οδήγησαν την ερευνητική κοινότητα στη διερεύνηση λύσεων απομακρυσμένου προγραμματισμού των συσκευών αυτών από το σταθμό βάσης [34]. Αρχικά, έγιναν κάποιες προτάσεις για την επέκταση του ενσωματωμένου λειτουργικού συστήματος των αισθητήρων ώστε να μπορεί να υποστηρίζει τη δυναμική φόρτωση νέου λογισμικού [35]. Στη συνέχεια διερευνήθηκαν τρόποι με τους οποίους θα γινόταν η φόρτωση και εκτέλεση του «ξένου» λογισμικού. Στην εργασία [36] το λογισμικό αυτό έχει τη μορφή αρθρωμάτων (object modules) τα οποία καλούν συναρτήσεις μιας βιβλιοθήκης που παίζει το ρόλο της διεπαφής με το λειτουργικό σύστημα. Αντίθετα, στην εργασία [37] προτείνεται μια αρχιτεκτονική όπου

εκτελούνται μικρές εφαρμογές σε γλώσσα σεναρίου στον αισθητήρα. Τέλος, στην εργασία [38] προτείνεται η υλοποίηση μιας πλατφόρμας Ισχυρών Ενεργών Δικτύων για Δίκτυα Αισθητήρων, η οποία εκτελεί κάψουλες σε μια εικονική μηχανή (virtual machine).

Σε κάθε περίπτωση η μετατροπή ενός Δικτύου Αισθητήρων σε μια προγραμματιζόμενη δικτυακή υποδομή μπορεί να ωφελήσει τόσο στην καλύτερη αξιοποίηση των διαθέσιμων πόρων (με την εφαρμογή εναλλακτικών προγραμμάτων λειτουργίας) όσο και στον περιορισμό των επιπτώσεων που μπορεί να επιφέρει η προβληματική λειτουργία κάποιων συσκευών (με την εφαρμογή φίλτρων και εναλλακτικών κανόνων δρομολόγησης). Οι δυνατότητες αυτές είναι ιδιαίτερα σημαντικές, ειδικά για δίκτυα αισθητήρων στα οποία είναι δύσκολη έως αδύνατη η φυσική πρόσβαση και κατ' επέκταση η αντικατάσταση του εξοπλισμού.

1.3 Προγραμματιζόμενα Δίκτυα Επικάλυψης

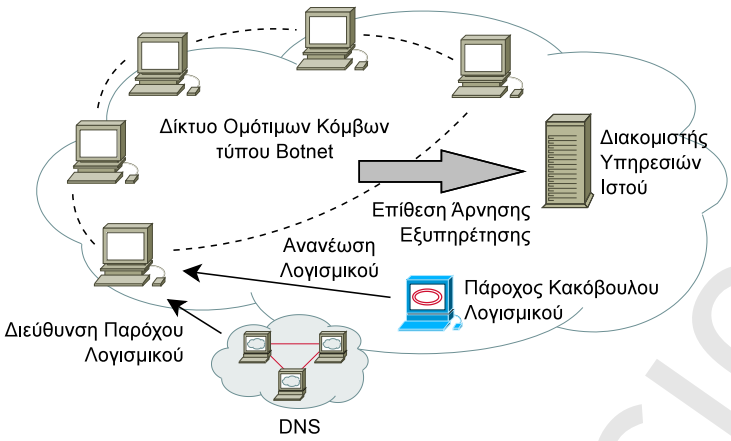
Τα Δίκτυα Επικάλυψης [39] είναι εικονικά δίκτυα που αποτελούνται από κόμβους που μπορεί να μην έχουν μεταξύ τους απευθείας δικτυακή σύνδεση. Την απαραίτητη εικονική ζεύξη μεταξύ των κόμβων την παρέχουν εφαρμογές, οι οποίες χρησιμοποιούν υπάρχοντα πρωτόκολλα επικοινωνίας (π.χ. TCP, UDP) για να μεταφέρουν πάνω από πραγματικά δίκτυα τα μηνύματα των κόμβων. Σε πολλές εφαρμογές όπου χρησιμοποιούνται δίκτυα επικάλυψης, οι κόμβοι λειτουργούν ως ομότιμοι [40], προσφέροντας όλοι τις ίδιες υπηρεσίες (δρομολόγηση, διαφήμιση στοιχείων άλλων κόμβων κλπ.) και συμβάλλοντας όλοι επί ίσοις όροις στη λειτουργία του εικονικού δικτύου.

Στη βιβλιογραφία έχουν προταθεί διάφορες προγραμματιζόμενες υποδομές για δίκτυα επικάλυψης οι οποίες επιτρέπουν μεταξύ άλλων την κατανεμημένη εκτέλεση εφαρμογών στους κόμβους του δικτύου επικάλυψης [41, 42] καθώς και την παροχή νέων δυναμικών υπηρεσιών μέσω του δικτύου επικάλυψης [43]. Σε σχέση με τα ενεργά δίκτυα και τα δίκτυα αισθητήρων, τα δίκτυα επικάλυψης προσφέρουν δυνατότητες προγραμματισμού των κόμβων σε υψηλότερο επίπεδο (επίπεδο εφαρμογής).

Δυστυχώς, η παραπάνω αρχιτεκτονική αξιοποιείται σήμερα και από κακόβουλες εφαρμογές. Επιτήδειοι που σχεδιάζουν κατανεμημένες επιθέσεις τύπου *άρνησης εξυπηρέτησης* [44] εκμεταλλεύονται ευπάθειες στο λογισμικό προσωπικών υπολογιστών και διακομιστών που συνδέονται στο διαδίκτυο, προκειμένου να τους εντάξουν σε ένα ειδικού τύπου δίκτυο επικάλυψης που ονομάζεται botnet [45]. Μέσω λογισμικού που έχουν εγκαταστήσει στους υπολογιστές των θυμάτων, οι επιτήδειοι μπορούν να υποκλέψουν ευαίσθητες πληροφορίες από αυτά τα συστήματα [46] αλλά και να τα χρησιμοποιήσουν ως ενδιάμεσους κόμβους σε διάφορους τύπους επιθέσεων προς τρίτα συστήματα [47]. Εφόσον το λογισμικό αυτό επιτρέπει την εκτέλεση οποιουδήποτε λογισμικού στους υπολογιστές των θυμάτων, κατ' επέκταση το botnet αποτελεί μια («παρασιτική») προγραμματιζόμενη υποδομή που λειτουργεί υπό την εποπτεία των επιτιθέμενων.

Το λογισμικό που εγκαθίσταται στους υπολογιστές των θυμάτων είναι μερικές φορές ιομορφικό και προσπαθεί να εγκαταστήσει τον εαυτό του και σε άλλους υπολογιστές στο Διαδίκτυο (βλ. worm [48]). Η αναβάθμιση του λογισμικού του botnet και η παραλαβή νέων εντολών σχετικών με επιθέσεις μπορεί να γίνει είτε από εξωτερική, για το δίκτυο, πηγή είτε από εσωτερική. Η εξωτερική πηγή συνήθως είναι κάποιος διακομιστής του οποίου η διεύθυνση IP δεν είναι σταθερή και την οποία μαθαίνουν οι κόμβοι του botnet μέσω ερωτημάτων σε διακομιστές υπηρεσιών DNS. Προκειμένου να κρύψουν τη διεύθυνση των διακομιστών που αποστέλλουν κακόβουλο λογισμικό, οι επιτιθέμενοι χρησιμοποιούν μια τεχνική γνωστή ως "fast flux" [49] κατά την οποία εναλλάσσουν μέσα σε μικρό χρονικό

διάστημα τις διευθύνσεις των διακομιστών που αντιστοιχούν σε ένα DNS όνομα (hostname).



Σχήμα 1.4: Προγραμματιζόμενο Δίκτυο Επικάλυψης κακόβουλης εφαρμογής

Οι επιτιθέμενοι χρησιμοποιούν, επίσης, διάφορες τεχνικές για να κρύψουν τη δική τους ταυτότητα στο Διαδίκτυο. Μία από αυτές είναι η χρήση της υπηρεσίας IRC για την επικοινωνία με τους κόμβους του botnet [45]. Μία νεότερη τεχνική εντάσσει τους κόμβους του botnet σε ένα δίκτυο ομότιμων κόμβων μέσω του οποίου οι κόμβοι μπορούν να λάβουν λογισμικό, εντολές αλλά και να επικοινωνήσουν μεταξύ τους. Έτσι, αρκεί ένας από αυτούς να επικοινωνήσει με το σύστημα του επιτιθέμενου (ή με ένα διακομιστή μέσω DNS fast flux) ώστε να λάβουν στη συνέχεια όλοι τις νέες εντολές ή το νέο λογισμικό. Η τεχνική αυτή χρησιμοποιήθηκε από το worm "Conficker C" [50] και παρουσιάζεται γραφικά στο σχήμα 1.4.

Τέλος, εφόσον το δίκτυο επικάλυψης ενός botnet με ομότιμους κόμβους έχει τη δυνατότητα να ανανεώσει το λογισμικό του, μπορεί ανά πάσα στιγμή να αλλάξει και τον τρόπο με τον οποίο επικοινωνούν οι κόμβοι αυτού. Επειδή οι αλλαγές αυτής της μορφής επηρεάζουν τη λειτουργία του δικτύου επικάλυψης, τα botnet που παρέχουν τέτοιες δυνατότητες χαρακτηρίζονται ως προγραμματιζόμενες δικτυακές υποδομές.

1.4 Στόχοι και δομή

Σκοπός αυτής της διατριβής είναι η μελέτη των προγραμματιζόμενων δικτυακών υποδομών και η διερεύνηση μιας σειράς θεμάτων που αφορούν στη λειτουργία αυτών. Αρχικά μελετάται η αρχιτεκτονική δικτυακών υποδομών που περιλαμβάνουν προγραμματιζόμενους κόμβους σε πρώτη φάση στον κορμό και σε δεύτερη φάση στα άκρα του δικτύου. Χρησιμοποιώντας ως παραδείγματα δύο καινοτόμες υπηρεσίες που βασίζονται σε προγραμματιζόμενες υποδομές, παρουσιάζονται τα σημαντικά εκείνα οφέλη που μπορούν να προσφέρουν οι υποδομές αυτού του τύπου στους χρήστες και στους διαχειριστές δικτύων πληροφοριών. Επίσης, προτείνονται λύσεις σε συγκεκριμένα προβλήματα ασφάλειας και απόδοσης που παρατηρούνται σε υποδομές που έχουν υλοποιηθεί σύμφωνα με τις δύο παραπάνω αρχιτεκτονικές.

Ως παράδειγμα υποδομής με προγραμματιζόμενους κόμβους στον κορμό του δικτύου χρησιμοποιείται αυτή της υπηρεσίας δυναμικής δρομολόγησης και επεξεργασίας μηνυμάτων ηλεκτρονικού ταχυδρομείου. Η υπηρεσία αυτή επιτρέπει τη δυναμική δρομολόγηση της ηλεκτρονικής αλληλογραφίας στο δίκτυο όπου συνδέεται κάθε φορά ένας χρήστης (δίκτυο υποδοχής), λύνοντας έτσι το πρόβλημα της τριγωνικής δρομολόγησης που συναντάται

σε τεχνολογίες περιαγωγής όπως η Mobile IP [51]. Επίσης, επιτρέπει στους παραλήπτες των μηνυμάτων να ορίσουν συγκεκριμένο λογισμικό το οποίο θα επεξεργαστεί τα εισερχόμενα μηνύματα καθώς αυτά μεταφέρονται προς το σταθμό εργασίας τους. Το λογισμικό αυτό μπορεί να συμβάλει σημαντικά στη μείωση της κίνησης που σχετίζεται με την ανεπιθύμητη αλληλογραφία αλλά και να επιτρέψει την αυτόματη προσαρμογή των μηνυμάτων σε μορφές πιο κατάλληλες για τις συσκευές των χρηστών. Η νέα αυτή υπηρεσία περιγράφεται αναλυτικά στο δεύτερο κεφάλαιο της διατριβής και προτείνονται λύσεις βασισμένες στην κρυπτογραφία δημοσίου κλειδιού για την ασφαλή μεταφορά λογισμικού και ρυθμίσεων στους προγραμματιζόμενους κόμβους της σχετικής υποδομής.

Παράδειγμα υποδομής με προγραμματιζόμενους κόμβους στα άκρα του δικτύου, αποτελεί αυτή της υπηρεσίας κατανεμημένης ανίχνευσης πολυμορφικού κακόβουλου λογισμικού σε δικτυακή κίνηση. Η νέα αυτή υπηρεσία, που εξετάζεται στο τρίτο κεφάλαιο της διατριβής, στηρίζεται στην επεξεργασία των πακέτων της δικτυακής κίνησης σε προστατευμένα περιβάλλοντα εικονικής εκτέλεσης (virtual execution environments) που βρίσκονται στους κόμβους της προγραμματιζόμενης υποδομής. Μία από τις τεχνικές ανίχνευσης που αξιοποιεί η προτεινόμενη υπηρεσία, εντοπίζει κακόβουλο λογισμικό παρατηρώντας την έκβαση «επικίνδυνων» κλήσεων συστήματος. Από τα αποτελέσματα της πειραματικής αξιολόγησης προκύπτει ότι η συγκεκριμένη μέθοδος μπορεί να εντοπίσει περισσότερες μορφές κακόβουλου λογισμικού εκμετάλλευσης τρωτοτήτων (polymorphic/metamorphic shellcode) και με μεγαλύτερη αξιοπιστία, από οποιαδήποτε άλλη μέθοδο έχει προταθεί μέχρι σήμερα στη σχετική βιβλιογραφία.

Στο τέταρτο κεφάλαιο εξετάζεται το πρόβλημα της αποδοτικής παροχής υπηρεσιών από μια πλατφόρμα της οποίας οι κόμβοι έχουν περιορισμένα αποθέματα ενέργειας. Συγκεκριμένα, προτείνονται δύο αλγόριθμοι για την παραγωγή ομάδων κάλυψης στόχων σε προγραμματιζόμενα δίκτυα αισθητήρων, οι οποίοι επεκτείνουν τη διάρκεια ζωής του δικτύου πέραν από την τυπική διάρκεια ζωής ενός αισθητήρα. Αρχικά μοντελοποιείται το πρόβλημα της κάλυψης στόχων και προτείνεται ένα ευρετικός αλγόριθμος που παράγει ομάδες κάλυψης δίχως κοινά μέλη. Κάθε ομάδα κόμβων μπορεί να τεθεί σε λειτουργία αυτόνομα και προσφέρει κάλυψη επί του συνόλου των στόχων. Στη συνέχεια παρουσιάζεται ένας βελτιωμένος αλγόριθμος ο οποίος επιτρέπει την παραγωγή ομάδων κάλυψης στις οποίες μπορούν να συμμετέχουν και κοινοί αισθητήρες. Τα πορίσματα της πειραματικής αξιολόγησης δείχνουν ότι ο πρώτος αλγόριθμος παράγει περισσότερες λύσεις και σε μικρότερο χρονικό διάστημα από τους αντίστοιχους αλγορίθμους που έχουν προταθεί στη βιβλιογραφία. Αντίστοιχα, ο δεύτερος αλγόριθμος παράγει περισσότερες λύσεις από τους ομολόγους του (και από τον πρώτο) αλλά με μια χρονική καθυστέρηση η οποία οφείλεται στο στάδιο βελτιστοποίησης των παραμέτρων του αλγορίθμου. Στο ίδιο κεφάλαιο περιλαμβάνεται επίσης μια σύντομη περιγραφή του λογισμικού που αναπτύχθηκε για την πειραματική αξιολόγηση των αλγορίθμων κάλυψης.

Το δεύτερο μέρος της διατριβής ασχολείται με θέματα ασφάλειας στις προγραμματιζόμενες δικτυακές υποδομές. Στο κεφάλαιο πέντε εξετάζονται τα μέτρα ασφάλειας που έχουν προταθεί στη βιβλιογραφία για την προστασία του λογισμικού, των υπηρεσιών, των χρηστών αλλά και των κόμβων μιας προγραμματιζόμενης δικτυακής υποδομής. Επίσης, εξετάζεται το πρόβλημα της επιλογής έμπιστων παρόχων υπηρεσιών σε ελεύθερα προγραμματιζόμενες υποδομές και κρίνεται ότι μιας τέτοιας μορφής επιλογή δε μπορεί να γίνει με στατικά κριτήρια.

Στο έκτο κεφάλαιο προτείνεται η χρήση ενός ειδικού δικτύου εμπιστοσύνης για την αυτόματη επιλογή έμπιστων παρόχων υπηρεσιών σε προγραμματιζόμενες υποδομές. Το προτεινόμενο δίκτυο εμπιστοσύνης επιτρέπει σε κόμβους να υπολογίσουν τιμές εμπιστοσύνης προς κόμβους-παρόχους, λαμβάνοντας υπόψη και τις αξιολογήσεις τρίτων κόμβων

του δικτύου. Χαρακτηριστκό αυτού του δικτύου είναι το γεγονός ότι περιορίζει το μήκος των μονοπατιών εμπιστοσύνης σε δύο βήματα, επιτρέποντας έτσι τον ταχύτερο υπολογισμό τιμών εμπιστοσύνης σε δίκτυα με μεγάλο αριθμό κόμβων. Σε αντίθεση με άλλες προτάσεις από τη βιβλιογραφία, το προτεινόμενο δίκτυο εμπιστοσύνης υποστηρίζει την ύπαρξη πολλαπλών ακμών μεταξύ των κόμβων, περιγράφοντας έτσι καλύτερα τις σχέσεις που προκύπτουν από τις διαφορετικές υπηρεσίες που μπορεί να παρέχει κάθε κόμβος. Το σύνολο των χαρακτηριστικών του προτεινόμενου δικτύου παρουσιάζεται επίσης με αλγεβρικό τρόπο ώστε να είναι ευκολότερη η σύγκριση αυτού με άλλες προτάσεις από τη βιβλιογραφία. Το κεφάλαιο αυτό ολοκληρώνεται με την πειραματική και ποιοτική αξιολόγηση του παραπάνω δικτύου εμπιστοσύνης. Τα πειραματικά δεδομένα δείχνουν ότι ο αλγόριθμος υπολογισμού εμπιστοσύνης λειτουργεί πιο αποδοτικά από αντίστοιχους που έχουν προταθεί στη βιβλιογραφία. Επίσης, το προτεινόμενο δίκτυο παρουσιάζεται ιδιαίτερα ανθεκτικό στις περισσότερες γνωστές μορφές επιθέσεων που προσβάλλουν δίκτυα τέτοιου τύπου. Εξαίρεση αποτελούν οι επιθέσεις στις οποίες οι επιτιθέμενοι παρουσιάζουν ψευδή στοιχεία ταυτοποίησης.

Στο έβδομο κεφάλαιο παρουσιάζεται μία νέα μέθοδος για την ασφαλή ταυτοποίηση προγραμματιζόμενων κόμβων η οποία συνδέει τα φυσικά χαρακτηριστικά ενός κόμβου με την ψηφιακή του ταυτότητα (ψηφιακό πιστοποιητικό). Η μέθοδος αυτή εξετάζει διάφορα χαρακτηριστικά των κόμβων, όπως το αποτύπωμα εκπομπής όταν αυτοί επικοινωνούν ασύρματα, ώστε το αποτέλεσμα της διαδικασίας ταυτοποίησης να είναι πιο αξιόπιστο. Περιγράφεται επίσης με ποιον τρόπο η παραπάνω μέθοδος μπορεί να προστατεύσει τους προγραμματιζόμενους κόμβους ασύρματων δικτύων από επιθέσεις τύπου «μεσάζοντα» (man in the middle) ή “Sybil” και προτείνονται χαρακτηριστικά που μπορεί να αξιοποιηθούν για αυτό το σκοπό.

Η διατριβή ολοκληρώνεται με την παρουσίαση στο όγδοο κεφάλαιο των συμπερασμάτων της παραπάνω έρευνας καθώς και των μελλοντικών κατευθύνσεων αυτής.

Κεφάλαιο 2

Προγραμματιζόμενες Υποδομές στο Δικτυακό Κορμό

Οι προγραμματιζόμενες υποδομές που αποτελούν μέρος του δικτυακού κορμού επιτρέπουν τη συνεργασία δρομολογητών και άλλων δικτυακών συσκευών στο πλαίσιο μιας υπηρεσίας. Χαρακτηριστικά παραδείγματα αποτελούν οι προγραμματιζόμενες υποδομές που προσφέρουν υπηρεσίες προστασίας ενάντια σε κατανεμημένες επιθέσεις τύπου άρνησης εξυπηρέτησης (DDoS attacks) [52, 53] καθώς και οι υποδομές που μετασχηματίζουν κατάλληλα πολυμεσικό υλικό ώστε αυτό να μπορεί να προβληθεί σε συσκευές περιορισμένων δυνατοτήτων [54].

Μιας και σε αυτές τις υποδομές συνεργάζονται δικτυακές συσκευές από διαφορετικούς οργανισμούς (administrative domains), ο τύπος του προγραμματισμού που μπορεί να επιφέρει μια συσκευή σε μια άλλη (ή ένας χρήστης σε μια συσκευή) είναι συνήθως πολύ περιορισμένος εξαιτίας των εφαρμοζόμενων πολιτικών ασφαλείας. Επίσης, εφόσον οι συσκευές αυτές αποτελούν μέρος του δικτυακού κορμού καλούνται συνήθως να επεξεργαστούν μεγάλο όγκο δεδομένων και έτσι η οποιαδήποτε επιβάρυνσή τους από ενεργές υπηρεσίες θα πρέπει να διατηρηθεί σε χαμηλά επίπεδα. Για τους παραπάνω λόγους, οι προγραμματιζόμενες υποδομές που βρίσκονται στον κορμό του δικτύου, παρέχουν ως επί το πλείστον ενεργές υπηρεσίες σε επίπεδο *εφαρμογής* (Application Layer Active Networking).

Οι ενεργές υπηρεσίες σε επίπεδο εφαρμογής δεν προγραμματίζονται από τα διερχόμενα πακέτα αλλά παραμετροποιούνται από αυτά. Έτσι, η διερχόμενη κίνηση καθορίζει τον τρόπο με τον οποίο θα λειτουργήσει η υπηρεσία αλλά δεν παρέχει την υλοποίηση αυτής. Υπεύθυνος για την επιλογή της υλοποίησης μιας υπηρεσίας παραμένει ο διαχειριστής του κόμβου.

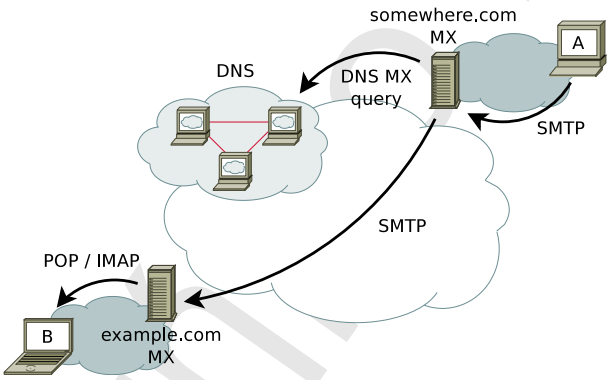
Στο κεφάλαιο αυτό θα παρουσιαστεί μια νέα υπηρεσία για τη δυναμική δρομολόγηση μηνυμάτων ηλεκτρονικού ταχυδρομείου η οποία στηρίζεται σε μια προγραμματιζόμενη πλατφόρμα που αποτελεί μέρος του δικτυακού κορμού. Η υπηρεσία αυτή λύνει τα προβλήματα τριγωνικής δρομολόγησης που παρουσιάζονται κατά την αποστολή μηνυμάτων προς συσκευές με δυνατότητα περιαγωγής (βλ. Mobile IP [51]). Επιπλέον, επιτρέπει στους παραλήπτες των μηνυμάτων να ορίσουν μια σειρά φίλτρων τα οποία θα εφαρμόζονται στην εισερχόμενη αλληλογραφία όταν αυτή διέρχεται από ενεργούς κόμβους. Μέσω αυτών των φίλτρων, το περιεχόμενο της αλληλογραφίας μπορεί να μετασχηματιστεί σε μορφές πιο κατάλληλες για τις συσκευές των χρηστών. Επίσης, μέσω των φίλτρων μπορεί να μειωθεί σημαντικά η δικτυακή κίνηση που οφείλεται σε ανεπιθύμητη αλληλογραφία.

Αρχικά, θα γίνει μια πρώτη ανάλυση του προβλήματος της τριγωνικής δρομολόγησης στην ηλεκτρονική αλληλογραφία και θα παρουσιαστεί η βασική αρχιτεκτονική της πλατ-

φόρμας στην οποία στηρίζεται η προτεινόμενη υπηρεσία. Στη συνέχεια, θα παρουσιαστούν μέθοδοι που επιτρέπουν στις προγραμματιζόμενες υποδομές του κορμού ενός δικτύου να ρυθμίζονται και να επαναπρογραμματίζονται με λογισμικό και δεδομένα από έμπιστες πηγές. Το κεφάλαιο ολοκληρώνεται με την παρουσίαση της πρότυπης υλοποίησης της προτεινόμενης υπηρεσίας και την αξιολόγηση αυτής.

2.1 Υπηρεσία δυναμικής δρομολόγησης μηνυμάτων ηλεκτρονικού ταχυδρομείου

Η υπηρεσία δυναμικής δρομολόγησης μηνυμάτων ηλ. ταχυδρομείου ή αλλιώς "Mobile Active Mail" [55] που προτείνεται στο κεφάλαιο αυτό, επιτρέπει τη δυναμική δρομολόγηση των μηνυμάτων ενός χρήστη στο δίκτυο υποδοχής όπου συνδέεται χωρίς να απαιτείται η προηγούμενη δρομολόγηση αυτών σε κάποιο κεντρικό εξυπηρετητή. Η δυναμική δρομολόγηση των μηνυμάτων γίνεται με τρόπο διαφανή για το χρήστη και συνεργάζεται με τεχνολογίες όπως η Mobile IP [51], οι οποίες επιτρέπουν στο χρήστη να διατηρήσει την ίδια δικτυακή ταυτότητα (διεύθυνση δικτύου) σε όποιο δίκτυο υποδοχής και αν συνδεθεί. Επιπρόσθετα, δίνει τη δυνατότητα στο χρήστη να ορίσει μια σειρά φίλτρων τα οποία θα εφαρμοστούν στα μηνύματα προτού αυτά φτάσουν στον τελικό τους προορισμό.

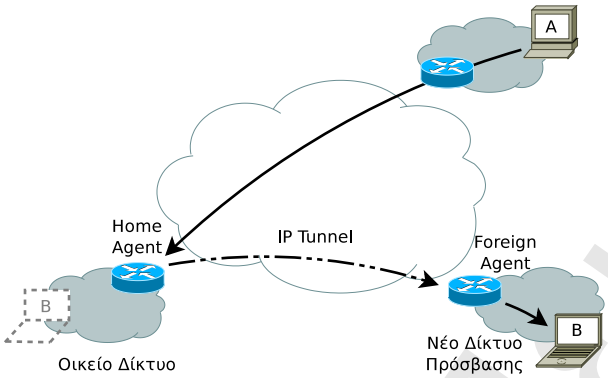


Σχήμα 2.1: Μεταφορά μηνύματος μέσω ηλ. ταχυδρομείου στο Διαδίκτυο

Στο σχήμα 2.1 παρουσιάζεται ο τρόπος με τον οποίο γίνεται σήμερα η αποστολή μηνυμάτων ηλεκτρονικού ταχυδρομείου. Έστω ότι ο χρήστης A του σχήματος με ηλ. διεύθυνση "user1@somewhere.com" επιθυμεί να αποστείλει ένα μήνυμα στο χρήστη B με ηλ. διεύθυνση "user1@example.com". Το λογισμικό υπεύθυνο για τη συγγραφή του μηνύματος (mail user agent) θα συνδεθεί στον τοπικό διακομιστή υπηρεσιών SMTP [56], ο οποίος είναι υπεύθυνος για την αποστολή μηνυμάτων (mail transfer agent) των χρηστών της διαχειριστικής ζώνης "somewhere.com". Ο διακομιστής αυτός (που παρουσιάζεται στο σχήμα 2.1 με το διακριτικό τίτλο "somewhere.com" MX) θα αναζητήσει μέσω της διαδικτυακής υπηρεσίας DNS [57] τον υπεύθυνο διακομιστή SMTP για τη διαχειριστική ζώνη "example.com". Μόλις βρεθεί η διεύθυνση αυτού, θα συνδεθεί σε αυτόν και θα μεταφέρει το μήνυμα του χρήστη A. Ο χρήστης B θα παραλάβει το μήνυμα από τον (συνήθως τοπικό) διακομιστή SMTP μέσω υπηρεσίας IMAP [58] ή POP [59]. Η παραλαβή αυτή μπορεί να γίνει μέσω λογισμικού τύπου "mail user agent" όπως αυτό που χρησιμοποίησε ο χρήστης A για να συντάξει το μήνυμα.

Στην περίπτωση όπου ο χρήστης B επιθυμεί να μετακινηθεί σε οποιοδήποτε άλλο δίκτυο πέραν του οικείου δικτύου, το λογισμικό τύπου "mail user agent" θα πρέπει να συνδεθεί

στον (απομακρυσμένο πλέον) διακομιστή POP/IMAP του οικείου δικτύου προκειμένου να παραλάβει τα νέα μηνύματα. Μια τέτοια ενέργεια θα δημιουργήσει κίνηση που θα επιβαρύνει το νέο δίκτυο πρόσβασης, το οικείο δίκτυο αλλά και τους ενδιάμεσους κόμβους που συνδέουν τα προηγούμενα δύο δίκτυα.



Σχήμα 2.2: Δρομολόγηση κίνησης μέσω Mobile IP

Δυστυχώς η επιβάρυνση αυτή θα υπάρχει και στην περίπτωση όπου ο χρήστης θα χρησιμοποιήσει την τεχνολογία Mobile IP [51] προκειμένου να διατηρήσει την ίδια δικτυακή ταυτότητα σε όποιο νέο δίκτυο πρόσβασης συνδεθεί. Συγκεκριμένα, στην τεχνολογία Mobile IP, δημιουργείται μια εικονική ζεύξη (IP tunnel) μεταξύ του οικείου δικτύου και του νέου δικτύου πρόσβασης ώστε η κίνηση που προοριζόταν για τη διεύθυνση δικτύου του χρήστη στο οικείο δίκτυο να μπορεί να δρομολογηθεί σε κάθε νέο δίκτυο πρόσβασης στο οποίο εκείνος συνδέεται. Για τη δημιουργία της εικονικής ζεύξης θα πρέπει να συνεργαστούν δύο πράκτορες (agents), ο "Foreign Agent" και ο "Home Agent", οι οποίοι βρίσκονται εγκατεστημένοι στους δρομολογητές των δύο δικτύων.

Ο πράκτορας Foreign Agent εκτελείται σε δρομολογητή του νέου δικτύου πρόσβασης και παρέχει πληροφορίες για κινητούς κόμβους που έχουν συνδεθεί στο δίκτυο αυτό. Η πιο σημαντική από αυτές τις πληροφορίες είναι η λεγόμενη "Care-of Address" η οποία αποτελεί τη διεύθυνση στην οποία θα τερματιστεί η εικονική ζεύξη για ένα κόμβο. Η διεύθυνση αυτή μπορεί να είναι η διεύθυνση του ίδιου του κινητού κόμβου ή η διεύθυνση του δρομολογητή στον οποίο εκτελείται ο Foreign Agent. Στη δεύτερη περίπτωση ο δρομολογητής θα αναλάβει να αφαιρέσει την ενθυλάκωση και να δρομολογήσει την κίνηση στον κινητό κόμβο.

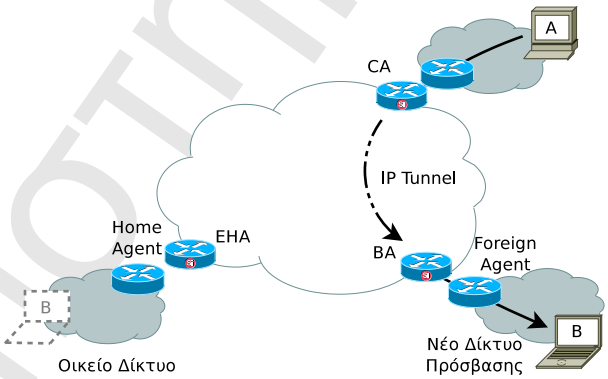
Ο πράκτορας Home Agent εκτελείται σε δρομολογητή του οικείου δικτύου και έχει ως στόχο την ανακατεύθυνση της κίνησης που προοριζόταν για τον κινητό κόμβο στο οικείο δίκτυο, στη νέα διεύθυνση αυτού στο δίκτυο υποδοχής. Με την εγγραφή ενός κινητού κόμβου σε ένα νέο δίκτυο πρόσβασης, ο Home Agent ενημερώνεται (είτε από τον ίδιο τον κόμβο είτε από τον Foreign Agent) για την τρέχουσα διεύθυνση Care-of Address που αντιστοιχεί στον κινητό κόμβο. Ο Home Agent στη συνέχεια θα δημιουργήσει μια εικονική ζεύξη τύπου IP tunnel με τον κόμβο που βρίσκεται στη διεύθυνση Care-of Address ώστε να προωθήσει μέσω αυτής όλη την κίνηση που αρχικά προοριζόταν για τον κινητό κόμβο στο οικείο δίκτυο.

Ένα παράδειγμα αυτής της τεχνολογίας παρουσιάζεται στο σχήμα 2.2. Ο κόμβος A επιθυμεί να επικοινωνήσει με τον κινητό κόμβο B χρησιμοποιώντας τη διεύθυνση που είχε ο δεύτερος στο οικείο δίκτυο. Τα πακέτα με προορισμό τον κόμβο B που θα φτάσουν στο οικείο δίκτυο θα συλλεχθούν από τον Home Agent, ο οποίος και θα τα προωθήσει στην κατάλληλη διεύθυνση Care-of Address που έμαθε μέσω του Foreign Agent. Μόλις τα πακέτα φτάσουν στο δρομολογητή με τη διεύθυνση Care-of Address, θα αφαιρεθεί η ενθυλάκωση αυτών και θα δρομολογηθούν στον κόμβο B. Στο σημείο αυτό θα πρέπει να σημειωθεί

ότι στον κόμβο B θα πρέπει να έχουν γίνει οι απαραίτητες ρυθμίσεις ώστε να μπορεί να παραλαμβάνει πακέτα που προορίζονται στη διεύθυνση δικτύου που εκείνος είχε στο οικείο δίκτυο.

Ο τρόπος με τον οποίο ο κόμβος B αποστέλλει πακέτα στον κόμβο Α ως απάντηση σε πακέτα που έλαβε στη διεύθυνσή του στο οικείο δίκτυο, εξαρτάται από τις δυνατότητες του δρομολογητή στο νέο δίκτυο πρόσβασης. Αν ο δρομολογητής επιτρέπει την προώθηση πακέτων προς το Διαδίκτυο με διεύθυνση αποστολέα που δεν ανήκει στο τοπικό δίκτυο, τότε ο κόμβος B μπορεί να απαντήσει *άμεσα* στον κόμβο A χρησιμοποιώντας ως διεύθυνση αποστολέα τη διεύθυνση δικτύου που είχε στο οικείο δίκτυο. Αν πάλι ο δρομολογητής δεν έχει αυτή τη δυνατότητα, τότε η εικονική ζεύξη θα πρέπει να καταλήγει στον κόμβο B ώστε τα πακέτα της απάντησης να φτάσουν στο οικείο δίκτυο μέσω αυτής και από εκεί στον κόμβο A.

Για την αποφυγή της *τριγωνικής δρομολόγησης* (triangular routing) μεταξύ του δικτύου του κόμβου A, του οικείου δικτύου του κόμβου B και του νέου δικτύου πρόσβασης του κόμβου B, μπορεί να χρησιμοποιηθεί η προγραμματιζόμενη δικτυακή πλατφόρμα ονόματι "Mobile Active Overlay" της εργασίας [60]. Στην πλατφόρμα αυτή εγκαθίσταται ένας Ενεργός Δρομολογητής σε κάθε δίκτυο που πρόκειται να συμμετέχει σε επικοινωνία τύπου Mobile IP. Ο δρομολογητής θα πρέπει να παρέχει τρεις υπηρεσίες οι οποίες λειτουργούν συνεργατικά με τους πράκτορες της τεχνολογίας Mobile IP. Ο πρώτος τύπος υπηρεσίας ονομάζεται "Extended Home Agent", παρέχεται από Ενεργό Δρομολογητή στο οικείο δίκτυο του χρήστη και ενημερώνει το Ενεργό Δίκτυο για τη νέα τοποθεσία του χρήστη. Ο δεύτερος τύπος υπηρεσίας ονομάζεται "Bridgehead Agent", παρέχεται από Ενεργό Δρομολογητή στο νέο δίκτυο πρόσβασης και διαμορφώνει την πρόσβαση του χρήστη στο Διαδίκτυο. Ο τρίτος και τελευταίος τύπος υπηρεσίας ονομάζεται "Correspondent Agent" και παρέχεται από Ενεργό Δρομολογητή ενός τρίτου δικτύου με το οποίο επικοινωνεί ο χρήστης. Η υπηρεσία αυτή αναλαμβάνει να συμβουλευθεί τον Extended Home Agent ώστε κάθε επικοινωνία με τον κινητό κόμβο να γίνεται μέσω του Bridgehead Agent που βρίσκεται στο δίκτυο υποδοχής αυτού.



Σχήμα 2.3: Δρομολόγηση κίνησης μέσω της πλατφόρμας Mobile Active Overlay

Στο σχήμα 2.3 παρουσιάζεται ένα παράδειγμα χρήσης της πλατφόρμας Mobile Active Overlay. Ο χρήστης A επικοινωνεί με το χρήστη Β στη νέα του τοποθεσία δίχως να απαιτείται η προηγούμενη δρομολόγηση πακέτων μέσω του οικείου δικτύου του B. Για να γίνει αυτή η άμεση επικοινωνία, η υπηρεσία Extended Home Agent (EHA) έχει πρωτύτερα καταγράψει (μέσω του Home Agent) τη διεύθυνση του Bridgehead Agent (BA) που είναι υπεύθυνος για τη νέα τοποθεσία του χρήστη B. Η διεύθυνση του Bridgehead Agent γίνεται πλέον μια διεύθυνση τύπου Care-of Address. Ο Correspondent Agent (CA) έχοντας ενημε-

ρωθεί για αυτή τη διεύθυνση από τον Extended Home Agent θα δημιουργήσει μια ειδική δικτυακή ζεύξη (IP tunnel) με τον Bridgehead Agent και θα προωθήσει μέσω αυτής την κίνηση που προορίζεται για τον κόμβο B στο νέο δίκτυο πρόσβασης.

Από το παράδειγμα του σχήματος 2.3 μπορεί κανείς εύκολα να διαπιστώσει ότι στην περίπτωση όπου ο χρήστης A αποστείλει μέσω ηλεκτρονικού ταχυδρομείου ένα μήνυμα στο χρήστη B, η επικοινωνία με το οικείο δίκτυο του χρήστη B θα είναι αναπόφευκτη καθώς:

α) το μήνυμα του χρήστη A θα πρέπει να παραδοθεί στο διακομιστή SMTP στο οικείο δίκτυο του B, και

β) ο χρήστης B θα πρέπει να παραλάβει το μήνυμα από το δίκτυο αυτό.

Η υπηρεσία Mobile Active Mail έρχεται να λύσει το παραπάνω πρόβλημα τριγωνικής δρομολόγησης (σε επίπεδο εφαρμογής) προσθέτοντας στην αρχιτεκτονική Mobile Active Overlay ενεργές υπηρεσίες για την προώθηση κίνησης ηλ. ταχυδρομείου στο δίκτυο υποδοχής του χρήστη. Επίσης, δίνει τη δυνατότητα στο χρήστη να ορίσει μια σειρά φίλτρων τα οποία θα εφαρμοστούν στα ηλ. μηνύματα προτού αυτά φτάσουν στον τελικό τους προορισμό. Με τον τρόπο αυτό μπορεί να μειωθεί περαιτέρω η κίνηση που θα φτάσει στο δίκτυο υποδοχής, καθώς τα φίλτρα του χρήστη μπορούν να εμποδίσουν από πολύ νωρίς (π.χ. στο δίκτυο του αποστολέα) τη μεταφορά ανεπιθύμητων μηνυμάτων (π.χ. διαφημιστικά μηνύματα τύπου spam).

Στις ενότητες που ακολουθούν περιγράφεται η αρχιτεκτονική της προτεινόμενης πλατφόρμας Mobile Active Mail και παρουσιάζονται τα πιο σημαντικά σημεία της υλοποίησης αυτής.

2.2 Πλατφόρμα Mobile Active Mail

Η πλατφόρμα Mobile Active Mail αποτελεί ένα τυπικό παράδειγμα προγραμματιζόμενης υποδομής που υλοποιείται στο δικτυακό κορμό. Για την υλοποίηση αυτής απαιτούνται δύο τύποι κόμβων, ο Ενεργός Δρομολογητής (Active Router) και ο Διακομιστής Ενεργών Υπηρεσιών (Active Server). Ο Ενεργός Δρομολογητής επεξεργάζεται δεδομένα σε επίπεδο δικτύου και δημιουργεί τις απαραίτητες ζεύξεις ώστε να προωθηθούν συγκεκριμένοι τύποι κίνησης σε άλλους Ενεργούς Δρομολογητές ή Διακομιστές. Ο Διακομιστής Ενεργών Υπηρεσιών λειτουργεί σε επίπεδο εφαρμογής και προσφέρει με δυναμικό τρόπο υπηρεσίες στους χρήστες του δικτύου.

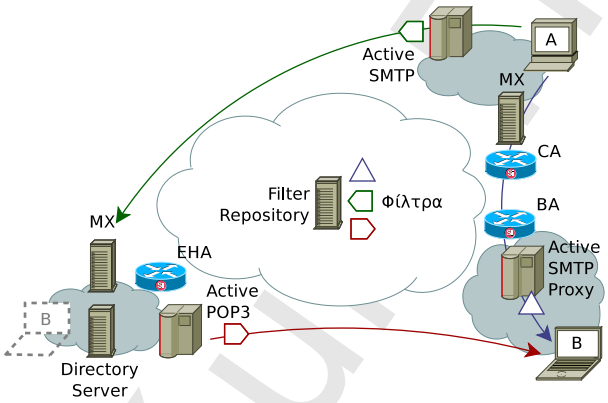
Η πλατφόρμα Mobile Active Mail προσφέρει τρεις βασικές υπηρεσίες. Η πρώτη από αυτές είναι η προώθηση μηνυμάτων ηλ. ταχυδρομείου στον προσωπικό υπολογιστή του χρήστη. Η προώθηση αυτή γίνεται μέσω της υποδομής Mobile Active Overlay και της υπηρεσίας Active SMTP Proxy, η οποία παρέχεται από Διακομιστή Ενεργών Υπηρεσιών του δικτύου υποδοχής.

Η δεύτερη υπηρεσία αφορά στην επεξεργασία μηνυμάτων ηλ. ταχυδρομείου προτού αυτά φτάσουν στο διακομιστή SMTP του παραλήπτη. Η υπηρεσία αυτή υλοποιείται από το λογισμικό Active SMTP το οποίο εκτελείται σε Διακομιστή Ενεργών Υπηρεσιών στο δίκτυο του αποστολέα. Μέσω της υπηρεσίας αυτής γίνεται δυνατός ο έλεγχος ηλ. μηνυμάτων (για κακόβουλο λογισμικό, ανεπιθύμητη αλληλογραφία κ.α) όταν τα μηνύματα αυτά βρίσκονται ακόμη στο δίκτυο του αποστολέα.

Η τρίτη και τελευταία υπηρεσία αφορά στην επεξεργασία μηνυμάτων κατά τη μεταφόρτωση αυτών από το διακομιστή POP3 του οικείου δικτύου του χρήστη. Η υπηρεσία αυτή ονομάζεται Active POP3 και είναι χρήσιμη σε περιπτώσεις όπου δεν είναι εφικτή η προώθηση μηνυμάτων απευθείας στο σταθμό εργασίας του χρήστη. Στις περιπτώσεις αυτές τα

μηνύματα παραλαμβάνονται από το διακομιστή SMTP του οικείου δικτύου και συλλέγονται από το χρήστη μέσω υπηρεσίας POP3. Η επεξεργασία των μηνυμάτων μπορεί σε αυτές τις περιπτώσεις να ελαττώσει σημαντικά τον όγκο των δεδομένων που θα μεταφερθούν από το οικείο δίκτυο στο δίκτυο υποδοχής.

Ένας χρήστης μπορεί να συμμετάσχει στην πλατφόρμα Mobile Active Mail ακόμη και όταν το δίκτυο υποδοχής στο οποίο συνδέεται δεν έχει κάποιο εγκατεστημένο Ενεργό Κόμβο. Σε αυτές τις περιπτώσεις ο Κινητός Κόμβος μπορεί να αξιοποιήσει μια Υπηρεσία Πληροφοριών Καταλόγου (Directory Server) του οικείου δικτύου (ή οποιουδήποτε άλλου δικτύου) ώστε να εντοπίσει ένα Διακομιστή Ενεργών Υπηρεσιών, κατά προτίμηση σε μικρή δικτυακή απόσταση από την παρούσα του θέση. Η Υπηρεσία Πληροφοριών Καταλόγου του έργου CASPIAN εντοπίζει τον κατάλληλο Διακομιστή για λογαριασμό ενός κινητού κόμβου συγκρίνοντας τα δικτυακά μονοπάτια που καταλήγουν στους Διακομιστές που περιέχονται στη βάση αυτής, με το δικτυακό μονοπάτι που καταλήγει στον Κινητό Κόμβο. Σε περιπτώσεις όπου είναι απαραίτητη η εύρεση του κοντινότερου (δικτυακά) Διακομιστή μπορεί επίσης να χρησιμοποιηθεί η τεχνολογία GNP [61] (Global Network Positioning) για τον ακριβέστερο υπολογισμό δικτυακών αποστάσεων.



Σχήμα 2.4: Η πλατφόρμα Mobile Active Mail

Στο σχήμα 2.4 παρουσιάζονται σε υψηλό επίπεδο τα κύρια συστατικά της πλατφόρμας Mobile Active Mail που περιγράφηκε παραπάνω. Μια πρότυπη υλοποίηση της πλατφόρμας δημιουργήθηκε στα πλαίσια του ερευνητικού προγράμματος CASPIAN (P926) του Ευρωπαϊκού φορέα Eurescom [62]. Για τις ανάγκες αυτής της υλοποίησης αναπτύχθηκε το παρακάτω λογισμικό:

- Λογισμικό Διακομιστή Ενεργών Υπηρεσιών (Active Server)
- Λογισμικό Ενεργού Δρομολογητή (Active Router)
- Ενεργές υπηρεσίες ηλ. ταχυδρομείου (Active SMTP Proxy, Active SMTP, Active POP3)
- Υπηρεσία πληροφοριών καταλόγου (Directory Service)
- Υπηρεσία διάθεσης φίλτρων (Filter Repository)
- Υπηρεσία ασφαλούς διάθεσης ρυθμίσεων χρηστών (Secure Configuration Delivery Service)

Οι ενότητες που ακολουθούν παρουσιάζουν πιο αναλυτικά το παραπάνω λογισμικό και τις τεχνικές που αυτό χρησιμοποιεί για να λύσει συγκεκριμένα προβλήματα ασφάλειας και απόδοσης.

2.3 Αρχιτεκτονική Ενεργών Κόμβων

Οι Ενεργοί Κόμβοι του έργου CASPIAN [62] είναι βασισμένοι στον πυρήνα Linux και στην τεχνολογία διερμηνέα / εικονικής μηχανής της γλώσσας προγραμματισμού Java (Java bytecode interpreter / Virtual Machine) [2]. Ένας κόμβος, στα πλαίσια της παραπάνω αρχιτεκτονικής, μπορεί να παίξει ταυτόχρονα τόσο το ρόλο του Ενεργού Δρομολογητή όσο και το ρόλο του Διακομιστή Ενεργών Υπηρεσιών.

2.3.1 Ενεργός Δρομολογητής

Ο Ενεργός Δρομολογητής που συναντάται στην πλατφόρμα Mobile Active Mail είναι υπεύθυνος για τη συλλογή δικτυακής κίνησης συγκεκριμένου τύπου και την προώθηση αυτής σε Διακομιστές Ενεργών Υπηρεσιών. Ο Ενεργός Δρομολογητής τοποθετείται στα άκρα δικτύων (αλλά και στο διαδικτυακό κορμό) προκειμένου να είναι αυξημένη η πιθανότητα να δρομολογηθεί από αυτόν κίνηση που προορίζεται για τον κινητό κόμβο.

Στην περίπτωση της πλατφόρμας Mobile Active Overlay οι Ενεργοί Δρομολογητές συλλέγουν την όποια κίνηση προορίζεται για το οικείο δίκτυο, την ενθυλακώνουν σε πακέτα UDP και την προωθούν σε Διακομιστή Ενεργών Υπηρεσιών στο δίκτυο υποδοχής του χρήστη. Εκεί αφαιρείται η παραπάνω ενθυλάκωση και η κίνηση προωθείται στο σταθμό εργασίας του χρήστη. Η διεύθυνση του Διακομιστή στον οποίο προωθείται η κίνηση λαμβάνεται δυναμικά από την υπηρεσία Extended Home Agent του οικείου δικτύου. Η ενθυλάκωση της κίνησης σε πακέτα UDP επιτρέπει μεταξύ άλλων την εισαγωγή νέας πληροφορίας σε αυτά, όπως για παράδειγμα το χαρακτηρισμό του τύπου της δικτυακής κίνησης ή την κωδική ονομασία των υπηρεσιών που θα ήταν κατάλληλες για να επεξεργαστούν τα δεδομένα της κίνησης στο Διακομιστή Ενεργών Υπηρεσιών.

Ο Ενεργός Δρομολογητής που χρησιμοποιείται στην πλατφόρμα Mobile Active Mail διαφέρει από τον παραπάνω Δρομολογητή κυρίως στον τρόπο με τον οποίο προωθεί την σχετική κίνηση στις υπηρεσίες ηλ. ταχυδρομείου του Διακομιστή Ενεργών Υπηρεσιών. Συγκεκριμένα, απαιτείται η προώθηση κίνησης τύπου TCP (κίνηση ηλεκτρονικού ταχυδρομείου) προς υπηρεσίες του Διακομιστή οι οποίες επικοινωνούν μέσω sockets. Για να μπορέσει το socket μιας υπηρεσίας να λάβει την ωφέλιμη πληροφορία από τα πακέτα της κίνησης, θα πρέπει να έχει προηγηθεί επεξεργασία των πακέτων αυτών από μια στοίβα πρωτοκόλλων TCP/IP. Για το λόγο αυτό, ο Ενεργός Δρομολογητής της πλατφόρμας Mobile Active Mail, αντί να ενθυλακώσει και να προωθήσει την κίνηση TCP σε μια υπηρεσία του Διακομιστή, την προωθεί δίχως ενθυλάκωση στη στοίβα του λειτουργικού συστήματος που φιλοξενεί το Διακομιστή. Εκεί, γίνεται η απαραίτητη επεξεργασία της κίνησης και τα δεδομένα αυτής μεταφέρονται (από το λειτουργικό σύστημα πλέον) στο socket της αντίστοιχης υπηρεσίας του Διακομιστή[1]. Θα πρέπει να σημειωθεί εδώ ότι το κανάλι επικοινωνίας που δημιουργείται μεταξύ Ενεργού Δρομολογητή και Διακομιστή Ενεργών Υπηρεσιών είναι κανάλι αμφίδρομης επικοινωνίας.

Ένας Ενεργός Δρομολογητής θα πρέπει να ενημερώνεται για το είδος της κίνησης που θα συλλέγει, το είδος των υπηρεσιών που θα αναλάβουν την επεξεργασία αυτής της κίνησης αλλά και τη δικτυακή διεύθυνση των κατάλληλων Διακομιστών που παρέχουν αυτές τις υπηρεσίες. Όπως αναφέρθηκε παραπάνω, στην περίπτωση της πλατφόρμας Mobile Active Overlay ο Δρομολογητής ενημερώνεται από τον Extended Home Agent του δικτύου όπου

[1] Σε μια εναλλακτική υλοποίηση, το λογισμικό του Διακομιστή Ενεργών Υπηρεσιών θα μπορούσε να αναλαμβάνει την αφαίρεση της ενθυλάκωσης UDP και την παράδοση των πακέτων στη στοίβα TCP/IP του λειτουργικού συστήματος μέσω διεπαφής τύπου tun/tap [63].

προορίζονται τα διερχόμενα πακέτα για τις προτιμήσεις των κινητών κόμβων που έχουν το δίκτυο αυτό ως οικείο δίκτυο. Στις προτιμήσεις αυτές συγκαταλέγονται η διεύθυνση του Διακομιστή στο νέο δίκτυο πρόσβασης, το είδος της κίνησης που θα πρέπει να προωθείται στο νέο δίκτυο πρόσβασης αλλά και το είδος της επεξεργασίας που θα πρέπει να γίνει στα πακέτα εκεί. Πριν ξεκινήσει την προώθηση, ο Δρομολογητής θα ερωτήσει το Διακομιστή για τη δυνατότητα επεξεργασίας των πακέτων αυτών. Ο Διακομιστής θα απαντήσει θετικά για όποια σχετική υπηρεσία παρέχει ήδη ή μπορεί να φορτώσει δυναμικά. Σε περίπτωση όπου ο ενδεδειγμένος Διακομιστής του νέου δικτύου πρόσβασης δεν παρέχει κάποια από τις ζητούμενες υπηρεσίες, θα γίνει σχετικό ερώτημα στον Bridgehead Agent του ίδιου δικτύου προκειμένου να εντοπιστεί κατάλληλος Διακομιστής.

Η συλλογή της κίνησης γίνεται μέσω της διεπαφής "netfilter" [64] που παρέχει ο πυρήνας του λειτουργικού συστήματος Linux. Η διεπαφή αυτή επιτρέπει τη συλλογή και επεξεργασία πακέτων με συγκεκριμένα χαρακτηριστικά και την προώθηση αυτών σε τοπικές διεργασίες αλλά και σε διεργασίες που εκτελούνται σε απομακρυσμένα συστήματα. Επίσης, επιτρέπει τη διαμόρφωση κανόνων που λειτουργούν ως δικτυακό τείχος προστασίας (firewall) για τις εφαρμογές ενός συστήματος.

Αρχικά ο Ενεργός Δρομολογητής χρησιμοποιεί την παραπάνω διεπαφή για να προδιαγράψει το είδος της κίνησης που επιθυμεί να συλλέξει. Η προδιαγραφή γίνεται μέσω ενός εικονικού αρχείου (`/proc/Caspian/config`) το οποίο τηρείται από σχετικό άρθρωμα του πυρήνα (Active Router kernel module). Η εφαρμογή του Ενεργού Δρομολογητή έχει δικαίωμα ανάγνωσης και εγγραφής στο αρχείο αυτό και μπορεί έτσι να λάβει τη λίστα των κανόνων που ορίζουν ποια πακέτα θα συλλεχθούν καθώς και να προσθέσει/αφαιρέσει κανόνες στη λίστα αυτή. Κάθε εγγραφή του αρχείου περιγράφει τις επιθυμητές τιμές που θα πρέπει να έχουν τα πακέτα ως προς το πρωτόκολλο (IP, TCP, UDP κλπ.), ως προς τη διεύθυνση IPv4 αποστολέα/παραλήπτη (ή το εύρος διευθύνσεων για μάσκα διαφορετική από /32) αλλά και ως προς τις πόρτες (ports) των εφαρμογών. Στην αρχή κάθε εγγραφής τοποθετείται το σύμβολο '+' ή '-' για να δηλωθεί η πρόσθεση ή η αφαίρεση ενός κανόνα. Παρακάτω παρουσιάζεται παράδειγμα μιας τέτοιας εγγραφής:

```
+6:0.0.0.0/0:0 3.3.3.3/32:25
```

Σχήμα 2.5: Παράδειγμα εγγραφής αρχείου /proc/Caspian/config

Στο παράδειγμα ζητείται η εισαγωγή ενός νέου κανόνα για τη συλλογή πακέτων που ανήκουν στο πρωτόκολλο TCP ("6"). Τα πακέτα αυτά μπορεί να προέρχονται από οποιοδήποτε αποστολέα (διεύθυνση IPv4 "0.0.0.0" και μάσκα "/0") και οποιαδήποτε πόρτα εφαρμογής (πόρτα "0"). Ως παραλήπτης των πακέτων θα πρέπει να παρουσιάζεται η υπηρεσία SMTP (πόρτα "25") του κόμβου (βλ. μάσκα "/32") με διεύθυνση IPv4 "3.3.3.3".

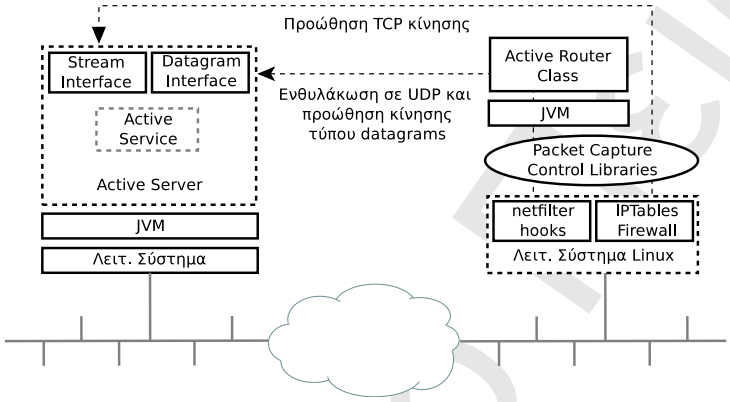
Τα πακέτα που συλλέγονται από το άρθρωμα του πυρήνα προωθούνται μέσω ενός socket τύπου "netlink" στο λογισμικό του Ενεργού Δρομολογητή. Εκεί λαμβάνουν διαφορετική επεξεργασία ανάλογα με το πρωτόκολλο στο οποίο ανήκουν:

- Αν πρόκειται για κίνηση τύπου TCP, προστίθεται ένας νέος κανόνας στο firewall του Δρομολογητή ώστε κάθε σχετικό πακέτο να προωθείται απευθείας στην αντίστοιχη υπηρεσία του κατάλληλου Διακομιστή (και όχι στο λογισμικό του Δρομολογητή). Το λογισμικό (του πελάτη) που είχε δημιουργήσει την κίνηση θα βρεθεί πλέον σε κατάσταση time-out (εφόσον δεν έχει λάβει απάντηση για το αρχικό πακέτο) και έτσι θα αποστείλει εκ νέου το πρώτο πακέτο της συνεδρίας. Το πακέτο αυτό καθώς και τα επόμενα πακέτα της συνεδρίας θα προωθηθούν στην αντίστοιχη υπηρεσία του επι-

λεγμένου Διακομιστή. Με το πέρας της συνεδρίας ο σχετικός κανόνας αφαιρείται από τον πυρήνα και το firewall του Δρομολογητή επανέρχεται στην αρχική του κατάσταση.

- Αν η κίνηση αποτελείται από πακέτα IP ή είναι τύπου UDP (δηλαδή είναι κίνηση τύπου αυτοδύναμων πακέτων), τότε αυτή ενθυλακώνεται σε πακέτα UDP και αποστέλλεται στον κατάλληλο Διακομιστή. Εκεί αφαιρείται η ενθυλάκωση και γίνεται η παράδοση στην αντίστοιχη υπηρεσία.

Στο σχήμα 2.6 παρουσιάζεται συνοπτικά η αρχιτεκτονική ενός Ενεργού Δρομολογητή της πλατφόρμας Mobile Active Mail καθώς και η μέθοδος προώθησης πακέτων προς ένα Διακομιστή Ενεργών Υπηρεσιών.



Σχήμα 2.6: Προώθηση πακέτων από Ενεργό Δρομολογητή της πλατφόρμας Mobile Active Mail

Η διαδικασία πρόσθεσης και αφαίρεσης κανόνων για την προώθηση πακέτων προς Διακομιστές στο Διαδίκτυο υλοποιείται μέσω μιας βιβλιοθήκης σε γλώσσα C ονόματι "iptables active router interface". Η βιβλιοθήκη αυτή συνεργάζεται με το ήδη υπάρχον σύστημα διαχείρισης κανόνων firewall του πυρήνα ("iptables" [64]) ώστε ο διαχειριστής να μπορεί ανά πάσα στιγμή να έχει συνολική εποπτεία των κανόνων που συνθέτουν το firewall του Δρομολογητή. Σε χαμηλό επίπεδο, οι κλήσεις που γίνονται από αυτή τη βιβλιοθήκη χρησιμοποιούν για άλλη μια φορά τη διεπαφή netfilter του πυρήνα.

Τέλος, το κυρίως λογισμικό του Ενεργού Δρομολογητή είναι υλοποιημένο σε γλώσσα Java και εκτελείται σε μια εικονική μηχανή τύπου Java Virtual Machine. Η επικοινωνία της εφαρμογής με εξωτερικές βιβλιοθήκες γίνεται μέσω διεπαφής JNI (Java Native Interface) [2]. Οι βιβλιοθήκες αυτές προσθέτουν περαιτέρω δυνατότητες στην εικονική μηχανή, όπως αυτή της δημιουργίας πακέτων και της προώθησης αυτών μέσω των τοπικών δικτυακών διεπαφών (raw socket).

2.3.2 Διακομιστής Ενεργών Υπηρεσιών

Οι υπηρεσίες της πλατφόρμας Mobile Active Mail εκτελούνται δυναμικά σε Διακομιστές Ενεργών Υπηρεσιών. Οι Διακομιστές αυτοί είναι διασκορπισμένοι σε διάφορα δίκτυα που συνήθως δεν υπόκεινται σε κάποια ενιαία διαχείριση. Ο αριθμός των εγκατεστημένων Διακομιστών ανά δίκτυο ποικίλει ανάλογα με τον αριθμό των χρηστών που καλούνται να εξυπηρετήσουν.

Το κυρίως λογισμικό του Διακομιστή είναι γραμμένο σε γλώσσα Java και εκτελείται σε μια εικονική μηχανή τύπου Java Virtual Machine. Στην ίδια μηχανή εκτελούνται και οι υπηρεσίες, οι οποίες έχουν τη μορφή νημάτων (Java Threads) και φορτώνονται δυναμικά όποτε

υπάρχει ζήτηση για αυτές. Όπως στον Ενεργό Δρομολογητή έτσι και εδώ,, οι επεκτάσεις των δυνατοτήτων της εικονικής μηχανής γίνονται μέσω βιβλιοθηκών σε γλώσσα C, οι οποίες επικοινωνούν με την εικονική μηχανή μέσω διεπαφής JNI.

Το κυρίως λογισμικό του Διακομιστή λειτουργεί τόσο συνεργατικά όσο και ρυθμιστικά ως προς τις προσφερόμενες υπηρεσίες. Συγκεκριμένα:

- ελέγχει ότι αυτές υλοποιούν ένα συγκεκριμένο πρότυπο πριν τις εκτελέσει (Active Service Interface),
- διαχειρίζεται την έναρξη, την παύση και τον τερματισμό τους,
- προστατεύει το περιβάλλον εκτέλεσης από αυτές,
- παρέχει σε αυτές βασικές λειτουργίες, όπως τη δημιουργία socket, την αποστολή/παράδοση δικτυακών μηνυμάτων, την τήρηση αρχείων καταγραφής συμβάντων (logs), και
- δημιουργεί ένα σαφώς ορισμένο δίαυλο επικοινωνίας μεταξύ των υπηρεσιών και τις προστατεύει από τη δράση τρίτων κακόβουλων υπηρεσιών.

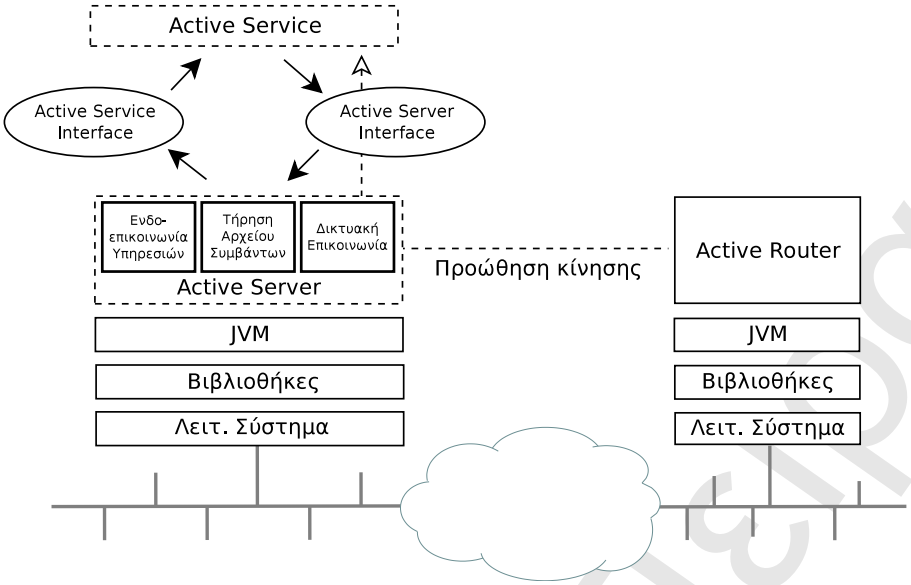
Η δυνατότητα επικοινωνίας μεταξύ των υπηρεσιών είναι ιδιαίτερα χρήσιμη καθώς επιτρέπει τη δημιουργία νέων σύνθετων υπηρεσιών από μικρότερες, πιο απλές υπηρεσίες.

Η προστασία του περιβάλλοντος εκτέλεσης καθώς και των ενεργών υπηρεσιών από τρίτες κακόβουλες υπηρεσίες υλοποιείται σε επίπεδο εικονικής μηχανής και στηρίζεται στα μέτρα προστασίας που προσφέρει η εικονική μηχανή της γλώσσας Java για τα νήματα αυτής. Ο Ενεργός Κόμβος της πλατφόρμας Mobile Active Mail δεν παρέχει άλλες μορφές προστασίας για τις Ενεργές Υπηρεσίες όπως είναι π.χ. η προστασία ενάντια σε επιθέσεις εξάντλησης πόρων (μνήμης, κύκλων επεξεργαστή κλπ.).

Οι Ενεργές Υπηρεσίες επικοινωνούν με το Διακομιστή μέσω συγκεκριμένης διεπαφής (Active Server Interface), η οποία παρέχει σε αυτές πρόσβαση στους απαραίτητους πόρους και τις λειτουργίες του Ενεργού Κόμβου. Χαρακτηριστικό παράδειγμα είναι η πρόσβαση στη δικτυακή κίνηση. Οι υπηρεσίες μέσω της παραπάνω διεπαφής ενημερώνουν το διακομιστή για το είδος της κίνησης που επιθυμούν να επεξεργαστούν. Ο Διακομιστής στη συνέχεια θα δημιουργήσει τα απαραίτητα sockets και θα προωθήσει τη σχετική κίνηση σε αυτές. Το σχήμα 2.7 παρουσιάζει συνοπτικά την αρχιτεκτονική ενός Διακομιστή Ενεργών Υπηρεσιών και τον τρόπο επικοινωνίας αυτού με τις Ενεργές Υπηρεσίες.

Όπως αναφέρθηκε στην προηγούμενη ενότητα, οι Διακομιστές Ενεργών Υπηρεσιών διατηρούν ένα δίαυλο αμφίδρομης επικοινωνίας με Ενεργούς Δρομολογητές, μέσω του οποίου λαμβάνουν δικτυακή κίνηση που προορίζεται για τους κινητούς κόμβους. Εξετάζοντας την ενθυλάκωση των πακέτων αυτής της κίνησης ο Διακομιστής θα ανακαλύψει την υπηρεσία στην οποία θα πρέπει να παραδοθούν τα περιεχόμενα αυτών. Αν η υπηρεσία αυτή είναι ήδη φορτωμένη τότε τα πακέτα θα παραδοθούν άμεσα σε αυτή. Σε αντίθετη περίπτωση, ο Διακομιστής θα συμβουλευτεί ένα τοπικό αρχείο ρυθμίσεων και εάν το επιτρέπουν οι κανόνες ασφάλειας του κόμβου τότε θα φορτώσει δυναμικά τη ζητούμενη υπηρεσία.

Παραδείγματα υπηρεσιών που εκτελούνται στους Ενεργούς Κόμβους του έργου CASPIAN είναι οι Enhanced Home Agent, Correspondent Agent και Bridgehead Agent που βοηθούν κατά την προώθηση κίνησης προς κινητούς κόμβους, καθώς και οι Active SMTP Proxy, Active SMTP και Active POP3 που επιτρέπουν τη δυναμική δρομολόγηση μηνυμάτων ηλ. ταχυδρομείου και την προληπτική εξέταση αυτών σύμφωνα με φίλτρα που ορίζει ο χρήστης. Στην ενότητα που ακολουθεί θα παρουσιαστούν αναλυτικά οι υπηρεσίες αυτές καθώς και ο τρόπος με τον οποίο αυτές αλληλεπιδρούν μεταξύ τους.



Σχήμα 2.7: Αρχιτεκτονική Διακομιστή Ενεργών Υπηρεσιών (Active Server) της πλατφόρμας Mobile Active Mail

2.4 Ενεργές Υπηρεσίες

Οι Ενεργές Υπηρεσίες που συνθέτουν την πλατφόρμα Mobile Active Mail έχουν τη μορφή αρχείων που περιέχουν Java bytecode. Κάθε Διακομιστής, μόλις υπάρξει ζήτηση για μια υπηρεσία, θα εντοπίσει τα κατάλληλα αρχεία και θα τα φορτώσει δυναμικά. Ένα αρχείο ρυθμίσεων ορίζει ποιες υπηρεσίες θα πρέπει να ξεκινήσουν αυτόματα κατά την εκκίνηση του Ενεργού Κόμβου αλλά και ποιες υπηρεσίες επιτρέπεται να φορτωθούν δυναμικά στο μέλλον. Η δυναμική έναρξη μιας υπηρεσίας παρέχεται από λογισμικό τύπου "Class Loader" το οποίο βασίζεται στη δυνατότητα δυναμικής εκτέλεσης bytecode που παρέχει η εικονική μηχανή της Java.

Τα αρχεία των υπηρεσιών βρίσκονται ως επί το πλείστον αποθηκευμένα σε τοπικό χώρο του Διακομιστή. Ωστόσο υπάρχει η δυνατότητα μεταφόρτωσης αρχείων υπηρεσιών και από άλλους Διακομιστές. Επιπρόσθετα, υπάρχει η δυνατότητα μεταφόρτωσης λογισμικού που αφορά μια συγκεκριμένη λειτουργικότητα που απαιτεί μια υπηρεσία. Χαρακτηριστικό παράδειγμα είναι η περίπτωση των φίλτρων που θα πρέπει να εφαρμοστούν στα μηνύματα των κινητών κόμβων.

Στην ενότητα αυτή θα παρουσιαστεί η υλοποίηση των βασικών υπηρεσιών της πλατφόρμας Mobile Active Mail, ξεκινώντας από την περιγραφή της διαδικασίας επεξεργασίας μηνυμάτων μιας και αυτή είναι κοινή τόσο για τις υπηρεσίες Active SMTP και Active SMTP Proxy, όσο και για την υπηρεσία Active POP3.

2.4.1 Διαδικασία επεξεργασίας μηνυμάτων

Η πλατφόρμα Mobile Active Mail παρέχει στο χρήστη ενός κινητού κόμβου τη δυνατότητα να εφαρμόσει μια σειρά φίλτρων στα εισερχόμενα μηνύματα ηλ. ταχυδρομείου όταν αυτά βρίσκονται ακόμη σε διακομιστές απομακρυσμένων δικτύων. Έτσι μπορεί να ελαττώσει την εισερχόμενη κίνηση (προς το νέο δίκτυο πρόσβασης) αλλά και να μετασχηματίσει το περιεχόμενο των μηνυμάτων σε μορφές πιο φιλικές προς το λογισμικό ή το υλικό του κινητού κόμβου.

Η διαδικασία εφαρμογής φίλτρων βασίζεται σε μια αρχιτεκτονική που αποτελείται από 5 συστατικά:

- ένα αρχείο ρυθμίσεων που βρίσκεται τοπικά αποθηκευμένο στον κινητό κόμβο,
- ένα διακομιστή αρχείων ρυθμίσεων που εκτελείται στον κινητό κόμβο,
- ένα σημείο στον κώδικα μιας Ενεργής Υπηρεσίας (π.χ. Active POP3) όπου εφαρμόζονται τα φίλτρα που επέλεξε ο χρήστης (filter hook),
- μια υπηρεσία διάθεσης φίλτρων, και τέλος,
- τα αρχεία που συνθέτουν κάθε φίλτρο (σε μορφή Java bytecode).

Κάθε υπηρεσία παράδοσης/παραλαβής ηλ. μηνυμάτων της πλατφόρμας Mobile Active Mail προτού αποστείλει ή μεταφέρει κάποιο μήνυμα, ελέγχει αν για τον παραλήπτη αυτού του μηνύματος θα πρέπει να εφαρμοστεί κάποιο φίλτρο. Ανάλογα με την τιμή που θα επιστρέψει κάθε φίλτρο, το μήνυμα μπορεί να συνεχίσει την πορεία του προς τον παραλήπτη (ίδιο ή μετασχηματισμένο), να αφαιρεθεί από την ουρά μηνυμάτων ή να επιστρέψει στον αποστολέα συνοδευόμενο από κάποιο μήνυμα σφάλματος, όταν αυτό είναι εφικτό (υπηρεσίες τύπου SMTP).

Η υπηρεσία που επεξεργάζεται τα μηνύματα για λογαριασμό ενός κινητού κόμβου, μαθαίνει τις προτιμήσεις του χρήστη μέσω ενός αρχείου ρυθμίσεων. Το αρχείο αυτό είναι σε μορφή XML [65] και βρίσκεται αποθηκευμένο στον κινητό κόμβο[2]. Στην παρούσα υλοποίηση για κάθε υπηρεσία υπάρχει και ξεχωριστό αρχείο ρυθμίσεων. Ωστόσο η XML δομή του αρχείου επιτρέπει την μελλοντική ένταξη όλων των προτιμήσεων του χρήστη σε ένα κοινό αρχείο ρυθμίσεων.

Η μεταφορά του αρχείου ρυθμίσεων από τον κινητό κόμβο στο Διακομιστή Ενεργών Υπηρεσιών μπορεί να γίνει με δύο τρόπους. Είτε μέσω HTTP, όταν ο κινητός κόμβος έχει εγκατεστημένο κάποιο διακομιστή ιστοσελίδων, είτε μέσω της υπηρεσίας "Secure Configuration Delivery". Και στις δύο περιπτώσεις το αρχείο λαμβάνεται μετά από μια αίτηση της μορφής:

```
GET /~user/pop3config.xml
```

όπου `user` είναι το όνομα χρήστη που παρουσιάζεται στην ηλ. διεύθυνση του παραλήπτη και `pop3config.xml` είναι το αρχείο ρυθμίσεων για την Ενεργή Υπηρεσία Active POP3.

Η υπηρεσία Secure Configuration Delivery επιτρέπει την κρυπτογραφημένη, πιστοποιημένη και ακέραια μεταφορά ενός αρχείου ρυθμίσεων από τον κινητό κόμβο στο Διακομιστή Ενεργών Υπηρεσιών. Η διαδικασία πιστοποίησης βασίζεται στην τεχνολογία PGP [66] και έχει ως εξής:

- Ο διακομιστής αρχείων ρυθμίσεων εξετάζει το πλήρες διαδικτυακό όνομα (Fully Qualified Domain Name) του Διακομιστή Ενεργών Υπηρεσιών που συνδέθηκε σε αυτόν.
- Με βάση το παραπάνω όνομα αναζητά στην παγκόσμια βάση κλειδιών PGP το δημόσιο κλειδί της αντίστοιχης υπηρεσίας (π.χ. `smtp_proxy@quasi.domain.net`).
- Αν βρεθεί το σχετικό κλειδί, τότε ελέγχεται αν αυτό φέρει υπογραφή από μια έμπιστη οντότητα.

[2]Εξαίρεση αποτελεί το αρχείο ρυθμίσεων της υπηρεσίας Active SMTP το οποίο θα πρέπει να βρίσκεται αποθηκευμένο στο διακομιστή SMTP υπεύθυνο για τη διεύθυνση του παραλήπτη.

- Σε περίπτωση που φέρει υπογραφή από έμπιστη οντότητα, τότε ο διακομιστής αρχείων ρυθμίσεων θεωρεί πιστοποιημένη την υπηρεσία που συνδέθηκε σε αυτόν και αποστέλλει κρυπτογραφημένο (με το δημόσιο κλειδί αυτής) το αρχείο ρυθμίσεων.

Σε περίπτωση όπου απαιτείται μεγαλύτερος βαθμός ασφάλειας, μπορεί ο διακομιστής αρχείων ρυθμίσεων να υπογράψει την αποστολή του αρχείου, ώστε ο Διακομιστής Ενεργών Υπηρεσιών να μπορέσει να επαληθεύσει ότι λαμβάνει όντως το αρχείο ρυθμίσεων από τον πραγματικό παραλήπτη του μηνύματος. Βέβαια στην περίπτωση αυτή ο διακομιστής αρχείων ρυθμίσεων θα πρέπει να έχει ανά πάσα στιγμή πρόσβαση στο ιδιωτικό κλειδί του χρήστη.

```
<!DOCTYPE SMTP_CONFIG SYSTEM "smtpconfig.dtd">
<SMTP_CONFIG>
 <FILTER name="SMTPHeaderFilter">
  <REJECT>
    <CONDITION>
      <SUBJECT type="contains" value="buy"/>
    </CONDITION>
  </REJECT>
 </FILTER>
</SMTP_CONFIG>
```

Σχήμα 2.8: Περιεχόμενα αρχείου ρυθμίσεων χρήστη για την υπηρεσία "Active SMTP"

Στο παράδειγμα 2.8 παρουσιάζονται τα περιεχόμενα του αρχείου ρυθμίσεων ενός χρήστη για την υπηρεσία Active SMTP. Ο χρήστης ζητά από κάθε Ενεργό Κόμβο που παρέχει αυτή την υπηρεσία, να μην προωθήσει μηνύματα που προορίζονται για τον ίδιο όταν στο θέμα (subject) αυτών περιλαμβάνεται η λέξη "buy". Το φίλτρο που χρησιμοποιεί ο χρήστης σε αυτό το παράδειγμα ονομάζεται "SMTPHeaderFilter" και επιτρέπει την επιλεκτική επεξεργασία μηνυμάτων ανάλογα με τις τιμές που παρουσιάζονται στις επικεφαλίδες τους. Εξαιτίας της επιλογής "REJECT" το φίλτρο θα δημιουργήσει ένα νέο μήνυμα (σφάλματος) το οποίο και θα αποστείλει στον αποστολέα του αρχικού μηνύματος προκειμένου να τον πληροφορήσει ότι το μήνυμα δεν έφτασε στον προορισμό του.

Στα πλαίσια του έργου CASPIAN αναπτύχθηκε και μια επέκταση του φίλτρου "SMTPHeaderFilter", ονόματι "SMTPAntispamFilter" το οποίο μετασχημάτιζε τα συνημμένα ενός ηλ. μηνύματος (π.χ. αρχεία PDF) σε μορφή κειμένου και εξέταζε αν αυτά περιείχαν λέξεις που παρέπεμπαν σε διαφημιστικά μηνύματα.

Ο χρήστης έχει τη δυνατότητα να ορίσει την εφαρμογή πολλαπλών φίλτρων τα οποία με τη σειρά τους μπορεί να ενεργήσουν με βάσει πολλαπλούς κανόνες. Η σειρά με την οποία περιγράφονται τα φίλτρα στο αρχείο ρυθμίσεων ορίζει και τη σειρά με την οποία αυτά θα εφαρμοστούν στα μηνύματα του χρήστη.

Κατά την επεξεργασία του αρχείου ρυθμίσεων η Ενεργή Υπηρεσία ελέγχει αν υπάρχει τοπικά αποθηκευμένο αντίγραφο (local filter cache) του φίλτρου που ζητείται. Αν κάτι τέτοιο δεν ισχύει τότε η Ενεργή Υπηρεσία αναζητά το εν λόγω φίλτρο μέσω μιας υπηρεσίας διάθεσης φίλτρων. Η υπηρεσία αυτή παρέχει μέσω διεπαφής RMI [2] σε μια Ενεργή Υπηρεσία την (ψηφιακά υπογεγραμμένη) υλοποίηση (bytecode) ενός φίλτρου. Ο κόμβος ο οποίος παρέχει την παραπάνω υπηρεσία δε χρειάζεται να βρίσκεται εγκατεστημένος στο ίδιο δίκτυο με τον κόμβο που θα χρησιμοποιήσει το φίλτρο. Η υπηρεσία διάθεσης φίλτρων μπορεί να παρέχεται μέσω Διαδικτύου από τους οργανισμούς ή τις εταιρίες που αναπτύσσουν τα φίλτρα. Η επιλογή του κόμβου από τον οποίο θα μεταφορτωθεί ένα φίλτρο μπορεί να γίνει βάσει σχετικών ρυθμίσεων του Περιβάλλοντος Εκτέλεσης ή βάσει στοιχείων που θα ληφθούν από υπηρεσία πληροφοριών καταλόγου (directory service) του δικτύου του

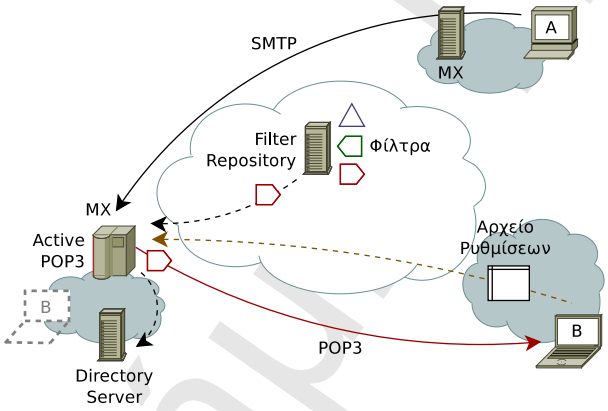
Ενεργού Κόμβου.

Η αρχιτεκτονική του συστήματος που περιγράφηκε παραπάνω απεικονίζεται στο παράδειγμα του σχήματος 2.9.

Θα πρέπει να σημειωθεί ότι όλα τα φίλτρα ακολουθούν ένα συγκεκριμένο πρότυπο υλοποίησης ("Mail Filter Interface"). Έτσι γίνεται δυνατή η χρήση των ίδιων φίλτρων από διαφορετικές Ενεργές Υπηρεσίες.

2.4.2 Υπηρεσία Active POP3

Για τους κινητούς κόμβους οι οποίοι δεν είναι εξοπλισμένοι με κάποιο διακομιστή υπηρεσιών SMTP ή δε βρίσκονται συνδεδεμένοι για μεγάλα χρονικά διαστήματα στο Διαδίκτυο, η πλατφόρμα Mobile Active Mail προσφέρει την υπηρεσία Active POP3 η οποία επιτρέπει την παραλαβή των μηνυμάτων του χρήστη από το οικείο δίκτυο αφού πρώτα εφαρμοστούν σε αυτά τα ζητούμενα (από το χρήστη) φίλτρα.



Σχήμα 2.9: Αρχιτεκτονική της υπηρεσίας Active POP3

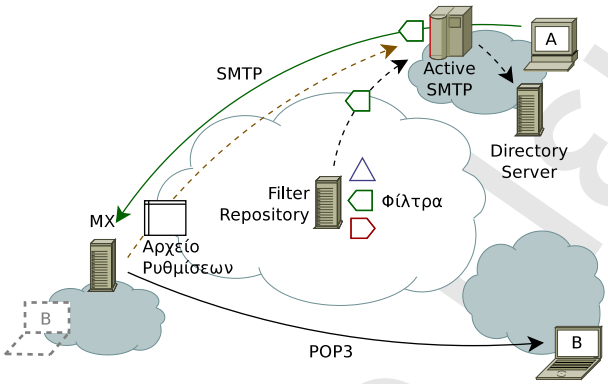
Η υπηρεσία είναι πλήρως συμβατή με το πρωτόκολλο POP3 και μπορεί να χρησιμοποιηθεί και ως αντικαταστάτης του διακομιστή υπηρεσιών POP3 σε ένα δίκτυο. Η εφαρμογή των φίλτρων ξεκινά αφού ο χρήστης ταυτοποιηθεί στην υπηρεσία. Κάθε φίλτρο μεταφέρεται στον Ενεργό Κόμβο με τον τρόπο που περιγράφηκε στην προηγούμενη υποενότητα. Μετά την εφαρμογή των φίλτρων, κάποια μηνύματα μπορεί να έχουν αλλάξει μορφή, άλλα μπορεί να έχουν διαγραφεί ενώ άλλα μπορεί να έχουν μετατραπεί σε κρυφά (δηλ. δεν θα εμφανιστούν σε ερώτημα τύπου "LIST" της mail user agent εφαρμογής του χρήστη). Αν προκύψει οποιοδήποτε πρόβλημα κατά την επεξεργασία ενός ή περισσοτέρων μηνυμάτων, θα δημιουργηθεί ένα νέο μήνυμα το οποίο θα περιγράφει το πρόβλημα και το οποίο θα παραλάβει ο χρήστης μαζί με τα υπόλοιπα μηνύματά του.

Μετά την εφαρμογή των φίλτρων το λογισμικό τύπου mail user agent θα μπορεί να χρησιμοποιήσει τις καθιερωμένες εντολές του πρωτοκόλλου POP3 για να διαχειριστεί τα μη κρυφά μηνύματα. Η δυνατότητα μεταβολής της κατάστασης ενός μηνύματος σε κρυφό επιτρέπει στο χρήστη να μεταφορτώσει μόνο τα μηνύματα που θεωρεί πιο σημαντικά, διατηρώντας παράλληλα τη δυνατότητα να εξετάσει τα υπόλοιπα σε κάποια άλλη χρονική στιγμή στο μέλλον.

Στο σχήμα 2.9 παρουσιάζεται συνοπτικά η αρχιτεκτονική της υπηρεσίας Active POP3.

2.4.3 Υπηρεσία Active SMTP

Η υπηρεσία Active SMTP αποτελεί μια πλήρη υλοποίηση του πρωτοκόλλου SMTP, ενώ παράλληλα επιτρέπει την εφαρμογή φίλτρων σύμφωνα με τις απαιτήσεις του παραλήπτη ενός μηνύματος. Σε αντίθεση με την υπηρεσία Active POP3, η Active SMTP επιτρέπει την επεξεργασία μηνυμάτων κατά το στάδιο αποστολής αυτών (δηλ. κατά το στάδιο παραλαβής αυτών από το διακομιστή SMTP του αποστολέα) και έτσι μπορεί να μειώσει σημαντικά τη σχετική κίνηση στο δίκτυο (αφού ένα μήνυμα, μετά την επεξεργασία αυτού, μπορεί να μη χρειαστεί να παραδοθεί στο διακομιστή SMTP του παραλήπτη και κατά συνέπεια στον κινητό κόμβο).



Σχήμα 2.10: Αρχιτεκτονική της υπηρεσίας Active SMTP

Η υπηρεσία μπορεί να εγκατασταθεί στους Ενεργούς Κόμβους οποιουδήποτε δικτύου και επεξεργάζεται τόσο την εισερχόμενη όσο και την εξερχόμενη SMTP κίνηση. Επίσης, έχει τη δυνατότητα τοπικής παράδοσης μηνυμάτων, επιτρέποντας έτσι τη χρήση της και ως κανονικού διακομιστή SMTP.

Η επεξεργασία ενός μηνύματος ξεκινά όταν παραληφθούν οι επικεφαλίδες και τα περιεχόμενα αυτού. Η υπηρεσία Active SMTP θα αναζητήσει το αρχείο με τις προτιμήσεις του παραλήπτη στον υπεύθυνο διακομιστή SMTP για τη διεύθυνση αυτού. Σε περίπτωση όπου δεν υπάρχει υπεύθυνος διακομιστής SMTP για τη διεύθυνση του παραλήπτη, θα αναζητηθούν οι προτιμήσεις του χρήστη στο σύστημα που παρουσιάζεται στο τμήμα domain της ηλ. διεύθυνσης αυτού (π.χ. στο σύστημα "some.example.com" για τη διεύθυνση "email@some.example.com"). Η εύρεση του υπεύθυνου SMTP γίνεται με ερώτημα τύπου "MX" στο σύστημα DNS, ενώ η μεταφορά του αρχείου ρυθμίσεων του χρήστη γίνεται με το μηχανισμό που περιγράφηκε στην ενότητα 2.4.1. Αν η μεταφορά του αρχείου ρυθμίσεων δεν είναι δυνατή, τότε η υπηρεσία προχωρά στην παραλαβή του μηνύματος και στην ενδεχόμενη προώθηση αυτού (αν δεν αφορά τοπικούς χρήστες).

Η επεξεργασία που μπορεί να γίνει σε ένα μήνυμα μέσω των φίλτρων έχει τα ακόλουθα αποτελέσματα:

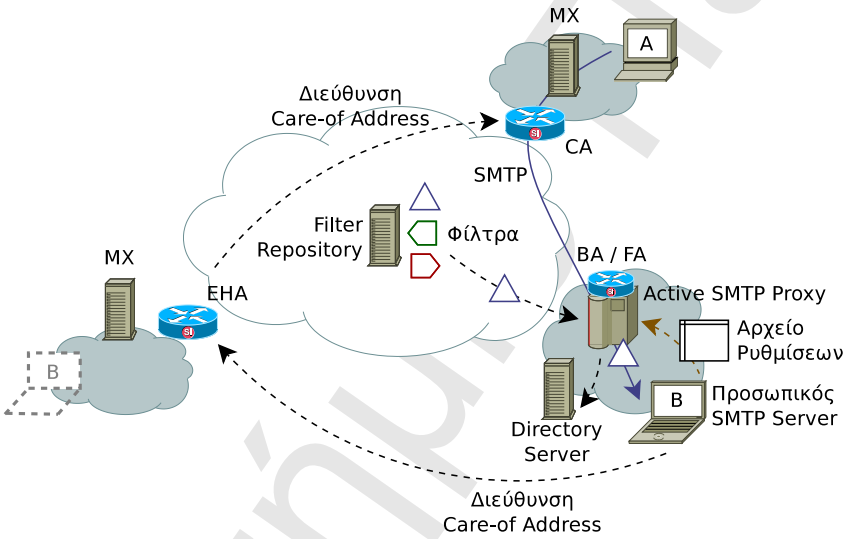
- την επιστροφή του μηνύματος στον αποστολέα με σχετικό μήνυμα σφάλματος
- τη σιωπηλή διαγραφή του μηνύματος
- το μετασχηματισμό των επικεφαλίδων ή περιεχομένων του μηνύματος
- τη δημιουργία και αποστολή νέου μηνύματος προς κάποιο παραλήπτη
- την προώθηση ή παράδοση του επεξεργασμένου μηνύματος

Τέλος, η υπηρεσία Active SMTP επιτρέπει και την εφαρμογή μιας σειράς φίλτρων στα εισερχόμενα μηνύματα, ανεξαρτήτως των προτιμήσεων των χρηστών. Τα φίλτρα αυτά ορίζονται από το διαχειριστή του Ενεργού Κόμβου και εφαρμόζονται πριν από τα φίλτρα που ορίζουν οι παραλήπτες των μηνυμάτων. Με τον τρόπο αυτό μπορεί να εφαρμοστούν πολιτικές όπως η μη προώθηση μηνυμάτων στα οποία έχει εντοπιστεί κακόβουλο λογισμικό (malware) ή διαφημιστικό περιεχόμενο (spam).

Στο σχήμα 2.10 παρουσιάζεται συνοπτικά η αρχιτεκτονική της υπηρεσίας Active SMTP.

2.4.4 Υπηρεσία Active SMTP Proxy

Η Ενεργή Υπηρεσία Active SMTP Proxy επιτρέπει τη δυναμική δρομολόγηση μηνυμάτων ηλ. ταχυδρομείου μέχρι τον Κινητό Κόμβο. Η υπηρεσία λειτουργεί ως ένας διαμεσολαβητής (transparent proxy) υπηρεσιών SMTP ο οποίος επεξεργάζεται και παραδίδει τα μηνύματα σε διακομιστή SMTP που εκτελείται στον Κινητό Κόμβο. Η παροχή της υπηρεσίας γίνεται από Ενεργούς Κόμβους που συμμετέχουν σε δίκτυα υποδοχής κινητών κόμβων.



Σχήμα 2.11: Αρχιτεκτονική της υπηρεσίας Active SMTP Proxy

Στο σχήμα 2.11 παρουσιάζεται η αρχιτεκτονική της υπηρεσίας. Οι Ενεργοί Δρομολογητές διαφόρων δικτύων, από τα οποία διέρχεται κίνηση με προορισμό τη διεύθυνση του κινητού κόμβου στο οικείο δίκτυο, ενημερώνονται από τον Extended Home Agent για τη νέα διεύθυνση του κόμβου στο δίκτυο υποδοχής. Αφού προωθήσουν την κίνηση αυτή στο δίκτυο υποδοχής, ένας Διακομιστής Ενεργών Υπηρεσιών (που παίζει και το ρόλο Ενεργού Δρομολογητή) θα συλλέξει την κίνηση που είναι σχετική με το πρωτόκολλο SMTP και με τη μέθοδο που περιγράφηκε στην υποενότητα 2.3.1 θα την προωθήσει στην τοπική υπηρεσία Active SMTP Proxy. Εκεί, θα εφαρμοστούν τα φίλτρα που όρισε ο χρήστης και το μήνυμα τελικά θα παραδοθεί στο διακομιστή SMTP του κινητού κόμβου.

Η υπηρεσία αυτή προϋποθέτει ότι ο χρήστης χρησιμοποιεί μια διεύθυνση ηλ. ταχυδρομείου της μορφής "user@host.domain.net", όπου user είναι το όνομα χρήστη που χρησιμοποιεί ο παραλήπτης στον κινητό κόμβο και host.domain.net είναι το όνομα DNS που αντιστοιχεί στη διεύθυνση του κόμβου στο οικείο δίκτυο. Επίσης, προϋποθέτει ότι ο κινητός κόμβος είναι εξοπλισμένος με ένα διακομιστή υπηρεσιών SMTP καθώς και με μία από τις δύο μεθόδους μεταφοράς του αρχείου ρυθμίσεων (web server ή secure configuration delivery service).

Στην περίπτωση όπου ο κινητός κόμβος βγει εκτός λειτουργίας, ή τεθεί εκτός δικτύου, η υπηρεσία Foreign Agent φροντίζει να ενημερώσει τον Extended Home Agent ώστε να μην προωθείται πλέον η σχετική κίνηση στο δίκτυο υποδοχής. Ένας Ενεργός Δρομολογητής ο οποίος δεν έχει ενημερωθεί για αυτή την αλλαγή και ο οποίος συνεχίζει να προωθεί την κίνηση ενός διακομιστή SMTP προς το δίκτυο υποδοχής, θα προκαλέσει την αποστολή πακέτου τύπου "ICMP Host Unreachable" προς τον εν λόγω διακομιστή. Ο διακομιστής τότε θα θέσει το μήνυμα προς αποστολή σε ουρά αναμονής και θα δοκιμάσει να το ξαναστείλει σε μεταγενέστερο χρόνο (queue and retransmit). Για κάθε μήνυμα που βρίσκεται στην ουρά αναμονής, ένας διακομιστής SMTP θα κάνει πολλαπλές προσπάθειες για να το αποστείλει. Οι προσπάθειες αυτές μπορεί να συνεχίσουν για αρκετές ημέρες. Έτσι γίνεται εφικτή η παράδοση μηνυμάτων ακόμη και σε κόμβους που βρίσκονται για εκτεταμένα χρονικά διαστήματα εκτός δικτύου.

Τέλος, η υπηρεσία Active SMTP Proxy μπορεί να λειτουργήσει και συνεργατικά με έναν ήδη υπάρχον διακομιστή υπηρεσιών SMTP στο δίκτυο του αποστολέα ενός μηνύματος. Σε αυτή την περίπτωση, ο τοπικός Ενεργός Δρομολογητής θα προωθήσει την εξερχόμενη από το δίκτυο SMTP κίνηση στην υπηρεσία Active SMTP Proxy και αυτή θα εφαρμόσει τα απαραίτητα φίλτρα προτού μεταφέρει τα μηνύματα στους διακομιστές υπηρεσιών SMTP των παραληπτών.

2.5 Ποιοτική αξιολόγηση

Η πλατφόρμα Mobile Active Mail αποτελεί ένα παράδειγμα χρήσης προγραμματιζόμενων υποδομών στο δικτυακό κορμό για την προσφορά νέων εξειδικευμένων υπηρεσιών στους χρήστες του Διαδικτύου. Το Ενεργό Δίκτυο της πλατφόρμας αποτελείται από Ενεργούς Δρομολογητές που προωθούν συγκεκριμένου τύπου κίνηση σε Ενεργούς Κόμβους, και Διακομιστές που φιλοξενούν Ενεργές Υπηρεσίες. Οι υπηρεσίες αυτές επιλέγονται από τους διαχειριστές των Ενεργών Κόμβων, φορτώνονται δυναμικά από έμπιστες πηγές λογισμικού και ο τρόπος λειτουργίας τους μεταβάλλεται ανάλογα με τη διερχόμενη κίνηση των χρηστών του δικτύου. Πρόκειται δηλαδή για ένα Ενεργό Δίκτυο που λειτουργεί σε επίπεδο Εφαρμογής (Application Layer Active Network).

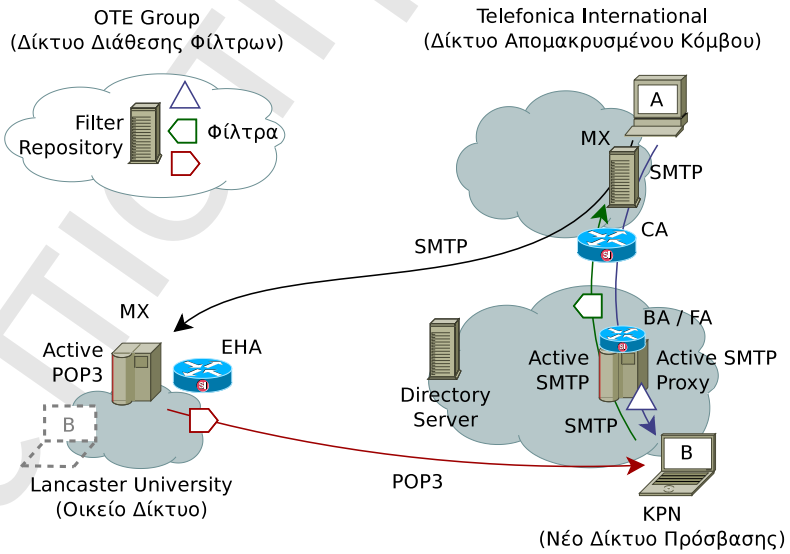
Η πλατφόρμα Mobile Active Mail επεκτείνει τις υπηρεσίες της αρχιτεκτονικής Mobile Active Overlay ώστε οι κινητοί χρήστες να μπορούν να παραλάβουν κατευθείαν στη συσκευή τους την ηλεκτρονική τους αλληλογραφία, χωρίς αυτή να χρειαστεί να περάσει από κάποιο διακομιστή στο οικείο δίκτυο αυτών. Οι υπηρεσίες της πλατφόρμας συμβάλλουν επίσης σημαντικά στην ελάττωση της συνολικής κίνησης που σχετίζεται με την ηλ. αλληλογραφία. Συγκεκριμένα, επιτρέπουν την επεξεργασία μηνυμάτων ακόμη και στο αρχικό στάδιο της πορείας τους προς τον παραλήπτη (δίκτυο αποστολέα), σύμφωνα με φίλτρα που ορίζονται τόσο από το διαχειριστή του κόμβου όσο και από τον παραλήπτη. Τα φίλτρα αυτά μπορούν να εμποδίσουν την ανεπιθύμητη αλληλογραφία να φτάσει στον παραλήπτη αλλά μπορούν και να μετασχηματίσουν το περιεχόμενο των μηνυμάτων κατά τέτοιο τρόπο ώστε να είναι πιο συμβατό με τις δυνατότητες του σταθμού εργασίας ή της κινητής συσκευής του παραλήπτη.

Για την ασφαλή εφαρμογή των φίλτρων και την ασφαλή μεταφορά των ρυθμίσεων αυτών, εφαρμόζεται μια μέθοδος που βασίζεται στην τεχνολογία PGP [66]. Η επιλογή αυτή επιτρέπει την αξιοποίηση μιας ήδη υπάρχουσας Υποδομής Δημοσίου Κλειδιού (αυτής του συστήματος PGP) για την εύρεση και επαλήθευση πιστοποιητικών χρηστών, διακομιστών αλλά και παρόχων λογισμικού. Μάλιστα, πολλοί από τους μελλοντικούς χρήστες της πλατφόρμας Mobile Active Mail ενδεχομένως να είναι ήδη μέλη της παραπάνω υποδομής καθώς

η τεχνολογία PGP αποτελεί μια από τις πιο διαδεδομένες μεθόδους για την κρυπτογράφηση ηλ. μηνυμάτων.

Η πλατφόρμα Mobile Active Mail δεν απαιτεί την εγκατάσταση Ενεργών Κόμβων σε κάθε δίκτυο που συμμετέχει σε αυτή. Υπηρεσίες όπως η Active POP3 επιτρέπουν σε ένα χρήστη να εφαρμόσει τα επιθυμητά φίλτρα ακόμη και στο τελευταίο στάδιο μεταφοράς της αλληλογραφίας, κατά την παραλαβή αυτής από τον οικείο διακομιστή υπηρεσιών POP3. Με τον τρόπο αυτό ο χρήστης μπορεί να εξασφαλίσει την εκτέλεση των φίλτρων στα μηνύματα ακόμη και στην περίπτωση όπου δε θα βρεθεί κάποιο δίκτυο μεταφοράς με εγκατεστημένο έναν Ενεργό Κόμβο της πλατφόρμας Mobile Active Mail. Επίσης, σε περίπτωση όπου ένας κινητός κόμβος βρεθεί εκτός δικτύου, η κίνηση που δρομολογείται προς αυτόν θα παραμείνει σε διακομιστή SMTP έως ότου εκείνος επανασυνδεθεί στο δίκτυο. Αυτή η λειτουργία επιτρέπει στους κινητούς κόμβους να αλλάζουν συνεχώς δίκτυο υποδοχής, όπως συμβαίνει με τις συσκευές κινητής τηλεφωνίας, δίχως να χάνονται τα μηνύματα που προορίζονται για τους χρήστες αυτών.

Η υλοποίηση της πλατφόρμας Mobile Active Mail έχει σχεδιαστεί με σπονδυλωτό (modular) τρόπο επιτρέποντας έτσι την παράλληλη δημιουργία νέων υπηρεσιών και φίλτρων από ανεξάρτητους φορείς. Στη σπονδυλωτή σχεδίαση συνεισφέρει ιδιαίτερα η ύπαρξη σαφών διεπαφών (APIs) μέσω των οποίων μία υπηρεσία επικοινωνεί με τον Ενεργό Κόμβο, με άλλες υπηρεσίες αλλά και με τα φίλτρα που εκτελεί για λογαριασμό των χρηστών. Τα φίλτρα που μεταφέρονται από τρίτους κόμβους διατηρούνται σε προσωρινό αποθηκευτικό χώρο (filter cache) για ικανό χρονικό διάστημα (το οποίο ορίζει ο διαχειριστής) ώστε να μην επιφορτίζεται το δίκτυο κάθε φορά που απαιτείται η εκτέλεση αυτών. Σε μια μελλοντική έκδοση της υλοποίησης, το αρχείο ρυθμίσεων των χρηστών για τα φίλτρα θα μπορούσε να αντικατασταθεί από μια περιορισμένη γλώσσα προγραμματισμού μέσω της οποίας ο χρήστης θα μπορούσε να δημιουργήσει πιο σύνθετες εφαρμογές με τα ήδη υπάρχοντα φίλτρα. Επίσης, ο Ενεργός Κόμβος θα μπορούσε να εισάγει πιο αυστηρούς περιορισμούς στους κοινόχρηστους πόρους που καταναλώνουν οι υπηρεσίες και τα φίλτρα (π.χ. μνήμη, χρόνος στον επεξεργαστή) ώστε ένα νήμα του κόμβου να μη μπορεί να στερήσει τους πόρους αυτούς από τα υπόλοιπα νήματα.



Σχήμα 2.12: Αρχιτεκτονική πειράματος έργου CASPIAN

Η πλατφόρμα Mobile Active Mail εγκαταστάθηκε και λειτούργησε πιλοτικά στα πλαίσια

πειράματος για το έργο CASPIAN του ευρωπαϊκού φορέα Eurescom. Στο πείραμα χρησιμοποιήθηκαν υποδομές του Πανεπιστημίου του Lancaster (Lancaster University), του Οργανισμού Τηλεπικοινωνιών Ελλάδος (OTE Group), του τηλεπικοινωνιακού ομίλου Telefonica International της Ισπανίας και του τηλεπικοινωνιακού παρόχου KPN της Ολλανδίας. Στο σχήμα 2.12 παρουσιάζονται οι υπηρεσίες που εγκαταστάθηκαν στο δίκτυο κάθε φορέα που συμμετείχε στο πείραμα. Το πείραμα περιλάμβανε τη μετακίνηση ενός κόμβου B από το οικείο δίκτυο στο Lancaster University, στο νέο δίκτυο πρόσβασης του παρόχου KPN. Στο πρώτο σενάριο του πειράματος, ένας δεύτερος κόμβος A που βρισκόταν στο δίκτυο της Telefonica International θα έστελνε κίνηση ηλ. ταχυδρομείου προς τον πρώτο κόμβο, η οποία μέσω Ενεργού Δρομολογητή (στο ίδιο δίκτυο) θα έφτανε στη νέα τοποθεσία του κινητού κόμβου. Στο δεύτερο σενάριο του πειράματος εξετάστηκε η υπηρεσία Active POP3 που ήταν εγκατεστημένη στο οικείο δίκτυο χρησιμοποιώντας και πάλι αλληλογραφία προερχόμενη από τον κόμβο A. Στο τρίτο και τελευταίο σενάριο εξετάστηκε η υπηρεσία Active SMTP που ήταν εγκατεστημένη στο νέο δίκτυο πρόσβασης. Αυτή τη φορά ο κόμβος B ήταν ο αποστολέας των μηνυμάτων και ο κόμβος A ο παραλήπτης.

Το εν λόγω πείραμα ολοκληρώθηκε με απόλυτη επιτυχία, επιβεβαιώνοντας το επίπεδο ωριμότητας της υλοποίησης της πλατφόρμας. Στην επιτυχή έκβαση του πειράματος συνετέλεσαν πολλοί παράγοντες, ένας από τους οποίους ήταν η λιτή αρχιτεκτονική των υπηρεσιών της πλατφόρμας, η οποία μείωσε στο ελάχιστο το διαχειριστικό κόστος λειτουργίας αυτής (π.χ. εργασίες εγκατάστασης, ρυθμίσεις υπηρεσιών, έλεγχος ορθής λειτουργίας κτλ.). Ένας άλλος παράγοντας ήταν η επιλογή ώριμων τεχνολογιών κατά την ανοικοδόμηση της προγραμματιζόμενης υποδομής (προώθηση πακέτων μέσω διεπαφής netfilter, κλήση απομακρυσμένων συναρτήσεων μέσω διεπαφής JNI κ.α.). Τέλος σημαντικό ρόλο έπαιξε και το γεγονός ότι οι υπηρεσίες της πλατφόρμας ήταν πλήρως συμβατές με τα πρωτόκολλα SMTP και POP3. Αυτό επέτρεψε την εύκολη ενσωμάτωση των Ενεργών Υπηρεσιών σαν απλούς διακομιστές των παραπάνω δύο πρωτοκόλλων στα αντίστοιχα δίκτυα, αλλά και την άμεση αξιοποίηση αυτών από οποιοδήποτε λογισμικό τύπου mail user agent.

Κεφάλαιο 3

Προγραμματιζόμενες Υποδομές στα Άκρα του Δικτύου

Το Διαδίκτυο σχεδιάστηκε με μια "end to end" φιλοσοφία, σύμφωνα με την οποία οι συσκευές που τοποθετούνται στα άκρα του δικτύου χαίρουν μεγαλύτερων επεξεργαστικών δυνατοτήτων από αυτές που συνθέτουν το δικτυακό κορμό. Είναι επίσης φυσικό σε ένα τέτοιο δίκτυο η κίνηση που παράγεται ή καταναλώνεται από κάποια συσκευή στα άκρα του δικτύου να είναι μικρότερη αυτής που διέρχεται από ένα κόμβο του δικτυακού κορμού. Οι παραπάνω δύο παράγοντες, σε συνδυασμό με τη διαχειριστική αυτοτέλεια που συνήθως έχουν οι οργανισμοί επί των συσκευών που τοποθετούν στα άκρα του δικτύου τους, κάνουν τις συσκευές αυτές ιδανικές για χρήση σε προγραμματιζόμενες δικτυακές υποδομές.

Μιας και οι συσκευές αυτές δεν παρέχουν απαραίτητα υπηρεσίες σε τρίτους κόμβους στο Διαδίκτυο, ο διαχειριστής αυτών μπορεί ελεύθερα να περιορίσει το είδος της κίνησης που θα φτάσει σε αυτές καθώς και τον όγκο της κίνησης που εκείνες θα επεξεργαστούν. Επίσης, η διαχειριστική αυτοτέλεια επί αυτών δίνει τη δυνατότητα σε ένα οργανισμό να ορίσει με σαφή τρόπο το πλαίσιο (σε υλικό και λογισμικό) εντός του οποίου οι συσκευές αυτές θα λειτουργήσουν ως μια ελεύθερα προγραμματιζόμενη υποδομή (δηλ. να προδιαγράψει το προγραμματιζόμενο περιβάλλον). Αν από τις δυναμικές υπηρεσίες, που θα προκύψουν μέσω του προγραμματισμού των συσκευών επωφεληθούν το δίκτυο και οι χρήστες αυτού, τότε πρόκειται για μια υποδομή που λειτουργεί σύμφωνα με τις αρχές των *Ισχυρών Ενεργών Δικτύων* (βλ. ενότητα 1.1).

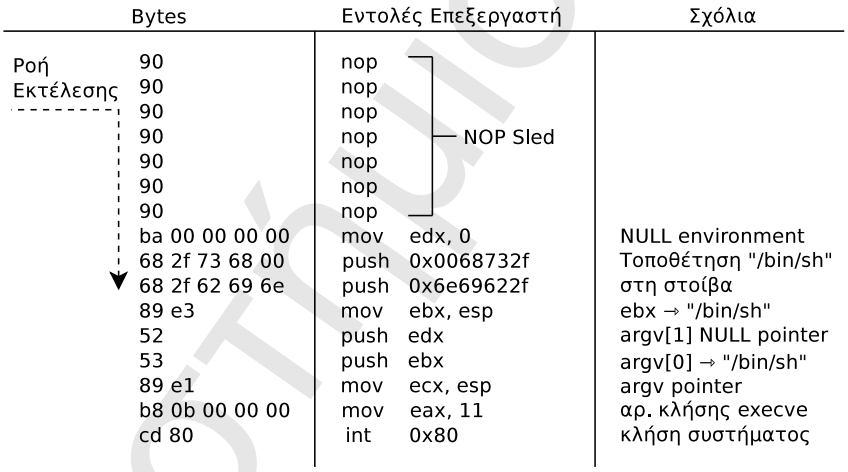
Τα Ισχυρά Ενεργά Δίκτυα προγραμματίζονται από ειδικό λογισμικό που εντοπίζουν στη διερχόμενη δικτυακή κίνηση (βλ. active capsules [6]). Χαρακτηριστική εφαρμογή Ισχυρών Ενεργών Δικτύων είναι η πλατφόρμα ANTS [14] που βοηθά στην υλοποίηση και τη μελέτη νέων δικτυακών πρωτοκόλλων. Οι διαχειριστές των κόμβων μιας τέτοιας υποδομής μπορούν να ορίσουν περιορισμούς ως προς τους πόρους που θα έχει πρόσβαση το «ξένο» λογισμικό αλλά και ως προς το είδος του λογισμικού που θα εκτελεστεί στους κόμβους. Οι περιορισμοί αυτοί περιγράφονται πιο αναλυτικά στο κεφάλαιο 5 όπου εξετάζονται τα μέτρα ασφάλειας του περιβάλλοντος εκτέλεσης ενός προγραμματιζόμενου δικτυακού κόμβου.

Στο παρόν κεφάλαιο θα παρουσιαστεί μια νέα προγραμματιζόμενη δικτυακή πλατφόρμα η οποία έχει ως στόχο τον εντοπισμό πολυμορφικού κακόβουλου λογισμικού στη δικτυακή κίνηση. Η πλατφόρμα έχει σχεδιαστεί σύμφωνα με τις αρχές των Ισχυρών Ενεργών Δικτύων και εγκαθίσταται στα άκρα των δικτύων που καλείται να προστατεύσει. Η πλατφόρμα λειτουργεί με πλήρως κατανεμημένο τρόπο και μπορεί εύκολα να επεκταθεί ώστε να καλύψει τις ανάγκες δικτύων υψηλών ταχυτήτων. Μέσα από τη διερεύνηση της παραπάνω πλατφόρμας θα σκιαγραφηθούν τα ιδιαίτερα χαρακτηριστικά των προγραμ-

ματιζόμενων υποδομών αυτού του τύπου και θα τονιστούν τα σημαντικά τους οφέλη για το δίκτυο και τους χρήστες αυτού. Επίσης, θα παρουσιαστούν ειδικοί μηχανισμοί για την ασφαλή εκτέλεση «ξένου» λογισμικού σε τέτοιες υποδομές καθώς και μέτρα για την αποδοτική παροχή δυναμικών υπηρεσιών.

3.1 Προστασία λογισμικού υπηρεσιών από διαδικτυακές επιθέσεις εκμετάλλευσης τρωτοτήτων

Με την εισαγωγή υπηρεσιών στο Διαδίκτυο γεννήθηκε μια νέα μορφή απειλής. Κακόβουλοι χρήστες εκμεταλλεύονται αδυναμίες στο λογισμικό των υπηρεσιών προκειμένου να δημιουργήσουν προβλήματα στην παροχή αυτών των υπηρεσιών (denial of service attacks) ή να εκτελέσουν δικά τους (συνήθως κακόβουλα) προγράμματα στα συστήματα που φιλοξενούν τις ευπαθείς υπηρεσίες. Η εκμετάλλευση μιας αδυναμίας γίνεται με την αποστολή ειδικών δεδομένων, τα οποία προκαλούν την εκτέλεση ενός ευπαθούς τμήματος του κώδικα της εφαρμογής. Αναλόγως με το είδος της ευπάθειας, το σύστημα που φιλοξενεί την υπηρεσία και επεξεργάζεται τα δεδομένα του κακόβουλου χρήστη μπορεί να οδηγηθεί ακόμη και στην εκτέλεση εντολών που συμπεριλαμβάνονται στα κακόβουλα δεδομένα. Οι εντολές αυτές ονομάζονται *payload* («φορτίο» της επίθεσης) ενώ τα (συνολικά) δεδομένα που απέστειλε ο κακόβουλος χρήστης προκειμένου να εκμεταλλευθεί την αδυναμία ονομάζονται *exploit* («δεδομένα εκμετάλλευσης»).



Σχήμα 3.1: Shellcode με NOP Sled για την αρχιτεκτονική x86

Μια αδυναμία μπορεί να επιτρέψει σε ένα κακόβουλο χρήστη να ανακατευθύνει την κανονική ροή εκτέλεσης ενός προγράμματος σε μια σειρά *εντολών επεξεργαστή* (CPU instructions) που εκείνος έχει ορίσει στο payload. Σε αυτή την περίπτωση το payload περιέχει εντολές που ονομάζονται *shellcode* («κώδικας κελύφους») καθώς τυπικά οδηγούν στην απόκτηση ενός «κελύφους» απομακρυσμένης πρόσβασης (χρησιμοποιώντας τις κατάλληλες κλήσεις συστήματος). Επειδή συχνά δεν είναι δυνατό για τον επιτιθέμενο να υπολογίσει με απόλυτη ακρίβεια τη θέση μνήμης στην οποία θα βρεθεί το shellcode κατά την απόπειρα εκμετάλλευσης μιας αδυναμίας, συνηθίζεται το πρώτο μέρος αυτού να αποτελείται από μια σειρά εντολών οι οποίες δεν έχουν κάποιο ειδικό σκοπό αλλά οδηγούν την εκτέλεση στο κυρίως σώμα του shellcode. Οι εντολές αυτές ονομάζονται "NOP sled" («έλκυθρο εντολών τύπου No Operation») και λειτουργούν σαν ένα «έλκυθρο» καθώς σε οποιαδήποτε

```
    jmp L2
L1:
    pop eax ; eax = L0
    ...
L2:
    call L1
L0:
```

(α′) GetPC με την εντολή Call

```
    fldz
    fnstenv [esp - 12]
    pop eax
    add al, 10 ; eax = L0
    nop
L0:
    ...
```

(β′) GetPC με την εντολή fnstenv

Σχήμα 3.2: Δύο παραδείγματα shellcode τύπου GetPC για την αρχιτεκτονική x86

από αυτές και αν βρεθεί ο επεξεργαστής, θα οδηγηθεί στην εκτέλεση του κυρίως σώματος του shellcode. Ένα παράδειγμα shellcode με NOP sled παρουσιάζεται στο σχήμα 3.1. Σε περιπτώσεις όπου το shellcode πρέπει να ανακαλύψει την ακριβή θέση μνήμης στην οποία βρίσκεται (π.χ. για να λάβει πρόσβαση σε δεδομένα που βρίσκονται σε σχετική με αυτή θέση) εισάγονται στον κώδικα αυτού εντολές οι οποίες βρίσκουν δυναμικά τη διεύθυνση εκτέλεσης (instruction pointer) του επεξεργαστή. Στις εντολές αυτές έχει δοθεί το χαρακτηριστικό όνομα GetPC ("Get Program Counter"). Στο σχήμα 3.2 παρουσιάζονται δύο παραδείγματα τέτοιου κώδικα τα οποία εντοπίζουν δυναμικά στη μνήμη τη θέση της ετικέτας L0.

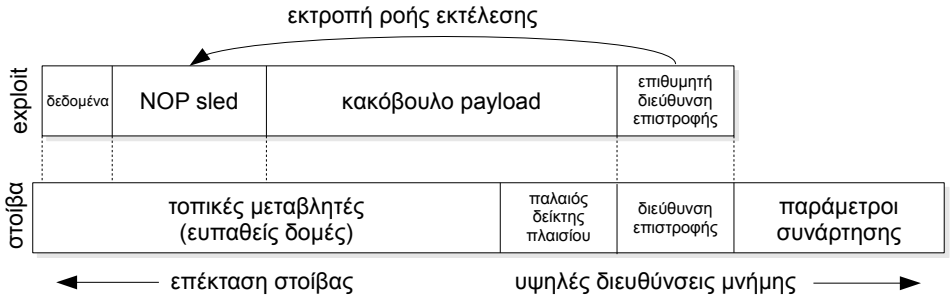
Οι επιθέσεις προς διαδικτυακές υπηρεσίες στις οποίες ένας επιτιθέμενος μπορεί να χρησιμοποιήσει shellcode χαρακτηρίζονται ως υψηλού κινδύνου, καθώς δύναται να οδηγήσουν στην αυτόματη, απομακρυσμένη και πλήρη εκμετάλλευση συστημάτων στο Διαδίκτυο. Για τον περιορισμό αυτής της απειλής έχουν προταθεί διάφορα προληπτικά μέτρα ασφάλειας τόσο σε επίπεδο εφαρμογής όσο και σε επίπεδο δικτύου.

Τα μέτρα ασφάλειας σε επίπεδο εφαρμογής δημιουργούν ειδικές συνθήκες στο περιβάλλον εκτέλεσης ώστε να γίνει πρακτικά αδύνατη η εκμετάλλευση με shellcode των όποιων αδυναμιών. Τα μέτρα αυτά δεν ανιχνεύουν το ίδιο το shellcode αλλά προκαλούν τον άμεσο τερματισμό μιας ευπαθούς διεργασίας όταν υπάρχει μια απόπειρα εκμετάλλευσης (π.χ. υπερχείλιση μνήμης, εκτροπή ροής εκτέλεσης κλπ.). Από τον τερματισμό αυτό ειδοποιείται έμμεσα ο διαχειριστής για την πιθανότητα ύπαρξης κάποιας ευπάθειας στο σύστημα. Σε κάθε περίπτωση πάντως, τα μέτρα αυτά δεν επιτρέπουν στο shellcode να εκτελεστεί κανονικά. Προτού παρουσιαστούν τα πιο σημαντικά από αυτά τα μέτρα, θα γίνει μια σύντομη παρουσίαση των αιτιών που οδηγούν στην εκτροπή της ροής εκτέλεσης ενός προγράμματος. Η παρουσίαση αυτή θα βοηθήσει τον αναγνώστη στην καλύτερη κατανόηση του τρόπου λειτουργίας των παραπάνω μέτρων προστασίας.

Η εκτροπή της ροής εκτέλεσης μιας εφαρμογής σε μια τυχαία διεύθυνση (που ενδεχομένως ελέγχει ο επιτιθέμενος) οφείλεται σε σφάλματα λογισμικού που επιτρέπουν την αυθαίρετη εγγραφή δεδομένων σε θέσεις μνήμης της εφαρμογής. Η μνήμη μιας διεργασίας παίζει ουσιαστικό ρόλο στην πορεία εκτέλεσής της, καθώς εκεί τοποθετούνται η διεύθυνση επιστροφής μιας συνάρτησης (return address), αλλά και δείκτες προς συναρτήσεις (της εφαρμογής και τρίτων βιβλιοθηκών). Αν οποιαδήποτε από αυτές τις πληροφορίες μεταβληθεί από τον επιτιθέμενο τότε είναι δυνατή η εκτροπή της ροής εκτέλεσης της εφαρμογής. Η μεταβολή μιας περιοχής μνήμης μπορεί να συμβεί για διάφορους λόγους, π.χ.

- εξαιτίας σφάλματος το οποίο θα προκαλέσει την εγγραφή δεδομένων πέραν των ορίων μιας δομής δεδομένων (buffer overflow [67]),
- εξαιτίας σφάλματος το οποίο θα επιτρέψει την εγγραφή δεδομένων σε μια τυχαία θέση μνήμης (π.χ. format string bug [68])

Τα παραπάνω σφάλματα συναντώνται συχνότερα σε γλώσσες προγραμματισμού όπως η C, οι οποίες δίνουν στον προγραμματιστή τη δυνατότητα να λάβει άμεσα πρόσβαση και να διαμορφώσει τα περιεχόμενα της μνήμης της εφαρμογής. Επίσης, πολλά σφάλματα τύπου format string είναι εύκολο να εντοπιστούν κατά τη μεταγλώττιση [69] ή την εκτέλεση [68]. Κάτι τέτοιο όμως δεν ισχύει για μεγάλη μερίδα σφαλμάτων τύπου buffer overflow. Τα σφάλματα αυτού του τύπου μπορούν να επηρεάσουν τόσο δομές που βρίσκονται στη *στοίβα* όσο και δομές που βρίσκονται στο *σωρό*.



Σχήμα 3.3: Υπερχείλιση στοίβας και εκτέλεση κακόβουλου payload

Στο παράδειγμα του σχήματος 3.3 ο επιτιθέμενος αποστέλλει ένα ειδικά διαμορφωμένο exploit το οποίο θα προκαλέσει την υπερχείλιση μιας τοπικής μεταβλητής (ή δομής) στη στοίβα μιας ευπαθούς συνάρτησης. Η υπερχείλιση θα οδηγήσει στην επικάλυψη της αποθηκευμένης διεύθυνσης επιστροφής με μια νέα διεύθυνση η οποία θα δείχνει στο NOP sled του exploit. Έτσι, κατά την έξοδο από την ευπαθή συνάρτηση, η εφαρμογή θα εκτελέσει τις εντολές που περιέχονται στο payload.

Για την αντιμετώπιση επιθέσεων υπερχείλισης μνήμης στη στοίβα [70], στο σωρό [71] αλλά και την αυθαίρετη επικάλυψη δεικτών σε συναρτήσεις [72], ο πυρήνας σύγχρονων λειτουργικών συστημάτων φορτώνει σε τυχαίες διευθύνσεις μνήμης τις σελίδες του κώδικα και των δεδομένων των εφαρμογών. Η τεχνική αυτή ονομάζεται "Address Space Layout Randomisation" (ASLR) [73] και αποκρύπτει από τον επιτιθέμενο τη διεύθυνση μνήμης στην οποία θα τοποθετηθούν τα δεδομένα του αλλά και τις διευθύνσεις μνήμης όπου έχουν τοποθετηθεί τα δεδομένα και ο κώδικας της εφαρμογής. Δυστυχώς, σε περιπτώσεις δικτυακών υπηρεσιών που χρησιμοποιούν την κλήση συστήματος `fork` ο επιτιθέμενος μπορεί να εντοπίσει τη σωστή διεύθυνση με μια εξαντλητική μέθοδο (brute-force), καθώς η ζητούμενη διεύθυνση παραμένει ίδια για κάθε νέα διεργασία που δημιουργείται για να εξυπηρετήσει μια αίτηση πελάτη. Επίσης, η τυχαιοποίηση των σελίδων μνήμης του κώδικα μιας εφαρμογής προϋποθέτει ότι η εφαρμογή έχει μεταγλωττιστεί με κατάλληλο τρόπο ώστε να μπορεί να εκτελεστεί από οποιαδήποτε θέση μνήμης (position independent executable - PIE). Ωστόσο, πολλές από τις διαδικτυακές υπηρεσίες που χρησιμοποιούνται σήμερα δεν έχουν μεταγλωττιστεί κατ' αυτό τον τρόπο. Θα πρέπει να σημειωθεί επίσης, ότι σε μερικές εφαρμογές (π.χ. φυλλομετρητές) ο επιτιθέμενος έχει τη δυνατότητα να προκαλέσει μια τέτοια σειρά από δυναμικές δεσμεύσεις μνήμης ώστε δικά του δεδομένα να τοποθετηθούν σε θέσεις μνήμης που εκείνος μπορεί να προβλέψει. Η τεχνική αυτή ονομάζεται "heap spraying" και χρησιμοποιείται πολύ συχνά για την εκμετάλλευση τρωτοτήτων τύπου "user after free" [74].

Εκτός από την τυχαιοποίηση της μνήμης μιας εφαρμογής, το λειτουργικό σύστημα μπορεί να προχωρήσει και στην απαγόρευση της εκτέλεσης εντολών από σελίδες μνήμης στις οποίες η εφαρμογή έχει δικαίωμα εγγραφής (βλ. exec-shield [75]). Στην περίπτωση όπου ο επεξεργαστής δεν υποστηρίζει την αφαίρεση του δικαιώματος εκτέλεσης από μια σελίδα μνήμης, τότε η δυνατότητα αυτή εξομοιώνεται με ειδικό λογισμικό. Για να προσπεράσουν

αυτό το μέτρο προστασίας, οι επιτιθέμενοι εκμεταλλεύονται τον ήδη υπάρχοντα κώδικα της εφαρμογής (ή των βιβλιοθηκών) που βρίσκεται σε εκτελέσιμες σελίδες μνήμης και ο οποίος είναι ισοδύναμος με το payload τους (επίθεση τύπου "return-to-libc" [76]). Μάλιστα, σε περιπτώσεις όπου το παρόν μέτρο συνδυάζεται με το μέτρο ASLR, οι επιτιθέμενοι εντοπίζουν πολλαπλά τμήματα κώδικα τα οποία βρίσκονται σε στατικές θέσεις μνήμης[1] και τα συνδέουν μεταξύ τους με μια σειρά από διευθύνσεις επιστροφής. Η τεχνική αυτή ονομάζεται "Return-oriented programming" (ROP) [77] και επιτρέπει στους επιτιθέμενους να πετύχουν το στόχο τους αξιοποιώντας πολλά μικρά τμήματα κώδικα που βρίσκονται σε γνωστές διευθύνσεις μνήμης. Όπως έχουν δείξει πρόσφατες εργασίες αυτά τα τμήματα κώδικα (γνωστά και ως "gadgets") δεν είναι απαραίτητο να καταλήγουν σε εντολές τύπου `return`. Μπορεί να συδεθούν μεταξύ τους και με άλλες ισοδύναμες εντολές [78].

Ένα τρίτο μέτρο για την προστασία των εφαρμογών είναι η εισαγωγή ειδικών τιμών στη στοίβα και το σωρό, η αλλαγή των οποίων θα σηματοδοτήσει την ύπαρξη ενός προβλήματος υπερχείλισης μνήμης. Η τεχνολογία StackGuard [79] προσθέτει μια τυχαία τιμή στη στοίβα (πριν την αποθηκευμένη διεύθυνση του προηγούμενου πλαισίου και μετά τις τοπικές μεταβλητές) την οποία και εξετάζει πριν ολοκληρωθεί η σχετική συνάρτηση. Αν η τιμή έχει αλλάξει τότε η εφαρμογή τερματίζεται. Παρόμοιες τεχνολογίες υπάρχουν και για το σωρό, μόνο που εκεί ο εντοπισμός μιας υπερχείλισης θα γίνει στην επόμενη κλήση μιας συνάρτησης διαμόρφωσης του σωρού [80]. Η εισαγωγή των εντολών για την εξέταση της ειδικής τιμής γίνεται κατά την μεταγλώττιση. Στο πλαίσιο προστασίας των περιεχομένων της στοίβας μερικοί μεταγλωττιστές προχωρούν και σε αλλαγή της σειράς με την οποία τοποθετούνται οι τοπικές μεταβλητές, ώστε να προστατευτούν δείκτες προς συναρτήσεις (function pointers) από πίνακες στους οποίους πιθανόν να έχει γίνει υπερχείλιση. Πάντως, όπως και στην περίπτωση του ASLR, αν μια διαδικτυακή υπηρεσία χρησιμοποιεί την κλήση συστήματος `fork` για να δημιουργήσει νέες διεργασίες για την εξυπηρέτηση πελατών, τότε είναι δυνατή (μέσω εξαντλητικής αναζήτησης) η εύρεση (ανά byte) της ειδικής τιμής που τοποθετήθηκε στη στοίβα ή στο σωρό.

Μέτρα σαν το ASLR, το StackGuard και το exec-shield βοηθούν επίσης στην προστασία πυρήνων λειτουργικών συστημάτων [81] από αντίστοιχες επιθέσεις.

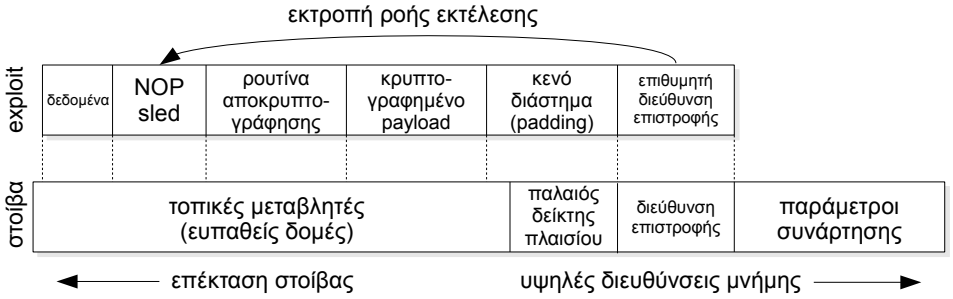
Στη βιβλιογραφία έχουν προταθεί διάφορες λύσεις [82] για την εποπτεία της ροής εκτέλεσης ενός προγράμματος (ή ενός πυρήνα λειτουργικού συστήματος) μέσω τρίτου λογισμικού το οποίο παρακολουθεί τα μονοπάτια εκτέλεσης. Αν η εκτέλεση παρεκκλίνει από τα προκαθορισμένα μονοπάτια τότε τερματίζεται το επικίνδυνο λογισμικό και ενημερώνεται ο διαχειριστής. Οι λύσεις αυτές παραμένουν ακόμη σε πειραματικό επίπεδο καθώς αυξάνουν σημαντικά το κόστος εκτέλεσης ανά εφαρμογή ενώ ταυτόχρονα απαιτούν από το διαχειριστή να διατηρεί ένα πλήρως ενημερωμένο προφίλ ασφάλειας ανά εφαρμογή. Μια διαφορετική προσέγγιση είναι η παρακολούθηση μιας διεργασίας σε επίπεδο κλήσης συστήματος. Εκεί, ο πυρήνας του λειτουργικού συστήματος μπορεί να ελέγξει αν ο τρόπος που έγινε η κλήση είναι συμβατός με το προφίλ ασφάλειας της εφαρμογής (π.χ. εξέταση αίτησης πρόσβασης σε αρχείο εκτός του καταλόγου από τον οποίο εκτελείται η εφαρμογή). Η προσέγγιση αυτή συνοδεύεται από μικρότερο κόστος κατά την εκτέλεση (καθώς οι έλεγχοι ασφάλειας γίνονται μόνο κατά τις κλήσεις συστήματος) και χρησιμοποιείται με επιτυχία για την προστασία διαφόρων κρίσιμων εφαρμογών και υπηρεσιών (βλ. SELinux [83], AppArmor [84]).

Προληπτικά μέτρα σαν τα παραπάνω έχουν εφαρμοστεί σε μικρό ποσοστό από τα συστήματα που συνδέονται σήμερα στο Διαδίκτυο. Τα περισσότερα συστήματα παραμένουν

[1] Είτε για λόγους συμβατότητας (backwards compatibility) σε περιπτώσεις βιβλιοθηκών είτε εξαιτίας της απουσίας ενός εκτελέσιμου τύπου PIE.

ευάλωτα σε αυτοματοποιημένες επιθέσεις που εκμεταλλεύονται γνωστές (και μη) αδυναμίες. Οι επιθέσεις αυτές συχνά παίρνουν τη μορφή «επιδημίας» (βλ. "worms" [48]), καθώς οι υπολογιστές που γίνονται θύματα των επιθέσεων μεταδίδουν με αυτόματο τρόπο το κακόβουλο λογισμικό σε άλλα ευπαθή συστήματα. Πολλά από τα μολυσμένα συστήματα θα χρησιμοποιηθούν αργότερα ως ενδιάμεσοι κόμβοι ("zombies") σε επιθέσεις τύπου άρνησης εξυπηρέτησης ή θα συμμετάσχουν σε προγραμματιζόμενα δίκτυα επικάλυψης (botnets) για την εκπλήρωση άλλων σκοπών (αποστολή διαφημιστικών μηνυμάτων κλπ.) [85]. Ο εντοπισμός shellcode στη διαδικτυακή κίνηση είναι ένα μέτρο που μπορεί να βοηθήσει στην έγκαιρη ανίχνευση και τον περιορισμό αυτών των αυτοματοποιημένων επιθέσεων. Εξετάζοντας την εισερχόμενη κίνηση, ένας οργανισμός μπορεί να εφαρμόσει με αποδοτικό τρόπο προληπτικά μέτρα ασφάλειας που θα προστατέψουν το σύνολο της πληροφοριακής του υποδομής.

Οι πρώτες προσπάθειες για τον εντοπισμό shellcode σε δικτυακή κίνηση αφορούσαν στον εντοπισμό συγκεκριμένων byte («υπογραφών») από γνωστά exploit στα διερχόμενα πακέτα. Η μέθοδος αυτή, αν και εξαιρετικά αποδοτική (ειδικά σε δίκτυα υψηλών ταχυτήτων), μπορεί να εντοπίσει ίχνη μόνο από γνωστές μορφές κακόβουλου λογισμικού. Ένα παράδειγμα αυτής της τεχνολογίας είναι το σύστημα Buttercup της εργασίας [86] το οποίο εντοπίζει ακεραίους που αντιστοιχούν σε διευθύνσεις επιστροφής από exploit που σχετίζονται με γνωστές τρωτότητες. Ένα άλλο παράδειγμα είναι ο μηχανισμός εντοπισμού γνωστών NOP sled και payload (αλλά και δεδομένων σχετικών με γνωστές τρωτότητες) που χρησιμοποιεί το σύστημα ανίχνευσης επιθέσεων Snort [87].



Σχήμα 3.4: Υπερχείλιση στοίβας και εκτέλεση πολυμορφικού shellcode

Οι επιτιθέμενοι για να αποφύγουν τον εντοπισμό από τέτοιου είδους συστήματα χρησιμοποιούν δύο τεχνικές μετασχηματισμού του shellcode: το *μεταμορφισμό* και τον *πολυμορφισμό*. Τα μεταμορφικά NOP sled και payload αποτελούν εναλλακτικές υλοποιήσεις οι οποίες αν εκτελεστούν φέρουν το ίδιο αποτέλεσμα με την αρχική τους μορφή. Ένα πολυμορφικό payload «αποκρύπτει» τον πραγματικό του κώδικα μέσω κρυπτογράφησης. Η ρουτίνα που θα αποκρυπτογραφήσει και θα εκτελέσει τον πραγματικό κώδικα μπορεί να υλοποιηθεί μεταμορφικά ώστε το σύνολο του payload (κρυπτογραφημένο μέρος, ρουτίνα αποκρυπτογράφησης και κενό διάστημα – padding) να μη μπορεί να εντοπιστεί βάσει κάποιας «υπογραφής». Στο σχήμα 3.4 παρουσιάζεται ένα παράδειγμα πολυμορφικού shellcode. Η έννοιες του μεταμορφισμού και του πολυμορφισμού έχουν προκύψει από αντίστοιχες τεχνικές που χρησιμοποιήθηκαν στο παρελθόν από συγγραφείς «ιομορφικού» λογισμικού (computer viruses). Στις εργασίες [88, 89] διαπιστώνεται η υπολογιστική δυσκολία (np-complete) του προβλήματος ανίχνευσης ενός γνωστού πολυμορφικού/μεταμορφικού «ιού» ακόμη και όταν η διαδικασία ανίχνευσης γνωρίζει τη γραμματική στην οποία ανήκει η νέα μορφή του «ιού».

Οι εργασίες [90] και [91] προσφέρουν μια διαφορετική αντιμετώπιση στο πρόβλημα της

ανίχνευσης επιθέσεων προς διαδικτυακές υπηρεσίες. Συγκεκριμένα, αντί να ανιχνεύσουν τα byte μιας επίθεσης (π.χ. ίχνη SQL injection, shellcode κλπ.), εντοπίζουν δικτυακή κίνηση η οποία παρεκκλίνει σημαντικά από την προβλεπόμενη (anomaly detection). Αυτή η κίνηση θα μπορούσε να θεωρηθεί ως σημάδι επίθεσης και θα μπορούσαν να ενεργοποιηθούν κατάλληλοι μηχανισμοί ώστε να ενημερωθεί ο διαχειριστής και να ληφθούν τα κατάλληλα μέτρα ασφάλειας (intrusion prevention system). Δυστυχώς τα συστήματα αυτά απαιτούν προηγούμενη εκπαίδευση προκειμένου να ελαττωθεί ο αριθμός των εσφαλμένων ειδοποιήσεων (false positives) σχετικά με επιθέσεις. Επίσης, δεν μπορούν να εντοπίσουν επιθέσεις όπου το shellcode έχει μετασχηματισθεί κατάλληλα ώστε να έχει τα ίδια στατιστικά χαρακτηριστικά με την προβλεπόμενη κίνηση [92, 93].

Από τα παραπάνω γίνεται εμφανές ότι το πολυμορφικό κακόβουλο λογισμικό (γνωστό και μη) δε μπορεί να εντοπιστεί εύκολα από τη στατική ανάλυση των byte που περιέχονται στα διερχόμενα πακέτα. Για το λόγο αυτό, νεότερες εργασίες όπως η [94] προχωρούν στη δυναμική ανάλυση των περιεχομένων των πακέτων, θεωρώντας τα περιεχόμενα αυτών «εν δυνάμει» κακόβουλο λογισμικό. Συγκεκριμένα, τοποθετούν τα byte κάθε πακέτου σε περιβάλλον εικονικής εκτέλεσης εντολών επεξεργαστή (CPU emulator) και δοκιμάζουν την εκτέλεση των δεδομένων (σαν να ήταν εντολές επεξεργαστή) από κάθε δυνατή θέση (offset). Αν κατά την εκτέλεση εντοπιστεί κάποια «ύποπτη» συνθήκη (π.χ. δραστηριότητα που παραπέμπει σε αποκρυπτογράφηση ή getPC), τότε ενημερώνεται το σχετικό σύστημα εντοπισμού επιθέσεων για το γεγονός αυτό. Η δυναμική ανάλυση αυτής της μορφής έχει βρει εφαρμογή και σε λογισμικό ασφάλειας που εγκαθίσταται σε εξυπηρετητές. Στην εργασία [95] χρησιμοποιείται για τη συλλογή νέου κακόβουλου λογισμικού στο πλαίσιο ενός *honeynet*, ενώ στην εργασία [96] χρησιμοποιείται για τον εντοπισμό συνθηκών εκμετάλλευσης τρωτοτήτων σε εξυπηρετητές. Παρόλα αυτά η χρήση εξομοίωσης για τον εντοπισμό shellcode σε επίπεδο δικτύου παραμένει πρωτίστης σημασίας καθώς μπορεί να προστατεύσει το σύνολο της πληροφοριακής υποδομής που βρίσκεται πίσω από το σύστημα ελέγχου της κίνησης.

Στην ενότητα που ακολουθεί θα παρουσιαστεί ένα νέο σύστημα εντοπισμού shellcode σε δικτυακή κίνηση το οποίο εκμεταλλεύεται τις δυνατότητες μιας ελεύθερα προγραμματιζόμενης υποδομής που βρίσκεται στα άκρα του δικτύου. Σε αντίθεση με το ενσωματωμένο σύστημα της εργασίας [94], το σύστημα αυτό μπορεί να υλοποιηθεί χρησιμοποιώντας κοινό υλικό και έχει τη δυνατότητα οριζόντιας επέκτασης προκειμένου να καλύψει τις ανάγκες δικτύων υψηλών ταχυτήτων. Επίσης, ο εντοπισμός του shellcode δε βασίζεται σε συνθήκες που προκύπτουν στη μνήμη του λογισμικού, αλλά σε συνθήκες που προκύπτουν μεταξύ αυτού και του λειτουργικού συστήματος. Το παραπάνω χαρακτηριστικό σε συνδυασμό με τη χρήση ενός προηγμένου περιβάλλοντος εξομοίωσης, επιτρέπουν στην πλατφόρμα να εντοπίσει περισσότερα (γνωστά και μη) πολυμορφικά shellcode από οποιαδήποτε άλλη δικτυακή πλατφόρμα έχει προταθεί μέχρι στιγμής στη βιβλιογραφία.

3.2 Ανίχνευση πολυμορφικού κακόβουλου λογισμικού σε δικτυακή κίνηση με την πλατφόρμα SEDUCE

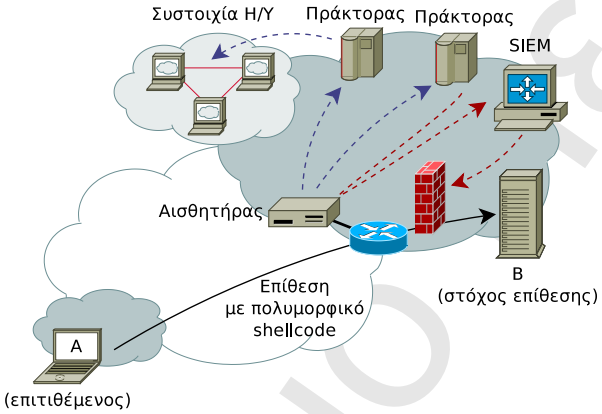
Η πλατφόρμα SEDUCE (Shellcode Detection Using CPU Emulation) συνδυάζει τον ελεύθερο προγραμματισμό των ισχυρών ενεργών δικτύων με την ισχυρή (και επεκτάσιμη) επεξεργαστική δυνατότητα των συστημάτων στα άκρα ενός δικτύου, προκειμένου να δώσει μια αποδοτική λύση στο πρόβλημα εντοπισμού κακόβουλου λογισμικού (shellcode) σε δίκτυα υψηλών ταχυτήτων.

Η πλατφόρμα αποτελείται από δύο τύπους κόμβων: τους *Αισθητήρες* και τους *Πράκτο-*

ρες. Οι Αισθητήρες συλλέγουν τη δικτυακή κίνηση που θα διερευνηθεί για ίχνη shellcode, ενώ οι Πράκτορες πραγματοποιούν τη διερεύνηση. Οι Πράκτορες έχουν τη δυνατότητα να συνεργαστούν και με συστοιχίες υπολογιστών ώστε να κατανεμηθεί ακόμη περισσότερο ο φόρτος εργασίας της διαδικασίας διερεύνησης.

Όταν ένας Πράκτορας εντοπίσει κάποιο shellcode, τότε ενημερώνεται ο Αισθητήρας που συνέλεξε τα αντίστοιχα πακέτα και αποστέλλεται σχετικό δελτίο στο Σύστημα Διαχείρισης Περιστατικών Ασφάλειας (Security Information & Event Management system – SIEM) του οργανισμού. Ανάλογα με την τοπική πολιτική ασφάλειας, το σύστημα αυτό μπορεί απλά να ειδοποιήσει το διαχειριστή και τον υπεύθυνο ασφάλειας ή να προβεί στην εφαρμογή προληπτικών μέτρων για την προστασία της υποδομής από τον επιτιθέμενο (π.χ. ενεργοποίηση σχετικών ρυθμίσεων στο firewall του οργανισμού).

Στο διάγραμμα 3.5 παρουσιάζεται συνοπτικά η αρχιτεκτονική της πλατφόρμας SEDUCE.



Σχήμα 3.5: Τοπολογία της αρχιτεκτονικής SEDUCE

Οι Αισθητήρες τοποθετούνται τυπικά σε κομβικά σημεία του δικτύου ενός οργανισμού προκειμένου να έχουν πρόσβαση σε όσο το δυνατόν περισσότερη εισερχόμενη κίνηση. Ο διαχειριστής μπορεί να ορίσει μια σειρά φίλτρων ώστε να συλλεχθεί συγκεκριμένου τύπου δικτυακή κίνηση. Η κίνηση αυτή στη συνέχεια θα διαμορφωθεί σε «πακέτα εργασίας» τα οποία θα κληθούν να επεξεργαστούν οι Πράκτορες. Κάθε Αισθητήρας μπορεί να συνεργαστεί με πολλαπλούς Πράκτορες και έχει τη δυνατότητα να διατηρήσει στη μνήμη του ένα (περιορισμένο) αριθμό πακέτων εργασίας έως ότου αυτά ζητηθούν από κάποιο Πράκτορα στο προσεχές μέλλον. Τα πακέτα εργασίας ομαδοποιούνται από τους Αισθητήρες κατά τέτοιο τρόπο ώστε επόμενα πακέτα της ίδιας συνεδρίας να είναι διαθέσιμα στον Πράκτορα που έχει αναλάβει τη διερεύνηση αυτής.

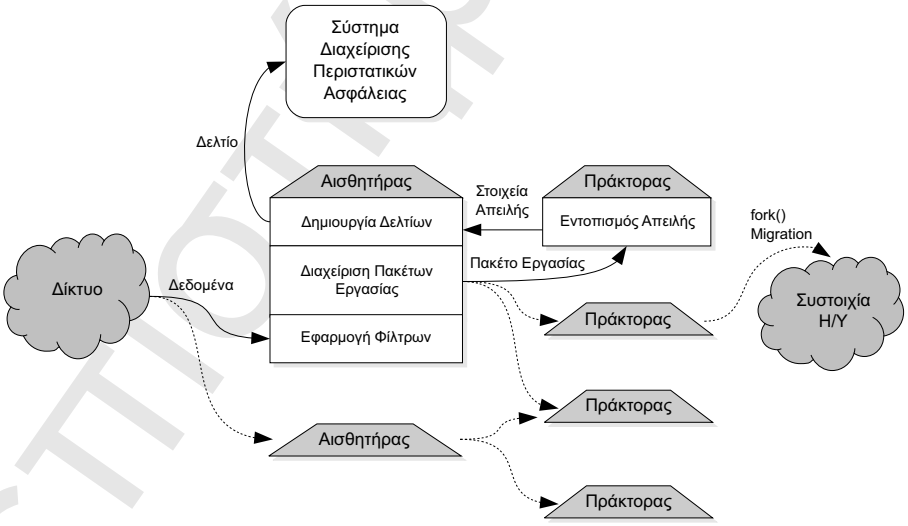
Οι Πράκτορες παίζουν το ρόλο των Ενεργών Κόμβων στην πλατφόρμα SEDUCE. Έχουν τη μορφή εφαρμογών και παρέχουν το περιβάλλον εκτέλεσης που απαιτείται για την διερεύνηση των πακέτων εργασίας. Ως εφαρμογές μπορεί να τοποθετηθούν σε οποιοδήποτε σημείο της πληροφοριακής υποδομής ενός οργανισμού αλλά και να συστεγαστούν με άλλες υπηρεσίες σε εξυπηρετητές. Προκειμένου να αποφευχθούν τυχόν διαρροές ευαίσθητων δεδομένων σε μη πιστοποιημένους Πράκτορες (ή άλλο κακόβουλο λογισμικό), ακολουθείται μια διαδικασία ταυτοποίησης των Πρακτόρων στους Αισθητήρες προτού ξεκινήσει η οποιαδήποτε προώθηση δεδομένων σε αυτούς.

Κάθε Πράκτορας ζητά διαρκώς νέα πακέτα εργασίας από μια ομάδα Αισθητήρων. Μόλις ένα πακέτο γίνει διαθέσιμο, εξετάζεται από μια σειρά μηχανισμών ανίχνευσης (detection engines) πολυμορφικού shellcode. Επειδή οι διαδικασίες των μηχανισμών ανίχνευσης ενδέχεται να είναι χρονοβόρες, η διεργασία του Πράκτορα δημιουργεί θυγατρικές διεργασίες

για την εξέταση κάθε πακέτου. Η χρήση θυγατρικών διεργασιών επιτρέπει τόσο την αποδοτικότερη χρήση των πολυεπεξεργαστικών δυνατοτήτων των συστημάτων που φιλοξενούν τους Πράκτορες όσο και την προώθηση εργασιών σε συστοιχίες υπολογιστών που υποστηρίζουν την τεχνολογία *fork migration* [97]. Εξάλλου η εξέταση πακέτων από πολλαπλές θέσεις (offsets) είναι μια πλήρως παραλληλοποιήσιμη διαδικασία (embarrassingly parallel task) καθώς δεν απαιτεί κάποιο συγχρονισμό μεταξύ των διεργασιών που θα εκτελέσουν την εξέταση. Συνεπώς, η χρήση πολυεπεξεργαστικών συστημάτων και συστοιχιών υπολογιστών για τη διερεύνηση πακέτων εργασίας έρχεται ως μια φυσική λύση στο πρόβλημα καταμερισμού του φόρτου εργασίας των Πρακτόρων.

Η δυναμική ανάλυση πακέτων εργασίας στην πλατφόρμα SEDUCE στηρίζεται στον εντοπισμό εντολών επεξεργαστή οι οποίες (εντός ενός συγκεκριμένου χρονικού διαστήματος) εκτελούν επιτυχώς κάποια «επικίνδυνη» κλήση συστήματος. Παράδειγμα μιας τέτοιας κλήσης συστήματος είναι η κλήση `open` η οποία δίνει σε μια εφαρμογή πρόσβαση σε κάποιο αρχείο. Αν εντοπιστεί μια επικίνδυνη κλήση συστήματος τότε τερματίζεται η εικονική εκτέλεση του πακέτου εργασίας και ειδοποιείται ο Αισθητήρας που ήταν υπεύθυνος για το ύποπτο πακέτο. Η παραπάνω προσέγγιση μειώνει σημαντικά την πιθανότητα να χαρακτηριστεί ως κακόβουλο λογισμικό μια σειρά από τυχαία byte δεδομένων, καθώς η εκτέλεση συγκεκριμένων κλήσεων συστήματος προϋποθέτει τη διαμόρφωση του περιβάλλοντος εκτέλεσης με ένα πολύ συγκεκριμένο τρόπο. Η διαμόρφωση αυτή είναι πρακτικά αδύνατο να προκύψει από την εκτέλεση τυχαίων byte ή byte δεδομένων. Επίσης, εφόσον έχει κριθεί ότι τα byte αυτά είναι εκτελέσιμα, τότε η ίδια η εκτέλεση μιας επικίνδυνης κλήσης συστήματος τα χαρακτηρίζει ως «εν δυνάμει» επικίνδυνο λογισμικό.

Η παραπάνω προσέγγιση σε συνδυασμό με τη χρήση ενός περιβάλλοντος εκτέλεσης που εξομοιώνει με υψηλή πιστότητα τη λειτουργία ενός επεξεργαστή, επιτρέπουν στην πλατφόρμα SEDUCE να εντοπίζει περισσότερα πολυμορφικά shellcode σε δικτυακή κίνηση από οποιοδήποτε άλλο σύστημα έχει προταθεί μέχρι στιγμής στη βιβλιογραφία.



Σχήμα 3.6: Βασικά συστατικά της αρχιτεκτονικής SEDUCE

Στο σχήμα 3.6 παρουσιάζονται τα βασικά συστατικά του κατανεμημένου συστήματος εντοπισμού πολυμορφικού shellcode της πλατφόρμας SEDUCE.

Ο φόρτος εργασίας που θα πρέπει να διαχειριστεί η πλατφόρμα SEDUCE εξαρτάται άμεσα από τα χαρακτηριστικά της εισερχόμενης κίνησης (όγκος δεδομένων, ρυθμός παρα-

λαβής πακέτων), από την επεξεργαστική και αποθηκευτική ικανότητα των Αισθητήρων, από τον αριθμό και τις επεξεργαστικές δυνατότητες των Πρακτόρων και, φυσικά, από το είδος της εξέτασης που θα πραγματοποιηθεί στα πακέτα εργασίας. Η αρχιτεκτονική που παρουσιάστηκε στο παράδειγμα του σχήματος 3.5 μπορεί να επεκταθεί οριζόντια προκειμένου να καλύψει τις όποιες ανάγκες ενός οργανισμού με πολλαπλές διαδικτυακές υπηρεσίες και δίκτυα υψηλών ταχυτήτων. Συγκεκριμένα, υπάρχει η δυνατότητα να προστεθούν πολλαπλοί Αισθητήρες ώστε να καταμεριστεί ο φόρτος εργασίας που συνδέεται με τη συλλογή της κίνησης και την εφαρμογή φίλτρων σε αυτή. Επίσης, μπορεί να αυξηθεί ο αριθμός των Πρακτόρων ώστε να είναι εφικτή η εξέταση όλων των πακέτων εργασίας που δημιουργούνται από τους Αισθητήρες.

Πέραν από την οριζόντια επέκταση της πλατφόρμας για την κάλυψη των αναγκών μιας πληροφοριακής υποδομής, η πλατφόρμα SEDUCE είναι επεκτάσιμη και σε επίπεδο λογισμικού. Η αρχιτεκτονική των Πρακτόρων επιτρέπει την εύκολη εισαγωγή νέων μηχανισμών διερεύνησης της δικτυακής κίνησης, καθιστώντας την πλατφόρμα ιδανική για χρήση και σε ερευνητικά περιβάλλοντα εργασίας. Ακόμη, οι μηχανισμοί εξέτασης κακόβουλου λογισμικού της πλατφόρμας, μπορούν να χρησιμοποιηθούν και αυτόνομα π.χ. στα πλαίσια πειραμάτων σχετικών με ιομορφικό λογισμικό.

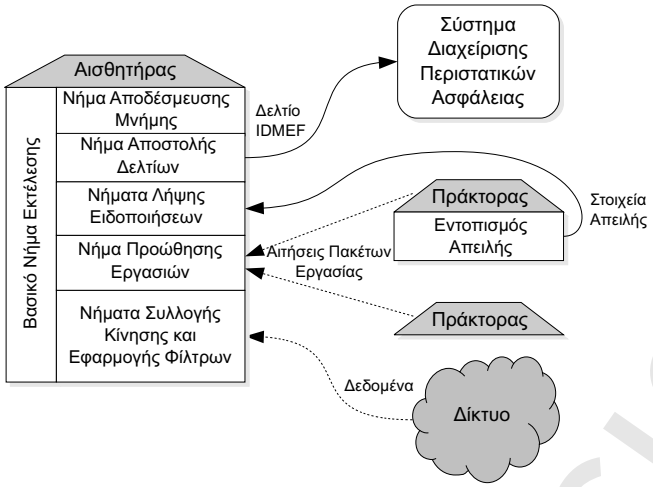
Στις ενότητες που ακολουθούν γίνεται μια πιο αναλυτική παρουσίαση της πλατφόρμας SEDUCE. Μεταξύ άλλων, παρουσιάζονται θέματα σχετικά με την ανάπτυξη νέων αρθρωμάτων για την πλατφόρμα και μελετώνται τρόποι με τους οποίους αυτή μπορεί να αξιοποιηθεί για την ανίχνευση επιθέσεων σε ειδικά πρωτόκολλα εφαρμογών. Τέλος, μέσα από την πειραματική και ποιοτική αξιολόγηση της πλατφόρμας σκιαγραφούνται τα ιδιαίτερα χαρακτηριστικά αυτής και τονίζονται τα πλεονεκτήματα που φέρουν οι προγραμματιζόμενες υποδομές στο χώρο της ασφάλειας τηλεπικοινωνιακών υποδομών.

3.3 Αρχιτεκτονική Αισθητήρων

Ο Αισθητήρας της πλατφόρμας SEDUCE είναι μια πολυνηματική εφαρμογή η οποία συλλέγει συγκεκριμένου τύπου δικτυακή κίνηση. Η κίνηση αυτή μετασχηματίζεται σε *πακέτα εργασίας* και παραδίδεται σε Πράκτορες προς εξέταση. Όταν κάποιος Πράκτορας διαπιστώσει ίχνη κακόβουλου λογισμικού σε κάποιο πακέτο εργασίας, τότε αποστέλλει ειδοποίηση στο σχετικό Αισθητήρα και αυτός με τη σειρά του ενημερώνει το Σύστημα Διαχείρισης Περιστατικών Ασφάλειας (SIEM). Η χρήση πολλαπλών νημάτων για την εκτέλεση των παραπάνω διαδικασιών επιτρέπει την καλύτερη αξιοποίηση των πολυεπεξεργαστικών δυνατοτήτων του σύγχρονου υλικού καθώς και την αποδοτικότερη επεξεργασία της εισερχόμενης δικτυακής κίνησης.

Όπως φαίνεται στο σχήμα 3.7, η εφαρμογή του Αισθητήρα αποτελείται από τα εξής νήματα:

- το κύριο νήμα εκτέλεσης της εφαρμογής,
- τα νήματα συλλογής δικτυακής κίνησης και εφαρμογής φίλτρων σε αυτή,
- το νήμα προώθησης εργασιών στους Πράκτορες,
- τα νήματα λήψης ειδοποιήσεων από τους Πράκτορες,
- το νήμα αποστολής δελτίων ασφάλειας στο Σύστημα Διαχείρισης Περιστατικών Ασφάλειας (SIEM),
- και το νήμα αποδέσμευσης μνήμης (“out of memory” handler).



Σχήμα 3.7: Πολυνηματική αρχιτεκτονική Αισθητήρα

Στην πρότυπη υλοποίηση της πλατφόρμας SEDUCE, η συλλογή πακέτων από το δίκτυο υλοποιείται μέσω της βιβλιοθήκης libpcap [98]. Στη βιβλιογραφία έχουν προταθεί διάφορες μέθοδοι για τη βελτίωση της ταχύτητας συλλογής πακέτων, μερικές από τις οποίες εκμεταλλεύονται ειδικά διαμορφωμένους οδηγούς συσκευών ώστε η κίνηση να παρακάμψει τη στοίβα πρωτοκόλλων και να φτάσει απευθείας στον Αισθητήρα (βλ. PF_RING [99]), ενώ άλλες εκμεταλλεύονται την πολυεπεξεργαστική ικανότητα των σύγχρονων επεξεργαστών με πολλαπλούς πυρήνες χρησιμοποιώντας ουρές που δεν απαιτούν κάποιο συγχρονισμό (βλ. PFQ [100]). Οι μέθοδοι αυτές επιτρέπουν σε ένα Αισθητήρα υλοποιημένο με μη εξειδικευμένο υλικό (commodity hardware) να συλλέξει δεδομένα με ταχύτητα έως και 42 εκατομμύρια πακέτα το δευτερόλεπτο. Η επεξεργαστική δυνατότητα αυτή είναι συχνά αναγκαία προκειμένου να καλυφθούν οι ανάγκες δικτύων υψηλών ταχυτήτων.

Τα δεδομένα που ενδιαφέρουν τον Αισθητήρα, βρίσκονται στο εσωτερικό πακέτων TCP και UDP. Τα δεδομένα αυτά ενδέχεται να φτάσουν στον Αισθητήρα με κατακερματισμένη μορφή (fragmented) είτε εξαιτίας κάποιου κατακερματισμού που συνέβη στο δίκτυο, είτε εξαιτίας κάποιου κατακερματισμού που συνέβη στην εφαρμογή, είτε εξαιτίας «δόλιου» κατακερματισμού που εφαρμόστηκε από κακόβουλο χρήστη προκειμένου το περιεχόμενο των πακέτων να μη γίνει αντιληπτό από κάποιο σύστημα ανίχνευσης επιθέσεων. Σε κάθε περίπτωση, τα επί μέρους δεδομένα θα πρέπει να συλλεχθούν ώστε να εξεταστούν στη μορφή που είχαν στο αρχικό πακέτο (πριν τον κατακερματισμό). Επίσης, πολλά από τα πακέτα που λαμβάνει ένας Αισθητήρας αφορούν στην κατάσταση ροών δεδομένων (π.χ. βοηθητικά πακέτα που επιβεβαιώνουν τη λήψη δεδομένων σε TCP συνδέσεις). Τα πακέτα αυτά ο Αισθητήρας θα πρέπει να τα λάβει υπόψιν ώστε να διαμορφώσει σωστή «εικόνα» επί των δεδομένων των ροών. Τόσο στην περίπτωση του κατακερματισμού όσο και στην περίπτωση των βοηθητικών πακέτων, ο Αισθητήρας καλείται να επεξεργαστεί τα εισερχόμενα πακέτα με τον ίδιο τρόπο που τα επεξεργάζεται μια στοίβα πρωτοκόλλων TCP/IP. Η libnids [101] είναι μια βιβλιοθήκη που παρέχει αυτή τη λειτουργικότητα στους Αισθητήρες της πλατφόρμας SEDUCE.

Ο Αισθητήρας εφαρμόζει φίλτρα στην εισερχόμενη κίνηση ώστε να διατηρήσει μόνο τα απολύτως απαραίτητα πακέτα. Τα φίλτρα αυτά εφαρμόζονται σε τρία διαφορετικά επίπεδα και λειτουργούν συνεργατικά μεταξύ τους. Στο πρώτο επίπεδο επιλέγονται τα πακέτα που θα πρέπει να συλλεχθούν από το δίκτυο και να προωθηθούν στη libnids για περαιτέρω επεξεργασία. Ο χρήστης μπορεί να ορίσει σε αυτό το επίπεδο την ομάδα των μηχανημάτων

που θα προστατεύεται από την πλατφόρμα SEDUCE («οικείο δίκτυο»). Τα σχετικά φίλτρα περιγράφονται στο αρχείο ρυθμίσεων του Αισθητήρα και μεταφράζονται σε εκφράσεις τύπου BPF (BSD packet filter [102]). Η εφαρμογή των φίλτρων BPF γίνεται σε ξεχωριστό νήμα ώστε να μην επιβαρύνεται ο επεξεργαστής που πραγματοποιεί τη συλλογή πακέτων από το δίκτυο. Επίσης, για να επιταχυνθεί η διαδικασία εξέτασης πολλαπλών πακέτων ως προς μια συγκεκριμένη σειρά φίλτρων, μπορεί να αξιοποιηθεί μια δομή τύπου *bloom filter* όπως προτείνεται στην εργασία [103]. Η δομή αυτή επιτρέπει στον Αισθητήρα να εντοπίσει με αποδοτικό τρόπο τα πακέτα που δεν ανήκουν στο σύνολο αυτών με τα επιθυμητά χαρακτηριστικά. Στο δεύτερο επίπεδο εφαρμογής φίλτρων ο Αισθητήρας επιλέγει από την έξοδο της libnids τα δεδομένα που προορίζονται για τα προστατευόμενα συστήματα. Έτσι π.χ. η απάντηση ενός προστατευόμενου εξυπηρετητή προς ένα αίτημα χρήστη, δε θα διατηρηθεί από τον Αισθητήρα. Η επιλογή αυτή περιορίζει σημαντικά τον όγκο των δεδομένων που θα κληθούν να εξετάσουν οι Πράκτορες. Στο τρίτο και τελευταίο επίπεδο εφαρμογής φίλτρων ο χρήστης μπορεί να περιορίσει περαιτέρω τα προς εξέταση πακέτα, ορίζοντας το είδος των υπηρεσιών (θύρες, τύπος σύνδεσης) που θα προστατεύονται από την πλατφόρμα SEDUCE.

Οι Πράκτορες βρίσκονται διαρκώς σε ένα ατέρμονα βρόχο, κατά τον οποίο ζητούν από τους Αισθητήρες νέα *πακέτα εργασίας* (work units). Το κάθε πακέτο εργασίας περιλαμβάνει δεδομένα που βρέθηκαν σε πακέτα TCP ή UDP (μετά το στάδιο επεξεργασίας από τη libnids). Υπεύθυνος για τη δημιουργία των πακέτων εργασίας είναι ο Αισθητήρας, ο οποίος τα αποθηκεύει σε ειδική λίστα στη μνήμη. Τα πακέτα εργασίας συνδέονται με *ομάδες δεδομένων* (data groups). Οι ομάδες δεδομένων περιγράφουν πακέτα που έχουν κάποια σχέση μεταξύ τους και στα οποία μπορεί να χρειαστεί να ανατρέξει ένας Αισθητήρας. Για λόγους απόδοσης, η ομάδα δεδομένων σε μια συνεδρία TCP περιλαμβάνει όλα τα πακέτα που έστειλε ένας χρήστης μέχρι να λάβει κάποια απάντηση από το διακομιστή. Αντίστοιχα, για κάθε πακέτο UDP δημιουργείται μια ξεχωριστή ομάδα δεδομένων. Με τον τρόπο αυτό, η κάθε δικτυακή συνεδρία χωρίζεται σε πολλαπλές ομάδες δεδομένων, η κάθε μία από τις οποίες μπορεί να εξεταστεί από διαφορετικό Πράκτορα.

Για να αποφευχθούν προβλήματα κατακερματισμού στο σωρό (heap fragmentation), η εφαρμογή αποθηκεύει πολλές από τις δομές αυτής σε πίνακες ήδη δεσμευμένης μνήμης (slab memory allocation). Επίσης, για τη γρηγορότερη εξυπηρέτηση των Πρακτόρων, οι ομάδες δεδομένων διατηρούνται σε λίστα τύπου LIFO (last in first out). Χρησιμοποιώντας μια συνάρτηση κατακερματισμού, ο Αισθητήρας μπορεί να εντοπίσει άμεσα τη δυναμική δομή που αντιστοιχεί στην ομάδα δεδομένων που επεξεργάζεται ένας Πράκτορας.

Μόλις εντοπιστεί κάποιο ίχνος κακόβουλου λογισμικού, ένα νέο νήμα θα αναλάβει να μετασχηματίσει τα στοιχεία της απειλής που δόθηκαν από τον Πράκτορα σε μορφή κατάλληλη για υποβολή στο Σύστημα Διαχείρισης Περιστατικών Ασφάλειας (SIEM). Η πρότυπη υλοποίηση της πλατφόρμας SEDUCE συνεργάζεται με το SIEM ανοιχτού λογισμικού prelude [104], στο οποίο υποβάλλονται δελτία σχετικά με κακόβουλο λογισμικό, σύμφωνα με το πρότυπο IDMEF [105]. Η δυνατότητα συνεργασίας με ένα σύστημα τύπου SIEM επιτρέπει στο διαχειριστή να συνδέσει την πλατφόρμα SEDUCE με την υπόλοιπη υποδομή ασφάλειας του οργανισμού και του παρέχει επίσης τη δυνατότητα κεντρικής εποπτείας επί του συνόλου των μηχανισμών ανίχνευσης εισβολών που μπορεί να διαθέτει ο οργανισμός.

Όταν ο ρυθμός με τον οποίο επεξεργάζονται πακέτα οι Πράκτορες είναι μικρότερος του ρυθμού συλλογής πακέτων ενός Αισθητήρα, τότε είναι πιθανή η εξάντληση της μνήμης του Αισθητήρα από πακέτα εργασίας που δεν έχουν τύχει εξέτασης. Προκειμένου σε τέτοιες περιπτώσεις να συνεχιστεί κανονικά η συλλογή πακέτων, η αρχιτεκτονική των Αισθητήρων περιλαμβάνει ένα ειδικό νήμα το οποίο είναι υπεύθυνο για την αποδέσμευση μνήμης (out-of-memory handler). Το νήμα αυτό εξετάζει αν η ποσότητα δεσμευμένης μνήμης ξε-

περνά ένα *ανώτατο όριο* (hard upper limit) και αν κάτι τέτοιο ισχύει τότε αφαιρεί τις παλαιότερες ομάδες δεδομένων έως ότου η ποσότητα δεσμευμένης μνήμης γίνει μικρότερη ενός *ανώτερου ορίου* (soft upper limit).

Τέλος, σε δίκτυα υψηλών ταχυτήτων, όπου υπάρχει μεγάλη επιφόρτιση από τον υψηλό ρυθμό άφιξης πακέτων, είναι δυνατή η σύνδεση πολλαπλών Αισθητήρων παράλληλα (ενδεχομένως εφαρμόζοντας και διαφορετικά φίλτρα) προκειμένου να γίνει επιμερισμός του φόρτου εργασίας που σχετίζεται με τη συλλογή και την επεξεργασία των πακέτων.

3.4 Αρχιτεκτονική Πρακτόρων

Ο Πράκτορας της πλατφόρμας SEDUCE είναι υπεύθυνος για την ανίχνευση κακόβουλου λογισμικού σε πακέτα εργασίας. Τα πακέτα εργασίας παρέχονται από μια ομάδα Αισθητήρων, τα μέλη της οποίας περιγράφονται στο αρχείο ρυθμίσεων κάθε Πράκτορα. Ο τρόπος με τον οποίο επιλέγει ένας Πράκτορας από ποιον Αισθητήρα θα αναζητήσει ένα νέο πακέτο εργασίας περιγράφεται επίσης στο αρχείο ρυθμίσεων και μπορεί να είναι είτε τυχαίος (random selection) είτε σειριακός (round-robin selection).

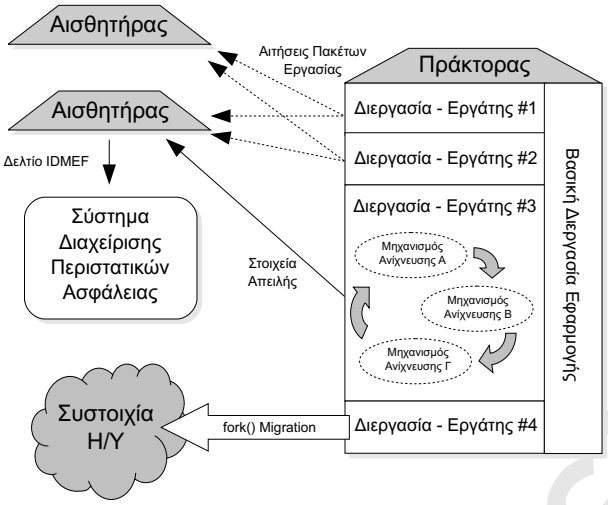
Ο Πράκτορας βρίσκεται σε μια διαρκή κατάσταση αναζήτησης νέων πακέτων εργασίας. Όταν έχει ολοκληρώσει την εξέταση ενός πακέτου τότε θα αναζητήσει άμεσα ένα νέο πακέτο. Αν ο Αισθητήρας με τον οποίο επικοινώνησε δεν έχει κάποιο πακέτο διαθέσιμο τότε θα κάνει μία ακόμη αίτηση σε αυτόν μετά από χρονικό διάστημα w δευτερολέπτων, προτού αποστείλει αίτηση σε άλλο Αισθητήρα. Αν μετά από k προσπάθειες δεν έχει βρεθεί κάποιο πακέτο εργασίας, τότε μένει ανενεργός για διάστημα w δευτερολέπτων. Οι τιμές του χρονικού διαστήματος w καθώς και του μέγιστου αριθμού προσπαθειών k μπορεί να διαμορφωθούν από το αρχείο ρυθμίσεων του Πράκτορα.

Η αναζήτηση, λήψη και επεξεργασία ενός πακέτου εργασίας γίνεται από μια διεργασία του Πράκτορα που ονομάζεται *Εργάτης* (worker). Οι Εργάτες λειτουργούν ουσιαστικά ως ξεχωριστοί Πράκτορες που φιλοξενούνται στο ίδιο σύστημα και έχουν κοινές ρυθμίσεις. Αυξάνοντας τον αριθμό των Εργατών σε ένα Πράκτορα, αυξάνεται και ο αριθμός των πακέτων εργασίας που θα εξεταστούν παράλληλα. Φυσικά, μια τέτοια παραλληλοποίηση διεργασιών, για να είναι αποδοτική θα πρέπει να υποστηρίζεται κατάλληλα και από το υλικό (π.χ. πολυεπεξεργαστικά συστήματα ή συστήματα με πολλούς πυρήνες).

Ωστόσο, η αντιστοίχιση διεργασίας – Εργάτη μας επιτρέπει να εκμεταλλευτούμε και τις δυνατότητες συστημάτων συστοιχιών υπολογιστών, οι οποίες δίνουν τη δυνατότητα σε διεργασίες να μεταναστεύσουν και τελικά να εκτελεστούν σε τρίτα συστήματα που έχουν χαμηλό επεξεργαστικό φόρτο. Η μετανάστευση αυτή ονομάζεται *fork migration* και υποστηρίζεται από συστοιχίες υπολογιστών τύπου *Single System Image* [97]. Στις συστοιχίες αυτές ο κάθε κόμβος έχει ένα εικονικό αριθμό επεξεργαστών που αντιστοιχεί στο σύνολο των επεξεργαστών των συστημάτων της συστοιχίας. Ομοίως, έχει και μια εικονική ποσότητα μνήμης που αντιστοιχεί στο σύνολο της μνήμης των συστημάτων της συστοιχίας. Επειδή το έργο ενός Εργάτη είναι κυρίως υπολογιστικής φύσης και δεν απαιτεί πρόσβαση σε τοπικούς πόρους ή σε (αργές) συσκευές εισόδου/εξόδου (του συστήματος που φιλοξενεί τον Πράκτορα), μπορεί να γίνει ιδιαίτερα αποδοτικά, μεταφέροντας την αντίστοιχη διεργασία σε μια συστοιχία υπολογιστών αυτού του τύπου.

Η δυνατότητα εγκατάστασης του Πράκτορα σε κοινό υλικό (commodity hardware) καθώς και η δυνατότητα εκμετάλλευσης συστοιχιών υπολογιστών επιτρέπουν στην πλατφόρμα SEDUCE να επεκταθεί *οριζόντια*, ώστε να καλύψει τις ανάγκες δικτύων υψηλών ταχυτήτων.

Όπως φαίνεται στο σχήμα 3.8 κάθε Εργάτης επεξεργάζεται ένα πακέτο εργασίας μέσω κάποιων *Μηχανισμών Ανίχνευσης Κακόβουλου Λογισμικού*. Οι Μηχανισμοί αυτοί (detection



Σχήμα 3.8: Αρχιτεκτονική Πράκτορα με πολλαπλές διεργασίες

engines) αξιοποιούνται εντός της διεργασίας του Εργάτη και κρίνουν κατά πόσο μια σειρά από byte αποτελεί κακόβουλο λογισμικό. Μόλις εντοπιστούν ίχνη κακόβουλου λογισμικού ο Εργάτης ενημερώνει τον Αισθητήρα από τον οποίο παρελήφθησαν αυτά τα byte ώστε τελικά να αποσταλεί σχετικό Δελτίο ειδοποίησης στο Σύστημα Διαχείρισης Περιστατικών Ασφάλειας του οργανισμού. Το είδος των Μηχανισμών (στατική ανάλυση, δυναμική ανάλυση κ.α.) που θα εξετάσουν τα byte καθώς και η σειρά με την οποία αυτοί θα λειτουργήσουν ορίζονται στο αρχείο ρυθμίσεων του Πράκτορα.

Για να γίνει πιο εύκολη η συγγραφή Μηχανισμών Ανίχνευσης Κακόβουλου Λογισμικού δημιουργήθηκε μια προγραμματιστική διεπαφή (API) μέσω της οποίας κάθε Πράκτορας θα καλεί τις συναρτήσεις των Μηχανισμών Ανίχνευσης (βλ. σχήμα 3.9).

```
typedef struct _Threat{
        unsigned char *payload;
        size_t length;
        unsigned short severity;
        char *msg;
} Threat;

typedef struct _DetectionEngine {
        char *name;
        char *descr;
        int  (*init)(void);
        void (*destroy)(void);
        void (*reset)(void);
        int  (*process)(char *, size_t, Threat *);
        void *params;
} DetectionEngine;
```

Σχήμα 3.9: Προγραμματιστική διεπαφή (API) Πράκτορα / Μηχανισμού Ανίχνευσης Κακόβουλου Λογισμικού

Κάθε Μηχανισμός Ανίχνευσης στην παραπάνω διεπαφή υλοποιεί μια δομή τύπου `DetectionEngine`, η οποία περιλαμβάνει το όνομα του Μηχανισμού (name), την περιγραφή αυτού (description), δείκτη σε αποθηκευτικό χώρο για δεδομένα του Μηχανισμού (params), δείκτη στη συνάρτηση αρχικοποίησης του Μηχανισμού (init), δείκτη στη συνάρτηση κατα-

στροφής του Μηχανισμού (destroy), δείκτη στη συνάρτηση αρχικοποίησης του περιβάλλοντος εξέτασης του Μηχανισμού (reset), και δείκτη στη συνάρτηση εξέτασης πακέτου εργασίας (process). Στην περίπτωση όπου εντοπιστούν ίχνη κακόβουλου λογισμικού σε κάποια byte, ο Μηχανισμός Ανίχνευσης που τα εντόπισε καλείται να συμπληρώσει μια δομή δεδομένων (`Threat`) με τα στοιχεία της απειλής. Τα στοιχεία αυτά περιλαμβάνουν το χαρακτηρισμό της απειλής (msg), την επικινδυνότητα αυτής (severity), τα byte στα οποία βρέθηκε το ίχνος κακόβουλου λογισμικού (payload) καθώς και το μήκος αυτών (length).

Καθορίζοντας τον τρόπο με τον οποίο θα επικοινωνεί ο Πράκτορας με τους Μηχανισμούς Ανίχνευσης, γίνεται δυνατή η εκμετάλλευση των Μηχανισμών Ανίχνευσης από τρίτες εφαρμογές. Για παράδειγμα, στην πρότυπη υλοποίηση της πλατφόρμας SEDUCE περιέχεται μια εφαρμογή που μπορεί να χρησιμοποιήσει οποιονδήποτε από τους διαθέσιμους Μηχανισμούς Ανίχνευσης, ώστε να εντοπίσει κακόβουλο λογισμικό σε ένα αρχείο δεδομένων. Επίσης, σε επίπεδο υλοποίησης, έγινε εφικτή η αξιοποίηση των Μηχανισμών Ανίχνευσης από υψηλότερου επιπέδου περιβάλλοντα προγραμματισμού (π.χ. γλώσσες σεναρίου). Αλλά και το αντίστροφο, υλοποιήθηκαν δηλαδή Μηχανισμοί Ανίχνευσης σε γλώσσες σεναρίου.

Η πρότυπη υλοποίηση της πλατφόρμας SEDUCE περιέχει τέσσερις Μηχανισμούς Ανίχνευσης Κακόβουλου Λογισμικού. Ο πρώτος εξ αυτών είναι βασισμένος στο άρθρωμα ονόματι *fnord* του συστήματος ανίχνευσης δικτυακών εισβολών Snort [87]. Το άρθρωμα αυτό επεκτάθηκε κατάλληλα ώστε να μπορεί να εντοπίσει και νεώτερα NOP sled που παράγονται από το πιο διαδεδομένο λογισμικό αυτόματης παραγωγής shellcode, το λεγόμενο metasploit framework [106]. Αν το πλήθος των NOP byte που θα εντοπιστούν ξεπεράσει ένα κατώφλι, τότε ο Μηχανισμός Ανίχνευσης αυτός θα θεωρήσει ότι τα byte που εξετάζονται περιέχουν κάποιο NOP sled και, επομένως, θα ειδοποιήσει τον αντίστοιχο Αισθητήρα για την πιθανή ύπαρξη κάποιου shellcode σε αυτά.

Με παρόμοιο τρόπο λειτουργεί και ο δεύτερος Μηχανισμός Ανίχνευσης Κακόβουλου λογισμικού, μόνο που αυτή τη φορά η στατική ανάλυση των byte γίνεται σε γλώσσα υψηλότερου επιπέδου (Python) και εντοπίζονται NOP sled που παράγονται από το άρθρωμα `opty2` του λογισμικού metasploit. Το άρθρωμα `opty2` παράγει ακολουθίες από byte (βάσει ενός πίνακα) τα οποία μπορούν να χρησιμοποιηθούν ως NOP sled. Ο σχετικός Μηχανισμός Ανίχνευσης χρησιμεύει ως παράδειγμα για την υλοποίηση αρθρωμάτων στην πλατφόρμα SEDUCE σε γλώσσες προγραμματισμού υψηλού επιπέδου (π.χ. γλώσσες σεναρίου).

Ο τρίτος Μηχανισμός Ανίχνευσης Κακόβουλου Λογισμικού χρησιμοποιεί τεχνικές που συνδυάζουν τόσο τη στατική όσο και τη δυναμική ανάλυση δεδομένων. Συγκεκριμένα, μέσω της βιβλιοθήκης *libemu* [107] υλοποιεί τη μέθοδο ανίχνευσης shellcode που προτείνουν οι συγγραφείς της εργασίας [94]. Αρχικά, εντοπίζονται όλες οι θέσεις του πακέτου εργασίας που περιέχουν κώδικα τύπου getPC (βλ. ενότητα 3.1). Στη συνέχεια δημιουργείται ένα γράφος με όλα τα μονοπάτια εκτέλεσης και σημειώνεται το μήκος αυτών που περιλαμβάνουν getPC κώδικα. Τα μονοπάτια ταξινομούνται ως προς το μέγεθός τους και επιλέγεται αυτό με το μεγαλύτερο μήκος. Αν το μήκος αυτό ξεπερνά ένα κατώφλι (σε αριθμό εντολών επεξεργαστή) τότε το πακέτο εργασίας θεωρείται ύποπτο και ενημερώνεται ο σχετικός Αισθητήρας.

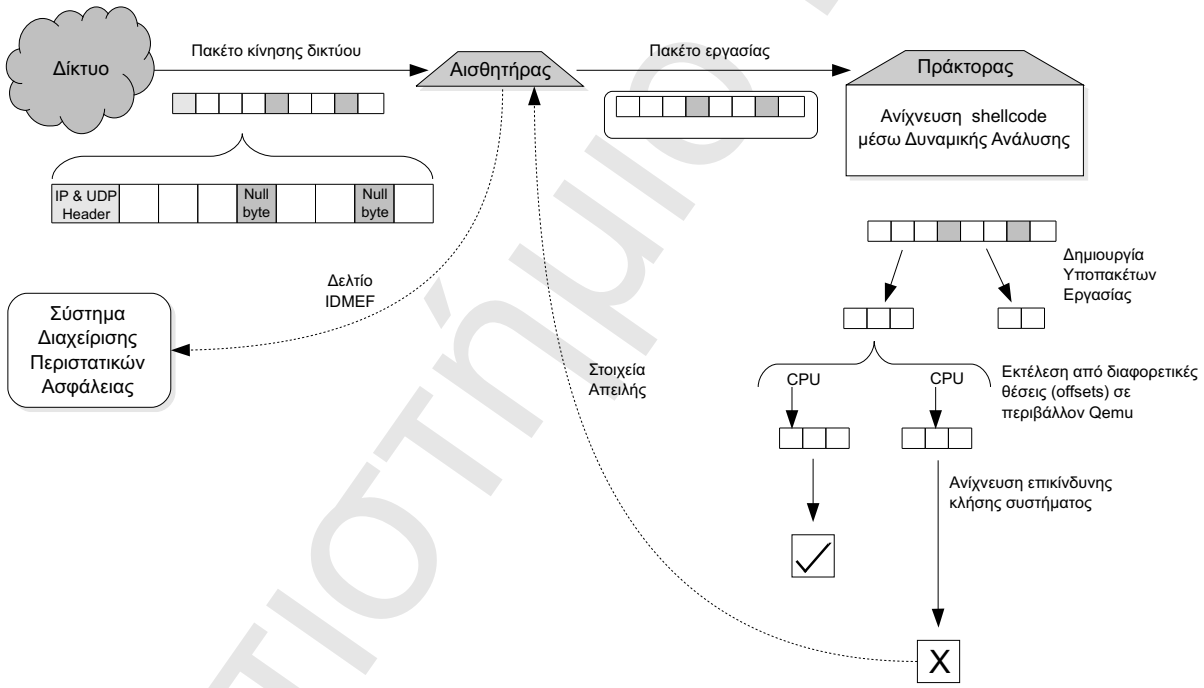
Ο τέταρτος και πιο σημαντικός Μηχανισμός Ανίχνευσης Κακόβουλου Λογισμικού βασίζεται στη δυναμική ανάλυση των περιεχομένων των πακέτων εργασίας και συγκεκριμένα στην εικονική εκτέλεση αυτών σε περιβάλλον *QEMU* [108]. Το λογισμικό QEMU εξομοιώνει με υψηλή πιστότητα τη λειτουργία διαφόρων επεξεργαστών και αρχιτεκτονικών υλικού. Στο πλαίσιο του παρόντος Μηχανισμού Ανίχνευσης παρέχεται ένα προστατευμένο περιβάλλον εκτέλεσης (sandbox) μέσα στο οποίο μπορούν να εκτελεστούν τα byte ενός πακέτου εργασίας, σαν αυτά να αποτελούσαν μέρος μιας εφαρμογής (userspace emulation).

Για τις ανάγκες της πλατφόρμας SEDUCE το QEMU διαμορφώθηκε κατάλληλα ώστε:

- να μπορεί να χρησιμοποιηθεί ως βιβλιοθήκη για την εξέταση μιας σειράς από byte,

- να μη χρησιμοποιεί τεχνικές caching κατά την εκτέλεση των byte (καθώς δημιουργούνται προβλήματα όταν το υπό διερεύνηση λογισμικό μεταλλάσσει τον εαυτό του),
- να παρέχει πληροφορίες ως προς τις κλήσεις συστήματος που εκτέλεσε το υπό διερεύνηση λογισμικό, και
- να παρέχει καλύτερη υποστήριξη για εντολές μαθηματικού συνεπεξεργαστή (καθώς χρησιμοποιούνται συχνά σε κώδικα τύπου getPC).

Ο τρόπος με τον οποίο λειτουργεί ο τέταρτος Μηχανισμός Ανίχνευσης παρουσιάζεται συνοπτικά στο σχήμα 3.10. Το πρώτο στάδιο του Μηχανισμού Ανίχνευσης είναι προαιρετικό και ρυθμίζεται από το αρχείο ρυθμίσεων του Πράκτορα. Συγκεκριμένα, επειδή συνηθίζεται στα shellcode να μην περιέχονται NULL byte (ώστε αυτά να μπορούν να χρησιμοποιηθούν και σε τρωτότητες όπου χρησιμοποιείται η συνάρτηση *strcpy* της βασικής βιβλιοθήκης της C), ο Μηχανισμός Ανίχνευσης χωρίζει το αρχικό πακέτο εργασίας σε *υποπακέτα εργασίας* ανά NULL byte (θεωρώντας ότι δεν αποτελεί μέρος του εκτελέσιμου κώδικα). Τα byte που θα τοποθετηθούν αργότερα στο περιβάλλον εκτέλεσης θα είναι τα byte του καθενός από τα υποπακέτα. Ωστόσο, υποπακέτα που έχουν μικρότερο μέγεθος από ένα κατώτατο όριο θ δε λαμβάνονται υπόψιν στα επόμενα στάδια της εξέτασης. Το όριο αυτό ορίζεται στο αρχείο ρυθμίσεων του Πράκτορα.



Σχήμα 3.10: Δυναμική ανάλυση και ανίχνευση πολυμορφικού κακόβουλου λογισμικού

Στη συνέχεια, ο Μηχανισμός Ανίχνευσης δοκιμάζει να εκτελέσει τα περιεχόμενα κάθε υποπακέτου εργασίας από όλες τις δυνατές θέσεις[2] (ξεκινώντας από το byte που ήταν πιο κοντά στην επικεφαλίδα του αρχικού πακέτου). Για άλλη μια φορά, οι θέσεις που απέχουν λιγότερο από θ byte από το τέλος της υποομάδας δεν εξετάζονται. Το κατώτατο όριο θ ουσιαστικά περιγράφει το μικρότερο δυνατό μέγεθος ενός shellcode.

[2]Η εκτέλεση από πολλαπλές θέσεις είναι μια διαδικασία που μπορεί να παραλληλιστεί πλήρως (embarrassingly parallel task).

Η εικονική εκτέλεση πραγματοποιείται στο προστατευμένο περιβάλλον που παρέχει το QEMU και μπορεί να διαρκέσει το πολύ t δευτερόλεπτα. Η τιμή t ορίζεται στο αρχείο ρυθμίσεων του Πράκτορα. Αν κατά τη διάρκεια της εκτέλεσης τα byte του πακέτου εργασίας έχουν προκαλέσει την εκτέλεση μιας «επικίνδυνης» κλήσης συστήματος, τότε η εκτέλεση τερματίζεται άμεσα και ειδοποιείται ο σχετικός Αισθητήρας. Ως «επικίνδυνες» κλήσεις συστήματος θεωρούνται αυτές που μπορούν να επιφέρουν αλλαγές στο περιβάλλον εκτέλεσης μιας διεργασίας ή που δίνουν στη διεργασία πρόσβαση σε εξωτερικούς πόρους ή πληροφορίες. Σε κάθε περίπτωση, όταν εκτελείται μια κλήση συστήματος στο εικονικό περιβάλλον του QEMU, η υλοποίηση αυτής δεν είναι πραγματική και επιστρέφονται ψευδή αποτελέσματα. Με τον τρόπο αυτό προστατεύεται το σύστημα που φιλοξενεί τον Πράκτορα από τυχόν κακόβουλες ενέργειες του υπό διερεύνηση λογισμικού. Επίσης, το υπό διερεύνηση λογισμικό εκτελείται (εικονικά) σε αυτόνομες σελίδες μνήμης με δική του στοίβα, ώστε να μην επηρεάζει την εκτέλεση του Πράκτορα.

Η εκτέλεση μιας «επικίνδυνης» κλήσης συστήματος είναι μια πολύ καλή ένδειξη για την ύπαρξη κακόβουλου λογισμικού, καθώς:

- η ίδια η εκτέλεση προϋποθέτει ότι έχει διαμορφωθεί μια πολύ συγκεκριμένη κατάσταση (στη μνήμη και στους καταχωρητές) στο περιβάλλον εκτέλεσης, και
- κρίνει το άγνωστο λογισμικό από τη συμπεριφορά του και όχι από τη μορφή του, δίνοντας έτσι τη δυνατότητα να εντοπιστεί και κακόβουλο λογισμικό που χρησιμοποιεί τεχνικές πολυμορφισμού/μεταμορφισμού.

Στην πρότυπη υλοποίηση της πλατφόρμας SEDUCE, ο παραπάνω Μηχανισμός Ανίχνευσης περιορίστηκε στον εντοπισμό κακόβουλου λογισμικού που στοχεύει σε τρωτότητες υπηρεσιών αρχιτεκτονικής Linux/x86. Βέβαια, εξαιτίας των αυξημένων δυνατοτήτων του εξομοιωτή QEMU η ανίχνευση αυτή μπορεί να επεκταθεί εύκολα και σε άλλα λειτουργικά συστήματα και αρχιτεκτονικές.

Τόσο το προστατευμένο περιβάλλον εκτέλεσης (sandbox) του QEMU όσο και ο χρονικός περιορισμός που εφαρμόζεται σε κάθε εκτέλεση ενός πακέτου εργασίας μπορούν να περιορίσουν δραστικά τους πόρους (επεξεργαστικούς, μνήμη, αποθηκευτικά μέσα κ.α.) στους οποίους θα έχει πρόσβαση το ξένο λογισμικό. Περισσότερες πληροφορίες για τα μέτρα ασφάλειας που μπορούν να εφαρμοστούν σε ελεύθερα προγραμματιζόμενες υποδομές δίνονται στο κεφάλαιο 5.

Τέλος, η δυναμική ανάλυση κίνησης για τον εντοπισμό πολυμορφικού κακόβουλου λογισμικού αποτελεί ένα εξαιρετικό παράδειγμα για το είδος των εξειδικευμένων υπηρεσιών που μπορεί να προσφέρει μια πλατφόρμα σαν τη SEDUCE, που στηρίζεται στις αρχές των Ισχυρών Ενεργών Δικτύων (εκτέλεση περιεχομένων πακέτων, προστατευμένο περιβάλλον εκτέλεσης, παροχή υπηρεσιών στο δίκτυο).

3.5 Ανίχνευση κακόβουλου λογισμικού σε πρωτόκολλα επικοινωνίας εφαρμογών

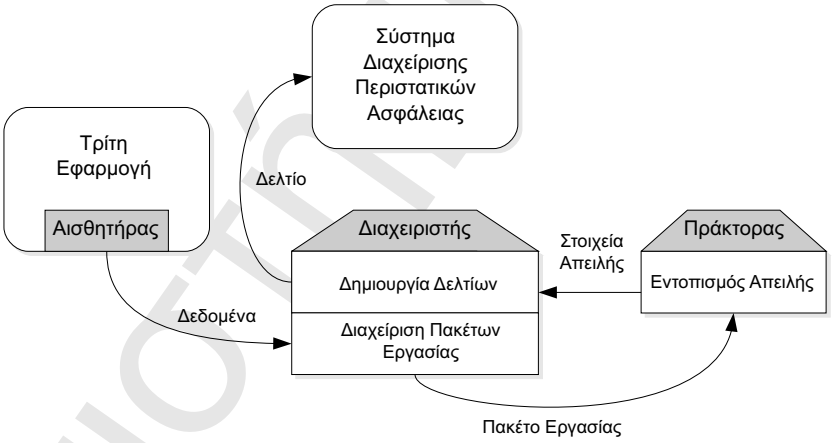
Στην αρχιτεκτονική 2 επιπέδων της πλατφόρμας SEDUCE η συλλογή κίνησης και η διαχείριση πακέτων εργασίας πραγματοποιούνται από το ίδιο λογισμικό, τον *Αισθητήρα*. Αυτό το χαρακτηριστικό της αρχιτεκτονικής μπορεί μεν να την καθιστά αποδοτική κατά την ανάλυση δεδομένων σε δίκτυα υψηλών ταχυτήτων περιορίζει όμως τον τύπο των δεδομένων που θα μπορούσε να εξετάσει η πλατφόρμα. Συγκεκριμένα, οι πράκτορες μπορούν να εντοπίσουν μόνο κακόβουλο λογισμικό που περιέχεται *ως έχει* σε δεδομένα κίνησης. Παρόλο που

η παραπάνω αρχιτεκτονική επιτρέπει την ανάπτυξη διαφορετικών μηχανισμών εντοπισμού κακόβουλου λογισμικού, δε βοηθά στον εντοπισμό λογισμικού που έχει μετασχηματισθεί και αποσταλεί στο πλαίσιο κάποιου πρωτοκόλλου επικοινωνίας ανώτερου επιπέδου στο πρότυπο OSI.

Για να μπορέσει η πλατφόρμα SEDUCE να εξετάσει δεδομένα που μεταφέρθηκαν στα πλαίσια πρωτοκόλλων επικοινωνίας υψηλότερων επιπέδων στο OSI (π.χ. πρωτόκολλα εφαρμογών) θα πρέπει:

- να ανεξαρτητοποιηθούν οι διαδικασίες συλλογής δεδομένων και διαχείρισης πακέτων εργασίας, και
- να χαρακτηριστεί το κάθε πακέτο εργασίας ως προς τον τύπο δεδομένων που περιλαμβάνει (π.χ. κίνηση πρωτοκόλλου HTTP).

Η ανεξαρτητοποίηση των διαδικασιών συλλογής δεδομένων και διαχείρισης πακέτων εργασίας επιτυγχάνεται με την υλοποίηση μιας αρχιτεκτονικής 3 επιπέδων. Στο σχήμα 3.11 παρουσιάζεται μια τέτοια αρχιτεκτονική όπου η συλλογή δεδομένων πραγματοποιείται από το λογισμικό που ονομάζεται *Αισθητήρας*, ενώ η διαχείριση και διανομή των πακέτων εργασίας πραγματοποιείται από το λογισμικό που ονομάζεται *Διαχειριστής*. Η επικοινωνία Διαχειριστή – Πράκτορα είναι όμοια με αυτή των Αισθητήρα – Πράκτορα στην αρχιτεκτονική 2 επιπέδων. Η βασική διαφορά της αρχιτεκτονικής 3 επιπέδων έγκειται στη δυνατότητα δημιουργίας Αισθητήρων διαφορετικών τύπων, οι οποίοι πλέον είναι δυνατό να ενσωματωθούν και σε τρίτες εφαρμογές προκειμένου να εξεταστούν δεδομένα που προκύπτουν μετά από κάποια επεξεργασία.



Σχήμα 3.11: Αρχιτεκτονική SEDUCE τριών επιπέδων

Οι Αισθητήρες της αρχιτεκτονικής 3 επιπέδων χαρακτηρίζουν την κίνηση που συλλέγουν ως προς τον τύπο αυτής. Ελέγχοντας τον τύπο των δεδομένων, οι Πράκτορες μπορούν να εφαρμόσουν τους μηχανισμούς ανίχνευσης που είναι πιο κατάλληλοι για αυτά τα δεδομένα. Επίσης, ανακοινώνοντας στο Διαχειριστή τους μηχανισμούς ανίχνευσης που υποστηρίζει κάθε Πράκτορας, γίνεται δυνατή η στοχευμένη προώθηση πακέτων εργασίας προς Πράκτορες που έχουν τις αντίστοιχες ικανότητες ανίχνευσης κακόβουλου λογισμικού. Για παράδειγμα, έστω ότι ένας Αισθητήρας έχει ενσωματωθεί σε ένα περιβάλλον διαχείρισης βάσεων δεδομένων (RDBMS) ή σε ένα σχετικό ενδιάμεσο εξυπηρετητή (RDBMS proxy) ώστε να συλλέγει δεδομένα που αφορούν ερωτήματα SQL. Τα δεδομένα αυτά θα μπορούσαν να

γίνουν αντικείμενο εξέτασης από Πράκτορες, τόσο για τον εντοπισμό shellcode όσο και για τον εντοπισμό άλλων κακόβουλων εντολών, π.χ. τύπου SQL injection [109].

Όπως στην αρχιτεκτονική 2 επιπέδων, έτσι και εδώ, μόλις εντοπιστεί κάποιο κακόβουλο λογισμικό από ένα Πράκτορα, προωθείται σχετική ειδοποίηση προς το Σύστημα Διαχείρισης Περιστατικών Ασφάλειας. Η ειδοποίηση αυτή συντάσσεται πλέον από το αρμόδιο υποσύστημα του Διαχειριστή.

Η πειραματική υλοποίηση της πλατφόρμας SEDUCE περιλαμβάνει και μια υλοποίηση της παραπάνω αρχιτεκτονικής 3 επιπέδων. Προκειμένου να απλουστευτεί η διαδικασία ανάπτυξης Αισθητήρων για αυτή την αρχιτεκτονική, δημιουργήθηκε μια βιβλιοθήκη σε γλώσσα C, η οποία αναλαμβάνει την επικοινωνία του Αισθητήρα με το Διαχειριστή. Η προγραμματιστική διεπαφή (API) της βιβλιοθήκης παρουσιάζεται στο σχήμα 3.12:

```
struct tuple4
{
        u_short source;
        u_short dest;
        u_int saddr;
        u_int daddr;
};

enum data_t { UNKNOWN, DGRAM_DATA, STREAM_DATA, HTTP, SMTP, DNS, ... };

int manager_connect(int *sockfd, in_addr_t addr, unsigned short port);
int manager_disconnect(int sockfd);

int new_stream_connection(int sockfd, const struct tuple4 *conn, unsigned **id,
                          enum data_t type);
int close_stream_connection(int sockfd, unsigned *id);
int send_stream_data(int sockfd, unsigned id, const void *data, size_t len);
int break_stream_data(int sockfd, unsigned id);

int send_dgram_data(int sockfd, const struct tuple4 *conn, const void *data,
                    size_t len, enum data_t type);
```

Σχήμα 3.12: Προγραμματιστική διεπαφή (API) Αισθητήρα / Διαχειριστή

Η διεπαφή απαρτίζεται από 7 βασικές συναρτήσεις:

manager_connect – Υπεύθυνη για τη σύνδεση του Αισθητήρα με το Διαχειριστή που βρίσκεται στη δικτυακή διεύθυνση *addr* και τη θύρα *port*. Όλη η επικοινωνία με το Διαχειριστή μετά το πέρας αυτής της συνάρτησης θα γίνεται μέσω του περιγραφέα *sockfd*.

manager_disconnect – Τερματίζει τη συνεδρία Αισθητήρα / Διακομιστή.

new_stream_connection – Ειδοποιεί το Διαχειριστή ότι οι δύο οντότητες που περιγράφονται από τη δομή *conn* ξεκινούν μια σύνδεση τύπου ροής δεδομένων (stream connection) και τα δεδομένα που θα προκύψουν από αυτή τη συνεδρία θα είναι τύπου *type*. Κάθε μελλοντική αναφορά σε αυτή τη συνεδρία θα γίνεται μέσω του ακεραίου *id*.

close_stream_connection – Ειδοποιεί το Διαχειριστή ότι η συνεδρία που συνδέεται με τον ακέραιο *id* ολοκληρώθηκε.

send_stream_data – Αποστέλλει στο Διαχειριστή τα δεδομένα *data*, μήκους *len* byte και τύπου ροής (stream data) της συνεδρίας που συνδέεται με τον ακέραιο *id*.

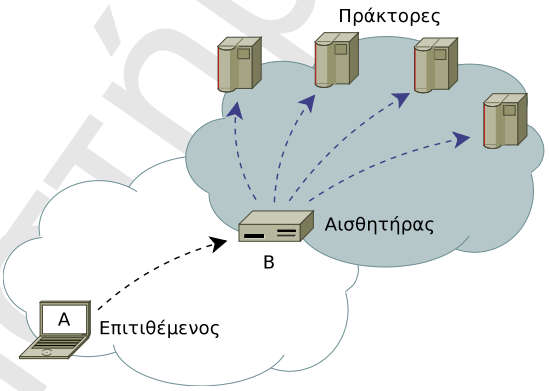
break_stream_data – Ειδοποιεί το Διαχειριστή ότι τα επόμενα δεδομένα της ροής δεδομένων που σχετίζεται με τον ακέραιο *id* θα ανήκουν σε νέα ομάδα δεδομένων (βλ. ενότητα 3.3).

send_dgram_data – Αποστέλλει στο Διαχειριστή τα δεδομένα *data*, μήκους *len* και τύπου *type* που αντάλλαξαν οι οντότητες που περιγράφονται από τη δομή *conn*. Τα δεδομένα αυτά ανήκουν σε πρωτόκολλο που στηρίζεται σε μηνύματα (datagrams).

Θα πρέπει να σημειωθεί ότι ο διαχωρισμός μεταξύ πρωτοκόλλων τύπου ροής δεδομένων (stream-based) και μηνυμάτων (datagram-based) επεκτείνεται πέραν από τον απλό διαχωρισμό μεταξύ κίνησης TCP και UDP. Κάθε πακέτο εργασίας που ανήκει σε πρωτόκολλο τύπου μηνυμάτων, μπορεί να εξεταστεί αυτόνομα, δίχως να απαιτείται πρόσβαση σε άλλα πακέτα εργασίας της ίδιας συνόδου. Κάτι τέτοιο όμως δεν ισχύει για πακέτα εργασίας που συνδέονται με πρωτόκολλα τύπου ροής δεδομένων. Εκεί, τα πακέτα που ανήκουν στην ίδια ομάδα δεδομένων, συχνά[3] τοποθετούνται σε προσωρινή μνήμη (buffering) προτού ξεκινήσει η διαδικασία εντοπισμού κακόβουλου λογισμικού σε αυτά.

3.6 Πειραματική αξιολόγηση

Στα πλαίσια της πειραματικής αξιολόγησης της πλατφόρμας SEDUCE, έγινε εγκατάσταση της πρότυπης υλοποίησης δύο επιπέδων σε δίκτυο ταχύτητας 1 gigabit/δευτερόλεπτο. Η τοπολογία που χρησιμοποιήθηκε για τα πειράματα παρουσιάζεται στο σχήμα 3.13 και περιλαμβάνει τον υπολογιστή A του επιτιθέμενου, τον Αισθητήρα B και τέσσερις Πράκτορες που εγκαταστάθηκαν σε ξεχωριστούς κόμβους. Τόσο ο υπολογιστής Α όσο και οι Πράκτορες συνδέονται με τον Αισθητήρα B μέσω gigabit σύνδεσης. Για λόγους απλότητος το ρόλο του θύματος στις επιθέσεις τον έπαιξε ο Αισθητήρας B.



Σχήμα 3.13: Τοπολογία πειραμάτων

Αρχικά έγιναν μετρήσεις ώστε να διαπιστωθεί ο μέγιστος ρυθμός πακέτων που μπορούσαν να φτάσουν σε μια εφαρμογή του Αισθητήρα B από τον υπολογιστή A. Ο εντοπισμός αυτού του ρυθμού είναι σημαντικός καθώς αποτελεί *άνω φράγμα* για το ρυθμό με τον οποίο θα μπορεί να συλλέξει πακέτα το λογισμικό του Αισθητήρα. Ο ρυθμός αυτός είναι μικρότερος από αυτόν που ορίζει η ταχύτητα του δικτυακού μέσου, καθώς υπάρχει ανώτατο όριο στα πακέτα που μπορούν να επεξεργαστούν τα δικτυακά στοιχεία που βρίσκονται μεταξύ των κόμβων A και B (δρομολογητές και κατανεμητές). Επιπρόσθετα, ο χρόνος που

[3] Ανάλογα με το μηχανισμό ανίχνευσης κακόβουλου λογισμικού.

δαπανάται στην επεξεργασία κάθε πακέτου από τον πυρήνα των λειτουργικών συστημάτων των κόμβων A και B περιορίζει τόσο το ρυθμό αποστολής των πακέτων, όσο και το ρυθμό λήψης αυτών από τις εφαρμογές.

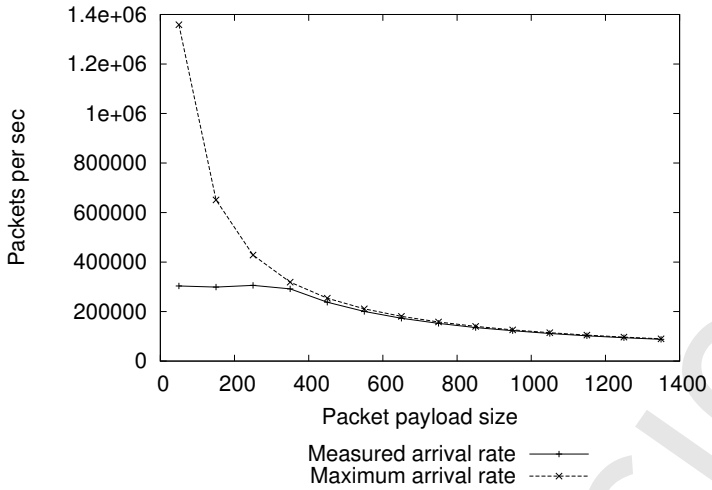
Χρησιμοποιώντας την εφαρμογή netperf [110] έγιναν μετρήσεις σχετικά με τη μαζική αποστολή (δίχως καθυστέρηση) πακέτων UDP από τον κόμβο A προς τον κόμβο B. Κάθε πείραμα περιλάμβανε διαφορετικό μέγεθος πακέτων ώστε να εξεταστούν διάφοροι ρυθμοί αποστολής πακέτων. Τόσο ο κόμβος A όσο και ο κόμβος B ήταν εξοπλισμένοι με gigabit ελεγκτές δικτύου (Intel 82566DM, Tigeon3), μνήμη RAM 1GB, επεξεργαστή Intel Pentium 4 (3.40Ghz, 2048kb cache) και χρησιμοποιούσαν το λειτουργικό σύστημα Linux. Επίσης, οι οδηγοί των ελεγκτών δικτύου είχαν ρυθμιστεί κατάλληλα ώστε να μπορούν να τοποθετήσουν σε ουρά 30000 πακέτα προτού αυτά γίνουν αντικείμενο επεξεργασίας από τον πυρήνα (ρύθμιση `netdev_max_backlog`). Τα αποτελέσματα των σχετικών πειραμάτων παρουσιάζονται στον πίνακα 3.1.

Μέγεθος Περιεχομένου (byte)	Αριθμός Απεσταλμένων Πακέτων	Αριθμός Ληφθέντων Πακέτων	Ρυθμός Αποστολής (mbit/sec)	Ρυθμός Λήψης (mbit/sec)	Ρυθμός Λήψης (πακέτα/sec)	Μεγ. Ρυθμός Λήψης για το Μέσο (πακέτα/sec)
50	3071657	3035477	122,87	121,42	303550,00	1358695,65
150	3002546	2989348	360,31	358,72	298933,33	651041,67
250	3077191	3058556	615,44	611,71	305855,00	428082,19
350	2964003	2918285	829,92	817,12	291828,57	318877,55
450	2388642	2379795	859,91	856,72	237977,78	254065,04
550	2020353	2006391	888,95	882,81	200638,64	211148,65
650	1731612	1731588	900,44	900,42	173157,69	180635,84
750	1524183	1521469	914,51	912,88	152146,67	157828,28
850	1356008	1354641	922,08	921,15	135463,24	140134,53
950	1228140	1226383	933,38	932,05	122638,16	126008,07
1050	1114307	1113096	936,02	935,00	111309,52	114468,86
1150	1026274	1022274	944,17	940,49	102227,17	104865,77
1250	945720	945720	945,72	945,72	94572,00	96749,23
1350	881695	880221	952,23	950,64	88022,22	89798,85

Πίνακας 3.1: Αποτελέσματα μετρήσεων με την εφαρμογή netperf

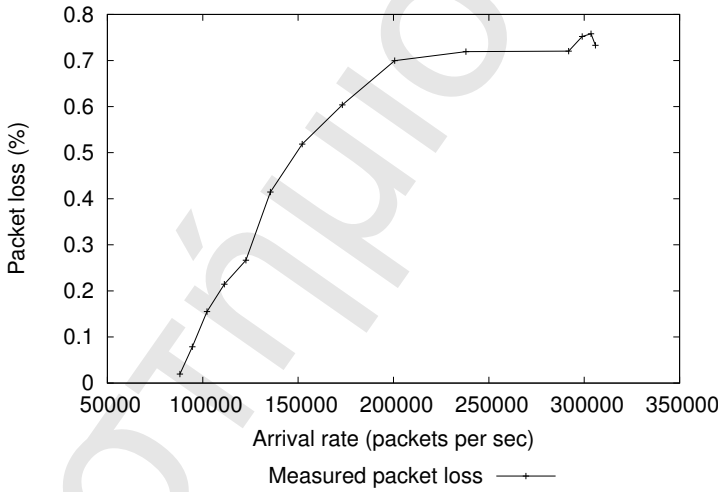
Από τα αποτελέσματα προκύπτει ότι ο μέγιστος ρυθμός λήψης πακέτων για τις εφαρμογές του κόμβου B κυμαίνεται περίπου στις τριακόσιες πέντε χιλιάδες πακέτα το δευτερόλεπτο, ενώ οι απώλειες πακέτων (βάσει των απεσταλμένων και ληφθέντων) είναι σχεδόν σε όλες τις περιπτώσεις μη μηδενικές, με μέγιστο ποσοστό απώλειας το 1%. Στο σχήμα 3.14 αποτυπώνεται η διαφορά μεταξύ του ρυθμού λήψης πακέτων που παρατηρήθηκε στα πειράματα (measured arrival rate) και του μέγιστου ρυθμού λήψης πακέτων, όπως αυτός διαμορφώνεται από την ταχύτητα μετάδοσης δεδομένων του δικτυακού μέσου (maximum arrival rate).

Στο επόμενο πείραμα ενεργοποιήθηκε το λογισμικό του Αισθητήρα και υπό τις ίδιες συνθήκες με τα προηγούμενα πειράματα (μαζική αποστολή UDP πακέτων δίχως καθυστέρηση) στάλθηκαν 10.000 πακέτα διαφόρων μεγεθών. Αυτή τη φορά στόχος ήταν η καταμέτρηση της απώλειας πακέτων που θα συνέβαινε εξαιτίας της επεξεργασίας των πακέτων από το λογισμικό του Αισθητήρα. Θα πρέπει να σημειωθεί ότι ο Αισθητήρας είχε ρυθμιστεί κατάλληλα ώστε να μπορεί να διατηρήσει το σύνολο των παραπάνω πακέτων στη μνήμη του. Στο σχήμα 3.15 παρουσιάζονται τα αποτελέσματα αυτού του πειράματος. Από το γράφημα φαίνεται καθαρά ότι όταν υπάρχει υψηλός ρυθμός άφιξης πακέτων, οι διαδικασίες της εφαρμογής του Αισθητήρα δημιουργούν τέτοιες καθυστερήσεις που η ποσοστιαία απώλεια πακέτων μπορεί να ξεπεράσει το 70%. Μάλιστα, όταν ο ρυθμός άφιξης πακέτων



Σχήμα 3.14: Σύγκριση μέγιστου και καταγεγραμμένου ρυθμού λήψης πακέτων για τις εφαρμογές του κόμβου B

πλησιάσει κοντά στο όριο που θέτει ο πυρήνας, τότε η απώλεια πακέτων αυξάνεται ακόμη περισσότερο και μπορεί να ξεπεράσει το 75%.



Σχήμα 3.15: Ρυθμός λήψης πακέτων και ποσοστιαία απώλεια πακέτων στον Αισθητήρα B

Στην επόμενη φάση της πειραματικής αξιολόγησης εξετάστηκαν οι Μηχανισμοί Ανίχνευσης Κακόβουλου Λογισμικού των Πρακτόρων. Οι τέσσερις Πράκτορες των επόμενων πειραμάτων βασίζονται σε όμοιο υλικό και λειτουργικό σύστημα με αυτά των κόμβων A και B. Κάθε Πράκτορας εκτελείται σε περιβάλλον με έναν επεξεργαστή και διαθέτει μία διεργασία – Εργάτη. Κατά την επικοινωνία με τον Αισθητήρα B, αν δεν υπάρχει κάποιο πακέτο εργασίας διαθέσιμο, περιμένει ένα δευτερόλεπτο προτού προχωρήσει σε νέα αίτηση για πακέτο εργασίας.

Οι Μηχανισμοί Ανίχνευσης που εξετάστηκαν ήταν οι εξής:

`qemu` – Δυναμική ανάλυση μέσω εικονικής εκτέλεσης σε περιβάλλον QEMU.

`libemu` – Υβριδική ανάλυση (στατική και δυναμική) μέσω της βιβλιοθήκης libemu.

pyopty2 – Στατική ανάλυση και εντοπισμός NOP sled με πολλαπλά byte μέσω σεναρίου σε γλώσσα Python.

fnord – Στατική ανάλυση και εντοπισμός απλών NOP sled από αυτόνομα byte.

Η εξέταση του Μηχανισμού qemu έγινε στα πλαίσια της διερεύνησης λύσεων εικονικής εκτέλεσης ως προς την καταλληλότητά τους για τον εντοπισμό πολυμορφικού κακόβουλου λογισμικού σε δικτυακή κίνηση. Αντίστοιχα, η βιβλιοθήκη libemu αποτελεί μέχρι στιγμής την πιο διαδεδομένη λύση για τον εντοπισμό shellcode από προγράμματα ανίχνευσης επιθέσεων ανοιχτού λογισμικού ([111, 112]). Μέσω αυτής γίνεται δυνατή η σύγκριση των δυνατοτήτων της πλατφόρμας SEDUCE με αυτές άλλων αντίστοιχων συστημάτων. Τέλος, η εξέταση των Μηχανισμών pyopty2 και fnord μας επιτρέπει να λάβουμε μια εικόνα για την απόδοση Μηχανισμών στατικής ανάλυσης κατά τον εντοπισμό πολυμορφικού shellcode αλλά και για την απόδοση Μηχανισμών που υλοποιούνται σε γλώσσες προγραμματισμού υψηλότερου επιπέδου.

Το πρώτο πείραμα με τους παραπάνω Μηχανισμούς Ανίχνευσης Κακόβουλου Λογισμικού εξέτασε την απόδοσή τους κατά την εξέταση πακέτων εργασίας διαφόρων μεγεθών με τυχαία δεδομένα. Τα αποτελέσματα του πειράματος παρουσιάζονται στον πίνακα 3.2. Θα πρέπει να σημειωθεί ότι κανένας Μηχανισμός δεν εντόπισε εσφαλμένα κάποιο shellcode στα πακέτα του πειράματος αυτού (false positive).

Μέγεθος Payload (byte)	**Χρόνος Επεξεργασίας (usec)**			
	qemu	libemu	pyopty2	fnord
50	5.379	40	4.771	5
150	125.858	46	14.073	7
250	28.060	58	23.403	15
350	112.040	63	27.049	13
450	128.104	81	42.729	46
550	191.489	87	45.503	40
650	669.322	89	52.672	33
750	429.489	104	67.143	72
850	329.058	117	74.895	95
950	330.407	125	84.231	130
1050	550.199	121	75.417	74
1150	361.498	8.878	97.814	186
1250	107.798	151	106.338	219
1350	615.420	147	102.284	123

Πίνακας 3.2: Χρόνος επεξεργασίας πακέτων διαφόρων μεγεθών με τυχαία δεδομένα

Από τα αποτελέσματα του πειράματος προκύπτει ότι ο χρόνος εξέτασης ενός πακέτου εξαρτάται τόσο από το μέγεθος του πακέτου όσο και από τα περιεχόμενα αυτού. Επίσης, η βιβλιοθήκη libemu κρίνεται εξαιρετικά αποδοτική κατά την επεξεργασία μεγάλων πακέτων καθώς παρουσιάζει χρόνους εκτέλεσης συγκρίσιμους με αυτούς μιας στατικής μεθόδου ανάλυσης (fnord). Αντίθετα, ο Μηχανισμός qemu, παρουσιάζει μια ιδιαίτερη καθυστέρηση κατά την επεξεργασία, που οφείλεται κυρίως στην ενδελεχή εξέταση του πακέτου από κάθε δυνατή θέση.

Στο επόμενο πείραμα έγινε σύγκριση των στατικών, δυναμικών αλλά και υβριδικών μεθόδων κατά την ανίχνευση shellcode που περιέχει NOP sled. Με βάση το "linux/x86/exec" payload του έργου metasploit [106] δημιουργήθηκαν 3 shellcode:

OPTY-10-EXEC – Shellcode στο οποίο προηγούνται 10 byte από NOP sled με ακολουθίες πολλαπλών byte.

RND-OPTY-10-EXEC – Shellcode όμοιο με το παραπάνω με τη διαφορά ότι προηγείται και έπεται αυτού μια ακολουθία 100 τυχαίων byte.

SINGLENOP-60-EXEC – Shellcode με NOP sled 60 αυτόνομων byte.

Τα αποτελέσματα του πειράματος με τα NOP sled παρουσιάζονται στον πίνακα 3.3.

	qemu		**libemu**		**pyopty2**		**fnord**	
Payload	Ανίχν.	Χρ. (usec)	Ανίχν.	Χρ. (usec)	Ανίχν.	Χρ. (usec)	Ανίχν.	Χρ. (usec)
OPTY-10-EXEC	✓	415	✓	28.128	✓	2.336		6
RND-OPTY-10-EXEC	✓	147.500	✓	36.712	✓	2.429		10
SINGLENOP-60-EXEC	✓	539	✓	22.964		6.880	✓	17

Πίνακας 3.3: Ανίχνευση shellcode με NOP sled

Παρατηρεί κανείς ότι στις περιπτώσεις όπου τα πρώτα byte του πακέτου είναι μέρος του shellcode, η εξέταση με το Μηχανισμό qemu ολοκληρώνεται άμεσα. Στην περίπτωση όμως όπου θα πρέπει να εντοπίσει το σημείο από όπου ξεκινάει το shellcode (RND-OPTY-10-EXEC), ο Μηχανισμός αυτός παρουσιάζει μια σημαντική καθυστέρηση. Τόσο ο Μηχανισμός qemu όσο και ο Μηχανισμός libemu πάντως, εντόπισαν το κακόβουλο payload σε όλα τα shellcode του πειράματος. Αντίθετα, οι μέθοδοι στατικής ανάλυσης εντόπισαν το κακόβουλο λογισμικό μόνο στις περιπτώσεις όπου αναγνώρισαν κάποια από τα byte του NOP sled.

Το επόμενο πείραμα εξέτασε τις δυνατότητες των μεθόδων δυναμικής και υβριδικής ανάλυσης κατά τον εντοπισμό shellcode με διαφορετικά κάθε φορά χαρακτηριστικά. Για το πείραμα αυτό, χρησιμοποιήθηκαν όλα τα payloads του έργου metasploit [106] που αφορούν συστήματα Linux αρχιτεκτονικής intel x86. Θα πρέπει να σημειωθεί ότι για τους σκοπούς αυτού του πειράματος ο κώδικας της βιβλιοθήκης libemu διαμορφώθηκε κατάλληλα ώστε να μπορεί να εντοπίσει shellcode σε δεδομένα λίγων byte. Συγκεκριμένα, αν ανιχνευόταν GetPC σε ροή κώδικα με περισσότερες από 10 εντολές επεξεργαστή τότε η βιβλιοθήκη θα χαρακτήριζε τα υπό διερεύνηση byte ως κακόβουλα[4].

Τα αποτελέσματα του πειράματος παρουσιάζονται στον πίνακα 3.4.

	qemu		**libemu**	
Payload	Ανίχνευση	Χρόνος Επεξ. (usec)	Ανίχνευση	Χρόνος Επεξ. (usec)
ADDUSER	✓	346	✓	44.705
CHMOD	✓	367	✓	5.946
EXEC	✓	370	✓	16.759
METERPRETER-BIND-IPv6-TCP	✓	364		26
METERPRETER-BIND-TCP	✓	362		25
METERPRETER-FIND-TAG	✓	381		24
METERPRETER-REVERSE-IPv6-TCP	✓	359		25
METERPRETER-REVERSE-TCP	✓	360		26
SHELL-BIND-IPv6-TCP	✓	360		26
SHELL-BIND-TCP	✓	368		26
SHELL-FIND-PORT	✓	373		24
SHELL-FIND-TAG	✓	377		24
SHELL-REVERSE-IPv6-TCP	✓	353		25
SHELL-REVERSE-TCP	✓	368		41
SHELL-REVERSE-TCP2	✓	357		25

Πίνακας 3.4: Ανίχνευση διαφορετικών payload

[4]Το όριο αυτό είχε τεθεί αρχικά στα 100 byte από τους συγγραφείς της βιβλιοθήκης.

Παρατηρεί κανείς ότι ο Μηχανισμός qemu εντόπισε σε όλες τις περιπτώσεις το shellcode και μάλιστα πιο γρήγορα από το Μηχανισμό libemu. Οι «επικίνδυνες» κλήσεις συστήματος που βρέθηκαν στα παραπάνω shellcode είχαν σχέση με δικτυακές συνδέσεις (socketcall), εκτέλεση εφαρμογών (execve), αλλαγή στοιχείων στο σύστημα αρχείων (chmod, write), αλλαγή στοιχείων ιδιοκτήτη διεργασίας (setreuid), και πρόσβαση σε δεδομένα του συστήματος αρχείων (open). Ο Μηχανισμός libemu αντίθετα, εντόπισε το shellcode μόνο σε 3 περιπτώσεις, στις οποίες εμφάνισε και αυξημένη καθυστέρηση.

		qemu		**libemu**	
Payload	**Μέγεθος (byte)**	Ανίχν.	Χρόνος Επεξ. (usec)	Ανίχν.	Χρόνος Επεξ. (usec)
ALPHA-MIXED	148	✓	6.010		47
ALPHA-UPPER	155	✓	6.547		46
CALL4-DWORD-XOR	68	✓	1.101	✓	13.103
COUNTDOWN	59	✓	2.914	✓	10.340
FNSTENV-MOV	66	✓	1.106	✓	15.806
JMP-CALL-ADDITIVE	73	✓	1.220	✓	28.189
NON-ALPHA	100	✓	2.624	✓	9.120
NON-UPPER	79	✓	426		3.257
SHIKATA-GA-NAI	70	✓	1.196		6.020
SHIKATA-GA-NAI-10	313	✓	29.217		9.304
SHIKATA-GA-NAI-10-RND	913	✓	559.879		10.999

Πίνακας 3.5: Ανίχνευση κωδικοποιημένου payload

Για να εξεταστεί περαιτέρω η δυνατότητα εντοπισμού πολυμορφικού λογισμικού, κωδικοποιήθηκε το 'linux/x86/exec' payload με όλες τις μεθόδους αυτόματης κωδικοποίησης που παρέχει το έργο metasploit. Στα παραγόμενα shellcode προστέθηκαν δύο ακόμη, ένα κρυπτογραφημένο 10 φορές με τον αλγόριθμο "Shikata Ga Nai" και ένα όμοιο που πλαισιώνεται από 300 byte επικεφαλίδας με τυχαία byte και 300 byte επιλόγου με τυχαία byte. Η επιλογή του Shikata Ga Nai για τα τελευταία 2 shellcode δεν είναι τυχαία. Ο αλγόριθμος κωδικοποίησης αυτός τοποθετεί στο shellcode έναν αποκωδικοποιητή που είναι μεταμορφικός. Έτσι, το παραγόμενο shellcode γίνεται δύσκολο να εντοπιστεί μέσω κάποιας τεχνολογίας που στηρίζεται σε υπογραφές (signature matching).

Τα αποτελέσματα του πειράματος με την κωδικοποίηση των payload παρουσιάζονται στον πίνακα 3.5. Παρόλη την κωδικοποίηση, ο Μηχανισμός qemu κατάφερε να εντοπίσει για άλλη μια φορά όλα τα payload. Στις 5 περιπτώσεις όπου η libemu εντόπισε και εκείνη το payload, ο Μηχανισμός qemu ολοκλήρωσε την εξέταση σε μικρότερο χρονικό διάστημα από εκείνη. Επίσης, μέσω δυναμικής ανάλυσης εντοπίστηκαν και shellcode σε πακέτα που περιείχαν 10 κωδικοποιήσεις αλλά και τυχαία δεδομένα. Τα αποτελέσματα αυτά είναι ιδιαίτερα ενθαρρυντικά καθώς φανερώνουν ότι η δυναμική ανάλυση μπορεί να βοηθήσει ουσιαστικά στον αυτόματο εντοπισμό κακόβουλου πολυμορφικού λογισμικού.

Μια πρόσφατη μέθοδος κωδικοποίησης που χρησιμοποιείται κυρίως σε εστιασμένες επιθέσεις είναι η λεγόμενη "Context Keyed Payload Encoding" [113]. Στην κωδικοποίηση αυτή ο επιτιθέμενος χρησιμοποιεί ως κλειδί αποκρυπτογράφησης κάποιο στοιχείο από το περιβάλλον της υπηρεσίας που θα αποτελέσει θύμα της επίθεσης. Επειδή τα συστήματα ανίχνευσης επιθέσεων δεν έχουν πρόσβαση στο περιβάλλον αυτό, δε μπορούν να αποκωδικοποιήσουν σωστά το shellcode και δε μπορούν κατ' επέκταση να εντοπίσουν το όποιο payload περιέχεται μέσα σε αυτό. Η εφαρμογή metasploit παρέχει μια σειρά από κωδικοποιητές αυτού του είδους, τρεις από τους οποίους εξετάστηκαν με την πλατφόρμα SEDUCE.

Ο πρώτος κωδικοποιητής (CONTEXT-TIME) κωδικοποιεί το shellcode κατά τέτοιο τρόπο ώστε η αποκωδικοποίηση να γίνεται μόνο εντός ενός συγκεκριμένου χρονικού παραθύρου.

Όπως φαίνεται στον πίνακα 3.6, η δυναμική μέθοδος ανάλυσης κατάφερε να εντοπίσει τον αποκωδικοποιητή σε αυτή την περίπτωση, επειδή εκτέλεσε την κλήση συστήματος *time* για να λάβει την τρέχουσα ώρα. Με παρόμοιο τρόπο εντοπίστηκε και ο κωδικοποιητής CONTEXT-STAT. Ο κωδικοποιητής αυτός λαμβάνει μέσω του συστήματος αρχείων δύο τιμές (ημερομηνία τροποποίησης αρχείου, μέγεθος αρχείου) τις οποίες χρησιμοποιεί ως κλειδί αποκρυπτογράφησης. Ο Μηχανισμός qemu εντόπισε σε αυτή την περίπτωση την κλήση συστήματος *stat64* και ενημέρωσε για την πιθανή ύπαρξη ενός shellcode. Αντίστοιχα η βιβλιοθήκη libemu εντόπισε το μηχανισμό GetPC που χρησιμοποιεί ο συγκεκριμένος αποκωδικοποιητής και ενημέρωσε και εκείνη με τη σειρά της για την πιθανή ύπαρξη κακόβουλου λογισμικού. Στην τελευταία περίπτωση όπου χρησιμοποιείται ο κωδικοποιητής CONTEXT-CPUID, το payload κωδικοποιείται με μια τιμή που προκύπτει από την έξοδο της εντολής επεξεργαστή *cpuid*. Σε αυτή την περίπτωση το payload ανιχνεύθηκε μόνο από τον Μηχανισμό libemu, καθώς ο qemu δεν περιλαμβάνει ελέγχους σχετικούς με συγκεκριμένες εντολές επεξεργαστή.

		qemu		**libemu**	
Payload	**Μέγεθος (byte)**	Ανίχν.	Χρόνος Επεξ. (usec)	Ανίχν.	Χρόνος Επεξ. (usec)
CONTEXT-TIME	75	✓	342		28.195
CONTEXT-STAT	105	✓	439	✓	59.452
CONTEXT-CPUID	97		264.548	✓	60.567

Πίνακας 3.6: Ανίχνευση payload κωδικοποιημένου με τη μέθοδο Context Keyed Payload Encoding

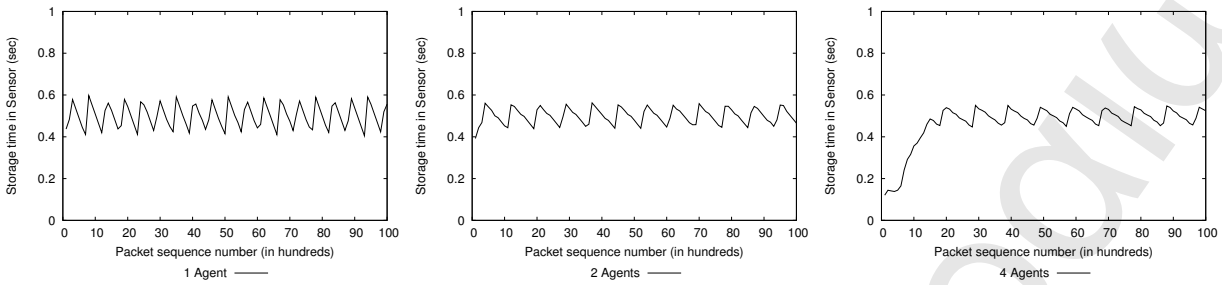
Η τελευταία ομάδα πειραμάτων βοήθησε στη διερεύνηση της επίδοσης της πλατφόρμας SEDUCE όταν αυτή καλείται να εξετάσει κίνηση διαφόρων ταχυτήτων. Στα πλαίσια αυτών των πειραμάτων δημιουργήθηκε σταθερή δικτυακή κίνηση από τον κόμβο A προς τον κόμβο B με πακέτα μεγέθους 200 byte[5]. Ως μονάδα μέτρησης της απόδοσης του συστήματος θεωρήθηκε ο χρόνος που παραμένει ένα πακέτο στην προσωρινή μνήμη του Αισθητήρα. Συγκεκριμένα, ο Αισθητήρας κατέγραφε το μέσο όρο παραμονής ανά 100 πακέτα που διαχειριζόταν.

Στο πρώτο πείραμα αυτής της ομάδας απεστάλη κίνηση σταθερής ταχύτητας 100 πακέτων το δευτερόλεπτο (100pps) προς τον κόμβο B και κατεγράφη ο μέσος χρόνος παραμονής όταν χρησιμοποιούνται ένας, δύο ή τέσσερις Πράκτορες. Στο σχήμα 3.16 παρουσιάζονται τα αποτελέσματα αυτών των μετρήσεων. Είναι εμφανές ότι η κίνηση αυτή δημιουργεί μικρό φορτίο για το σύστημα και έτσι ο χρόνος παραμονής είναι περίπου ο ίδιος και για τις τρεις περιπτώσεις. Οι τιμές μεταξύ των οποίων κυμαίνεται ο χρόνος παραμονής έχουν άμεση σχέση τόσο με την ταχύτητα λήψης πακέτων όσο και με το χρόνο που περιμένουν οι Πράκτορες όταν δεν υπάρχει πακέτο εργασίας για αυτούς.

Στο δεύτερο πείραμα αυτής της ομάδας, εξετάστηκε η απόδοση της πλατφόρμας όταν έφτανε σε αυτή κίνηση χιλίων πακέτων το δευτερόλεπτο (1000pps). Σε αυτή την περίπτωση, αν η πλατφόρμα έχει μόνο ένα Πράκτορα, τότε ο χρόνος παραμονής στον Αισθητήρα αυξάνεται γραμμικά ως προς το χρόνο που περνά. Αν η κίνηση δεν ελαττωθεί, η προσωρινή μνήμη του Αισθητήρα θα γεμίσει και έτσι το νήμα το οποίο είναι υπεύθυνο για τη διαχείριση της μνήμης θα αναγκαστεί να αφαιρέσει κάποιες από τις πιο παλιές ομάδες δεδομένων στη μνήμη. Ευτυχώς κάτι τέτοιο δεν είναι απαραίτητο όταν υπάρχουν δύο ή τέσσερις Πράκτορες. Όπως φαίνεται στο σχήμα 3.17 οι Πράκτορες αυτοί θα καταφέρουν να μοιραστούν το φορτίο μεταξύ τους και δε θα υπάρξει κάποια αύξηση στο χρόνο παραμονής.

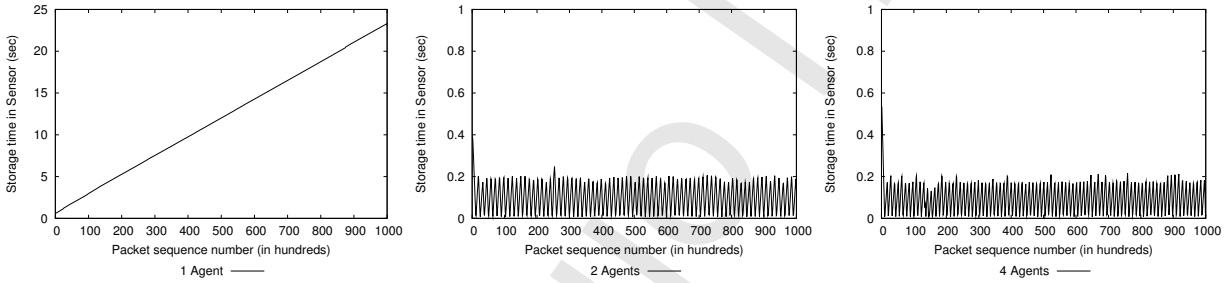
Στο τρίτο και τελευταίο πείραμα αυτής της ομάδας εξετάστηκε η απόδοση του συστήματος όταν σε αυτό φτάνει σταθερή κίνηση δέκα χιλιάδων πακέτων το δευτερόλεπτο

[5]Τα πακέτα αυτά μοιράζονταν το ίδιο περιεχόμενο από τυχαία byte.



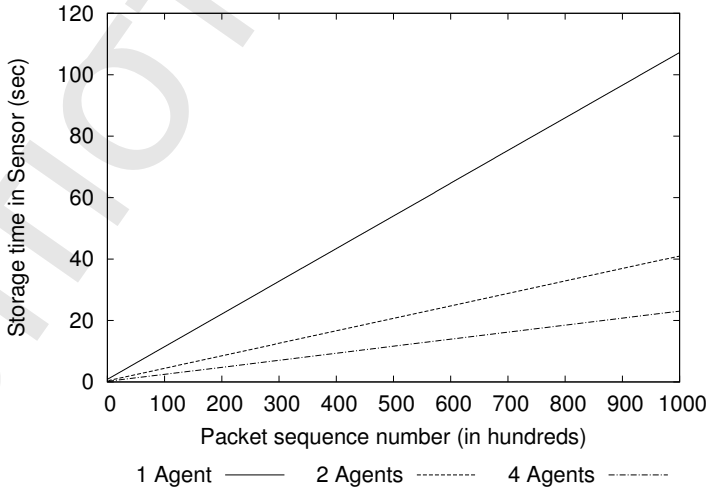
(α΄) (β΄) (γ΄)

Σχήμα 3.16: Μέσος χρόνος παραμονής στον Αισθητήρα για πακέτα κίνησης 100pps



(α΄) (β΄) (γ΄)

Σχήμα 3.17: Μέσος χρόνος παραμονής στον Αισθητήρα για πακέτα κίνησης 1000pps



Σχήμα 3.18: Μέσος χρόνος παραμονής στον Αισθητήρα για πακέτα κίνησης 10000pps

(10000pps). Τα αποτελέσματα του πειράματος, που παρουσιάζονται στο σχήμα 3.18, έδειξαν ότι υπήρξε γραμμική αύξηση του χρόνου παραμονής σε όλες τις περιπτώσεις με μικρότερη αυτή που σημειώθηκε στην τοπολογία με τους τέσσερις Πράκτορες. Πάντως, ως επί το πλείστον οι εξυπηρετητές δε λαμβάνουν αιτήσεις με σταθερή ταχύτητα και έτσι οι Πράκτορες μπορούν να εκμεταλλευτούν μια δικτυακή παύση (ή χαμηλή κίνηση) ώστε να επεξεργαστούν πακέτα που έφτασαν στον Αισθητήρα όταν υπήρχε αυξημένη κίνηση.

3.7 Ποιοτική αξιολόγηση

Η πλατφόρμα SEDUCE αποτελεί ένα κατανεμημένο σύστημα για τον εντοπισμό κακόβουλου πολυμορφικού λογισμικού σε δικτυακή κίνηση. Χαρακτηριστικό της πλατφόρμας είναι η ευκολία με την οποία αυτή μπορεί να επεκταθεί προκειμένου να καλύψει τις ανάγκες δικτύων μεγάλων οργανισμών αλλά και η δυνατότητα αυτής να εξετάσει τη δικτυακή κίνηση εις βάθος, μέσω διαφόρων μεθόδων ανάλυσης. Η υλοποίηση αυτής εκμεταλλεύεται τόσο τις επεξεργαστικές δυνατότητες των συστημάτων που τη φιλοξενούν (συστήματα με πολλαπλούς πυρήνες και επεξεργαστές) όσο και τις επεξεργαστικές δυνατότητες τρίτων συστημάτων όπως είναι οι συστοιχίες υπολογιστών.

Οι Πράκτορες της πλατφόρμας λειτουργούν ως Ενεργοί Κόμβοι ενός Ισχυρού Ενεργού Δικτύου και αναλαμβάνουν την εκτέλεση των περιεχομένων των πακέτων που καλείται να εξετάσει η πλατφόρμα. Εφόσον πρόκειται για την εκτέλεση μη έμπιστου κώδικα, αυτή περιορίζεται σε ένα εικονικό περιβάλλον εκτέλεσης το οποίο προστατεύει τόσο τον Πράκτορα όσο και το δίκτυο που τον περιβάλλει από τις συνέπειες της εκτέλεσης του μη έμπιστου λογισμικού. Επίσης, οι επεξεργαστικοί πόροι που παρέχονται σε κάθε εκτέλεση είναι περιορισμένοι, καθώς το ύποπτο λογισμικό θα πρέπει να εκτελεστεί εντός ενός συγκεκριμένου χρονικού διαστήματος.

Πέραν από την εξέταση της διερχόμενης κίνησης, η πλατφόρμα SEDUCE μπορεί να λειτουργήσει και ως ερευνητικό εργαλείο, για τον εντοπισμό και την ανάλυση κακόβουλου λογισμικού. Ο Μηχανισμός δυναμικής ανάλυσης qemu μπορεί να παρέχει χρήσιμες πληροφορίες για όλες τις καταστάσεις του υπό διερεύνηση λογισμικού (π.χ. καταγραφή των μονοπατιών εκτέλεσης κλπ.), ενώ ο Μηχανισμός pyopty2 απέδειξε ότι είναι ιδιαίτερα εύκολο να υλοποιηθεί ένα άρθρωμα της πλατφόρμας σε γλώσσα υψηλότερου επιπέδου χωρίς να θυσιάζεται σημαντικά η ταχύτητα εξέτασης των πακέτων. Επίσης, η ύπαρξη συγκεκριμένων διεπαφών μέσω των οποίων επικοινωνούν τα διάφορα συστατικά της πλατφόρμας κάνει πιο εύκολη τη χρήση μέρους αυτής (π.χ. των Μηχανισμών Ανίχνευσης Κακόβουλου Λογισμικού) από τρίτες εφαρμογές.

Η πλατφόρμα SEDUCE τριών επιπέδων επιτρέπει την αξιοποίηση της κατανεμημένης πλατφόρμας και από άλλους μηχανισμούς ανίχνευσης κακόβουλου λογισμικού, οι οποίοι μπορούν να εξετάσουν μεταξύ άλλων και πρωτόκολλα σε επίπεδο εφαρμογής.

Κατά την πειραματική αξιολόγηση της πλατφόρμας εντοπίστηκε πρόβλημα κατά την επεξεργασία κίνησης υψηλών ταχυτήτων. Για να μειωθεί σημαντικά η απώλεια πακέτων σε αυτές τις ταχύτητες, θα μπορούσαν να χρησιμοποιηθούν τεχνικές όπως αυτές που περιγράφονται στις εργασίες [99] και [100], οι οποίες εγγυώνται ταχύτερη μεταφορά δεδομένων από το δικτυακό ελεγκτή στην εφαρμογή. Επίσης, κάθε πακέτο που λαμβάνει ο Αισθητήρας θα μπορούσε να τοποθετείται σε ήδη δεσμευμένη μνήμη (SLAB memory allocation) ώστε να μη καταναλώνεται χρόνος για την αναδιοργάνωση και διαχείριση του σωρού.

Μέσω της πειραματικής αξιολόγησης αποδείχθηκε ότι ο Μηχανισμός Ανίχνευσης qemu μπορεί να εντοπίσει τη συντριπτική πλειοψηφία κακόβουλων payload, πολυμορφικών και μη. Η μέθοδος υβριδικής ανάλυσης αν και πιο αποδοτική, δεν κατάφερε να εντοπίσει με-

γάλο μέρος των κακόβουλων payload, ακόμη και όταν αυτά δεν είχαν κωδικοποιηθεί. Το πρόβλημα αυτό οφείλεται στο γεγονός ότι η βιβλιοθήκη libemu στηρίζεται στον εντοπισμό κώδικα τύπου GetPC κατά την ανίχνευση shellcode, ο οποίος κώδικας δεν αποτελεί πάντα μέρος ενός shellcode. Ο χρόνος ανάλυσης για το Μηχανισμό qemu μπορεί να βελτιωθεί με τη χρήση διαφορετικών νημάτων για την εξέταση του payload από διαφορετικές θέσεις.

Η μόνη περίπτωση κωδικοποίησης που στάθηκε εμπόδιο για τη μέθοδο δυναμικής ανάλυσης ήταν η κωδικοποίηση τύπου "Context-keyed payload encoding". Συγκεκριμένα όταν η μέθοδος αυτή χρησιμοποιούσε τα αποτελέσματα εντολών επεξεργαστή για να αποκωδικοποιήσει το payload τότε το εικονικό περιβάλλον εκτέλεσης δεν μπορούσε να εκτελέσει το payload με σωστό τρόπο, καθώς δε γνώριζε τις τιμές που θα έπρεπε να επιστρέψει η σχετική εντολή επεξεργαστή. Μια μέθοδος για τον εντοπισμό τέτοιων εντολών είναι ο χαρακτηρισμός συγκεκριμένων εντολών ως επικίνδυνων, όπως ακριβώς γίνεται δηλαδή για τις κλήσεις συστήματος. Βέβαια κάτι τέτοιο δε μπορεί να εφαρμοστεί σε μια άλλη μορφή κωδικοποίησης τύπου "context-keyed payload encoding", όπου ως κλειδί αποκρυπτογράφησης χρησιμοποιούνται τα περιεχόμενα μιας θέσης μνήμης της υπηρεσίας που αποτελεί στόχο της επίθεσης.

Τέλος, μια άλλη κατηγορία από exploit που δεν είναι δυνατόν να ανιχνευθούν από πλατφόρμες σαν τη SEDUCE είναι αυτά που έχουν υλοποιηθεί με τη μέθοδο ROP (return oriented programming) [77]. Ένα ROP exploit μπορεί να εκτελεστεί δίχως να βρεθεί σε εκτελέσιμες σελίδες στη στοίβα (ή το σωρό). Συγκεκριμένα, μέσω διευθύνσεων επιστροφής που τοποθετεί στη στοίβα, το ROP exploit εκτελεί κώδικα που υπάρχει ήδη στη μνήμη μιας εφαρμογής. Επειδή θεωρητικά ο κώδικας ενός ROP exploit δεν περιλαμβάνεται στο ίδιο το exploit, είναι αδύνατο να εντοπιστεί από μεθόδους όπως η δυναμική ανάλυση που προσφέρει η πλατφόρμα SEDUCE. Ωστόσο, επειδή το payload που θα εκτελέσει ένας επιτιθέμενος με ένα ROP exploit είναι συνήθως τυποποιημένο, το ROP τμήμα του exploit περιορίζεται μόνο στην υλοποίηση των διαδικασιών φόρτωσης και εκτέλεσης του payload. Έτσι, αν ο ROP κώδικας δεν περιλαμβάνει κάποιο αποκωδικοποιητή για το payload, τότε το payload θα είναι ανιχνεύσιμο από το Μηχανισμό qemu, αφού θα εμπεριέχεται και αυτό (ως έχει) στο exploit που απέστειλε ο επιτιθέμενος.

Κεφάλαιο 4

Διαχείριση Προγραμματιζόμενων Δικτυακών Υποδομών

Ο δυναμικός τρόπος με τον οποίο οι υπηρεσίες αξιοποιούν τις συσκευές μιας προγραμματιζόμενης δικτυακής πλατφόρμας βοηθά μεν στην επίλυση διαφόρων διαχειριστικών θεμάτων σχετικών με το δίκτυο που συγκροτεί η πλατφόρμα (π.χ. διευκολύνεται η συλλογή στατιστικών κίνησης από τους κόμβους μέσω της εκτέλεσης ειδικού λογισμικού σε αυτούς [114]), ταυτόχρονα όμως δημιουργεί πλήθος διαχειριστικών θεμάτων για την ίδια την προγραμματιζόμενη πλατφόρμα. Ο αρχιτέκτονας κάθε προγραμματιζόμενης δικτυακής πλατφόρμας καλείται να αποφασίσει μεταξύ άλλων:

- Ποιοι κόμβοι θα έχουν τη δυνατότητα να συμμετάσχουν στην προγραμματιζόμενη υποδομή.
- Ποιο λογισμικό θα επιτρέπεται να εκτελεστεί σε αυτούς.
- Ποιοι χρήστες (ή υπηρεσίες) θα έχουν τη δυνατότητα να αξιοποιήσουν την υποδομή.
- Ποια μέρη της υποδομής θα συμβάλλουν στην παροχή της κάθε υπηρεσίας.
- Με ποιο τρόπο θα αξιοποιούνται οι εξειδικευμένες λειτουργίες που παρέχουν συγκεκριμένοι κόμβοι.
- Πώς θα γίνεται ο εντοπισμός και η διαχείριση συμβάντων ασφαλείας.
- Πώς θα γίνεται η διαχείριση κάποιας βλάβης και τι επιπτώσεις θα έχει αυτή στους χρήστες των υπηρεσιών.

Η εύκολη πρόσβαση σε συστοιχίες υπολογιστών (τεχνολογίες Πλέγματος - Grid [115], Σύννεφου - Cloud [27] κλπ.), η δυνατότητα παράλληλης εκτέλεσης πολλών εικονικών συστημάτων σε σύγχρονους επεξεργαστές [116] αλλά και το μικρό κόστος παραγωγής ενσωματωμένων συστημάτων (όπως αυτά που χρησιμοποιούνται σε Δίκτυα Αισθητήρων [28]), καθιστούν πλέον δυνατή τη συμμετοχή ενός τεράστιου αριθμού συσκευών σε μια προγραμματιζόμενη πλατφόρμα. Η διαχείριση ενός δικτύου τέτοιων διαστάσεων με χειροκίνητο τρόπο είναι προφανώς απαγορευτική. Απαιτείται λοιπόν, τόσο ο αυτοματισμός των διαχειριστικών διαδικασιών όσο και η αυτο-οργάνωση των κόμβων κατά την παροχή υπηρεσιών.

Στην εργασία των Fu et al. [54] παρουσιάζεται μια πλατφόρμα η οποία επιτρέπει την οργάνωση και σύνθεση ευρύτερων υπηρεσιών από τις επί μέρους υπηρεσίες που προσφέρουν οι προγραμματιζόμενοι κόμβοι. Ως παράδειγμα δίνεται μια υπηρεσία μετάδοσης video η

οποία μεταλλάσσει το ψηφιακό σήμα ανάλογα με τις ανάγκες του πελάτη (ποιότητα σήματος, επιλογή κωδικοποίησης κλπ.). Προκειμένου να γίνει η καλύτερη δυνατή αξιοποίηση των προγραμματιζόμενων κόμβων (αλλά και του εύρους ζώνης του δικτύου) θα πρέπει το δίκτυο που θα προσφέρει αυτή την υπηρεσία να αυτο-οργανωθεί με δύο τρόπους:

- να εντοπίσει τις κατάλληλες επί μέρους υπηρεσίες που θα επεξεργαστούν το περιεχόμενο με τη μορφή που επιθυμεί ο χρήστης, και
- να δημιουργήσει μια εικονική ζεύξη μεταξύ των κατάλληλων κόμβων ώστε ο χρήστης να λάβει την ποιότητα υπηρεσίας που επιθυμεί (quality of service) και παράλληλα το δίκτυο να υποστεί τη μικρότερη δυνατή επιβάρυνση.

Η πρώτη μορφή αυτο-οργάνωσης συχνά απαιτεί την αλληλεπίδραση μεταξύ των υπηρεσιών προκειμένου η έξοδος κάθε υπηρεσίας να είναι συμβατή με την είσοδο της επόμενης υπηρεσίας στο εικονικό κανάλι ζεύξης. Στη δεύτερη μορφή αυτο-οργάνωσης, το δίκτυο θα πρέπει να επιλέξει το μικρότερο δυνατό μονοπάτι επικοινωνίας μεταξύ της πηγής και του κόμβου-πελάτη. Πολλές φορές αυτό σημαίνει τη μεταφορά κάποιων υπηρεσιών σε κόμβους που βρίσκονται πιο κοντά δικτυακά στον πελάτη.

Σημαντικά είναι και τα θέματα ασφάλειας που προκύπτουν κατά τη χρήση μιας προγραμματιζόμενης δικτυακής πλατφόρμας. Όπως κάθε δικτυακή πλατφόρμα, έτσι και αυτή, είναι ευάλωτη τόσο σε απομακρυσμένες όσο και σε τοπικές επιθέσεις. Στόχος αυτών των επιθέσεων μπορεί να είναι η αλλοίωση της ποιότητας μιας υπηρεσίας, η υποκλοπή ευαίσθητων στοιχείων, η εξαπάτηση χρηστών ή υπηρεσιών με χρήση πλαστών πιστοποιητικών, η μη νόμιμη χρήση μιας πλατφόρμας εκτέλεσης κ.α. Στην εργασία [117] προτείνεται μια σειρά μέτρων προστασίας που θα πρέπει να υλοποιεί μια προγραμματιζόμενη δικτυακή πλατφόρμα. Μεταξύ αυτών αναφέρονται η λειτουργία μιας υπηρεσίας σύμφωνα με κάποια συγκεκριμένη πολιτική (policy enforcement), η χρήση «ασφαλών γλωσσών προγραμματισμού» που μειώνουν την πιθανότητα ευπαθειών εξαιτίας σφαλμάτων στο λογισμικό καθώς και η χρήση μεθόδων εξέτασης του «ξένου» λογισμικού, προτού αυτό εκτελεστεί σε μια πλατφόρμα εκτέλεσης. Στο κεφάλαιο 5 θα γίνει μια πιο αναλυτική εξέταση των προβλημάτων ασφάλειας που συναντώνται στις προγραμματιζόμενες δικτυακές υποδομές καθώς και των μέτρων προστασίας που μπορούν να εφαρμοστούν σε αυτές.

Η ανάγκη διαχείρισης μιας προγραμματιζόμενης πλατφόρμας ως μια «δικτυακή συσκευή» ενός ευρύτερου δικτύου υπολογιστικών συστημάτων, οδήγησε στη δημιουργία λογισμικού όπως αυτό της εργασίας [118], το οποίο επιτρέπει σε μια πλατφόρμα εκτέλεσης να δηλώσει την κατάστασή της μέσω του καθιερωμένου πρωτοκόλλου SNMP. Το πρωτόκολλο αυτό επιτρέπει τόσο την εξέταση χαρακτηριστικών ενός κόμβου (στοιχεία επεξεργαστικού φορτίου, φορτίου δικτυακής κίνησης, πλήθος παρεχόμενων υπηρεσιών κ.α.) από μια εξωτερική οντότητα, όσο και την ενημέρωση μιας εξωτερικής οντότητας όταν κάποιο χαρακτηριστικό λάβει μια τιμή με ειδικό ενδιαφέρον (π.χ. κάποια τιμή η οποία υπερβαίνει ένα όριο το οποίο έχει τεθεί από το διαχειριστή του κόμβου). Με τη χρήση του λογισμικού αυτού, γίνεται πλέον δυνατή η επίβλεψη μιας προγραμματιζόμενης πλατφόρμας χρησιμοποιώντας τις υπάρχουσες εφαρμογές διαχείρισης δικτυακού εξοπλισμού που υποστηρίζουν το πρωτόκολλο SNMP.

Ένα ιδιαίτερα ενδιαφέρον πρόβλημα αποτελεί η βέλτιστη επιλογή των κόμβων που θα βοηθήσουν στην παροχή μιας υπηρεσίας. Κάθε υπηρεσία μπορεί να αποτελείται από περισσότερες από μία διαδικασίες οι οποίες μπορεί να μην είναι δυνατό να εκτελεστούν από όλους τους κόμβους του δικτύου. Στην εργασία [25] παρουσιάζεται μια πλατφόρμα χρονοπρογραμματισμού διαδικασιών, η οποία επιτρέπει τη "δρομολόγηση" των διαδικασιών σε κοντινούς δικτυακά κόμβους ώστε να ελαχιστοποιείται η επιβάρυνση που προκαλείται

στο δίκτυο κατά τη μεταφορά δεδομένων. Αντίστοιχα, στην εργασία [119] παρουσιάζεται ένα σύστημα που χρησιμοποιεί περισσότερους του ενός κόμβους για την εκτέλεση κάθε διαδικασίας προκειμένου να εγγυηθεί στο χρήστη ένα συγκεκριμένο επίπεδο ποιότητας εξυπηρέτησης ακόμη και σε περιπτώσεις όπου υπάρχουν βλάβες στο λογισμικό ή στο υλικό της υποδομής.

Στο κεφάλαιο αυτό θα παρουσιαστεί μία νέα μέθοδος χρονοπρογραμματισμού η οποία έχει ως στόχο την καλύτερη ενεργειακή αξιοποίηση των κόμβων μιας προγραμματιζόμενης υποδομής. Η μέθοδος αυτή βρίσκει άμεση εφαρμογή στα Ασύρματα Δίκτυα Αισθητήρων και επιτρέπει την επέκταση του χρόνου ζωής του δικτύου πέραν του τυπικού χρόνου ζωής των κόμβων αυτού.

4.1 Οργάνωση κόμβων για την εξοικονόμηση ενέργειας σε Δίκτυα Αισθητήρων

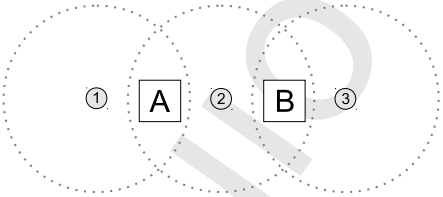
Οι προγραμματιζόμενες συσκευές που συναντώνται στα Ασύρματα Δίκτυα Αισθητήρων [28] έχουν περιορισμένα αποθέματα ενέργειας. Ο τύπος αισθητήρα που εξετάζεται σε αυτή την ενότητα, δεν έχει δυνατότητα κίνησης και επικοινωνεί με άλλους αισθητήρες με ad hoc τρόπο, δηλαδή χωρίς να βασίζεται σε κάποιο σταθερό δίκτυο κορμού για τις υπηρεσίες επικοινωνίας. Σε πολλές περιπτώσεις η τοποθέτηση τέτοιων αισθητήρων γίνεται με τυχαίο τρόπο (π.χ. ρίψη από αεροπλάνο σε στρατιωτικές επιχειρήσεις) και σε σημεία όπου είναι δύσκολο να επέμβει ο άνθρωπος (π.χ. ηφαιστειογενείς περιοχές), οπότε καθίσταται δύσκολη (ή επικίνδυνη) η συντήρηση των συσκευών αυτών. Συστήματα όπως αυτό που παρουσιάζουν στην εργασία τους οι Jiang et al. [120], χρησιμοποιούν την ηλιακή ενέργεια για να επεκτείνουν το χρόνο ζωής των αισθητήρων. Δυστυχώς όμως η χρήση τέτοιων ανενεώσιμων πηγών ενέργειας προϋποθέτει ότι ισχύουν ορισμένες συνθήκες στο περιβάλλον του Δικτύου Αισθητήρων. Μιας και το κόστος παραγωγής των αισθητήρων μειώνεται κάθε χρόνο, θεωρείται πιο συμφέρουσα λύση η αντικατάσταση των αισθητήρων που έχουν εξαντλήσει το ενεργειακό τους απόθεμα, παρά η συντήρηση αυτών. Έτσι, το ενδιαφέρον της ερευνητικής κοινότητας έχει στραφεί στην ανάπτυξη αλγορίθμων και τεχνικών που στοχεύουν στη χαμηλότερη κατανάλωση ενέργειας, επιτρέποντας στο δίκτυο να λειτουργήσει με τους ίδιους αισθητήρες για μεγαλύτερο χρονικό διάστημα.

Οι τεχνικές εξοικονόμησης ενέργειας που προτείνονται στη βιβλιογραφία, αφορούν κυρίως στην πρόσβαση στο δικτυακό μέσο (MAC) [121], στη δρομολόγηση πακέτων σε Δίκτυα Αισθητήρων [122], στη συλλογή πληροφορίας από τους αισθητήρες [123], στη διαχείριση γεγονότων στο λειτουργικό σύστημα των αισθητήρων [124] αλλά και στο χρονοπρογραμματισμό που θα ορίσει τελικά ποιοι αισθητήρες θα παρέχουν μετρήσεις από το περιβάλλον και για ποιο χρονικό διάστημα [125]. Η αναζήτηση για το βέλτιστο χρονοδιάγραμμα λειτουργίας των αισθητήρων είναι ένα ιδιαίτερα ενδιαφέρον πρόβλημα, καθώς η λύση αυτού μπορεί να εφαρμοστεί σε οποιοδήποτε δίκτυο προγραμματιζόμενων οντοτήτων, όπου υπάρχουν πολλαπλοί διακομιστές για κάθε υπηρεσία και κάθε διακομιστής λειτουργεί εντός συγκεκριμένων ορίων που ορίζονται από το διαχειριστή του (μέγιστο πλήθος εξυπηρετήσεων, συνολική κατανάλωση ενέργειας κλπ.). Στην περίπτωση του Δικτύου Αισθητήρων, οι διακομιστές που προσφέρουν κοινή υπηρεσία είναι οι αισθητήρες που παρέχουν κάλυψη σε μια συγκεκριμένη περιοχή, ενώ το όριο που εφαρμόζεται σε κάθε αισθητήρα είναι η μέγιστη κατανάλωση ενέργειας, που δε θα πρέπει να υπερβαίνει τα αποθέματα ενέργειας αυτού.

Έστω ότι κάθε αισθητήρας ενός δικτύου μπορεί να λειτουργήσει σε κατάσταση *active* (βλ. ενότητα 1.2) επί h ώρες. Ενεργοποιώντας όλους τους αισθητήρες του δικτύου ταυτόχρονα,

το δίκτυο θα προσφέρει (ιδανικά) h ώρες *κάλυψης* των υπό παρατήρηση περιοχών (θεωρώντας ότι όλοι οι αισθητήρες έχουν όμοια αποθέματα ενέργειας και ίδια κατανάλωση). Ανάλογα με την τοπολογία του δικτύου και τον τύπο της κάλυψης (πλήρης ή μερική) που αυτό προσφέρει, είναι δυνατό να επεκταθεί ο συνολικός χρόνος κάλυψης χρησιμοποιώντας την τεχνική του χρονοπρογραμματισμού. Συγκεκριμένα, ο χρόνος κάλυψης χωρίζεται σε βάρδιες και για την εκτέλεση κάθε βάρδιας είναι αποκλειστικά υπεύθυνο ένα υποσύνολο των διαθέσιμων αισθητήρων, το οποίο ονομάζεται *ομάδα κάλυψης* (cover set). Αν μία βάρδια διαρκεί h' ώρες και $|C|$ είναι το πλήθος των ομάδων κάλυψης, τότε το δίκτυο μπορεί να προσφέρει συνολικά $h' \cdot |C|$ ώρες κάλυψης. Κατά τη διάρκεια μίας βάρδιας, οι αισθητήρες που προσφέρουν υπηρεσίες κάλυψης βρίσκονται σε κατάσταση *active*, ενώ οι υπόλοιποι αισθητήρες βρίσκονται σε κατάσταση *sleeping* ώστε να εξοικονομείται ενέργεια.

Το πρόβλημα παραγωγής του μέγιστου αριθμού ομάδων κάλυψης είναι γνωστό στο χώρο των Δικτύων Αισθητήρων ως πρόβλημα της ενεργειακά αποδοτικής κάλυψης αισθητήρων (energy-efficient sensor coverage problem) [125]. Ανάλογα με τις ανάγκες του διαχειριστή του δικτύου, οι παραγόμενες ομάδες μπορεί να έχουν κοινά μέλη (disjoint cover sets) ή όχι (non-disjoint cover sets). Στην περίπτωση των ομάδων δίχως κοινά μέλη, δίνεται έμφαση στην αξιοπιστία του δικτύου καθώς οποιαδήποτε βλάβη σε μια ομάδα κάλυψης δε θα επηρεάσει τη λειτουργία των υπόλοιπων ομάδων. Αντίθετα, στην περίπτωση που επιτρέπεται σε δύο ή περισσότερες ομάδες να έχουν κοινά μέλη, δίνεται έμφαση στη μακροβιότητα του δικτύου, καθώς παράγονται περισσότερες (αλλά εξαρτημένες) ομάδες κάλυψης.

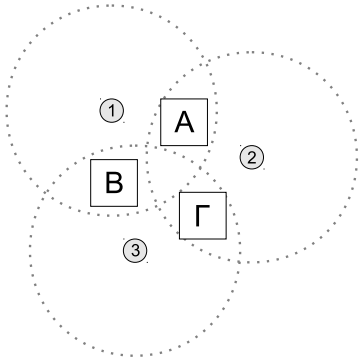


Σχήμα 4.1: Παράδειγμα τοπολογίας με 2 στόχους και 3 αισθητήρες

Στο σχήμα 4.1 παρουσιάζεται ένα δίκτυο αισθητήρων και στόχων. Οι στόχοι περιγράφονται με τα γράμματα του ελληνικού αλφαβήτου ενώ οι αισθητήρες με αριθμούς. Οι δίσκοι γύρω από κάθε αισθητήρα συμβολίζουν την περιοχή κάλυψης του αισθητήρα. Έτσι οι αισθητήρες 1 και 2 μπορούν να παρατηρήσουν το στόχο A, ή αλλιώς $A = \{1, 2\}$. Ομοίως έχουμε $B = \{2, 3\}$. Χρησιμοποιώντας έναν αλγόριθμο εύρεσης ομάδων κάλυψης δίχως κοινά στοιχεία (disjoint) στο σενάριο αυτό, προκύπτουν οι ομάδες $\{2\}$ και $\{1, 3\}$. Οι δύο αυτές ομάδες μας δίνουν συνολικό χρόνο κάλυψης των στόχων $2 \cdot h$, όπου h είναι ο χρόνος που μπορεί ένας τέτοιος αισθητήρας να παραμείνει σε κατάσταση *active*.

Στο παράδειγμα του σχήματος 4.2 ένας αλγόριθμος που φτιάχνει ομάδες κάλυψης δίχως κοινά στοιχεία, είναι ικανός να προτείνει μία, το πολύ, ομάδα κάλυψης (π.χ. $\{1, 2\}$) και έτσι ο συνολικός χρόνος ζωής του δικτύου παραμένει h. Αντίθετα, ένας αλγόριθμος που θα μπορούσε να χρησιμοποιήσει κάθε αισθητήρα σε δύο ομάδες, θα μπορούσε να προτείνει τρεις ομάδες κάλυψης, π.χ. $\{1, 2\}$, $\{2, 3\}$, $\{1, 3\}$. Σε αυτή την περίπτωση, επειδή ο κάθε αισθητήρας θα πρέπει να χρησιμοποιηθεί δύο φορές, ο χρόνος ζωής της μίας ομάδας κάλυψης (δηλ. ο χρόνος που θα διαρκέσει η βάρδια της) θα είναι $0.5 \cdot h$. Συνεπώς, ο συνολικός χρόνος κάλυψης που θα προσφέρει το δίκτυο (θεωρώντας αμελητέα την κατανάλωση ενέργειας όταν οι αισθητήρες είναι σε κατάσταση *sleeping*), θα είναι $1.5 \cdot h$.

Τα σενάρια κάλυψης στα ασύρματα δίκτυα αισθητήρων χωρίζονται σε σενάρια *κάλυψης σημείων* (point coverage) και σενάρια *κάλυψης περιοχών* (area coverage). Σε ένα σενάριο κάλυψης σημείων, μια ομάδα κάλυψης παρακολουθεί συγκεκριμένα σημεία του υπό επιτή-



Σχήμα 4.2: Παράδειγμα τοπολογίας με 3 στόχους και 3 αισθητήρες

ρηση χώρου. Αντίθετα, σε ένα σενάριο κάλυψης περιοχών, η ομάδα κάλυψης παρακολουθεί συγκεκριμένες περιοχές του υπό επιτήρηση χώρου, οι οποίες ονομάζονται *πεδία* (fields). Συγκεκριμένα, δύο σημεία ανήκουν στο ίδιο πεδίο αν και μόνο αν καλύπτονται από κοινούς αισθητήρες. Η περιοχή που καταλαμβάνει ένα πεδίο οριοθετείται από την τομή της ευρύτερης υπό παρακολούθηση περιοχής και των περιοχών που μπορούν να καλύψουν οι αισθητήρες που ανήκουν στο πεδίο αυτό. Κάθε αισθητήρας μπορεί να καλύπτει περισσότερα από ένα πεδία και ένα πεδίο καλύπτεται από τουλάχιστον ένα αισθητήρα.

Οι συγγραφείς της εργασίας [126] παρατηρούν ότι αν τα πεδία θεωρηθούν ως «σημεία», τότε κάθε σενάριο κάλυψης περιοχών ανάγεται σε ένα σενάριο κάλυψης σημείων. Συνεπώς, μία μέθοδος παραγωγής ομάδων κάλυψης που είναι ικανή να επιλύσει το πρόβλημα της κάλυψης σημείων, αρκεί για να επιλύσει και το πρόβλημα της κάλυψης περιοχών. Στην παρούσα διατριβή θα χρησιμοποιηθεί ο γενικότερος όρος «κάλυψη στόχων» (target coverage) για να περιγράψει τόσο τα σενάρια κάλυψης σημείων όσο και τα σενάρια κάλυψης περιοχών.

4.2 Έρευνα πεδίου σε θέματα κάλυψης

Οι Cardei και Wu ταξινομούν τους αλγόριθμους παραγωγής ομάδων κάλυψης, σύμφωνα με τα παρακάτω χαρακτηριστικά [125]:

1. Στόχος του αλγορίθμου (π.χ. η επέκταση του χρόνου ζωής του δικτύου, η χρήση του ελάχιστου δυνατού αριθμού αισθητήρων κ.α.)
2. Τοπολογία των αισθητήρων (τυχαία τοποθέτηση ή τοποθέτηση σύμφωνα με κάποιο μοντέλο).
3. Ομοιογένεια (ή μη) στα τεχνικά χαρακτηριστικά των αισθητήρων.
4. Κεντρικοποιημένη ή κατανεμημένη λειτουργία του αλγορίθμου.
5. Περαιτέρω χαρακτηριστικά που αφορούν την κατανάλωση ενέργειας και τη διασφάλιση της επικοινωνίας των αισθητήρων με τη *βάση* (βλ. ενότητα 1.2).

Στη λίστα αυτή θα μπορούσε κανείς να προσθέσει και τη δυνατότητα παραγωγής ομάδων κάλυψης με κοινούς αισθητήρες (non-disjoint cover sets), μιας και μερικοί από τους αλγόριθμους που έχουν προταθεί στη βιβλιογραφία παρέχουν τη δυνατότητα αυτή.

Στις επόμενες υποενότητες θα εξεταστεί μια σειρά κεντρικοποιημένων αλγορίθμων για την παραγωγή ομάδων κάλυψης. Οι αλγόριθμοι αυτοί θεωρούν ότι όλοι οι αισθητήρες έχουν

ομοιογενή τεχνικά χαρακτηριστικά και η τοποθέτησή τους έχει γίνει με τυχαίο τρόπο. Στη βιβλιογραφία υπάρχουν και κατανεμημένοι αλγόριθμοι για το χρονοπρογραμματισμό αισθητήρων (βλ. [127], [128]) οι οποίοι διαμορφώνουν το συνολικό πρόγραμμα λειτουργίας του δικτύου μέσω τοπικών αποφάσεων. Οι αλγόριθμοι αυτοί παρουσιάζουν μεγαλύτερη κατανάλωση ενέργειας σε σύγκριση με τους κεντρικοποιημένους, καθώς απαιτείται περαιτέρω επικοινωνία των αισθητήρων μεταξύ τους, προκειμένου να αποφευχθούν προβλήματα συγχρονισμού. Επίσης, λειτουργούν με τρόπο βέλτιστης προσπάθειας (*best–effort*), δίχως δηλαδή να παρέχουν εγγυήσεις για πλήρη κάλυψη ή κάποιας μορφής εγγυημένη κάλυψη σε συγκεκριμένες περιοχές. Τέλος, σε περιπτώση βλάβης ενός αισθητήρα είναι δυνατό να χαθεί ένα μονοπάτι επικοινωνίας με τη *βάση*, δημιουργώντας έτσι *τυφλά σημεία* στις υπό παρακολούθηση περιοχές. Τα προβλήματα αυτά οφείλωνται κυρίως στην περιορισμένη "εικόνα" που έχουν οι αισθητήρες για το δίκτυο στο οποίο συμμετέχουν. Συγκεκριμένα, προκειμένου να διατηρηθεί σε χαμηλά επίπεδα η κατανάλωση ενέργειας, κάθε αισθητήρας ενημερώνεται μόνο για την κατάσταση των γειτονικών του αισθητήρων. Έτσι όμως, γίνεται δύσκολη έως αδύνατη η εξεύρεση λύσεων σε προβλήματα που αφορούν το δίκτυο συνολικά (π.χ. εναλλακτική δρομολόγηση, αποφυγή *τυφλών σημείων* κ.α.). Για τους λόγους αυτούς, στη διατριβή αυτή, θα μελετηθούν κυρίως κεντρικοποιημένοι αλγόριθμοι χρονοπρογραμματισμού, οι οποίοι βασίζονται στις επεξεργαστικές δυνατότητες της *βάσης* και προσφέρουν λύσεις εξετάζοντας παραμέτρους του συνολικού δικτύου αισθητήρων.

4.2.1 Παραγωγή ομάδων κάλυψης δίχως κοινούς αισθητήρες

Στην εργασία των Slijepcevic και Potkonjak [129] παρουσιάζεται ένας κεντρικοποιημένος αλγόριθμος, ο οποίος παρέχει ομάδες κάλυψης δίχως κοινούς αισθητήρες. Στόχος του αλγορίθμου είναι η πλήρης κάλυψη συγκεκριμένων περιοχών, οι οποίες χωρίζονται σε *πεδία* (βλ. ενότητα 4.1). Ο αλγόριθμος ξεκινά καλύπτοντας πρώτα τα πεδία με τη μικρότερη δυνατότητα κάλυψης (δηλαδή, τα πεδία που καλύπτονται από το μικρότερο αριθμό αισθητήρων). Τα πεδία αυτά ονομάζονται *κρίσιμα*. Μόλις επιλεχθεί ένας αισθητήρας που μπορεί να καλύψει ένα Κρίσιμο Πεδίο, ο αλγόριθμος λαμβάνει τα απαραίτητα μέτρα ώστε να μην επιλεχθεί, στα πλαίσια της ίδιας ομάδας κάλυψης, άλλος αισθητήρας που θα καλύψει το πεδίο αυτό. Η πολυπλοκότητα του αλγορίθμου των Slijepcevic και Potkonjak είναι $O(n^2)$, όπου n είναι το πλήθος αισθητήρων που εξετάζονται από τον αλγόριθμο.

Οι Cardei et al. παρέχουν μια μοντελοποίηση του ίδιου προβλήματος στην εργασία [130], χρησιμοποιώντας γράφους. Συγκεκριμένα, δημιουργείται ένας γράφος $G = (V, E)$, όπου V το σύνολο των αισθητήρων και E ένα σύνολο ακμών, τέτοιο ώστε η ακμή $(u, v) \in E$ αν και μόνο αν οι αισθητήρες u, v βρίσκονται ο ένας εντός της ακτίνας παρακολούθησης του άλλου. Στο γράφο αυτό εφαρμόζεται μια τεχνική «χρωματισμού» (graph colouring) προκειμένου να παραχθεί ο μέγιστος αριθμός «κυρίαρχων συνόλων» (dominating sets). Οι συγγραφείς της εργασίας προτείνουν έναν ευρετικό αλγόριθμο με πολυπλοκότητα $O(n^3)$ για τη λύση αυτού του προβλήματος. Όπως διαπιστώνεται όμως στην εργασία [131], ο προτεινόμενος αλγόριθμος παρόλο που παράγει περισσότερες ομάδες κάλυψης από αυτόν που προτάθηκε στην εργασία [129], δεν είναι ικανός να παρέχει πλήρη κάλυψη σε όλες τις επιθυμητές περιοχές του υπό παρατήρηση χώρου.

Στην εργασία [126] οι Cardei και Du αποδεικνύουν ότι το πρόβλημα κάλυψης στόχων σε δίκτυα ασύρματων αισθητήρων είναι ένα *NP-Complete* πρόβλημα, καθώς αποτελεί μια γενίκευση του γνωστού προβλήματος 3-SAT [132]. Στην ίδια εργασία, προτείνουν τη μοντελοποίηση του προβλήματος ως πρόβλημα «μέγιστης ροής» (maximum flow). Χρησιμοποιώντας τη μέθοδο μαθηματικού προγραμματισμού *Mixed Integer Programming*, επιλύουν το πρόβλημα «μέγιστης ροής» και παράγουν μια σειρά από ομάδες κάλυψης. Τα αποτε-

λέσματα της προτεινόμενης μεθόδου, αν και είναι καλύτερα από αυτά της εργασίας [129], παράγονται με πιο αργό τρόπο. Επίσης, η πολυπλοκότητα της προτεινόμενης μεθόδου δεν είναι συγκεκριμένη, καθώς εξαρτάται από την πολυπλοκότητα της μεθόδου μαθηματικού προγραμματισμού που θα χρησιμοποιηθεί στα δεδομένα του προβλήματος «μέγιστης ροής».

Οι Abrams et al. [133] προτείνουν έναν τυχαιοποιημένο αλγόριθμο για την επίλυση του προβλήματος παραγωγής ομάδων κάλυψης περιοχών. Όμως, ο προτεινόμενος αλγόριθμος δεν παράγει ομάδες κάλυψης οι οποίες μπορούν να προσφέρουν πλήρη κάλυψη, ανεξάρτητα η μία από την άλλη. Επίσης οι ομάδες που παράγονται θα πρέπει να χρονοπρογραμματισθούν σειριακά, ώστε να επιτευχθεί κάλυψη ενός σημαντικού τμήματος (80%) των υπό παρατήρηση περιοχών. Η πολυπλοκότητα του προτεινόμενου αλγόριθμου είναι $O(nm|P_{max}|)$, όπου n είναι ο αριθμός διαθέσιμων αισθητήρων, m ο αριθμός παραγόμενων ομάδων κάλυψης και $|P_{max}|$ το μέγιστο πλήθος πεδίων που μπορεί να καλύψει ένας αισθητήρας.

4.2.2 Παραγωγή ομάδων κάλυψης με κοινούς αισθητήρες

Η εργασία [134] των Cardei et al. προτείνει μία λύση στο πρόβλημα παραγωγής ομάδων κάλυψης η οποία εκμεταλλεύεται τη δυνατότητα χρήσης ενός αισθητήρα σε περισσότερες από μία ομάδες κάλυψης (non-disjoint cover sets). Για την παραγωγή των ομάδων κάλυψης οι συγγραφείς χρησιμοποιούν μια μέθοδο βασισμένη στο γραμμικό προγραμματισμό. Ο προτεινόμενος αλγόριθμος έχει πολυπλοκότητα $O(m^3n^3)$, όπου m είναι ο αριθμός των παραγόμενων ομάδων κάλυψης και n το πλήθος των διαθέσιμων αισθητήρων. Στην ίδια εργασία παρουσιάζεται επίσης ένας «άπληστος» αλγόριθμος (greedy algorithm) μικρότερης πολυπλοκότητας $O(dk^2n)$, όπου d είναι το πλήθος των αισθητήρων που καλύπτουν στόχους με μικρή δυνατότητα κάλυψης και k είναι το πλήθος των στόχων που πρέπει να καλυφθούν.

Η τεχνική *Linear Programming* συναντάται και στην εργασία [135] των Berman et al. Στη συγκεκριμένη περίπτωση, ο αλγόριθμος παράγει πρώτα ένα πλήθος ομάδων κάλυψης και στη συνέχεια επιλέγει το βέλτιστο χρονικό προγραμματισμό για αυτές. Η μέθοδος αυτή βασίζεται σε μια προσέγγιση τύπου $(1+\epsilon)$-approximation του αλγορίθμου των Garg και Könemann [136], με βαθμό προσέγγισης (approximation factor) ίσο με $(1+\epsilon)(1+2\ln n)$ για κάθε $\epsilon > 0$.

4.3 Μοντελοποίηση του προβλήματος πλήρους κάλυψης

Στην ενότητα αυτή θα παρουσιαστεί το πρόβλημα *πλήρους κάλυψης* καθώς και ένα γενικό μοντέλο για την ανάπτυξη ευρετικών αλγορίθμων παραγωγής ομάδων κάλυψης. Με βάση το μοντέλο αυτό, θα παρουσιαστεί στην ενότητα 4.4 ένας κεντρικοποιημένος αλγόριθμος υψηλών επιδόσεων που παρέχει πλήρη κάλυψη στόχων.

4.3.1 Παράμετροι του προβλήματος

Έστω $T_0 = \{t_1, t_2, \ldots, t_k\}$ το σύνολο των στόχων και $S_0 = \{s_1, s_2, \ldots, s_n\}$ το σύνολο των αισθητήρων. Κάθε στόχος του συνόλου T_0 καλύπτεται από τουλάχιστον έναν αισθητήρα του συνόλου S_0.

Κάθε αισθητήρας μπορεί να είναι εξοπλισμένος με πλήθος υποσυστημάτων τα οποία παρακολουθούν διάφορα χαρακτηριστικά του περιβάλλοντος. Κάθε υποσύστημα από αυτά έχει μια ακτίνα παρακολούθησης, η μικρότερη εκ των οποίων είναι R_s. Αυτό σημαίνει ότι ένας αισθητήρας s μπορεί να λάβει πλήρεις μετρήσεις από οποιοδήποτε στόχο (ή αλλιώς

να *καλύψει* οποιοδήποτε στόχο) βρίσκεται εντός του κύκλου με κέντρο τον αισθητήρα και ακτίνα R_s.

Καλούμε N_i το σύνολο των *γειτονικών αισθητήρων* του στόχου t_i. Κάθε γειτονικός αισθητήρας $s_j \in N_i$ είναι ικανός να καλύψει τον στόχο t_i, δηλ.:

$$\forall\ s_j \in N_i : |t_i - s_j| \leq R_s, N_i \subseteq S_0, t_i \in T_0,$$

όπου $|t_i - s_j|$ είναι η απόσταση του στόχου t_i από τον αισθητήρα s_j.

Η *είσοδος* I ενός αλγορίθμου παραγωγής ομάδων κάλυψης είναι ένα σύνολο δυάδων της μορφής:

$$I = \{(t_1, N_1), \ldots, (t_k, N_k)\}, t_i \in T_0, k = |T_0|,$$

όπου N_i είναι το σύνολο γειτονικών αισθητήρων του στόχου t_i. Το σύνολο I αποτελείται από μία δυάδα ανά στόχο του συνόλου T_0. Θα πρέπει να σημειωθεί ότι ένας αισθητήρας μπορεί να συμμετέχει σε περισσότερα από ένα σύνολα γειτονικών αισθητήρων.

Η έξοδος ενός τέτοιου αλγορίθμου είναι μια συλλογή $C = \{C_1, \ldots, C_m\}$ από m ομάδες κάλυψης. Κάθε ομάδα κάλυψης C_p αποτελεί υποσύνολο των διαθέσιμων αισθητήρων ($C_p \subseteq S_0$) και καλύπτει όλους τους στόχους που υπάρχουν στο T_0. Ο αριθμός εμφανίσεων w ($w \in \mathbb{N}^*$) ενός αισθητήρα στις παραγόμενες ομάδες κάλυψης, εξαρτάται από το είδος των ομάδων κάλυψης που επιθυμεί να παράγει ο χρήστης. Με $w = 1$ ο αλγόριθμος παράγει ομάδες κάλυψης που δεν έχουν κοινά στοιχεία (node disjoint cover sets), δηλ.:

$$\forall\ i, j : C_i \cap C_j = \emptyset,\ \ i, j \in [1, m], i \neq j.$$

Αντίθετα, με $w > 1$ ο αλγόριθμος επιτρέπει την παραγωγή ομάδων κάλυψης με κοινά στοιχεία (non-disjoint cover sets).

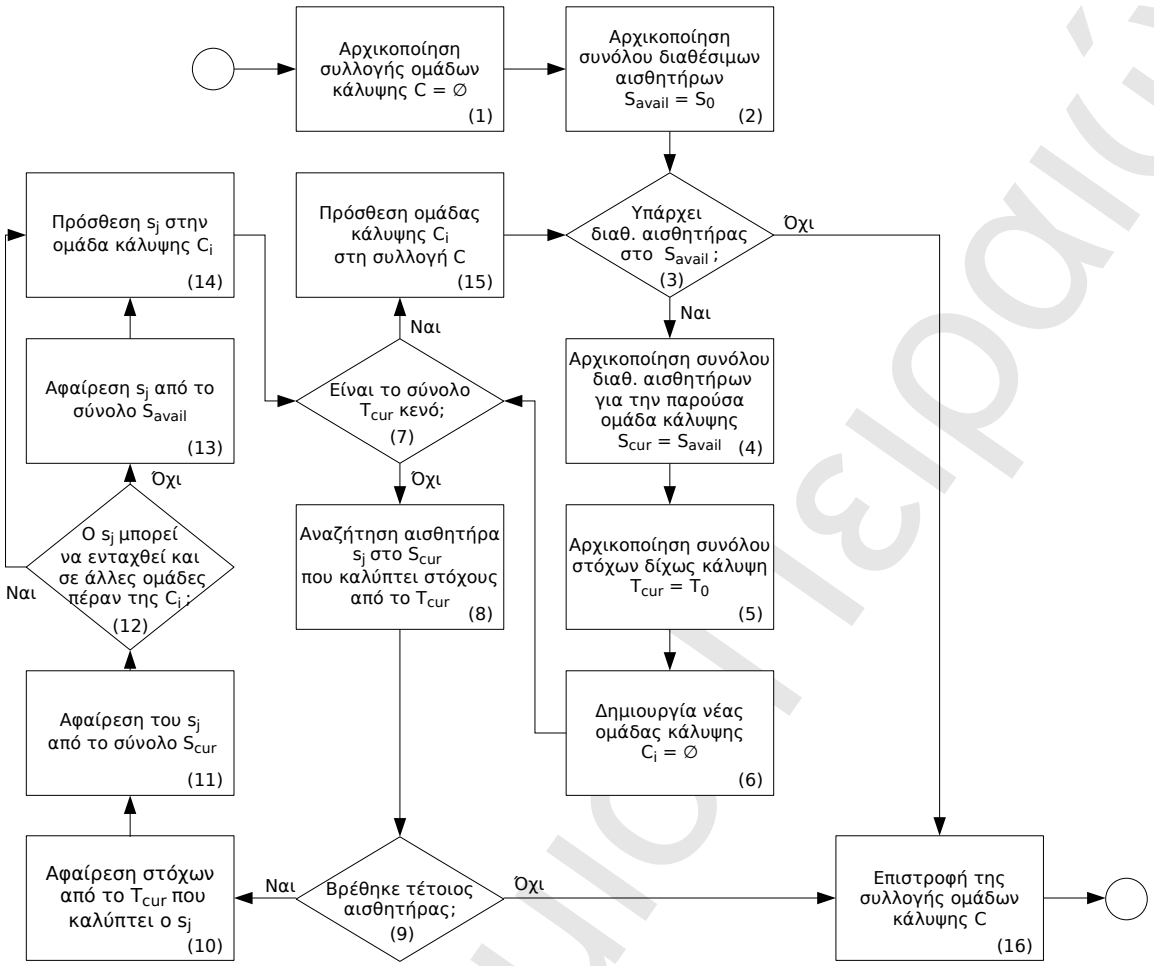
Ο στόχος ενός αλγορίθμου παραγωγής ομάδων κάλυψης είναι η επέκταση του χρόνου ζωής του δικτύου αισθητήρων, δηλαδή η μεγιστοποίηση του πλήθους $|C|$ των ομάδων κάλυψης που υπάρχουν στη παραγόμενη συλλογή C.

4.3.2 Παραγωγή ομάδων κάλυψης

Στην ενότητα 4.2 παρουσιάστηκαν διάφοροι αλγόριθμοι από τη βιβλιογραφία, που έχουν ως στόχο την παραγωγή ομάδων κάλυψης για χρήση σε δίκτυα αισθητήρων. Μερικοί από αυτούς τους αλγόριθμους ανήκουν στην οικογένεια των άπληστων ευρετικών αλγορίθμων (greedy heuristic algorithms). Παρόλο που οι αλγόριθμοι αυτοί εξετάζουν μέρος των δυνατών λύσεων του προβλήματος, παράγουν σε μικρό χρονικό διάστημα πολύ καλά αποτελέσματα που συχνά ταυτίζονται με τα βέλτιστα.

Στο σχήμα 4.3 παρουσιάζεται η γενική μορφή ενός άπληστου ευρετικού αλγορίθμου για την παραγωγή ομάδων κάλυψης. Κάθε αλγόριθμος αυτής της οικογένειας αποτελείται από δύο εμφωλευμένους βρόχους. Η συνθήκη στην οποία βασίζεται ο εξωτερικός βρόχος είναι η ύπαρξη διαθέσιμων αισθητήρων (βήμα 3), δηλαδή αισθητήρων που θα μπορούσαν να συμπεριληφθούν σε μια νέα ομάδα κάλυψης. Η συνθήκη στην οποία βασίζεται ο εσωτερικός βρόχος είναι η ύπαρξη στόχων που δεν έχουν καλυφθεί ακόμη από τους αισθητήρες μιας ομάδας κάλυψης (βήμα 7).

Κατά το μοντέλο αυτό, σε κάθε επανάληψη του εσωτερικού βρόχου, αναζητάται ένας αισθητήρας για εισαγωγή στην υπο κατασκευή ομάδα κάλυψης (βήμα 8). Η ομάδα κάλυψης θεωρείται πλήρης μόλις αποκτήσει ένα σύνολο αισθητήρων ικανό για να παρέχει το είδος της κάλυψης που επιθυμεί ο χρήστης (βήμα 7). Μόλις μια ομάδα κάλυψης γίνει πλήρης, προστίθεται στη συλλογή ομάδων κάλυψης (βήμα 15) και ξεκινούν οι διαδικασίες για την παραγωγή μιας νέας ομάδας κάλυψης (βήματα 3, 4, 5 & 6). Ο αλγόριθμος τερματίζει όταν



Σχήμα 4.3: Παραγωγή ομάδων κάλυψης χρησιμοποιώντας έναν άπληστο ευρετικό αλγόριθμο

δεν υπάρχουν πλέον άλλοι διαθέσιμοι αισθητήρες (βήματα 3 & 9) και επιστρέφει στο χρήστη τη συλλογή ομάδων κάλυψης όπως έχει διαμορφωθεί μέχρι εκείνη τη στιγμή (βήμα 16).

Οι διαθέσιμοι αισθητήρες περιγράφονται από δύο σύνολα. Το σύνολο S_{avail} περιέχει τους αισθητήρες που θα είναι διαθέσιμοι για χρήση και σε επόμενες ομάδες κάλυψης, ενώ το σύνολο S_{cur} περιέχει τους αισθητήρες που είναι διαθέσιμοι για την κατασκευή της ομάδας κάλυψης που εξετάζεται εκείνη τη στιγμή. Προκειμένου να αποφευχθεί η χρήση ενός αισθητήρα περισσότερες από μία φορές σε μια ομάδα κάλυψης, κάθε αισθητήρας που εντάσσεται σε μια ομάδα, αφαιρείται από το σύνολο S_{cur} (βήμα 11). Όταν ο αλγόριθμος παράγει ομάδες που έχουν μεταξύ τους κοινά στοιχεία (non-disjoint cover sets), τότε το βήμα 12 ελέγχει αν ο επιλεγμένος αισθητήρας θα μπορούσε να χρησιμοποιηθεί και σε άλλες ομάδες. Αν κάτι τέτοιο δεν είναι εφικτό, ο αισθητήρας αφαιρείται από το σύνολο των συνολικά διαθέσιμων αισθητήρων S_{avail} (βήμα 13). Προφανώς ο έλεγχος αυτός δεν απαιτείται για τους αλγόριθμους που παράγουν ομάδες κάλυψης δίχως κοινά στοιχεία. Στην περίπτωση αυτή ο κάθε αισθητήρας αφαιρείται άμεσα από το σύνολο των συνολικά διαθέσιμων αισθητήρων S_{avail}, μόλις χρησιμοποιηθεί.

Το πλήθος των παραγόμενων ομάδων εξαρτάται κυρίως από τη στρατηγική που θα ακολουθηθεί κατά την επιλογή αισθητήρων (βήμα 8). Στις επόμενες υποενότητες θα παρουσιαστούν αναλυτικά οι διάφορες παράμετροι που επηρεάζουν το αποτέλεσμα του αλγορίθμου

και θα προταθούν τεχνικές που μπορούν να βελτιώσουν τον αριθμό των παραγόμενων ομάδων κάλυψης.

4.3.3 Περιοριστικοί παράγοντες

Ο στόχος με το μικρότερο αριθμό γειτονικών αισθητήρων εισάγει ένα άνω όριο στον αριθμό των ομάδων κάλυψης που μπορούν να παραχθούν. Συγκεκριμένα, αν το μικρότερο σύνολο γειτονικών αισθητήρων περιέχει x αισθητήρες, τότε ένας αλγόριθμος θα μπορεί να παράγει το πολύ $x \cdot w$ ομάδες κάλυψης (όπου w ο μέγιστος αριθμός εμφανίσεων ενός αισθητήρα στις παραγόμενες ομάδες). Στην εργασία [137], το όριο αυτό ονομάστηκε *Θεωρητικό Μέγιστο* (theoretical maximum) και μπορεί να υπολογιστεί πολύ εύκολα από την είσοδο I του αλγορίθμου (βλ. ενότητα 4.3.1). Η γνώση του Θεωρητικού Μέγιστου είναι ιδιαίτερα χρήσιμη σε ευρετικούς αλγορίθμους και αλγορίθμους τύπου "επέκτασης & οριοθέτησης" (branch & bound) καθώς τους επιτρέπει να διαπιστώσουν άμεσα αν κάποια λύση είναι βέλτιστη, δίχως να απαιτείται περαιτέρω εξέταση άλλων λύσεων του προβλήματος.

Δυστυχώς, το Θεωρητικό Μέγιστο δεν ισούται με το πραγματικό μέγιστο πλήθος ομάδων που θα μπορούσαν να παραχθούν από μια είσοδο I. Όπως προαναφέρθηκε, μόλις χρησιμοποιηθούν όλοι οι αισθητήρες που καλύπτουν στόχους με μικρά σύνολα γειτονικών αισθητήρων, θα γίνει αδύνατη η παραγωγή περαιτέρω ομάδων κάλυψης. Για το λόγο αυτό, οι στόχοι που είναι συνδεδεμένοι με μικρά σύνολα γειτονικών αισθητήρων ονομάζονται *Κρίσιμοι Στόχοι*. Η πιθανότητα επιλογής ενός αισθητήρα που καλύπτει έναν συγκεκριμένο στόχο αυξάνεται σε *πυκνά δομημένα* δίκτυα αισθητήρων όπου ο κάθε αισθητήρας καλύπτει μεγάλο ποσοστό των υπό παρατήρηση στόχων. Έτσι, κατά την παραγωγή μιας ομάδας κάλυψης σε ένα πυκνό δίκτυο, είναι πιθανό να συμπεριληφθούν περισσότεροι του ενός γειτονικοί αισθητήρες ενός στόχου. Όμως, κάθε επιπλέον αισθητήρας που καλύπτει έναν Κρίσιμο Στόχο κατά τη διάρκεια της ίδιας βάρδιας, μειώνει κατά μία μονάδα το πλήθος των παραγόμενων ομάδων κάλυψης. Συνεπώς, σε πυκνά δίκτυα αισθητήρων, το μέγιστο πλήθος των *δυνατών* ομάδων κάλυψης μπορεί να είναι μικρότερο του Θεωρητικού Μέγιστου. Από την έρευνα που έγινε στη βιβλιογραφία, δε βρέθηκε κάποια αποδοτική μέθοδος υπολογισμού του πραγματικού μέγιστου αριθμού ομάδων κάλυψης που μπορούν να παραχθούν για ένα δίκτυο αισθητήρων.

Προκειμένου να μειωθεί ο αριθμός των μη αναγκαίων καλύψεων των Κρίσιμων Στόχων, μπορεί να χρησιμοποιηθεί μία μέθοδος που συναντάται στην εργασία [129]. Συγκεκριμένα, κατά την κατασκευή μιας ομάδας κάλυψης, επιλέγεται πρώτα ένας αισθητήρας που καλύπτει τον πιο Κρίσιμο Στόχο (δηλ. το στόχο με το μικρότερο σύνολο γειτονικών αισθητήρων) και έπειτα επιλέγονται αισθητήρες που δε συμμετέχουν στο σύνολο των γειτονικών αισθητήρων του στόχου αυτού. Η μέθοδος αυτή απαιτεί την αναζήτηση διαθέσιμων αισθητήρων που καλύπτουν Κρίσιμους Στόχους, κάθε φορά που δημιουργείται μία νέα ομάδα κάλυψης. Μια διαφορετική αντιμετώπιση του προβλήματος αυτού βασίζεται στην ταξινόμηση των αισθητήρων σύμφωνα με ένα κριτήριο το οποίο περιγράφει τη σχέση τους με τους Κρίσιμους Στόχους. Κατά την επιλογή αισθητήρων, ο αλγόριθμος παραγωγής ομάδων κάλυψης, μπορεί να επιλέξει από την ταξινομημένη λίστα αισθητήρων, τον αισθητήρα που καλύπτει τους λιγότερους δυνατούς Κρίσιμους Στόχους. Η τεχνική αυτή που χρησιμοποιήθηκε στην εργασία [137], προσδίδει ευελιξία στη διαδικασία επιλογής αισθητήρων, ειδικά στις περιπτώσεις όπου η βέλτιστη σύνθεση των ομάδων κάλυψης απαιτεί την επιλογή περισσότερων του ενός αισθητήρων από τα σύνολα γειτονικών αισθητήρων των Κρίσιμων Στόχων.

Ο αλγόριθμος παραγωγής ομάδων κάλυψης τερματίζει είτε όταν το πλήθος των παραγόμενων ομάδων είναι ίσο με το Θεωρητικό Μέγιστο είτε όταν δεν υπάρχουν πλέον διαθέσιμοι αισθητήρες για να καλύψουν τους υπό επιτήρηση στόχους. Η εξάντληση των αι-

σθητήρων δεν οφείλεται όμως μόνο στους Κρίσιμους Στόχους. Η χρήση μιας λανθασμένης πολιτικής επιλογής αισθητήρων μπορεί να οδηγήσει στην κάλυψη στόχων με πλεονάζοντα αριθμό αισθητήρων, μειώνοντας έτσι το πλήθος των δυνατών ομάδων κάλυψης. Για το λόγο αυτό, η πολιτική επιλογής αισθητήρων θα πρέπει να διασφαλίζει την καλύτερη δυνατή χρήση κάθε αισθητήρα, παράγοντας ομάδες με το μικρότερο δυνατό πλήθος αισθητήρων, αφήνοντας το μέγιστο δυνατό αριθμό αισθητήρων διαθέσιμους για χρήση στις επόμενες ομάδες κάλυψης.

4.3.4 Στρατηγικές διαμόρφωσης των ομάδων κάλυψης

Κατά τη δημιουργία μιας ομάδας κάλυψης, η διαδικασία επιλογής αισθητήρων επεξεργάζεται τέσσερις τύπους υποψήφιων αισθητήρων. Οι τύποι αυτοί διαφέρουν μεταξύ τους στο είδος της κάλυψης που προσφέρουν:

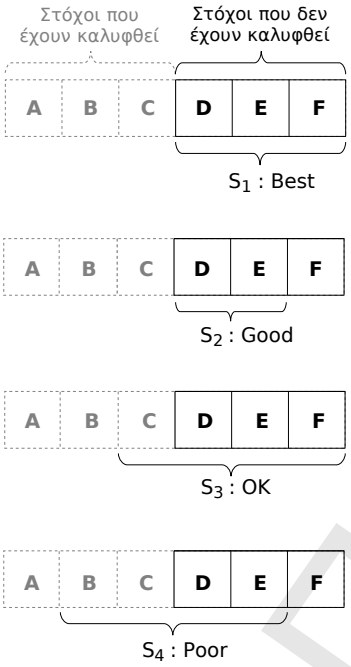
- στους στόχους που παραμένουν ακάλυπτοι, και
- στους στόχους που έχουν ήδη καλυφθεί από μέλη της "υπό κατασκευήν" ομάδας κάλυψης.

Στην εργασία [137] δίνονται τα εξής χαρακτηριστικά ονόματα στους τύπους υποψήφιων αισθητήρων:

- **τύπος Best**: Ο αισθητήρας καλύπτει όλους τους στόχους που παραμένουν ακάλυπτοι και κανέναν από τους ήδη καλυμμένους.
- **τύπος Good**: Ο αισθητήρας καλύπτει μέρος των ακάλυπτων στόχων και κανέναν εκ των ήδη καλυμμένων.
- **τύπος OK**: Ο αισθητήρας καλύπτει όλους τους ακάλυπτους στόχους και μέρος εκ των ήδη καλυμμένων.
- **τύπος Poor**: Ο αισθητήρας καλύπτεί μέρος των ακάλυπτων στόχων καθώς και μέρος (ή το σύνολο) των ήδη καλυμμένων.

Στο σχήμα 4.4 απεικονίζονται παραδείγματα των παραπάνω τύπων υποψήφιων αισθητήρων. Όπως γίνεται αντιληπτό, οι αισθητήρες τύπου *Best* προτιμώνται σε σχέση με τους αισθητήρες άλλων τύπων καθώς καλύπτουν επακριβώς το σύνολο των ακάλυπτων στόχων. Σε περιπτώσεις όμως όπου ο αλγόριθμος παραγωγής ομάδων κάλυψης δεν έχει στη διάθεση του αισθητήρες αυτού του τύπου θα πρέπει να επιλέξει κάποιον αισθητήρα άλλου τύπου. Αν ακολουθηθεί μια αυστηρή σειρά προτίμησης με βάση τον τύπο (π.χ. $Good \rightarrow OK \rightarrow Poor$) τότε τα αποτελέσματα του αλγορίθμου δε θα είναι σε καμμία περίπτωση βέλτιστα. Το πρόβλημα αυτό πηγάζει από το γεγονός ότι ο τύπος ενός αισθητήρα δεν είναι το μοναδικό κριτήριο σύμφωνα με το οποίο θα πρέπει να αξιολογείται η καταλληλότητα αυτού. Για παράδειγμα, ένας αισθητήρας που καλύπτει 5 ακάλυπτους στόχους και μόλις 1 ήδη καλυμμένο (τύπος OK) είναι συνήθως πιο χρήσιμος από έναν αισθητήρα που καλύπτει 1 ακάλυπτο στόχο και κανέναν ήδη καλυμμένο (τύπος $Good$).

Φαίνεται λοιπόν, ότι ο αριθμός καλυμμένων/ακάλυπτων στόχων που καλύπτει ο κάθε αισθητήρας, θα πρέπει να ληφθεί και αυτός υπόψιν (πέραν του τύπου του αισθητήρα) κατά τη διαδικασία αξιολόγησης των υποψήφιων αισθητήρων. Στην εργασία [137] οι αισθητήρες των τύπων $Good$, OK και $Poor$ θεωρούνται ισότιμοι και αξιολογούνται με βάση μια αντικειμενική συνάρτηση η οποία λαμβάνει υπόψιν μεταξύ άλλων παραμέτρων και το



Σχήμα 4.4: Οι τέσσερις τύποι υποψήφιων αισθητήρων: Best, Good, OK και Poor

πλήθος των ήδη καλυμμένων και ακάλυπτων στόχων που οι αισθητήρες καλύπτουν. Η αντικειμενική συνάρτηση προσδίδει στον αλγόριθμο μεγαλύτερη "ευελιξία" κατά την επιλογή αισθητήρων, επιτρέποντας του να μειώσει (όποτε είναι αυτό εφικτό) τον αριθμό των στόχων που καλύπτονται (άσκοπα) από περισσότερους του ενός αισθητήρες αλλά και τον αριθμό αισθητήρων ανά ομάδα κάλυψης.

4.4 Αλγόριθμοι πλήρους κάλυψης

Στην ενότητα αυτή θα παρουσιαστούν τρεις κεντρικοποιημένοι αλγόριθμοι για την παραγωγή ομάδων *πλήρους κάλυψης*, που βασίζονται στο μοντέλο που περιγράφηκε στην ενότητα 4.3.2.

Ο πρώτος αλγόριθμος, ονόματι B{GOP}, αποτελεί ένα γρήγορο ευρετικό αλγόριθμο για την παραγωγή ομάδων κάλυψης δίχως κοινά στοιχεία. Κατά τη διαδικασία επιλογής αισθητήρων για μια ομάδα κάλυψης, ο αλγόριθμος αυτός χρησιμοποιεί μια αντικειμενική συνάρτηση προκειμένου να ταξινομήσει τους υποψήφιους αισθητήρες. Ο αλγόριθμος B{GOP} παρουσιάστηκε για πρώτη φορά στην εργασία [137] και προσφέρει καλύτερα αποτελέσματα από αυτά του αντίστοιχου αλγορίθμου των Slijepcevic et al [129] και μάλιστα σε μικρότερο χρονικό διάστημα.

Στο δεύτερο αλγόριθμο, που ονομάζεται BGOP-random, κάθε αισθητήρας επιλέγεται σύμφωνα με κάποια πιθανότητα, στον υπολογισμό της οποίας συμβάλλει η αντικειμενική συνάρτηση που αναφέρθηκε παραπάνω. Στόχος του αλγορίθμου αυτού είναι η μελέτη της επίδρασης της τυχαίας επιλογής αισθητήρων στο πλήθος των παραγόμενων ομάδων κάλυψης.

Ο τρίτος αλγόριθμος που θα παρουσιαστεί, ονομάζεται CCF και αποτελεί μετεξέλιξη του B{GOP}. Ο αλγόριθμος αυτός χρησιμοποιεί μια πιο σύνθετη αντικειμενική συνάρτηση η οποία του επιτρέπει να παράγει καλύτερα αποτελέσματα από αυτά του B{GOP}. Επίσης,

έχει τη δυνατότητα να παράγει και ομάδες κάλυψης με κοινά στοιχεία. Από τα πειραματικά δεδομένα της εργασίας [138] προκύπτει ότι ο αλγόριθμος CCF υπερτερεί του αντίστοιχου αλγορίθμου των Cardei et al [134].

4.4.1 Ο αλγόριθμος B{GOP}

Ο ευρετικός αλγόριθμος B{GOP} έχει ως στόχο τη γρήγορη παραγωγή ομάδων κάλυψης δίχως κοινά στοιχεία.

Βασικά χαρακτηριστικά

Η ονομασία του αλγορίθμου B{GOP} προκύπτει από το γεγονός ότι ο αλγόριθμος αυτός επιλέγει κόμβους (αισθητήρες) σύμφωνα με μια στρατηγική η οποία λαμβάνει υπόψιν της τον τύπο (*Best*, *Good*, *OK*, *Poor*) των κόμβων. Συγκεκριμένα, ο αλγόριθμος B{GOP} προτιμά πάντα την επιλογή κόμβων τύπου *Best*. Σε περίπτωση απουσίας ενός τέτοιου κόμβου επιλέγει τον καλύτερο υποψήφιο από τους υπόλοιπους τύπους κόμβων, σύμφωνα με το αποτέλεσμα μιας διαδικασίας αξιολόγησης.

Σε αντίθεση με τον αλγόριθμο των Slijepcevic et al. [129], ο B{GOP} δε διαχωρίζει τους στόχους σε Κρίσιμους και Μη Κρίσιμους. Αντίθετα, η "κρισιμότητα" ενός στόχου αυξάνεται όσο ο αριθμός των αισθητήρων που τον καλύπτουν μειώνεται. Έτσι, ένας στόχος που καλύπτεται από x αισθητήρες είναι "πιο κρίσιμος" από κάποιον που καλύπτεται από $x+1$. Ο βαθμός αυτός "κρισιμότητας" επιτρέπει στο B{GOP} να ταξινομήσει τους αισθητήρες με τέτοιο τρόπο, ώστε να επιλέγονται με προτεραιότητα οι αισθητήρες που καλύπτουν στόχους με μικρή "κρισιμότητα". Αποφεύγεται έτσι, κατά μεγάλο βαθμό, η άσκοπη κάλυψη κρίσιμων στόχων και επιτυγχάνεται η παραγωγή περισσότερων ομάδων κάλυψης. Επίσης, σε αντίθεση με την στατική πολιτική επιλογής κόμβων που εφαρμόζεται στον αλγόριθμο των Slijepcevic et al., επιτρέπεται η επιλογή ενός αισθητήρα που καλύπτει στόχους μεγάλης "κρισιμότητας" όταν μια τέτοια επιλογή είναι η μόνη λύση για να συμπληρωθεί μια ομάδα κάλυψης.

Το χαρακτηριστικό *badness* B_j περιγράφει το συνολικό βαθμό "κρισιμότητας" των στόχων που καλύπτει ένας αισθητήρας s_j και δίνεται από τη σχέση:

$$B_j = \sum_{i=1}^{|P_j|} (\mu - |N_i| + 1)^3 \tag{4.1}$$

όπου N_i είναι το σύνολο των γειτονικών αισθητήρων του i-οστού μέλους του συνόλου P_j. Το σύνολο P_j ονομάζεται *σύνολο κάλυψης στόχων* του αισθητήρα s_j και περιέχει τους στόχους που καλύπτει ο αισθητήρας s_j, δηλ. $t_i \in P_j$ αν και μόνο αν $s_j \in N_i$. Το σύμβολο μ αποτελεί τον πληθάριθμο του μεγαλύτερου συνόλου γειτονικών αισθητήρων, ($\mu = \max(|N_1|, \ldots, |N_k|)$, $k = |T_0|$). Τέλος, η ποσότητα $(\mu - |N_i| + 1)$ προσδίδει ένα βαθμό "κρισιμότητας" ακόμη και στους στόχους που καλύπτονται από το μέγιστο αριθμό αισθητήρων, ενώ ο εκθέτης επιτρέπει στο *badness* να αποδώσει διαφορετικές τιμές στον αισθητήρα που καλύπτει λίγους κρίσιμους στόχους και στον αισθητήρα που καλύπτει πολλους μη κρίσιμους στόχους.

Όταν ο B{GOP} πρέπει να επιλέξει μεταξύ δύο αισθητήρων τύπου $Best$ (βλ. ενότητα 4.3.4), τότε επιλέγει αυτόν με τη μικρότερη τιμή *badness*. Όμως, όταν θα πρέπει να επιλέξει μεταξύ αισθητήρων άλλων τύπων τότε θα πρέπει οι αισθητήρες να αξιολογηθούν με βάση το αποτέλεσμα μιας *αντικειμενικής συνάρτησης*. Στη συνάρτηση αυτή εκτός από το χαρακτηριστικό *badness*, λαμβάνεται υπόψιν και το πλήθος των ήδη καλυμμένων αλλά και των

ακάλυπτων στόχων που ο υποψήφιος αισθητήρας καλύπτει, καθώς και μια ένδειξη ως προς το βαθμό ολοκλήρωσης του αλγορίθμου.

Η αντικειμενική συνάρτηση, ονόματι *benefit* έχει την εξής μορφή:

$$\begin{aligned} benefit(B_j, P_j, T_{cur}, T_0) &= \frac{uncovered}{(covered+1)^r} + (1 - normalised\_badness) \\ &= \frac{cov(P_j, T_{cur})}{(cov(P_j, T_0) - cov(P_j, T_{cur}) + 1)^r} + \left(1 - \frac{B_j}{max\_badness}\right), \end{aligned} \tag{4.2}$$

όπου οι παράμετροι B_j, P_j, T_{cur} και T_0 είναι η τιμή του χαρακτηριστικού *badness* του υπό εξέταση αισθητήρα, οι στόχοι που καλύπτει ο υπό εξέταση αισθητήρας, οι στόχοι που παραμένουν ακάλυπτοι από την υπό κατασκευή ομάδα κάλυψης και το σύνολο των στόχων του υπό παρατήρηση χώρου, αντίστοιχα.

Η αντικειμενική συνάρτηση χρησιμοποιεί τη βοηθητική συνάρτηση $cov(A, B) = |A \cap B|$, για να ορίσει το πλήθος των ακάλυπτων ($uncovered$) καθώς και των ήδη καλυμμένων ($covered$) στόχων που καλύπτονται από τον υπό εξέταση αισθητήρα. Συγκεκριμένα, οι στόχοι της αντικειμενικής συνάρτησης είναι: α) η ανάδειξη κόμβων που καλύπτουν μεγάλο πλήθος ακάλυπτων στόχων (ώστε κάθε ομάδα κάλυψης να περιέχει το μικρότερο δυνατό πλήθος αισθητήρων), β) η αποφυγή κόμβων που καλύπτουν ήδη καλυμμένους στόχους (ώστε να αποφεύγεται η «διπλή κάλυψη» στόχων και ιδίως «κρίσιμων στόχων») και γ) η ανάδειξη κόμβων που καλύπτουν όσο το δυνατόν λιγότερους «κρίσιμους στόχους».

Οι στόχοι (α) και (β) επιτυγχάνονται μέσω της διαίρεσης[1] $\frac{uncovered}{covered+1}$. Η δύναμη $r = \frac{|C|}{max\_sets}$ (με $r \in [0, 1]$, $|C|$ το πλήθος των ήδη σχηματισμένων ομάδων κάλυψης και max_sets το θεωρητικό μέγιστο πλήθος δυνατών ομάδων κάλυψης) που εφαρμόζεται στο διαιρέτη του παραπάνω κλάσματος, αυξάνει την «ποινή» των κόμβων που καλύπτουν ήδη καλυμμένους στόχους, όσο ο αλγόριθμος οδεύει προς την κάλυψη πιο «κρίσιμων στόχων». Εφόσον ο αλγόριθμος δίνει μεγαλύτερη προτεραιότητα στην επιλογή κόμβων που καλύπτουν στόχους μικρής «κρισιμότητας», οι κόμβοι που καλύπτουν πιο «κρίσιμους» στόχους εξετάζονται στα τελευταία στάδια εκτέλεσης του αλγορίθμου, όταν δηλαδή η ποσότητα r τείνει στη μονάδα.

Ο στόχος (γ) επιτυγχάνεται με τη χρήση του χαρακτηριστικού *badness* (B_j) στον ορισμό της συνάρτησης *benefit* (βλ. τύπο 4.2). Συγκεκριμένα, χρησιμοποιείται μια κανονικοποιημένη μορφή ($normalised\_badness$) της τιμής *badness* ώστε να προωθούνται κόμβοι που έχουν χαμηλότερες τιμές στο χαρακτηριστικό αυτό. Η μέθοδος κανονικοποίησης χρησιμοποιεί τη σταθερά $max\_badness$ η οποία αντιστοιχεί στη μέγιστη καταγεγραμμένη τιμή για το χαρακτηριστικό *badness*. Η τιμή της σταθεράς αυτής γίνεται γνωστή στον αλγόριθμο κατά το στάδιο αρχικοποίησης, όπως θα παρουσιαστεί στην επόμενη υποενότητα.

Στάδιο αρχικοποίησης - Setup

Ο αλγόριθμος χρησιμοποιεί την είσοδο I (βλ. παρ. 4.3.1) για να δημιουργήσει/υπολογίσει:

- Το αρχικό σύνολο στόχων T_0.
- Το αρχικό σύνολο αισθητήρων S_0.
- Το σύνολο γειτονικών αισθητήρων N_i κάθε στόχου $t_i \in T_0$.

[1] η πρόσθεση με τη μονάδα στο διαιρέτη επιτρέπει την αποφυγή διαιρέσεων με το μηδέν.

- Το σύνολο κάλυψης στόχων P_j κάθε αισθητήρα $s_j \in S_0$.
- Τον πληθάριθμο του μεγαλύτερου συνόλου γειτονικών αισθητήρων, $\mu = \max(|N_1|, \ldots, |N_k|)$, $k = |T_0|$.
- Το χαρακτηριστικό *badness* B_j κάθε αισθητήρα $s_j \in S_0$.
- Το μέγιστο *badness*: $max\_badness = \max(B_1, \ldots, B_n)$, $n = |S_0|$.
- Το *Θεωρητικό Μέγιστο* πλήθος δυνατών ομάδων κάλυψης (βλ. παρ. 4.3.3), $max\_sets = \min(|N_1|, \ldots, |N_k|),\ k = |T_0|$.

Περιγραφή αλγορίθμου

Ο B{GOP} είναι ένας «άπληστος» ευρετικός αλγόριθμος κάλυψης στόχων. Η έξοδος του αλγορίθμου B{GOP} είναι μια συλλογή C από ομάδες κάλυψης δίχως κοινά στοιχεία, οι οποίες μπορούν να καλύψουν ανεξάρτητα όλους τους υπό παρατήρηση στόχους. Η υλοποίηση του αλγορίθμου B{GOP} βασίζεται σε τρεις εμφωλευμένους βρόχους: το βρόχο "Ελέγχου διαθέσιμων αισθητήρων", το βρόχο "Ελέγχου ακάλυπτων στόχων" και το βρόχο "Ελέγχου καταλληλότητας αισθητήρα" (βλ. αλγόριθμο 1).

Ο βρόχος "Ελέγχου διαθέσιμων αισθητήρων" είναι υπεύθυνος για την πρόσθεση ομάδων κάλυψης στη συλλογή C και εκτελείται μόνο όταν υπάρχουν διαθέσιμοι αισθητήρες. Αρχικά, δημιουργεί μια νέα κενή ομάδα κάλυψης C_{cur} και αρχικοποιεί το σύνολο των ακάλυπτων στόχων T_{cur} (με τα περιεχόμενα του αρχικού συνόλου στόχων T_0) και το σύνολο των διαθέσιμων αισθητήρων S_{cur} (με τα περιεχόμενα του αρχικού συνόλου αισθητήρων S_0). Έπειτα, καλεί το βρόχο "Ελέγχου ακάλυπτων στόχων" για να συμπληρώσει τη νέα ομάδα κάλυψης C_{cur}. Μόλις αυτή είναι έτοιμη την προσθέτει στη συλλογή C. Ο αλγόριθμος τερματίζει είτε σε περίπτωση που το πλήθος των ομάδων που περιέχονται στο C γίνει ίσο με το Θεωρητικό Μέγιστο πλήθος δυνατών ομάδων κάλυψης (max_sets) είτε σε περίπτωση εξάντλησης των διαθέσιμων αισθητήρων. Και στις δύο περιπτώσεις, ο αλγόριθμος επιστρέφει στο χρήστη τα περιεχόμενα της συλλογής C, όπως αυτή έχει διαμορφωθεί μέχρι εκείνη τη στιγμή.

Ο βρόχος "Ελέγχου ακάλυπτων στόχων" είναι υπεύθυνος για τη συγκρότηση μιας ομάδας κάλυψης. Κάθε αισθητήρας της ομάδας αυτής επιλέγεται από ένα σύνολο υποψήφιων αισθητήρων που έχει διαμορφωθεί κατά το βρόχο "Ελέγχου καταλληλότητας αισθητήρα". Συγκεκριμένα, ο βρόχος "Ελέγχου καταλληλότητας αισθητήρα" εντοπίζει τον καλύτερο υποψήφιο της κλάσης *Best* (όταν αυτός υπάρχει) καθώς και τον καλύτερο υποψήφιο από τις υπόλοιπες κλάσεις (βλ. τύπους υποψήφιων αισθητήρων παρ. 4.3.4). Αν εντοπιστεί υποψήφιος κόμβος της κλάσης *Best* τότε αυτός εισάγεται άμεσα στην υπό κατασκευή ομάδα κάλυψης. Διαφορετικά, επιλέγεται ο καλύτερος υποψήφιος από τις υπόλοιπες κλάσεις. Ένας αισθητήρας που έχει επιλεγεί για μια ομάδα κάλυψης αφαιρείται από το σύνολο των διαθέσιμων αισθητήρων S_{cur} και προστίθεται στο σύνολο C_{cur}, που αντιπροσωπεύει την ομάδα κάλυψης. Επίσης, οι στόχοι που κάλυπτε ο αισθητήρας αυτός αφαιρούνται από το σύνολο ακάλυπτων στόχοι T_{cur}. Ο βρόχος "Ελέγχου ακάλυπτων στόχων" τερματίζει όταν δεν υπάρχουν πλέον ακάλυπτοι στόχοι στο σύνολο T_{cur}. Το γεγονός αυτό καθορίζει επίσης ότι η ομάδα κάλυψης C_{cur} είναι πλέον έτοιμη να γίνει μέρος της συλλογής C.

Ο βρόχος "Ελέγχου καταλληλότητας αισθητήρα" αναλαμβάνει το διαχωρισμό των υποψήφιων κόμβων σε τρεις κατηγορίες, βάσει των δυνατοτήτων κάλυψης που αυτοί έχουν. Στην πρώτη κατηγορία ανήκουν οι κόμβοι οι οποίοι δεν καλύπτουν κάποιον από τους ακάλυπτους κόμβους και μπορούν επομένως να παραβλεφτούν κατά τη σύνθεση της παρούσας ομάδας κάλυψης. Στη δεύτερη κατηγορία ανήκουν οι κόμβοι τύπου *Best*, ο καλύτερος εκ των οποίων παρακολουθείται από τη μεταβλητή $best$. Ως καλύτερος υποψήφιος τύπου

Αλγόριθμος 1 B{GOP}

Require: $S_0 \neq \emptyset$, $T_0 \neq \emptyset$, $P \neq \emptyset$, $B \neq \emptyset$, $max\_sets > 0$, $max\_badness > 0$

$C = \emptyset$
$S_{cur} = S_0$
while $S_{cur} \neq \emptyset$ **do** {Έλεγχος διαθέσιμων αισθητήρων}
  $C_{cur} = \emptyset$
  $T_{cur} = T_0$
  while $T_{cur} \neq \emptyset$ **do** {Έλεγχος ακάλυπτων στόχων}
    $best := none$
    $other := none$
    $selected := none$
    $min\_badness := max\_badness + 1$
    $max\_benefit := 0$
    for all $s \in S_{cur}$ **do** {Έλεγχος καταλληλότητας αισθητήρα}
      if $cov\,(P_s, T_{cur}) = 0$ **then**
        $S_{cur} = S_{cur} - \{s\}$ {παράβλεψη αισθητήρα s}
      else if $cov\,(P_s, T_{cur}) = cov\,(P_s, T_0)$ **then**
        if $B_s < min\_badness$ **then**
          $min\_badness := B_s$
          $best := s$
        end if
      else
        $in := cov\,(P_s, T_{cur})$
        $out := cov\,(P_s, T_0) - in$

        $$\alpha := \frac{|C|}{max\_sets}$$
        $$\beta := 1 - \frac{B_s}{max\_badness}$$
        $$benefit := \frac{in}{(out+1)^{\alpha}} + \beta$$

        if $benefit > max\_benefit$ **then**
          $max\_benefit := benefit$
          $other := s$
        end if
      end if
    end for {Έλεγχος καταλληλότητας αισθητήρα}
    $selected := best$ **or** $other$
    if $selected = none$ **then**
      return C
    end if
    $S_{cur} = S_{cur} - \{selected\}$
    $T_{cur} = T_{cur} - P_{selected}$
    $C_{cur} = C_{cur} \cup \{selected\}$
  end while {Έλεγχος ακάλυπτων στόχων}
  $C = C \cup \{C_{cur}\}$
  if $|C| = max\_sets$ **then**
    return C
  end if
end while {Έλεγχος διαθέσιμων αισθητήρων}
return C

Best θεωρείται ο κόμβος που συνοδεύεται από τη χαμηλότερη τιμή του χαρακτηριστικού *badness*. Τέλος, στην τρίτη κατηγορία ανήκουν οι κόμβοι τύπων *Good*, *OK* και *Poor*, ο καλύτερος εκ των οποίων παρακολουθείται από τη μεταβλητή $other$. Ως καλύτερος υποψήφιος αυτής της κατηγορίας θεωρείται αυτός που συγκεντρώνει την υψηλότερη βαθμολογία ως προς την αντικειμενική συνάρτηση *benefit*.

Ανάλυση αλγορίθμου

Κατά τη διαδικασία επιλογής του κατάλληλου αισθητήρα για μια ομάδα κάλυψης, ο αλγόριθμος B{GOP} εξετάζει τις δυνατότητες κάλυψης όλων των διαθέσιμων αισθητήρων, ως προς τους στόχους που παραμένουν ακάλυπτοι. Μόλις επιλεγεί ο κατάλληλος αισθητήρας και προστεθεί στην υπό κατασκευή ομάδα κάλυψης, θα εξεταστεί η κάλυψη που προσφέρουν οι εναπομείναντες αισθητήρες ως προς τους στόχους που ο επιλεγμένος αισθητήρας δεν κάλυψε. Με τον τρόπο αυτό προστίθενται αισθητήρες στην ομάδα κάλυψης έως ότου αυτή καλύψει όλους τους στόχους. Μόλις η ομάδα κάλυψης είναι έτοιμη, ξεκινά η παραγωγή μιας νέας ομάδας και τότε αρχικοποιείται το σύνολο των ακάλυπτων στόχων T_{cur} με τους στόχους που περιέχονται στο αρχικό σύνολο στόχων T_0.

Ο αλγόριθμος B{GOP} τερματίζει είτε όταν δεν υπάρχουν άλλοι διαθέσιμοι αισθητήρες ($S_{cur} = \emptyset$) για χρήση σε ομάδες κάλυψης, είτε όταν το πλήθος των παραγμένων ομάδων κάλυψης ισούται με το Θεωρητικό Μέγιστο δυνατό πλήθος ομάδων κάλυψης ($|C| = max\_sets$). Όμως, όπως αναφέρθηκε στην παράγραφο 4.3.3, πολλές φορές είναι αδύνατο να παραχθεί το Θεωρητικό Μέγιστο πλήθος ομάδων κάλυψης, καθώς ο αλγόριθμος τερματίζει πρόωρα λόγω απουσίας διαθέσιμων αισθητήρων. Συνεπώς, ο αριθμός των διαθέσιμων αισθητήρων θέτει ένα άνω όριο στη διάρκεια εκτέλεσης του αλγορίθμου. Συγκεκριμένα, η πιο χρονοβόρα εκτέλεση του αλγορίθμου B{GOP} θα αφορούσε ένα σενάριο όπου όλοι οι διαθέσιμοι αισθητήρες θα συμμετείχαν στις παραγόμενες ομάδες κάλυψης. Στην περίπτωση αυτή, ο χρόνος εκτέλεσης του αλγορίθμου θα ήταν ανάλογος του γινομένου:

$$\sum_{i=0}^{n-1}(n-i)(k-i \bmod k),$$

όπου n το πλήθος των αισθητήρων ($n = |S_0|$) και k το πλήθος των στόχων ($k = |T_0|$). Από την παραπάνω παράσταση προκύπτει ότι ο χρόνος εκτέλεσης του αλγορίθμου B{GOP} είναι της τάξης $O(n^2k)$.

Ο ευρετικός αλγόριθμος B{GOP} εκτός από μικρό χρόνο εκτέλεσης, κατέχει και μια ακόμη ενδιαφέρουσα ιδιότητα. Είναι αποδεδειγμένα ικανός να παραγάγει τουλάχιστον μία ομάδα κάλυψης, όταν υπάρχουν διαθέσιμοι οι αναγκαίοι αισθητήρες προκειμένου να συγκροτηθεί μια τέτοια ομάδα. Η απόδειξη της πρότασης αυτής δίνεται αναλυτικά στο παράρτημα Α΄.1. Η πρόταση αυτή ισχύει και για τους αλγορίθμους B{GOP}-random και CCF που αποτελούν μετεξελίξεις του αλγορίθμου B{GOP} και οι οποίοι θα παρουσιαστούν σε επόμενες ενότητες.

Βελτιστοποιήσεις

Κατά τη διαδικασία "Ελέγχου καταλληλότητας" ενός αισθητήρα (βλ. αλγόριθμο 1) η συνάρτηση $cov(P, T)$ χρησιμοποιείται για να εξεταστούν οι δυνατότητες κάλυψης του αισθητήρα ως προς τους ακάλυπτους καθώς και ως προς τους ήδη καλυμμένους στόχους. Η συνάρτηση αυτή υπολογίζει τον αριθμό των μελών που ανήκουν στην τομή των συνόλων P και T και έτσι απαιτεί τουλάχιστον $min(|P|, |T|)$ ελέγχους[2]. Όπως είναι φανερό, περιορίζοντας τις κλήσεις στη συνάρτηση $cov(P, T)$ θα μπορούσε να μειωθεί σημαντικά ο χρόνος εκτέλεσης του αλγορίθμου.

Για παράδειγμα, το αρχικό πλήθος στόχων που καλύπτει ένας κόμβος ($cov(P_s, T_0)$) μπορεί να υπολογισθεί μία φορά για κάθε κόμβο, στο στάδιο αρχικοποίησης του αλγορίθμου, καθώς η τιμή αυτή παραμένει σταθερή κατά την εκτέλεση του αλγορίθμου.

[2]αν χρησιμοποιηθεί κάποια μέθοδος βασισμένη σε πίνακα κατακερματισμού για τον έλεγχο της ύπαρξης ενός μέλους σε ένα σύνολο.

Επίσης, αντί να υπολογίζεται η τιμή $cov(P_s, T_{cur})$ κάθε φορά που ο αλγόριθμος χρειάζεται πρόσβαση στο πλήθος των ακάλυπτων στόχων που καλύπτει ένας κόμβος, μπορεί η τιμή αυτή να προϋπολογιστεί όποτε μεταβάλλονται τα περιεχόμενα του συνόλου ακάλυπτων στόχων T_{cur}. Δηλαδή:

- η ποσότητα $cov(P_s, T_{cur})$ θα γίνεται ίση με $cov(P_s, T_0)$, όποτε δημιουργείται μια νέα κενή ομάδα κάλυψης και αρχικοποιείται το σύνολο ακάλυπτων στόχων T_{cur} με τα δεδομένα του αρχικού συνόλου στόχων T_0.
- μόλις επιλεγεί ένας αισθητήρας για χρήση σε μια ομάδα κάλυψης, οι στόχοι που αυτός καλύπτει αφαιρούνται από το σύνολο ακάλυπτων στόχων T_{cur} και ανανεώνονται οι τιμές $cov(P_s, T_{cur})$ για όλους τους διαθέσιμους κόμβους που κάλυπταν αυτούς τους στόχους.

Οι βελτιστοποιήσεις αυτές επιταχύνουν σημαντικά την εκτέλεση του ευρετικού αλγορίθμου και για το λόγο αυτό εφαρμόστηκαν στο λογισμικό που χρησιμοποιήθηκε στο πλαίσιο της πειραματικής αξιολόγησης (βλ. παρ. 4.6). Τέλος, η φύση των βελτιστοποιήσεων αυτών είναι τόσο γενική που επιτρέπει την εφαρμογή τους και σε αλγορίθμους που βασίζονται στον B{GOP}. Δύο τέτοιοι αλγόριθμοι είναι οι B{GOP}-random και CCF, οι οποίοι θα παρουσιαστούν στις επόμενες ενότητες.

4.4.2 Ο αλγόριθμος B{GOP}-random

Ο αλγόριθμος B{GOP}–random αποτελεί μια παραλλαγή του αλγορίθμου B{GOP}, στην οποία κάθε αισθητήρας επιλέγεται τυχαία βάσει κάποιας πιθανότητας εκλογής. Ο αλγόριθμος αυτός αναπτύχθηκε στα πλαίσια μελέτης σχετικά με τις επιπτώσεις της τυχαιοποίησης (randomisation) στη διαδικασία σύνθεσης ομάδων κάλυψης.

Βασικά Χαρακτηριστικά

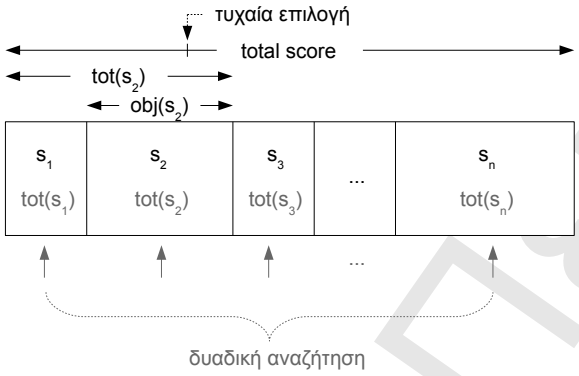
Στον αλγόριθμο B{GOP}–random, ένας κόμβος έχει πιθανότητα εκλογής ανάλογη με το αποτέλεσμα μιας αντικειμενικής συνάρτησης, που εφαρμόστηκε στα χαρακτηριστικά αυτού. Για την περίπτωση των κόμβων τύπου *Best* χρησιμοποιείται η παρακάτω αντικειμενική συνάρτηση:

$$obj_{Best}\left(B_s, max\_badness\right) = max\_badness - B_s + 1,$$

όπου $max\_badness$ είναι η μέγιστη καταγεγραμμένη τιμή $badness$, ενώ B_s είναι η τιμή του χαρακτηριστικού $badness$ του εξεταζόμενου κόμβου s. Με τον τρόπο αυτό δίνεται μεγαλύτερη πιθανότητα εκλογής σε κόμβους με μικρότερες τιμές στο χαρακτηριστικό *badness*. Αντίστοιχα, για τους κόμβους τύπου *Good*, *OK* και *Poor* χρησιμοποιείται ως αντικειμενική συνάρτηση η συνάρτηση *benefit* που παρουσιάστηκε στην προηγούμενη ενότητα, στο πλαίσιο του αλγορίθμου B{GOP}. Στο σημείο αυτό θα πρέπει να σημειωθεί ότι ο αλγόριθμος B{GOP}–random, τηρεί την απόδοση απόλυτης προτεραιότητας σε υποψήφιους κόμβους τύπου *Best*, που εισήγαγε ο αλγόριθμος B{GOP}.

Όπως αναφέρθηκε και παραπάνω, κατά τη διαδικασία επιλογής κόμβων, ο αλγόριθμος B{GOP}–random δίνει σε κάθε κόμβο πιθανότητα εκλογής ανάλογη της τιμής που έλαβε αυτός ο κόμβος από την αντικειμενική συνάρτηση (για χάρη συντομίας, έστω $obj(s)$). Προκειμένου να επιτευχθεί αυτή η πολιτική εκλογής, οι υποψήφιοι κόμβοι εισάγονται σε μια "νοητή λωρίδα" (βλ. σχήμα 4.5) όπου ο κάθε κόμβος καταλαμβάνει απόσταση ίση με

το αποτέλεσμα της αντικειμενικής συνάρτησης $obj(s)$. Το συνολικό μήκος της "λωρίδας" $total\_score$, είναι ίσο με το άθροισμα των τιμών που έχουν λάβει οι υποψήφιοι κόμβοι από την αντικειμενική συνάρτηση. Με τυχαίο τρόπο επιλέγεται ένας πραγματικός αριθμός $\tau \in (0, total\_score]$. Ο αριθμός τ αναπαριστά ένα σημείο στη νοητή "λωρίδα" που αντιστοιχεί στον προς επιλογή κόμβο. Με τον τρόπο αυτό η πιθανότητα επιλογής ενός συγκεκριμένου κόμβου ισούται με το μερίδιο αυτού του κόμβου στη νοητή "λωρίδα", δηλαδή με το αποτέλεσμα της αντικειμενικής συνάρτησης για τον κόμβο αυτό.



Σχήμα 4.5: Επιλογή τυχαίου κόμβου

Η ποσότητα $tot(s)$ αποτελεί ένα μερικό άθροισμα των τιμών της αντικειμενικής συνάρτησης, που αντιστοιχούν στον κόμβο s και στους κόμβους που προηγούνται αυτού στη "λωρίδα". Αν μαζί με κάθε κόμβο s στη νοητή "λωρίδα" αποθηκευθεί και η αντίστοιχη τιμή $tot(s)$, τότε ο κόμβος στον οποίο αντιστοιχεί το σημείο τ μπορεί να εντοπιστεί με τη μέθοδο της δυαδικής αναζήτησης, καθώς οι τιμές $tot(s)$ θα βρίσκονται αποθηκευμένες στη "λωρίδα" με αύξουσα σειρά.

Περιγραφή αλγορίθμου

Τόσο η διαδικασία αρχικοποίησης όσο και οι παράμετροι του αλγορίθμου B{GOP}–random είναι όμοιες με τις αντίστοιχες του αλγορίθμου B{GOP}.

Ο αλγόριθμος B{GOP}–random δεν ταξινομεί τους κόμβους σύμφωνα με το χαρακτηριστικό $badness$ ή το αποτέλεσμα της συνάρτησης $benefit()$. Αντίθετα, οι τιμές αυτές αποθηκεύονται, ώστε να αξιοποιηθούν αργότερα από τη συνάρτηση τυχαίας επιλογής $prob\_pick()$. Συγκεκριμένα, κατά το βρόχο "Ελέγχου ακάλυπτων στόχων" (βλ. αλγόριθμο 2) δημιουργούνται δύο λίστες, η S_{best} για κόμβους τύπου *Best* και η S_{other} για κόμβους τύπων *Good*, *OK* ή *Poor*. Κάθε φορά που εξετάζεται ένας υποψήφιος κόμβος στο βρόχο "Ελέγχου καταλληλότητας αισθητήρα", αποθηκεύεται στο τέλος της κατάλληλης λίστας μια δυάδα της μορφής $[s, tot(s)]$, η οποία περιγράφει τον κόμβο αυτό καθώς και το μερικό άθροισμα των τιμών της αντικειμενικής συνάρτησης για όλους τους κόμβους της λίστας μέχρι εκείνο το σημείο. Η πρόσθεση στοιχείων στη λίστα γίνεται μέσω τη συνάρτησης $push(A, b)$ η οποία προσθέτει το στοιχείο b στο τέλος της λίστας A και επιστρέφει τη νέα λίστα. Μόλις οι λίστες είναι έτοιμες (δηλ. έχουν εξεταστεί όλοι οι υποψήφιοι κόμβοι), δίδονται ως παράμετροι στη συνάρτηση $prob\_pick()$, σε συνδυασμό με το άθροισμα όλων των καταγεγραμμένων τιμών της αντικειμενικής συνάρτησης για την αντίστοιχη οικογένεια κόμβων (τιμές $total\_best\_badness$ για τους κόμβους τύπου *Best* και $total\_other\_benefit$ για τους κόμβους τύπου *Good*, *OK* και *Poor*).

Η συνάρτηση $prob\_pick()$ είναι υπεύθυνη για την τυχαία επιλογή ενός αισθητήρα. Η

Αλγόριθμος 2 B{GOP}–random

Require: $S_0 \neq \emptyset$, $T_0 \neq \emptyset$, $P \neq \emptyset$, $B \neq \emptyset$, $max\_sets > 0$, $max\_badness > 0$
  $C = \emptyset$
  $S_{cur} = S_0$
  while $S_{cur} \neq \emptyset$ **do** {Έλεγχος διαθέσιμων αισθητήρων}
    $C_{cur} = \emptyset$
    $T_{cur} = T_0$
    while $T_{cur} \neq \emptyset$ **do** {Έλεγχος ακάλυπτων στόχων}
      $selected := none$
      $total\_best\_badness := 0$
      $total\_other\_benefit := 0$
      $S_{best} = []$
      $S_{other} = []$
      for all $s \in S_{cur}$ **do** {Έλεγχος καταλληλότητας αισθητήρα}
        if $cov\,(P_s, T_{cur}) = 0$ **then**
          $S_{cur} = S_{cur} - \{s\}$ {παράβλεψη αισθητήρα s}
        else if $cov\,(P_s, T_{cur}) = cov\,(P_s, T_0)$ **then**
          $badness := max\_badness - B_s + 1$
          $total\_best\_badness := total\_best\_badness + badness$
          $S_{best} = push\left(S_{best}, [s, total\_best\_badness]\right)$
        else
          $in := cov\,(P_s, T_{cur})$
          $out := cov\,(P_s, T_0) - in$

          $$\alpha := \frac{|C|}{max\_sets}$$

          $$\beta := 1 - \frac{B_s}{max\_badness}$$

          $$benefit := \frac{in}{(out+1)^{\alpha}} + \beta$$

          $total\_other\_benefit := total\_other\_benefit + benefit$
          $S_{other} = push\left(S_{other}, [s, total\_other\_benefit]\right)$
        end if
      end for {Έλεγχος καταλληλότητας αισθητήρα}
      if $S_{best} \neq []$ **then**
        $selected = prob\_pick(S_{best}, total\_best\_badness)$
      else if $S_{other} \neq []$ **then**
        $selected = prob\_pick(S_{other}, total\_other\_benefit)$
      end if
      if $selected = none$ **then**
        return C
      end if
      $S_{cur} = S_{cur} - \{selected\}$
      $T_{cur} = T_{cur} - P_{selected}$
      $C_{cur} = C_{cur} \cup \{selected\}$
    end while {Έλεγχος ακάλυπτων στόχων}
    $C = C \cup \{C_{cur}\}$
    if $|C| = max\_sets$ **then**
      return C
    end if
  end while {Έλεγχος διαθέσιμων αισθητήρων}
  return C

συνάρτηση αυτή δέχεται ως παραμέτρους τη λίστα από δυάδες (S) που περιγράφει τους υποψήφιους αισθητήρες, καθώς και το άθροισμα των τιμών της αντικειμενικής συνάρτησης ($total\_score$) για τους αισθητήρες αυτούς. Όπως φαίνεται στον αλγόριθμο 3, η συνάρτηση αρχικά παράγει έναν τυχαίο πραγματικό αριθμό $score \in (0, total\_score]$ χρησιμοποιώντας τη συνάρτηση παραγωγής ψευδοτυχαίων αριθμών $rand()$. Έπειτα, χρησιμοποιεί δυαδική

Αλγόριθμος 3 Συνάρτηση prob_pick

Require: $S \neq \emptyset$, $total\_score > 0$

$low := 0$
$high := |S|$

$score := rand(total\_score)$

while $low < high$ **do** {Δυαδική Αναζήτηση}

$mid := int\left(low + \frac{high - low}{2}\right)$

if $score > S[mid][score]$ **then**
$low := mid + 1$
else
$high := mid$
end if

end while {Δυαδική Αναζήτηση}

return $S[low][sensor]$

αναζήτηση[3] για να εντοπίσει τον κόμβο στον οποίο ανήκει το σημείο με απόσταση $score$ από την αρχή της νοητής "λωρίδας".

Ανάλυση αλγορίθμου

Η διαφορά στην πολυπλοκότητα των αλγορίθμων B{GOP}-random και B{GOP} οφείλεται στη δυαδική αναζήτηση που ο αλγόριθμος B{GOP}-random εισάγει στο βρόχο "Ελέγχου ακάλυπτων στόχων".

Ο χρόνος εκτέλεσης του αλγορίθμου B{GOP}-random είναι ανάλογος του αθροίσματος:

$$\sum_{i=0}^{n-1} \left((n-i)(k - i \bmod k) + \log_2(n-i)\right),$$

και συνεπώς η πολυπλοκότητα του αλγορίθμου ανάγεται σε $O(n^2k + n\log_2 n)$, όπου n το πλήθος των διαθέσιμων αισθητήρων και k το πλήθος των στόχων.

4.4.3 Ο αλγόριθμος CCF

Ο αλγόριθμος CCF επιτρέπει τη δημιουργία ομάδων πλήρους κάλυψης που μπορούν να περιέχουν και κοινούς αισθητήρες. Η δομή του αλγορίθμου βασίζεται σε αυτή του ευρετικού αλγορίθμου B{GOP}, μόνο που εδώ χρησιμοποιείται μια κοινή αντικειμενική συνάρτηση για την αξιολόγηση όλων των υποψήφιων κόμβων, ανεξαρτήτως τύπου.

Βασικά χαρακτηριστικά

Το βασικότερο χαρακτηριστικό του αλγορίθμου CCF είναι η δυνατότητα παραγωγής ομάδων πλήρους κάλυψης, χρησιμοποιώντας κοινούς αισθητήρες (non-disjoint sets). Όπως διαπιστώθηκε στην ενότητα 4.1 η τακτική αυτή μπορεί να επεκτείνει το συνολικό χρόνο κάλυψης που προσφέρει ένα δίκτυο αισθητήρων.

[3]Στον αλγόριθμο 3 το σύμβολο $S[mid][score]$ αναφέρεται στην τιμή της αντικειμενικής συνάρτησης που αντιστοιχεί στο mid-οστό στοιχείο της λίστας S. Αντίστοιχα, το σύμβολο $S[low][sensor]$ αναφέρεται στο όνομα του κόμβου που βρίσκεται στη low-οστή θέση της λίστας S.

Κάθε αισθητήρας που εξετάζεται από τον αλγόριθμο CCF μπορεί να συμμετάσχει σε w, το πολύ, ομάδες κάλυψης (βλ. *αριθμός εμφανίσεων* w, παρ. 4.3.1). Όταν $w = 1$ τότε ο αλγόριθμος παράγει ομάδες δίχως κοινούς αισθητήρες, αντίθετα όταν $w > 1$, ο αλγόριθμος παράγει ομάδες με κοινούς αισθητήρες. Εφόσον ο κάθε αισθητήρας έχει περιορισμένη διάρκεια ζωής, έστω h ώρες, η μία από τις w βάρδιες, θα διαρκέσει $\frac{h}{w}$ ώρες.

Κατά τη διάρκεια εκτέλεσης του αλγορίθμου, το *δυναμικό συμμετοχής* L_s περιγράφει το πλήθος των περαιτέρω ομάδων κάλυψης στις οποίες επιτρέπεται να συμμετάσχει ένας κόμβος s. Μόλις το δυναμικό συμμετοχής ενός κόμβου γίνει μηδέν, ο κόμβος αυτός δε θα μπορέσει να χρησιμοποιηθεί ξανά σε κάποια ομάδα κάλυψης. Προκειμένου να υπάρχει μια σαφής διάκριση μεταξύ των κόμβων που έχουν αποθέματα ενέργειας και αυτών που τα έχουν εξαντλήσει, εισάγεται το σύνολο S_{avail}, το οποίο περιέχει τους κόμβους s με $L_s > 0$.

Το δεύτερο βασικό χαρακτηριστικό του αλγορίθμου CCF είναι η χρήση μιας κοινής αντικειμενικής συνάρτησης για την ταξινόμηση των υποψήφιων κόμβων, ανεξαρτήτως τύπου. Η χρήση της κοινής αντικειμενικής συνάρτησης έχει ως αποτέλεσμα την κατάργηση των ειδικών ελέγχων σχετικά με τον τύπο των κόμβων και συνεπώς την επιτάχυνση της διαδικασίας επιλογής αισθητήρων. Οι στόχοι της αντικειμενικής συνάρτησης είναι οι παρακάτω:

1. Η ανάδειξη κόμβων που καλύπτουν μεγάλο πλήθος ακάλυπτων στόχων.
2. Η αποφυγή κόμβων που καλύπτουν ήδη καλυμμένους στόχους.
3. Η ανάδειξη κόμβων που καλύπτουν όσο το δυνατόν λιγότερους «κρίσιμους στόχους» (από το στόχο αυτό προκύπτει και η ονομασία του αλγορίθμου CCF – *Critical Control Factor*).
4. Η ανάδειξη κόμβων που έχουν διαθέσιμα υψηλά αποθέματα ενέργειας.

Στην εργασία [138] προτείνεται η παρακάτω μορφή για την αντικειμενική συνάρτηση του αλγορίθμου CCF:

$$\begin{aligned} obj_{CCF}\left(T_{cur}, P_s, B_s, L_s\right) = \alpha \cdot & \frac{uncovered}{(covered+1)^r \cdot |T_{cur}|} \\ & + \beta \cdot \left(1 - \frac{B_s}{max\_badness}\right) \\ & + \gamma \cdot \frac{L_s}{w} \end{aligned} \tag{4.3}$$

όπου $uncovered = cov(P_s, T_{cur})$, $covered = cov(P_s, T_0) - cov(P_s, T_{cur})$ και $r = 1 - \frac{|T_{cur}|}{T_0}$. Για τις σταθερές α, β και γ, ισχύει:

$$\alpha, \beta, \gamma \in (0,1) \text{ και } \alpha + \beta + \gamma = 1 \tag{4.4}$$

Ο τρόπος με τον οποίο προκύπτουν οι τιμές αυτών των σταθερών θα αναλυθεί σε επόμενη παράγραφο.

Εφόσον η αντικειμενική συνάρτηση καλείται μόνο σε περιπτώσεις κόμβων που καλύπτουν τουλάχιστον ένα ακάλυπτο στόχο, ισχύει:

$$0 < \frac{uncovered}{(covered+1)^r \cdot |T_{cur}|} \leq 1 \tag{4.5}$$

Επίσης, εφόσον η αντικειμενική συνάρτηση χρησιμοποιείται για τον έλεγχο κόμβων που έχουν τα απαραίτητα ενεργειακά αποθέματα για να συμμετάσχουν σε μία τουλάχιστον ομάδα κάλυψης, ισχύει:

$$0 < \frac{L_s}{w} \leq 1 \tag{4.6}$$

Τέλος, ισχύει:

$$0 \leq \left(1 - \frac{B_s}{max\_badness}\right) < 1 \tag{4.7}$$

Από τις 4.3, 4.4, 4.5, 4.6 και 4.7 προκύπτει ότι:

$$obj_{CCF}\left(T_{cur}, P_s, B_s, L_s\right) \rightarrow (0, 1) \tag{4.8}$$

Στάδιο αρχικοποίησης – Setup

Όμοια με τον αλγόριθμο B{GOP}, κατά το στάδιο αρχικοποίησης, ο αλγόριθμος CCF χρησιμοποιεί την είσοδο I για να παραγάγει τα σύνολα T_0, S_0, N, P, το χαρακτηριστικό B_s και τις σταθερές $\mu, max\_badness$ και max_sets. Θα πρέπει να σημειωθεί ότι εφόσον κάθε αισθητήρας μπορεί να συμμετάσχει σε (το πολύ) w ομάδες κάλυψης, η τιμή της σταθεράς max_sets διαμορφώνεται ως εξής:

$$max\_sets = w \cdot min\left(|N_1|, \ldots, |N_k|\right), k = |T_0|.$$

Οι τιμές των σταθερών w ($w > 0$) και α, β, γ δίνονται από το χρήστη ως παράμετροι στον αλγόριθμο.

Περιγραφή αλγορίθμου

Ο αλγόριθμος CCF (βλ. αλγόριθμο 4) αρχικά καταγράφει το δυναμικό συμμετοχής L_s για κάθε ένα από τους διαθέσιμους κόμβους s. Εάν όλοι οι κόμβοι έχουν τα ίδια αποθέματα ενέργειας και συνεπώς μπορούν να συμμετάσχουν σε w ομάδες κάλυψης, τότε:

$$L_s = w, \ \forall s \in S_0.$$

Αν κάτι τέτοιο όμως δεν ισχύει, τότε οι τιμές L_s θα πρέπει να προσαρμοστούν κατάλληλα από το χρήστη του αλγορίθμου, ώστε να ανταποκρίνονται στις πραγματικές δυνατότητες κάλυψης των διαθέσιμων αισθητήρων.

Κατά τη διάρκεια εκτέλεσης του αλγορίθμου το σύνολο S_{avail} περιλαμβάνει τους αισθητήρες που έχουν τα απαραίτητα αποθέματα ενέργειας προκειμένου να συμμετάσχουν σε μία τουλάχιστον ομάδα κάλυψης. Αρχικά, το σύνολο αυτό ισούται με το αρχικό σύνολο αισθητήρων, δηλαδή: $S_{avail} = S_0$.

Σε κάθε επανάληψη του βρόχου "Ελέγχου διαθέσιμων αισθητήρων" δημιουργείται μια νέα ομάδα κάλυψης C_{cur} η οποία προστίθεται στη συλλογή C. Ο βρόχος εκτελείται μόνο εφόσον υπάρχουν διαθέσιμοι αισθητήρες στο σύνολο S_{avail} (δηλ. αισθητήρες με τα απαιτούμενα αποθέματα ενέργειας). Σε κάθε επανάληψη του βρόχου, το σύνολο S_{cur} με τους διαθέσιμους αισθητήρες για τη νέα ομάδα, αρχικοποιείται με τα περιεχόμενα του συνόλου S_{avail}. Έτσι, στη νέα ομάδα μπορούν να συμμετάσχουν μόνο κόμβοι που έχουν τα απαραίτητα αποθέματα ενέργειας. Όπως συμβαίνει και στον αλγόριθμο B{GOP}, ο βρόχος "Ελέγχου διαθέσιμων αισθητήρων" καλεί το βρόχο "Ελέγχου ακάλυπτων στόχων" προκειμένου να προστεθούν κόμβοι στη νέα ομάδα κάλυψης.

Ο βρόχος "Ελέγχου ακάλυπτων στόχων" προσθέτει τον κόμβο που έλαβε τη μεγαλύτερη βαθμολογία από την αντικειμενική συνάρτηση obj_{CCF} (στα πλαίσια του βρόχου "Ελέγχου

Αλγόριθμος 4 CCF

Require: $S_0 \neq \emptyset$, $T_0 \neq \emptyset$, $P \neq \emptyset$, $max\_sets > 0$, $B \neq \emptyset$, $max\_badness > 0$, $w > 0$, $\alpha, \beta, \gamma \in (0, 1)$

```
C = ∅
for all s ∈ S_0 do
    L_s := w
end for
S_avail = S_0
while S_avail ≠ ∅ do {Έλεγχος διαθέσιμων αισθητήρων}
    S_cur = S_avail
    T_cur = T_0
    C_cur = ∅
    while T_cur ≠ ∅ do {Έλεγχος ακάλυπτων στόχων}
        selected := none
        max_obj_CCF := 0
        for all s ∈ S_cur do {Έλεγχος καταλληλότητας αισθητήρα}
            if cov(P_s, T_cur) ≠ 0 then
                uncovered := cov(P_s, T_cur)
                covered := cov(P_s, T_0) − uncovered

                r := 1 − |T_cur| / |T_0|

                obj_CCF := α · uncovered / ((covered + 1)^r · |T_cur|) + β · (1 − B_s / max_badness) + γ · L_s / w

                if obj_CCF > max_obj_CCF then
                    max_obj_CCF := obj_CCF
                    selected := s
                end if
            end if
        end for {Έλεγχος καταλληλότητας αισθητήρα}
        if selected = none then
            return C
        end if
        T_cur = T_cur − P_selected
        S_cur = S_cur − {selected}
        L_selected := L_selected − 1
        if L_selected = 0 then
            S_avail = S_avail − {selected}
        end if
        C_cur = C_cur ∪ {selected}
    end while {Έλεγχος ακάλυπτων στόχων}
    C = C ∪ {C_cur}
    if |C| = max_sets then
        return C
    end if
end while {Έλεγχος διαθέσιμων αισθητήρων}
return C
```

καταλληλότητας αισθητήρα") στη νέα ομάδα κάλυψης. Επίσης, ενημερώνει το δυναμικό συμμετοχής του κόμβου αυτού ($L_{selected}$) ώστε σε περίπτωση που δεν μπορεί να χρησιμοποιηθεί ο κόμβος αυτός σε άλλη ομάδα κάλυψης, να αφαιρεθεί από το σύνολο S_{avail}.

Τέλος, ο βρόχος "Ελέγχου καταλληλότητας αισθητήρα" είναι υπεύθυνος για την εφαρμογή της αντικειμενικής συνάρτησης σε όλους τους υποψήφιους κόμβους που καλύπτουν τουλάχιστον έναν από τους ακάλυπτους στόχους. Όποτε προκύπτει μια βαθμολογία μεγαλύτερη από την υψηλότερη τρέχουσα, τότε ενημερώνεται η υψηλότερη τρέχουσα και ο αντίστοιχος κόμβος σημαδεύεται ως προτεινόμενος ($selected = s$).

Ανάλυση αλγορίθμου

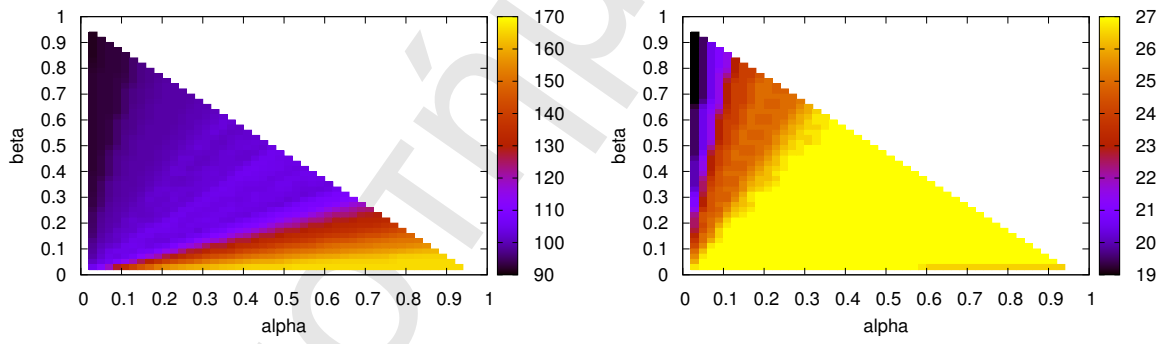
Όπως διαπιστώθηκε και για τον αλγόριθμο B{GOP}, η μεγαλύτερη (σε χρονική διάρκεια) εκτέλεση του αλγορίθμου CCF, περιλαμβάνει την εισαγωγή όλων των διαθέσιμων αισθητήρων σε ομάδες κάλυψης. Μάλιστα, στην περίπτωση όπου κάθε αισθητήρας μπορεί να χρησιμοποιηθεί w φορές, ο συνολικός χρόνος εκτέλεσης του αλγορίθμου γίνεται ανάλογος του γινομένου:

$$w \sum_{i=0}^{n-1} (n-i)(k - i \bmod k),$$

όπου $n = |S_0|$ και $k = |T_0|$. Συνεπώς, ο συνολικός χρόνος εκτέλεσης του αλγορίθμου CCF είναι της τάξης $O(wn^2k)$.

Εύρεση βέλτιστων τιμών για τις σταθερές α, β και γ της αντικειμενικής συνάρτησης

Οι σταθερές α, β και γ λαμβάνουν τιμές οι οποίες επιλέγονται ανάλογα με τη φύση του προβλήματος κάλυψης που εξετάζεται. Για παράδειγμα, ενισχύοντας την τιμή της σταθεράς α στην αντικειμενική συνάρτηση 4.3, δίνεται μεγαλύτερη έμφαση στις δυνατότητες κάλυψης ενός κόμβου και έτσι παράγονται ομάδες κάλυψης με λιγότερους κόμβους-μέλη. Αντίστοιχα, ενισχύοντας την τιμή β, ο αλγόριθμος αποφεύγει όλο και περισσότερο τους κόμβους που καλύπτουν "κρίσιμους στόχους". Τέλος, η ενίσχυση της τιμής της σταθεράς γ έχει ως αποτέλεσμα την απόδοση προτεραιότητας σε υποψήφιους κόμβους με μεγαλύτερα αποθέματα ενέργειας.



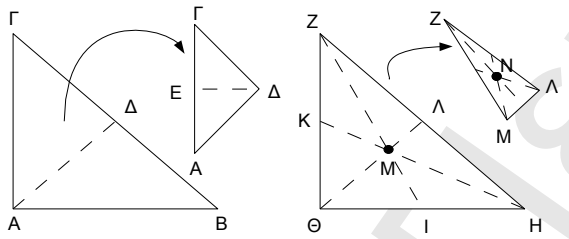
(α′) (β′)

Σχήμα 4.6: Η σχέση μεταξύ των σταθερών α, β και του πλήθους των παραγόμενων ομάδων κάλυψης (χρωματισμός άξονα z) σε πυκνά δομημένες (4.6α′) και αραιά δομημένες (4.6β′) τοπολογίες αισθητήρων.

Στην παράγραφο αυτή θα μελετηθεί η σχέση μεταξύ των σταθερών α, β, γ και του πλήθους των παραγόμενων ομάδων κάλυψης $|C|$. Επίσης, με τη χρήση μεθόδων βελτιστοποίησης θα αναζητηθούν οι βέλτιστες τιμές των σταθερών, οι οποίες επιτρέπουν την παραγωγή του μέγιστου δυνατού πλήθους ομάδων κάλυψης. Εφόσον $\gamma = 1 - \alpha - \beta$, αρκεί να βρεθούν βέλτιστες τιμές για τις δύο από τις τρεις σταθερές.

Στο σχήμα 4.6 παρουσιάζεται το πλήθος των παραγόμενων ομάδων κάλυψης (χρωματισμός άξονα z) για συγκεκριμένες τιμές των σταθερών α και β. Συγκεκριμένα, στο σχήμα

4.6α΄ απεικονίζονται τα αποτελέσματα για μια "πυκνή" τοπολογία, με 350 αισθητήρες και 40 στόχους σε μια επιφάνεια $361m^2$. Αντίστοιχα, στο σχήμα 4.6β΄ απεικονίζονται τα αποτελέσματα για μια "αραιή" τοπολογία, με τον ίδιο αριθμό αισθητήρων και στόχων αλλά αυτή τη φορά διεσπαρμένων σε επιφάνεια $961m^2$. Παρατηρεί κανείς ότι και στις δύο περιπτώσεις οι βέλτιστες τιμές βρίσκονται εντός μιας τριγωνικής "ζώνης". Επίσης, όσο απομακρύνεται κανείς από τη "ζώνη" με τις βέλτιστες τιμές, φθίνει και η ποιότητα του αποτελέσματος. Αυτό σημαίνει ότι μια αναζήτηση για το τοπικό μέγιστο (από οποιοδήποτε σημείο του χώρου λύσεων) μπορεί με μεγάλη πιθανότητα να οδηγήσει στο καθολικό μέγιστο. Αυτό το χαρακτηριστικό του χώρου λύσεων εντοπίστηκε στη συντριπτική πλειοψηφία των σεναρίων κάλυψης που εξετάστηκαν.



Σχήμα 4.7: Υποδιαίρεση τριγώνου και εύρεση έγκεντρου

Επειδή οι σταθερές α και β είναι πραγματικοί αριθμοί, μπορούν να λάβουν άπειρους συνδυασμούς τιμών. Προκειμένου να περιοριστεί το πλήθος των υπό εξέταση σημείων του χώρου λύσεων, ο τριγωνικός χώρος λύσεων μπορεί να διαιρεθεί σε μικρότερα τρίγωνα, από τα οποία θα εξεταστούν τα έγκεντρα αυτών[4]. Το έγκεντρο ενός τριγώνου είναι το σημείο στο οποίο τέμνονται οι διχοτόμοι των γωνιών αυτού και βρίσκεται πάντοτε εντός του τριγώνου (βλ. σημείο Μ του τριγώνου ΖΗΘ στο σχήμα 4.7).

Η βέλτιστη τιμή για τις σταθερές α και β, μπορεί να εντοπιστεί με μια μέθοδο τύπου "Διαίρει και Βασίλευε" (Divide and Conquer), όπου ο αρχικός χώρος λύσεων διαιρείται σε μικρότερους χώρους, και οι οποίοι στη συνέχεια εξετάζονται (με τη χρήση μιας αντικειμενικής συνάρτησης) ως προς την ικανότητα τους να δώσουν καλές λύσεις. Κάθε τρίγωνο που προκύπτει από την αρχική υποδιαίρεση, διαιρείται και αυτό με τη σειρά του ώστε να εντοπιστεί με μεγαλύτερη ακρίβεια η βέλτιστη λύση. Ενδιαφέρον παρουσιάζουν δύο μέθοδοι υποδιαίρεσης των τριγώνων του χώρου λύσεων: α) η διχοτόμηση ως προς μια τυχαία κορυφή και β) η υποδιαίρεση ως προς τις διχοτόμους. Όπως φαίνεται στο σχήμα 4.7, η υποδιαίρεση (α) δημιουργεί δύο τρίγωνα προς εξέταση (ΑΒΔ, ΑΓΔ), ενώ η υποδιαίρεση (β) δημιουργεί 6 τρίγωνα (ΖΙΘ, ΖΙΗ, ΘΖΛ, ΘΛΗ, ΗΚΖ, ΗΚΘ – 2 ανά διχοτομημένη γωνία του τριγώνου).

Χρησιμοποιώντας την τεχνική εξέτασης έγκεντρων και τη μέθοδο "Διαίρει και Βασίλευε" μπορεί κανείς με αλγοριθμικό τρόπο να εντοπίσει τις βέλτιστες τιμές για τις σταθερές α και β. Συγκεκριμένα, στα πλαίσια της διατριβής αυτής αναπτύχθηκε ένας αλγόριθμος τύπου *Best First Search*, ο οποίος υποδιαιρεί τον τριγωνικό χώρο λύσεων σε μικρότερα τρίγωνα και εξετάζει με προτεραιότητα τα τρίγωνα των οποίων τα έγκεντρα συγκεντρώνουν την καλύτερη βαθμολογία ως προς μια αντικειμενική συνάρτηση. Ο αλγόριθμος αυτός ονομάστηκε *Best-First Search* και χρησιμοποιεί μια ουρά προτεραιότητας για να ταξινομήσει (σε φθίνουσα σειρά) τα εξεταζόμενα τρίγωνα ως προς τη βαθμολόγησή τους. Η αντικειμενική συνάρτηση f που εφαρμόζεται στο έγκεντρο κάθε τριγώνου, δεν είναι άλλη από την απλή

[4]ως αντιπροσωπευτικές τιμές των περιοχών που καταλαμβάνουν τα τρίγωνα που προέκυψαν από την υποδιαίρεση.

απαρίθμηση των ομάδων κάλυψης που παράγονται από τον αλγόριθμο CCF, όταν αυτός εκτελεστεί με τις τιμές α και β που αντιστοιχούν στις συντεταγμένες του έγκεντρου.

Αλγόριθμος 5 Αναζήτηση Best-First

Require: $tr_0, f, queue\_size, divs, max\_depth, max\_checks$

$best\_tr := None$
$best\_val := 0$
$checks := 0$
$Q = [(tr_0, f(centroid(tr_0)), 0)]$ {Ουρά προτεραιότητας}

while $empty(Q) = False$ **do** {Βρόχος εξέτασης αντικειμένων της ουράς}
  $e := dequeue(Q)$
  $tr := get\_triangle(e)$
  $depth := get\_depth(e)$
  $val := get\_prio(e)$
  if $val > best\_val$ **then**
    $best\_val := val$
    $best\_tr := tr$
  end if
  $checks := checks + 1$
  if $checks = max\_checks$ **then**
    break
  end if
  if $depth = max\_depth$ **or** $full(Q, divs, queue\_size) = True$ **then**
    continue
  end if
  $T = split(tr, divs)$
  $depth := depth + 1$
  for all $tri \in T$ **do** {Βρόχος ελέγχου τριγώνου}
    $val := f(centroid(tri))$
    $Q = enqueue(Q, (tri, depth, val))$
  end for {Βρόχος ελέγχου τριγώνου}
end while {Βρόχος εξέτασης αντικειμένων της ουράς}
return $(centroid(best\_tr), best\_val)$

Ο αλγόριθμος 5 παρουσιάζει τα βασικά βήματα της αναζήτησης *Best-First Search*. Ο χρήστης δίνει ως παραμέτρους το αρχικό τρίγωνο λύσεων tr_0 ([1,0], [0,0], [0,1]), την αντικειμενική συνάρτηση f, το μέγιστο μέγεθος της ουράς προτεραιότητας $queue\_size$, το πλήθος $divs$ των τριγώνων που θα προκύπτουν σε κάθε υποδιαίρεση (2 ή 6), το μέγιστο πλήθος υποδιαιρέσεων max_depth καθώς και το μέγιστο πλήθος των στοιχείων της ουράς που θα εξεταστούν, max_checks. Τα στοιχεία που εισάγονται στην ουρά προτεραιότητας έχουν τη μορφή $(tr, f(centroid(tr)), depth)$ όπου tr είναι οι συντεταγμένες ενός τριγώνου, $f(centroid(tr))$ είναι το αποτέλεσμα της αντικειμενικής συνάρτησης f για τις τιμές α, β που προκύπτουν από το έγκεντρο του τριγώνου tr ($centroid(tr)$) και $depth$ είναι το επίπεδο υποδιαίρεσης στο οποίο ανήκει το τρίγωνο tr. Όταν διαιρείται ένα τρίγωνο, ο *βαθμός υποδιαίρεσης* των τριγώνων που προκύπτουν είναι κατά μία μονάδα αυξημένος από τον αντίστοιχο βαθμό υποδιαίρεσης του αρχικού τριγώνου. Χρησιμοποιώντας τον καταγεγραμμένο βαθμό υποδιαίρεσης και το ανώτατο όριο max_depth, ο αλγόριθμος μπορεί να περιορίσει το "βάθος" του δέντρου των λύσεων που θα εξετάσει.

Αρχικά, ο αλγόριθμος εισάγει στην ουρά προτεραιότητας τα στοιχεία που αντιστοιχούν στο συνολικό τρίγωνο λύσεων tr_0. Ο *Βρόχος εξέτασης αντικειμένων της ουράς* αναλαμβάνει σε κάθε επανάληψή του, να αφαιρέσει ένα στοιχείο από την ουρά, να το εξετάσει, να καταγράψει αν πρόκειται για το καλύτερο στοιχείο μέχρι στιγμής (τρίγωνο $best\_tr$, τιμή αντ. συνάρτησης $best\_val$) και, αν οι συνθήκες το επιτρέπουν, να το υποδιαιρέσει και να προσθέσει τα παραγόμενα τρίγωνα στην ουρά προτεραιότητας για περαιτέρω εξέταση. Οι

συνθήκες που επηρεάζουν την υποδιαίρεση ενός τριγώνου είναι τρεις:

- Αν έχει ήδη εξεταστεί max_checks πλήθος στοιχείων της ουράς, τότε ο αλγόριθμος τερματίζει πρόωρα επιστρέφοντας τις συντεταγμένες (δηλ. τις τιμες α και β) του έγκεντρου του τριγώνου που έχει συγκεντρώσει μέχρι εκείνη τη στιγμή τη μεγαλύτερη βαθμολογία, καθώς και τη βαθμολογία αυτή (δηλ. το πλήθος των παραγόμενων ομάδων κάλυψης).

- Αν η υποδιαίρεση του τριγώνου πρόκειται να αυξήσει το βαθμό υποδιαίρεσης των παραγόμενων τριγώνων πέραν της τιμής max_depth, τότε η υποδιαίρεση ακυρώνεται και εξετάζεται το επόμενο στοιχείο στην ουρά προτεραιότητας.

- Αν η ουρά προτεραιότητας είναι ήδη γεμάτη (δηλ. περιέχει πλήθος στοιχείων μεγαλύτερο του $queue\_size - divs$) τότε η υποδιαίρεση του τριγώνου ακυρώνεται και εξετάζεται το επόμενο στοιχείο της ουράς.

Σε περίπτωση που οι συνθήκες επιτρέπουν την υποδιαίρεση ενός τριγώνου, τότε αυτή πραγματοποιείται μέσω της συνάρτησης $split(tr, divs)$ η οποία δέχεται ως παραμέτρους το εν λόγω τρίγωνο tr και το πλήθος των τριγώνων $divs$ που θα παραχθούν. Η συνάρτηση $split()$ επιστρέφει το σύνολο με τα τρίγωνα που προέκυψαν από την υποδιαίρεση. Κάθε ένα από τα τρίγωνα αυτά θα βαθμολογηθεί μέσω της αντικειμενικής συνάρτησης και θα τοποθετηθεί στην κατάλληλη θέση της ουράς προτεραιότητας σύμφωνα με την βαθμολογία που έλαβε.

Όπως μπορεί κανείς εύκολα να διαπιστώσει, ο αλγόριθμος αυτός θα συνέχιζε την εκτέλεσή του επ άπειρον, αν ο χρήστης δεν έθετε περιορισμούς στο πλήθος των στοιχείων της ουράς που μπορούν να εξεταστούν (έλεγχος max_checks) καθώς και στο μέγιστο δυνατό "βάθος" του δέντρου λύσεων (έλεγχος max_depth).

Επειδή είναι περιορισμένος ο αριθμός των στοιχείων που εξετάζονται σε κάθε εκτέλεση του αλγορίθμου, δεν είναι δυνατή η εξέταση όλων των κόμβων του δέντρου λύσεων. Με τη ρύθμιση πάντως, της παραμέτρου max_depth ο χρήστης μπορεί να επηρεάσει το βάθος του δέντρου λύσεων και να πετύχει την εύρεση μιας ικανοποιητικής λύσης, με μικρό αριθμό επαναλήψεων (βλ. Βρόχο εξέτασης αντικειμένων της ουράς). Το πρόβλημα όμως που παραμένει είναι η ύπαρξη δύο διαφορετικών παραμέτρων, τις οποίες θα πρέπει να ρυθμίσει ο χρήστης προκειμένου ο αλγόριθμος να βρει σε μικρό χρονικό διάστημα τη βέλτιστη (ή μια ικανοποιητική) λύση.

Παρατηρώντας την εκτέλεση του αλγορίθμου *Best-First Search* σε διάφορα σενάρια δικτύων αισθητήρων, διαπιστώθηκε ότι τις περισσότερες φορές η ταχύτερη εκτέλεση του αλγορίθμου (η οποία συγχρόνως παρήγαγε τις βέλτιστες τιμές για τις σταθερές α και β) αφορούσε στην εξέταση ενός και μόνο μονοπατιού του δέντρου λύσεων, το οποίο μονοπάτι μπορούσε να προκύψει από τοπικές "αποφάσεις" σε κάθε κόμβο του δέντρου. Λαμβάνοντας υπόψιν αυτή την πειραματική διαπίστωση, δημιουργήθηκε ένας ευρετικός αλγόριθμος για την ταχύτερη βελτιστοποίηση των τιμών των σταθερών α και β. Ο ευρετικός αυτός αλγόριθμος ονομάζεται *Split-Triangle* και ο χρόνος εκτέλεσής του περιορίζεται από το μέγιστο "βάθος" του δέντρου λύσεων που επιθυμεί να εξετάσει ο χρήστης.

Ο αλγόριθμος *Split-Triangle* δέχεται ως παραμέτρους τον αρχικό τριγωνικό χώρο λύσεων (tr_0), την αντικειμενική συνάρτηση (f), το μέγιστο βάθος ($steps$) του δένδρου λύσεων που θα εξεταστεί, καθώς και το πλήθος των τριγώνων ($divs$) που θα προκύπτουν σε κάθε υποδιαίρεση.

Ο αλγόριθμος *Split-Triangle* αποτελείται, όπως και ο *Best-First Search*, από δύο εμφωλευμένους βρόχους (βλ. αλγόριθμο 6). Ο εξωτερικός βρόχος (*Βρόχος υποδιαιρέσεων*) αναλαμ-

Αλγόριθμος 6 Αναζήτηση Split-Triangle

Require: $tr_0, f, steps, divs$

```
best_tr := tr_0
best_val := f(centroid(tr_0))
tr := tr_0

for i = 1 to steps do {Βρόχος υποδιαιρέσεων}
   T = split(tr, divs)
   max_val := 0
   for all tri ∈ T do {Βρόχος ελέγχου τριγώνου}
      val := f(centroid(tri))
      if val ≥ max_val then
         max_val := val
         tr := tri
         if max_val > best_val then
            best_val := max_val
            best_tr := tri
         end if
      end if
   end for {Βρόχος ελέγχου τριγώνου}
end for {Βρόχος υποδιαιρέσεων}
return (centroid(best_tr), best_val)
```

βάνει να διαιρέσει το τρίγωνο που εξετάζεται εκείνη τη στιγμή (tr) σε υπο-τρίγωνα. Αυτή η διαίρεση θα συμβεί $steps$ φορές.

Ο εσωτερικός βρόχος (*Βρόχος ελέγχου τριγώνου*) θα εξετάσει σειριακά τα τρίγωνα αυτά και το τρίγωνο του οποίου το έγκεντρο θα λάβει την υψηλότερη βαθμολόγηση από την αντικειμενική συνάρτηση, θα γίνει το επόμενο τρίγωνο προς εξέταση (tr). Αν κάποιο έγκεντρο λάβει βαθμολογία υψηλότερη από οποιοδήποτε άλλο έχει εξεταστεί μέχρι εκείνη τη στιγμή, η τοποθεσία του καταγράφεται μαζί με την βαθμολογία του, ώστε να επιστραφούν κατά τον τερματισμό του αλγορίθμου.

Η βασική διαφορά μεταξύ των δύο αλγορίθμων έγγειται στη μέθοδο διαχείρισης των τριγώνων που προκύπτουν από μια υποδιαίρεση. Ενώ και στους δύο αλγορίθμους βαθμολογούνται τα τρίγωνα αυτά με βάση την αντικειμενική συνάρτηση, στον αλγόριθμο *Best-First Search* τα τρίγωνα αυτά γίνονται μέρος της ουράς προτεραιότητας ανεξαρτήτως της βαθμολογίας που έλαβαν. Αυτο σημαίνει ότι κάθε ένα από αυτά θα προκαλέσει μια επιπλέον επανάληψη[5] του εξωτερικού βρόχου και επίσης θα προκαλέσει μια καθυστέρηση, σχετική με την τοποθέτηση αυτού στο κατάλληλο σημείο της ουράς προτεραιότητας. Στην περίπτωση του αλγορίθμου *Split-Triangle*, μετά τη βαθμολόγηση των τριγώνων, επιλέγεται ένα από αυτά για περαιτέρω υποδιαίρεση μειώνοντας έτσι σημαντικά το χρόνο εκτέλεσης του αλγορίθμου. Όμως, εφόσον δεν εξετάζονται όλα τα τρίγωνα που παρήχθηκαν από την υποδιαίρεση, υπάρχει πάντοτε ο κίνδυνος να μη βρεθεί η βέλτιστη λύση στο υπό εξέταση πρόβλημα.

Στον πίνακα 4.1 παρουσιάζονται συνοπτικά τα αποτελέσματα μιας σειράς πειραμάτων, στα οποία οι αλγόριθμοι *Best-First Search* και *Split-Triangle* χρησιμοποιήθηκαν για τη βελτιστοποίηση των τιμών των σταθερών α, β και γ. Χρησιμοποιώντας τις βελτιστοποιημένες τιμές, ο αλγόριθμος CCF κατάφερε σε όλες τις περιπτώσεις να παραγάγει το μέγιστο δυνατό πλήθος ομάδων κάλυψης. Οι γραμμές του πίνακα παρουσιάζουν διαφορετικά σενάρια κάλυψης. Σε κάθε σενάριο 350 αισθητήρες προσφέρουν πλήρη κάλυψη σε 40 στόχους, όμως

[5] Μια επανάληψη του εξωτερικού βρόχου μπορεί να οδηγήσει στην υποδιαίρεση ενός τριγώνου και, συνεπώς, σε πολλαπλές χρήσεις της χρονοβόρας συνάρτησης f, αν οι συνθήκες το επιτρέπουν.

	Αναζήτηση Best-First			Αναζήτηση Split-Triangle
Μέγεθος Περιοχής (m^2)	**Επαναλήψεις Βρόχου**	**Πλήθος Υποδιαιρέσεων**	**Χρόνος Εκτέλ. (sec)**	**Χρόνος Εκτέλ. (sec)**
361	3	2	10.71	5.76
484	2	1	4.02	2.20
625	3	2	6.62	3.65
778.4	2	1	2.74	1.52
961	2	1	2.35	1.34
1156	2	1	2.12	1.18
1369	2	1	2.27	1.26
1600	5	4	8.72	4.88
1764	3	2	2.68	1.53
1989	2	1	1.88	1.05

Πίνακας 4.1: Σύγκριση μεθόδων βελτιστοποίησης κατά την παραγωγή του μέγιστου δυνατού πλήθους ομάδων κάλυψης σε σενάρια με 350 αισθητήρες, 40 στόχους και μεταβλητού μεγέθους πεδίο κάλυψης.

το συνολικό πεδίο κάλυψης (terrain) μεταβάλλεται ανά περίπτωση (βλ. 1^{η} στήλη του πίνακα 4.1).

Κατά την εκτέλεση των αλγορίθμων βελτιστοποίησης, επιλέχθηκε η υποδιαίρεση των τριγωνικών περιοχών σε 6 τρίγωνα. Η δεύτερη και τρίτη στήλη του πίνακα 4.1 περιγράφουν το πλήθος των επαναλήψεων του εξωτερικού βρόχου και το πλήθος των υποδιαιρέσεων που απαιτήθηκαν προκειμένου ο αλγόριθμος *Best-First Search* να εντοπίσει τις βέλτιστες λύσεις στο συντομότερο δυνατό χρονικό διάστημα. Το πλήθος των απαιτούμενων υποδιαιρέσεων των τριγώνων είναι κοινό με αυτό του αλγορίθμου *Split-Triangle*, όμως στο δεύτερο αλγόριθμο ο αριθμός των εξωτερικών επαναλήψεων ισούται με το πλήθος των υποδιαιρέσεων. Όπως φαίνεται από την τέταρτη και πέμπτη στήλη του πίνακα, ο χρόνος εκτέλεσης του αλγορίθμου *Split-Triangle* βρέθηκε σε όλες τις περιπτώσεις μικρότερος εκείνου του αλγορίθμου *Best-First Search*.

4.5 Βοηθητικό λογισμικό

Προκειμένου να εξεταστούν οι παραπάνω αλγόριθμοι αναπτύχθηκε σχετικό λογισμικό[6] σε γλώσσα προγραμματισμού Perl[7]. Το λογισμικό αυτό περιλαμβάνει:

- εφαρμογές παραγωγής και απεικόνισης ψευδοτυχαίων τοπολογιών αισθητήρων και στόχων,
- υλοποιήσεις των αλγορίθμων κάλυψης που εξετάστηκαν στην ενότητα 4.4
- εφαρμογή αυτόματης εκτέλεσης αλγορίθμων κάλυψης, και συλλογής στατιστικών και ελέγχου των παραγόμενων ομάδων κάλυψης ως προς την ορθότητα τους.

Στην ενότητα αυτή θα περιγραφούν η δομή και οι λειτουργίες του λογισμικού αυτού.

Αξίζει να σημειωθεί ότι η παραπάνω πλατφόρμα προσομοίωσης χρησιμοποιήθηκε και σε εργασίες οι οποίες αφορούσαν θέματα πέραν του προβλήματος της πλήρους κάλυψης, όπως η εργασία [139] η οποία εξετάζει το πρόβλημα μερικής κάλυψης σε δίκτυα όπου απαιτείται σταθερή επικοινωνία με τη βάση.

[6] Το λογισμικό αυτό διατίθεται ελεύθερα υπό την άδεια GPLv3 από την ηλ. διεύθυνση: `http://rainbow.cs.unipi.gr/projects/sensors`

[7] `http://www.perl.org`

4.5.1 Παραγωγή τοπολογιών

Προκειμένου να αξιολογηθεί πειραματικά ένας αλγόριθμος κάλυψης απαιτείται η εφαρμογή αυτού σε πλήθος διαφορετικών τοπολογιών δικτύων αισθητήρων καθώς και σε τοπολογίες με διαφορετικά χαρακτηριστικά (π.χ. πυκνή / αραιή δόμηση).

Στο πλαίσιο της διατριβής αυτής αναπτύχθηκε λογισμικό αυτόματης παραγωγής τοπολογιών 2 διαστάσεων, 3 διαστάσεων αλλά και τοπολογιών που δεν περιορίζονται από συγκεκριμένο αριθμό διαστάσεων. Οι τοπολογίες 2 διαστάσεων είναι χρήσιμες για την εξέταση σεναρίων όπου δεν απαιτείται ακριβής μοντελοποίηση του πεδίου στο οποίο θα τοποθετηθούν οι αισθητήρες, ενώ οι τοπολογίες 3 διαστάσεων είναι προφανώς πιο κατάλληλες στην αντίθετη περίπτωση. Στις τοπολογίες 2 και 3 διαστάσεων, ο χρήστης ορίζει τον αριθμό των αισθητήρων που θα μετέχουν στην παραγόμενη τοπολογία, τον αριθμό των στόχων και το μέγεθος του χώρου που θα καταληφθεί από το δίκτυο αισθητήρων. Από τις παραμέτρους αυτές η εφαρμογή υπολογίζει την πυκνότητα δόμησης του επιθυμητού δικτύου αισθητήρων.

Το λογισμικό δημιουργεί εικονικούς αισθητήρες με ακτίνα επικοινωνίας $R_c = 50m$ και ακτίνα παρακολούθησης $R_s = 2m$. Η τοποθέτηση τους στο χώρο γίνεται σύμφωνα με τα παρακάτω βήματα:

1. Αρχικά, ο επιθυμητός αριθμός αισθητήρων και στόχων τοποθετείται τυχαία στην επιφάνεια που έχει ορίσει ο χρήστης, ακολουθώντας ομοιόμορφη κατανομή (uniform distribution) στο χώρο. Η *βάση* τοποθετείται πάντα στο σημείο $[0, \frac{y}{2}]$ σε χώρους 2 διαστάσεων και στο σημείο $[0, \frac{y}{2}, \frac{z}{2}]$ σε χώρους 3 διαστάσεων.

2. Οι στόχοι που δεν καλύπτονται από κανένα αισθητήρα αφαιρούνται από το πεδίο.

3. Οι αισθητήρες που δεν καλύπτουν κανένα στόχο αφαιρούνται από το πεδίο.

4. Κάθε στόχος ορίζει μια κυκλική *περιοχή* (ή σφαιρική στην περίπτωση των 3-διάστατων πεδίων) με κέντρο τον ίδιο το στόχο. Ένας αισθητήρας *ανήκει* σε μια τέτοια περιοχή αν το κέντρο αυτής βρίσκεται σε απόσταση από αυτόν μικρότερη του R_s. Κάθε αισθητήρας μπορεί να ανήκει σε περισσότερες από μία τέτοιες περιοχές. Προκειμένου δύο οποιοιδήποτε αισθητήρες (που ανήκουν σε διαφορετικές ενδεχομένως περιοχές) να μπορούν να ανταλλάξουν δεδομένα άμεσα (δίχως τη μεσολάβηση τρίτου αισθητήρα), θα πρέπει να μετέχουν σε τουλάχιστον ένα ζεύγος περιοχών, των οποίων τα κέντρα θα έχουν ευκλείδεια απόσταση μικρότερη από $(R_c - 2 \cdot R_s)$. Στόχοι και αισθητήρες *απομονωμένων περιοχών*, των οποίων οι αισθητήρες δε μπορούν να επικοινωνήσουν άμεσα με αισθητήρες άλλης περιοχής (ή τη *βάση*), αφαιρούνται από το πεδίο.

5. Δημιουργείται ένας γράφος με τις υπόλοιπες περιοχές. Οι κόμβοι του γράφου είναι οι περιοχές καθαυτές ενώ οι ακμές του γράφου συνδέουν τις περιοχές που μπορούν να επικοινωνήσουν άμεσα μεταξύ τους. Η *βάση* προστίθεται και αυτή ως κόμβος στο γράφο, με τις αντίστοιχες ακμές προς τις περιοχές με τις οποίες έχει άμεση επικοινωνία. Υπολογίζοντας τη *μεταβατική κλειστότητα* (transitive closure) του γράφου, εντοπίζονται περιοχές οι οποίες δεν έχουν κάποιο μονοπάτι επικοινωνίας με τη *βάση*. Οι στόχοι και οι αισθητήρες των περιοχών αυτών αφαιρούνται από το πεδίο.

6. Η παραγόμενη τοπολογία απαρτίζεται από τους εναπομείναντες αισθητήρες και στόχους.

Εξαιτίας των βημάτων 2 και 3 η τοπολογία που παράγεται είναι συμβατή με την είσοδο ενός αλγορίθμου κάλυψης όπως αυτός περιγράφηκε στην ενότητα 4.3.1. Δηλαδή, κάθε

στόχος θα πρέπει να καλύπτεται από τουλάχιστον ένα αισθητήρα και κάθε αισθητήρας θα πρέπει να καλύπτει τουλάχιστον ένα στόχο. Ο έλεγχος επικοινωνίας των βημάτων 4 και 5 εγγυάται ότι θα υπάρχει στην παραγόμενη τοπολογία μονοπάτι επικοινωνίας μεταξύ οποιουδήποτε αισθητήρα και της *βάσης*. **Μεταφέροντας τον έλεγχο αυτό στο στάδιο παραγωγής της τοπολογίας, δίνεται η δυνατότητα, σε οποιονδήποτε αλγόριθμο πλήρους κάλυψης, να δημιουργήσει ομάδες κάλυψης, οι οποίες θα έχουν εγγυημένα μονοπάτια επικοινωνίας με τη *βάση***. Επίσης, το βήμα 1 μπορεί να αντικατασταθεί με την είσοδο δεδομένων από μια πραγματική τοπολογία δικτύου αισθητήρων. Στην περίπτωση αυτή, η παραπάνω διαδικασία θα παίξει το ρόλο ενός "φίλτρου", δημιουργώντας ένα υποσύνολο του αρχικού δικτύου, το οποίο όμως θα έχει εγγυημένη επικοινωνία με τη *βάση*.

Στο σχήμα 4.8 παρουσιάζεται ένα παράδειγμα εκτέλεσης της εφαρμογής παραγωγής διδιάστατων τοπολογιών (`create_nodes_2d`). Ο χρήστης ζητά την παραγωγή μιας τοπολογίας με 500 αισθητήρες και 100 στόχους, διεσπαρμένους στο 20% μιας επιφάνειας $1km^2$. Η εφαρμογή θα δημιουργήσει την επιθυμητή τοπολογία και τα στοιχεία αυτής θα αποθηκευθούν στο αρχείο `topology.txt`. Κατά τη διάρκεια παραγωγής της τοπολογίας, η εφαρμογή τυπώνει στην καθιερωμένη έξοδο λαθών (standard error) βοηθητικές πληροφορίες σχετικά με την έκβαση της διαδικασίας.

```
glynos@host:~/bgop$ ./create_nodes_2d.pl
usage: ./create_nodes_2d.pl <num_of_nodes> <num_of_points> <area_of_interest%>
glynos@host:~/bgop$ ./create_nodes_2d.pl 500 100 20 > topology.txt
                  Generating Points: [..........] Done!
                 Generating Sensors: [..........] Done!
              Sensors in area check: [..........] Done!
            Deleting Solitary Areas: [.......] Done!
    Transitive Closure for 13 areas (this might take a while)
   Deleting base-unreachable areas: [......] Done!
glynos@host:~/bgop$
```

Σχήμα 4.8: Παραγωγή τοπολογίας 2 διαστάσεων

Στο σχήμα 4.10 παρουσιάζονται τα περιεχόμενα του αρχείου `topology.txt` που παράχθηκε από την παραπάνω εφαρμογή. Το αρχείο αυτό περιέχει πληροφορίες που θα αποτελέσουν την είσοδο I για έναν αλγόριθμο παραγωγής ομάδων κάλυψης. Θα πρέπει να σημειωθεί εδώ, ότι ο αριθμός των αισθητήρων και των στόχων που περιλαμβάνονται στο αρχείο μπορεί να μην είναι ο ίδιος με αυτούς που δόθηκαν ως παράμετροι στην εφαρμογή παραγωγής τοπολογιών. Αυτό συμβαίνει επειδή η εφαρμογή παραγωγής τοπολογιών "φιλτράρει" τα δεδομένα (με τον τρόπο που παρουσιάστηκε παραπάνω) ώστε η τελική τοπολογία να εξασφαλίζει σε όλους τους αισθητήρες τη δυνατότητα επικοινωνίας με τη βάση. Στην συγκεκριμένη περίπτωση, το αρχείο περιγράφει ένα διδιάστατο πεδίο $0.2km^2$ με 12 αισθητήρες και 10 στόχους.

Το πρώτο μέρος του αρχείου χρησιμοποιεί την παρακάτω σύνταξη για να περιγράψει τους αισθητήρες που θα μπορούσαν να προσφέρουν κάλυψη σε κάθε στόχο:

```
<Σύμβολο Στόχου #1> <Αριθμός Αισθητήρα #1> <Αριθμός Αισθητήρα #2> ...
<Σύμβολο Στόχου #2> ...
...
```

Στο δεύτερο μέρος του αρχείου, περιέχονται εγγραφές-σχόλια[8] που περιγράφουν τον τρόπο με τον οποίο παράχθηκε το αρχείο καθώς και βοηθητικές πληροφορίες και στατιστικά σχετικά με την τοπολογία που αντιπροσωπεύει. Συγκεκριμένα, αναφέρονται οι διαστάσεις

[8]Κάθε γραμμή που περιέχει εγγραφές-σχόλια ξεκινά με το χαρακτήρα #

του διδιάστατου πεδίου (σε dm), οι συντεταγμένες των αισθητήρων και των στόχων, οι συντεταγμένες της βάσης, η εφαρμογή που παρείγαγε το αρχείο (καθώς και οι παράμετροι αυτής), το πλήθος των κόμβων και αισθητήρων, το ελάχιστο πλήθος αισθητήρων που μπορούν να καλύψουν κάποιο στόχο, το μέγιστο / ελάχιστο / μέσο πλήθος στόχων που καλύπτουν οι αισθητήρες, το μέγεθος της επιφάνειας του διδιάστατου πεδίου, το μέγεθος κάθε αισθητήρα, η ακτίνα παρακολούθησης αυτού (R_s) καθώς και η ακτίνα επικοινωνίας του (R_c).

```
A 342
B 275
C 30
D 75
E 380
F 498
G 328
H 20 148 202
I 238
J 410
# terrain map [2000 x 2000]
# sensor coords: 238 [1742 1856] 328 [1561 1179] 75 [169 1133] 342 [449 1265]
                 202 [1436 1916] 20 [1454 1925] 498 [908 1327] 30 [1225 1552]
                 275 [1563 1278] 148 [1443 1914] 380 [998 1766] 410 [1705 870]
# target coords: A [455 1276] B [1568 1262] C [1242 1558] D [173 1129]
                 E [992 1770] F [908 1339] G [1568 1193] H [1436 1921]
                 I [1748 1873] J [1707 882]
# base station coords: [1 1000]
# generated with: ./create_nodes_2d.pl 500 100 20
# stats: sensors=12 points=10 min_cardinality=1 min_node_occur=1
         max_node_occur=1 mean_node_occur=1.0 terrain=40000.0m^2
         sensor_sz=0.01m^2 sensor_reading_radius=2.00m sensor_comm_radius=50.00m
```

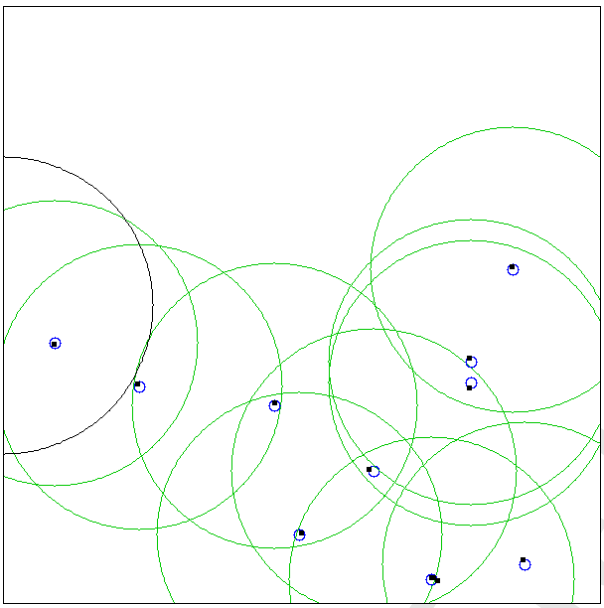
Σχήμα 4.9: Αρχείο περιγραφής τοπολογίας 2 διαστάσεων

Οι εφαρμογές `draw_terrain_2d` και `draw_terrain_3d` χρησιμοποιούν την παραπάνω βοηθητική πληροφορία για να αναπαραστήσουν με γραφικό τρόπο διδιάστατες και τρισδιάστατες (αντίστοιχα) τοπολογίες. Στο σχήμα 4.10 φαίνεται η γραφική απεικόνιση της διδιάστατης τοπολογίας του αρχείου `topology.txt`. Οι μικροί κύκλοι αντιπροσωπεύουν τις *κυκλικές περιοχές* γύρω από τους στόχους (βλ. διαδικασία παραγωγής τοπολογιών – βήμα 4), ενώ τα μικρά τετράγωνα σχήματα αντιπροσωπεύουν τους αισθητήρες. Οι μεγαλύτεροι κύκλοι περιγράφουν την εμβέλεια επικοινωνίας κάθε κόμβου, ενώ με έντονο μαύρο ημικύκλιο, σημειώνεται στα αριστερά του πεδίου, η εμβέλεια επικοινωνίας της βάσης.

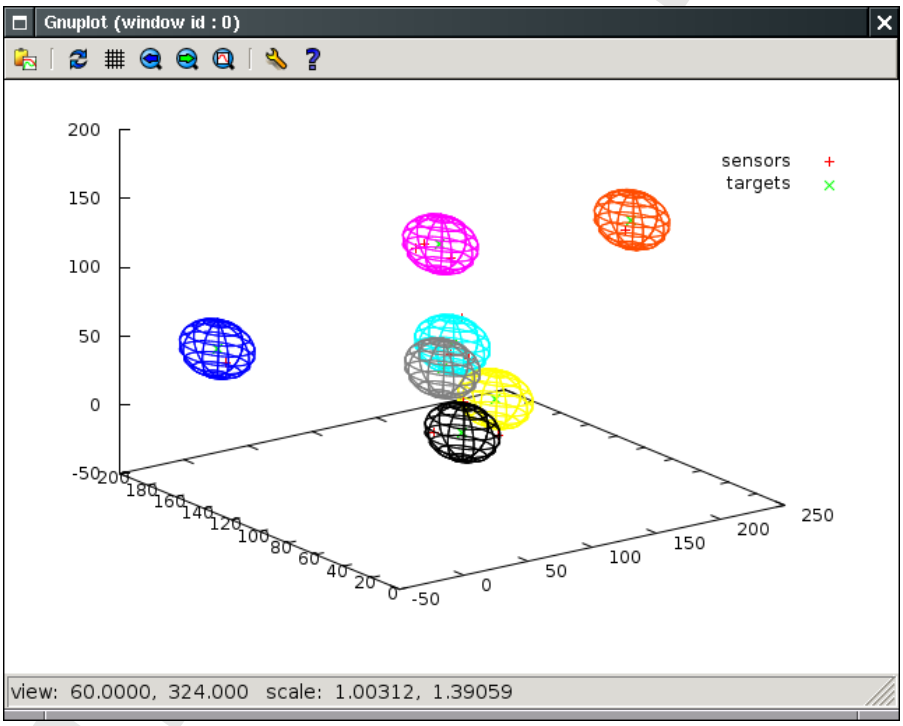
Στο σχήμα 4.11 παρουσιάζεται η γραφική απεικόνιση μιας τρισδιάστατης τοπολογίας με τη βοήθεια των εφαρμογών `draw_terrain_3d` (παραγωγή τρισδιάστατου σχεδίου) και `gnuplot`[9] (απεικόνιση και πλοήγηση στο τρισδιάστατο σχέδιο). Οι σφαίρες γύρω από τους αισθητήρες αποτελούν την τρισδιάστατη εκδοχή των κυκλικών περιοχών γύρω από τους στόχους, που περιγράφηκαν παραπάνω. Στην περίπτωση αυτή, το δίκτυο αποτελείται από 11 αισθητήρες και 7 στόχους, τοποθετημένους σε μια περιοχή $0.008km^3$.

Η παραγωγή τοπολογιών που δεν περιορίζονται από ορισμένο αριθμό διαστάσεων είναι ιδιαίτερα χρήσιμη κατά την εξέταση της απόδοσης ενός αλγορίθμου κάλυψης. Συγκεκριμένα, αφαιρώντας τον περιορισμό των διαστάσεων, το λογισμικό μπορεί να κατασκευάσει πιο πολύπλοκες τοπολογίες από αυτές των 2 και 3 διαστάσεων και μπορεί έτσι να ελέγξει περισσότερες πτυχές του αλγορίθμου κάλυψης. Επίσης, με τις τοπολογίες αυτές είναι δυνατό να ελεγχθούν και αλγόριθμοι κάλυψης που στοχεύουν στην επίλυση διαφορετικών

[9] `http://www.gnuplot.info`



Σχήμα 4.10: Παρουσίαση τοπολογίας 2 διαστάσεων



Σχήμα 4.11: Παρουσίαση τοπολογίας 3 διαστάσεων

προβλημάτων (πέραν των δικτύων αισθητήρων) με δεδομένα που δεν περιορίζονται σε 2 ή 3 διαστάσεις. Το σχετικό λογισμικό (`create_nodes_kd`) δέχεται από το χρήστη τον επιθυμητό αριθμό αισθητήρων, τον αριθμό των στόχων, καθώς και το μέγιστο αριθμό στόχων που θα μπορεί να καλύψει ένας αισθητήρας ($max\_sensor\_targets$). Για κάθε αισθητήρα, το λογισμικό παράγει έναν τυχαίο ακέραιο τ (από μια ομοιόμορφη κατανομή), ο οποίος αναπαριστά τον πραγματικό αριθμό από στόχους που θα καλύψει ο αισθητήρας αυτός

```
glynos@host:~/bgop$ ./create_nodes_kd.pl
usage: create_nodes_kd.pl <num_of_nodes> <num_of_points> <max_node_coverage>
glynos@host:~/bgop$ ./create_nodes_kd.pl 10 6 3
A 3 5 9 10
B 3
C 4 6 7 8 10
D 1 2 5 8
E 4 6 7
F 3 4 6 7 8 9 10
# stats: sensors=10 points=6 min_cardinality=1 min_node_occur=1 max_node_occur=3
         mean_node_occur=2.4
# generated with: create_nodes_kd.pl 10 6 3
```

Σχήμα 4.12: Παραγωγή τοπολογίας δίχως περιορισμούς στις διαστάσεις

(δηλ. $\tau \in [1, max\_sensor\_targets]$). Οι στόχοι που θα καλυφθούν από αυτό τον αισθητήρα επιλέγονται και αυτοί τυχαία (από μια ομοιόμορφη κατανομή).

Στο σχήμα 4.12 παρουσιάζεται ένα παράδειγμα εκτέλεσης της εφαρμογής που παράγει ομάδες κάλυψης δίχως περιορισμούς ως προς τις διαστάσεις της παραγόμενης τοπολογίας. Ο χρήστης ζητά την παραγωγή μιας τοπολογίας με 10 αισθητήρες, και 6 στόχους καθώς και την κάλυψη το πολύ 3 στόχων από κάθε αισθητήρα.

4.5.2 Υλοποίηση αλγορίθμων κάλυψης

Στο πλαίσιο της πειραματικής αξιολόγησης των αλγορίθμων πλήρους κάλυψης της ενότητας 4.4, αναπτύχθηκε νέο λογισμικό σε γλώσσα Perl, το οποίο υλοποιεί τους αλγορίθμους B{GOP}, B{GOP}-random, CCF και Cardei Greedy MSC [134]. Στο λογισμικό αυτό προστέθηκε η υλοποίηση του ευρετικού αλγορίθμου των Slijepcevic et al. [129] σε γλώσσα Java[10], η οποία είναι δημόσια διαθέσιμη από την ιστοσελίδα του εν λόγω έργου[11].

Η εφαρμογή `bgop`

Η υλοποίηση του αλγορίθμου B{GOP} δέχεται ως παράμετρο (ή ως δεδομένα από την καθιερωμένη είσοδο – standard input) το αρχείο που περιγράφει την τοπολογία του δικτύου αισθητήρων. Η λειτουργία της εν λόγω εφαρμογής χωρίζεται στα εξής 4 στάδια:

1. Επεξεργασία της εισόδου I και υπολογισμός των συνόλων S_0, T_0, P και N. Για κάθε κόμβο υπολογίζεται το χαρακτηριστικό B_s καθώς και η τιμή $cov(P_s, S_0)$. Επίσης, υπολογίζονται σταθερές όπως το $max\_badness$ αλλά και το Θεωρητικό Μέγιστο πλήθος των δυνατών ομάδων κάλυψης (max_sets).

2. Εκτέλεση του αλγορίθμου B{GOP}.

3. Έλεγχος ορθότητας των παραγόμενων ομάδων κάλυψης.

4. Προβολή αποτελέσματος (καθώς και λοιπών στατιστικών) στην καθιερωμένη έξοδο (standard output).

Αν ο χρήστης το επιθυμεί, μπορεί να ενεργοποιήσει την προβολή στατιστικών στοιχείων σχετικών με την πρόοδο του αλγορίθμου ανά τακτά χρονικά διαστήματα. Η επιλογή αυτή

[10] `http://java.sun.com`

[11] `http://www.cs.ucla.edu/~sascha/Software/coverage.zip`

είναι συνήθως απενεργοποιημένη καθώς επηρεάζει την ακρίβεια μετρήσεων όπως εκείνη για το χρόνο εκτέλεσης του αλγορίθμου.

Επειδή ο αλγόριθμος απαιτεί συχνή αναζήτηση στοιχείων στα σύνολα P, N και S_{cur}, τα σύνολα αυτά υλοποιήθηκαν με τη μορφή πινάκων κατακερματισμού (hash tables). Επίσης, στην υλοποίηση εντάχθηκαν και οι τεχνικές βελτιστοποίησης του αλγορίθμου που προτάθηκαν στην ενότητα 4.4.1.

```
glynos@host:~/bgop$ ./bgop.pl < ~/thesis/create_nodes_2d_20_5_0.4.txt
# min_extra_covers=0pts, max_extra_covers=2pts
C1    : 6 13
C2    : 2 18
C3    : 3 8
C4    : 12 4
C5    : 15 16 7
C6    : 1 14 17
# Algorithm running time: 0.001333 secs
# Number of generated sets 6 (of 6 maximum)
# $Id: bgop.pl,v 1.31 2006/03/08 18:24:50 glynos Exp $
```

Σχήμα 4.13: Παράδειγμα εκτέλεσης της εφαρμογής `bgop`

Στο σχήμα 4.13 παρουσιάζεται ένα παράδειγμα χρήσης της εφαρμογής `bgop`, που υλοποιεί τον αλγόριθμο B{GOP}. Ο χρήστης παρέχει μέσω της καθιερωμένης εισόδου τα δεδομένα μιας τοπολογίας που απαρτίζεται από 20 αισθητήρες και 5 στόχους, τοποθετημένους σε μια έκταση $16m^2$. Η εφαρμογή τυπώνει στην καθιερωμένη έξοδο τις παραγόμενες ομάδες κάλυψης (C1, ..., C6) αλλά και στατιστικά όπως ο χρόνος εκτέλεσης του αλγορίθμου, το Θεωρητικό Μέγιστο πλήθος ομάδων κάλυψης, καθώς και το ελάχιστο και το μέγιστο πλήθος μη απαιτούμενων καλύψεων στόχων (min_extra_covers και max_extra_covers αντίστοιχα).

Η εφαρμογή `bgop-random`

Η δομή της εφαρμογής `bgop-random`, που υλοποιεί τον αλγόριθμο B{GOP}-random, είναι όμοια με αυτή της εφαρμογής `bgop`.

```
glynos@host:~/bgop/bgop-random.pl < ~/thesis/create_nodes_2d_20_5_0.4.txt
# min_extra_covers=0pts, max_extra_covers=2pts
C1    : 14 6
C2    : 10 4 9
C3    : 2 18
C4    : 8 12
C5    : 7 11 17
C6    : 3 16 1
# Algorithm running time: 0.001795 secs
# Number of generated sets 6 (of 6 maximum)
# $Id: bgop-random.pl,v 1.31 2006/03/08 18:24:50 glynos Exp $
```

Σχήμα 4.14: Παράδειγμα εκτέλεσης της εφαρμογής `bgop-random`

Στο σχήμα 4.14 παρουσιάζεται ένα παράδειγμα εκτέλεσης της εν λόγω εφαρμογής. Ως δεδομένα εισόδου χρησιμοποιήθηκαν τα δεδομένα του προηγούμενου παραδείγματος. Όπως μπορεί να παρατηρήσει κανείς, η τυχαιότητα κατά την επιλογή αισθητήρων οδηγεί στη διαμόρφωση διαφορετικών ομάδων κάλυψης από αυτές του αλγορίθμου B{GOP}.

Η εφαρμογή ccf

Η εφαρμογή `ccf` που υλοποιεί τον αλγόριθμο CCF, δέχεται ως παραμέτρους το μέγιστο πλήθος εμφανίσεων w ενός αισθητήρα στις παραγόμενες ομάδες, τις τιμές των σταθερών α και β, καθώς και το όνομα του αρχείου που περιγράφει την τοπολογία του δικτύου αισθητήρων. Η δομή της εφαρμογής είναι όμοια με αυτή της εφαρμογής `bgop`, με τη μόνη διαφορά ότι στο στάδιο επεξεργασίας των δεδομένων εισόδου και της αρχικοποίησης των σταθερών του αλγορίθμου, αρχικοποιείται το χαρακτηριστικό L_s για κάθε κόμβο s, που περιγράφει το πλήθος των ομάδων κάλυψης στις οποίες μπορεί ο κόμβος αυτός να συμμετάσχει.

```
glynos@host:~/ccf/ccf.pl 2 0.17 0.5 ~/thesis/create_nodes_2d_20_5_0.4.txt
# min_extra_covers=1pts, max_extra_covers=3pts
C1    : 11 16 17 18
C2    : 20 19 9 8
C3    : 13 10 2 4
C4    : 14 3 16 7
C5    : 5 12 19 1
C6    : 15 10 18
C7    : 11 17 8
C8    : 20 9 4
C9    : 13 2 7
C10   : 14 3 1
C11   : 5 12 6
C12   : 15 6
# Algorithm running time: 0.004150 secs
# Number of generated sets 12 (of 12 maximum)
# Total network lifetime: 6.00 * sensor lifetime
```

Σχήμα 4.15: Παράδειγμα εκτέλεσης της εφαρμογής ccf

Στο σχήμα 4.15 παρουσιάζεται το αποτέλεσμα της εκτέλεσης της εφαρμογής `ccf`, με παραμέτρους $w = 2$, $\alpha = 0.17$ και $\beta = 0.5$. Ως αρχείο περιγραφής της τοπολογίας χρησιμοποιείται για άλλη μια φορά το αρχείο που παρουσιάστηκε στο παράδειγμα της εφαρμογής `bgop`. Τόσο ο αλγόριθμος B{GOP} όσο και οι αλγόριθμοι B{GOP}-random και CCF κατάφεραν να παραγάγουν στο παράδειγμα αυτό, το μέγιστο πλήθος των δυνατών ομάδων κάλυψης. Στην περίπτωση του αλγορίθμου CCF, το πλήθος αυτό είναι διπλάσιο από το αντίστοιχο που προέκυψε από τους B{GOP} και B{GOP}-random, καθώς κάθε κόμβος μπορεί να χρησιμοποιηθεί δύο φορές (αλλά σε βάρδιες με χρονική διάρκεια ελαττωμένη κατά το ήμισυ).

Οι εφαρμογές `best-first-search` και `split-triangle-search`

Όπως αναφέρθηκε στο κεφάλαιο 4.4.3, οι τιμές των σταθερών α και β του αλγορίθμου CCF μπορεί να υπολογιστούν με τη χρήση μεθόδων βελτιστοποίησης. Οι εφαρμογές `best-first-search` και `split-triangle-search` υλοποιούν τις μεθόδους βελτιστοποίησης *Best-First Search* και *Split-Triangle* που παρουσιάστηκαν στο παραπάνω κεφάλαιο. Είναι υλοποιημένες σε γλώσσα Python[12] και αξιοποιούν τις παρακάτω κλάσεις:

`PQueue` – Υλοποίηση ουράς προτεραιότητας.

`CoverSetGenerator` – Κλάση "ενθυλάκωσης" του αλγορίθμου παραγωγής ομάδων κάλυψης, που παρέχει μια αντικειμενική συνάρτηση (μέτρηση πλήθους παραγόμενων ο-

[12] `http://www.python.org`

μάδων κάλυψης) και μια μέθοδο υπολογισμού του Θεωρητικού Μέγιστου πλήθους δυνατών ομάδων κάλυψης.

Line – Μοντελοποίηση ευθύγραμμου τμήματος και υπολογισμός μέσου αυτού.

PointTwoD – Μοντελοποίηση σημείου σε διδιάστατο χώρο.

Triangle – Μοντελοποίηση τριγώνου, υποδιαίρεση αυτού και υπολογισμός έγκεντρου.

Η εφαρμογή `best-first-search` δέχεται ως παραμέτρους το μέγιστο πλήθος στοιχείων της ουράς που θα εξεταστούν (max_checks), το μέγιστο πλήθος υποδιαιρέσεων που θα γίνουν στον τριγωνικό χώρο λύσεων (max_depth), την εφαρμογή που θα χρησιμοποιηθεί για να παραγάγει τις ομάδες κάλυψης (δηλ. `ccf`), το μέγιστο πλήθος εμφανίσεων w κάθε αισθητήρα στις ομάδες κάλυψης και, τέλος, το αρχείο περιγραφής της τοπολογίας. Αρχικά, δημιουργεί ένα στιγμιότυπο της κλάσης `CoverSetGenerator` γύρω από τον επιλεγμένο αλγόριθμο παραγωγής ομάδων κάλυψης. Μέσω του στιγμιοτύπου αυτού υπολογίζει το Θεωρητικό Μέγιστο πλήθος των δυνατών ομάδων κάλυψης. Επίσης, λαμβάνει πρόσβαση σε ένα στιγμιότυπο της αντικειμενικής συνάρτησης, το οποίο και θα αξιοποιήσει αργότερα, στα πλαίσια της διαδικασίας βελτιστοποίησης των τιμών α και β. Μόλις ολοκληρωθεί η διαδικασία βελτιστοποίησης, εκτυπώνονται στην καθιερωμένη έξοδο οι τιμές που προέκυψαν για τις σταθερές α και β, το πλήθος των παραγόμενων ομάδων κάλυψης, το Θεωρητικό Μέγιστο πλήθος ομάδων κάλυψης και, τέλος, ο συνολικός χρόνος εκτέλεσης της διαδικασίας βελτιστοποίησης.

Το μέγεθος της ουράς προτεραιότητας και το πλήθος των τριγώνων που προκύπτουν σε κάθε υποδιαίρεση είναι δύο σταθερές που μπορεί εύκολα να παραμετροποιηθούν από τον πηγαίο κώδικα της εφαρμογής. Η έκδοση της εφαρμογής που χρησιμοποιήθηκε στα πειράματα της ενότητας 4.4.3, αξιοποιούσε μια ουρά προτεραιότητας με 400 θέσεις και κάθε τρίγωνο χωριζόταν σε 6 υπο-τρίγωνα.

```
glynos@host:~/ccf$ ./split-triangle-search.py  3 ccf.pl 1 350-40-1.9-a.txt
A(1.000000,0.000000), B(0.000000,1.000000), C(0.000000,0.000000)
65.0
80.0
82.0
A(1.000000,0.000000), B(0.000000,0.000000), C(0.500000,0.500000)
83.0
85.0
A(1.000000,0.000000), B(0.000000,0.000000), C(0.750000,0.250000)
85.0
85.0
85.0
a=0.58 b=0.08 sets=85 of=85
time=16.34 secs
```

Σχήμα 4.16: Παράδειγμα χρήσης της μεθόδου εύρεσης βέλτιστων τιμών Split-Triangle για τις παραμέτρους α, β του αλγορίθμου CCF

Η εφαρμογή `split-triangle-search` έχει όμοια δομή με αυτή της `best-first-search`. Εφόσον εδώ δε χρησιμοποιείται κάποια ουρά προτεραιότητας, δε χρειάζεται να δηλωθεί ο μέγιστος αριθμός στοιχείων αυτής που θα εξεταστούν. Στο σχήμα 4.16 φαίνεται ένα παράδειγμα εκτέλεσης της εφαρμογής `split-triangle-search` για ένα δίκτυο με 350 αισθητήρες, 40 στόχους και συνολική επιφάνεια $361m^2$. Ο χρήστης στο παράδειγμα αυτό επιλέγει μέγιστο αριθμό υποδιαιρέσεων ($steps$) ίσο με 3 και μέγιστο πλήθος εμφανίσεων (σε ομάδες κάλυψης) w ίσο με 1. Κατά τη διάρκεια εκτέλεσης του αλγορίθμου τυπώνονται

στην καθιερωμένη έξοδο λαθών, το τρίγωνο που εξετάζεται εκείνη τη στιγμή καθώς και οι υψηλότερες καταγεγραμμένες τιμές της αντικειμενικής συνάρτησης, όταν αυτή εξετάζει το έγκεντρο των υποδιαιρέσεων του τριγώνου αυτού.

Η εφαρμογή `cardei-greedy-msc`

Η εφαρμογή `cardei-greedy-msc` αποτελεί μια υλοποίηση του αλγορίθμου Cardei Greedy MSC [134] σε γλώσσα Perl. Η δομή της εφαρμογής `cardei-greedy-msc` είναι όμοια με αυτή της εφαρμογής `ccf`. Εφόσον πρόκειται για την υλοποίηση ενός αλγορίθμου με δυνατότητα παραγωγής ομάδων κάλυψης με κοινά στοιχεία, ο χρήστης θα πρέπει να παρέχει στην εφαρμογή, ως παράμετρο, την επιθυμητή τιμή για το μέγιστο πλήθος εμφανίσεων w.

Σύμφωνα με τον αλγόριθμο που περιγράφεται στην εργασία [134], μόλις ανιχνευθεί ένας *κρίσιμος στόχος*, επιλέγεται ένας αισθητήρας που τον καλύπτει. Σε περίπτωση όπου υπάρχουν περισσότεροι του ενός αισθητήρες που καλύπτουν τον κρίσιμο στόχο, τότε επιλέγεται ο αισθητήρας που λαμβάνει τη μεγαλύτερη βαθμολογία ως προς μια αντικειμενική συνάρτηση, ονόματι *contribution*. Η μορφή της αντικειμενικής συνάρτησης *contribution* που χρησιμοποιήθηκε στην εφαρμογή `cardei-greedy-msc` είναι η ακόλουθη:

$$contribution(s) = cov(P_s, T_{cur}) + L_s,$$

όπου $cov(P_s, T_{cur})$ είναι το πλήθος των ακάλυπτων στόχων που μπορεί να καλύψει ο υπό εξέταση αισθητήρας και L_s είναι το *δυναμικό συμμετοχής* του αισθητήρα.

```
glynos@host:~/ccf$ ./cardei-greedy-msc.pl 2 ~/thesis/create_nodes_2d_20_5_0.4.txt
C1    : 6 11
C2    : 6 3
C3    : 1 12
C4    : 7 2
C5    : 8 15
C6    : 4 3
C7    : 1 9 20
C8    : 18 17 14
C9    : 7 12
C10   : 8 2
C11   : 4 15
C12   : 18 9 10
# Algorithm running time: 0.001874 secs
# Number of generated sets 12 (of 12 maximum)
# Total network lifetime: 6.00 * sensor lifetime
```

Σχήμα 4.17: Παράδειγμα εκτέλεσης της εφαρμογής `cardei-greedy-msc`

Στο σχήμα 4.17 παρουσιάζεται ένα παράδειγμα εκτέλεσης της εφαρμογής `cardei-greedy-msc` με $w = 2$ και αρχείο εισόδου όμοιο με αυτό που χρησιμοποιήθηκε στο παράδειγμα της εφαρμογής `bgop`.

Η εφαρμογή `slijepcevic`

Η εφαρμογή `slijepcevic` αποτελεί μια υλοποίηση σε Java του ευρετικού αλγορίθμου που παρουσιάστηκε στην εργασία [129]. Η εφαρμογή αυτή στηρίζεται σε κώδικα των δημιουργών της εργασίας [129], ο οποίος όμως έχει τροποποιηθεί κατάλληλα ώστε να υποστηρίζει τη σύνταξη του αρχείου εισόδου που περιγράφηκε παραπάνω.

```
glynos@host:~/bgop$ java Simulation f ~/thesis/create_nodes_2d_20_5_0.4.txt
# time 0.073
C1 : 6 2
C2 : 1 3
C3 : 7 12
C4 : 4 15
C5 : 8 9 5
C6 : 18 17 10
# generated sets 6
```

Σχήμα 4.18: Παράδειγμα εκτέλεσης της εφαρμογής `slijepcevic`

Το σχήμα 4.18 παρουσιάζει την εκτέλεση της εφαρμογής `slijepcevic` με παράμετρο το αρχείο περιγραφής τοπολογίας που χρησιμοποιήθηκε στο παράδειγμα της εφαρμογής `bgop`.

4.5.3 Περιβάλλον προσομοίωσης

Κατά την πειραματική αξιολόγηση των αλγορίθμων που παρουσιάστηκαν στις προηγούμενες ενότητες, ήταν συχνά αναγκαία η (επανειλημένη) εκτέλεση ενός αλγορίθμου σε διαφορετικές τοπολογίες, καθώς και η εξαγωγή στατιστικών στοιχείων από τα αποτελέσματα της κάθε εκτέλεσης. Προκειμένου να επιταχυνθεί αυτή η διαδικασία, δημιουργήθηκε ένα περιβάλλον προσομοίωσης σε γλώσσα Perl το οποίο επιτρέπει την πλήρη αυτοματοποίηση των παραπάνω εργασιών.

Για κάθε αλγόριθμο κάλυψης που θα εξεταστεί δημιουργείται ένα προφίλ, το οποίο περιέχει:

- ένα σύντομο κωδικό όνομα για αυτό τον αλγόριθμο (π.χ. `bgop`),
- τον τρόπο κλήσης της εφαρμογής που υλοποιεί αυτό τον αλγόριθμο (π.χ. `'./bgop.pl %f'`),
- τον παρόντα κατάλογο εργασίας (present working directory) που θα χρησιμοποιηθεί κατά την εκτέλεση της εφαρμογής,
- την κανονική έκφραση (regular expression) που αντιστοιχεί στο χρόνο εκτέλεσης του αλγορίθμου, όπως αυτός περιγράφεται στην καθιερωμένη έξοδο της εφαρμογής, και
- την κανονική έκφραση που αντιστοιχεί στο πλήθος των παραγόμενων ομάδων κάλυψης.

Κάθε προφίλ έχει τη μορφή ενός πίνακα κατακερματισμού σε γλώσσα Perl. Όπως φαίνεται στο παράδειγμα του σχήματος 4.19 το πεδίο με κλειδί `cmdline` περιγράφει τον τρόπο εκτέλεσης του αλγορίθμου. Το πεδίο με κλειδί `working_dir` περιγράφει τον επιθυμητό παρόντα κατάλογο εργασίας για την εφαρμογή. Τέλος, οι συμβολοσειρές που αντιστοιχούν στα κλειδιά `time_pattern` και `sets_pattern`, είναι οι κανονικές εκφράσεις που θα χρησιμοποιηθούν για την εξαγωγή του χρόνου εκτέλεσης και του πλήθους των παραγόμενων ομάδων κάλυψης.

Το ίδιο το προφίλ ενός αλγορίθμου αποτελεί μια τιμή σε ένα ευρύτερο πίνακα κατακερματισμού, ονόματι `simulation`, που έχει ως κλειδιά τα κωδικά ονόματα των αλγορίθμων που θα εξεταστούν κατά την προσομοίωση. Στο παραπάνω παράδειγμα έχει δοθεί το κωδικό όνομα `random` στον αλγόριθμο B{GOP}-random.

```
$simulation{"random"}{"cmdline"} = "./bgop-random.pl %f";
$simulation{"random"}{"working_dir"} = ".";
$simulation{"random"}{"time_pattern"} = '# Algorithm running time: ([0-9]+\.[0-9]+)';
$simulation{"random"}{"sets_pattern"} = '# Number of generated sets ([0-9]+)';
```

Σχήμα 4.19: Το προφίλ του αλγορίθμου B{GOP}-random στο περιβάλλον προσομοίωσης

Ο τρόπος εκτέλεσης μιας εφαρμογής περιγράφεται με μια συμβολοσειρά η οποία περιέχει το μονοπάτι για το εκτελέσιμο της εφαρμογής, τυχόν παραμέτρους που απαιτούνται από την εφαρμογή, καθώς και τα χαρακτηριστικά σύμβολα `'%f'`. Τα σύμβολα αυτά, θα αντικατασταθούν από το περιβάλλον προσομοίωσης με το όνομα του αρχείου τοπολογίας, για το οποίο ο αλγόριθμος θα παραγάγει ομάδες κάλυψης.

Έχοντας έτοιμα τα προφίλ των αλγορίθμων κάλυψης που επιθυμεί να εξετάσει, ο χρήστης μπορεί πλέον να ξεκινήσει τη διαδικασία προσομοίωσης. Η βασική εφαρμογή του περιβάλλοντος προσομοίωσης `batch-simulation`, δέχεται ως παραμέτρους το πλήθος των επιθυμητών εκτελέσεων[13] ανά αρχείο τοπολογίας, καθώς και τα μονοπάτια των αρχείων τοπολογίας στα οποία θα εξεταστούν οι υπό προσομοίωση αλγόριθμοι.

```
glynos@host:~/bgop$ ./batch-simulation.pl 3 350-40-1.9-a.txt 350-40-1.9-b.txt
350-40-1.9-a.txt            bgop   84 sets 0.792 sec 0.009 sec/set
350-40-1.9-a.txt     slijepcevic   85 sets 1.679 sec 0.020 sec/set
350-40-1.9-b.txt            bgop   96 sets 0.864 sec 0.009 sec/set
350-40-1.9-b.txt     slijepcevic   97 sets 1.724 sec 0.018 sec/set
```

Σχήμα 4.20: Παράδειγμα εκτέλεσης της εφαρμογής προσομοίωσης `batch-simulation`

Η προσομοίωση των αλγορίθμων γίνεται ανά αρχείο τοπολογίας. Για κάθε αλγόριθμο σημειώνεται ο ελάχιστος χρόνος εκτέλεσης, το μέγιστο πλήθος παραγόμενων ομάδων κάλυψης και ο ελάχιστος χρόνος εκτέλεσης ανά παραγόμενη ομάδα κάλυψης. Τέλος, οι πληροφορίες αυτές γίνονται διαθέσιμες στο χρήστη από την καθιερωμένη έξοδο της εφαρμογής προσομοίωσης, όπως φαίνεται στο σχήμα 4.20.

4.6 Πειραματική αξιολόγηση

Χρησιμοποιώντας το περιβάλλον που περιγράφηκε στην προηγούμενη ενότητα πραγματοποιήθηκε μια σειρά προσομοιώσεων προκειμένου να μελετηθούν οι επιδόσεις των αλγορίθμων B{GOP}, B{GOP}-random, CCF, Slijepcevic [129] και Cardei Greedy MSC [134] κατά την παραγωγή ομάδων κάλυψης με ή δίχως κοινούς αισθητήρες.

Για κάθε τύπο τοπολογίας που θα εξεταζόταν από τους παραπάνω αλγόριθμους δημιουργήθηκαν 5 παραλλαγές. Από τις 5 αυτές παραλλαγές υπολογίστηκε ο μέσος χρόνος παραγωγής μίας ομάδας καθώς και ο μέσος αριθμός παραγόμενων ομάδων ανά αλγόριθμο.

Για τον αλγόριθμο Slijepcevic [129] θα πρέπει να σημειωθεί ότι είναι υλοποιημένος σε διαφορετική γλώσσα προγραμματισμού από τους υπόλοιπους και έτσι θα εξεταστούν για αυτόν οι σχετικοί χρόνοι παραγωγής ομάδων κάλυψης.

Όλες οι προσομοιώσεις πραγματοποιήθηκαν σε Η/Υ τύπου Pentium IV 3.4GHz με 2GB RAM και λειτουργικό σύστημα GNU/Linux.

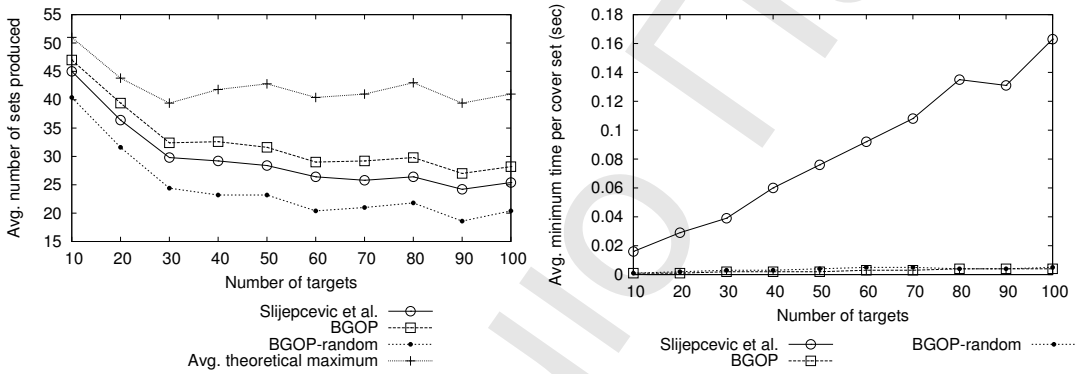
[13] Μέσω της ελάχιστης καταγεγραμμένης καθυστέρησης από τις πολλαπλές εκτελέσεις, μπορεί να προσεγγιστεί ο πραγματικός χρόνος εκτέλεσης ενός αλγορίθμου (δηλ. ο χρόνος που δεν περιλαμβάνει τις καθυστερήσεις που οφείλονται σε διαδικασίες του λειτουργικού συστήματος).

4.6.1 Παραγωγή ομάδων κάλυψης δίχως κοινούς αισθητήρες

Στην πρώτη φάση της πειραματικής αξιολόγησης εξετάστηκαν οι αλγόριθμοι B{GOP}, B{GOP}-random και Slijepcevic [129] οι οποίοι παράγουν ομάδες κάλυψης που δεν περιλαμβάνουν κοινούς αισθητήρες.

Σενάριο Α' - Κάλυψη Στόχων

Με το λογισμικό `create_nodes_kd` δημιουργήθηκε μια σειρά τοπολογιών όπου 100 αισθητήρες παρακολουθούν ένα μεταβλητό αριθμό από στόχους οι οποίοι κυμαίνονται μεταξύ 10 και 100. Κάθε τοπολογία απαρτίζεται από συγκεκριμένο αριθμό αισθητήρων και στόχων, όμως κάθε αισθητήρας που μετέχει σε αυτή καλύπτει τυχαίο αριθμό στόχων με μέγιστο το σύνολο των στόχων της τοπολογίας. Ο αριθμός των αισθητήρων και των στόχων σε αυτές τις τοπολογίες παραπέμπει σε δίκτυα αισθητήρων που προσφέρουν κάλυψη στόχων.



(α′) (β′)

Σχήμα 4.21: Μέσος αριθμός παραγόμενων ομάδων κάλυψης και μέσος χρόνος παραγωγής ομάδας σε δίκτυα με 100 αισθητήρες και μεταβλητό πλήθος στόχων.

Τα αποτελέσματα των προσομοιώσεων στις παραπάνω τοπολογίες παρουσιάζονται στα σχήματα 4.21α′ και 4.21β′. Συγκεκριμένα, στο σχήμα 4.21α′ παρουσιάζεται το μέσο πλήθος παραγόμενων ομάδων κάλυψης για κάθε αλγόριθμο και η μέση τιμή του θεωρητικού μέγιστου για κάθε τύπο τοπολογίας. Αντίστοιχα, στο σχήμα 4.21β′ παρουσιάζεται ο μέσος χρόνος παραγωγής μιας ομάδας για κάθε ένα από τους αλγόριθμους που συμμετείχαν στην προσομοίωση.

Τα αποτελέσματα των προσομοιώσεων δείχνουν ότι ο αλγόριθμος B{GOP} παρήγαγε σε κάθε περίπτωση περισσότερες ομάδες κάλυψης από τον αλγόριθμο των Slijepcevic et al [129]. Από το διάγραμμα των χρόνων εκτέλεσης παρατηρεί κανείς ότι ο αλγόριθμος Slijepcevic χρειάστηκε δεκαπλάσιο χρόνο για να επεξεργαστεί κάθε ομάδα κάλυψης όταν οι στόχοι δεκαπλασιάστηκαν. Ο αντίστοιχος χρόνος για τον αλγόριθμο B{GOP} μόλις τετραπλασιάστηκε, καθιστόντας τον πιο κατάλληλο για χρήση σε δίκτυα επιτήρησης με μεγάλο πλήθος στόχων. Οι μέσοι χρόνοι παραγωγής ομάδων κάλυψης παρουσιάζονται αναλυτικά στον πίνακα 4.2.

Ο αλγόριθμος B{GOP}-random παρουσίασε παρόμοιους χρόνους εκτέλεσης με τον B{GOP}, παραμένει όμως πιο αργός από τον δεύτερο καθώς χρησιμοποιεί μια πιο πολύπλοκη μέθοδο επιλογής αισθητήρων. Επίσης, οι επιδόσεις του στην παραγωγή ομάδων κάλυψης ήταν αρκετά μειωμένες σε σχέση με τον αλγόριθμο Slijepcevic.

	Slijepcevic		B{GOP}		B{GOP}-random		
Πλήθος Στόχων	Ομάδες[1]	Χρόνος[2]	Ομάδες[1]	Χρόνος[2]	Ομάδες[1]	Χρόνος[2]	Θεωρ. Μέγιστο[3]
10	45,000	0,016	47,000	0,001	40,400	0,001	51,000
20	36,400	0,029	39,400	0,001	31,600	0,002	43,800
30	29,800	0,039	32,400	0,002	24,400	0,003	39,400
40	29,200	0,060	32,600	0,002	23,200	0,003	41,800
50	28,400	0,076	31,600	0,002	23,200	0,004	42,800
60	26,400	0,092	29,000	0,003	20,400	0,005	40,400
70	25,800	0,108	29,200	0,003	21,000	0,005	41,000
80	26,400	0,135	29,800	0,004	21,800	0,004	43,000
90	24,200	0,131	27,000	0,004	18,600	0,004	39,400
100	25,400	0,163	28,200	0,004	20,400	0,005	41,000

[1] Μέσο πλήθος παραγόμενων ομάδων κάλυψης

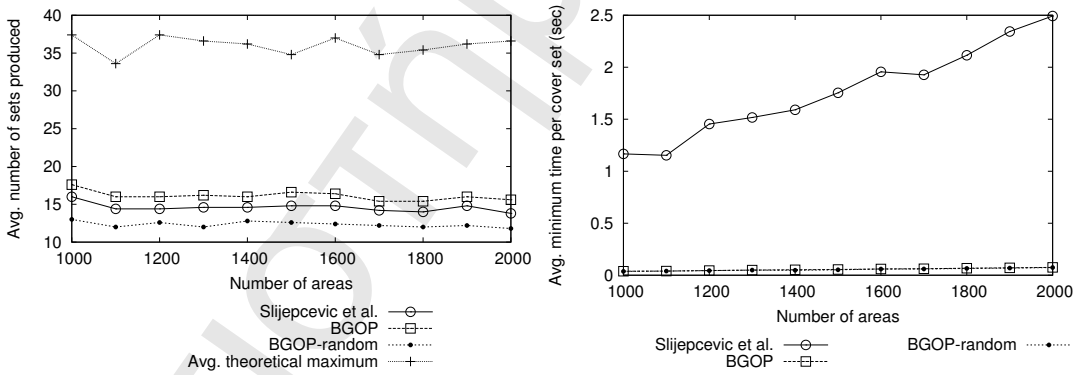
[2] Μέσος χρόνος παραγωγής μίας ομάδας κάλυψης (δευτερόλεπτα)

[3] Μέσος όρος θεωρητικού μεγίστου

Πίνακας 4.2: Αποτελέσματα προσομοίωσης με 100 αισθητήρες και μεταβλητό πλήθος στόχων.

Σενάριο Β' - Κάλυψη Περιοχών

Το δεύτερο σενάριο που εξετάστηκε αφορούσε την προσομοίωση δικτύων που προσφέρουν κάλυψη περιοχών. Σε αυτά τα δίκτυα ο αριθμός των αισθητήρων είναι τυπικά μικρότερος από τον αριθμό των περιοχών. Έτσι, δημιουργήθηκαν τοπολογίες με 100 αισθητήρες και μεταβλητό αριθμό περιοχών κάλυψης οι οποίες κυμαίνονταν μεταξύ 1000 και 2000. Ο κάθε αισθητήρας επιτηρούσε τυχαίο αριθμό περιοχών, με μέγιστο πλήθος επιτηρούμενων περιοχών το σύνολο των περιοχών της κάθε τοπολογίας.



(α΄) (β΄)

Σχήμα 4.22: Μέσος αριθμός παραγόμενων ομάδων κάλυψης και μέσος χρόνος παραγωγής ομάδας σε δίκτυα με 100 αισθητήρες και μεταβλητό πλήθος περιοχών.

Τα αποτελέσματα που παρουσιάζονται στα σχήματα 4.22α΄ και 4.22β΄ δείχνουν ότι ο αλγόριθμος B{GOP} παραμένει ταχύτερος και πιο αποδοτικός από τον Slijepcevic και τον B{GOP}-random σε κάθε περίπτωση του σεναρίου κάλυψης περιοχών. Επίσης, όλοι οι αλγόριθμοι χρειάστηκαν περιπού το διπλάσιο χρόνο εκτέλεσης για να παραγάγουν μια ομάδα κάλυψης όταν ο αριθμός των περιοχών διπλασιάστηκε. Τα αναλυτικά αποτελέσματα των προσομοιώσεων παρουσιάζονται στον πίνακα 4.3.

Ο αυξημένος χρόνος εκτέλεσης που παρουσιάζεται σε αυτές τις τοπολογίες οφείλεται

	Slijepcevic		B{GOP}		B{GOP}-random		
Πλήθος Περιοχών	Ομάδες[1]	Χρόνος[2]	Ομάδες[1]	Χρόνος[2]	Ομάδες[1]	Χρόνος[2]	Θεωρ. Μέγιστο[3]
1000	16,000	1,167	17,600	0,038	13,000	0,037	37,400
1100	14,400	1,153	16,000	0,040	12,000	0,040	33,600
1200	14,400	1,454	16,000	0,045	12,600	0,044	37,400
1300	14,600	1,517	16,200	0,049	12,000	0,049	36,600
1400	14,600	1,591	16,000	0,052	12,800	0,048	36,200
1500	14,800	1,754	16,600	0,055	12,600	0,052	34,800
1600	14,800	1,956	16,400	0,060	12,400	0,059	37,000
1700	14,200	1,927	15,400	0,062	12,200	0,060	34,800
1800	14,000	2,115	15,400	0,067	12,000	0,065	35,400
1900	14,800	2,344	16,000	0,071	12,200	0,069	36,200
2000	13,800	2,493	15,600	0,075	11,800	0,074	36,600

[1] Μέσο πλήθος παραγόμενων ομάδων κάλυψης

[2] Μέσος χρόνος παραγωγής μίας ομάδας κάλυψης (δευτερόλεπτα)

[3] Μέσος όρος θεωρητικού μεγίστου

Πίνακας 4.3: Αποτελέσματα προσομοίωσης με 100 αισθητήρες και μεταβλητό πλήθος περιοχών κάλυψης.

κυρίως στο μεγάλο αριθμό περιοχών ανά αισθητήρα που πρέπει να επεξεργαστεί ο εκάστοτε αλγόριθμος. Στα σενάρια που ακολουθούν θα εξεταστεί η σχέση του χρόνου εκτέλεσης των αλγορίθμων με το πλήθος των αισθητήρων, το πλήθος των περιοχών/στόχων και την πυκνότητα δόμησης του δικτύου.

Σενάριο Γ' - Μεταβλητό πλήθος αισθητήρων

Μέσω της εφαρμογής `create_nodes_kd` δημιουργήθηκε μια σειρά τοπολογιών προκειμένου να μελετηθεί η σχέση μεταξύ του αριθμού των εξεταζόμενων αισθητήρων και του χρόνου παραγωγής μιας ομάδας κάλυψης. Οι τοπολογίες αυτές περιελάμβαναν 50 στόχους και μεταβλητό αριθμό αισθητήρων, ο οποίος κυμαινόταν μεταξύ 100 και 1000 αισθητήρες. Κάθε αισθητήρας κάλυπτε τυχαίο αριθμό στόχων, με μέγιστο αριθμό καλυπτόμενων στόχων το πλήθος των στόχων της τοπολογίας.

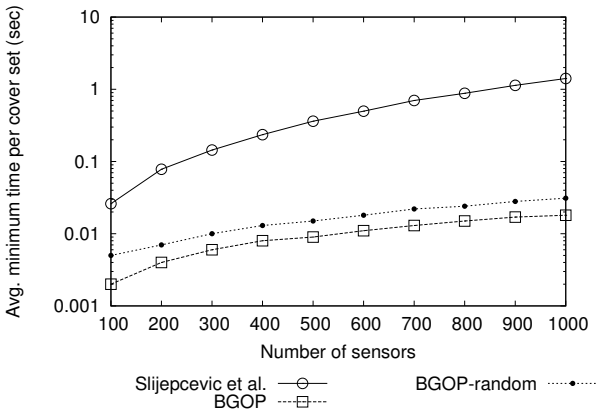
	Slijepcevic		B{GOP}		B{GOP}-random		
Αρ. Αισθητήρων	Ομάδες[1]	Χρόνος[2]	Ομάδες[1]	Χρόνος[2]	Ομάδες[1]	Χρόνος[2]	Θεωρ. Μέγιστο[3]
100	11,800	0,026	13,000	0,002	9,200	0,005	16,600
200	29,600	0,078	30,800	0,004	19,800	0,007	39,600
300	45,200	0,144	48,200	0,006	29,200	0,010	62,800
400	63,000	0,235	65,600	0,008	40,200	0,013	82,400
500	80,400	0,361	84,600	0,009	50,400	0,015	109,800
600	98,400	0,498	102,400	0,011	60,200	0,018	129,600
700	117,800	0,700	122,600	0,013	71,400	0,022	154,600
800	134,600	0,881	140,800	0,015	82,400	0,024	182,800
900	153,400	1,133	160,200	0,017	91,200	0,028	203,800
1000	174,000	1,410	182,400	0,018	103,000	0,031	238,200

[1] Μέσο πλήθος παραγόμενων ομάδων κάλυψης

[2] Μέσος χρόνος παραγωγής μίας ομάδας κάλυψης (δευτερόλεπτα)

[3] Μέσος όρος θεωρητικού μεγίστου

Πίνακας 4.4: Αποτελέσματα προσομοίωσης με 50 στόχους και μεταβλητό αριθμό αισθητήρων.



Σχήμα 4.23: Μέσος χρόνος παραγωγής ομάδας κάλυψης σε δίκτυα με 50 στόχους και μεταβλητό αριθμό αισθητήρων.

Ο πίνακας 4.4 περιλαμβάνει τα αποτελέσματα των σχετικών προσομοιώσεων. Τα δεδομένα δείχνουν ότι ο αλγόριθμος Slijepcevic παρουσιάζει μια αύξηση στο χρόνο που απαιτεί για την παραγωγή μίας ομάδας κάλυψης, όταν ο αριθμός των εξεταζόμενων αισθητήρων αυξάνει. Συγκεκριμένα, ο λόγος του μέσου χρόνου (που χρειάστηκε για την παραγωγή μίας ομάδας) στις τοπολογίες 200, 500 και 1000 αισθητήρων προς τον αντίστοιχο χρόνο για τις τοπολογίες 100 αισθητήρων είναι περίπου 3:1, 14:1 και 54:1 αντίστοιχα.

Οι αλγόριθμοι B{GOP} και B{GOP}-random παρουσιάζουν και αυτοί αύξηση στο χρόνο παραγωγής ομάδων κάλυψης αλλά η αύξηση αυτή πραγματοποιείται με μικρότερους ρυθμούς απ'ότι αυτή που παρατηρήθηκε στον αλγόριθμο Slijepcevic. Για τον B{GOP} οι λόγοι αύξησης για τις τοπολογίες 200, 500 και 1000 αισθητήρων είναι περίπου 2:1, 5:1 και 9:1, ενώ για τον B{GOP}-random οι αντίστοιχοι λόγοι είναι περίπου 7:5, 3:1 και 6:1. Θα πρέπει να σημειωθεί ότι ο B{GOP}-random απαιτεί σε κάθε περίπτωση περισσότερο χρόνο για την παραγωγή μίας ομάδας κάλυψης από ότι ο αλγόριθμος B{GOP}.

Στο σχήμα 4.23 παρουσιάζεται και γραφικά η καθυστέρηση που σημειώνεται από τους τρεις αλγορίθμους κατά την παραγωγή ομάδων κάλυψης, όταν ο αριθμός των διαθέσιμων αισθητήρων αυξάνει. Οι τιμές του κατακόρυφου άξονα του σχήματος δίνονται σε λογαριθμική κλίμακα.

Σενάριο Δ' - Μεταβλητό πλήθος στόχων

Παρόλο που και οι τρεις αλγόριθμοι επηρεάζονται πρωτίστως από το πλήθος των διαθέσιμων αισθητήρων, θα ήταν ενδιαφέρον να διερευνηθεί κατά πόσο επηρεάζονται και από το πλήθος των στόχων μιας τοπολογίας.

Για το σκοπό αυτό δημιουργήθηκε μια νεά σειρά τοπολογιών με το λογισμικό `create_nodes_kd`, με σταθερό αριθμό 250 αισθητήρων και μεταβλητό αριθμό στόχων. Κάθε αισθητήρας κάλυπτε τυχαίο αριθμό στόχων, με μέγιστο πλήθος καλυπτόμενων στόχων ανά αισθητήρα το συνολικό πλήθος των διαθέσιμων στόχων.

Στον πίνακα 4.5 παρουσιάζονται τα αποτελέσματα των σχετικών προσομοιώσεων. Ο αλγόριθμος Slijepcevic παρουσιάζει μια αρχική επιβάρυνση στο χρόνο παραγωγής των ομάδων η οποία οφείλεται στο γεγονός ότι στις τοπολογίες 50 και 100 στόχων υπάρχει μεγάλο πλήθος στόχων που καλύπτονται από κοινούς αισθητήρες. Στη συνέχεια, η καθυστέρηση στο χρόνο παραγωγής μειώνεται στην τοπολογία των 200 αισθητήρων και προοδευτικά αυξάνει μέχρι την τοπολογία των 500. Η αύξηση αυτή οφείλεται στο πλήθος των στόχων που εξε-

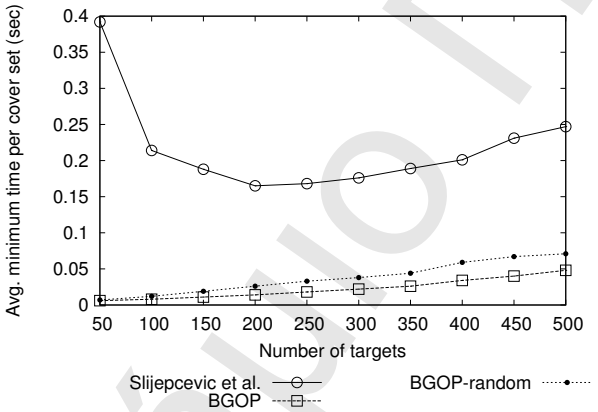
	Slijepcevic		B{GOP}		B{GOP}-random		
Πλήθος Στόχων	Ομάδες[1]	Χρόνος[2]	Ομάδες[1]	Χρόνος[2]	Ομάδες[1]	Χρόνος[2]	Θεωρ. Μέγιστο[3]
50	80,200	0,392	81,800	0,006	56,000	0,007	109,800
100	29,000	0,214	30,600	0,008	20,000	0,012	45,800
150	17,800	0,188	19,000	0,011	12,600	0,019	27,600
200	11,800	0,165	12,600	0,014	8,600	0,026	17,200
250	8,600	0,168	8,800	0,018	6,000	0,033	11,200
300	7,200	0,176	6,800	0,022	5,200	0,038	9,400
350	5,200	0,189	5,800	0,026	4,000	0,044	6,800
400	4,400	0,201	3,800	0,034	3,200	0,059	5,800
450	3,400	0,231	3,400	0,040	3,000	0,067	4,400
500	3,400	0,247	3,000	0,048	2,800	0,071	4,200

[1] Μέσο πλήθος παραγόμενων ομάδων κάλυψης

[2] Μέσος χρόνος παραγωγής μίας ομάδας κάλυψης (δευτερόλεπτα)

[3] Μέσος όρος θεωρητικού μεγίστου

Πίνακας 4.5: Αποτελέσματα προσομοίωσης με 250 αισθητήρες και μεταβλητό αριθμό στόχων.



Σχήμα 4.24: Μέσος χρόνος παραγωγής ομάδας κάλυψης σε δίκτυα με 250 αισθητήρες και μεταβλητό αριθμό στόχων.

τάζει ο αλγόριθμος και σε κάθε περίπτωση είναι μικρότερη της αύξησης που παρατηρήθηκε στο Σενάριο Γ' όπου υπήρχε μεταβολή του πλήθους των αισθητήρων.

Αντίστοιχα, οι αλγόριθμοι B{GOP} και B{GOP}-random δεν παρουσιάζουν κάποια καθυστέρηση εξαιτίας των αρχικών κοινών καλύψεων στόχων, αλλά παρουσιάζουν μια μικρή (σχεδόν σταθερή) αύξηση στην καθυστέρηση όσο ο αριθμός των αισθητήρων αυξάνεται. Στο σχήμα 4.24 που παρουσιάζονται γραφικά τα αποτελέσματα της προσομοίωσης φαίνεται καθαρά ότι ο ρυθμός αύξησης της καθυστέρησης για τον αλγόριθμο B{GOP}-random είναι μεγαλύτερος αυτού του B{GOP}. Επίσης, ο ρυθμός αύξησης της καθυστέρησης του B{GOP} είναι μικρότερος του ρυθμού αύξησης που παρατηρείται στον αλγόριθμο Slijepcevic.

Σενάριο Ε' - Μεταβλητή πυκνότητα δικτύου

Επίσης ενδιαφέρουσα είναι η σχέση μεταξύ πυκνής / αραιής δόμησης ενός δικτύου και χρόνου εκτέλεσης των αλγορίθμων. Ένα πυκνά δομημένο δίκτυο αισθητήρων παρέχει κάλυψη σε στόχους μέσω πολλών αισθητήρων. Όπως περιγράφηκε στην παράγραφο 4.3.3, στα δίκτυα αυτά υπάρχει μεγαλύτερη πιθανότητα δύο ή περισσότεροι κόμβοι που επιλέγονται

για χρήση σε μια ομάδα κάλυψης να καλύπτουν τον ίδιο κρίσιμο στόχο. Για το λόγο αυτό πολλοί αλγόριθμοι όπως ο Slijepcevic καταφεύγουν σε στρατηγικές αποφυγής των κόμβων που καλύπτουν κρίσιμους στόχους. Εκτός από την καθυστέρηση που οφείλεται σε αυτές τις στρατηγικές, οι αλγόριθμοι συναντούν στα πυκνά δομημένα δίκτυα περισσότερους αισθητήρες ανά στόχο ή/και περισσότερους στόχους ανά αισθητήρα.

	Slijepcevic		**B{GOP}**		**B{GOP}-random**		
Πεδίο[1]	Ομάδες[2]	Χρόνος[3]	Ομάδες[2]	Χρόνος[3]	Ομάδες[2]	Χρόνος[3]	Θεωρ. Μέγιστο[4]
100	75,600	0,269	77,200	0,003	78,200	0,003	84,000
121	63,400	0,240	65,200	0,003	64,400	0,004	75,200
144	52,200	0,196	54,800	0,003	52,200	0,004	58,800
169	48,800	0,164	48,800	0,003	43,200	0,004	51,600
193	41,600	0,140	43,800	0,003	37,600	0,004	45,000
225	38,000	0,136	38,000	0,003	34,000	0,003	38,600
256	32,400	0,104	32,400	0,003	27,800	0,004	32,600
289	30,200	0,110	30,200	0,003	26,600	0,004	30,400
324	26,800	0,094	27,200	0,003	24,200	0,004	27,400
361	21,200	0,086	22,800	0,003	21,000	0,004	24,400
400	22,200	0,086	22,200	0,003	21,000	0,004	22,600

[1] Μέγεθος πεδίου σε τ.μ.

[1] Μέσο πλήθος παραγόμενων ομάδων κάλυψης

[2] Μέσος χρόνος παραγωγής μίας ομάδας κάλυψης (δευτερόλεπτα)

[3] Μέσος όρος θεωρητικού μεγίστου

Πίνακας 4.6: Αποτελέσματα προσομοίωσης με 100 αισθητήρες, 50 στόχους και μεταβλητή πυκνότητα δικτύου.

Στον πίνακα 4.6 παρουσιάζονται τα αποτελέσματα προσομοίωσης η οποία έγινε σε δίκτυα μεταβλητής πυκνότητας με σταθερό πλήθος κόμβων (100) και στόχων (50). Συγκεκριμένα τα δίκτυα αυτά βασίζονται σε διδιάστατες τοπολογίες οι οποίες παράχθηκαν από το λογισμικό `create_nodes_2d`. Οι τοπολογίες αυτές διαφέρουν μεταξύ τους ως προς την τοποθέτηση των κόμβων και στόχων αλλά και ως προς το συνολικό πεδίο τοποθέτησης. Για να προσομοιωθούν συνθήκες πυκνής και αραιής δόμησης τοποθετήθηκε ο ίδιος αριθμός κόμβων και στόχων σε ολοένα και μεγαλύτερη επιφάνεια. Η έκταση της επιφάνειας αυτής για κάθε εξεταζόμενη τοπολογία περιγράφεται στην πρώτη στήλη του πίνακα 4.6.

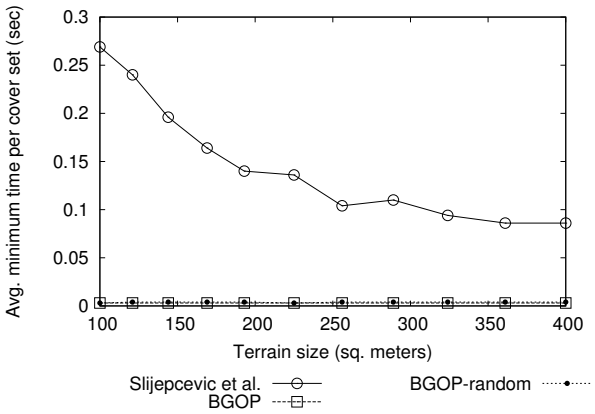
Όπως φαίνεται από τα αποτελέσματα των πειραμάτων η πυκνή δόμηση του διδιάστατου πεδίου δεν επηρέασε το χρόνο που χρειάστηκαν οι αλγόριθμοι B{GOP} και B{GOP}-random για να παράξουν μία ομάδα κάλυψης. Αντίθετα, ο αλγόριθμος Slijepcevic παρουσίασε αυξημένη καθυστέρηση κατά την παραγωγή ομάδων κάλυψης σε πεδία με λίγα τετραγωνικά μέτρα και πυκνή δόμηση.

Στο σχήμα 4.25 παρουσιάζεται γραφικά ο χρόνος που απαιτήθηκε από κάθε αλγόριθμο για την παραγωγή μίας ομάδας κάλυψης, συναρτήσει του μεγέθους του πεδίου τοποθέτησης των αισθητήρων και των στόχων.

4.6.2 Παραγωγή ομάδων κάλυψης με κοινούς αισθητήρες

Στη δεύτερη φάση της πειραματικής αξιολόγησης εξετάστηκαν οι αλγόριθμοι CCF και Cardei Greedy MSC [134] οι οποίοι επιτρέπουν την παραγωγή ομάδων κάλυψης με κοινά στοιχεία.

Όπως είχε αναφερθεί στις ενότητες 4.1 και 4.4.3, αν w είναι ο μέγιστος επιτρεπτός αριθμός ομάδων στις οποίες μπορεί να μετέχει ένας αισθητήρας, τότε οι παραγόμενες ομάδες C προσφέρουν $\frac{|C|}{w}$ χρόνο κάλυψης (ή αλλιώς χρόνο *ζωής*) στο δίκτυο. Ως μονάδα



Σχήμα 4.25: Μέσος χρόνος παραγωγής ομάδας κάλυψης σε δίκτυα με 100 αισθητήρες, 50 στόχους και μεταβλητή πυκνότητα δικτύου.

μέτρησης του χρόνου ζωής του δικτύου στα παρακάτω πειράματα θα θεωρηθεί ο τυπικός χρόνος ζωής ενός αισθητήρα.

Σενάριο Α' - Μεταβλητός αριθμός συμμετοχών

Η πρώτη τοπολογία στην οποία εξετάστηκαν οι αλγόριθμοι CCF και Cardei Greedy MSC ήταν ένα δίκτυο με 1000 αισθητήρες και 50 στόχους, όπου κάθε αισθητήρας μπορούσε να καλύψει το πολύ 25 στόχους. Το δίκτυο αυτό παράχθηκε με το λογισμικό `create_nodes_kd` και αποτελεί παράδειγμα ενός εξαιρετικά πυκνού δικτύου. Στόχος του πειράματος ήταν να μετρηθεί η αποδοτικότητα των παραπάνω αλγορίθμων (δηλ. ο αριθμός των παραγόμενων ομάδων κάλυψης) σε σχέση με τον πλήθος των συμμετοχών κάθε αισθητήρα στις παραγόμενες ομάδες. Σε κάθε εκτέλεση του αλγορίθμου CCF οι σταθερές α και β είχαν τις τιμές $\alpha = 0.35$ και $\beta = 0.02$.

Στον πίνακα 4.7 παρουσιάζονται τα αποτελέσματα του σχετικού πειράματος. Οι αλγόριθμοι CCF και Cardei Greedy MSC εξετάζονται κάθε φορά με διαφορετικό αριθμό μέγιστων συμμετοχών, ο οποίος κυμαίνεται μεταξύ 1 συμμετοχής και 20 συμμετοχών. Σαν σημείο αναφορά δίνεται επίσης η απόδοση του αλγορίθμου B{GOP} στο ίδιο σενάριο.

Τα αποτελέσματα δείχνουν ότι ο αλγόριθμος Cardei Greedy MSC απαιτεί μεγαλύτερο αριθμό συμμετοχών ανά αισθητήρα προκειμένου να παραγάγει τις ίδιες ομάδες κάλυψης με τον αλγόριθμο CCF. Ανάλογα με τον χρονοπρογραμματισμό των ομάδων κάλυψης, αυτή η επιβάρυνση μπορεί να μεταφραστεί και σε ενεργειακή επιφόρτηση για τους αισθητήρες καθώς καλούνται να αλλάξουν κατάσταση περισσότερες φορές. Επίσης, παρόλο που ο αλγόριθμος Cardei Greedy MSC είναι εν γένει ταχύτερος από τον CCF, εξαιτίας της παραπάνω μειωμένης απόδοσής του ως προς τις ομάδες κάλυψης απαιτεί τελικά περισσότερο χρόνο για την παραγωγή του ίδιου αριθμού ομάδων κάλυψης με τον CCF.

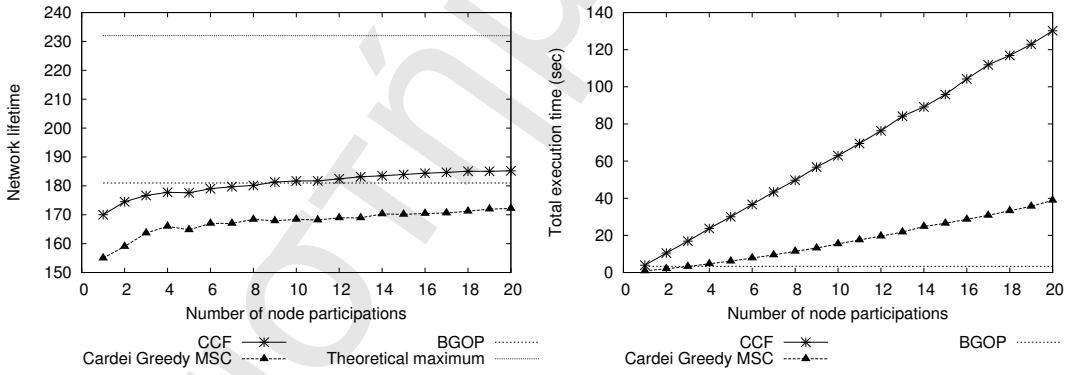
Ο αλγόριθμος CCF κατάφερε να ξεπεράσει τον παραγόμενο από τον B{GOP} αριθμό ομάδων όταν ο αριθμός συμμετοχών ήταν ίσος ή μεγαλύτερος των 9. Μάλιστα, με 20 συμμετοχές παρείχε περίπου 2% περισσότερες ομάδες κάλυψης από αυτές του αλγορίθμου B{GOP}. Η αντίστοιχη καθυστέρηση όμως στην εκτέλεση του αλγορίθμου CCF ήταν 40 φορές μεγαλύτερη αυτής του αλγορίθμου B{GOP}.

Τα αποτελέσματα του πειράματος παρουσιάζονται γραφικά στα σχήματα 4.26α΄ και 4.26β΄. Στο σχήμα 4.26α΄ απεικονίζεται μεταξύ άλλων και ο χρόνος ζωής δικτύου που αντιστοιχεί στο θεωρητικό μέγιστο αριθμό παραγόμενων ομάδων κάλυψης. Τέλος, στο σχήμα

	CCF		Cardei Greedy MSC		B{GOP}	
Συμ.[1]	ΧΔ[2]	ΧΕ[3]	ΧΔ[2]	ΧΕ[3]	ΧΔ[2]	ΧΕ[3]
1	170.000	3.935	155.000	0.902	181.000	3.315
2	174.500	10.489	159.000	2.016	-	-
3	176.670	16.887	163.670	3.241	-	-
4	177.750	23.679	166.000	4.697	-	-
5	177.600	29.999	164.800	6.133	-	-
6	179.000	36.632	167.000	7.829	-	-
7	179.710	43.372	167.000	9.475	-	-
8	180.120	49.719	168.380	11.423	-	-
9	181.330	56.768	167.890	13.184	-	-
10	181.700	62.910	168.400	15.428	-	-
11	181.730	69.512	168.180	17.523	-	-
12	182.420	76.304	168.920	19.608	-	-
13	183.150	84.184	168.920	21.792	-	-
14	183.500	89.164	170.290	24.772	-	-
15	183.930	95.832	170.130	26.490	-	-
16	184.380	104.229	170.440	28.630	-	-
17	184.650	111.854	170.590	30.780	-	-
18	185.060	116.928	171.220	33.256	-	-
19	185.050	122.891	171.950	35.651	-	-
20	185.200	130.207	172.200	38.905	-	-

[1] Δυνατότητα συμμετοχών ανά αισθητήρα σε ομάδες κάλυψης
[2] Χρόνος ζωής δικτύου (μονάδα μέτρησης: χρόνος ζωής αισθητήρα)
[3] Συνολικός χρόνος εκτέλεσης αλγορίθμου σε δευτερόλεπτα

Πίνακας 4.7: Αποτελέσματα προσομοίωσης με 1000 αισθητήρες και 50 στόχους.



(α′) (β′)

Σχήμα 4.26: Χρόνος ζωής δικτύου και χρόνος εκτέλεσης σε δίκτυο 1000 αισθητήρων και 50 στόχων.

4.26β′ απεικονίζεται μια γραμμική αύξηση στο χρόνο εκτέλεσης και των δύο αλγορίθμων όσο το πλήθος των συμμετοχών αυξάνεται στις παραγόμενες ομάδες κάλυψης.

Σενάριο Β' - Σταθερός αριθμός συμμετοχών

Το δεύτερο σενάριο πειραμάτων ασχολήθηκε με την σύγκριση των αλγορίθμων CCF και Cardei Greedy MSC όταν αυτοί χρησιμοποιούνται με προδιαγεγραμμένο αριθμό συμμετοχών σε πολλαπλές τοπολογίες. Το σενάριο αυτό εξετάζει την τυπική χρήση των παραπάνω

αλγορίθμων σε τοπολογίες με διαφορετικά χαρακτηριστικά (αραιά / πυκνά δομημένες κλπ.). Οι τοπολογίες αυτές παράχθηκαν με το λογισμικό `create_nodes_kd` και χρησιμοποιούν 100 αισθητήρες και μεταβλητό πλήθος στόχων που κυμαίνεται μεταξύ 10 και 100 στόχους. Κάθε αισθητήρας μπορεί να καλύψει τυχαίο αριθμό στόχων, με μέγιστο πλήθος το σύνολο των στόχων της τοπολογίας.

Το σενάριο αυτό εξετάζει την απόδοση των αλγορίθμων CCF και Cardei Greedy MSC όταν αυτοί μπορούν να αξιοποιήσουν ένα αισθητήρα δέκα φορές στις παραγόμενες ομάδες αλλά και μία μόνο φορά. Η θέση του περιορισμού αυτού (δηλ. της μοναδικής χρήσης ενός αισθητήρα σε μία από τις παραγόμενες ομάδες κάλυψης) επιτρέπει την περαιτέρω εξέταση των παραπάνω αλγορίθμων και σε σενάρια όπου απαιτείται η παραγωγή ομάδων κάλυψης δίχως κοινά στοιχεία. Για άλλη μια φορά θα χρησιμοποιηθούν τα αποτελέσματα του αλγορίθμου B{GOP} σαν σημείο αναφοράς για το χρόνο ζωής δικτύου που μπορεί να προσφέρει ένας αλγόριθμος στην υπό διερεύνηση τοπολογία αισθητήρων.

Οι εκτελέσεις του αλγορίθμου CCF όταν μπορούσε να χρησιμοποιηθεί κάθε αισθητήρας μία μόνο φορά έγιναν με τις σταθερές α, β να έχουν μηδενικές τιμές (δηλ. $\gamma = 1$). Αντίθετα, οι τιμές για τις αντίστοιχες σταθερές, όταν κάθε αισθητήρας μπορούσε να χρησιμοποιηθεί δέκα φορές, ήταν $\alpha = 0.35$ και $\beta = 0.02$.

	CCF				**Cardei Greedy MSC**				**BGOP**		
	10 Συμμετοχές		1 Συμμετοχή		10 Συμμετοχές		1 Συμμετοχή				
Στόχοι	ΧΔ[1]	ΧΕ[2]	ΧΔ[1]	ΧΕ[2]	ΧΔ[1]	ΧΕ[2]	ΧΔ[1]	ΧΕ[2]	ΧΔ[1]	ΧΕ[2]	Μ[3]
10	48,200	0,764	48,200	0,067	45,800	0,272	42,800	0,020	47,000	0,040	51,000
20	39,800	0,846	39,200	0,080	37,200	0,297	33,600	0,025	39,400	0,049	43,800
30	33,000	0,894	32,400	0,086	30,800	0,303	28,800	0,029	32,400	0,054	39,400
40	32,200	1,027	32,000	0,103	30,400	0,374	28,400	0,040	32,600	0,073	41,800
50	31,000	1,161	30,200	0,110	29,800	0,448	28,000	0,049	31,600	0,084	42,800
60	28,000	1,205	28,000	0,123	27,800	0,486	25,600	0,054	29,000	0,086	40,400
70	27,400	1,301	27,600	0,096	27,400	0,543	26,200	0,061	29,200	0,099	41,000
80	27,400	1,443	28,000	0,120	27,400	0,653	25,800	0,068	29,800	0,114	43,000
90	24,600	1,387	24,000	0,134	25,400	0,592	23,600	0,074	27,000	0,108	39,400
100	25,800	1,603	26,400	0,138	26,400	0,755	25,200	0,076	28,200	0,122	41,000

[1] Μέσος χρόνος ζωής δικτύου (μονάδα μέτρησης: χρόνος ζωής αισθητήρα)

[2] Μέσος συνολικός χρόνος εκτέλεσης αλγορίθμου σε δευτερόλεπτα

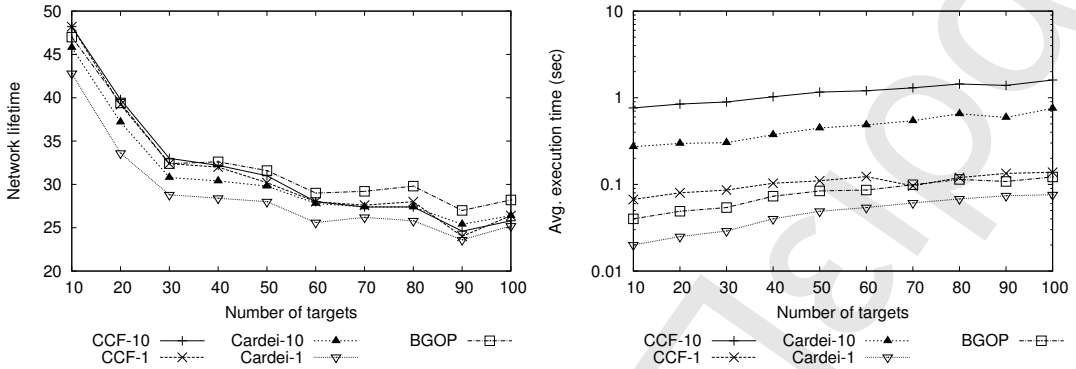
[3] Μέσος όρος θεωρητικού μεγίστου

Πίνακας 4.8: Αποτελέσματα προσομοίωσης με 100 αισθητήρες και μεταβλητό πλήθος στόχων.

Στον πίνακα 4.8 παρουσιάζονται τα αποτελέσματα των σχετικών πειραμάτων. Οι χρόνοι που φαίνονται στον πίνακα αποτελούν τους μέσους όρους των χρόνων (συνολικός χρόνος εκτέλεσης, χρόνος ζωής δικτύου, χρόνος ζωής δικτύου σύμφωνα με το Θεωρητικό Μέγιστο πλήθος παραγόμενων ομάδων κάλυψης) των προσομοιώσεων ανά περίπτωση τοπολογίας. Σύμφωνα, με τα στοιχεία του πίνακα ο αλγόριθμος των Cardei et al. [134] προσέφερε στα σενάρια 10 συμμετοχών καλύτερες λύσεις στο 20% των περιπτώσεων. Μάλιστα φαίνεται ότι οι επιδόσεις του γίνονται καλύτερες όσο το πλήθος των στόχων αυξάνει. Αντίθετα, οι επιδόσεις του ίδιου αλγορίθμου όταν ο κάθε αισθητήρας μπορούσε να χρησιμοποιηθεί μία φορά ήταν πολύ κατώτερες αυτών του αλγοριθμού B{GOP} και του αλγορίθμου CCF.

Στα σενάρια των 10 συμμετοχών, ο αλγόριθμος CCF παρήγαγε καλύτερες λύσεις από τον Cardei Greedy MSC στο 60% των περιπτώσεων. Αντίθετα στα σενάρια της μίας συμμετοχής, ήταν σε κάθε περίπτωση καλύτερος από τον Cardei Greedy MSC και μόλις στο 10% των περιπτώσεων καλύτερος από τον B{GOP}.

Εξετάζοντας συνολικά όλες τις τιμές του πίνακα διαπιστώνει κανείς ότι ο αλγόριθμος B{GOP} παράγει καλύτερα αποτελέσματα στο 70% των περιπτώσεων. Βέβαια θα πρέπει να επισημανθεί ότι στις σχετικές προσομοιώσεις χρησιμοποιήθηκε περιορισμένος αριθμός από συμμετοχές ανά αισθητήρα, όπως επίσης οι τιμές των σταθερών α και β του αλγορίθμου CCF δεν είχαν τύχει βελτιστοποίησης.



(α′) (β′)

Σχήμα 4.27: Μέσος χρόνος ζωής δικτύου και μέσος χρόνος εκτέλεσης αλγορίθμων σε δίκτυα με 100 αισθητήρες και μεταβλητό πλήθος στόχων.

Τα σχήματα 4.27α′ και 4.27β′ παρουσιάζουν γραφικά τα αποτελέσματα των προσομοιώσεων των αλγορίθμων στις τοπολογίες του παρόντος σεναρίου. Με την ονομασία "CCF-10" και "Cardei-10" σημειώνονται τα αποτελέσματα των αλγορίθμων CCF και Cardei Greedy MSC όταν αυτοί χρησιμοποιήθηκαν με ανώτατο όριο 10 συμμετοχών ανά αισθητήρα. Τέλος, με την ονομασία "CCF-1" και "Cardei-1" σημειώνονται τα αποτελέσματα των ίδιων αλγορίθμων όταν αυτοί χρησιμοποιήθηκαν με ανώτατο όριο μίας συμμετοχής ανά αισθητήρα.

Σενάριο Γ' - Βελτιστοποίηση παραμέτρων CCF

Στόχος του σεναρίου αυτού ήταν η εξέταση της απόδοσης του αλγορίθμου CCF όταν οι τιμές των σταθερών α και β είχαν τύχει βελτιστοποίησης μέσω του αλγορίθμου `split-triangle-search`. Συγκεκριμένα, χρησιμοποιήθηκαν 10 τοπολογίες από το προηγούμενο σενάριο οι οποίες είχαν σταθερό πλήθος αισθητήρων και μεταβλητό πλήθος στόχων. Στις τοπολογίες αυτές εφαρμόστηκαν οι αλγόριθμοι B{GOP}, Slijepcevic, Cardei Greedy MSC και CCF. Ο αλγόριθμος B{GOP}-random δεν εξετάστηκε εξαιτίας της μειωμένης απόδοσης που παρουσίασε σε προηγούμενα πειράματα (βλ. παράγραφο 4.6.1).

Για κάθε μία τοπολογία ο αλγόριθμος `split-triangle-search` δοκίμασε 20 υποδιαιρέσεις του χώρου λύσεων (*steps* $= 20$). Μαζί με το χρόνο ζωής των παραγόμενων δικτύων καταγράφηκε και ο συνολικός χρόνος εκτέλεσης της διαδικασίας βελτιστοποίησης.

Όπως και στα προηγούμενα σενάρια προσομοίωσης έτσι και εδώ, οι αλγόριθμοι παραγωγής ομάδων με κοινά στοιχεία εξετάστηκαν σε 2 φάσεις. Η πρώτη φάση ήταν αυτή της παραγωγής ομάδων δίχως κοινά στοιχεία ($w = 1$) ενώ η δεύτερη φάση ήταν αυτή της παραγωγής ομάδων με κοινά στοιχεία ($w = 10$). Επίσης, έγινε μια επιπλέον εκτέλεση του αλγορίθμου CCF ανά τοπολογία, δίχως βελτιστοποιημένες τιμές για τις σταθερές α και β. Για αυτές τις εκτελέσεις οι παραπάνω σταθερές έλαβαν την τιμή μηδέν (δηλ. $\gamma = 1$) και παράχθηκαν ομάδες δίχως κοινούς αισθητήρες.

Στον πίνακα 4.9 παρουσιάζονται τα αποτελέσματα των σχετικών πειραμάτων. Ο αλ-

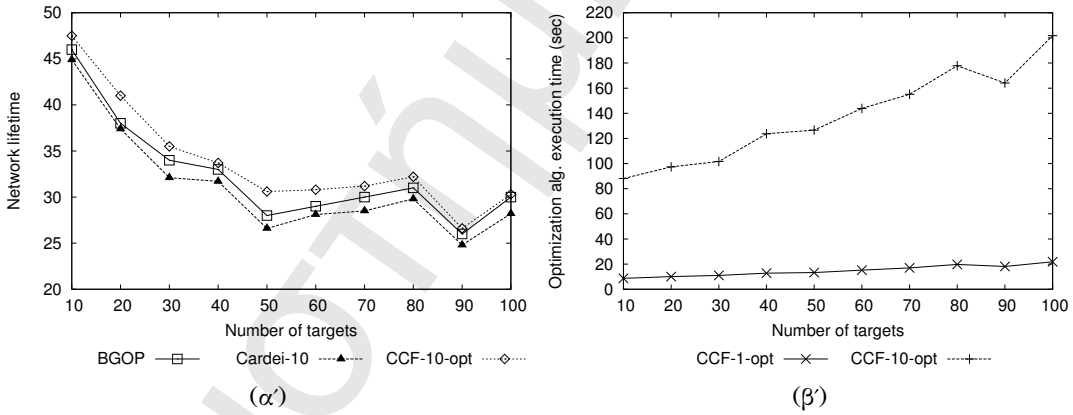
	B{GOP}	Slijepcevic	Cardei Greedy MSC		CCF				
	1 Συμ.	1 Συμ.	1 Συμ.	10 Συμ.	1 Συμ. ($\alpha=0,\beta=0$)	1 Συμ. ($\alpha,\beta=$ βελτ.)		10 Συμ. ($\alpha,\beta=$ βελτ.)	
Στόχοι	ΧΔ[1]	ΧΔ[1]	ΧΔ[1]	ΧΔ[1]	ΧΔ[1]	ΧΔ[1]	ΧΕ[2]	ΧΔ[1]	ΧΕ[2]
10	46	43	41	44,90	47	47	8,69	47,5	88,10
20	38	35	34	37,40	39	39	10,07	41,0	97,37
30	34	30	29	32,10	33	35	10,99	35,5	101,62
40	33	28	28	31,70	31	32	12,79	33,7	123,68
50	28	26	26	26,60	28	29	13,30	30,6	126,62
60	29	26	27	28,10	28	28	15,23	30,8	143,86
70	30	27	27	28,50	29	29	17,00	31,2	155,13
80	31	29	28	29,80	30	31	19,76	32,2	177,78
90	26	23	23	24,80	23	25	18,16	26,6	164,11
100	30	27	27	28,20	28	28	21,85	30,3	201,77

[1] Χρόνος ζωής δικτύου (μονάδα μέτρησης: χρόνος ζωής αισθητήρα)

[2] Χρόνος εκτέλεσης αλγορίθμου βελτιστοποίησης σε δευτερόλεπτα

Πίνακας 4.9: Αποτελέσματα προσομοίωσης με 100 αισθητήρες και μεταβλητό πλήθος στόχων.

γόριθμος CCF με βελτιστοποιημένες τιμές για τις σταθερές α και β και με ανώτατο όριο τις 10 συμμετοχές ανά αισθητήρα στις παραγόμενες ομάδες κάλυψης προσφέρει σε όλες τις περιπτώσεις την καλύτερη διάρκεια ζωής για το δίκτυο αισθητήρων. Δεύτερος σε κατάταξη καλύτερων λύσεων (ως προς την προσφερόμενη διάρκεια ζωής του δικτύου) έρχεται ο αλγόριθμος B{GOP} και τρίτος ο αλγόριθμος CCF για ομάδες κάλυψης χωρίς κοινά στοιχεία με βελτιστοποίημένες τιμές για τις σταθερές α και β.



(α′) (β′)

Σχήμα 4.28: Χρόνος ζωής δικτύου και χρόνος εκτέλεσης αλγορίθμων βελτιστοποίησης σε δίκτυα με 100 αισθητήρες και μεταβλητό πλήθος στόχων.

Στο σχήμα 4.28α′ παρουσιάζονται τα αποτελέσματα ως προς το χρόνο ζωής του δικτύου για τους αλγόριθμους B{GOP}, Cardei Greedy MSC και CCF με όριο 10 συμμετοχές ανά αισθητήρα. Αντίστοιχα στο σχήμα 4.28β′ παρουσιάζονται οι χρόνοι εκτέλεσης των αλγορίθμων βελτιστοποίησης για τον αλγόριθμο CCF όταν χρησιμοποιήθηκε με όριο συμμετοχών τον 1 (CCF-1-opt) και τους 10 (CCF-10-opt) αισθητήρες. Από τα αποτελέσματα φαίνεται ότι η διάρκεια εκτέλεσης του αλγορίθμου βελτιστοποίησης αυξάνεται με ταχύτερο ρυθμό όταν αυξάνεται το όριο συμμετοχών.

Από το πείραμα αυτό αλλά και από τα προηγούμενα πειράματα μπορεί κανείς να βγάλει το συμπέρασμα ότι ο αλγόριθμος B{GOP} παράγει στις περισσότερες περιπτώσεις

πολύ καλές λύσεις στο πρόβλημα της κάλυψης στόχων. Σε τοπολογίες με ιδιαίτερα πυκνή δόμηση, οι λύσεις αυτές παράγονται γρήγορα ακόμη και όταν ο αλγόριθμος επεξεργάζεται μεγάλο πλήθος αισθητήρων και στόχων.

Τέλος, σε περιπτώσεις όπου απαιτείται η μεγιστοποίηση του χρόνου ζωής ενός δικτύου, τα πειράματα δείχνουν ότι ενδείκνυται η χρήση του βελτιστοποιημένου αλγορίθμου CCF για την παραγωγή ομάδων κάλυψης με κοινούς αισθητήρες. Οι ομάδες που θα παραχθούν θα απαιτούν πολύ λιγότερες συμμετοχές από τους αισθητήρες σε σχέση με αυτές που θα παράγονταν από τον αλγόριθμο των Cardei et al [134] και έτσι εξοικονομείται σημαντική ποσότητα ενέργειας.

4.7 Ποιοτική αξιολόγηση

Οι ευρετικοί αλγόριθμοι που εξετάστηκαν στο κεφάλαιο αυτό προσφέρουν ικανές λύσεις για το πρόβλημα της κάλυψης στόχων σε δίκτυα αισθητήρων, το οποίο σύμφωνα με την εργασία [126] είναι ένα *NP-Complete* πρόβλημα. Είναι κεντρικοποιημένοι και έτσι δεν απαιτούν από τους αισθητήρες να έχουν πλήρη εποπτεία όλου του δικτύου. Επίσης, εκτελούνται στο σταθμό βάσης οπότε δεν επιβαρύνουν τη λειτουργία των αισθητήρων.

Αρχικά εξετάστηκε το πρόβλημα κάλυψης και η σχετική βιβλιογραφία προκειμένου να αναδειχθούν τα βασικά χαρακτηριστικά που διέπουν οποιονδήποτε ευρετικό αλγόριθμο παραγωγής ομάδων κάλυψης στόχων ή περιοχών. Από τη μοντελοποίηση αυτή προέκυψε ο αλγόριθμος B{GOP} ο οποίος, όπως διαπιστώθηκε και κατά την πειραματική αξιολόγηση, προσφέρει εξαιρετικά αποδοτικές λύσεις στο πρόβλημα της παραγωγής ομάδων κάλυψης δίχως κοινά στοιχεία. Ο B{GOP} έχει τη δυνατότητα να αποφύγει με δυναμικό τρόπο τόσο τις διπλοκαλύψεις στόχων όσο και την επιλογή κρίσιμων αισθητήρων. Η δυνατότητα αυτή του επιτρέπει να παράγει περισσότερες ομάδες κάλυψης από αυτές του αντίστοιχου αλγορίθμου των Slijepcevic et al [129]. Επίσης, ο χρόνος εκτέλεσης αυτού επηρεάζεται λιγότερο από μεταβολές στον αριθμό των αισθητήρων και των στόχων μιας τοπολογίας, απ'ότι ο αντίστοιχος χρόνος του αλγορίθμου Slijepcevic.

Ο B{GOP}-random αποτελεί μια παραλαγή του αλγορίθμου B{GOP}, όπου η επιλογή του αισθητήρα γίνεται με τυχαίο τρόπο στον οποίο συμβάλλει η αντικειμενική συνάρτηση του αλγορίθμου B{GOP}. Η πειραματική αξιολόγηση έδειξε ότι ο αλγόριθμος αυτός παρήγαγε επί το πλείστον λιγότερες ομάδες κάλυψης από τους αλγορίθμους της βιβλιογραφίας αλλά και από τους προτεινόμενους αλγορίθμους. Επίσης, η πολυπλοκότητά του ήταν μεγαλύτερη αυτής του B{GOP}.

Ο αλγόριθμος CCF αποτελεί μια μετεξέλιξη του αλγορίθμου B{GOP}. Παρέχει τη δυνατότητα επέκτασης του παραγόμενου αριθμού ομάδων μέσω της επαναχρησιμοποίησης αισθητήρων και αξιοποιεί μια νέα αντικειμενική συνάρτηση για τη βαθμολόγηση όλων των αισθητήρων. Τα αποτελέσματα της πειραματικής αξιολόγησης έδειξαν ότι μπορεί να παραγάγει περισσότερες ομάδες κάλυψης από τον αντίστοιχο αλγόριθμο των Cardei et al [134], χρησιμοποιώντας κάθε αισθητήρα λίγοτερες φορές. Έτσι παράγονται πιο γρήγορα τα σχετικά αποτελέσματα και εξοικονομείται περισσότερη ενέργεια.

Για την παραγωγή ακόμη περισσότερων ομάδων κάλυψης, μπορεί να χρησιμοποιηθεί η μέθοδος βελτιστοποίησης `split-triangle-search` στις παραμέτρους α και β του αλγορίθμου CCF. Τα σχετικά πειράματα έδειξαν ότι μέσω της βελτιστοποίησης ο αλγόριθμος CCF κατάφερε να παραγάγει περισσότερες ομάδες κάλυψης από κάθε άλλο αλγόριθμο που συμμετείχε στα πειράματα αυτά.

Οι προσομοιώσεις που έγιναν έλεγξαν τους αλγόριθμους με πλήθος σεναρίων. Από τους ελέγχους αυτούς διαπιστώθηκε ότι ο αλγόριθμος Slijepcevic παράγει ελαττωμένο αριθμό

ομάδων κάλυψης σε πυκνά δομημένα δίκτυα εξαιτίας του αυστηρού τρόπου επιλογής των αισθητήρων που χρησιμοποιεί. Ομοίως, ο αλγόριθμος Cardei Greedy MSC χρειάζεται σε τέτοιου είδους τοπολογίες όλο και περισσότερους κόμβους προκειμένου να καλύψει το σύνολο των στόχων του δικτύου.

Η συνολική εξέταση των πειραματικών δεδομένων έδειξε ότι στις περισσότερες περιπτώσεις ο αλγόριθμος B{GOP} παρείχε λύσεις πολύ καλής ποιότητας στο χρήστη. Σε περιπτώσεις όμως όπου ο χρήστης θα επιθυμούσε να εξάγει το μέγιστο δυνατό αριθμό από ομάδες, τότε θα έπρεπε να επιλέξει τη βελτιστοποιημένη έκδοση του αλγορίθμου CCF. Βέβαια η επιλογή αυτή έρχεται με ένα αυξημένο κόστος στο χρόνο εκτέλεσης του αλγορίθμου.

Για την υλοποίηση του συνόλου των πειραμάτων χρησιμοποιήθηκε το ειδικό λογισμικό προσομοίωσης αλγορίθμων κάλυψης που παρουσιάστηκε στην ενότητα 4.5. Το λογισμικό αυτό επέτρεψε την ταχεία εξέταση των αλγορίθμων σε τοπολογίες δικτύων με διαφορετικά χαρακτηριστικά (τοπολογίες 2/3/k-διαστάσεων, πυκνή/αραιή δόμηση κλπ.). Επίσης, επέτρεψε την παραγωγή ομάδων κάλυψης με συγκεκριμένες ιδιότητες όπως η εγγυημένη συνδεσιμότητα όλων των αισθητήρων με τη βάση. Ως έργο ανοιχτού λογισμικού, το περιβάλλον προσομοίωσης διατίθεται ελεύθερα από ιστοσελίδα του πανεπιστημίου και μπορεί να χρησιμοποιηθεί για την εξέταση οποιουδήποτε αλγορίθμου παραγωγής ομάδων κάλυψης. Χαρακτηριστικά θα μπορούσε κανείς να σημειώσει τη χρήση αυτού στην εργασία [139] όπου μελετούνται αλγόριθμοι παραγωγής ομάδων μερικής κάλυψης.

Τέλος, οι επιδόσεις των αλγορίθμων B{GOP} και CCF καθώς και ο χρόνος εκτέλεσης αυτών τους καθιστά κατάλληλους για χρήση και σε άλλα προβλήματα, όπως αυτό του χρονοπρογραμματισμού διεργασιών σε μεγάλες διαδικτυακές υποδομές.

Κεφάλαιο 5

Ασφάλεια σε Προγραμματιζόμενες Υποδομές

Η δυνατότητα επαναπρογραμματισμού των δικτυακών συσκευών μπορεί να θέσει σε κίνδυνο την ασφάλεια του προγραμματιζόμενου δικτύου και να μειώσει την εμπιστοσύνη των χρηστών αυτού προς τις παρεχόμενες υπηρεσίες. Σε συστήματα βασισμένα σε Ανοιχτά Πρότυπα (open systems architectures) το λογισμικό που εκτελείται στους προγραμματιζόμενους κόμβους δεν ανήκει απαραίτητα στις διαχειριστικές αρχές που επιβλέπουν τους κόμβους και έτσι θα πρέπει να συνοδεύεται από μια σειρά εγγυήσεων προτού τεθεί σε λειτουργία. Ομοίως, οι συγγραφείς υπηρεσιών θα πρέπει να λάβουν κάποιες εγγυήσεις προτού επιτρέψουν στο λογισμικό τους να εκτελεστεί σε προγραμματιζόμενες πλατφόρμες καθώς υπάρχει κίνδυνος να αλλοιωθεί η παρεχόμενη υπηρεσία από το περιβάλλον εκτέλεσης (host). Τέτοια θέματα ασφάλειας που αφορούν την εκτέλεση μη έμπιστου λογισμικού σε δικτυακές πλατφόρμες αποτελούν αντικείμενο εξέτασης τόσο στη βιβλιογραφία των Ενεργών Δικτύων [140] όσο και στη βιβλιογραφία των *Κινητών Πρακτόρων* (Mobile Agents) [141] και των *Αρχιτεκτονικών Πλέγματος* (Grid Computing) [142].

Συγκεκριμένα, στην εργασία των Jansen και Karygiannis [141] περιγράφονται τέσσερις μορφές προστασίας που θα πρέπει να παρέχονται από συστήματα που επιτρέπουν την εκτέλεση «ξένου» λογισμικού:

- προστασία της πλατφόρμας εκτέλεσης από το λογισμικό,
- προστασία του λογισμικού από την πλατφόρμα εκτέλεσης,
- προστασία μεταξύ των εφαρμογών που συνυπάρχουν σε μια πλατφόρμα εκτέλεσης, και
- προστασία της πλατφόρμας εκτέλεσης και του λογισμικού από εξωγενείς παράγοντες.

Στην παραπάνω λίστα θα μπορούσε κανείς να προσθέσει και την προστασία του χρήστη από κακόβουλους παρόχους υπηρεσιών ή παρόχους που προσφέρουν υπηρεσίες χαμηλής ποιότητας. Πρέπει να τονιστεί εδώ ότι σε ένα προγραμματιζόμενο δίκτυο, η έννοια του παρόχου μιας υπηρεσίας περιλαμβάνει τόσο τις πλατφόρμες εκτέλεσης, όσο και τις οντότητες λογισμικού που συνεργάστηκαν στο πλαίσιο του δικτύου αυτού για να εξυπηρετήσουν το χρήστη.

Στις υποενότητες που ακολουθούν θα παρουσιαστούν μερικές τεχνικές από τη βιβλιογραφία που στοχεύουν στην παροχή των παραπάνω μορφών προστασίας.

5.1 Μέθοδοι προστασίας της πλατφόρμας εκτέλεσης από το λογισμικό

Ο έλεγχος της υπογραφής σε ψηφιακά υπογεγραμμένο λογισμικό [143] μπορεί να εγγυηθεί την ακεραιότητα του προς εκτέλεση λογισμικού καθώς και την προηγούμενη επεξεργασία αυτού (π.χ. δημιουργία, διαμόρφωση κλπ.) από κάποια έμπιστη οντότητα. Σε καμμία περίπτωση δεν εγγυάται ότι το λογισμικό αυτό θα λειτουργήσει μέσα σε κάποια «νόμιμα» πλαίσια και ότι δε θα προξενήσει προβλήματα στην πλατφόρμα εκτέλεσης και κατ'επέκταση στο δίκτυο και στους χρήστες αυτού. Μάλιστα, ο έλεγχος μιας ψηφιακής υπογραφής προϋποθέτει την ύπαρξη μιας αρχής πιστοποίησης η οποία θα πιστοποιήσει ότι το δημόσιο κλειδί με το οποίο έχει υπογραφεί το λογισμικό είναι έγκυρο και ότι αντιστοιχεί σε έμπιστο πάροχο λογισμικού. Αυτή η εξάρτηση από τρίτες έμπιστες οντότητες είναι ασύμβατη με τη φύση μιας αρχιτεκτονικής ανοιχτών προτύπων, όπου ο οποιοσδήποτε που ακολουθεί το συμφωνημένο πρωτόκολλο λειτουργίας θεωρείται εν δυνάμει πάροχος υπηρεσιών. Για να προστατευθεί λοιπόν η πλατφόρμα εκτέλεσης θα πρέπει να βρεθούν (αποδοτικές) μέθοδοι ώστε η ίδια η πλατφόρμα να είναι σε θέση να εξετάσει την ποιότητα του κώδικα που θα εκτελέσει.

Μια τέτοια μέθοδος είναι η παροχή κώδικα με *συνημμένες αποδείξεις* (proof carrying code) [144],[145]. Στην περίπτωση αυτή ο συγγραφέας του λογισμικού περιγράφει μια σειρά αποδείξεων σε μια γλώσσα περιγραφής λογικών προτάσεων (όπως είναι η LF [146], μια typed λ-calculus γλώσσα). Το περιβάλλον εκτέλεσης θα χρησιμοποιήσει τις αποδείξεις αυτές για να επαληθεύσει την ύπαρξη μιας σειράς επιθυμητών χαρακτηριστικών στο προς εκτέλεση λογισμικό. Δυστυχώς, η εγγύηση για την ορθή λειτουργία του λογισμικού δεν θα προέλθει από την απόδειξη καθαυτή αλλά από το γεγονός ότι το λογισμικό που θα εξετάσει την απόδειξη δε θα εντοπίσει κάποιο λάθος σε αυτή.

Μια άλλη μέθοδος είναι η χρήση μιας περιορισμένης διεπαφής μεταξύ λογισμικού και πλατφόρμας εκτέλεσης, όπου οι λειτουργίες που θα επιτρέπονται μέσω της διεπαφής θα αποτελούν στην ουσία και την πολιτική εκτέλεσης «ξένου» λογισμικού. Προκειμένου να περιοριστούν οι δυνατότητες του λογισμικού, θα πρέπει οι λειτουργίες που παρέχονται από την πλατφόρμα να είναι πρωτεύουσας σημασίας για την εκτέλεση του λογισμικού. Ένα τέτοιο μοντέλο εφαρμόζεται στην εικονική μηχανή (virtual machine) της γλώσσας προγραμματισμού Java, όπου η δικτυακή επικοινωνία μιας εφαρμογής ή η πρόσβασή της σε μέσα αποθήκευσης μπορεί να περιοριστεί μέσω μιας προγραμματιστικής διεπαφής (βλ. `SecurityManager`) [147].

Η πλατφόρμα εκτέλεσης μπορεί επίσης να «καταγράψει» τις ενέργειες του λογισμικού (tracing) και να εξαγάγει από το αρχείο αυτό ποιοτικά συμπεράσματα για τη λειτουργία του. Για παράδειγμα, η πλατφόρμα RCANE [148] για ενεργά δίκτυα χρησιμοποιεί μια διεπαφή σαν αυτή που περιγράφηκε παραπάνω για να δώσει στο «ξένο» λογισμικό πρόσβαση σε τοπικούς πόρους και υπηρεσίες. Η διεπαφή μπορεί επίσης να καταγράψει την ποσότητα των πόρων (μνήμη, κύκλοι του επεξεργαστή κ.α.) που εκμεταλλεύεται το «ξένο» λογισμικό και να θέσει κάποια όρια στην εκμετάλλευση αυτών. Οι περιορισμοί αυτοί θα δράσουν τελικά ως μέτρα ασφάλειας σε περιπτώσεις επιθέσεων τύπου *Άρνησης Εξυπηρέτησης* (Denial of Service) προς την πλατφόρμα εκτέλεσης.

Η εξαγωγή συμπερασμάτων από την εκτέλεση «ξένου» λογισμικού μπορεί να επεκταθεί και στην παρακολούθηση των δραστηριοτήτων του σε περισσότερες από μία πλατφόρμες. Συγκεκριμένα, στην τεχνική *καταγραφής μονοπατιών* (path histories) [149] κάθε πλατφόρμα στην οποία εκτελείται το λογισμικό τοποθετεί σε αυτό μια υπογραφή ώστε οι επόμενες πλατφόρμες να αξιολογήσουν το λογισμικό με βάση την εμπιστοσύνη τους προς τις προη-

γούμενες πλατφόρμες που εκτέλεσαν το λογισμικό αυτό. Έτσι, η τελική αξιολόγηση του λογισμικού επηρεάζεται από το δίκτυο εμπιστοσύνης που έχουν δημιουργήσει οι πλατφόρμες εκτέλεσης μεταξύ τους.

5.2 Μέθοδοι προστασίας του λογισμικού από την πλατφόρμα εκτέλεσης

Η πλατφόρμα εκτέλεσης έχει τον απόλυτο έλεγχο της ροής των προγραμμάτων που εκτελούνται σε αυτή. Δυστυχώς, ο έλεγχος αυτός της δίνει τη δυνατότητα να αλλοιώσει την εκτέλεση μιας εφαρμογής, να αλλοιώσει τα δεδομένα που επεξεργάζεται ή παράγει μια εφαρμογή, καθώς και να υποκλέψει ευαίσθητα δεδομένα που μπορεί να διαχειρίζεται η εκάστοτε εφαρμογή. Για να προστατευθεί το λογισμικό από ένα τέτοιο κακόβουλο περιβάλλον εκτέλεσης και τελικά να προστατευθεί το επίπεδο της παρεχόμενης υπηρεσίας και ο χρήστης αυτής απαιτούνται κάποιες εγγυήσεις από το περιβάλλον εκτέλεσης.

Η *Tamper-proof environment* (TPE) [150] αποτελεί μια πλατφόρμα εκτέλεσης η οποία αποτελεί μέρος ενός ευρύτερου περιβάλλοντος εκτέλεσης. Η πλατφόρμα αυτή είναι υλοποιημένη σε υλικό (hardware) και οι λειτουργίες της πιστοποιούνται από τον κατασκευαστή της. Το λογισμικό που θα εκτελέσει μια TPE δίνεται σε αυτή κρυπτογραφημένο με το δημόσιο κλειδί που της αντιστοιχεί. Σύμφωνα με τη μέθοδο της ασύμμετρης κρυπτογράφησης, η TPE θα αποκρυπτογραφήσει το λογισμικό χρησιμοποιώντας ένα ιδιωτικό κλειδί το οποίο έχει αποθηκευμένο τοπικά στο υλικό και το οποίο δεν είναι γνωστό στο ευρύτερο περιβάλλον εκτέλεσης. Μαζί με το λογισμικό της εφαρμογής, ο πάροχος υπηρεσιών μπορεί να δώσει και συνοδευτικά στοιχεία στην TPE, όπως παραμέτρους για την εκτέλεση αυτής ή το δημόσιο κλειδί με το οποίο θα κρυπτογραφηθεί η έξοδος της εφαρμογής. Επίσης, τα δεδομένα που παράγει μια TPE μπορούν να υπογραφούν έτσι ώστε να γίνεται δυνατή η πιστοποίηση της αυθεντικότητας αυτών από απομακρυσμένες οντότητες λογισμικού. Στο μοντέλο εμπιστοσύνης που εισάγουν οι TPE, ο συγγραφέας λογισμικού δε συνάπτει πλέον σχέσεις με τους κόμβους όπου εκτελείται το λογισμικό, αλλά με τους κατασκευαστές των TPE. Η ευελιξία που παρέχει το μοντέλο αυτό περιορίζεται σε κάποιο βαθμό από τις μειωμένες δυνατότητες της συσκευής και το κόστος αναβάθμισης αυτής.

Σε περιβάλλοντα εκτέλεσης όπου δεν υπάρχει εγκατεστημένη κάποια TPE, υπάρχει ο κίνδυνος υποκλοπής ευαίσθητων δεδομένων (ή διαδικασιών) από τη μνήμη της υπό εκτέλεση εφαρμογής. Για να κρύψει τμήματα ενός αλγορίθμου ή δεδομένα από μια τέτοια πλατφόρμα εκτέλεσης, ο συγγραφέας λογισμικού μπορεί να χρησιμοποιήσει *κρυπτογραφημένες συναρτήσεις* [151]. Συγκεκριμένα, έστω f μια συνάρτηση η οποία υλοποιείται από το πρόγραμμα $P(f)$ και η οποία υπολογίζει το αποτέλεσμα $f(x)$ για την είσοδο δεδομένων x. Έστω $E(f)$ μια κρυπτογραφημένη (διαφορετική) μορφή της f η οποία δέχεται ως όρισμα το x και υπολογίζει το $f(x)$ (δηλ. $E(f(x)) = f(x)$). Χρησιμοποιώντας την $E(f)$, ο συγγραφέας λογισμικού μπορεί να αποστείλει το πρόγραμμα $P(E(f))$ στην πλατφόρμα εκτέλεσης και αυτό θα υπολογίσει το σωστό αποτέλεσμα για κάθε είσοδο x, χωρίς να φανερωθεί κάποιο μέρος της αρχικής υλοποίησης $P(f)$. Στην εργασία των Κοτζανικολάου et al. [152], παρουσιάζεται μια συνάρτηση $E(f)$ η οποία είναι ομομορφική της f και η υλοποίηση της οποίας κάνει δυνατή την παραγωγή μιας ψηφιακής υπογραφής τύπου RSA χωρίς να αποκαλυφθεί το ιδιωτικό κλειδί που χρησιμοποιήθηκε σε αυτή.

Τέλος, για υπηρεσίες που εκτελούνται για μικρό χρονικό διάστημα και για τις οποίες απαιτείται μυστικότητα ως προς τη δομή του αλγορίθμου τους, μπορεί να χρησιμοποιηθεί μια τεχνική *συσκότισης* (obfuscation) του αρχικού αλγορίθμου, όπως προτείνεται στην εργασία [153] για το λογισμικό κινητών πρακτόρων. Οι τεχνικές συσκότισης αλγορίθμων

έχουν ως στόχο τη διαστρέβλωση του κώδικα μιας εφαρμογής με τέτοιο τρόπο ώστε να γίνεται δύσκολη η αναγνώριση του αρχικού αλγορίθμου κατά την στατική (static) ή δυναμική (run-time) ανάλυση αυτού. Για την απόκρυψη του αλγορίθμου βραχυπρόθεσμων εφαρμογών προγραμματιζόμενων δικτύων απαιτούνται τεχνικές συσκότισης που θα επιτρέψουν στο λογισμικό να ολοκληρώσει τη λειτουργία του προτού η πλατφόρμα εκτέλεσης αντιληφθεί το είδος του αλγορίθμου που εκτελέστηκε.

5.3 Αμοιβαία προστασία των εφαρμογών μιας πλατφόρμας εκτέλεσης

Η πλατφόρμα εκτέλεσης διαχειρίζεται ένα σύνολο κοινών πόρων για λογαριασμό των διεργασιών που εκτελούνται στον προγραμματιζόμενο κόμβο και οι οποίες μπορεί να ανήκουν σε διαφορετικές υπηρεσίες. Προκειμένου να επιτευχθεί η αρμονική «συμβίωση» αυτών των διεργασιών μέσα στην ίδια πλατφόρμα εκτέλεσης, θα πρέπει να προστατευθεί τόσο η λειτουργία αυτών όσο και το δικαίωμά τους για πρόσβαση στους κοινούς πόρους. Αυτό το μοντέλο λειτουργίας θυμίζει το αντίστοιχο που εφαρμόζεται στα λειτουργικά συστήματα Η/Υ που επιτρέπουν την εκτέλεση πολλαπλών διεργασιών σε ένα υπολογιστικό περιβάλλον με κοινό επεξεργαστή, μνήμη και αποθηκευτικό χώρο. Συγκεκριμένα, το περιβάλλον (sandbox) μέσα στο οποίο εκτελούνται οι εφαρμογές ενός προγραμματιζόμενου κόμβου δρα όπως ο πυρήνας ενός λειτουργικού συστήματος Η/Υ. Διαχειρίζεται δηλαδή την πρόσβαση στους κοινούς πόρους και παρέχει μια διεπαφή ώστε οι διεργασίες να μπορούν να επικοινωνήσουν μεταξύ τους. Μάλιστα, πολλές φορές οι πλατφόρμες εκτέλεσης υλοποιούνται ως τροποποιήσεις πυρήνων λειτουργικών συστημάτων ώστε οι εφαρμογές τους να μπορούν να αξιοποιήσουν τα ήδη υπάρχοντα μέτρα ασφάλειας (και δυνατότητες επικοινωνίας) που παρέχονται από αυτά τα λειτουργικά συστήματα (βλ. [154], [155]).

Η βασική μέθοδος προστασίας της εκτέλεσης μιας εφαρμογής είναι η απόδοση ξεχωριστών σελίδων μνήμης ανά διεργασία και η προστασία των σελίδων αυτών με κάποιο μηχανισμό ο οποίος ελέγχεται είτε από το λειτουργικό σύστημα είτε από την εικονική μηχανή του περιβάλλοντος εκτέλεσης. Σε περιβάλλοντα εκτέλεσης όπου δεν υπάρχει η δυνατότητα παραχώρησης ξεχωριστών σελίδων μνήμης ανά διεργασία και όπου κάθε εφαρμογή έχει άμεση πρόσβαση στην μνήμη κάθε άλλης εφαρμογής (π.χ. εφαρμογές που τρέχουν σε επίπεδο πυρήνα λειτουργικού συστήματος [156]), εισάγονται κάποιοι περιορισμοί στον τύπο του λογισμικού που μπορεί να εκτελεστεί. Συγκεκριμένα, γλώσσες όπως η PLAN [12], επιτρέπουν τη δημιουργία λογισμικού το οποίο δεν μπορεί να αναφερθεί σε συγκεκριμένες θέσεις μνήμης (pointer-safe), δίνοντας έτσι τη δυνατότητα σε περιβάλλοντα όπως τα παραπάνω να εκτελέσουν περισσότερες από μία διεργασίες, χωρίς να υπάρχει κίνδυνος αλλοίωσης της μνήμης των εφαρμογών κατά τη διάρκεια εκτέλεσής τους.

Όσον αφορά στην κατανάλωση πόρων, αρχιτεκτονικές όπως αυτή του RCANE [148] μπορούν να μειώσουν την πιθανότητα να μονοπωλήσει κάποια συγκεκριμένη εφαρμογή τους πόρους ενός προγραμματιζόμενου συστήματος. Όπως προαναφέρθηκε στην παράγραφο 5.1, οι αρχιτεκτονικές αυτές θέτουν όρια χρήσης/κατανάλωσης στους διάφορους πόρους, τα οποία οι εφαρμογές δεν επιτρέπεται να υπερβούν. Έτσι εξασφαλίζεται η «δίκαιη» κατανομή των πόρων στις διάφορες εφαρμογές και αποφεύγονται επιθέσεις που έχουν ως στόχο την εξάντληση των πόρων (resource starvation) και τη δημιουργία συνθηκών Άρνησης Εξυπηρέτησης. Η πιο απλή μορφή ενός τέτοιου συστήματος κατανομής πόρων είναι ο χρονοπρογραμματιστής της πλατφόρμας εκτέλεσης. Ο χρονοπρογραμματιστής εναλλάσσει ανά τακτά χρονικά διαστήματα τις διεργασίες που εκτελούνται στον επεξεργαστή (ή στους επεξεργαστές) της πλατφόρμας εκτέλεσης, παρέχοντας ισότιμη κατανομή χρόνου ε-

πεξεργασίας (processing time) σε κάθε διεργασία. Στην περίπτωση των Ενεργών Δικτύων όπου οι εφαρμογές συχνά επεξεργάζονται δικτυακή κίνηση, απαιτείται μια ενιαία πολιτική χρονοπρογραμματισμού που λαμβάνει υπόψιν της τόσο το χρόνο επεξεργασίας που δαπανήθηκε όσο και την ποσότητα της δικτυακής κίνησης που παράχθηκε. Οι Ramachandran et al. προτείνουν έναν τέτοιο χρονοπρογραμματιστή στο [157].

Επειδή η πλατφόρμα εκτέλεσης αποτελεί η ίδια ένα κοινό πόρο που θα αξιοποιηθεί από πλήθος δικτυακών υπηρεσιών, απαιτείται ο ανάλογος χρονοπρογραμματισμός και για το συνολικό χρόνο εκτέλεσης των εφαρμογών. Για το λόγο αυτό δημιουργήθηκαν ειδικές γλώσσες προγραμματισμού οι οποίες επιτρέπουν την πρόβλεψη του συνολικού χρόνου εκτέλεσης μιας εφαρμογής. Μια τέτοια γλώσσα είναι η SNAP [158], η οποία δεν υποστηρίζει βρόχους (loops) καθώς δε διαθέτει τελεστή για την επαναφορά του προγράμματος σε κάποιο προηγούμενο στάδιο εκτέλεσης (no backward jumps). Στην SNAP ο συνολικός χρόνος εκτέλεσης ενός προγράμματος είναι ανάλογος του πλήθους των εντολών αυτού του προγράμματος. Το περιβάλλον εκτέλεσης της SNAP μπορεί να μετρήσει ανά πάσα στιγμή τον αριθμό των εντολών που έχουν εκτελεστεί από μια εφαρμογή και συνεπώς το χρόνο που απαιτείται μέχρι να ολοκληρωθεί η εκτέλεση της εφαρμογής. Χρησιμοποιώντας αυτή την πληροφορία το περιβάλλον εκτέλεσης μπορεί να προϋπολογίσει την κατάλληλη χρονική στιγμή για να θέσει σε λειτουργία μια καινούρια εφαρμογή, χωρίς να απαιτείται ο πρόωρος τερματισμός εφαρμογών που ήδη βρίσκονται υπό εκτέλεση.

5.4 Μέθοδοι προστασίας της πλατφόρμας εκτέλεσης και του λογισμικού από εξωγενείς παράγοντες

Τόσο η πλατφόρμα εκτέλεσης όσο και το λογισμικό που εκτελείται σε αυτή μπορεί να γίνουν στόχος επιθέσεων από εξωτερικές οντότητες. Το είδος της προστασίας που απαιτείται σε αυτές τις περιπτώσεις είναι το ίδιο με αυτό που απαιτείται για οποιαδήποτε υπηρεσία παρέχεται πάνω από ένα δίκτυο μη έμπιστων κόμβων, όπως είναι το Διαδίκτυο.

Ένας κόμβος που βρίσκεται στο δικτυακό μονοπάτι επικοινωνίας μεταξύ δύο πλατφορμών εκτέλεσης, μπορεί να εκτελέσει μια «επίθεση μεσάζοντα» (man in the middle attack) κατά την οποία:

- να συλλέξει ευαίσθητες πληροφορίες από την επικοινωνία των δύο κόμβων,
- να δημιουργήσει μια συνθήκη «άρνησης εξυπηρέτησης», αρνούμενος να προωθήσει τα δεδομένα της μίας πλατφόρμας στην άλλη,
- να μεταβάλει τα δεδομένα που απέστειλε η πλατφόρμα–αποστολέας, ώστε αυτά να ληφθούν αλλοιωμένα από την πλατφόρμα–παραλήπτη,
- να παρουσιαστεί ως η πλατφόρμα-παραλήπτης στην πλατφόρμα-αποστολέα,
- να παρουσιαστεί ως η πλατφόρμα-αποστολέας στην πλατφόρμα-παραλήπτη,
- να δρομολογήσει την κίνηση που παράγουν οι δύο κόμβοι σύμφωνα με κάποια πολιτική με την οποία δε συμφωνεί ο ένας (τουλάχιστον) εκ των δύο κόμβων, ή
- να δώσει στους κόμβους μια ψεύτικη εικονα της τοπολογίας του δικτύου (π.χ. κάνοντας «σιωπηλή» προώθηση πακέτων[1], παριστάνοντας μη υπαρκτούς κόμβους, δημιουργώντας μη υπαρκτές ζεύξεις κ.α.).

[1] Ο δρομολογητής προωθεί πακέτα δίχως να δηλώσει την ύπαρξή του στο μονοπάτι

Με τη χρήση κρυπτογραφίας, τα ευαίσθητα δεδομένα, που ανταλλάσσονται μεταξύ των δύο πλατφόρμων εκτέλεσης, μπορεί να προστατευθούν από ενδιάμεσους κόμβους. Μάλιστα, χρησιμοποιώντας μεθόδους ασύμμετρης κρυπτογράφησης, οι κόμβοι μπορούν να ελέγξουν αν τα δεδομένα που έλαβαν δημιουργήθηκαν από έναν συγκεκριμένο κόμβο (τον κάτοχο του δημοσίου κλειδιού που τα υπέγραψε ψηφιακά) και εάν αυτά έφτασαν άθικτα στον κόμβο–παραλήπτη (έλεγχος υπογεγραμμένης τιμής συνάρτησης κατακερματισμού). Επίσης, ένας κόμβος μπορεί ανά πάσα στιγμή να αποδείξει ότι έλαβε δεδομένα που παρήχθησαν από τον κάτοχο ενός ιδιωτικού κλειδιού, κάνοντας δύσκολη τη διάψευση του γεγονότος αυτού (non-repudiation).

Η πλατφόρμα SANE [140] για ενεργά δίκτυα, χρησιμοποιεί τις παραπάνω τεχνικές για να μεταφέρει με ασφάλεια δεδομένα και λογισμικό, από μια πλατφόρμα εκτέλεσης σε μια άλλη. Συγκεκριμένα, κάθε πλατφόρμα SANE ξεκινά την εκτέλεσή της, χρησιμοποιώντας το σύστημα AEGIS [159] το οποίο πιστοποιεί ότι η εκκίνηση έγινε σε ένα ασφαλές περιβάλλον (non-tainted environment). Το δημόσιο κλειδί της πλατφόρμας, αλλά και το κάθε δημόσιο κλειδί ενός «σταδίου» εκκίνησης, υπογράφεται από το προηγούμενο στάδιο εκκίνησης, εξασφαλίζοντας έτσι μια «αλυσίδα εμπιστοσύνης» μεταξύ του περιβάλλοντος εκτέλεσης και του υλικού. Κατά την επικοινωνία μεταξύ δύο πλατφορμών εκτέλεσης SANE, χρησιμοποιείται το πρωτόκολλο Diffie–Hellman [160] για να οριστεί με ασφάλεια ένα κοινό συμμετρικό κλειδί με το οποίο θα γίνει η κρυπτογράφηση δεδομένων για μία συνεδρία. Η συμμετρική κρυπτογράφηση των δεδομένων κρίνεται αναγκαία, καθώς μια ασύμμετρη κρυπτογράφηση αυτών θα καθυστερούσε κατά μεγάλο βαθμό τη μεταφορά τους από το ένα σύστημα στο άλλο. Για να αποφευχθούν πιθανές «επιθέσεις μεσάζοντα» κατά τη διαδικασία Diffie-Hellman, οι πλατφόρμες SANE εισάγουν ένα στάδιο αυθεντικοποίησης στοιχείων όπου ελέγχονται οι ψηφιακές υπογραφές (τύπου DSA [161]) των δημοσίων κλειδιών των κόμβων.

Η χρήση ψηφιακών υπογραφών από τους δρομολογητές ενός προγραμματιζόμενου δικτύου μπορεί να επιβεβαιώσει μεν ότι ένα πακέτο πέρασε από κάποιους συγκεκριμένους δρομολογητές, δεν μπορεί όμως να εξασφαλίσει κάποιας μορφής άμυνα προς επιθέσεις τύπου «άρνησης εξυπηρέτησης» ή «σιωπηλής δρομολόγησης». Οι επιθέσεις αυτές μπορεί να ολοκληρωθούν με επιτυχία ανεξαρτήτως των όποιων βοηθητικών δεδομένων συνοδεύουν τα πακέτα των κόμβων. Προκειμένου να εξασφαλιστεί μια ασφαλέστερη δρομολόγηση των πακέτων μέχρι τον κόμβο–παραλήπτη, είναι αναγκαίο να εξεταστούν εναλλακτικά μονοπάτια δρομολόγησης τα οποία θα επιτρέψουν στα πακέτα να φτάσουν στον προορισμό τους μέσω τουλάχιστον ενός μονοπατιού που δεν περιέχει κακόβουλους κόμβους. Δυστυχώς, η δυνατότητα εναλλακτικής δρομολόγησης (δυνατότητα ορισμού συγκεκριμένου μονοπατιού δρομολόγησης) δεν παρέχεται από όλα τα πρωτόκολλα δικτύωσης και συνήθως απουσιάζει από τα δίκτυα με σταθερή υποδομή. Στην εργασία [162] των Μαυροπόδη et al. παρουσιάζεται ένα πρωτόκολλο δρομολόγησης που προσφέρει ασφαλή δρομολόγηση δεδομένων από άκρο σε άκρο χρησιμοποιώντας πολλαπλά μονοπάτια. Το πρωτόκολλο αυτό είναι σχεδιασμένο για χρήση σε κόμβους ασυρμάτων ad-hoc δικτύων και εκμεταλλεύεται μονοπάτια τα οποία δεν έχουν κοινούς κόμβους. Για να αποφευχθεί η «σιωπηλή δρομολόγηση» δεδομένων, το παραπάνω πρωτόκολλο μπορεί να συνδυαστεί με μια τεχνική αυθεντικοποίησης γειτονικών κόμβων, η οποία στηρίζεται στην εξέταση διαφόρων (φυσικών και μη) χαρακτηριστικών των κόμβων αυτών. Η τεχνική αυτή παρουσιάστηκε στην εργασία [163] και θα αναλυθεί περαιτέρω στο κεφάλαιο 7.2.

Τόσο η πλατφόρμα εκτέλεσης όσο και το λογισμικό που εκτελείται σε αυτή είναι επιρρεπή σε αλγοριθμικά σφάλματα (bugs) τα οποία μπορεί να οδηγήσουν σε κενά ασφαλείας. Ένας κακόβουλος χρήστης θα μπορούσε να χρησιμοποιήσει ένα τέτοιο σφάλμα για να παρακάμψει κάποιο έλεγχο ασφαλείας της πλατφόρμας εκτέλεσης ή να εκτρέψει την σειρά

εκτέλεσης των εντολών του λογισμικού. Προκειμένου να αποφευχθούν τέτοια σφάλματα κατά την παροχή υπηρεσιών, έχουν προταθεί διάφορες «ασφαλείς» γλώσσες προγραμματισμού (safe languages) οι οποίες παρέχουν κάποια μέτρα προστασίας στις εφαρμογές τους. Μια τέτοια γλώσσα είναι η Cyclone [164], η οποία προσφέρει στον προγραμματιστή παρόμοιες δυνατότητες με τη γλώσσα C, αλλά ταυτόχρονα παρέχει και μέτρα προστασίας, όπως τον έλεγχο ορίων μνήμης (bounds checking) σε πράξεις με δείκτες (pointers). Άλλες γλώσσες, όπως η PLAN [12], θωρακίζουν την εφαρμογή και το περιβάλλον εκτέλεσης, περιορίζοντας τις δυνατότητες της ίδιας της γλώσσας (π.χ. απαγορεύοντας την άμεση πρόσβαση σε συγκεκριμένες διευθύνσεις μνήμης). Τέλος, στα μέτρα προστασίας του λογισμικού των υπηρεσιών μπορεί να προστεθούν και τεχνικές που θωρακίζουν το γενικότερο περιβάλλον εκτέλεσης, όπως είναι η προστασία της στοίβας από σφάλματα υπερχείλισης (stack protection) [79] και η «τυχαιοποίηση» των διευθύνσεων των σελίδων μνήμης μιας εφαρμογής (Address Space Layout Randomisation – ASLR) [73].

5.5 Μέθοδοι προστασίας του χρήστη από κακόβουλους παρόχους υπηρεσιών

Η τελευταία μορφή προστασίας που θα εξεταστεί αφορά στο χρήστη μιας υπηρεσίας και προστατεύει αυτόν από παρόχους υπηρεσιών που προσφέρουν υπηρεσίες κακής ποιότητας ή υπηρεσίες που λειτουργούν με τρόπο μη αποδεκτό για το χρήστη. Στις «Υπηρεσιοστρεφείς Αρχιτεκτονικές» (Service Oriented Architectures) ο χρήστης μπορεί να αποτελεί λογισμικό που αξιοποιεί τις υπηρεσίες άλλου λογισμικού, όπως π.χ. συμβαίνει κατά την εκτέλεση «ξένου» κώδικα σε μια προγραμματιζόμενη πλατφόρμα.

Στην περίπτωση όπου ο χρήστης δεν εμπιστεύεται τον πάροχο υπηρεσιών ως προς την ορθότητα των πληροφοριών που παρέχει, μπορεί να μετατρέψει τη συναλλαγή με τον πάροχο σε ένα «τεστ», με στόχο να μειώσει την πιθανότητα να λάβει λανθασμένα δεδομένα. Συγκεκριμένα, στην περίπτωση όπου ο πάροχος εκτελεί κάποιο υπολογισμό για λογαριασμό του χρήστη, ο χρήστης μπορεί να υποβάλει σε αυτόν N αιτήσεις από τις οποίες η μία μόνο περιέχει τις πραγματικές παραμέτρους που πρέπει να εξεταστούν από τον πάροχο. Στις υπόλοιπες $N-1$ αιτήσεις θα δοθούν παράμετροι στις οποίες ο χρήστης γνωρίζει ήδη το σωστό αποτέλεσμα. Υποβάλλοντας με τυχαία σειρά αυτές τις αιτήσεις στον πάροχο, ο χρήστης μπορεί να εξασφαλίσει ότι η πιθανότητα να λάβει ψευδή απάντηση είναι $\frac{1}{N}$, αν ο πάροχος απαντήσει σωστά στις υπόλοιπες $N-1$ αιτήσεις για τις οποίες ο χρήστης ξέρει τη σωστή απάντηση. Η διαδικασία αυτή είναι πολλές φορές χρονοβόρα καθώς ο χρήστης θα πρέπει να περιμένει (στη χειρότερη περίπτωση) απάντηση για κάθε μία από τις επιπλέον $N-1$ αιτήσεις.

Σε δικτυακές εφαρμογές όπου υπάρχει επιλογή μεταξύ πολλών διαφορετικών διακομιστών για την παροχή της ίδιας υπηρεσίας, μπορεί να χρησιμοποιηθεί ένα σύστημα «ψήφων» για να επικυρώσει την ορθότητα του αποτελέσματος. Οι παράμετροι της αίτησης, στην περίπτωση αυτή, δίνονται *παράλληλα* σε ένα πλήθος n διακομιστών και το πρώτο αποτέλεσμα στο οποίο θα συμφωνήσουν m διακομιστές θα θεωρηθεί αληθές. Η τεχνική αυτή ονομάζεται *m-first voting scheme* [165] και μπορεί να χρησιμοποιηθεί σε εφαρμογές όπως η SETI@home[2], που εξετάζουν μεγάλο όγκο δεδομένων χρησιμοποιώντας τους υπολογιστές (μη έμπιστων) εθελοντών. Φυσικά, η τεχνική αυτή είναι ευπαθής σε επιθέσεις όπου m κακόβουλοι κόμβοι συνεργάζονται ώστε να αποδείξουν ότι μια ψευδής τιμή αποτελεί το σωστό αποτέλεσμα.

Σε δίκτυα όπου υπάρχει μικρή πιθανότητα να εμφανιστούν κακόβουλοι κόμβοι μπορεί

[2] `http://setiathome.berkeley.edu`

να χρησιμοποιηθεί η τεχνική *spot-testing* [165], σύμφωνα με την οποία ο χρήστης/πελάτης στέλνει «τεστ» σε τυχαίους παρόχους σύμφωνα με κάποια πιθανότητα. Όποιος πάροχος απαντήσει λάθος σε κάποιο τεστ, τοποθετείται σε μια «μαύρη λίστα» (blacklist) και τα αποτελέσματα που είχε δώσει μέχρι εκείνη τη στιγμή ανακαλώνται. Η τεχνική αυτή μπορεί να χρησιμοποιηθεί μόνο σε εφαρμογές οι οποίες έχουν κάποια περιθώρια ανοχής σε σφάλματα στους υπολογισμούς τους ή δεν απαιτούν άμεση επικύρωση των αποτελεσμάτων των υπολογισμών.

Η «μαύρη λίστα» που χρησιμοποιείται στην τεχνική *spot-testing* εισάγει την έννοια της «φήμης» ενός κόμβου, καθώς όποιος κόμβος συμμετέχει στη «μαύρη λίστα» ενός χρήστη, δε θα αξιοποιηθεί ξανά από το χρήστη αυτό (λόγω της κακής του «φήμης»). Η φήμη όμως ενός κόμβου δεν είναι απαραίτητο να έχει δυαδική μορφή (καλή/κακή). Αν ένας χρήστης βαθμολογεί την κάθε αλληλεπίδραση που είχε με έναν πάροχο υπηρεσιών χρησιμοποιώντας μια αριθμητική τιμή, η φήμη του παρόχου αυτού θα μπορούσε να περιγραφεί με το μέσο όρο των τελευταίων k βαθμολογήσεων του χρήστη [166].

Η τιμή που προκύπτει από το μέσο όρο θα μπορούσε να θεωρηθεί, επίσης, ως μια μετρική που περιγράφει την «εμπιστοσύνη» που έχει ο χρήστης στις υπηρεσίες του παρόχου. Έτσι ένας χρήστης μπορεί να αποφασίσει να σταματήσει να χρησιμοποιεί την υπηρεσία ενός συγκεκριμένου παρόχου, όταν η τιμή εμπιστοσύνης προς τον πάροχο αυτό γίνει μικρότερη από κάποιο κατώφλι θ. Επίσης, ταξινομώντας τους παρόχους υπηρεσιών σύμφωνα με τις τιμές εμπιστοσύνης, ο χρήστης μπορεί να επιλέξει για κάποια συναλλαγή τον πιο αξιόπιστο πάροχο (δηλ. τον πάροχο με την υψηλότερη τιμή εμπιστοσύνης). Στην εργασία [167] οι κόμβοι μιας Αρχιτεκτονικής Πλέγματος (Grid Architecture) δημιουργούν ένα *δίκτυο εμπιστοσύνης* στο οποίο μοιράζονται μεταξύ τους τις αξιολογήσεις που έχουν κάνει για τις διάφορες πλατφόρμες εκτέλεσης. Από τις αξιολογήσεις αυτές, οι χρήστες του δικτύου μπορούν να ανακαλύψουν νέες αξιόπιστες πλατφόρμες εκτέλεσης αλλά και να αποφύγουν πλατφόρμες εκτέλεσης με κακή φήμη. Μάλιστα, με την πάροδο του χρόνου, δημιουργούνται διμερείς σχέσεις εμπιστοσύνης μεταξύ «νησίδων» από πλατφόρμες εκτέλεσης (Grid Domains), οι οποίες επιτρέπουν στις νησίδες αυτές να φέρουν εις πέρας απο κοινού τις όποιες εργασίες (tasks), σαν να αποτελούσαν μέρος μιας ενιαίας υποδομής πλέγματος (Virtual Organisation).

Η χρήση δικτύων εμπιστοσύνης για την επιλογή έμπιστων παρόχων υπηρεσιών αποτελεί μια ιδανική λύση για δυναμικά περιβάλλοντα, όπου οι σχέσεις μεταξύ χρηστών και παρόχων υπηρεσιών δεν είναι στατικές, αλλά διαμορφώνονται κατά την αλληλεπίδρασή τους. Τέτοια περιβάλλοντα είναι οι προγραμματιζόμενες δικτυακές πλατφόρμες που βασίζονται σε ανοιχτά πρότυπα (open systems architectures), στις οποίες ένας οποιοσδήποτε κόμβος μπορεί να λάβει το ρόλο του παρόχου υπηρεσιών. Στο κεφάλαιο που ακολουθεί θα εξεταστούν διάφοροι τύποι δικτύων εμπιστοσύνης που έχουν προταθεί στη βιβλιογραφία και θα παρουσιαστεί μία νέα αρχιτεκτονική, η οποία επιτρέπει την αξιολόγηση υπηρεσιών βάσει στοιχείων που προέρχονται από ομότιμους κόμβους. Η αρχιτεκτονική αυτή ονόματι TwoHop, λειτουργεί με πλήρως κατανεμημένο τρόπο και μπορεί να χρησιμοποιηθεί για την αυτοματοποιημένη αξιολόγηση υπηρεσιών σε δίκτυα προγραμματιζόμενων κόμβων.

Κεφάλαιο 6

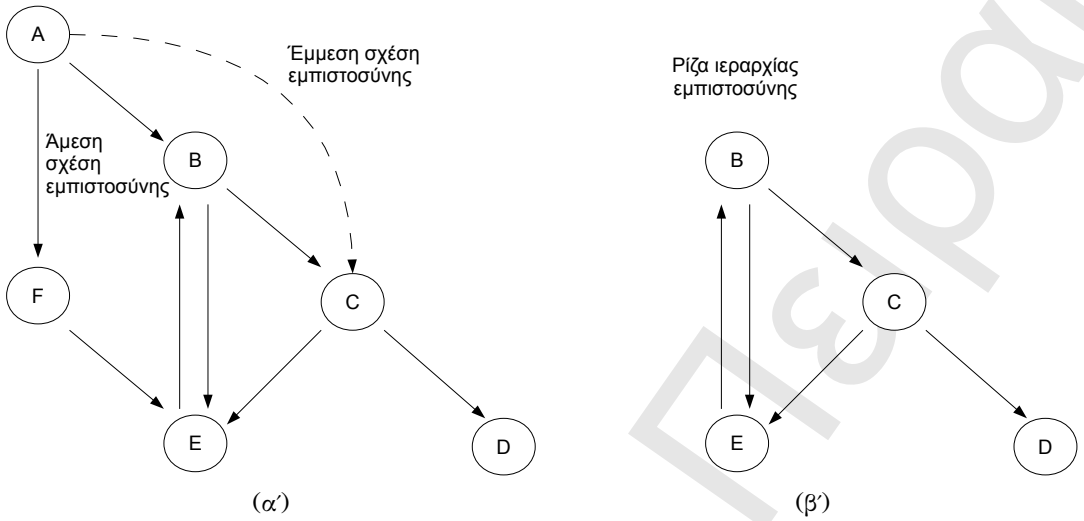
Επιλογή Διακομιστών Υπηρεσιών μέσω Δικτύων Εμπιστοσύνης

Συχνά, η αναζήτηση για «έμπιστους» παρόχους υπηρεσιών αφορά στην εξέταση ψηφιακών πιστοποιητικών που συνοδεύουν τις υπηρεσίες αυτών. Η αναγνώριση κάποιων παρόχων ως «έμπιστων» μέσω ψηφιακών πιστοποιητικών, δηλώνει ότι οι πάροχοι αυτοί, σε κάποια χρονική στιγμή, πληρούσαν κάποιες προϋποθέσεις που είχε θέσει μια αρχή έκδοσης πιστοποιητικών. Όμως, τα πιστοποιητικά αυτά δεν μπορούν να αποτελέσουν εγγυήσεις για το επίπεδο υπηρεσιών που θα προσφέρουν οι εν λόγω πάροχοι σε κάποια άλλη χρονική στιγμή. Μάλιστα, κάθε χρήστης που θα ελέγξει το πιστοποιητικό ενός παρόχου δέχεται *a priori* ότι η αρχή πιστοποίησης αυτού αποτελεί μια τρίτη έμπιστη οντότητα (trusted third party), γεγονός που έρχεται σε ρήξη με την πλήρως αποκεντρωμένη λειτουργία δικτύων όπως αυτά των ομότιμων κόμβων. Συνεπώς, η επιλογή του κατάλληλου παρόχου υπηρεσιών σε ένα δυναμικό σύστημα δεν μπορεί να επιτευχθεί βάσει κάποιου στατικού μηχανισμού πιστοποίησης.

Τα *Συστήματα Διαχείρισης Φήμης* (reputation systems) επεξεργάζονται αξιολογήσεις, που έχουν γίνει από χρήστες ενός δικτύου, προκειμένου να εξαχθούν συμπεράσματα για την ποιότητα των υπηρεσιών κάποιων παρόχων του ίδιου δικτύου. Οι αξιολογήσεις αποθηκεύονται είτε τοπικά σε κάθε κόμβο του δικτύου (π.χ. ο κάθε κόμβος αποθηκεύει τις αξιολογήσεις που έχει κάνει στο παρελθόν) είτε σε κάποια κεντρική βάση δεδομένων. Χρησιμοποιώντας τις αξιολογήσεις αυτές μπορεί κανείς να υπολογίσει μια *μετρική εμπιστοσύνης* προς ένα πάροχο υπηρεσιών, η οποία θα περιγράφει κατά πόσο ο πάροχος αυτός παρέχει μια συγκεκριμένη υπηρεσία με τρόπο που ο χρήστης θεωρεί ικανοποιητικό.

Ένα *Δίκτυο Εμπιστοσύνης* αποτελεί έναν κατευθυντικό γράφο (directed graph), του οποίου οι ακμές συνδέουν κόμβους που έχουν συνάψει κάποια σχέση εμπιστοσύνης. Οι ακμές του γράφου συχνά συνοδεύονται από βάρη που περιγράφουν με ποσοτικό τρόπο τη σχέση μεταξύ δύο γειτονικών κόμβων. Οι σχέσεις αυτές λέγονται *άμεσες σχέσεις εμπιστοσύνης* και προκύπτουν μετά από κάποια αλληλεπίδραση που είχαν οι κόμβοι αυτοί. Ένας κόμβος μπορεί να συνάψει επίσης *έμμεσες σχέσεις εμπιστοσύνης* με κόμβους με τους οποίους δεν είχε κάποια προηγούμενη αλληλεπίδραση στο παρελθόν. Οι έμμεσες σχέσεις εμπιστοσύνης προκύπτουν από την «αλυσιδωτή» φύση της εμπιστοσύνης, σύμφωνα με την οποία η σχέση εμπιστοσύνης του κόμβου A με το γειτονικό κόμβο B, θα επιτρέψει στον A να συνάψει «ιδεατούς» δεσμούς εμπιστοσύνης με άγνωστους κόμβους οι οποίοι είναι γείτονες του B. Επειδή όμως ο κόμβος A δεν είχε κάποια προηγούμενη αλληλεπίδραση με αυτούς τους κόμβους, η έμμεση τιμή εμπιστοσύνης προς αυτούς μπορεί να υπολογισθεί με διαφορετικό τρόπο από ότι υπολογίσθηκε για τον κόμβο B. Έτσι, π.χ. η τιμή αυτή μπορεί να επηρεάζε-

ται από το βαθμό εμπιστοσύνης του A προς το B αλλά και από το βαθμό εμπιστοσύνης του B προς αυτούς. Στο σχήμα 6.1α΄ φαίνεται ένα παράδειγμα έμμεσης σχέσης εμπιστοσύνης μεταξύ των κόμβων A και C, που προέκυψε λόγω της άμεσης σχεσης εμπιστοσύνης που είχε συνάψει ο A με τον κόμβο B.



Σχήμα 6.1: Παράδειγμα ενός δικτύου εμπιστοσύνης και της αντίστοιχης ιεραρχίας εμπιστοσύνης για τον κόμβο B.

Οι έμμεσες σχέσεις εμπιστοσύνης μπορεί να επεκταθούν ώστε να συμπεριλάβουν τις ιδεατές σχέσεις μεταξύ κόμβων που βρίσκονται σε απόσταση μεγαλύτερη των 2 βημάτων (hops), όπως συμβαίνει για τους κόμβους A και D, του σχήματος 6.1α΄. Οι κόμβοι για τους οποίους ένας χρήστης μπορεί να υπολογίσει μια έμμεση ή άμεση τιμή εμπιστοσύνης, δημιουργούν ένα ασθενώς συνεκτικό γράφο, ο οποίος στην εργασία [168] ονομάζεται *ιεραρχία εμπιστοσύνης*. Η *ρίζα* της ιεραρχίας αυτής είναι ο κόμβος που αντιπροσωπεύει τον ίδιο το χρήστη. Στο σχήμα 6.1β΄ φαίνεται η ιεραρχία εμπιστοσύνης που αντιστοιχεί στον κόμβο B του σχήματος 6.1α΄.

Τα μονοπάτια που οδηγούν από τη ρίζα της ιεραρχίας σε οποιοδήποτε κόμβο αυτής ονομάζονται *μονοπάτια εμπιστοσύνης*. Κάθε μονοπάτι περιγράφει μια σχέση εμπιστοσύνης και από τα βάρη που αντιστοιχούν στις ακμές του, μπορεί κανείς να εξαγάγει μια ποσοτική εκτίμηση της εμπιστοσύνης αυτής. Επειδή ένας κόμβος στην ιεραρχία εμπιστοσύνης μπορεί να είναι προσβάσιμος μέσω πολλαπλών μονοπατιών, πολλά δίκτυα εμπιστοσύνης χρησιμοποιούν μια συνάρτηση συγκερασμού των επί μέρους τιμών εμπιστοσύνης που προκύπτουν από κάθε μονοπάτι, προκειμένου να υπολογίσουν την τελική τιμή εμπιστοσύνης που αντιστοιχεί στον κόμβο αυτό. Η τιμή αυτή ονομάζεται *δικτυακή τιμή εμπιστοσύνης* και διαφέρει από την καθολική τιμή εμπιστοσύνης καθώς υπολογίζεται με βάση τα στοιχεία που προκύπτουν από την ιεραρχία εμπιστοσύνης ενός κόμβου.

Αν ο δεσμός εμπιστοσύνης που σχηματίζεται μεταξύ δύο κόμβων περιγράφει την ποιότητα μιας υπηρεσίας που παρείχε ο ένας κόμβος στον άλλο, τότε μπορεί κανείς να χρησιμοποιήσει το δίκτυο εμπιστοσύνης που προκύπτει από τέτοιους δεσμούς για να λάβει εκτιμήσεις ως προς την ποιότητα υπηρεσιών γνωστών και άγνωστων παρόχων (αλλά και για να ανακαλύψει νέους έμπιστους παρόχους). Φυσικά, προκειμένου να έχει ο οποιοσδήποτε κόμβος πρόσβαση στις αξιολογήσεις των υπόλοιπων κόμβων, αυτές θα πρέπει να γίνουν δημόσια διαθέσιμες. Για να γίνει όμως αυτό, δεν απαιτείται κάποια κεντρικοποιημένη υπηρεσία διάθεσης των αξιολογήσεων. Οι κόμβοι μπορούν ως μέλη μιας αρχιτεκτονικής ομότιμων

κόμβων να διαθέσουν ο καθένας τις δικές του αξιολογήσεις στους υπόλοιπους. Έτσι, με πλήρως κατανεμημένο τρόπο, ο κάθε κόμβος μπορεί να υπολογίσει μια τιμή εμπιστοσύνης για έναν πάροχο, βασισμένη στις αξιολογήσεις των κόμβων που συμμετέχουν στην ιεραρχία εμπιστοσύνης του. Η τιμή που θα υπολογιστεί θα είναι μια δικτυακή τιμή εμπιστοσύνης, η οποία θα αποτυπώνει την άποψη που έχουν εκείνη τη στιγμή οι κόμβοι της ιεραρχίας για το συγκεκριμένο πάροχο.

Ένα τέτοιο δυναμικό σύστημα μπορεί να αποτελέσει τη βάση για μια κατανεμημένη υπηρεσία αξιολόγησης παρόχων υπηρεσιών σε δίκτυα προγραμματιζόμενων κόμβων. Στις ενότητες που ακολουθούν θα εξεταστούν σχετικές προτάσεις από τη βιβλιογραφία και θα παρουσιαστεί ένας νέος τύπος δικτύου εμπιστοσύνης, ο οποίος σχεδιάστηκε με γνώμονα τις απαιτήσεις μιας προγραμματιζόμενης δικτυακής αρχιτεκτονικής.

6.1 Έρευνα πεδίου σε θέματα Δικτύων Εμπιστοσύνης

Στόχος της παρούσας ενότητας είναι να παρουσιάσει συνοπτικά το ερευνητικό έργο που έχει γίνει τα τελευταία χρόνια στο χώρο των Δικτύων Εμπιστοσύνης και να αποτυπώσει τους λόγους για τους οποίους τα προτεινόμενα μοντέλα εμπιστοσύνης δεν καλύπτουν πλήρως τις ανάγκες των ομότιμων κόμβων μιας δικτυακής προγραμματιζόμενης υποδομής.

Η χρήση και η διαχείριση τιμών εμπιστοσύνης με στόχο την διεκπεραίωση αλγοριθμικών διαδικασιών παρουσιάζεται για πρώτη φορά στη διδακτορική διατριβή του Stephen P. Marsh [166]. Στην εργασία αυτή εξετάζονται κοινωνιολογικές και ψυχολογικές πτυχές της έννοιας της εμπιστοσύνης και προτείνεται ένα μοντέλο το οποίο επιτρέπει σε οντότητες να υπολογίσουν τιμές εμπιστοσύνης προς τρίτες οντότητες βάσει προηγούμενων συναλλαγών με αυτές. Σύμφωνα με το μοντέλο αυτό, η εμπιστοσύνη $\widehat{T_A(B)}$ που *αντιλαμβάνεται* μια οντότητα A προς μία οντότητα B (perceptual trust) ισούται με το μέσο όρο των τελευταίων k αξιολογήσεων που έγιναν από την οντότητα A προς την οντότητα B. Το μέγεθος της «μνήμης» προηγούμενων αξιολογήσεων $k \in \mathbb{N}^*$ επιλέγεται ανάλογα με τις ανάγκες της εφαρμογής. Επίσης, στην εργασία αυτή εισάγεται και η έννοια της περιστασιακής εμπιστοσύνης (situational trust), όπου για την κατάσταση x, η εμπιστοσύνη $\mathrm{T_A(B}, x)$ της οντότητας A προς τη B ισούται με το γινόμενο $U_A(x) \cdot I_A(x) \cdot \widehat{T_A(B)}$, όπου το $I_A(x)$ μετρά τη σημαντικότητα της κατάστασης x για την οντότητα A και το $U_A(x)$ μετρά τη χρησιμότητα που έχει για την A ο ορθός χειρισμός της κατάστασης x. Δυστυχώς το μοντέλο αυτό υπολογίζει τιμές εμπιστοσύνης με βάση τα τοπικά και μόνο δεδομένα κάθε οντότητας (προηγούμενες αξιολογήσεις). Έτσι, δεν μπορεί να εκμεταλλευτεί τυχόν αξιολογήσεις που έχουν κάνει άλλες οντότητες στο δίκτυο εμπιστοσύνης αλλά ούτε και τις σχέσεις εμπιστοσύνης μεταξύ οντοτήτων που συνήθως υπάρχουν σε δικτυακά συνεργατικά περιβάλλοντα. Τέλος, στο μοντέλο αυτό θεωρείται προβληματικό σενάριο ο υπολογισμός εμπιστοσύνης προς ένα πάροχο για τον οποίο δεν υπάρχουν τοπικά δεδομένα από προηγούμενες αξιολογήσεις.

Μία από τις πρώτες συνεργατικές μεθόδους υπολογισμού εμπιστοσύνης χρησιμοποιήθηκε για την αξιολόγηση δημοπρατών στην ηλεκτρονική πλατφόρμα δημοπρασίας eBay [169]. Όπως και στο μοντέλο του Marsh, έτσι και εδώ η εμπιστοσύνη προς ένα δημοπράτη υπολογίζεται από τις προηγούμενες αξιολογήσεις που έκαναν οι χρήστες που αγόρασαν κάποιο αντικείμενο από αυτόν. Όμως, στην περίπτωση του eBay, οι αξιολογήσεις κάθε χρήστη (που έχουν τη μορφή ακεραίων από το σύνολο $\{-1, 0, 1\}$) προστίθενται και το άθροισμα αυτών γίνεται άμεσα διαθέσιμο στο ευρύ κοινό από τη σελίδα–προφίλ του δημοπράτη. Επίσης οι χρήστες έχουν τη δυνατότητα να σχολιάσουν δημόσια τις υπηρεσίες που έλαβαν από ένα δημοπράτη. Έτσι, ένας χρήστης μπορεί να συμβουλευθεί προηγούμενες δημόσιες κρίσεις για ένα δημοπράτη πρίν αγοράσει κάποιο αντικείμενο από αυτόν. Το βασικό μειονέκτημα

του μοντέλου αυτού είναι η κεντρικοποιημένη λειτουργία του, η οποία καθιστά δύσκολη την επέκταση της χρήσης του σε δίκτυα με μεγάλο πλήθος κόμβων αλλά και σε δίκτυα όπου όλοι οι κόμβοι θα πρέπει να λειτουργούν ως ισότιμοι πάροχοι υπηρεσιών (όπως ισχύει στα δίκτυα ομότιμων κόμβων). Επίσης, το μοντέλο αυτό επιτρέπει επιθέσεις τύπου *karma suicide*, όπου ένας δημοπράτης χρησιμοποιεί την καλή φήμη που έχει για να φέρει εις πέρας μια συναλλαγή όπου θα πωλήσει ένα αντικείμενο χαμηλής αξίας σε υψηλή τιμή. Μόλις η συναλλαγή ολοκληρωθεί ο δημοπράτης θα αλλάξει ταυτότητα και θα συνδεθεί στο δίκτυο εμπιστοσύνης με νέα ταυτότητα αποφεύγοντας έτσι τις όποιες αρνητικές συνέπειες της συναλλαγής που έκανε.

Το Poblano είναι ένα κατανεμημένο σύστημα υπολογισμού εμπιστοσύνης για κόμβους που συμμετέχουν στην πλατφόρμα JXTA [170]. Κάθε κόμβος διατηρεί μια δημόσια βάση δεδομένων με αξιολογήσεις που έχει κάνει σε υπηρεσίες άλλων κόμβων. Ένας κόμβος, που επιθυμεί να υπολογίσει μια τιμή εμπιστοσύνης προς ένα πάροχο υπηρεσιών, μπορεί να συμβουλευτεί τις βάσεις δεδομένων γνωστών του κόμβων και από αυτές να εξαγάγει συμπεράσματα για το επίπεδο της παρεχόμενης υπηρεσίας. Μάλιστα, μπορεί επίσης να χρησιμοποιήσει αυτές τις βάσεις δεδομένων, για να ανακαλύψει νέους κόμβους, οι οποίοι με τη σειρά τους θα τον οδηγήσουν σε νέες αξιολογήσεις. Έτσι, σχηματίζεται ένας γράφος εμπιστοσύνης, από τον οποίο μπορούν να υπολογιστούν *καθολικές τιμές εμπιστοσύνης* (global trust values) για τους κόμβους. Αυτό σημαίνει ότι οποιοσδήποτε κόμβος και αν υπολογίσει την τιμή εμπιστοσύνης για έναν κόμβο A, ανεξάρτητα από το δίκτυο γνωστών του κόμβων, η τιμή αυτή θα είναι πάντα η ίδια. Το κύριο μειονέκτημα της αρχιτεκτονικής αυτής είναι η ύπαρξη μίας και μόνο μορφής ακμών η οποία ενώνει τους κόμβους στο γράφο εμπιστοσύνης. Το μειονέκτημα αυτό συναντάται σε όλες τις αρχιτεκτονικές εμπιστοσύνης στις οποίες επιτρέπεται η δημιουργία μονοπατιών με αυθαίρετο αριθμό κόμβων. Περιορίζοντας το είδος των σχέσεων που μπορούν να έχουν δύο κόμβοι, γίνεται αδύνατη η διαφοροποίηση μεταξύ π.χ. της αξιολόγησης της υπηρεσίας ενός κόμβου και της ποιότητας μιας αξιολόγησης. Στην ουσία, οι χρήστες ενός τέτοιου δικτύου αναγκάζονται να περιγράψουν διαφορετικές έννοιες με μια *μονοδιάστατη μετρική εμπιστοσύνης*.

Σε αντίθεση με τις *καθολικές τιμές εμπιστοσύνης* που παράγει το Poblano, η πλατφόρμα GNUnet [171] στηρίζεται σε *τοπικές τιμές εμπιστοσύνης*. Οι χρήστες της πλατφόρμας καταγράφουν την ποιότητα προηγούμενων συναλλαγών με κάποιους κόμβους–παρόχους, ώστε στο μέλλον να μπορούν να αξιοποιήσουν αυτές τις αξιολογήσεις ως δείκτες εμπιστοσύνης προς τις υπηρεσίες των παραπάνω παρόχων. Οι αξιολογήσεις στο σύστημα αυτό είναι ιδιωτικές και δεν δημοσιοποιούνται στο υπόλοιπο δίκτυο εμπιστοσύνης. Ως εκ τούτου, ο κάθε κόμβος υπολογίζει βάσει τοπικών δεδομένων την τιμή εμπιστοσύνης προς μια υπηρεσία, δίχως να λαμβάνονται υπόψιν τυχόν αξιολογήσεις που έχουν γίνει από άλλους κόμβους του δικτύου. Με παρόμοιο τρόπο, στο σύστημα PGP [66] ο κάθε χρήστης αποδίδει κάποιο βαθμό εμπιστοσύνης σε σχέσεις μεταξύ των ταυτοτήτων φυσικών προσώπων και των δημοσίων κλειδιών. Επειδή ο βαθμός εμπιστοσύνης αυτός παραμένει ιδιωτικός, δε μπορούν να εξαχθούν ασφαλή συμπεράσματα για τις σχέσεις μεταξύ ταυτοτήτων και κλειδιών αγνώστων χρηστών.

Στο σύστημα Advogato [172], το δίκτυο εμπιστοσύνης μοντελοποιείται ως ένας γράφος με κατευθυνόμενες ακμές, όπου ο βαθμός εμπιστοσύνης ενός κόμβου A προς ένα κόμβο B ισούται με την πιθανότητα ένας τυχαίος περίπατος του γράφου (random walk) που ξεκινά από τον κόμβο A να καταλήξει στον κόμβο B. Το μοντέλο αυτό στηρίζεται σε δύο παραδοχές. Η πρώτη παραδοχή απαιτεί τη χρήση γράφων εμπιστοσύνης όπου όλοι οι κόμβοι έχουν τον ίδιο εσωτερικό βαθμό (in–degree). Η δεύτερη παραδοχή απαιτεί από κάθε κόμβο του δικτύου να έχει *πλήρη γνώση* της δομής του γράφου. Οι παραδοχές αυτές είναι αρκετά περιοριστικές και δεν επιτρέπουν τη χρήση του Advogato σε σύγχρονες εφαρμογές όπως

αυτές που αφορούν δίκτυα ομότιμων κόμβων, όπου υπάρχει άναρχη δόμηση των σχέσεων μεταξύ των κόμβων και το πλήθος των κόμβων είναι τέτοιο που δεν επιτρέπει σε κάθε κόμβο να έχει μια πλήρη εικόνα του συνολικού δικτύου.

Το TrustFlow είναι ένα δίκτυο εμπιστοσύνης που χρησιμοποιείται από την κοινότητα ιστολογίων LiveJournal [173]. Κάθε χρήστης αυτού του δικτύου διατηρεί μια ταξινομημένη λίστα από έμπιστους «φίλους». Σκοπός του συστήματος TrustFlow είναι η εισαγωγή σε αυτή τη λίστα άγνωστων μελών του δικτύου (μελών που δεν είχαν προηγούμενη επαφή με το χρήστη). Σε κάθε χρήστη δίνονται κάποια «κουπόνια» εμπιστοσύνης καθώς και ένα «δοχείο» που μπορεί να χωρέσει συγκεκριμένο αριθμό κουπονιών. Ο χρήστης τοποθετεί κουπόνια στα δοχεία των φίλων του με τη σειρά που εκείνοι εμφανίζονται στην παραπάνω ταξινομημένη λίστα. Μόλις γεμίσει από κουπόνια το δοχείο του χρήστη A, τα περισσευούμενα κουπόνια θα τοποθετηθούν στα δοχεία των φίλων του A με τη σειρά που εκείνοι εμφανίζονται στην ταξινομημένη λίστα αυτού. Με τον τρόπο αυτό παράγεται μια νέα ταξινομημένη λίστα για κάθε χρήστη, η οποία περιέχει τους χρήστες που έχουν το μεγαλύτερο αριθμό κουπονιών. Ένας χρήστης του δικτύου εμπιστοσύνης TrustFlow δε χρειάζεται να γνωρίζει τη συνολική δομή του γράφου εμπιστοσύνης, παρά μόνο το υποδίκτυο που απαρτίζεται από εκείνον και τους φίλους αυτού (*μερική γνώση του γράφου*). Το βασικό μειονέκτημα αυτής της αρχιτεκτονικής είναι η λανθασμένη ταξινόμηση που προκύπτει στις περιπτώσεις όπου ένας χρήστης έχει μικρό αριθμό από φίλους (σε σχέση με τα άλλα μέλη του δικτύου) και έτσι οι φίλοι αυτού θα λάβουν υψηλές θέσεις στην ταξινομημένη λίστα.

Η υπηρεσία OpenPrivacy [174] δημιουργεί σχέσεις εμπιστοσύνης μεταξύ αγνώστων κόμβων, δηλαδή κόμβων που δεν είχαν κάποια επαφή στο παρελθόν. Συγκεκριμένα, ο χρήστης του συστήματος αυτού επικοινωνεί με μία οντότητα διαχείρισης φήμης (reputation authority) η οποία τον ενημερώνει για τη γνώμη άλλων χρηστών σχετικά με κάποιο χρήστη του δικτύου. Επειδή οι χρήστες μπορούν να καταθέσουν τις απόψεις τους σε αυτή την οντότητα, το σύστημα γίνεται πιο δυναμικό από ότι θα ήταν αν η οντότητα αυτή έπαιζε απλά το ρόλο ενός Τρίτου Έμπιστου Προσώπου (Trusted Third Party). Δυστυχώς όμως, το OpenPrivacy παραμένει ένα κεντρικοποιημένο σύστημα που στηρίζεται στην παροχή υπηρεσιών από συγκεκριμένους κόμβους του δικτύου (οντότητες διαχείρισης φήμης).

Το σύστημα Free Haven [175] επιτρέπει σε χρήστες να δημοσιεύσουν με ανώνυμο τρόπο έγγραφα σε ειδικούς εξυπηρετητές. Οι εξυπηρετητές αυτοί, που αποτελούν μέρος ενός ανώνυμου δικτύου υπηρεσιών, δημιουργούν μεταξύ τους «συμβόλαια» για την αποθήκευση δεδομένων. Ένας εξυπηρετητής ολοκληρώνει την εργασία που περιγράφεται σε ένα συμβόλαιο όταν έχει αποθηκεύσει για ένα συγκεκριμένο χρονικό διάστημα κάποια δεδομένα ενός άλλου εξυπηρετητή. Με την επιτυχή ολοκλήρωση αυτής της εργασίας, ο εξυπηρετητής αυτός λαμβάνει κάποιους βαθμούς «καλής διαγωγής», τους οποίους «εξαργυρώνει» αποθηκεύοντας δικά του δεδομένα σε άλλους εξυπηρετητές. Το Free Haven επιτρέπει τη χρήση είτε ενός κεντρικοποιημένου συστήματος για τη διαχείριση των βαθμών «καλής διαγωγής», είτε τη χρήση τοπικών βάσεων δεδομένων ανά εξυπηρετητή, δίνοντας έτσι τη δυνατότητα να διατηρήσει κάθε εξυπηρετητής τη δική του οπτική για τη διαγωγή των υπόλοιπων μελών του δικτύου.

Το P2PRep [176] είναι ένα σύστημα διαχείρισης τιμών εμπιστοσύνης για δίκτυα ομότιμων κόμβων. Κάθε κόμβος διατηρεί μια τοπική βάση δεδομένων με αξιολογήσεις για άλλους κόμβους του δικτύου, με τους οποίους έχει έρθει σε επαφή στο παρελθόν. Η βάση αυτή είναι προσβάσιμη από όλους τους κόμβους του δικτύου και οι τιμές της ανανεώνονται όποτε ο κόμβος έλθει σε επαφή με κάποιον άλλο κόμβο. Η ύπαρξη δημόσια διαθέσιμων αξιολογήσεων επιτρέπει σε ένα κόμβο να ελέγξει τις βάσεις δεδομένων άλλων κόμβων και από τις τιμές αυτές να εξαγάγει ένα *δικτυακό μέτρο εμπιστοσύνης* για έναν κόμβο με τον οποίο μπορεί να μην είχε κάποια προηγούμενη αλληλεπίδραση στο παρελθόν. Το σύστη-

μα XRep [177] προσθέτει στο P2PRep τη δυνατότητα αξιολόγησης πόρων. Με τον τρόπο αυτό μπορεί να αποφευχθεί η χρήση κάποιας υπηρεσίας χαμηλής ποιότητας ή η εκτέλεση κακόβουλου λογισμικού, αν βρεθούν για τους πόρους αυτούς αντίστοιχες ενδεικτικές αξιολογήσεις. Σοβαρό πρόβλημα των P2PRep και XRep, αποτελεί η έλλειψη αξιολογήσεων προς κόμβους που παρέχουν τιμές εμπιστοσύνης. Δίχως τις αξιολογήσεις αυτές, οι χρήστες των δικτύων P2PRep/XRep παραμένουν ευάλωτοι σε επιθέσεις κόμβων που παρέχουν ψευδείς τιμές εμπιστοσύνης.

Το δίκτυο εμπιστοσύνης EigenTrust [178] επιτρέπει τον υπολογισμό καθολικών τιμών εμπιστοσύνης για κόμβους ομότιμων δικτύων, βάσει αξιολογήσεων που γίνονται από κόμβους των δικτύων αυτών. Ο βασικός αλγόριθμος του EigenTrust υπολογίζει τις τιμές εμπιστοσύνης προς όλους τους κόμβους του δικτύου και έτσι απαιτεί πλήρη γνώση του γράφου εμπιστοσύνης. Μέσα από μια επαναληπτική διαδικασία, σταθμίζει τις τιμές των αξιολογήσεων, πολλαπλασιάζοντας την αξιολόγηση που έκανε ένας κόμβος με την καθολική τιμή εμπιστοσύνης προς τον κόμβο αυτό. Ως καθολική τιμή εμπιστοσύνης προς ένα κόμβο ορίζεται το άθροισμα των σταθμισμένων αξιολογήσεων που αφορούν στον κόμβο αυτό. Ο αλγόριθμος υπολογισμού εμπιστοσύνης που χρησιμοποιεί το EigenTrust απαιτεί τη χρήση αξιολογήσεων με κανονικοποιημένες τιμές, ώστε το σύνολο των αξιολογήσεων που έχει κάνει ένας κόμβος να έχει άθροισμα ίσο με τη μονάδα (1). Η μέθοδος κανονικοποίησης που εφαρμόζεται φαίνεται παρακάτω:

$$c_{i,j} = \frac{max(s_{ij}, 0)}{\sum_j max(s_{ij}, 0)},$$

όπου $c_{i,j}$ είναι η κανονικοποιημένη τιμή της αξιολόγησης s_{ij} που έκανε ο κόμβος i για τον κόμβο j. Η κανονικοποίηση αυτή οδηγεί πολλές φορές σε τιμές οι οποίες δεν έχουν σχέση με τις αρχικές τιμές των αξιολογήσεων, όπως π.χ. στην περίπτωση όπου ο υπεύθυνος για μια αξιολόγηση έχει κάνει μόνο αυτή την αξιολόγηση και μέσω της παραπάνω κανονικοποίησης η τιμή της αξιολόγησης μετατρέπεται σε μονάδα (1).

Στην εργασία των Jiang et al. [179] παρουσιάζεται μια κατανεμημένη μέθοδος υπολογισμού τιμών εμπιστοσύνης η οποία αξιοποιεί «ψήφους» των μελών του δικτύου εμπιστοσύνης. Οι ψήφοι αυτές αθροίζονται και διαμορφώνουν την τελική τιμή εμπιστοσύνης σύμφωνα με κάποια βάρη. Οι συγγραφείς της εργασίας προτείνουν δύο μορφές ψηφίσματος. Η πρώτη μορφή αφορά στην περίπτωση όπου είναι δυνατό να συλλεχθούν οι πληροφορίες από όλους τους κόμβους του δικτύου, ενώ η δεύτερη αφορά στην περίπτωση όπου προτιμάται να αξιοποιηθούν μόνο τοπικά δεδομένα. Η αρχιτεκτονική που προτείνεται παρουσιάζει δυστυχώς μεγάλες καθυστερήσεις στο χρόνο σύγκλισης του αλγορίθμου υπολογισμού των τελικών τιμών εμπιστοσύνης. Επίσης, είναι επιρρεπής σε επιθέσεις όπου κακόβουλοι κόμβοι επηρεάζουν το αποτέλεσμα του ψηφίσματος καταθέτοντας ψήφους με την ίδια τιμή.

Οι Raya et al. παρουσιάζουν στην εργασία [180] ένα δίκτυο εμπιστοσύνης που έχει ως στόχο την ανάδειξη αξιόπιστων μετρήσεων. Συγκεκριμένα, εξετάζονται μετρήσεις που αφορούν την ακριβή γεωγραφική θέση ενός στόχου ή τη χρονική στιγμή που έλαβε χώρα κάποιο συμβάν. Ως πιο αξιόπιστες θεωρούνται οι μετρήσεις που έγιναν από αισθητήρες που βρίσκονταν πιο κοντά (τοπολογικά ή χρονικά) στο συμβάν. Η αρχιτεκτονική που προτείνεται στην εργασία αυτή, αρχικά συλλέγει όλες τις μετρήσεις από τους αισθητήρες και εν συνεχεία τις επεξεργάζεται βάσει της θεωρίας Dempster-Shafer [181], η οποία επιτρέπει τη λήψη τοπικών αποφάσεων ακόμη και όταν τα στοιχεία (δηλ. οι μετρήσεις) δε συμφωνούν μεταξύ τους.

Στην εργασία [182] παρουσιάζεται ένα δυναμικό σύστημα διαχείρισης τιμών εμπιστοσύνης που έχει ως στόχο την υποστήριξη αξιόπιστων υπηρεσιών διάθεσης αρχείων πάνω από δίκτυα ομότιμων κόμβων. Το σύστημα αυτό, που βασίζεται σε μια βελτιωμένη έκδο-

ση του συστήματος PeerTrust [183], χρησιμοποιεί μια αρχιτεκτονική δύο επιπέδων, με το πρώτο επίπεδο να απαρτίζεται από κόμβους–παρόχους και το δεύτερο επίπεδο να απαρτίζεται από κόμβους–αξιολογητές. Οι κόμβοι–πάροχοι διαθέτουν αρχεία στους κόμβους του δικτύου. Ένας κόμβος A που επιθυμεί να υπολογίσει το επίπεδο των υπηρεσιών (βαθμός εμπιστοσύνης) ενός κόμβου–παρόχου B, θα πρέπει να υπολογίσει το σταθμισμένο μέσο όρο όλων των αξιολογήσεων που έχουν γίνει από τους κόμβους–αξιολογητές για τον B. Κάθε αξιολόγηση συμμετέχει στο μέσο όρο με ένα βάρος το οποίο περιγράφει την αξιοπιστία του αξιολογητή, όπως αυτή γίνεται αντιληπτή από τον A. Συγκεκριμένα, η αξιοπιστία υπολογίζεται με βάση τον αριθμό των κοινών παρόχων που ο αξιολογητής και ο Α είχαν αξιολογήσει στο παρελθόν αλλά και από το βαθμό ομοιότητας των σχετικών αξιολογήσεών για τους παρόχους αυτούς. Το συνολικό σύστημα έχει ένα δυναμικό τρόπο λειτουργίας σε σχέση με το χρόνο, καθώς δίνει μεγαλύτερη έμφαση σε πρόσφατες αξιολογήσεις. Θα πρέπει όμως να σημειωθεί ότι στηρίζεται υπερβολικά στην εθελοντική διαφήμιση των κόμβων–αξιολογητών αλλά και στα δεδομένα που αυτοί θα δημοσιοποιήσουν στο δίκτυο, καθώς από αυτά θα προκύψει ο αριθμός των κοινών αξιολογήσεων και τελικά θα οριστεί ο βαθμός εμπιστοσύνης προς τον κάθε κόμβο–πάροχο. Κακόβουλοι κόμβοι μπορούν εύκολα να εκμεταλλευτούν αυτή τη σχέση με στόχο να βλάψουν την υπόληψη συγκεκριμένων κόμβων–παρόχων.

Στην εργασία των Theodorakopoulos και Baras [184] παρουσιάζεται ένα δίκτυο εμπιστοσύνης βασισμένο σε κατευθυνόμενο γράφο με σύνθετα βάρη. Το βάρος μιας ακμής αποτελείται από δύο στοιχεία: α) το βαθμό εμπιστοσύνης ενός κόμβου προς ένα άλλο και β) το βαθμό εγκυρότητας της παραπάνω μέτρησης εμπιστοσύνης. Χρησιμοποιώντας τη θεωρία των *ημιδακτυλίων* οι συγγραφείς υπολογίζουν με αλγεβρικό τρόπο το βαθμό εμπιστοσύνης μεταξύ κόμβων που δεν είχαν κάποια αλληλεπίδραση στο παρελθόν και εντοπίζουν το μονοπάτι με το μεγαλύτερο βαθμό εμπιστοσύνης ανάμεσα στους δύο κόμβους. Το μονοπάτι αυτό είναι ιδιαίτερα χρήσιμο σε εφαρμογές δρομολόγησης σε δίκτυα κινητών κόμβων όπου δεν υπάρχει κάποια σταθερή υποδομή για τη δρομολόγηση δεδομένων και οι κόμβοι αναγκάζονται να προωθήσουν τα δεδομένα τους μέσω ενός δικτύου μη έμπιστων κόμβων. Η αλγεβρική μέθοδος περιγραφής του αλγορίθμου που παρουσιάστηκε στην εργασία [184] επεκτείνεται από τους ίδιους συγγραφείς στην εργασία [185], ώστε να αποτελέσει ένα μέσο περιγραφής και αξιολόγησης αλγορίθμων υπολογισμού εμπιστοσύνης. Περισσότερες πληροφορίες για τη μέθοδο αυτή μπορεί να βρεθούν στην ενότητα 6.5, όπου η μέθοδος των Theodorakopoulos και Baras χρησιμοποιείται για την αξιολόγηση των χαρακτηριστικών του προτεινόμενου δικτύου εμπιστοσύνης TwoHop.

6.2 Η αρχιτεκτονική TwoHop

Το δίκτυο εμπιστοσύνης TwoHop, που παρουσιάστηκε για πρώτη φορά στην εργασία [168], αποτελεί μια νέα πλατφόρμα για τη διαχείριση σχέσεων εμπιστοσύνης σε δίκτυα ομότιμων κόμβων. Η πλατφόρμα TwoHop επιτρέπει τον υπολογισμό τιμών εμπιστοσύνης προς παρόχους υπηρεσιών με τρόπο γρήγορο και πλήρως κατανεμημένο, καθιστώντας την κατάλληλη για χρήση σε προγραμματιζόμενα δικτυακά περιβάλλοντα. Στην ενότητα αυτή θα παρουσιαστούν οι βασικές αρχές που επηρεάζουν την αρχιτεκτονική της πλατφόρμας καθώς και οι βασικές διαφορές του δικτύου TwoHop σε σχέση με τα αντίστοιχα δίκτυα εμπιστοσύνης που έχουν προταθεί μέχρι σήμερα στη βιβλιογραφία.

Ασύμμετρες σχέσεις εμπιστοσύνης

Οι σχέσεις που περιγράφει ένα δίκτυο εμπιστοσύνης TwoHop είναι ασύμμετρες. Αυτό σημαίνει ότι η σχέση εμπιστοσύνης που ο κόμβος A έχει αναπτύξει για τον κόμβο B δεν

συνεπάγεται κάποια σχέση εμπιστοσύνης του κόμβου B για τον κόμβο A. Η αρχή αυτή συναντάται στις περισσότερες αρχιτεκτονικές δικτύων εμπιστοσύνης και οδηγεί στη μοντελοποίηση των δικτύων αυτών με χρήση κατευθυντικών γράφων.

Μονοπάτια εμπιστοσύνης περιορισμένου μήκους

Η έννοια της εμπιστοσύνης δεν έχει απόλυτα μεταβατικό χαρακτήρα. Έτσι, η φύση της εμπιστοσύνης που περιγράφεται από μια άμεση σχέση εμπιστοσύνης δεν είναι η ίδια με αυτή που περιγράφεται από μια έμμεση σχέση εμπιστοσύνης. Όπως διαπιστώνουν και οι συγγραφείς της εργασίας [186], με την πρόσθεση ενός κόμβου σε ένα μονοπάτι εμπιστοσύνης, το νέο μονοπάτι περιγράφει μια διαφορετικής μορφής εμπιστοσύνη από ότι το μονοπάτι δίχως το νέο κόμβο. Μάλιστα, στην εργασία [187] αναφέρεται ότι κάθε σύστημα που αγνοεί την αλλοίωση που επιφέρουν οι σχέσεις των ενδιάμεσων κόμβων ενός μονοπατιού στην τιμή της αρχικής αξιολόγησης, υπονομεύει τη σπουδαιότητα της πηγής μιας αξιολόγησης.

Εφόσον η αλλοίωση στη φύση της εμπιστοσύνης είναι δεδομένη θα πρέπει αυτή να περιοριστεί εντός κάποιων λογικών ορίων. Ένας τρόπος για να επιτευχθεί αυτό είναι με τον περιορισμό του μέγιστου μήκους ενός μονοπατιού εμπιστοσύνης. Ο περιορισμός αυτός επιφέρει και ένα ακόμη πλεονέκτημα, την επιτάχυνση των αλγορίθμων υπολογισμού δικτυακών τιμών εμπιστοσύνης καθώς πλέον θα πρέπει καλούνται να επεξεργαστούν λιγότερα βάρη ανά μονοπάτι εμπιστοσύνης.

Στην περίπτωση του TwoHop, τα μονοπάτια εμπιστοσύνης περιορίστηκαν ώστε να έχουν μέγιστο μήκος δύο βημάτων, μεταξύ του κόμβου που υπολογίζει την τιμή εμπιστοσύνης και του παρόχου υπηρεσιών. Ο γράφος που προκύπτει από αυτόν τον περιορισμό, επιτρέπει την περιγραφή συναλλαγών μεταξύ ομότιμων κόμβων, όπου ένας κόμβος είτε έρχεται άμεσα σε επαφή με κάποιον κόμβο (μονοπάτι ενός βήματος) είτε μαθαίνει για κάποιο τρίτο κόμβο μέσω ενός κόμβου με τον οποίο ήρθε άμεσα σε επαφή (μονοπάτι δύο βημάτων).

Διαφοροποίηση τύπων εμπιστοσύνης

Ο γράφος εμπιστοσύνης του TwoHop υποστηρίζει πολλαπλές ακμές μεταξύ γειτονικών κόμβων. Κάθε ακμή περιγράφει ένα διαφορετικό τύπο εμπιστοσύνης και έτσι, διαφοροποιούνται οι βαθμοί που αποδόθηκαν σε κάποιον κόμβο για την παροχή μιας υπηρεσίας (αξιολόγηση), από τους βαθμούς που αποδόθηκαν στον ίδιο κόμβο για την ποιότητα των αξιολογήσεων που έχει κάνει μέχρι στιγμής. Χρησιμοποιώντας τις βαθμολογίες αυτές, ένας κόμβος του δικτύου μπορεί να προστατευθεί από επιθέσεις κακόβουλων κόμβων που παρέχουν ψευδείς αξιολογήσεις ή αξιολογήσεις σύμφωνα με κάποια κακόβουλη πολιτική. Περισσότερες πληροφορίες σχετικά με την αντιμετώπιση αυτής της επίθεσης μπορεί να βρεθούν στην ενότητα 6.7.2.

Δυστυχώς τα περισσότερα δίκτυα εμπιστοσύνης που προτείνονται στη βιβλιογραφία υποστηρίζουν το πολύ δύο τύπους εμπιστοσύνης: την τιμή εμπιστοσύνης μεταξύ δύο κόμβων (αξιολόγηση) και μια εκτίμηση της εγκυρότητας της τιμής αυτής (confidence value). Περιορίζοντας όμως έτσι τις ακμές ενός γράφου δεν δίνεται η δυνατότητα να περιγραφούν σωστά όλες οι υπάρχουσες σχέσεις μεταξύ των κόμβων και πολλές φορές χρησιμοποιούνται κάποια βάρη με καταχρηστικό τρόπο, περιγράφοντας λάθος σχέσεις.

Στο TwoHop, η μέθοδος υπολογισμού της τιμής εμπιστοσύνης για ένα μονοπάτι λαμβάνει υπόψιν της μία ακμή ανά ζεύγος κόμβων. Σε κάθε βήμα του μονοπατιού, επιλέγεται η κατάλληλη ακμή (βάρος) ανάλογα με το ρόλο που έπαιξε ο κόμβος αυτού του βήματος στο μονοπάτι εμπιστοσύνης. Έτσι, αν ένας κόμβος προσέφερε μια κρίση ως προς τις αξιολογήσεις που παρέχει ο κόμβος–αξιολογητής, τότε η κρίση αυτή θα είναι η τιμή που θα

συμπεριληφθεί στην τελική τιμή εμπιστοσύνης και όχι κάποια άλλη βαθμολόγηση που έκανε ο κόμβος αυτός για τις υπηρεσίες του κόμβου–αξιολογητή. Περισσότερες λεπτομέρειες σχετικά με τους ρόλους που παίζουν οι κόμβοι σε ένα δίκτυο εμπιστοσύνης TwoHop μπορεί κανείς να βρει στην ενότητα 6.3. Επίσης, στην ενότητα 6.4 δίνεται μια πιο αναλυτική περιγραφή της μέθοδου που χρησιμοποιείται για τον υπολογισμό των τιμών εμπιστοσύνης.

Τοπικές τιμές εμπιστοσύνης

Κάθε κόμβος στο δίκτυο εμπιστοσύνης TwoHop διατηρεί μια τοπική βάση με βαθμολογίες. Κάθε μία από αυτές τις βαθμολογίες περιγράφει με ποσοτικό τρόπο μια σχέση εμπιστοσύνης που έχει συνάψει ο κόμβος–αξιολογητής με κάποιο άλλο κόμβο του δικτύου. Στο δίκτυο TwoHop οι κόμβοι μοιράζονται μεταξύ τους τις τοπικές αυτές βάσεις, προκειμένου τα δεδομένα τους να μπορούν να χρησιμοποιηθούν κατά τον υπολογισμό των τιμών εμπιστοσύνης προς γνωστούς και άγνωστους παρόχους υπηρεσιών. Η κοινοποίηση των βάσεων δεδομένων με τις βαθμολογίες, επιτρέπει τη δημιουργία μιας κοινής δικτυακής «μνήμης» μέσω της οποίας οι κόμβοι του δικτύου με συνεργατικό τρόπο μπορούν να αναδείξουν έμπιστους παρόχους υπηρεσιών καθώς και αξιολογητές υπηρεσιών.

Οι τιμές που ανακτώνται από βάσεις τρίτων κόμβων δε χρησιμοποιούνται ως έχουν, αλλά διαμορφώνονται σύμφωνα με βάρη (βαθμολογίες) που περιγράφονται στην τοπική βάση δεδομένων. Αν ένας κόμβος δεν έχει δεχθεί κάποια βαθμολόγηση για τις αξιολογήσεις του, τότε οι αξιολογήσεις του δεν μπορούν να επηρεάσουν τους υπόλοιπους κόμβους του δικτύου. Γενικότερα, ο κάθε κόμβος έχει τη δυνατότητα να υπολογίσει μια «τοπική τιμή εμπιστοσύνης» για κάποιον άλλο κόμβο, βάσει στοιχείων που προκύπτουν τόσο από τις εμπειρίες άλλων κόμβων όσο και από τις δικές του. Με τον τρόπο αυτό αναγνωρίζεται το δικαίωμα των κόμβων να διαμορφώνουν διαφορετικές απόψεις για συγκεκριμένους παρόχους υπηρεσιών, καθώς επίσης και το δικαίωμά τους να δημιουργούν συστάδες κόμβων (clusters) με παρόμοιες «πεποιθήσεις». Αυτή η δυνατότητα δεν υπάρχει στα δίκτυα εμπιστοσύνης όπου χρησιμοποιούνται «καθολικές τιμές εμπιστοσύνης», καθώς η τιμή εμπιστοσύνης που αντιστοιχεί σε κάποιο κόμβο είναι η ίδια ανεξαρτήτως του κόμβου που την υπολογίζει. Επίσης, τα δίκτυα αυτά δεν μπορούν να ανταπεξέλθουν στις απαιτήσεις δικτύων ομότιμων κόμβων με πολλά μέλη, καθώς απαιτούν πλήρη γνώση του γράφου εμπιστοσύνης.

Δικτυακό μέτρο εμπιστοσύνης

Στο δίκτυο TwoHop, η μετρική που περιγράφει την εμπιστοσύνη προς ένα πάροχο βασίζεται σε πληροφορίες που έχουν συλλεχθεί από τους κόμβους μιας *ιεραρχίας εμπιστοσύνης*. Η ιεραρχία ξεκινά από τις σχέσεις εμπιστοσύνης που έχει συνάψει ένας κόμβος του δικτύου και επεκτείνεται βάσει των σχέσεων που έχουν συνάψει οι γειτονικοί κόμβοι αυτού. Χρησιμοποιώντας τα βάρη του υποδικτύου που σχηματίζεται από τις παραπάνω σχέσεις, μπορεί κανείς να υπολογίσει ένα *δικτυακό μέτρο εμπιστοσύνης*.

Τα περισσότερα δίκτυα εμπιστοσύνης δεν επιβάλλουν περιορισμούς στη σύνθεση της ιεραρχίας και έτσι υπολογίζουν δικτυακές τιμές εμπιστοσύνης βάσει εκτιμήσεων από κόμβους που ανήκουν στο ευρύτερο δίκτυο εμπιστοσύνης. Στην περίπτωση του TwoHop, η ιεραρχία αυτή αποτελεί μια «γειτονιά» κόμβων με μονοπάτια δύο το πολύ βημάτων. Η μικρή «απόσταση» μεταξύ των κόμβων στο γράφο της ιεραρχίας, εξασφαλίζει ότι τα βάρη και οι αξιολογήσεις που θα ληφθούν υπόψιν κατά τον υπολογισμό της τιμής εμπιστοσύνης, θα προέρχονται από κόμβους με στενές σχέσεις εμπιστοσύνης. Έτσι αποφεύγεται σε μεγάλο βαθμό η εξέταση ψευδών ή κακόβουλων βαρών που προέρχονται από τυχαίους κόμβους του ευρύτερου δικτύου εμπιστοσύνης. Επίσης, με τον περιορισμό αυτό δίνεται περισσότε-

ρη έμφαση στις απόψεις κόμβων που ενδεχομένως συμμετέχουν στην ίδια «κλίκα» με το χρήστη.

Τέλος, η εξέταση ενός περιορισμένου υποδικτύου του συνολικού δικτύου εμπιστοσύνης επιτρέπει τη γρηγορότερη εκτέλεση του αλγορίθμου υπολογισμού εμπιστοσύνης (καθώς μειώνεται η πολυπλοκότητα αυτού), δίνοντας έτσι τη δυνατότητα στο χρήστη να λάβει άμεσα μια εκτίμηση για την ποιότητα των υπηρεσιών ενός παρόχου.

Εκμετάλλευση πολλαπλών μονοπατιών

Ο αλγόριθμος υπολογισμού εμπιστοσύνης του TwoHop, λαμβάνει υπόψιν του τα βάρη που προκύπτουν από πολλαπλά μονοπάτια εμπιστοσύνης. Όμως, σε αντίθεση με άλλα δίκτυα εμπιστοσύνης, δεν αξιοποιεί την πληροφορία που προκύπτει από όλα τα μονοπάτια.

Συγκεκριμένα, κάθε αξιολόγηση της υπηρεσίας ενός παρόχου λαμβάνεται υπόψιν μία φορά και σε περίπτωση που αυτή συνδέεται με περισσότερα του ενός μονοπάτια, χρησιμοποιούνται τα βάρη του μονοπατιού με το μικρότερο μήκος. Με τον τρόπο αυτό δίνεται έμφαση στις αξιολογήσεις κόμβων που έχουν στενότερη σχέση με τον κόμβο που υπολογίζει την τιμή εμπιστοσύνης. Επίσης, αν δύο μονοπάτια καταλήγουν στην ίδια αξιολόγηση αλλά έχουν το ίδιο μήκος, τότε προτιμούνται τα βάρη του μονοπατιού που ξεκινά με τον πιο έμπιστο κόμβο.

6.3 Βασικές έννοιες της αρχιτεκτονικής TwoHop

Τα μέλη ενός δικτύου εμπιστοσύνης TwoHop παίζουν το ρόλο του *παρόχου* υπηρεσιών, του *αξιολογητή* υπηρεσιών, του *κριτή* αξιολογήσεων καθώς και του *επιθεωρητή* κρίσεων σχετικών με ομάδες αξιολογητών. Ένα μέλος μπορεί να αναλάβει ανά πάσα στιγμή παραπάνω από έναν ρόλους, ανάλογα με τις δραστηριότητες αυτού στο δίκτυο TwoHop.

Ο *πάροχος* υπηρεσιών είναι ένα μέλος του δικτύου που παρέχει υπηρεσίες σε άλλα μέλη αυτού. Για κάθε τύπο υπηρεσίας έχει καθιερωθεί ένα χαρακτηριστικό κωδικό όνομα (identifier) s, το οποίο περιγράφει σαφώς τον τύπο της υπηρεσίας. Η μορφή αυτού εξαρτάται από τις ανάγκες της εφαρμογής που αξιοποιεί το δίκτυο TwoHop (π.χ. κείμενο – "ζωντανή μετάδοση video", URL – "`http: //srv. org/type`", αριθμός – 0x34123 κλπ.).

Το ρόλο του *αξιολογητή* υπηρεσιών παίζει οποιοδήποτε μέλος του δικτύου έχει χρησιμοποιήσει κάποια υπηρεσία και έχει βαθμολογήσει την ποιότητα αυτής με μια εγγραφή (αξιολόγηση) σαν την παρακάτω:

$$[i,\ s,\ v],\ i \in I, s \in S, v \in (0,1),$$

όπου v είναι ο βαθμός που έλαβε η υπηρεσία s του παρόχου i κατά την αξιολόγηση. Το σύνολο I περιέχει όλες τις δυνατές *ταυτότητες* των μελών του δικτύου, ενώ το σύνολο S περιέχει όλους τους δυνατούς τύπους υπηρεσιών.

Ένα μέλος του δικτύου που λειτουργεί ως *κριτής*, βαθμολογεί τις αξιολογήσεις που έχουν κάνει μέλη του δικτύου για ένα συγκεκριμένο τύπο υπηρεσίας. Οι κρίσεις αυτές αποθηκεύονται σε εγγραφές της μορφής:

$$[i,\ s,\ a],\ i \in I, s \in S, a \in (0,1),$$

όπου a είναι ο βαθμός που έλαβε ο αξιολογητής i για αξιολογήσεις που έκανε σε παρόχους που προσέφεραν τον τύπο υπηρεσίας s.

Οι *επιθεωρητές* ελέγχουν την ποιότητα και την ποσότητα των βαθμολογήσεων των κριτών και επιβραβεύουν κριτές που ανακαλύπτουν νέους αξιόπιστους αξιολογητές. Οι βαθμολογήσεις των επιθεωρητών έχουν τη μορφή:

$$[i,\ s,\ r],\ i \in I, s \in S, r \in (0,1),$$

όπου r είναι ο βαθμός που αποδόθηκε στον κριτή i για τις κρίσεις που έκανε σχετικά με αξιολογητές της υπηρεσίας s.

Κάθε μέλος του δικτύου εμπιστοσύνης TwoHop διατηρεί μια συλλογή C από *πορτφόλιο εμπιστοσύνης*, με ένα πορτφόλιο P_s ανά τύπο υπηρεσίας s:

$$C = (P_j, \ldots, P_k), j, k \in S, j \neq k$$

Το πορτφόλιο εμπιστοσύνης περιέχει τις αξιολογήσεις, κρίσεις και επιθεωρήσεις που έχει κάνει το μέλος – ιδιοκτήτης του και έχει τη μορφή:

$$P_s = \{(i_0, r_0, a_0, v_0), \ldots, (i_k, r_k, a_k, v_k)\},\ k \in \mathbb{N}^*, s \in S,$$

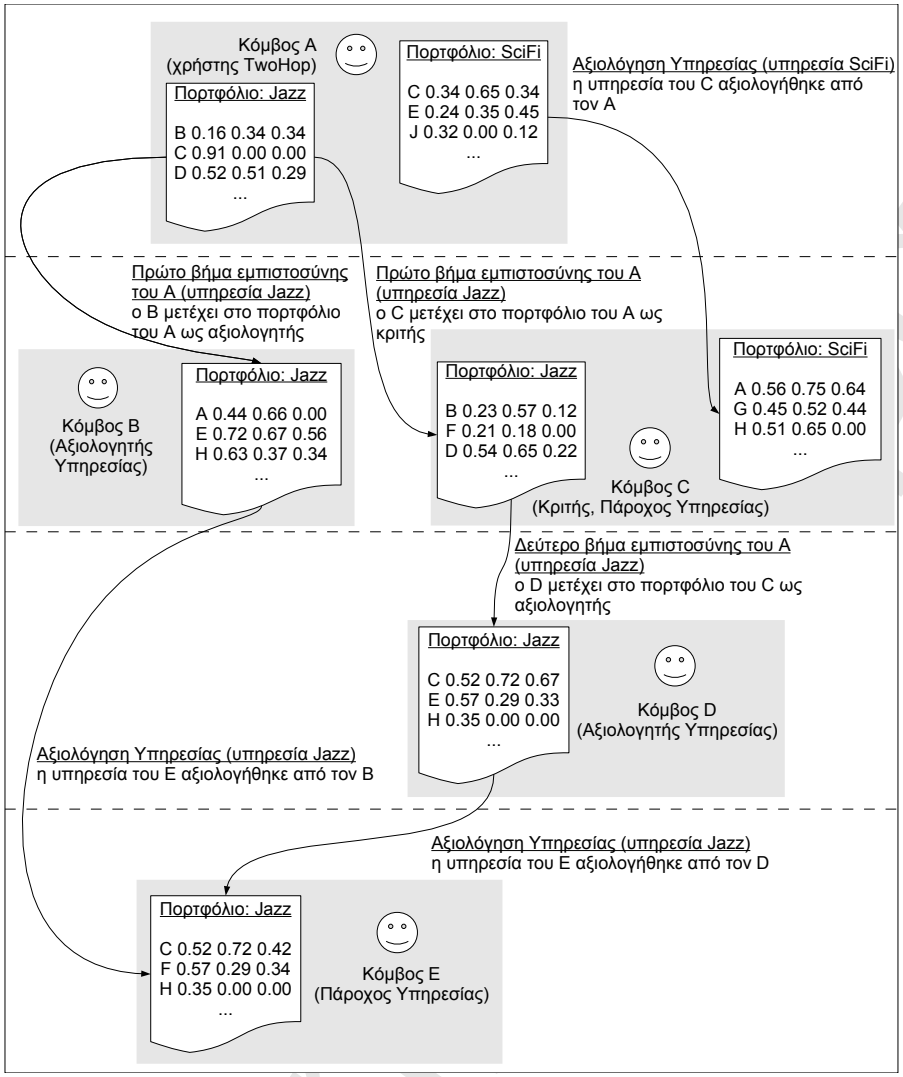
όπου v_k είναι η αξιολόγηση της υπηρεσίας τύπου s που προσφέρει ο πάροχος i_k, a_k είναι η κρίση των αξιολογήσεων που προσέφερε το μέλος i_k και r_k είναι η επιθεώρηση των κρίσεων του μέλους i_k για αξιολογήσεις υπηρεσιών τύπου s. Σε περίπτωση που κάποια από αυτές τις τιμές (r, a, v) δεν είναι διαθέσιμη, αρχικοποιείται με 0.

Ο τρόπος και η συχνότητα με την οποία ανανεώνονται αυτές οι τιμές εξαρτώνται από το είδος της εφαρμογής που αξιοποιεί το δίκτυο εμπιστοσύνης. Για παράδειγμα, στην ενότητα 6.6 θα παρουσιαστεί μια εφαρμογή στην οποία ο χρήστης δεν ανανεώνει άμεσα τις εγγραφές του πορτφόλιο αλλά οι ίδιες ανανεώνονται αυτόματα σε κάθε αλληλεπίδραση του χρήστη με τις υπηρεσίες των παρόχων. Περισσότερες πληροφορίες σχετικά με στρατηγικές ανανέωσης των βαρών σε δίκτυα εμπιστοσύνης μπορεί κανείς να βρει στην εργασία [188] των Uddin et al., όπου μαζί με τα προτερήματα και τα μειονεκτήματα των διαφόρων στρατηγικών που προτείνονται, περιγράφονται και τύποι εφαρμογών που ταιριάζουν σε κάθε στρατηγική.

Στο σχήμα 6.2 παρουσιάζεται ένα παράδειγμα δικτύου εμπιστοσύνης TwoHop, όπου φαίνονται συνοπτικά τα πορτφόλιο των κόμβων, οι ρόλοι τους καθώς και οι μεταξύ τους σχέσεις.
Οι τρεις τιμές που απεικονίζονται σε κάθε πορτφόλιο είναι οι βαθμολογίες που αποδίδονται σε τρίτους κόμβους που έπαιξαν το ρόλο των κριτών, των αξιολογητών και των παρόχων. Συγκεκριμένα, οι κόμβοι B και D είναι αξιολογητές της υπηρεσίας τύπου 'Jazz' που παρέχεται από τον κόμβο E. Για να υπολογίσει ο κόμβος A την τιμή εμπιστοσύνης για την υπηρεσία αυτή, θα πρέπει να χρησιμοποιήσει την πληροφορία που περιέχεται σε δύο μονοπάτια εμπιστοσύνης ($A \rightarrow B \rightarrow E$ και $A \rightarrow C \rightarrow D \rightarrow E$). Η τιμή εμπιστοσύνης του πρώτου μονοπατιού προκύπτει από την κρίση του κόμβου B από τον A αλλά και από την αξιολόγηση της υπηρεσίας του E από τον B. Αντίστοιχα, στο δεύτερο μονοπάτι συνυπολογίζεται η βαθμολογία που έλαβε ο κριτής C από τον κόμβο A, η βαθμολογία που έλαβε ο αξιολογητής D από τον κριτή C καθώς και η βαθμολογία που έλαβε ο πάροχος E από τον αξιολογητή D. Στο σχήμα παρουσιάζεται και μια ακόμη ιεραρχία εμπιστοσύνης η οποία αφορά στις υπηρεσίες τύπου 'SciFi'. Σε αυτή την ιεραρχία ο κόμβος C συμμετέχει ως πάροχος υπηρεσιών, ενώ ο κόμβος A ως αξιολογητής των υπηρεσιών αυτών.

6.4 Μέθοδος υπολογισμού εμπιστοσύνης στην αρχιτεκτονική TwoHop

Ο αλγόριθμος υπολογισμού εμπιστοσύνης της αρχιτεκτονικής TwoHop υλοποιείται από τη συνάρτηση `calculateTrust(`P_{root}`, i, s)`. Η συνάρτηση αυτή δέχεται ως παραμέτρους το



Σχήμα 6.2: Παράδειγμα των σχέσεων μεταξύ κόμβων σε ένα δίκτυο εμπιστοσύνης TwoHop

βασικό πορτφόλιο P_{root} από το οποίο θα ξεκινήσει η αναζήτηση, την ταυτότητα του παρόχου i και το χαρακτηριστικό κωδικό της υπηρεσίας s που παρέχει, για την οποία θα υπολογιστεί η τιμή εμπιστοσύνης. Για να υπολογίσει η `calculateTrust` αυτή την τιμή χρησιμοποιεί δύο βοηθητικές συναρτήσεις:

- `getPortfolio(`i`,` s`)`: Υπεύθυνη για την εύρεση, μεταφορά και τοπική αποθήκευση του πορτφόλιο που διατηρεί ο χρήστης i για τον τύπο υπηρεσίας s. Επιστρέφει τα περιεχόμενα του πορτφόλιο αυτού.
- `peerExists(`P`,` i`)`: Επιστρέφει την Bool τιμή 'Αληθές' αν το πορτφόλιο P περιέχει κάποια εγγραφή που αφορά τον κόμβο i.

Το *βασικό πορτφόλιο* P_{root} (root trust portfolio) είναι, όπως προαναφέρθηκε, το πορτφόλιο από το οποίο θα ξεκινήσει η αναζήτηση για στοιχεία σχετικά με την υπηρεσία s του παρόχου i. Το πορτφόλιο αυτό είναι συνήθως το προσωπικό πορτφόλιο του χρήστη που επιθυμεί να υπολογίσει την τιμή εμπιστοσύνης για την υπηρεσία του παρόχου. Σε περίπτωση που ο χρήστης είναι νέο μέλος του δικτύου TwoHop και δεν έχει (αρκετές)

εγγραφές στο πορτφόλιο του σχετικά με το συγκεκριμένο τύπο υπηρεσίας, τότε μέσω της `getPortfolio` μπορεί να προμηθευθεί το πορτφόλιο μιας τρίτης έμπιστης οντότητας και να χρησιμοποιήσει αυτό σαν βασικό πορτφόλιο P_{root} (portfolio bootstrapping).

Αλγόριθμος 7 Υπολ. Εμπιστοσύνης (συνάρτηση `calculateTrust`)

Require: $P_{root} \neq \emptyset, i \in I, s \in S$
  $sum_{trust} \Leftarrow 0$, $num_{evals} \Leftarrow 0$, $old_{review0} \Leftarrow \emptyset$, $old_{assess1} \Leftarrow \emptyset$
  for all $entry_0 \in P_{root}$ **do** {Βρόχος Βασικού Πορτφόλιο}
    $peer_0 \Leftarrow entry_0[peer\_id]$
    $assess_0 \Leftarrow entry_0[assessment]$
    $review_0 \Leftarrow entry_0[review]$
    if $peer_0 = i$ **then**
      continue {Βρόχος Βασικού Πορτφόλιο}
    end if
    $P_{onehop} \Leftarrow getPortfolio(peer_0, s)$
    for all $entry_1 \in P_{onehop}$ **do** {Βρόχος $1^{ου}$ βήματος}
      if $entry_1[peer\_id] = i$ **then**
        $sum_{trust} \Leftarrow sum_{trust} + (assess_0 \times entry_1[evaluation])$
        $num_{evals} \Leftarrow num_{evals} + 1$
      else
        $peer_1 \Leftarrow entry_1[peer\_id]$
        $assess_1 \Leftarrow entry_1[assessment]$
        if $peerExists(P_{root}, peer_1) = TRUE$ **then**
          continue {Βρόχος $1^{ου}$ βήματος}
        end if
        if $peer_1 \notin old_{review0}$ **then**
          $old_{review0} \Leftarrow old_{review0} \cup (peer_1)$
          $old_{review0}[peer_1] \Leftarrow 0$
          $old_{assess1} \Leftarrow old_{assess1} \cup (peer_1)$
          $old_{assess1}[peer_1] \Leftarrow 0$
        end if
        $P_{twohop} \Leftarrow getPortfolio(peer_1, s)$
        for all $entry_2 \in P_{twohop}$ **do** {Βρόχος $2^{ου}$ βήματος}
          if $entry_2[peer\_id] = i$
          AND $old_{review0}[peer_1] < review_0$ **then**
            $sum_{trust} \Leftarrow sum_{trust} + (review_0 \times$
              $assess_1 \times entry_2[evaluation]) -$
                $(old_{review0}[peer_1] \times old_{assess1}[peer_1] \times$
                  $entry_2[evaluation])$
            if $old_{review0}[peer_1] = 0$ **then**
              $num_{evals} \Leftarrow num_{evals} + 1$
            end if
            $old_{review0}[peer_1] \Leftarrow review_0$
            $old_{assess1}[peer_1] \Leftarrow assess_1$
          end if
        end for {Βρόχος $2^{ου}$ βήματος}
      end if
    end for {Βρόχος $1^{ου}$ βήματος}
  end for {Βρόχος Βασικού Πορτφόλιο}
  if $num_{evals} \neq 0$ **then**
    return $\frac{sum_{trust}}{num_{evals}}$
  else
    return 0
  end if

Ο αλγόριθμος υπολογισμού εμπιστοσύνης του TwoHop αποτελείται από τρεις εμφωλευμένους βρόχους. Στον εξωτερικό βρόχο (βλ. αλγόριθμο 7, «Βρόχος Βασικού Πορτφόλιο») αναζητώνται στο πορτφόλιο P_{root} εγγραφές που αφορούν αξιολογητές και κριτές, σχετικούς με τον τύπο υπηρεσίας που παρέχει ο πάροχος i. Οι εγγραφές αυτές θα χρησιμοποιηθούν ως αναφορές σε άλλα πορτφόλιο που θα εξετάσει ο αλγόριθμος στον αμέσως επόμενο εμ-

φωλευμένο βρόχο. Προηγούμενες αξιολογήσεις του ίδιου του ιδιοκτήτη του P_{root} για την υπηρεσία του i δε λαμβάνονται υπ'όψιν σε αυτό το σημείο (θα ληφθούν όμως υπ'όψιν σε επόμενη φάση, όπως θα φανεί παρακάτω). Επίσης, αν βρεθεί αναφορά στο πορτφόλιο του παρόχου i, αυτή δε θα ληφθεί υπ'όψιν καθώς το πορτφόλιο αυτό ενδέχεται να περιέχει υποκειμενικές αξιολογήσεις του παρόχου για την υπηρεσία του, καθώς και αναφορές σε «φιλικούς» (ως προς τον πάροχο) κόμβους, οι οποίοι ενδεχομένως να παρέχουν ψευδή στοιχεία για την ποιότητα υπηρεσιών αυτού.

Στόχος του πρώτου εμφωλευμένου βρόχου (βλ. «Βρόχος $1^{ου}$ βήματος») είναι να εξετάσει τα πορτφόλιο που συλλέχθηκαν από τις αναφορές του P_{root}. Τα πορτφόλιο αυτά (P_{onehop}) αποτελούν στην ουσία το αποτέλεσμα του πρώτου βήματος εμπιστοσύνης (1^{st} trust hop) από το βασικό πορτφόλιο P_{root}. Ο αλγόριθμος χρησιμοποιεί τη συνάρτηση `getPortfolio` για να μεταφέρει τοπικά την πληροφορία αυτών των πορτφόλιο και τα εξετάζει αναζητώντας εγγραφές δύο ειδών: α) αξιολογήσεις της υπηρεσίας s του παρόχου i και β) κρίσεις αξιολογητών για αυτό τον τύπο υπηρεσίας. Αν βρεθεί κάποια αξιολόγηση για την εν λόγω υπηρεσία στο πορτφόλιο, έστω, του $peer_0$, τότε η τιμή αυτής πολλαπλασιάζεται με την καταγεγραμμένη στο βασικό πορτφόλιο κρίση για τον $peer_0$ ($assess_0$) και το αποτέλεσμα προστίθεται στο άθροισμα sum_{trust}. Κάθε αξιολόγηση που συμπεριλαμβάνεται σε αυτό το άθροισμα αυξάνει τον αριθμό των καταγεγραμμένων αξιολογήσεων num_{evals} κατά μία μονάδα.

Οι κρίσεις που βρέθηκαν στα πορτφόλιο P_{onehop} θα χρησιμοποιηθούν ως αναφορές σε πορτφόλιο που βρίσκονται σε απόσταση δύο βημάτων εμπιστοσύνης από το βασικό πορτφόλιο (P_{twohop}). Συγκεκριμένα, από τις κρίσεις αυτές, θα αξιοποιηθούν μονάχα εκείνες που αναφέρονται σε κόμβους οι οποίοι δεν περιλαμβάνονται στο βασικό πορτφόλιο (έλεγχος μέσω της `peerExists`). Με τον τρόπο αυτό, αν το πορτφόλιο ενός κόμβου εμφανίζεται τόσο σε πρώτο όσο και σε δεύτερο βήμα εμπιστοσύνης, τότε προτιμάται να εξεταστεί με τα βάρη του βασικού πορτφόλιο (ως κόμβος πρώτου βήματος) και όχι μέσω βαρών που έθεσαν τρίτοι. Επίσης, με τον έλεγχο αυτό αποφεύγεται η διπλή εξέταση ενός πορτφόλιο όταν ένας κόμβος πρώτου βήματος περιλαμβάνει στο πορτφόλιο του αναφορά που περιγράφει τον ίδιο.

Ο δεύτερος εμφωλευμένος βρόχος (βλ. «Βρόχος $2^{ου}$ βήματος») αναζητά στα πορτφόλιο που βρίσκονται σε απόσταση δύο βημάτων εμπιστοσύνης από το βασικό πορτφόλιο, αξιολογήσεις για την υπηρεσία s του παρόχου i. Αν μια τέτοια αξιολόγηση βρεθεί στο πορτφόλιο ενός κόμβου $peer_1$, τότε η τιμή της πολλαπλασιάζεται με την κρίση του κόμβου $peer_0$ για τον $peer_1$ ($assess_1$) εξαιτίας της οποίας βρέθηκε αυτή η αξιολόγηση, καθώς και με τους βαθμούς επιθεώρησης ($review_0$) που έχει λάβει ο $peer_0$ στο βασικό πορτφόλιο. Όπως και στον προηγούμενο βρόχο, έτσι και εδώ, οι τελικές τιμές των αξιολογήσεων προστίθενται στο άθροισμα sum_{trust} και για κάθε αξιολόγηση που προστίθεται, αυξάνεται κατά ένα ο αριθμός των καταγεγραμμένων αξιολογήσεων num_{evals}. Αν ο αλγόριθμος φτάσει σε κάποια αξιολόγηση μέσω δύο μονοπατιών εμπιστοσύνης τότε προτιμάται το μονοπάτι που ξεκινά με τον πιο έμπιστο κόμβο (δηλ. τον κόμβο που έχει τον υψηλότερο βαθμό επιθεώρησης στο βασικό πορτφόλιο).

Η τελική τιμή εμπιστοσύνης T που επιστρέφει η συνάρτηση `calculateTrust` προέρχεται από το μέσο όρο των (σταθμισμένων) αξιολογήσεων που συλλέχθηκαν από το δίκτυο εμπιστοσύνης του χρήστη (ακτίνας δύο βημάτων). Δηλαδή,

$$T = \frac{sum_{trust}}{num_{evals}} = \frac{\sum_{j=1}^{|V_{i,s}|} w_j v_j}{|V_{i,s}|}, \tag{6.1}$$

$$\text{με } w_j = \begin{cases} r_j a_j & \text{αν η } v_j \text{ συλλέχθηκε από κόμβο } 2^{\text{ου}} \text{ βήματος} \\ a_j & \text{αν η } v_j \text{ συλλέχθηκε από κόμβο } 1^{\text{ου}} \text{ βήματος} \end{cases},$$

$$\text{και } i \in I,\ s \in S,\ v_j \in V_{i,s},\ r_j, a_j, v_j \in (0,1),\ T \in (0,1).$$

Το σύνολο $V_{i,s}$ περιέχει όλες τις αξιολογήσεις v_j που συλλέχθηκαν από το δίκτυο εμπιστοσύνης και αφορούν στην υπηρεσία s του παρόχου i. Τα βάρη r_j και a_j αντιστοιχούν στους πόντους επιθεώρησης και κρίσης που συνδέονται με το μονοπάτι εμπιστοσύνης που οδήγησε στην αξιολόγηση v_j.

Όπως διευκρινίστηκε και παραπάνω, κατά το «Βρόχο Βασικού Πορτφόλιο» δε λαμβάνονται υπόψιν τυχόν αξιολογήσεις της υπηρεσίας s του παρόχου i που έγιναν από τον κάτοχο του βασικού πορτφόλιο. Οι αξιολογήσεις αυτές όμως, ενδέχεται να ενσωματωθούν στο αποτέλεσμα της συνάρτησης υπολογισμού εμπιστοσύνης, αν στο πορτφόλιο κάποιου τρίτου κόμβου (πρώτου βήματος) βρεθεί αναφορά στο βασικό πορτφόλιο. Αυτό σημαίνει πρακτικά ότι οι αξιολογήσεις ενός οποιουδήποτε κόμβου έχουν σημασία μόνο όταν αυτές αναγνωρίζονται και από άλλους κόμβους του δικτύου.

Σε περίπτωση όπου δεν εντοπιστεί κάποιο μονοπάτι εμπιστοσύνης για την υπηρεσία s του παρόχου i τότε (εφόσον το επιτρέπει ο τύπος της εφαρμογής) μπορεί να χρησιμοποιηθεί ως τιμή εμπιστοσύνης το αποτέλεσμα μιας παλαιότερης (τοπικής) αξιολόγησης της υπηρεσίας. Εάν στο παρελθόν δεν είχε συμβεί μια τέτοια αξιολόγηση, τότε ο ενδιαφερόμενος κόμβος μπορεί να προχωρήσει σε μια σειρά δοκιμαστικών αλληλεπιδράσεων με τον πάροχο ώστε να υπολογίσει την αρχική τιμή της σχετικής αξιολόγησης.

Η ανίχνευση μιας αξιολόγησης για ένα συγκεκριμένο πάροχο σε πορτφόλιο δεύτερου βήματος δεν απαιτεί σειριακή εξέταση όλων των εγγραφών αυτού του πορτφόλιο. Εφόσον κάθε πορτφόλιο περιγράφει τιμές που αφορούν σε ένα συγκεκριμένο είδος υπηρεσίας, η εγγραφή για έναν πάροχο θα είναι μοναδική (αν υπάρχει) και θα μπορούσε να ανακτηθεί σε χρόνο $O(1)$ χρησιμοποιώντας κάποια τεχνική ευρετηριοποίησης (indexing). Με την τεχνική αυτή, στην ουσία καταργούνται οι επαναλήψεις του δεύτερου εμφωλευμένου βρόχου στον αλγόριθμο 7 και μειώνεται σημαντικά η πολυπλοκότητα του αλγορίθμου αυτού.

Αν κάθε κόμβος διαθέτει ένα πορτφόλιο με μέσο όρο εγγραφών k, τότε η συνάρτηση `calculateTrust` θα ελέγξει κατά μέσο όρο k εγγραφές στο βασικό πορτφόλιο και για κάθε μία από αυτές, θα πρέπει να ελέγξει επίσης k εγγραφές στο πορτφόλιο πρώτου βήματος στο οποίο αναφέρεται. Συνεπώς, χρησιμοποιώντας την παραπάνω τεχνική ευρετηριοποίησης, ο μέσος χρόνος εκτέλεσης του αλγορίθμου γίνεται $O(k^2)$, θεωρώντας ότι οι συναρτήσεις `peerExists` και `getPortfolio` ολοκληρώνονται σε χρόνο $O(1)$.

Σε μια εφαρμογή όπου και τα τρία πορτφόλιο (βασικό, πρώτου βήματος, και δεύτερου βήματος) βρίσκονται στη μνήμη ταυτόχρονα, ο αλγόριθμος θα χρειαστεί τουλάχιστον $3 \cdot \lambda \cdot k$ μονάδες μνήμης για να αποθηκεύσει τα δεδομένα των πορτφόλιο (όπου λ είναι το μέγεθος μιας εγγραφής σε μονάδες μνήμης). Συνεπώς, η μνήμη που απαιτείται από τον αλγόριθμο για τον υπολογισμό τιμών εμπιστοσύνης είναι κατά μέσο όρο $O(k)$. Στο σημείο αυτό θα πρέπει να σημειωθεί ότι ο αριθμός των εγγραφών k περιορίζεται από τον αριθμό των επαφών ενός κόμβου ανά τύπο υπηρεσίας. Αν π.χ. ένα πορτφόλιο εμπιστοσύνης αντιπροσωπεύει τις επαφές που έχει ένας χρήστης σε ένα πραγματικό κοινωνικό δίκτυο, τότε το μέγεθος z αυτού του δικτύου λειτουργεί ως άνω φράγμα για την τιμή του k, δηλ. $k \in [0, z], k \in \mathbb{N}, z \in \mathbb{N}$. Σύμφωνα με την εργασία [189] των Hill και Dunbar, για τα κοινωνικά δίκτυα ανθρώπων ισχύει ότι $\ln z \in (1, 6)$ και $\lceil\ \overline{z}\ \rceil = 154$.

Επειδή η αρχιτεκτονική TwoHop περιορίζεται στη συλλογή στοιχείων από πορτφόλιο που βρίσκονται σε ακτίνα δύο, το πολύ, βημάτων εμπιστοσύνης από το βασικό πορτφόλιο, υπάρχει η πιθανότητα κάποια αξιολόγηση που ανήκει σε κόμβο του δικτύου εμπιστοσύνης να μη ληφθεί υπόψιν από την `calculateTrust`. Συγκεκριμένα, σε ένα δίκτυο με N

κόμβους, όπου ο καθένας διατηρεί πορτφόλιο k εγγραφών, η πιθανότητα να βρεθεί μια εγγραφή που αφορά σε έναν τυχαίο κόμβο α σε υποδίκτυο ακτίνας δύο βημάτων είναι:

$$P(\alpha) = 1 - (\frac{N-k}{N})^{k+k^2} \tag{6.2}$$

Αυτό σημαίνει ότι σε ένα μεγάλο δίκτυο με 1.000.000 κόμβους, όπου ο κάθε κόμβος διατηρεί πορτφόλιο 150 εγγραφών, υπάρχει 96% πιθανότητα να εντοπιστεί μια εγγραφή που αφορά σε έναν τυχαίο κόμβο του δικτύου αυτού. Η απόδειξη του τύπου (6.2) δίνεται στο παράρτημα Α΄.2.

6.5 Αλγεβρική περιγραφή των χαρακτηριστικών της αρχιτεκτονικής TwoHop

Στην εργασία [190] χρησιμοποιήθηκε η αλγεβρική μέθοδος περιγραφής αλγορίθμων υπολογισμού εμπιστοσύνης των Theodorakopoulos και Baras [185] για να αναδειχθούν και να εξεταστούν τα ιδιαίτερα χαρακτηριστικά του αλγορίθμου TwoHop.

Οι Theodorakopoulos και Baras αναφέρουν ότι παρόλο που οι αλγόριθμοι υπολογισμού εμπιστοσύνης διαφέρουν μεταξύ τους στην υλοποίηση, στα είδη των βαρών που εφαρμόζονται στο γράφο εμπιστοσύνης αλλά και στη συνάρτηση υπολογισμού εμπιστοσύνης, παρατηρείται κάποια ομοιότητα στον τρόπο με τον οποίο διαχειρίζονται τα μονοπάτια εμπιστοσύνης. Συγκεκριμένα, κατά τον υπολογισμό της έμμεσης εμπιστοσύνης μεταξύ δύο μελών ενός δικτύου εμπιστοσύνης, πραγματοποιούνται δύο βασικές διαδικασίες: ο *συγκερασμός βαρών* και ο *συγκερασμός τιμών εμπιστοσύνης*.

Για να περιγραφούν αλγεβρικά οι δύο αυτές διαδικασίες, εισάγονται δύο νέοι τελεστές:

- ο τελεστής *συνένωσης* $\otimes$, ο οποίος κατά τον συγκερασμό βαρών, συνδυάζει τα βάρη από γειτονικές ακμές κατά μήκος ενός μονοπατιού εμπιστοσύνης, και
- ο τελεστής *περίληψης* $\oplus$, ο οποίος κατά τον συγκερασμό τιμών εμπιστοσύνης, συνδυάζει τις τιμές εμπιστοσύνης που προέρχονται από διαφορετικά μονοπάτια.

Οι πράξεις που εκτελούν οι τελεστές αυτοί συναντώνται σε όλους τους αλγόριθμους υπολογισμού εμπιστοσύνης, με διαφορετική όμως υλοποίηση. Θα μπορούσε λοιπόν κανείς να συγκρίνει μια σειρά αλγορίθμων εμπιστοσύνης χρησιμοποιώντας ως μέτρο σύγκρισης την υλοποίηση των παραπάνω τελεστών καθώς και τον τύπο των τιμών στις οποίες επενεργούν. Επίσης, χρησιμοποιώντας τους παραπάνω τελεστές, μπορεί κανείς να περιγράψει μια σειρά προϋποθέσεων τις οποίες θα πρέπει να πληροί ένας αλγόριθμος υπολογισμού εμπιστοσύνης προκειμένου να καλύψει τις απαιτήσεις μιας εφαρμογής.

Δυστυχώς, το αλγεβρικό μοντέλο που προτείνεται στην εργασία [185] θεωρεί ότι δύο κόμβοι συνδέονται άμεσα μέσω μίας το πολύ ακμής (και συνεπώς σχέσης). Για να μπορέσουν να μελετηθούν πιο σύνθετα δίκτυα, όπως το TwoHop, με το μοντέλο αυτό, έπρεπε να εξομοιωθούν με κάποιο τρόπο οι πολλαπλές σχέσεις που μπορεί να είχαν δύο άμεσα συνδεδεμένοι κόμβοι του δικτύου. Αυτό έγινε με την εφαρμογή πολλαπλών βαρών ανά ακμή του γράφου εμπιστοσύνης.

Έστω ότι το σύνολο V_i^j περιέχει όλες τις αξιολογήσεις μιας υπηρεσίας ενός κόμβου–παρόχου j, που βρέθηκαν σε απόσταση δύο το πολύ βημάτων από το πορτφόλιο του κόμβου i. Για κάθε αξιολόγηση $v_k \in V_i^j$, υπάρχει ένα μονοπάτι p_k το οποίο ξεκινά από τον κόμβο i και καταλήγει στην αξιολόγηση αυτή. Το μονοπάτι p_k αποτελείται από κατευθυνόμενες ακμές τύπου (x, y), όπου ο κόμβος x αποτελεί το σημείο έναρξης της κατευθυνόμενης ακμής και ο

κόμβος y το σημείο λήξης αυτής. Στο σύστημα TwoHop, κάθε ακμή $e : (x, y)$ συνδέεται με (το πολύ) τρία βάρη: τους πόντους επιθεώρησης ($r_{x,y}$), κρίσης ($a_{x,y}$) και αξιολόγησης ($v_{x,y}$) που ο κόμβος x έχει απονείμει στον κόμβο y για τις υπηρεσίες του (βλ. ενότητα 6.3).

Για να υπολογιστεί η εμπιστοσύνη προς την υπηρεσία ενός παρόχου με δεδομένα από ένα μονοπάτι εμπιστοσύνης, θα πρέπει να συνδυαστούν τα κατάλληλα βάρη. Συγκεκριμένα, αν ονομάσουμε $w^j_{x,y}$ το βάρος με το οποίο η ακμή (x, y) θα συμβάλλει στον υπολογισμό για τον πάροχο j, τότε έχουμε:

$$w^j_{x,y} = \begin{cases} v_{x,y} & \text{αν ο } y \text{ είναι ο πάροχος } j \text{ και ο } x \text{ είναι αξιολογητής αυτού} \\ a_{x,y} & \text{αν ο } x \text{ είναι κριτής για τον αξιολογητή } y \\ r_{x,y} & \text{αν ο } x \text{ είναι επιθεωρητής για τον κριτή } y \end{cases} \tag{6.3}$$

Στην αρχιτεκτονική TwoHop, ο συνδυασμός βαρών κατά μήκος ενός μονοπατιού εμπιστοσύνης γίνεται με πολλαπλασιασμό. Συγκεκριμένα, ο βαθμός εμπιστοσύνης προς την υπηρεσία του παρόχου j, όπως αυτή περιγράφεται από τις ακμές του μονοπατιού p_k : $\{(x_0, x_1), (x_1, x_2), (x_2, x_3)\}$, θα είναι:

$$t^j_{p_k} = w^j_{x_0,x_1} \otimes w^j_{x_1,x_2} \otimes w^j_{x_2,x_3} = r^j_{x_0,x_1} \cdot a^j_{x_1,x_2} \cdot v^j_{x_2,x_3}. \tag{6.4}$$

Επειδή $w^j_{x,y} \in (0, 1)$ συνεπάγεται ότι $t^j_{p_k} \in (0, 1)$.

Το μέτρο εμπιστοσύνης προς ένα πάροχο, όπως αυτό καταγράφεται από ένα δίκτυο εμπιστοσύνης, προκύπτει από την περίληψη των μονοπατιών εμπιστοσύνης που οδηγούν σε αξιολογήσεις της υπηρεσίας του παρόχου, δηλ.:

$$T = t_{p_1} \oplus t_{p_2} \oplus \ldots \oplus t_{p_n}, \tag{6.5}$$

όπου t_{p_k} είναι η τιμή της εμπιστοσύνης που προκύπτει από το μονοπάτι p_k. Όπως προαναφέρθηκε στην υποενότητα 6.4, ο αλγόριθμος υπολογισμού εμπιστοσύνης που χρησιμοποιείται στο TwoHop, υπολογίζει την περίληψη των μονοπατιών, χρησιμοποιώντας το μέσο όρο των τιμών εμπιστοσύνης που προέκυψαν από αυτά τα μονοπάτια. Δυστυχώς όμως, εάν ο τελεστής περίληψης αναλάβει και τον υπολογισμό του μέσου όρου, τότε δε θα είναι δυνατό να συμπεριληφθεί ο μέσος όρος νέων μονοπατιών στον ήδη υπολογισμένο μέσο όρο παλαιότερων. Για το λόγο αυτό, ο τελεστής περίληψης διαμορφώθηκε κατάλληλα ώστε να μπορεί να υπολογίσει το άθροισμα ζευγών της μορφής $(\bar{t}, n)$, όπου $\bar{t}$ είναι το *άθροισμα* των τιμών εμπιστοσύνης που προέρχονται από n μονοπάτια:

$$(\bar{t}_1, n_1) \oplus (\bar{t}_2, n_2) = (\bar{t}_1 + \bar{t}_2, n_1 + n_2) \tag{6.6}$$

Έτσι, ο τελεστής περίληψης αποκτά την προσεταιριστική ιδιότητα της πρόσθεσης και μπορεί να χρησιμοποιηθεί για να ενσωματώσει την πληροφορία των ήδη επεξεργασμένων μονοπατιών με αυτή που προκύπτει από νέα μονοπάτια, όπως φαίνεται παρακάτω:

$$\begin{aligned} &(\bar{t}_1, n_1) \oplus (\bar{t}_2, n_2) \oplus (\bar{t}_3, n_3) \oplus (\bar{t}_4, n_4) \\ &= \Big((\bar{t}_1, n_1) \oplus (\bar{t}_2, n_2)\Big) \oplus \Big((\bar{t}_3, n_3) \oplus (\bar{t}_4, n_4)\Big) \\ &= \Big((\bar{t}_1, n_1) \oplus (\bar{t}_2, n_2) \oplus (\bar{t}_3, n_3)\Big) \oplus (\bar{t}_4, n_4) \\ &= (\bar{t}_1, n_1) \oplus \Big((\bar{t}_2, n_2) \oplus (\bar{t}_3, n_3) \oplus (\bar{t}_4, n_4)\Big) \\ &= (\bar{t}_1 + \bar{t}_2 + \bar{t}_3 + \bar{t}_4, n_1 + n_2 + n_3 + n_4) \end{aligned} \tag{6.7}$$

Οι συναρτήσεις *accum* και *paths* εξάγουν τα μέλη $\bar{t}$ και n, αντίστοιχα, από ένα ζεύγος $(\bar{t}, n)$:

$$accum((\bar{t}, n)) = \bar{t} \tag{6.8}$$

$$paths((\bar{t}, n)) = n \tag{6.9}$$

Με τη βοήθεια των (6.8) και (6.9), του νέου τελεστή περίληψης (6.6) και της γενικευμένης μορφής του μέτρου εμπιστοσύνης για ένα δίκτυο εμπιστοσύνης (6.5), η συνάρτηση υπολογισμού εμπιστοσύνης του TwoHop μπορεί να γραφεί ως εξής:

$$\begin{aligned} T_i^j &= \frac{accum\Big((t_{p_1}^j, 1) \oplus (t_{p_2}^j, 1) \oplus \ldots \oplus (t_{p_n}^j, 1)\Big)}{paths\Big((t_{p_1}^j, 1) \oplus (t_{p_2}^j, 1) \oplus \ldots \oplus (t_{p_n}^j, 1)\Big)} \\ &= \frac{accum\Big((t_{p_1}^j + t_{p_2}^j + \ldots + t_{p_n}^j, n)\Big)}{paths\Big((t_{p_1}^j + t_{p_2}^j + \ldots + t_{p_n}^j, n)\Big)} \\ &= \frac{t_{p_1}^j + t_{p_2}^j + \ldots + t_{p_n}^j}{n} \end{aligned} \tag{6.10}$$

Εφόσον $n \in \mathbb{N}^*$, και $t_{p_k}^j \in (0,1)\ \forall\ k \in [1,n]$, συνεπάγεται ότι $0 < (t_{p_1}^j + t_{p_2}^j + \ldots + t_{p_n}^j) < n$ και, συνεπώς $T_i^j \in (0,1)$.

Στην εργασία [185] οι Theodorakopoulos και Baras προτείνουν δύο βασικές προϋποθέσεις για τους τελεστές συνένωσης και περίληψης των αλγορίθμων υπολογισμού εμπιστοσύνης. Η πρώτη προϋπόθεση έχει ως εξής:

$$a \otimes b \leq a, b \tag{6.11}$$

Η προϋπόθεση αυτή εξασφαλίζει ότι:

- οι αξιολογήσεις που θα βρεθούν σε ένα πορτφόλιο ενός χρήστη, δε θα θεωρηθούν πιο έμπιστες από τον ίδιο το χρήστη, και
- η τιμή της εμπιστοσύνης προς μια παρεχόμενη υπηρεσία, όπως αυτή έχει καταγραφεί από ένα μονοπάτι εμπιστοσύνης, δε μπορεί να είναι υψηλότερη από την τιμή της αξιολόγησης που συλλέχθηκε από το μονοπάτι αυτό.

Το TwoHop πληροί την παραπάνω προϋπόθεση καθώς $a \otimes b = a \cdot b$ και $a, b \in (0,1)$.

Η δεύτερη προϋπόθεση που θέτουν οι Theodorakopoulos και Baras αφορά στον τελεστή περίληψης. Συγκεκριμένα, κατά τη συλλογή παρόμοιων τιμών εμπιστοσύνης από διάφορα μονοπάτια εμπιστοσύνης, οι τιμές που έχουν συλλεχθεί από ανεξάρτητα μονοπάτια θα πρέπει να ενισχύουν το *βαθμό εγκυρότητας* (confidence level) της τιμής εμπιστοσύνης που θα προκύψει κατά την περίληψη των μονοπατιών και όχι την ίδια την τιμή εμπιστοσύνης. Μάλιστα, η τελική τιμή εμπιστοσύνης που θα προκύψει για ένα πάροχο, δε θα πρέπει να ξεπερνά την υψηλότερη τιμή εμπιστοσύνης που καταγράφηκε για τον πάροχο αυτό στο δίκτυο εμπιστοσύνης, δηλ.:

$$T_i^j \leq max(t_{p_1}^j, t_{p_2}^j, \ldots, t_{p_n}^j). \tag{6.12}$$

Επίσης, στην περίπτωση όπου οι αξιολογήσεις της υπηρεσίας ενός παρόχου έχουν μεγάλη απόκλιση, τότε είτε ο βαθμός εγκυρότητας της τιμής εμπιστοσύνης θα πρέπει να μειωθεί,

είτε η τελική τιμή εμπιστοσύνης θα πρέπει να προκύψει από το μέσο όρο αυτών των αξιολογήσεων.

Στην περίπτωση του TwoHop, σύμφωνα με την αρχή *εκμετάλλευσης πολλαπλών μονοπατιών* (βλ. ενότητα 6.2), σε κάθε αξιολόγηση $v_k \in V_i^j$ αντιστοιχεί ένα μονοπάτι εμπιστοσύνης. Έτσι, η τελική τιμή εμπιστοσύνης υπολογίζεται με βάση αξιολογήσεις που βρέθηκαν σε διαφορετικά πορτφόλιο και ο βαθμός εγκυρότητας της τιμής αυτής αυξάνεται όσο οι αξιολογήσεις που βρέθηκαν συμφωνούν. Επίσης, για να προστατευθούν οι χρήστες από περιπτώσεις όπου οι αξιολογήσεις έχουν διαφορετικές τιμές με μεγάλη απόκλιση, χρησιμοποιείται στον αλγόριθμο υπολογισμού εμπιστοσύνης η τεχνική του μέσου όρου που αναφέρεται παραπάνω. Προκειμένου ο τελεστής περίληψης να διατηρήσει την προσεταιριστική του ιδιότητα και συνάμα η συνάρτηση υπολογισμού της τελικής εμπιστοσύνης να αξιοποιήσει το μέσο όρο, τμήμα του υπολογισμού του μέσου όρου (η τελική διαίρεση μεταξύ του αθροίσματος των τιμών εμπιστοσύνης και του αριθμού των εξεταζόμενων μονοπατιών) μεταφέρθηκε από τον τελεστή περίληψης στη συνάρτηση υπολογισμού εμπιστοσύνης. Τέλος, ο μέσος όρος αυτός βοηθά το TwoHop να καλύψει την προϋπόθεση (6.12). Συγκεκριμένα, από το (6.10) και (6.13) προκύπτει ότι:

$$\frac{t^j_{p_1} + t^j_{p_2} + \ldots + t^j_{p_n}}{n} \leq \frac{n \cdot max(t^j_{p_1}, t^j_{p_2}, \ldots, t^j_{p_n})}{n}, \tag{6.13}$$

δηλαδή ότι η πρόταση (6.12) είναι αληθής για τη συνάρτηση υπολογισμού εμπιστοσύνης που χρησιμοποιεί το TwoHop.

Στην εργασία [185] αναλύονται επίσης μερικά επιθυμητά χαρακτηριστικά για τους τελεστές συνένωσης και περίληψης. Συγκεκριμένα, αν το σύνολο M περιλαμβάνει όλες τις πιθανές τιμές που μπορεί να λάβει το βάρος μιας ακμής στο γράφο εμπιστοσύνης, τότε οι συγγραφείς της εργασίας [185] συνιστούν το σύνολο M να είναι κλειστό ως προς τους τελεστές συνένωσης και περίληψης, δηλ. αν $a, b \in M$, τότε $a \otimes b \in M$ και $a \oplus b \in M$. Με τη συνθήκη αυτή οι τελεστές παράγουν τιμές που είναι άμεσα αναγνωρίσιμες από το χρήστη καθώς μπορούν να συγκριθούν με τις τιμές που έχουν τα βάρη του γράφου εμπιστοσύνης. Στην περίπτωση του TwoHop η συνθήκη αυτή ισχύει μόνο για τον τελεστή συνένωσης:

$$w^j_{x_0,x_1} \otimes w^j_{x_1,x_2} \in M, \quad \forall\ w^j_{x,y} \in M, \quad 0 < w^j_{x,y} < 1, \tag{6.14}$$

οπότε το ζεύγος $(M, \otimes)$ είναι ένα αλγεβρικό *μάγμα*. Ο τελεστής περίληψης στο TwoHop δέχεται διαφορετικής μορφής ορίσματα (ζεύγη $(\bar{t}, n)$) από τον τελεστή συνένωσης. Όμως, η συνάρτηση υπολογισμού της τιμής εμπιστοσύνης (που χρησιμοποιεί τον τελεστή περίληψης) παράγει τελικά τιμές $T^j_i \in M$.

Άλλο ένα ενδιαφέρον χαρακτηριστικό των τελεστών είναι η *μεταθετική* ιδιότητα. Στην περίπτωση του τελεστή συνένωσης, η μεταθετική ιδιότητα μπορεί να εκφραστεί ως:

$$a \otimes b = b \otimes a, \quad \forall\ a, b \in M. \tag{6.15}$$

Στην αρχιτεκτονική TwoHop, η μεταθετική ιδιότητα ισχύει τόσο στον τελεστή συνένωσης όσο και στον τελεστή περίληψης, όπως φαίνεται παρακάτω:

$$\forall\ w^j_{x_0,x_1}, w^j_{x_1,x_2} \in M,\ \bar{t}_1, \bar{t}_2 \in \mathbb{R}^+ \text{ και } n_1, n_2 \in \mathbb{N}^*,$$

$$
\begin{aligned}
w^j_{x_0,x_1} \otimes w^j_{x_1,x_2} &= w^j_{x_0,x_1} \cdot w^j_{x_1,x_2} \\
&= w^j_{x_1,x_2} \cdot w^j_{x_0,x_1} \\
&= w^j_{x_1,x_2} \otimes w^j_{x_0,x_1}
\end{aligned} \tag{6.16}
$$

$$
\begin{aligned}
(\bar{t}_1, n_1) \oplus (\bar{t}_2, n_2) &= (\bar{t}_1 + \bar{t}_2, n_1 + n_2) \\
&= (\bar{t}_2 + \bar{t}_1, n_2 + n_1) \\
&= (\bar{t}_2, n_2) \oplus (\bar{t}_1, n_1)
\end{aligned} \tag{6.17}
$$

Η μεταθετική ιδιότητα ενός τελεστή περίληψης είναι ιδιαίτερα χρήσιμη καθώς επιτρέπει σε έναν αλγόριθμο υπολογισμού εμπιστοσύνης την επεξεργασία μονοπατιών εμπιστοσύνης με τυχαία σειρά.

Η *προσεταιριστική* ιδιότητα των τελεστών διευκολύνει την ενσωμάτωση νέων τιμών με αυτές που είχαν συλλεχθεί παλαιότερα. Η ιδιότητα αυτή για τον τελεστή συνένωσης εκφράζεται ως εξής:

$$
(a \otimes b) \otimes c = a \otimes (b \otimes c), \quad \forall\ a, b, c \in M. \tag{6.18}
$$

Στην αρχιτεκτονική TwoHop η προσεταιριστική ιδιότητα ισχύει τόσο για τον τελεστή συνένωσης, όσο και για τον τελεστή περίληψης. Η ιδιότητα αυτή παρουσιάστηκε στην (6.7) για τον τελεστή περίληψης και παρακάτω παρουσιάζεται για τον τελεστή συνένωσης:

$$
\forall\ w^j_{x,y} \in M,
$$

$$
\begin{aligned}
(w^j_{x_0,x_1} \otimes w^j_{x_1,x_2}) \otimes w^j_{x_2,x_3} &= (w^j_{x_0,x_1} \cdot w^j_{x_1,x_2}) \cdot w^j_{x_2,x_3} \\
&= w^j_{x_0,x_1} \cdot (w^j_{x_1,x_2} \cdot w^j_{x_2,x_3}) \\
&= w^j_{x_0,x_1} \otimes (w^j_{x_1,x_2} \otimes w^j_{x_2,x_3})
\end{aligned} \tag{6.19}
$$

Όπως προαναφέρθηκε παραπάνω, η προσεταιριστική ιδιότητα επιτρέπει σε έναν αλγόριθμο υπολογισμού εμπιστοσύνης να ενσωματώσει εύκολα τις τιμές που συνέλεξε από καινούρια μονοπάτια, σε αυτές που ήδη έχει, χωρίς να απαιτείται κάποιος υπολογισμός εκ νέου πάνω σε όλες τις τιμές που έχουν συλλεχθεί μέχρι εκείνη τη στιγμή. Επίσης, επιτρέπει την παράλληλη εξέταση μονοπατιών, καθώς τα στοιχεία ενός μονοπατιού μπορεί να συμπεριληφθούν άμεσα στο «άθροισμα» των τιμών εμπιστοσύνης, μόλις αυτά γίνουν διαθέσιμα.

Τέλος, στην εργασία [185] εξετάζεται και η *επιμεριστική* ιδιότητα των τελεστών συνένωσης και περίληψης, για παράδειγμα: $a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c)$. Η ιδιότητα αυτή επιτρέπει τη μείωση του αριθμού των πράξεων που γίνονται κατά τον υπολογισμό της τιμής εμπιστοσύνης. Δυστυχώς, επειδή οι τελεστές της αρχιτεκτονικής TwoHop δέχονται διαφορετικές μορφές ορισμάτων, η βελτιστοποίηση αυτή δε μπορεί να εφαρμοστεί.

Συνοψίζοντας, τα κύρια χαρακτηριστικά του Δικτύου Εμπιστοσύνης TwoHop είναι τα εξής:

- Κάθε ακμή του γράφου εμπιστοσύνης έχει πολλαπλά βάρη (βαθμός επιθεώρησης, κρίσης και αξιολόγησης).
- Τα βάρη είναι πραγματικοί αριθμοί από το σύνολο $M : (0, 1)$.
- Κάθε μονοπάτι εμπιστοσύνης έχει το πολύ 3 ακμές.

- Ο τελεστής συνένωσης $\otimes$ πολλαπλασιάζει τις τιμές των βαρών. Ισχύει για αυτόν η προσεταιριστική και η μεταθετική ιδιότητα. Επίσης, το σύνολο M είναι κλειστό ως προς αυτόν και ισχύει η συνθήκη $a \otimes b \leq a, b$.

- Η τιμή εμπιστοσύνης $t_p \in M$ που αντιστοιχεί σε ένα μονοπάτι p, προκύπτει από τον συγκερασμό των κατάλληλων βαρών που συνδέονται με τις ακμές του μονοπατιού αυτού.

- Ο τελεστής περίληψης $\oplus$ προσθέτει ζεύγη της μορφής $(\bar{t}, n)$, όπου $\bar{t}$ είναι το άθροισμα των τιμών εμπιστοσύνης που έχουν συλλεχθεί από n μονοπάτια. Επίσης, ισχύει για αυτόν η προσεταιριστική και η μεταθετική ιδιότητα.

- Η τελική τιμή εμπιστοσύνης $T \in M$ για μια υπηρεσία ενός παρόχου υπολογίζεται με βάση την περίληψη μονοπατιών που οδηγούν σε αξιολογήσεις που έκαναν διαφορετικοί κόμβοι του δικτύου. Επειδή $T = \frac{t_{p_1}+t_{p_2}+\ldots+t_{p_n}}{n}$, η συνθήκη $T \leq max(t_{p_1}, t_{p_2}, \ldots, t_{p_n})$ είναι αληθής.

6.6 Θέματα υλοποίησης και εφαρμογών

Στην ενότητα αυτή θα παρουσιαστεί μια σειρά ειδικών θεμάτων που σχετίζονται με την υλοποίηση ενός δικτύου εμπιστοσύνης TwoHop καθώς και με τις εφαρμογές αυτού.

6.6.1 Ταυτότητα και πιστοποιητικά ενός κόμβου

Ένας κόμβος που συμμετέχει σε ένα δίκτυο εμπιστοσύνης TwoHop συνδέεται σε αυτό χρησιμοποιώντας μια *ταυτότητα* (peer id) η οποία είναι μοναδική σε όλο το δίκτυο. Η ταυτότητα αυτή μπορεί να προκύψει από την εφαρμογή μιας κρυπτογραφικής συνάρτησης κατακερματισμού (για παράδειγμα SHA-1) σε κάποιο στοιχείο που χαρακτηρίζει μοναδικά τον κόμβο αυτό, όπως είναι το δημόσιο κλειδί του.

Σε εφαρμογές όπου οι κόμβοι είναι εξοπλισμένοι με δημόσια/ιδιωτικά κλειδιά, τα πορτφόλιο εμπιστοσύνης υπογράφονται ψηφιακά από το δημιουργό τους πρίν κοινοποιηθούν. Έτσι, τρίτες οντότητες μπορούν να ελέγξουν την ακεραιότητα των δεδομένων που περιέχουν καθώς και να επαληθεύσουν τη σχέση τους με τα μέλη που φαίνονται ως ιδιοκτήτες. Στο σημείο αυτό θα πρέπει να σημειωθεί ότι για τις ενέργειες αυτές δεν απαιτείται κάποια υποδομή δημοσίου κλειδιού (PKI) για την επαλήθευση των ταυτοτήτων, καθώς οι ταυτότητες των μελών είναι στην ουσία τα ίδια τα κλειδιά τους.

6.6.2 Κατανομή πληροφορίας και φόρτου εργασίας

Η αρχιτεκτονική TwoHop έχει σχεδιαστεί κατά τέτοιο τρόπο ώστε να είναι συμβατή με δίκτυα ομότιμων κόμβων (όπως είναι το Chord [191] και το Tapestry [192]). Στα δίκτυα αυτά όλοι οι κόμβοι θεωρούνται ισότιμοι και μπορούν ανά πάσα στιγμή να παρέχουν κάποια υπηρεσία είτε στο ίδιο το δίκτυο είτε στους χρήστες αυτού. Με τη χρήση κατανεμημένων πινάκων κατακερματισμού (distributed hash tables) [193], οι πόροι και οι υπηρεσίες συνδέονται με μέλη του δικτύου. Η κατανομή αυτή εξασφαλίζει τόσο τη μακροβιότητα των υπηρεσιών ενός δικτύου όσο και τη μείωση του φόρτου εργασίας ανά κόμβο.

Τα *δίκτυα επικάλυψης* είναι δίκτυα που σχηματίζονται από εικονικές ζεύξεις μεταξύ ομότιμων κόμβων. Οι ζεύξεις αυτές πραγματοποιούνται στο επίπεδο εφαρμογής (σύμφωνα με το μοντέλο OSI) και οι βασικές δικτυακές υπηρεσίες (δρομολόγηση, μεταφορά

δεδομένων, διευθυνσιοδότηση κλπ.) των ιδεατών αυτών δικτύων παρέχονται από τους ίδιους τους ομότιμους κόμβους. Στην εργασία [194] παρουσιάζεται ένα σύστημα ονόματι CFS, το οποίο επιτρέπει την κατανεμημένη αποθήκευση δεδομένων σε κόμβους δικτύων επικάλυψης. Το TwoHop θα μπορούσε να χρησιμοποιήσει μια τέτοια υπηρεσία για να δημιουργήσει σε πολλαπλούς κόμβους αντίγραφα ασφαλείας (backup) των δεδομένων ενός πορτφόλιο, αυξάνοντας έτσι το χρόνο ζωής αυτής της πληροφορίας στο δίκτυο. Επίσης, εφόσον η ανάκτηση αυτών των δεδομένων θα μπορεί να γίνει από περισσότερες από μία πηγές, θα μπορούσε να επιταχυνθεί η διαδικασία μεταφοράς του πορτφόλιο (και να μειωθεί η παραγόμενη δικτυακή κίνηση) παραλαμβάνοντας τα δεδομένα από κόμβους–πηγές που βρίσκονται πιο κοντά (δικτυακά) στον ενδιαφερόμενο κόμβο. Η επιλογή του κοντινότερου κόμβου μπορεί να γίνει βάσει στοιχείων που θα παρέχει μια υπηρεσία μέτρησης δικτυακών αποστάσεων, όπως είναι το σύστημα GNP [61] για δίκτυα ομότιμων κόμβων.

6.6.3 Προσωρινή αποθήκευση δεδομένων

Ένα πορτφόλιο που έχει ήδη μεταφερθεί μέσω δικτύου μπορεί *προσωρινά* να αποθηκευθεί *τοπικά* (portfolio cache) προκειμένου:

- να επιταχυνθεί η διαδικασία υπολογισμού της τιμής εμπιστοσύνης (επαναχρησιμοποιώντας τα δεδομένα του),
- να μειωθεί το πλήθος των δεδομένων που μεταφέρονται πάνω από το δίκτυο,
- να εξοικονομηθούν πόροι από το σύστημα του ιδιοκτήτη αυτού του πορτφόλιο,
- να γίνει δυνατή η χρήση των δεδομένων του και σε περιόδους όπου ο ιδιοκτήτης αυτού δεν είναι διαθέσιμος στο δίκτυο, και
- να βελτιστοποιηθεί η διαδικασία διανομής των πορτφόλιο.

Στο TwoHop, κάθε αίτηση του κόμβου A για το πορτφόλιο ενός γειτονικού κόμβου $1^{ου}$ βήματος B, ακολουθείται από περαιτέρω αιτήσεις του A για τα πορτφόλιο των κόμβων $2^{ου}$ βήματος που περιγράφονται στο πορτφόλιο του κόμβου B. Επειδή όμως ο κόμβος B είναι και αυτός χρήστης της συνάρτησης υπολογισμού εμπιστοσύνης του TwoHop, υπάρχει μεγάλη πιθανότητα να έχει τοπικά αποθηκευμένα τα πορτφόλιο των γειτονικών του κόμβων $1^{ου}$ βήματος. Τα πορτφόλιο αυτά θα μπορούσαν να αποσταλούν στον A μέσω του B ως βοηθητικά δεδομένα, συνημμένα στην απάντηση της αρχικής αίτησης του A για το πορτφόλιο του B.

Προκειμένου να γίνει προσωρινή αποθήκευση των πορτφόλιο σε αποθηκευτικό χώρο περιορισμένου μεγέθους, θα πρέπει κάθε πορτφόλιο να συνοδεύεται από την ημερομηνία τελευταίας ενημέρωσης αυτού καθώς και από την ψηφιακή υπογραφή του ιδιοκτήτη του. Έτσι, ένας κόμβος θα μπορεί να ελέγξει ανά πάσα στιγμή αν έχει την τελευταία έκδοση ενός πορτφόλιο καθώς και αν τα δεδομένα αυτής της έκδοσης είναι έγκυρα. Η πολιτική σύμφωνα με την οποία θα αντικαθιστώνται τα πορτφόλιο που περιλαμβάνονται στο χώρο προσωρινής αποθήκευσης εξαρτάται άμεσα από το είδος της εφαρμογής που αξιοποιεί το δίκτυο TwoHop. Στις περισσότερες περιπτώσεις πάντως, θα μπορούσε να χρησιμοποιηθεί μια πολιτική LRU (αντικατάσταση των πορτφόλιο που δεν έχουν χρησιμοποιηθεί για μεγάλο χρονικό διάστημα) σε συνδυασμό με αιτήσεις τύπου `If-Modified-Since`, όπως συμβαίνει στις εφαρμογές προσωρινής αποθήκευσης ιστοσελίδων (web caches) [195].

6.6.4 Συμπίεση και επιλεκτική κοινοποίηση δεδομένων ενός πορτφόλιο

Η συνάρτηση `getPortfolio` θα μπορούσε να τροποποιηθεί κατάλληλα ώστε να επιστρέφει συγκεκριμένους τύπους εγγραφών ενός πορτφόλιο (π.χ. αξιολογήσεις, κρίσεις κ.α.). Η δυνατότητα αυτή σε συνδυασμό με τη συμπίεση των δεδομένων των πορτφόλιο (κατά τη μεταφορά τους) θα μπορούσε να μειώσει σημαντικά την κίνηση του δικτύου που οφείλεται στο TwoHop και να βελτιώσει το χρόνο εκτέλεσης του αλγορίθμου υπολογισμού εμπιστοσύνης. Ενθαρρυντικά αποτελέσματα για τη χρήση συμπίεσης σε δίκτυα εμπιστοσύνης, παρουσιάζονται στην εργασία [196], όπου εφαρμόζεται η μέθοδος της «Κωδικοποίησης Δικτύου» (network coding) σε όλα τα δεδομένα που κοινοποιούν οι κόμβοι.

Οι χρήστες της πλατφόρμας TwoHop είναι ελεύθεροι να επιλέξουν το είδος της πληροφορίας που θα αποστείλουν σε τρίτους κόμβους (στα πλαίσια της `getPortfolio`). Π.χ. ένας κόμβος θα μπορούσε να αποκρύψει μια αξιολόγηση ή ακόμη και τη σχέση που έχει με έναν άλλο κόμβο. Η απόκρυψη αυτή είναι ισοδύναμη με την τήρηση πολλαπλών πορτφόλιο ανά τύπο υπηρεσίας και την επιλογή ενός από αυτά για χρήση σύμφωνα με μια πολιτική πρόσβασης (access control list). Παρόλο που αυτή η συμπεριφορά μπορεί να φαίνεται εκ πρώτης όψεως ως κακόβουλη, θα μπορούσε να θεωρηθεί ως μια μέθοδος προστασίας σε περιβάλλοντα όπου οι κόμβοι μπορούν να «στιγματιστούν» από τα δεδομένα που κοινοποιούν σε τρίτους (όπως π.χ. συμβαίνει σε κόμβους δικτύων κοινωνικής δικτύωσης). Σε περιπτώσεις πάντως όπου δεν συντρέχουν τέτοιοι κίνδυνοι, συνιστάται η πλήρης κοινοποίηση των περιεχομένων των πορτφόλιο, καθώς αυτή η ενέργεια προσδίδει στο δίκτυο μια πιο πλούσια πηγή πληροφορίας. Μέσω αυτής της πληροφορίας το δίκτυο μπορεί να «θωρακιστεί» ενάντια σε επιθέσεις κακόβουλων κόμβων που κοινοποιούν ψευδείς πληροφορίες (βλ. παράγραφο 6.7.3).

6.6.5 Πρότυπη υλοποίηση

Στα πλαίσια της έρευνας γύρω από το Δίκτυο Εμπιστοσύνης TwoHop, δημιουργήθηκε μια πρότυπη υλοποίηση[1] αυτού σε γλώσσα Python[2]. Η υλοποίηση αυτή αποτελεί μια πλατφόρμα στην οποία μπορούν να βασιστούν τρίτες εφαρμογές για να υπολογίσουν και να διαχειριστούν τιμές εμπιστοσύνης. Η πρότυπη υλοποίηση απαρτίζεται από τις εξής κλάσεις:

`TrustComputer` – Υλοποιεί τον αλγόριθμο υπολογισμού εμπιστοσύνης του TwoHop.

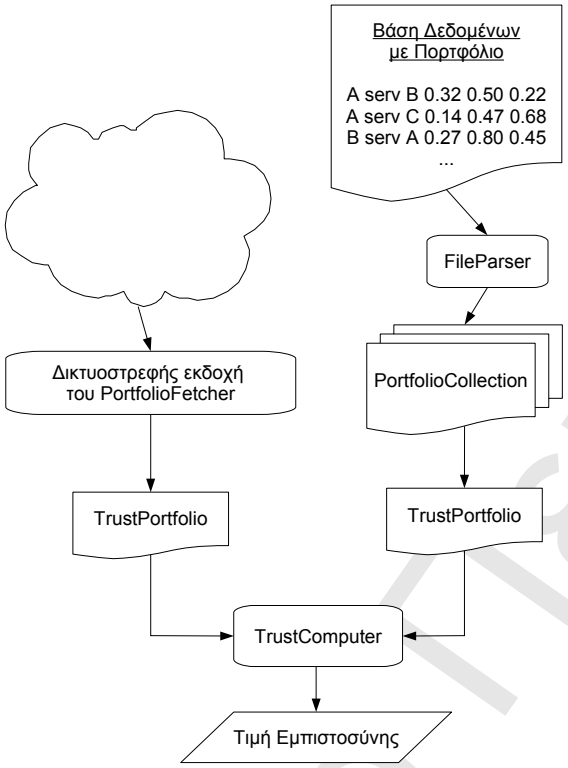
`TrustPortfolio` – Διαχειρίζεται τα δεδομένα ενός πορτφόλιο εμπιστοσύνης.

`PortfolioFetcher` – Διεπαφή που ορίζει τη μέθοδο αναζήτησης και μεταφοράς ενός πορτφόλιο. Η κλάση `TrustComputer` χρησιμοποιεί κάθε φορά την κατάλληλη υποκλάση της `PortfolioFetcher`, ώστε η μεταφορά να γίνεται βάσει των πρωτοκόλλων που ισχύουν για το δίκτυο επικάλυψης που αξιοποιεί το TwoHop.

`PortfolioCollection` – Υλοποιεί τη διεπαφή `PortfolioFetcher` και προσφέρει ένα μέσο αποθήκευσης για τα πορτφόλιο των κόμβων του δικτύου εμπιστοσύνης.

`FileParser` – Ανακτά πληροφορίες για κάποια πορτφόλιο από το σύστημα αρχείων.

`ReviewMutator` – Παρέχει μεθόδους αναδιαμόρφωσης του γράφου εμπιστοσύνης και των βαρών αυτού.



Σχήμα 6.3: Συλλογή των πορτφόλιο από διαφορετικά μέσα (δίκτυο, βάση δεδομένων).

Το σχήμα 6.3 παρουσιάζει τον τρόπο με τον οποίο μπορεί κανείς να εκμεταλλευθεί την αντικειμενοστρεφή σχεδίαση της πρότυπης υλοποίησης προκειμένου να καλύψει τις ανάγκες διαφορετικών εφαρμογών. Στα αριστερά φαίνεται η περίπτωση μιας εφαρμογής που αντλεί στοιχεία (πορτφόλιο) από κόμβους που μετέχουν σε ένα δίκτυο επικάλυψης. Τα πορτφόλιο των κόμβων, που αναπαριστώνται ως αντικείμενα τύπου `TrustPortfolio`, μεταφέρονται μέσω δικτύου χρησιμοποιώντας μια υλοποίηση της διεπαφής `PortfolioFetcher` που εξειδικεύεται στη μεταφορά δεδομένων μέσω του συγκεκριμένου δικτύου επικάλυψης. Η κλάση `TrustComputer` θα μελετήσει τα σχετικά αντικείμενα `TrustPortfolio` και θα παραγάγει την επιθυμητή τιμή εμπιστοσύνης σύμφωνα με τον αλγόριθμο υπολογισμού που ορίζει το TwoHop. Αντίστοιχα, στη δεξιά παράσταση του σχήματος 6.3, φαίνεται μια εφαρμογή που αντλεί τα δεδομένα των πορτφόλιο από μια τοπική βάση δεδομένων. Η κλάση `FileParser` είναι υπεύθυνη για τη δημιουργία αντικειμένων τύπου `TrustPortfolio`, από τα δεδομένα που βρέθηκαν στην τοπική βάση. Τα αντικείμενα `TrustPortfolio` που θα προκύψουν θα τοποθετηθούν σε μια συλλογή από πορτφόλιο, τύπου `PortfolioCollection`, η οποία θα είναι και υπεύθυνη (ως υλοποίηση της διεπαφής `PortfolioFetcher`) να τα παραδώσει στο αντικείμενο της κλάσης `TrustPortfolio`, όποτε αυτό τα χρειαστεί για τον υπολογισμό της τιμής εμπιστοσύνης.

Η πρότυπη υλοποίηση του TwoHop συνοδεύεται από μια εφαρμογή–παράδειγμα η οποία χρησιμοποιεί την πλατφόρμα TwoHop για να υπολογίσει τη "μουσική συμβατότητα" ενός χρήστη/ακροατή με κάποιο καλλιτέχνη, βάσει δεδομένων που αντλούνται από την η-

[1]Το λογισμικό αυτό διατίθεται ελεύθερα υπό την άδεια GPLv3 από την ηλ. διεύθυνση: `http://rainbow.cs.unipi.gr/projects/twohop`

[2]`http://www.python.org`

λεκτρονική υπηρεσία "Last.fm". Το Last.fm[3] αποτελεί μια υπηρεσία κοινωνικής δικτύωσης (social networking), η οποία συγκεντρώνει στατιστικά δεδομένα σχετικά με τη μουσική που ακούν οι χρήστες της. Ένας χρήστης της υπηρεσίας Last.fm μπορεί να παρακολουθήσει τη λίστα των τραγουδιών που άκουσε κάποιος άλλος χρήστης αλλά και να συνάψει μια αμοιβαία σχέση εμπιστοσύνης με αυτόν (εισαγωγή στην ομάδα "φίλων"). Στο πλαίσιο της πλατφόρμας TwoHop, ένας καλλιτέχνης του οποίου τραγούδια εμφανίζονται στα στατιστικά του Last.fm, αποτελεί έναν πάροχο υπηρεσιών. Κάθε χρήστης που ακούει ένα τραγούδι αυτού του καλλιτέχνη, αποτελεί αξιολογητή της υπηρεσίας που παρέχει ο καλλιτέχνης. Ο βαθμός αξιολόγησης προκύπτει από τον αριθμό των ακροάσεων των τραγουδιών αυτού του καλλιτέχνη και η τιμή της αξιολόγησης κανονικοποιείται κατάλληλα ώστε να ανήκει στο διάστημα $[0.01, 0.99]$. Η υπηρεσία "tasteometer" του Last.fm, μετράει τη μουσική συμβατότητα μεταξύ δύο χρηστών. Η συμβατότητα αυτή μπορεί να χρησιμοποιηθεί ως κρίση προς ένα χρήστη–αξιολογητή. Επίσης, η μέση τιμή του "tasteometer" μεταξύ του χρήστη A και των φίλων του χρήστη B, μπορεί να θεωρηθεί ως βαθμός επιθεώρησης του A προς τον B. Χρησιμοποιώντας το μοντέλο αυτό, μπορεί πλέον ένας χρήστης της υπηρεσίας Last.fm να υπολογήσει τη μουσική του συμβατότητα με κάποιο άγνωστο καλλιτέχνη, αξιοποιώντας τα δεδομένα που προκύπτουν από το TwoHop δίκτυο εμπιστοσύνης που σχηματίζουν οι φίλοι αυτού. Τέλος, θα πρέπει να σημειωθεί ότι στην εφαρμογή αυτή η ανανέωση των βαρών του γράφου εμπιστοσύνης δε συμβαίνει άμεσα από το χρήστη, αλλά έμμεσα, καθώς οι τιμές των βαρών διαμορφώνονται αυτόματα σύμφωνα με τις ακροάσεις του χρήστη και τις ακροάσεις των γειτονικών του χρηστών.

Μέρος της πρότυπης υλοποίησης αποτελεί και μια σειρά προγραμμάτων σε γλώσσα Python τα οποία έχουν ως στόχο την παραγωγή δεδομένων ιδεατών πορτφόλιο για χρήση σε πειράματα. Τα δεδομένα αυτά μπορεί να είναι είτε τυχαία είτε βασισμένα σε πραγματικά δεδομένα διαδικτυακών υπηρεσιών που περιγράφουν κάποιο δίκτυο εμπιστοσύνης (π.χ. δεδομένα χρηστών της υπηρεσίας Last.fm, δεδομένα του γράφου εμπιστοσύνης που σχηματίζεται από τα υπογεγραμμένα κλειδιά της εφαρμογής OpenPGP[4]).

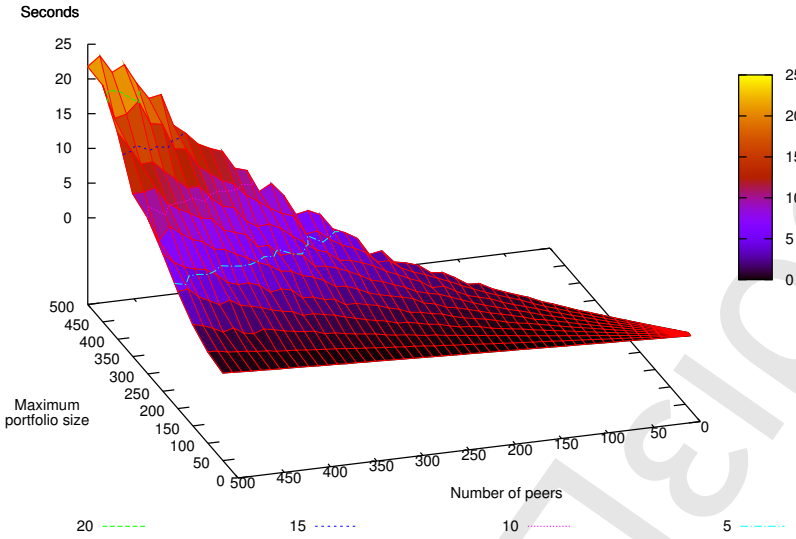
6.7 Πειραματική αξιολόγηση

Στην ενότητα αυτή θα παρουσιαστούν τα αποτελέσματα μιας σειράς πειραμάτων που έχουν ως στόχο την εξέταση των επιδόσεων του αλγορίθμου υπολογισμού εμπιστοσύνης TwoHop. Τα αποτελέσματα αυτά θα συγκριθούν με τα αποτελέσματα αντίστοιχων αλγορίθμων που υπολογίζουν τοπικές και καθολικές τιμές εμπιστοσύνης. Θα διερευνηθεί επίσης, η συμπεριφορά ενός δικτύου εμπιστοσύνης TwoHop, όταν μέλη αυτού δρουν με κακόβουλες προθέσεις. Τα σχετικά πειράματα πραγματοποιήθηκαν σε Η/Υ τύπου Pentium IV 3GHz με 1GB RAM και χρησιμοποιήθηκε ειδικό λογισμικό βασισμένο στην πλατφόρμα πρότυπης υλοποίησης του TwoHop.

Σε κάθε πείραμα χρησιμοποιούνται δίκτυα εμπιστοσύνης που χαρακτηρίζονται από τρεις παραμέτρους: τον τύπο της υπηρεσίας που περιγράφουν, το πλήθος των κόμβων που περιέχουν (δηλ. το πλήθος των πορτφόλιο) και το μέγιστο πλήθος εγγραφών k ανά πορτφόλιο. Κατά τη δημιουργία ενός δικτύου εμπιστοσύνης, κάθε κόμβος αποκτά ένα πορτφόλιο που τον συνδέει με k (το πολύ) τυχαίους κόμβους, σύμφωνα με μια ομοιόμορφη κατανομή (uniform distribution). Στο γράφο εμπιστοσύνης που προκύπτει, τα βάρη των ακμών αποκτούν και αυτά τυχαίες τιμές από το σύνολο $[0.01, 0.99]$.

[3] http://last.fm

[4] http://www.lysator.liu.se/~jc/wotsap



Σχήμα 6.4: Χρόνοι εκτέλεσης του αλγορίθμου TwoHop σε δίκτυα με μεταβλητό αριθμό κόμβων και εγγραφών ανά πορτφόλιο.

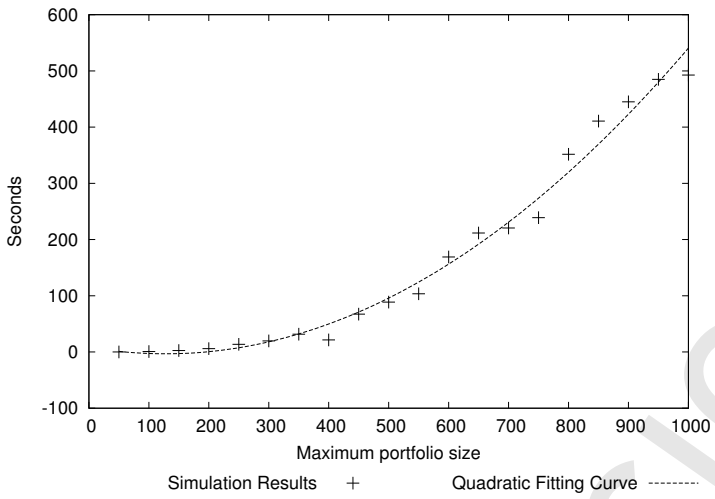
Τα δεδομένα των παρακάτω πειραμάτων έχουν προκύψει από την εφαρμογή του αλγορίθμου υπολογισμού εμπιστοσύνης του TwoHop σε 5 τυχαία δίκτυα εμπιστοσύνης. Στα πειράματα όπου εξετάζεται η ταχύτητα εκτέλεσης του αλγορίθμου, καταγράφεται η μέγιστη χρονική καθυστέρηση που είχε αυτός όταν εφαρμόστηκε στα 5 δίκτυα εμπιστοσύνης. Αντίθετα, στα πειράματα όπου εξετάζεται η συμπεριφορά του δικτύου TwoHop όταν σε αυτό συμμετέχουν κακόβουλοι χρήστες, καταγράφεται η μέση διαφορά που σημειώθηκε στην τιμή εμπιστοσύνης για κάποιο πάροχο των παραπάνω 5 δικτύων.

6.7.1 Αξιολόγηση επιδόσεων

Το πρώτο πείραμα καταγράφει το χρόνο εκτέλεσης του αλγορίθμου υπολογισμού εμπιστοσύνης TwoHop σε δίκτυα μικρών κοινοτήτων με μέγιστο αριθμό συμμετοχών 500 κόμβους. Το σχήμα 6.4 παρουσιάζει τα αποτελέσματα του πειράματος, με τον καταγεγραμμένο χρόνο στον άξονα y (seconds), τον αριθμό των κόμβων στον άξονα x (number of peers) και το μέγιστο αριθμό εγγραφών ανά πορτφόλιο στον άξονα z (maximum portfolio size). Κάθε μέτρηση αντιστοιχεί στο μέγιστο χρόνο εκτέλεσης της συνάρτησης `calculateTrust`, όταν αυτή κλήθηκε για να υπολογίσει την τιμή εμπιστοσύνης για έναν τυχαίο πάροχο υπηρεσιών, χρησιμοποιώντας ένα τυχαίο βασικό πορτφόλιο (root trust portfolio).

Οι απόλυτοι χρόνοι του πειράματος δεν είναι τόσο σημαντικοί όσο οι σχετικοί χρόνοι, καθώς οι απόλυτοι χρόνοι επηρεάζονται από την συγκεκριμένη υλοποίηση του αλγορίθμου υπολογισμού εμπιστοσύνης. Μια υλοποίηση σε γλώσσα προγραμματισμού χαμηλότερου επιπέδου (π.χ. γλώσσα C) θα μπορούσε να προσφέρει ακόμη πιο σύντομους χρόνους εκτέλεσης. Από τα πειραματικά αποτελέσματα, πάντως, διαφαίνεται ότι ο χρόνος εκτέλεσης επηρεάζεται κυρίως από το μέγεθος των πορτφόλιο.

Για να εξεταστεί περαιτέρω η σχέση μεταξύ πλήθους εγγραφών και χρόνου εκτέλεσης, έγινε ένα νέο πείραμα σε δίκτυα εμπιστοσύνης με σταθερό αριθμό κόμβων. Το σχήμα 6.5 παρουσιάζει τα αποτελέσματα του πειράματος για ένα δίκτυο με 5000 κόμβους και μέγιστο



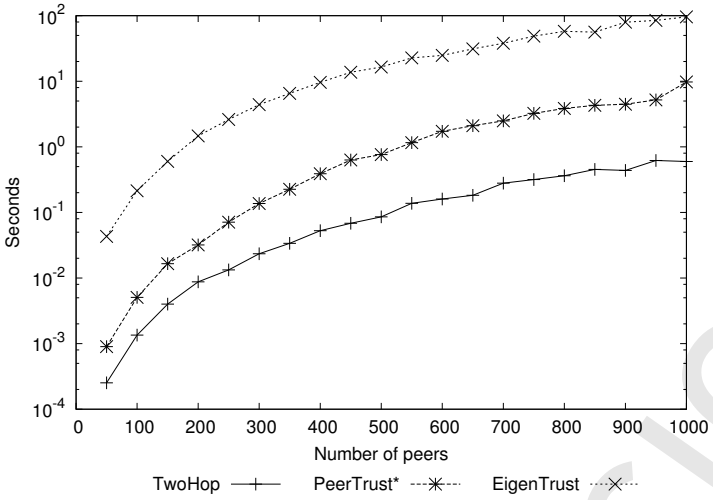
Σχήμα 6.5: Εξέταση της επίδρασης του μεγέθους των πορτφόλιο σε ένα δίκτυο με 5000 κόμβους.

πλήθος εγγραφών ανά πορτφόλιο, 1000. Με προσαρμογή των δεδομένων χρησιμοποιώντας ένα πολυώνυμο δευτέρου βαθμού (quadratic fitting curve) παρατηρεί κανείς ότι τα πειραματικά δεδομένα έχουν μικρές αποκλίσεις από τη γραφική παράσταση του πολυωνύμου. Η παρατήρηση αυτή έρχεται να επιβεβαιώσει το θεωρητικό χρόνο εκτέλεσης $O(k^2)$ που υπολογίστηκε για τη συνάρτηση `calculateTrust` στην ενότητα 6.4.

Στο τρίτο πείραμα εξετάζεται ο χρόνος εκτέλεσης του αλγορίθμου υπολογισμού εμπιστοσύνης TwoHop σε σχέση με το χρόνο εκτέλεσης δύο άλλων αλγορίθμων, του EigenTrust [178] και του αλγορίθμου των Prasad et al. [182]. Για χάρη συντομίας στον αλγόριθμο των Prasad et al δόθηκε η ονομασία PeerTrust*, μιας και αποτελεί βελτιωμένη έκδοση του αλγορίθμου PeerTrust [183].

Ο *κεντρικοποιημένος* αλγόριθμος του δικτύου EigenTrust υπολογίζει μια προσέγγιση της τιμής εμπιστοσύνης για ένα πάροχο υπηρεσιών μέσω μιας επαναληπτικής διαδικασίας. Σε κάθε επανάληψη υπολογίζεται με μεγαλύτερη ακρίβεια η *καθολική τιμή* εμπιστοσύνης προς τους αξιολογητές του παρόχου και, έτσι, προκύπτει ένα πιο ακριβές αποτέλεσμα για την καθολική τιμή εμπιστοσύνης προς τον πάροχο. Ο αλγόριθμος τερματίζει είτε όταν η ακρίβεια του αποτελέσματος έχει γίνει μικρότερη από κάποιο κατώφλι ϵ, είτε όταν έχουν ολοκληρωθεί μ επαναλήψεις. Η ακρίβεια της τιμής του αποτελέσματος υπολογίζεται βάσει της διαφοράς μεταξύ των τιμών εμπιστοσύνης που προκύπτουν για τον πάροχο σε δύο διαδοχικές επαναλήψεις. Επίσης, προκειμένου ο αλγόριθμος να συγκλίνει σε κάποιο αποτέλεσμα, χρησιμοποιείται μια βοηθητική σταθερά $\alpha \in (0, 1)$ στη συνάρτηση υπολογισμού της τιμής εμπιστοσύνης, η οποία κάνει μη περιοδικό τον πίνακα με τις καταγεγραμμένες αξιολογήσεις. Στο πλαίσιο του παρόντος πειράματος, ο αλγόριθμος EigenTrust υλοποιήθηκε σε γλώσσα Python, χρησιμοποιώντας τις παρακάτω τιμές για τις αντίστοιχες σταθερές: $\alpha = 0.5$, $\epsilon = 10^{-5}$ και $\mu = 100$.

Ο αλγόριθμος PeerTrust*, όπως και ο TwoHop, υπολογίζει *τοπικές τιμές* εμπιστοσύνης με *αποκεντρωμένο* τρόπο. Κατά τον υπολογισμό της τιμής για ένα πάροχο, γίνεται αναζήτηση για όλους τους κόμβους που έχουν αξιολογήσει τον εν λόγω πάροχο. Για κάθε ένα από αυτούς τους αξιολογητές ξεκινά νέα έρευνα προκειμένου να διαπιστωθούν οι «κοινοί πάροχοι», δηλαδή οι πάροχοι που έλαβαν αξιολόγηση τόσο από τον αιτούντα κόμβο όσο και από τον υπο διερεύνηση αξιολογητή. Στη συνέχεια, για κάθε αξιολογητή υπολογίζεται ο βαθμός αξιοπιστίας με βάση τον αριθμό των «κοινών παρόχων», την απόκλιση των αξιολο-



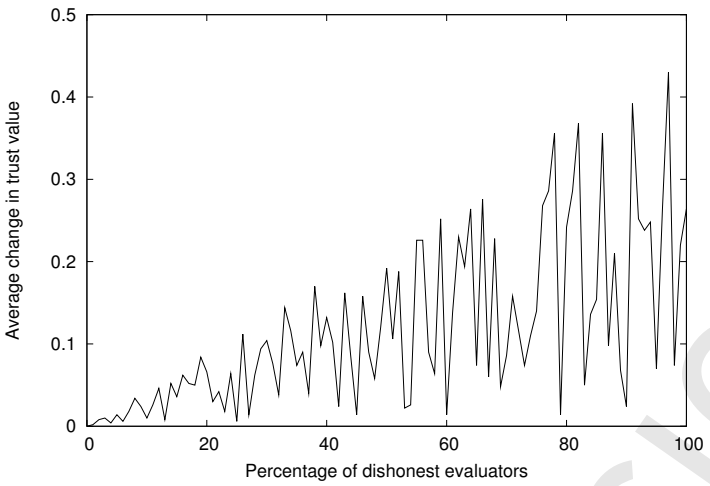
Σχήμα 6.6: TwoHop, PeerTrust* και EigenTrust σε δίκτυα με μεταβλητό πλήθος κόμβων και εγγραφών ανά πορτφόλιο.

γήσεων (του αξιολογητή και του αιτούντος κόμβου) για τους κοινούς παρόχους, το πλήθος των αξιολογήσεων που έχει κάνει ο αιτών κόμβος, καθώς και το πλήθος των αξιολογήσεων που έχει κάνει ο συγκεκριμένος αξιολογητής. Η τελική τιμή εμπιστοσύνης για τον πάροχο προκύπτει από το σταθμισμένο μέσο όρο των αξιολογήσεων που συλλέχθηκαν. Το ρόλο των βαρών σε αυτό το μέσο όρο παίζουν ο βαθμός αξιοπιστίας κάθε αξιολογητή.

Στο σχήμα 6.6 παρουσιάζονται οι χρόνοι εκτέλεσης των αλγορίθμων των δικτύων TwoHop, PeerTrust* και EigenTrust, καθώς αυτοί υπολογίζουν την τιμή εμπιστοσύνης για έναν τυχαίο πάροχο σε δίκτυα με μεταβλητό αριθμό κόμβων. Το πλήθος των κόμβων παρουσιάζεται στον άξονα x, ενώ οι χρόνοι εκτέλεσης στον άξονα y. Κάθε κόμβος στα δίκτυα αυτά έχει στην κατοχή του ένα πορτφόλιο εμπιστοσύνης με αριθμό εγγραφών ίσο με το 10% του συνολικού πλήθους των κόμβων του δικτύου.

Τα αποτελέσματα δείχνουν ότι ο αλγόριθμος του δικτύου EigenTrust βρέθηκε μέχρι και 100 φορές πιο αργός από τον αντίστοιχο του TwoHop σε κάθε πειραματικό σενάριο. Η μεγάλη καθυστέρηση που παρατηρείται στην εκτέλεση του EigenTrust, τον χαρακτηρίζει ως ακατάλληλο για χρήση σε *online* εφαρμογές και περιορίζει τη χρήση αυτού σε δίκτυα εμπιστοσύνης με λίγους κόμβους. Όμως, θα πρέπει να σημειωθεί ότι η καθυστέρηση του αλγορίθμου αυτού οφείλεται κυρίως στο γεγονός ότι υπολογίζει πρώτα τις τιμές εμπιστοσύνης για όλους τους κόμβους του δικτύου πρίν αποφανθεί για την τελική τιμή εμπιστοσύνης για τον πάροχο υπηρεσιών. Οι ήδη υπολογισμένες τιμές εμπιστοσύνης θα μπορούσαν να αποθηκευτούν προσωρινά προκειμένου να χρησιμοποιηθούν σε μελλοντικούς υπολογισμούς, δίνοντας έτσι τη δυνατότητα στον αλγόριθμο του δικτύου EigenTrust να υπολογίσει στιγμιαία την τιμή εμπιστοσύνης για έναν οποιοδήποτε πάροχο. Βέβαια, η τιμή που θα παραχθεί σε αυτή την περίπτωση ενδέχεται να βασίζεται σε παλαιά δεδομένα και έτσι μπορεί να μην αποτυπώνει την πραγματική εικόνα που έχουν εκείνη τη στιγμή οι συμμετέχοντες στο δίκτυο για τον πάροχο υπηρεσιών.

Ο αλγόριθμος PeerTrust* παρουσίασε καλύτερες επιδόσεις από τον EigenTrust σε όλα τα πειραματικά σενάρια. Αυτό οφείλεται κυρίως στο γεγονός ότι ο αλγόριθμος PeerTrust* υπολογίζει μία μόνο τιμή εμπιστοσύνης, λαμβάνοντας πληροφορίες από ένα τμήμα του συνολικού γράφου εμπιστοσύνης. Παραμένει όμως βραδύτερος από τον TwoHop, καθώς αναζητά «κοινούς παρόχους» κάθε φορά που εντοπίζει έναν κατάλληλο αξιολογητή.



Σχήμα 6.7: Εξέταση των επιπτώσεων που έχουν οι ανειλικρινείς αξιολογήσεις σε ένα δίκτυο 1000 κόμβων με 200 (το πολύ) εγγραφές ανά πορτφόλιο εμπιστοσύνης.

6.7.2 Ανειλικρινείς Αξιολογητές

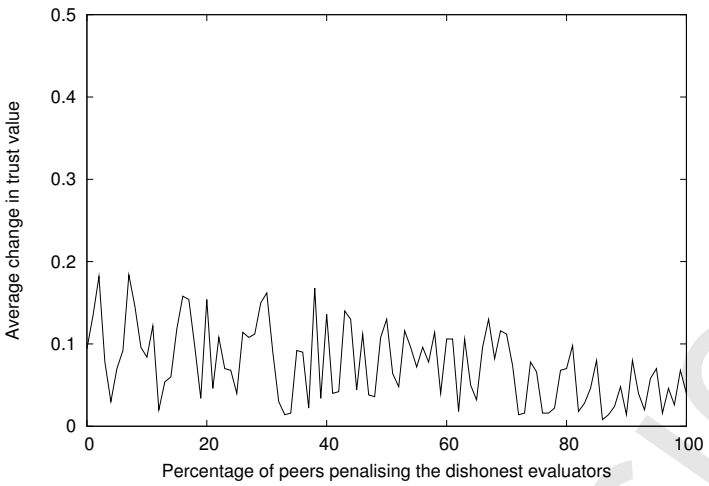
Στο πείραμα αυτό μελετάται η επίδραση που έχει μια ομάδα *ανειλικρινών αξιολογητών* στην τιμή εμπιστοσύνης που υπολογίζεται για κάποιο πάροχο υπηρεσιών. Οι ανειλικρινείς αξιολογητές παρέχουν αξιολογήσεις οι οποίες είναι ψευδείς και δεν αντικατοπτρίζουν το πραγματικό επίπεδο υπηρεσίας που παρέχει ένας κόμβος.

Προκειμένου να προσομοιωθεί αυτή η μορφή επίθεσης προς ένα δίκτυο εμπιστοσύνης TwoHop, δημιουργείται ένα τυχαίο δίκτυο με 1000 κόμβους, στο οποίο ο κάθε κόμβος έχει στη διάθεσή του ένα πορτφόλιο εμπιστοσύνης με 200 (το πολύ) εγγραφές. Στη συνέχεια, επιλέγεται ο κόμβος με τις περισσότερες αξιολογήσεις ως ο πάροχος υπηρεσιών που θα αποτελέσει το θύμα της επίθεσης. Όλοι οι υπολογισμοί τιμών εμπιστοσύνης προς τον πάροχο αυτό γίνονται χρησιμοποιώντας ως βασικό πορτφόλιο το πορτφόλιο ενός τυχαίου κόμβου του δικτύου.

Σε κάθε εκτέλεση της προσομοίωσης καταγράφεται η μεταβολή Δ_T στην τιμή εμπιστοσύνης του πάροχου, όταν ένα μέρος των μελών του δικτύου παρέχουν ανειλικρινείς αξιολογήσεις της υπηρεσίας αυτού. Η μεταβολή στην τιμή εμπιστοσύνης ορίζεται ως $\Delta_T = |T_0 - T_y|$, όπου T_0 είναι η τιμή εμπιστοσύνης όταν το δίκτυο δεν περιέχει ανειλικρινείς αξιολογητές και T_y είναι η τιμή εμπιστοσύνης όταν ένα ποσοστό από τους αξιολογητές του κόμβου–θύματος παρέχει ψευδείς αξιολογήσεις.

Οι ειλικρινείς αξιολογητές βαθμολογούν την υπηρεσία του παρόχου με μια τυχαία τιμή από το εύρος $[\rho - 0.1, \rho + 0.1]$, όπου $\rho \in (0.1, 0.9)$. Το ρ είναι μια τυχαία τιμή η οποία παραμένει ίδια για όλους τους ειλικρινείς αξιολογητές. Με τον τρόπο αυτό προσομοιώνεται μια «συμφωνία» μεταξύ των αξιολογήσεων των κόμβων αυτών, η οποία περιγράφει το πραγματικό επίπεδο της υπηρεσίας του θύματος. Αντίθετα, οι ψευδείς αξιολογήσεις των ανειλικρινών αξιολογητών προσομοιώνονται με τυχαίες τιμές από το εύρος (0, 1), χωρίς να απαιτείται κάποια ομοιότητα ή συμφωνία μεταξύ των τιμών αυτών.

Στο σχήμα 6.7 φαίνεται η επίδραση των ανειλικρινών αξιολογητών στο παραπάνω δίκτυο 1000 κόμβων. Ο άξονας y παρουσιάζει τη μεταβολή Δ_T στην τιμή εμπιστοσύνης του κόμβου–θύματος, η οποία αυξάνει καθώς το ποσοστό των κόμβων που μετέχουν στην ομάδα ανειλικρινών αξιολογητών (άξονας x) γίνεται μεγαλύτερο. Οι μεταβολές Δ_T στις τιμές εμπιστοσύνης που παρουσιάζονται στο σχήμα 6.7, προέρχονται από το μέσο όρο των



Σχήμα 6.8: Περιορίζοντας την επίδραση των ανειλικρινών αξιολογήσεων μιας ομάδας κόμβων σε ένα δίκτυο εμπιστοσύνης με $1,000$ μέλη και 200 (το πολύ) εγγραφές ανά πορτφόλιο εμπιστοσύνης.

μεταβολών που σημειώθηκαν σε 5 τυχαία σεναρία του δικτύου εμπιστοσύνης. Τα σενάρια αυτά είχαν το ίδιο ποσοστό ανειλικρινών αξιολογητών, αλλά όχι απαραίτητα τους ίδιους αξιολογητές.

6.7.3 Περιορίζοντας την επιρροή των Ανειλικρινών Αξιολογητών

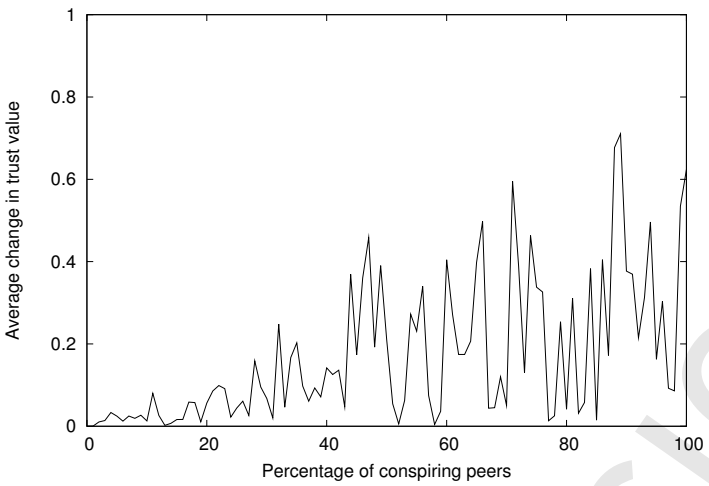
Οι κόμβοι ενός γράφου εμπιστοσύνης TwoHop μπορούν να χρησιμοποιήσουν τα βάρη τους (δηλ. τις τιμές που περιέχονται στα πορτφόλιο τους) ως μέσο άμυνας προς τις αξιολογήσεις των ανειλικρινών κόμβων. Για να εξεταστεί η αποτελεσματικότητα αυτής της μεθόδου άμυνας, θα χρησιμοποιηθεί το δίκτυο εμπιστοσύνης της προηγούμενης παραγράφου σε ένα νέο πείραμα, όπου το 40% των αξιολογητών του κόμβου–θύματος θα παρέχουν ανειλικρινείς αξιολογήσεις.

Στο σχήμα 6.8 φαίνεται ότι η μεταβολή Δ_T στην τιμή εμπιστοσύνης (άξονας y) που οφείλεται στους ανειλικρινείς αξιολογητές, μπορεί να ελαχιστοποιηθεί αν οι υπόλοιποι κόμβοι του δικτύου επιβάλλουν «κυρώσεις» στους ανειλικρινείς αξιολογητές. Στο συγκεκριμένο πείραμα, ένα ποσοστό (που καταγράφεται στον άξονα x) των κριτών που είχαν βαθμολογήσει στο παρελθόν τους ανειλικρινείς αξιολογητές, ανανεώνει τις τιμές των παλαιών κρίσεων του ($weight_{old}$) με νέες χαμηλότερες τιμές ($0.2 \times weight_{old}$). Όπως φαίνεται στο σχήμα 6.8, όσο μεγαλύτερο γίνεται το ποσοστό των κριτών που επιβάλλουν κυρώσεις, τόσο περισσότερο μειώνεται η επιρροή των ανειλικρινών αξιολογητών στην τιμή εμπιστοσύνης.

Αντί να πολλαπλασιαστεί η παλαιά κρίση προς ένα ανειλικρινή αξιολογητή με κάποιο συντελεστή, θα μπορούσε αυτή να τεθεί σε μια κατώτατη τιμή (π.χ. 0.01) ως ένδειξη ότι ο κόμβος αυτός παρέχει ψευδείς αξιολογήσεις. Όμως, ένα τέτοιο μέτρο ενδεχομένως να μην μπορεί να εφαρμοστεί σε μικρά δίκτυα εμπιστοσύνης, καθώς αναιρεί όλες τις αξιολογήσεις ενός κόμβου που στο παρελθόν μπορεί να είχε λάβει πολύ καλές κριτικές.

6.7.4 Συνομωτικοί Κόμβοι

Στο πείραμα αυτό θα εξεταστεί μια διαφορετική μορφή επίθεσης προς δίκτυα εμπιστοσύνης, από κόμβους που δρούν οργανωμένοι σε ομάδες. Έστω ότι ένα σύνολο κόμβων έχει δημιουργήσει ένα υποδίκτυο κοινής εμπιστοσύνης, όπου οι κόμβοι αυτού του υποδικτύου



Σχήμα 6.9: Η επίδραση συνομωτικών κόμβων σε ένα δίκτυο με 1000 μέλη και 200 (το πολύ) εγγραφές ανά πορτφόλιο εμπιστοσύνης.

συνδέονται όλοι μεταξύ τους με υψηλά βάρη (βαθμολογίες κρίσης). Τα μέλη αυτού του υποδικτύου αποφασίζουν να αξιολογήσουν έναν πάροχο διαφορετικά από ότι τα υπόλοιπα μέλη του ευρύτερου δικτύου εμπιστοσύνης, ακολουθώντας κάποια πολιτική. Επειδή οι κόμβοι αυτοί δρουν *συνομωτικά*, ονομάζονται *Συνομωτικοί Κόμβοι*.

Όπως διαπιστώθηκε στην προηγούμενη παράγραφο, η δράση συνομωτικών κόμβων που περιορίζεται σε αξιολογήσεις, μπορεί να μην επηρεάσει σε μεγάλο βαθμό την τιμή εμπιστοσύνης προς κάποιον πάροχο, αν τρίτοι κόμβοι–κριτές χρησιμοποιήσουν τα κατάλληλα βάρη εναντίον τους. Όμως, εάν κάποιος από τους συνομωτικούς κόμβους αποτελέσει μέρος ενός βασικού πορτφόλιο, τότε ως κριτής θα μπορέσει να διαμορφώσει κατά μεγάλο βαθμό την υπολογιζόμενη τιμή εμπιστοσύνης, καθώς θα εισάγει σε αυτή κρίσεις και αξιολογήσεις από το συνομωτικό υποδίκτυο.

Για να εξεταστεί η επιρροή που ασκούν οι κόμβοι–κριτές του συνομωτικού δικτύου, στον υπολογισμό της τιμής εμπιστοσύνης, έγινε ένα νέο πείραμα χρησιμοποιώντας το δίκτυο εμπιστοσύνης της προηγούμενης παραγράφου. Συγκεκριμένα, κάθε ανειλικρινής αξιολογητής μετατράπηκε σε συνομωτικό κόμβο και συνδέθηκε με τους υπόλοιπους συνομωτικούς κόμβους μέσω βαρών–κρίσεων από το σύνολο $[0.8, 1.0)$. Αντίστοιχα με τους ειλικρινείς αξιολογητές της προηγούμενης παραγράφου, οι συνομωτικοί κόμβοι αξιολογούν την υπηρεσία του παρόχου–θύματος με τυχαίες τιμές από το σύνολο $[\sigma - 0.1, \sigma + 0.1]$, όπου $\sigma \in (0.1, 0.9)$. Το σ είναι μια τυχαία τιμή που ισχύει για όλα τα μέλη της συνομωτικής ομάδας ανά πειραματικό σενάριο. Στο σχήμα 6.9 φαίνεται ότι η μεταβολή Δ_T στην τιμή εμπιστοσύνης (άξονας y) αυξάνει αναλογικά ως προς το ποσοστό των συμμετοχών στην ομάδα συνομωτικών κόμβων (άξονας x).

Θα πρέπει να σημειωθεί εδώ ότι κάθε κόμβος του δικτύου που περιγράφεται σε ένα πορτφόλιο, αποτελεί ένα γειτονικό κόμβο του οποίου η υπηρεσία έχει ελεγχθεί στο παρελθόν και έχει σχηματιστεί μια συγκεκριμένη «άποψη» για αυτή. Έτσι, εάν ένας τέτοιος κόμβος αποφασίσει να συμμετάσχει σε μια συνομωτική ομάδα, ο ιδιοκτήτης του πορτφόλιο είτε θα εισχωρήσει και αυτός στην ομάδα (αν συμφωνεί με την πολιτική αυτής), είτε θα ανανεώσει τα βάρη του[5] ώστε τα μέλη της συνομωτικής ομάδας να μην επηρεάσουν τους επόμενους υπολογισμούς τιμών εμπιστοσύνης.

[5]Η θα αφαιρέσει τον συνομωτικό κόμβο από το πορτφόλιο

Παρόλα αυτά, οι προθέσεις μιας ομάδας κόμβων που βαθμολογεί διαφορετικά από το υπόλοιπο δίκτυο εμπιστοσύνης, δεν είναι απαραίτητα κακές. Μέλη μιας τέτοιας ομάδας μπορούν να αναδείξουν με τις αξιολογήσεις τους, τις υπηρεσίες νέων παρόχων υπηρεσιών, οι οποίοι δεν έχουν γίνει ευρύτερα γνωστοί μέχρι εκείνη τη στιγμή. Είναι σημαντικό λοιπόν, να επιτρέπεται η σύσταση τέτοιων υποομάδων, καθώς αυτές μπορούν να εκφράσουν νέες τάσεις στο δίκτυο εμπιστοσύνης και παράλληλα μπορούν να προσαρμόσουν κατάλληλα τη δομή του δικτύου ώστε να καλύπτονται οι ανάγκες των χρηστών αυτού.

6.8 Ποιοτική αξιολόγηση

6.8.1 Σύγκριση χαρακτηριστικών δικτύων εμπιστοσύνης

Ο πίνακας 6.1 παρουσιάζει συνοπτικά τις βασικές διαφορές του TwoHop, σε σχέση με τα δίκτυα εμπιστοσύνης που εξετάστηκαν στην ενότητα 6.1. Η σύγκριση γίνεται με βάση τις αρχές της αρχιτεκτονικής TwoHop, όπως αυτές περιγράφηκαν στην ενότητα 6.2. Με την ένδειξη "N/A" σημειώνονται τα δίκτυα των οποίων η σχεδίαση είναι ασύμβατη ως προς κάποια αρχή. Εφόσον όλα τα δίκτυα που εξετάστηκαν εκφράζουν την εμπιστοσύνη ως μια ασύμμετρη σχέση, το χαρακτηριστικό αυτό έχει παραληφθεί από τον πίνακα.

Δίκτυο Εμπιστοσύνης	**Τοπικές τιμές εμπ.**	**Δικτυακό μέτρο εμπ.**	**Μονοπάτια εμπ. περιορισμένου μήκους**	**Διαφοροποίηση τύπων εμπ.**	**Εκμετάλλευση πολλαπλών μονοπατιών**
Marsh [166]	√		N/A		
eBay [169]			N/A		
Poblano [170]	√	√			
GNUnet [171]	√		N/A		
PGP [66]	√	√	√		
Advogato [172]		√			√
TrustFlow [173]	√	√			
OpenPrivacy [174]					
Free Haven [175]	Τοπικές & Καθολικές				
P2PRep [176]	√				√
XRep [177]	√				√
EigenTrust [178]		√			√
Jiang et al. [179]	√				√
Raya et al. [180]	√				√
PeerTrust* [182]	√		N/A	√	√
Theodorakopoulos et al. [184]	√	√			√
TwoHop [168]	√	√	√	√	√

Πίνακας 6.1: Σύγκριση χαρακτηριστικών Δικτύων Εμπιστοσύνης.

6.8.2 Προστασία ενάντια σε ενεργές επιθέσεις

Η αρχιτεκτονική TwoHop προσφέρει προστασία από μια σειρά επιθέσεων που προσβάλλουν δίκτυα εμπιστοσύνης και συστήματα διαχείρισης φήμης. Στην παρούσα ενότητα θα περιγραφούν οι κύριες μορφές ενεργών επιθέσεων προς τέτοιου είδους δίκτυα και θα εξεταστεί η ανθεκτικότητα της αρχιτεκτονικής TwoHop ως προς αυτές.

Η πιο απλή μορφή επίθεσης σε δίκτυα όπου λαμβάνονται υπόψιν αξιολογήσεις τρίτων είναι η επίθεση *αυτο-προώθησης* (self-promotion attack) [197]. Κατά την επίθεση αυτή ένας κόμβος προσπαθεί να βελτιώσει το βαθμό εμπιστοσύνης (ή τη φήμη) του στο δίκτυο,

κοινοποιώντας κάλπικες θετικές αξιολογήσεις των υπηρεσιών του. Η αρχιτεκτονική TwoHop αντιμετωπίζει αυτού του είδους την παραβατική συμπεριφορά με τα εξής δύο μέτρα:

α. κατά τον υπολογισμό της τιμής εμπιστοσύνης ενός κόμβου, δε λαμβάνει υπόψιν τις αξιολογήσεις που προέρχονται από τον κόμβο αυτό, και

β. επιτρέπει την επιβεβαίωση της γνησιότητας της πηγής μιας αξιολόγησης με τη χρήση ψηφιακών υπογραφών.

Έτσι, ένα δίκτυο TwoHop προστατεύεται από αυτό-αξιολογήσεις κακόβουλων κόμβων αλλά και από αξιολογήσεις που φέρουν κάλπικα στοιχεία αξιολογητή (spoofed source evaluations).

Μια άλλη μορφή επίθεσης που εκμεταλλεύεται τη δυνατότητα κοινοποίησης αξιολογήσεων με ψευδείς τιμές, είναι η επίθεση *δυσφήμισης* (bad mouthing attack) [198]. Σε αυτή την επίθεση ένας ή περισσότεροι ανεξάρτητοι κόμβοι κοινοποιούν ψευδείς αρνητικές αξιολογήσεις για έναν πάροχο υπηρεσιών προκειμένου να βλάψουν τη φήμη αυτού. Όπως παρουσιάστηκε στην παράγραφο 6.7.3, οι κόμβοι ενός δικτύου TwoHop έχουν τη δυνατότητα να περιορίσουν από κοινού τις επιπτώσεις μιας τέτοιας επίθεσης, βαθμολογώντας αρνητικά τους αξιολογητές που συμμετείχαν σε αυτή.

Παρόμοια επίθεση είναι και αυτή των *συνομωτικών κόμβων* (collusion attack) [183] όπου κακόβουλοι κόμβοι οργανωμένοι σε ομάδες βαθμολογούν θετικά ή αρνητικά συγκεκριμένους παρόχους ανάλογα με τα συμφέροντά τους. Για να ενισχυθούν περαιτέρω αυτές οι αξιολογήσεις, τα μέλη της κάθε ομάδας βαθμολογούνται και μεταξύ τους με υψηλές τιμές. Όπως περιγράφηκε στην ενότητα 6.7.4, τα μέλη ενός δικτύου TwoHop μπορούν να περιορίσουν σημαντικά τις επιπτώσεις μιας τέτοιας επίθεσης μέσω της δράσης τους ως κριτές και επιθεωρητές. Επίσης, μπορούν να προχωρήσουν και σε αφαίρεση των παραπάνω κακόβουλων κόμβων από τα πορτφόλιό τους ώστε να μειωθεί όσο το δυνατόν περισσότερο η επίδρασή αυτών στις διάφορες ιεραρχίες εμπιστοσύνης.

Μια διαφορετική μορφή επίθεσης είναι αυτή της *αντιφατικής συμπεριφοράς* (conflicting behavior attack) [198]. Έστω μια ομάδα κακόβουλων παρόχων υπηρεσιών οι οποίοι παρέχουν υπηρεσίες υψηλής ποιότητας στην ομάδα κόμβων A και υπηρεσίες χαμηλής ποιότητας στην ομάδα κόμβων B. Οι κακές αξιολογήσεις που θα κοινοποιήσει η ομάδα B για τις υπηρεσίες των παρόχων, θα οδηγήσουν την ομάδα A στο συμπέρασμα ότι τα μέλη της ομάδας B είναι κακοί αξιολογητές, αλλά και το αντίστροφο. Τελικά, οι κοινοποιήσεις των αξιολογήσεων θα δημιουργήσουν λανθασμένες εντυπώσεις στο δίκτυο και θα πληγούν τόσο τα μέλη της ομάδας A όσο και τα μέλη της ομάδας B. Δυστυχώς, το δίκτυο TwoHop δε μπορεί να προσφέρει κάποιο μέτρο προστασίας ενάντια σε αυτού του είδους την επίθεση. Είναι ευθύνη της εκάστοτε εφαρμογής να εντοπίσει κόμβους που συμπεριφέρονται κατ'αυτό τον τρόπο και ευθύνη των χρηστών να δράσουν ως αξιολογητές, ώστε να περιθωριοποιηθούν οι παραπάνω πάροχοι και να μη βλάψουν περαιτέρω το δίκτυο εμπιστοσύνης.

Στην επίθεση *on-off* [198] κακόβουλοι κόμβοι συμπεριφέρονται με καλό και με κακό τρόπο εναλλάξ. Αυτού του είδους η συμπεριφορά είναι ιδιαίτερα συμφέρουσα για τους κακόβουλους κόμβους σε δίκτυα όπου η τιμή της εμπιστοσύνης υπολογίζεται βάσει των τελευταίων N συναλλαγών με έναν πάροχο. Η αρχιτεκτονική TwoHop δεν ορίζει συγκεκριμένο τρόπο με τον οποίο ανανεώνονται τα βάρη του γράφου εμπιστοσύνης και έτσι δεν μπορεί να αντιμετωπίσει άμεσα το παραπάνω πρόβλημα. Μια εφαρμογή όμως, μπορεί να προστατευτεί από την επίθεση αυτή, διατηρώντας για μεγαλύτερο χρονικό διάστημα τις κακές αξιολογήσεις μέσα στην προσωρινή μνήμη (buffer) των τελευταίων N συναλλαγών. Έτσι γίνεται όλο και πιο δύσκολο για έναν κακόβουλο πάροχο να επανακτήσει την καλή του φήμη, εάν έχουν προηγηθεί μια σειρά από κακές συναλλαγές.

Έφόσον το δίκτυο εμπιστοσύνης TwoHop δεν μπορεί να αποτρέψει έναν κόμβο από το να συμμετέχει σε αυτό με περισσότερες από μία ταυτότητες, είναι ευάλωτο σε επιθέσεις τύπου *Sybil* [199]. Στην επίθεση αυτή ένα κόμβος εμφανίζεται με πολλαπλές ταυτότητες σε ένα δίκτυο προκειμένου να υποστηρίξει κάποια θέση του σε μια διαδικασία όπου λαμβάνονται δημοκρατικά αποφάσεις Όπως αναφέρθηκε στην παράγραφο 6.7.4 οι επιπτώσεις τέτοιων επιθέσεων μπορούν να περιοριστούν σε ένα βαθμό με τη χρήση κατάλληλων βαρών και την άσκηση μιας συντηρητικής πολιτικής κατά την εισαγωγή νέων κόμβων στα πορτφόλιο εμπιστοσύνης. Μια πιο εκτεταμένη παρουσίαση της επίθεσης *Sybil* και των επιπτώσεων αυτής θα δοθεί στην παράγραφο 7.1.1 όπου παρουσιάζεται μια νέα πλατφόρμα για την αντιμετώπιση επιθέσεων πλαστοπροσωπείας, σε περιβάλλοντα όπου επιτρέπεται η ταυτοποίηση κόμβων βάσει πολλαπλών χαρακτηριστικών.

Τέλος, ένας χρήστης του δικτύου TwoHop που έχει λάβει στο παρελθόν πολλές αρνητικές αξιολογήσεις (π.χ. εξαιτίας μιας επίθεσης τύπου *Karma Suicide*) μπορεί να αποχωρήσει από το δίκτυο, καταστρέφοντας την παλιά του ταυτότητα και επιστρέφοντας με μια νέα. Η ταυτότητα αυτή δε θα είναι πλέον συνδεδεμένη με τις αρνητικές αξιολογήσεις που είχε λάβει στο παρελθόν και θα μπορεί να δράσει στο δίκτυο δίχως κάποια αρνητική «φήμη». Αυτού του είδους η επίθεση ονομάζεται επίθεση *αθώωσης* (whitewashing attack) [197]. Προκειμένου να τιμωρηθούν οι κόμβοι που ακολουθούν αυτή την στρατηγική, θα πρέπει οι τιμές εμπιστοσύνης που λαμβάνει ένας κόμβος στο δίκτυο TwoHop να μην είναι μικρότερες από την τιμή εμπιστοσύνης που μπορεί να λάβει ένας νέος κόμβος. Θα πρέπει να σημειωθεί πάντως ότι με την καταστροφή της ταυτότητας ενός κόμβου στο δίκτυο TwoHop, χάνεται μέρος του κέρδους που είχε αποκτήσει ο κόμβος αυτός από το δίκτυο, καθώς δεν υπάρχουν πλέον πορτφόλιο που αναφέρονται σε αυτόν.

6.8.3 Πλεονεκτήματα της αρχιτεκτονικής TwoHop

Το βασικό πλεονέκτημα του δικτύου εμπιστοσύνης TwoHop είναι η αποκεντρωμένη λειτουργία αυτού, η οποία στηρίζεται στη συνδρομή ομότιμων κόμβων. Κάθε κόμβος μπορεί ανά πάσα στιγμή να υπολογίσει μια μετρική εμπιστοσύνης προς ένα πάροχο, βασιζόμενος σε δεδομένα που προκύπτουν από τη δική του ιεραρχία εμπιστοσύνης. Ο τρόπος με τον οποίο διαμορφώνεται η ιεραρχία εμπιστοσύνης εξαρτάται σε μεγάλο βαθμό από τον ίδιο τον κόμβο και τις σχέσεις αυτού με τους άλλους κόμβους στο δίκτυο εμπιστοσύνης. Εφόσον λοιπόν ο κάθε κόμβος υπολογίζει μέσω τοπικών κριτηρίων την τιμή εμπιστοσύνης προς κάποιο πάροχο, έχει την ελευθερία να διατηρήσει μια δική του οπτική ως προς τους «έμπιστους» παρόχους του δικτύου.

Ο αλγόριθμος υπολογισμού των τιμών εμπιστοσύνης λειτουργεί με *online* τρόπο και δίνει τη δυνατότητα σε ένα κόμβο να λάβει μια εκτίμηση της εικόνας που έχει το δίκτυο εμπιστοσύνης για έναν πάροχο, εκείνη τη στιγμή. Αυτή η πληροφορία είναι ιδιαίτερα σημαντική για συστήματα που πρέπει να αντιδράσουν σε μικρό χρονικό διάστημα σε αλλαγές του δικτυακού περιβάλλοντος (π.χ. συστήματα εντοπισμού δικτυακών επιθέσεων). Ο μικρός χρόνος εκτέλεσης του αλγορίθμου οφείλεται κυρίως στη μικρή πολυπλοκότητα αυτού καθώς και στο γεγονός ότι εξετάζει ένα υπο-γράφο του συνολικού δικτύου εμπιστοσύνης, που έχει περιορισμένο μήκος μονοπατιών. Όπως προκύπτει και από τα αποτελέσματα των πειραμάτων που παρουσιάστηκαν στην ενότητα 6.7.1, οι χρόνοι εκτέλεσης του TwoHop είναι σαφώς μικρότεροι των αντίστοιχων χρόνων του κεντρικοποιημένου αλγορίθμου EigenTrust και του αποκεντρωμένου PeerTrust* και κυμαίνονται σε τέτοια επίπεδα που γίνεται δυνατή η χρήση του αλγορίθμου σε διαδικτυακές εφαρμογές προγραμματιζόμενων δικτύων μεγάλης κλίμακας. Όπως αναφέρθηκε στην ενότητα 6.6.3, ο χρόνος εκτέλεσης του αλγορίθμου μπορεί να βελτιωθεί περαιτέρω χρησιμοποιώντας κάποια τεχνική προσωρινής

αποθήκευσης των πορτφόλιο που έχουν ήδη εξεταστεί.

Η εισαγωγή ενός νέου κόμβου στο δίκτυο εμπιστοσύνης TwoHop δε συνοδεύεται από κάποιο διαχειριστικό κόστος. Ο νέος κόμβος απλά χρησιμοποιεί και αξιολογεί τις υπηρεσίες των υπόλοιπων κόμβων και με την πάροδο του χρόνου διαμορφώνει το προσωπικό του πορτφόλιο. Ένας νέος κόμβος, που δεν έχει ακόμη εγγραφές στο πορτφόλιο του, μπορεί να χρησιμοποιήσει το πορτφόλιο κάποιου άλλου κόμβου για να ανακαλύψει παρόχους και αξιολογητές. Για να γίνει ο κόμβος αυτός μέρος μιας ιεραρχίας εμπιστοσύνης, θα πρέπει να επικοινωνήσει με κάποιο άλλο κόμβο και να παρέχει σε αυτόν κάποια υπηρεσία ή αξιολόγηση. Αν το επίπεδο της προσφοράς του είναι ικανοποιητικό τότε θα εισαχθεί στην ιεραρχία εμπιστοσύνης του κόμβου με τον οποίο ήρθε σε επαφή και οι τιμές του πορτφόλιο του θα επηρεάσουν τους υπολογισμούς των τιμών εμπιστοσύνης άλλων κόμβων.

Οι έμμεσες και οι δικτυακές τιμές εμπιστοσύνης που υπολογίζονται στο δίκτυο TwoHop είναι συγκρίσιμες με τις άμεσες τιμές εμπιστοσύνης (αξιολογήσεις κλπ.) που καταχωρίζουν οι κόμβοι στα πορτφόλιο τους. Συγκεκριμένα όλες οι τιμές αυτές ανήκουν στο σύνολο $(0, 1)$. Έτσι, ο χρήστης μπορεί να αντιληφθεί εύκολα τη σημασία μιας τιμής εμπιστοσύνης και να την αξιοποιήσει άμεσα δίχως κάποια περαιτέρω επεξεργασία (κανονικοποίηση κλπ.). Όπως επισημαίνουν οι Theodorakopoulos και Baras [185], αυτό είναι ένα σημαντικό χαρακτηριστικό που βοηθά στη χρήση ενός αλγορίθμου υπολογισμού εμπιστοσύνης, αλλά το οποίο δυστυχώς απουσιάζει από σημαντικό μέρος των δικτύων εμπιστοσύνης που έχουν προταθεί στη βιβλιογραφία. Θα πρέπει να σημειωθεί εδώ ότι ο τοπικός υπολογισμός τιμών εμπιστοσύνης βάσει αξιολογήσεων που έχουν γίνει από τρίτους κόμβους, προϋποθέτει ότι όλοι οι κόμβοι έχουν συμφωνήσει σε μια ενιαία ερμηνεία της βαθμολογίας αξιολόγησης (π.χ. μια ικανοποιητική παροχή υπηρεσίας περιγράφεται με 0.9) γιατί στην αντίθετη περίπτωση δεν είναι δυνατός ο συγκερασμός βαρών καθώς θα συνδυάζει τιμές οι οποίες έχουν διαφορετική ερμηνεία για κάθε χρήστη.

Τέλος, οι πολλαπλές σχέσεις εμπιστοσύνης μεταξύ των κόμβων ενός δικτύου TwoHop επιτρέπουν την καλύτερη περιγραφή των σχέσεων μεταξύ των κόμβων. Μάλιστα, με τη χρήση των σχετικών βαρών μπορεί να αναδειχθούν έμπιστοι αξιολογητές υπηρεσιών αλλά και να μειωθεί η επίδραση κακόβουλων αξιολογητών, όπως διαπιστώθηκε από τα πειραματικά δεδομένα της ενότητας 6.7.3. Φυσικά τα βάρη αυτά μπορεί να χρησιμοποιηθούν και για να προωθήσουν μια συγκεκριμένη βαθμολόγηση μιας υπηρεσίας ενός παρόχου. Αυτό όμως δεν είναι απαραίτητα κακό, καθώς δίνει στο δίκτυο κάποια ευελιξία η οποία με τη σειρά της επιτρέπει τη δημιουργία τάσεων, οι οποίες εκφράζουν μέλη του δικτύου.

Κεφάλαιο 7

Προστασία ενάντια σε Επιθέσεις Πλαστοπροσωπείας

Κατά την επίθεση πλαστοπροσωπείας ένας κόμβος παρουσιάζεται σε ένα δίκτυο με μια ταυτότητα η οποία δεν του ανήκει. Η ταυτότητα αυτή μπορεί να συνδέεται είτε με κάποιον ήδη υπάρχοντα κόμβο (κλοπή / δανεισμός ταυτότητας), είτε όχι. Σε όλες τις περιπτώσεις ο κόμβος που εκτελεί την επίθεση παρουσιάζεται με μια ταυτότητα η οποία δεν είναι νόμιμα δική του, σύμφωνα με το πρωτόκολλο λειτουργίας του δικτύου.

Με το δανεισμό μιας διαφορετικής ταυτότητας ένας κόμβος μπορεί να εξαπατήσει τρίτους κόμβους και υπηρεσίες, χρησιμοποιώντας τις σχέσεις εμπιστοσύνης που εκείνοι είχαν αναπτύξει προς τον πραγματικό ιδιοκτήτη της κλεμμένης ταυτότητας. Επίσης, η χρήση μιας διαφορετικής ταυτότητας θα επιτρέψει σε ένα κακόβουλο κόμβο να αποφύγει την αρνητική φήμη ή τις αρνητικές πολιτικές που είχαν σχηματιστεί γύρω από την παλαιά του ταυτότητα. Μάλιστα, σε περίπτωση που η νέα ταυτότητα αντιστοιχεί σε κάποιον ήδη υπάρχοντα κόμβο, οι ενέργειες του κακόβουλου κόμβου υπό αυτή την ταυτότητα θα επηρεάσουν τη φήμη του πραγματικού ιδιοκτήτη αυτής.

Μια διαφορετική περίπτωση εξαπάτησης αφορά τον τρόπο συμμετοχής ενός κόμβου σε ένα δίκτυο. Σε "δημοκρατικές" διαδικασίες όπου λαμβάνονται αποφάσεις βάσει ψήφων από τους κόμβους ενός δικτύου, ένας κόμβος που εμφανίζεται στο δίκτυο με περισσότερες από μία ταυτότητες θα μπορεί να δώσει ένα δυσανάλογο πλεονέκτημα ψήφων στη δική του επιλογή και να διαμορφώσει τελικά το αποτέλεσμα μιας διαδικασίας σύμφωνα με κάποια δική του πολιτική.

Η ασφαλής ταυτοποίηση ενός κόμβου αποτελεί ένα πρόβλημα που αφορά στον ευρύτερο χώρο των δικτύων υπολογιστών. Σε αυτό το πρόβλημα συντελούν κυρίως τρείς λόγοι:

- Η ταυτότητα ενός κόμβου δηλώνεται από τον ίδιο τον κόμβο και όχι από κάποια εξειδικευμένη (έμπιστη) υποδομή η οποία αναλαμβάνει την αποστολή και προώθηση δεδομένων.

- Τα διάφορα επίπεδα επικοινωνίας (βλ. μοντέλο OSI) επιτρέπουν σε κάθε κόμβο να έχει μια διαφορετική ταυτότητα για κάθε επίπεδο, δίχως οι ταυτότητες αυτές να συνδέονται μεταξύ τους με κάποιο τρόπο.

- Όταν δύο κόμβοι δεν έχουν κάποιο άμεσο κανάλι επικοινωνίας, θα πρέπει να εμπιστευθούν τρίτους κόμβους ώστε τα πακέτα τους να προωθηθούν στον τελικό τους προορισμό.

Συνεπώς, ένας κόμβος δε μπορεί να εμπιστευθεί *a priori* την αναγραφόμενη ως ταυτότητα αποστολέα σε κάποια δεδομένα. Θα πρέπει να επιβεβαιώσει σε κάθε περίπτωση ότι τα δεδομένα αυτά παρήχθησαν από τον κόμβο που κατείχε την εν λόγω ταυτότητα.

Στην ενότητα 5.4 παρουσιάστηκαν οι επιπτώσεις των επιθέσεων πλαστοπροσωπείας σε δίκτυα προγραμματιζόμενων κόμβων, ενώ στην ενότητα 6.8 τονίστηκε η σημασία της ορθής ταυτοποίησης των κόμβων στα πλαίσια ενός δικτύου εμπιστοσύνης. Στην παρούσα ενότητα θα εξεταστούν αναλυτικότερα οι διάφοροι τύποι των επιθέσεων πλαστοπροσωπείας και θα προταθεί μια νέα πλατφόρμα η οποία προσφέρει ασφαλή ταυτοποίηση κόμβων σε δίκτυα προγραμματιζόμενων υποδομών. Η πλατφόρμα αυτή βασίζεται στην ταυτοποίηση βάσει πολλαπλών χαρακτηριστικών και μπορεί να εφαρμοστεί σε ασύρματα δίκτυα κινητών κόμβων (Mobile Ad Hoc Networks) για να διασφαλίσει (μεταξύ άλλων) την αξιόπιστη δρομολόγηση δεδομένων.

7.1 Τύποι επιθέσεων πλαστοπροσωπείας

Στην ενότητα αυτή θα εξεταστούν οι τρεις βασικοί τύποι επιθέσεων πλαστοπροσωπείας: η επίθεση *Sybil*, η επίθεση του *Αόρατου Κόμβου* και η επίθεση *Κλοπής Πιστοποιητικών*.

7.1.1 Η επίθεση Sybil

Κατά την επίθεση τύπου Sybil [199] ένας κόμβος εμφανίζεται σε ένα δίκτυο με μία ή περισσότερες ψεύτικες ταυτότητες. Υπάρχουν τρεις μορφές αυτής της επίθεσης:

- Ο κακόβουλος κόμβος χρησιμοποιεί την ταυτότητα ενός πραγματικού κόμβου για να προσποιηθεί ότι αποτελεί τον κόμβο αυτό σε κάποιο άλλο μέλος του δικτύου.
- Ο κακόβουλος κόμβος χρησιμοποιεί μία ή περισσότερες νέες ταυτότητες οι οποίες δεν αντιστοιχούν σε πραγματικούς κόμβους του δικτύου. Με τον τρόπο αυτό μπορεί να επηρεάσει το αποτέλεσμα δημοκρατικών διαδικασιών σε δίκτυα ομότιμων κόμβων. Επίσης, μπορεί να δημιουργήσει τη ψευδαίσθηση ότι υπάρχουν εναλλακτικοί δίοδοι επικοινωνίας μεταξύ κόμβων σε δίκτυα ad hoc, το οποίο θα μείωνε την αξιοπιστία αλγορίθμων δρομολόγησης που αξιοποιούν πολλαπλά μονοπάτια επικοινωνίας.
- Ο κακόβουλος κόμβος μπορεί να αλλάζει συστηματικά την ταυτότητά του σε κάποια άλλη προκειμένου να αποφύγει τις επιπτώσεις της κακής «φήμης» που τον ακολουθεί. Το σενάριο αυτό εξετάστηκε στην ενότητα 6.8.2 στο πλαίσιο των Δικτύων Εμπιστοσύνης και της επίθεσης τύπου *Karma Suicide*.

Η πρώτη μορφή της επίθεσης (δηλ. η χρήση μιας ταυτότητας που αντιστοιχεί σε πραγματικό κόμβο του δικτύου) μπορεί να αποφευχθεί με τη χρήση κρυπτογραφικών μεθόδων ταυτοποίησης. Εφόσον ο κακόβουλος κόμβος δε θα έχει στη διάθεσή του το ιδιωτικό κλειδί που αντιστοιχεί στην ταυτότητα που έχει «δανειστεί», δε θα είναι σε θέση να παραγάγει έγκυρες υπογραφές με την ταυτότητα αυτή. Έτσι, ο παραλήπτης των μηνυμάτων του κακόβουλου κόμβου θα μπορεί να διαπιστώσει εύκολα ότι τα μηνύματα αυτά δεν προέρχονται από τον κόμβο με την αναγραφόμενη ταυτότητα, καθώς θα απουσιάζει η αντίστοιχη υπογραφή.

Για να αντιμετωπιστούν οι υπόλοιπες δύο μορφές της επίθεσης Sybil απαιτείται η χρήση κλειδιών που φέρουν την υπογραφή μιας αρχής πιστοποίησης. Η αρχή πιστοποίησης λειτουργεί εδώ σαν φορέας που μπορεί να βεβαιώσει ότι δύο πιστοποιητικά ανήκουν στον ίδιο κόμβο του δικτύου.

Σε δυναμικά και πλήρως κατανεμημένα περιβάλλοντα, όπου δεν είναι δυνατή η χρήση μιας κεντρικής αρχής πιστοποίησης, η χρήση μιας κρυπτογραφικής μεθόδου ταυτοποίησης θα εξασφαλίσει μονάχα τη σύνδεση των υπογεγραμμένων μηνυμάτων με ένα συγκεκριμένο πιστοποιητικό. Η σύνδεση όμως αυτή, θα επιτρέψει την καταγραφή των ενεργειών ενός κόμβου υπό μια συγκεκριμένη ταυτότητα και έτσι τα μέλη του δικτύου θα είναι σε θέση να αξιολογήσουν τη συμπεριφορά αυτού του κόμβου χρησιμοποιώντας ένα Δίκτυο Εμπιστοσύνης (σαν αυτό που περιγράφηκε στην ενότητα 6.2). Το Δίκτυο Εμπιστοσύνης δε θα εμποδίσει έναν κόμβο από το να συμμετάσχει με περισσότερες από μία ταυτότητες σε ένα δίκτυο. Σε περίπτωση όμως που διαπιστωθεί μια τέτοια κακόβουλη πράξη, το Δίκτυο Εμπιστοσύνης θα αναπροσαρμόσει κατάλληλα τα βάρη στο γράφο εμπιστοσύνης, ώστε οι ενέργειες του κακόβουλου κόμβου να μην επηρεάσουν περαιτέρω τις υπηρεσίες και τους χρήστες του δικτύου.

Για να περιοριστεί ο αριθμός των ταυτοτήτων που μπορεί να παραγάγει ένας κόμβος μπορεί να χρησιμοποιηθεί μια μέθοδος τύπου *hashcash* [200] όπου η δημιουργία μιας νέας ταυτότητας θα συνοδεύεται πάντα και από μια "χρονοβόρα εργασία" την οποία θα πρέπει να φέρει εις πέρας ο αιτών κόμβος. Παράδειγμα μιας τέτοιας εργασίας είναι η εύρεση μιας συμβολοσειράς της οποίας η κρυπτογραφικά κατακερματισμένη μορφή (SHA1 sum) ξεκινά με k bit που έχουν την τιμή μηδέν (0). Για να εξασφαλιστεί η επανάληψη της εργασίας σε κάθε αντίστοιχη περίσταση, στη συμβολοσειρά θα πρέπει να περιέχονται πληροφορίες όπως η τρέχουσα ημερομηνία και ώρα, μια τυχαία τιμή (nonce) καθώς και μια συμβολοσειρά που περιγράφει το πιστοποιητικό που θα συνδεθεί με την ταυτότητα του κόμβου (ή η κρυπτογραφικά κατακερματισμένη έκδοση αυτού). Το επεξεργαστικό κόστος της ζητούμενης εργασίας εξαρτάται από τον αριθμό των k bits που θα πρέπει να έχουν την τιμή μηδέν (0) στο αποτέλεσμα της κρυπτογραφικής συνάρτησης. Δυστυχώς, όπως διαπιστώνει και ο συγγραφέας της εργασίας [201], αυτή η μέθοδος μπορεί να βρει περιορισμένη μόνο χρήση, καθώς η επεξεργαστική ικανότητα των συσκευών που συμμετέχουν στα σημερινά δίκτυα δεδομένων ποικίλει. Επίσης, θα πρέπει να σημειωθεί ότι μια ομάδα κακόβουλων κόμβων με online πρόσβαση σε ένα ισχυρό υπολογιστικό σύστημα θα μπορούσε ενδεχομένως να παραγάγει πολλαπλάσιο αριθμό ταυτοτήτων, αξιοποιώντας τις αυξημένες υπολογιστικές δυνατότητες του συστήματος αυτού.

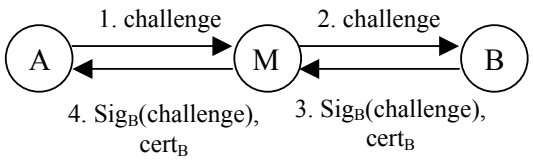
7.1.2 Η επίθεση του Αόρατου Κόμβου

Ένας «Αόρατος Κόμβος» δρομολογεί «σιωπηλά» τα πακέτα που λαμβάνει, δίχως να αποκαλύπτει την ύπαρξή του σε ένα ή περισσότερα μονοπάτια επικοινωνίας. Με τον τρόπο αυτό δίνει μια λανθασμένη εικόνα της τοπολογίας του δικτύου στους γύρω κόμβους.

Κατά την επίθεση του «Αόρατου Κόμβου» δύο ή περισσότεροι κόμβοι θεωρούν ότι έχουν άμεση επικοινωνία μεταξύ τους, ενώ στην πραγματικότητα παρεμβάλλεται μεταξύ αυτών ένας «Αόρατος Κόμβος». Σε ένα ασύρματο δίκτυο ad hoc, όπου ο κάθε κόμβος εξετάζει τον αριθμό και το είδος των μονοπατιών μέσω των οποίων μπορούν να φτάσουν τα πακέτα του στον προορισμό τους, η επίθεση του «Αόρατου Κόμβου» θα έχει δυσάρεστες συνέπειες καθώς θα δώσει την εντύπωση σε κάποιο κόμβο ότι έχει περισσότερα εναλλακτικά μονοπάτια επικοινωνίας από ότι πραγματικά υπάρχουν.

Η επίθεση του «Αόρατου Κόμβου» ανήκει στην ευρύτερη οικογένεια των *Επιθέσεων Μεσάζοντα* (man-in-the-middle attacks). Στην ουσία, ο «Αόρατος Κόμβος» M που μεταφέρει σιωπηλά τα πακέτα ενός κόμβου A σε ένα κόμβο B, παριστάνει τον κόμβο A στον κόμβο B, με διαφορετική όμως θέση στο δίκτυο (γείτονας $1^{ου}$ βήματος). Επειδή πρόκειται για μια επίθεση στο επίπεδο μεταφοράς δεδομένων (σύμφωνα με τη στοίβα πρωτοκόλλων OSI), η χρήση ενός κρυπτογραφικού μέσου ελέγχου των ταυτοτήτων των κόμβων A και B

δεν αρκεί για να αποκαλύψει την απάτη του κόμβου M.



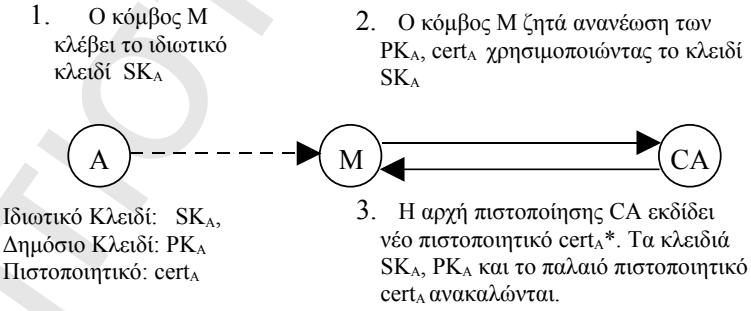
Σχήμα 7.1: Επίθεση «Αόρατου Κόμβου»

Όπως φαίνεται και στο σχήμα 7.1, μια διαδικασία τύπου *challenge–response* θα ολοκληρωθεί επιτυχώς μεταξύ των κόμβων A και B, πιστοποιώντας την ταυτότητα του κόμβου B στον A, παρόλη την ύπαρξη του «Αόρατου Κόμβου» M. Ο κόμβος M θα παραμείνει κρυφός, προωθώντας «σιωπηλά» το μήνυμα τύπου *challenge* του κόμβου A στον κόμβο B και ομοίως την απάντηση τύπου *response* του κόμβου B (με την υπογραφή $Sig_B(challenge)$ και το αντίστοιχο πιστοποιητικό $cert_B$) στον κόμβο Α.

7.1.3 Η επίθεση Κλοπής Πιστοποιητικών

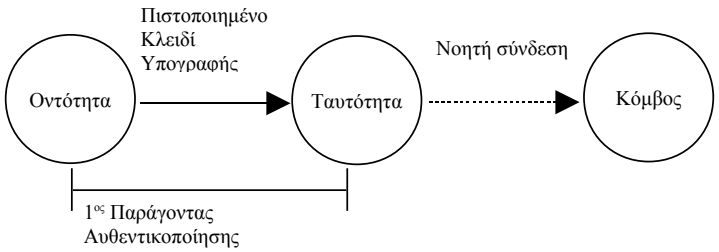
Η επίθεση κλοπής πιστοποιητικών γίνεται σε δύο φάσεις. Στην πρώτη φάση ένας κακόβουλος κόμβος κλέβει τα ιδιωτικά στοιχεία ενός κόμβου-θύματος που θα χρησίμευαν κατά την ταυτοποίηση του κόμβου-θύματος σε κάποιο τρίτο κόμβο (π.χ. το ιδιωτικό του κλειδί). Χρησιμοποιώντας τα ιδιωτικά αυτά στοιχεία μπορεί πλέον να υποδυθεί τον κόμβο-θύμα σε οποιονδήποτε τρίτο κόμβο.

Στη δεύτερη φάση της επίθεσης ο κακόβουλος κόμβος θα προσπαθήσει να ανανεώσει το πιστοποιητικό του θύματος χρησιμοποιώντας τα κλεμμένα ιδιωτικά στοιχεία. Αν προλάβει και έρθει σε επαφή με την αρχή πιστοποίησης πρίν έρθει σε επαφή με αυτή ο κόμβος–θύμα, τότε θα λάβει ένα νέο πιστοποιητικό με νέα κλειδιά, στα οποία δε θα έχει πλέον πρόσβαση το θύμα. Ουσιαστικά, με τον τρόπο αυτό ο κακόβουλος κόμβος «κλέβει» την ταυτότητα του θύματος.



Σχήμα 7.2: Επίθεση Κλοπής Πιστοποιητικών

Στο σχήμα 7.2 παρουσιάζεται μια τέτοια επίθεση. Θα πρέπει να σημειωθεί ότι πέραν της ταυτότητας του θύματος, ο κακόβουλος κόμβος θα μπορεί μετά την επίθεση να εκμεταλλευτεί και όποιες σχέσεις εμπιστοσύνης είχε δημιουργήσει το θύμα με άλλους κόμβους του δικτύου.



Σχήμα 7.3: Το παραδοσιακό μοντέλο ταυτοποίησης κόμβων.

Όπως και στην επίθεση του «Αόρατου Κόμβου» έτσι και εδώ, η χρήση κρυπτογραφικών μεθόδων ταυτοποίησης δε μπορεί να αποτρέψει τις συνέπειες των δράσεων των κακόβουλων κόμβων.

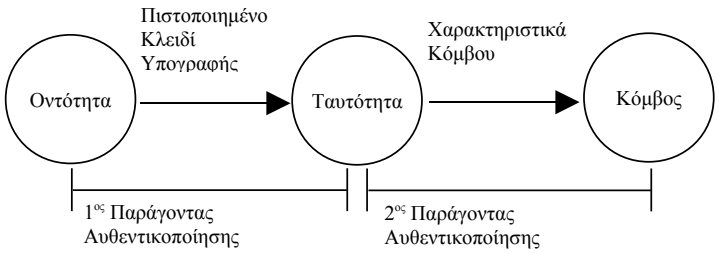
7.2 Ταυτοποίηση κόμβων βάσει πολλαπλών χαρακτηριστικών

Όπως φάνηκε από τα συμπεράσματα της προηγούμενης ενότητας, η χρήση κρυπτογραφικών μεθόδων ταυτοποίησης δεν αποτελεί από μόνη της μια ικανή λύση για την αντιμετώπιση επιθέσεων πλαστοπροσωπείας. Ο χρήστης μιας κρυπτογραφικής μεθόδου ταυτοποίησης δέχεται ως δεδομένο ότι ο χρήστης ενός συγκεκριμένου ιδιωτικού κλειδιού (ή ενός κοινού μυστικού στην περίπτωση συμμετρικής κρυπτογραφίας) είναι ο ιδιοκτήτης αυτού. Αυτή η παραδοχή δεν είναι όμως πάντοτε αληθής. Το ίδιο πρόβλημα συναντάται σε όλες τις μεθόδους ταυτοποίησης που βασίζονται στην εξέταση ενός και μόνο χαρακτηριστικού (single-factor authentication). Οποιοσδήποτε χρήστης μπορέσει να αποδείξει ότι κατέχει μια συγκεκριμένη τιμή (ή μορφή) αυτού του χαρακτηριστικού, θα αποκτήσει αυτόματα (μέσω της διαδικασίας ταυτοποίησης) την ταυτότητα που συνδέεται με αυτή την τιμή του χαρακτηριστικού.

Επεκτείνοντας αυτό το σκεπτικό θα μπορούσε κανείς να πει ότι οι χρήστες κρυπτογραφικών μεθόδων ταυτοποίησης συνδέουν μια οντότητα με μια ταυτότητα, όμως η ταυτότητα αυτή δε συνδέεται ποτέ με το σύστημα που συμμετέχει στη διαδικασία ταυτοποίησης (βλ. σχήμα 7.3). Η σύνδεση αυτή, παρόλο που δεν συμβαίνει, θεωρείται δεδομένη.

Σε προγραμματιζόμενα συστήματα όπου επιτρέπεται να εκτελεστεί «ξένο» λογισμικό, η ταυτοποίηση σύμφωνα με κάποια μυστική αλληλουχία από byte είναι μια ιδιαίτερα επισφαλής διαδικασία, καθώς τα byte αυτά μπορεί να υποκλαπούν (ή να αλλοιωθούν) από κακόβουλο λογισμικό. Μάλιστα, η χρήση ενός τέτοιου μυστικού από κάποιο λογισμικό δεν εξασφαλίζει ότι το λογισμικό αυτό είχε κάποια συγκεκριμένη μορφή (π.χ. αποτελεί γνωστό, έμπιστο λογισμικό) ή ότι εκτελέστηκε σε κάποιο συγκεκριμένο περιβάλλον εκτέλεσης. Αυτή η έλλειψη ταυτοποίησης του χρήστη του μυστικού γίνεται ιδιαίτερα προφανής στα Ενεργά Δίκτυα αλλά και στους Κινητούς Πράκτορες, όπου το λογισμικό είναι ικανό να μεταναστεύσει μεταξύ διαφορετικών πλατφόρμων εκτέλεσης, παίρνοντας μαζί του δεδομένα όπως την παραπάνω μυστική αλληλουχία από byte.

Μια διαφορετική μέθοδος ταυτοποίησης είναι αυτή που στηρίζεται στην εξέταση πολλαπλών χαρακτηριστικών (multi-factor authentication). Όπως και στη φύση έτσι και εδώ, μια σειρά από «αισθητήρες» θα καταγράψουν τη μορφή κάποιων χαρακτηριστικών μιας οντότητας. Εξετάζοντας τα χαρακτηριστικά αυτά, η διαδικασία ταυτοποίησης θα «αναγνω-



Σχήμα 7.4: Μοντέλο ταυτοποίησης βάσει πολλαπλών χαρακτηριστικών.

ρίσει» μια οντότητα, δηλαδή θα τη συνδέσει με μια συγκεκριμένη ταυτότητα (με κάποια πιθανότητα σφάλματος). Αυτή η μέθοδος ταυτοποίησης, κάνει πλέον δυνατή τη σύνδεση μιας ταυτότητας με ένα σύστημα, καθώς μπορεί να εξετάσει χαρακτηριστικά του εν λόγω συστήματος στο πλαίσιο της διαδικασίας ταυτοποίησης. Επιπλέον, εξετάζοντας τα φυσικά χαρακτηριστικά ενός συστήματος, μπορεί πλέον η φυσική υπόσταση του συστήματος να συνδεθεί με την ταυτότητα αυτού.

Στην εργασία [163] παρουσιάστηκε για πρώτη φορά μια πλατφόρμα, η οποία χρησιμοποιεί την παραπάνω μέθοδο ταυτοποίησης για να προστατέψει κινητούς κόμβους από επιθέσεις πλαστοπροσωπείας σε ασύρματα ad hoc δίκτυα (MANET). Εξετάζοντας μεταξύ άλλων και φυσικά χαρακτηριστικά των κόμβων, η πλατφόρμα επιτρέπει την ασφαλέστερη σύνδεση ταυτοτήτων με τις συσκευές ενός ad hoc δικτύου.

Η πλατφόρμα αυτή, παρόλο που μπορεί να επεκταθεί κατάλληλα ώστε να χρησιμοποιηθεί από οποιασδήποτε μορφής δίκτυα, σχεδιάστηκε με γνώμονα τη χρήση από κινητούς κόμβους ασύρματων ad hoc δικτύων, για τρεις κυρίως λόγους:

- Ο τρόπος μετάδοσης δεδομένων σε ασύρματα δίκτυα (εκπομπή σε κοινό μέσο διάδοσης) επιτρέπει σε οποιονδήποτε κόμβο να υποδυθεί κάποιον άλλο κόμβο και καθιστά δύσκολο τον εντοπισμό του πραγματικού κόμβου που απέστειλε κάποια πληροφορία στο δίκτυο.

- Στα δίκτυα ad hoc η εξακρίβωση της ταυτότητας ενός κόμβου είναι ιδιαίτερα σημαντική καθώς η δρομολόγηση των πακέτων εξαρτάται κατά μεγάλο βαθμό από τις ταυτότητες των γειτονικών κόμβων.

- Όπως θα φανερωθεί και παρακάτω, τα ασύρματα ad hoc δίκτυα με κινητούς κόμβους προσφέρουν πλήθος χαρακτηριστικών που μπορούν να εξεταστούν στο πλαίσιο μιας πλατφόρμας ταυτοποίησης πολλαπλών χαρακτηριστικών.

Στόχος της πλατφόρμας είναι η ασφαλής ταυτοποίηση γειτονικών κόμβων που βρίσκονται εντός της επικοινωνιακής εμβέλειας (communication range). Η μέθοδος ταυτοποίησης που υλοποιεί η πλατφόρμα, αποτελείται από δύο φάσεις. Στην πρώτη φάση μια οντότητα στο ad hoc δίκτυο συνδέεται με μια ταυτότητα χρησιμοποιώντας μια ψηφιακή υπογραφή ενός πιστοποιημένου κλειδιού. Στη δεύτερη φάση, εξετάζονται χαρακτηριστικά του κόμβου που παρήγε την ψηφιακή υπογραφή και αν αυτά συμφωνούν με αυτά που περιγράφονται στο υπογεγραμμένο (από την αρχή πιστοποίησης) πιστοποιητικό του κόμβου, τότε συνδέεται η φαινόμενη ως ταυτότητα αυτού με τη φυσική του υπόσταση. Όπως φαίνεται και στο

σχήμα 7.4, με τον τρόπο αυτό ένας κόμβος ταυτοποιείται βάσει δύο ανεξάρτητων παραγόντων.

7.3 Τύποι εξεταζόμενων χαρακτηριστικών

Προκειμένου το αποτέλεσμα μιας διαδικασίας ταυτοποίησης να συνοδεύεται από ένα υψηλό βαθμό αξιοπιστίας θα πρέπει να επιλεχθεί με προσοχή το είδος των χαρακτηριστικών που αυτή θα εξετάσει.

Χαρακτηριστικά που μπορούν να επαληθευθούν από τρίτες οντότητες

Είναι σημαντικό η τιμή ενός χαρακτηριστικού να μη βασίζεται μονάχα σε μια δήλωση του υπό ταυτοποίηση κόμβου καθώς η δήλωση αυτή μπορεί να είναι ψευδής. Για το λόγο αυτό, προτιμούνται χαρακτηριστικά που μπορούν να επαληθευθούν από τρίτες οντότητες είτε με άμεσο τρόπο, είτε με έμμεσο τρόπο. Ένα χαρακτηριστικό που επιδέχεται άμεση εξέταση, μπορεί να ελεγχθεί από κόμβους που βρίσκονται εντός της επικοινωνιακής εμβέλειας του υπο ταυτοποίηση κόμβου. Αντίθετα, ένα χαρακτηριστικό που επιδέχεται έμμεση εξέταση, απαιτεί τη συνεργασία πολλαπλών (έμπιστων) κόμβων κατά τη διερεύνηση της τιμής (ή μορφής) αυτού.

Χαρακτηριστικά που βασίζονται σε ιδιότητες του υλικού και του λογισμικού

Η εξέταση χαρακτηριστικών του υλικού ενός κόμβου είναι συχνά πιο αξιόπιστη από την εξέταση χαρακτηριστικών του λογισμικού αυτού, καθώς το λογισμικό μπορεί να διαμορφωθεί πολύ πιο εύκολα από μια κακόβουλη οντότητα, από ότι το υλικό. Έτσι, κατά την ταυτοποίηση κόμβων προτιμάται η εξέταση χαρακτηριστικών που είναι συνδεδεμένα κυρίως με το υλικό ενός κόμβου.

Κατά τη συμμετοχή όμως ενός κόμβου σε ένα δίκτυο (και ειδικότερα σε ένα προγραμματιζόμενο δίκτυο), το λογισμικό αυτού καθορίζει κατά μεγάλο βαθμό τη συμπεριφορά του και έτσι δε μπορεί να αποκλειστεί πλήρως από τα χαρακτηριστικά που θα εξεταστούν σε μια διαδικασία ταυτοποίησης.

Χρήση σταθμισμένων χαρακτηριστικών

Ο βαθμός αξιοπιστίας ενός χαρακτηριστικού πρακτικά δεν είναι ο ίδιος για κάθε χαρακτηριστικό. Αν ένα χαρακτηριστικό επιτρέπει σε έναν κόμβο να εξαπατήσει με ευκολία κάποιον άλλο κόμβο, τότε το χαρακτηριστικό αυτό λαμβάνει μικρότερο βαθμό αξιοπιστίας από κάποιο άλλο, του οποίου η τιμή έχει μικρότερη πιθανότητα να είναι ψευδής. Για παράδειγμα, χαρακτηριστικά που χρειάζονται ειδικές τροποποιήσεις στο υλικό προκειμένου να αποδώσουν ψευδείς τιμές (π.χ. τιμές που αντιστοιχούν σε άλλους κόμβους) λαμβάνουν υψηλό βαθμό αξιοπιστίας, ενώ χαρακτηριστικά που μπορούν να διαμορφωθούν από κακόβουλους κόμβους με απλές αλλαγές στο λογισμικό, λαμβάνουν μικρότερο βαθμό αξιοπιστίας.

Ο βαθμός αξιοπιστίας κάθε χαρακτηριστικού μπορεί να επηρεαστεί και από αλλαγές στο περιβάλλον. Έτσι, η τιμή ενός χαρακτηριστικού που αφορά το σήμα που εκπέμπει ένας κόμβος, θα συνοδεύεται από χαμηλό βαθμό αξιοπιστίας αν η εξέταση έγινε σε περιβάλλον με χαμηλή ποιότητα σήματος (low signal-to-noice ratio – SNR).

Προτεινόμενα Χαρακτηριστικά

Με βάση τις παραπάνω παρατηρήσεις, μπορεί κανείς να επιλέξει μια σειρά χαρακτηριστικών του υλικού, τα οποία σε συνδυασμό, θα περιγράψουν την ταυτότητα μιας συσκευής σε ένα ασύρματο ad hoc δίκτυο. Τέτοια χαρακτηριστικά είναι τα εξής:

Αποτύπωμα συχνότητας εκπομπής: Οι τεχνικές "Radio Frequency Fingerprint" (RFF) εξάγουν μοναδικά «χαρακτηριστικά» (features) από το σήμα ενός κόμβου, παρέχοντας έτσι ένα «αποτύπωμα» το οποίο μπορεί να χρησιμοποιηθεί για την ταυτοποίηση μιας συσκευής σε ένα δίκτυο MANET. Στις εργασίες [202, 203, 204] αναφέρονται τρείς μέθοδοι (Threshold, Bayesian Step Change Detector και Signal Phase) για την εξαγωγή τέτοιων χαρακτηριστικών από το σήμα ενός κόμβου, όταν αυτό βρίσκεται στα πρώτα στάδια εκπομπής (turn-on transient). Το κάθε αποτύπωμα αντιστοιχεί στο κύκλωμα μιας συγκεκριμένης συσκευής και όχι στο κύκλωμα ενός συγκεκριμένου μοντέλου. Κόμβοι που κατέχουν το κατάλληλο υποσύστημα ώστε να καταγράψουν αυτά τα αποτυπώματα, μπορούν να τα χρησιμοποιήσουν για να ελέγξουν την αυθεντικότητα της ταυτότητας ενός κόμβου που βρίσκεται εντός επικοινωνιακής εμβέλειας. Μάλιστα, τα αποτυπώματα αυτά θα μπορούσαν να χρησιμοποιηθούν για να αντιμετωπιστεί η επίθεση «Κλοπής Πιστοποιητικών» (βλ. ενότητα 7.1.3), καθώς ο κακόβουλος κόμβος θα έχει διαφορετικό αποτύπωμα από αυτό που περιγράφεται στο πιστοποιητικό του κόμβου που υποδύεται.

Εντοπισμός υδατογραφήματος στη συχνότητα εκπομπής: Ένας κόμβος ενός MANET δικτύου μπορεί να διαμορφώσει την εκπομπή του κατάλληλα ώστε να περιλαμβάνει ένα (μοναδικό) υδατογράφημα (radio frequency watermark). Μια τέτοια τεχνική παρουσιάζεται στην εργασία [205], επιτρέποντας σε τρίτους κόμβους να επαληθεύσουν την ταυτότητα κόμβων που μεταδίδουν πληροφορίες εντός της επικοινωνιακής τους εμβέλειας.

Μελέτη των καθυστερήσεων των διαδικασιών του υλικού ενός κόμβου: Στην εργασία [206] περιγράφεται μια τεχνική αναγνώρισης κόμβων η οποία βασίζεται στην καταγραφή των καθυστερήσεων που προκύπτουν κατά την ταυτοποίηση σε δίκτυα κόμβων που επικοινωνούν σύμφωνα με το πρωτόκολλο `IEEE 802.11`[1]. Στα δίκτυα αυτά συμβαίνει μια διμερής ταυτοποίηση μεταξύ ενός κόμβου και του Σημείου Πρόσβασης (Access Point). Μελετώντας το χρονικό διάστημα μεταξύ της αποδοχής της αίτησης ταυτοποίησης ενός κόμβου (authentication acknowledgement) και της αποστολής της αίτησης ταυτοποίησης του Σημείου Πρόσβασης (authentication request), μπορεί κανείς να εξάγει ένα αποτύπωμα για τη συσκευή που παίζει το ρόλο του Σημείου Πρόσβασης. Επειδή η μέθοδος αυτή ασχολείται με το Επίπεδο Πρόσβασης (MAC) δεν απαιτεί την ύπαρξη εξειδικευμένου υλικού στο σύστημα που επιθυμεί να ταυτοποιήσει τα Σημεία Πρόσβασης ενός δικτύου. Θα πρέπει να σημειωθεί πάντως, ότι η μέθοδος της εργασίας [206] συνοδεύεται από τρεις περιορισμούς: α) μπορεί να χρησιμοποιηθεί μόνο σε δίκτυα κόμβων που επικοινωνούν σύμφωνα με το πρωτόκολλο 802.11, β) υποστηρίζει την ταυτοποίηση μόνο των Σημείων Πρόσβασης (και άρα δε μπορεί να χρησιμοποιηθεί αυτούσια σε δίκτυα τύπου MANET) και γ) απαιτεί την υλοποίηση της λειτουργίας MAC της διεπαφής δικτύου του υπό ταυτοποίηση κόμβου, σε υλικό ή firmware (καθώς σε περίπτωση που είναι υλοποιημένη σε λογισμικό θα μπορούσε να διαμορφωθεί η καθυστέρηση κατά τέτοιο τρόπο ώστε να μοιάζει με αυτή κάποιου άλλου κόμβου). Παρόλα αυτά η βασική αρχή αυτής της μεθόδου (δηλαδή η εξέταση της καθυστέρησης

[1] http://www.ieee802.org/11

γνωστού λογισμικού ή πρωτοκόλλου σε ένα συγκεκριμένο σύστημα) είναι αξιοσημείωτη.

Γεωγραφική κάλυψη: Η *γεωγραφική κάλυψη* ενός κόμβου είναι ένα σύνθετο χαρακτηριστικό που προέρχεται από το συνδυασμό της *γεωγραφικής* του *θέσης* και της *επικοινωνιακής* του *εμβέλειας*. Η ακριβής θέση ενός κόμβου (υπογεγραμμένα στοιχεία χρόνου και τοποθεσίας) μπορεί να ανακαλυφθεί είτε με τη χρήση τεχνολογίας GPS είτε με τριγωνοποίηση, χρησιμοποιώντας δεδομένα από έμπιστους κόμβους του δικτύου [207]. Η μέγιστη ακτίνα επικοινωνίας ενός κόμβου εξαρτάται από το μοντέλο και τον κατασκευαστή της MANET συσκευής. Χρησιμοποιώντας το συνδυασμό αυτών των δύο πληροφοριών, μπορεί κανείς να εξετάσει τη γεωγραφική κάλυψη ενός κόμβου και να διαπιστώσει αν δύο κόμβοι έχουν τη δυνατότητα να επικοινωνήσουν μεταξύ τους απευθείας. Σε περίπτωση που κάτι τέτοιο υπονοείται από την επικοινωνία τους αλλά δεν διαπιστώνεται από τη γεωγραφική τους κάλυψη, ενδέχεται να αποτελούν θύματα μιας επίθεσης «Αόρατου Κόμβου» (βλ. ενότητα 7.1.2).

Σε περιβάλλοντα όπου απαιτείται ο έλεγχος χαρακτηριστικών του λογισμικού ενός κόμβου μπορούν να χρησιμοποιηθούν τεχνικές εντοπισμού της έκδοσης του λειτουργικού συστήματος του κόμβου [208, 209] καθώς και των υπηρεσιών αυτού [210]. Οι τεχνικές αυτές βασίζονται στο είδος των απαντήσεων που αποστέλλει ένας κόμβος όταν λαμβάνει συγκεκριμένες αιτήσεις. Επίσης, μπορούν να χρησιμοποιηθούν παραδοσιακές τεχνικές όπως ο έλεγχος ψηφιακών υπογραφών σε υποδομές δημοσίου κλειδιού. Όμως, όπως περιγράφηκε και παραπάνω, τα χαρακτηριστικά αυτά λαμβάνουν χαμηλό βαθμό αξιοπιστίας καθώς μπορούν εύκολα να γίνουν αντικείμενο παραποίησης. Μια τεχνική που συνδυάζει την ευελιξία της υποδομής δημοσίου κλειδιού με την ασφαλή λειτουργία του υλικού είναι οι πλατφόρμες *TPE* που εξετάστηκαν στην ενότητα 5.2, οι οποίες εκτελούν τη διαδικασία υπογραφής σε ειδικό υποσύστημα του υλικού.

Στον πίνακα 7.1 παρουσιάζονται συνοπτικά τα προτερήματα και τα μειονεκτήματα των χαρακτηριστικών που εξετάστηκαν στην ενότητα αυτή.

7.4 Πλατφόρμα ταυτοποίησης

Η προτεινόμενη πλατφόρμα προσφέρει μια υπηρεσία ταυτοποίησης η οποία συνδυάζει την κρυπτογραφική ταυτοποίηση κόμβων (έλεγχος ψηφιακών υπογραφών και πιστοποιητικών) με την εξέταση χαρακτηριστικών (φυσικών και μη) αυτών. Στόχος της υπηρεσίας είναι η ταυτοποίηση γειτονικών κόμβων και συγκεκριμένα, ο υπολογισμός μιας τιμής η οποία περιγράφει το βαθμό αξιοπιστίας που κατέχει η σύνδεση ενός κόμβου με μια ταυτότητα.

7.4.1 Βασικές παραδοχές

Οι κόμβοι που θα συμμετάσχουν στην πλατφόρμα ταυτοποίησης θα πρέπει να έχουν τη δυνατότητα να δημιουργήσουν και να ελέγξουν ψηφιακές υπογραφές. Σε περίπτωση όπου οι κόμβοι έχουν περιορισμένες υπολογιστικές δυνατότητες (π.χ. αισθητήρες), μπορούν να χρησιμοποιηθούν αλγόριθμοι κρυπτογραφίας ελλειπτικών καμπύλων (elliptic curve cryptography), οι οποίοι έχουν μειωμένες απαιτήσεις σε επεξεργαστική ισχύ [211].

Επίσης, όλοι οι κόμβοι θα πρέπει να είναι εξοπλισμένοι με τα απαραίτητα υποσυστήματα που επιτρέπουν τον έλεγχο των χαρακτηριστικών τρίτων κόμβων. Το είδος και το πλήθος των υποσυστημάτων αυτών εξαρτάται από το πεδίο χρήσης της πλατφόρμας ταυτοποίησης. Έτσι, σε στρατιωτικές εφαρμογές όπου απαιτείται υψηλότερος βαθμός αξιοπιστίας προς

Χαρακτηριστικό	Συνδέεται με συγκεκριμένη συσκευή;	Μπορεί να πλαστογραφηθεί;	Επηρεάζεται από τη χαμηλή ποιότητα σήματος; (SNR)	Απαιτείται η ύπαρξη $3^{ης}$ έμπιστης οντότητας;
Ψηφιακή Υπογραφή (PKI)	Όχι[1]	Ναι, βλ. εν. 7.1.3	Όχι	Ναι, η Αρχή Πιστοποίησης
Αναγνώριση Λειτ. Συστήματος	Όχι	Ναι	Όχι	Όχι
Αναγνώριση Εφαρμογών	Όχι	Ναι	Όχι	Όχι
Αποτύπωμα Συχνότητας Εκπομπής (RFF)	Ναι	Όχι[2]	Ναι, βλ. [204]	Όχι
Υδατογράφημα στη Συχνότητα Εκπομπής (RF watermark)	Ναι	Όχι[3]	Ναι[4]	Όχι
Μελέτη Καθυστερήσεων των Διαδικασιών του Υλικού	Ναι	Όχι[2]	Όχι	Όχι
Γεωγραφική Κάλυψη	Ναι	Όχι[5]	Όχι	Ναι

[1] με εξαίρεση τις TPE

[2] απαιτείται ειδική διαμόρφωση του υλικού

[3] απαιτείται η ύπαρξη ενός παραμετροποιήσιμου συστήματος μετάδοσης

[4] σε συνθήκες με χαμηλό SNR απαιτείται μεγαλύτερος ρυθμός μετάδοσης του υδατογραφήματος

[5] υπάρχει η πιθανότητα να παρέχει ανακριβή δεδομένα εξαιτίας της κίνησης των κόμβων

Πίνακας 7.1: Προτερήματα και μειονεκτήματα χρήσης διαφόρων χαρακτηριστικών των κόμβων σε διαδικασίες ταυτοποίησης.

τη διαδικασία ταυτοποίησης, προβλέπεται η χρήση περισσότερων υποσυστημάτων ελέγχου που εξετάζουν φυσικά χαρακτηριστικά των κόμβων.

7.4.2 Στάδιο αρχικοποίησης

Αρχικά, κάθε κόμβος εφοδιάζεται με ένα ζεύγος κλειδιών (δημόσιο PK_i, ιδιωτικό SK_i) κατάλληλων για χρήση σε σύστημα κρυπτογράφησης δημοσίου κλειδιού.

Προτού εισαχθεί ένας νέος κόμβος στο δίκτυο MANET, θα πρέπει να καταγραφούν τα χαρακτηριστικά αυτού, μέσω μιας σειράς μετρήσεων που θα λάβουν χώρα σε ελεγχόμενο περιβάλλον. Η διαδικασία αυτή μπορεί να γίνει είτε από τον κατασκευαστή του κόμβου, είτε από κάποια άλλη έμπιστη οντότητα.

Τα καταγεγραμμένα χαρακτηριστικά ενός κόμβου καθώς και το δημόσιο κλειδί αυτού γίνονται μέρος ενός πιστοποιητικού που εκδίδει (και υπογράφει) μια κατανεμημένη Αρχή Πιστοποίησης [212]. Το πιστοποιητικό αυτό θα συνοδεύει τον κόμβο κατά τη συμμετοχή του στο δίκτυο και κατ'επέκταση, κατά τη συμμετοχή του στην πλατφόρμα ταυτοποίησης.

7.4.3 Διαδικασία ταυτοποίησης

Έστω οι κόμβοι A και B, οι οποίοι πληρούν τις προϋποθέσεις που περιγράφηκαν στην προηγούμενη παράγραφο και συμμετέχουν σε ένα δίκτυο με τα υπογεγραμμένα πιστοποιητικά

$Cert_A$ και $Cert_B$ αντίστοιχα.

Έστω, ότι ο κόμβος A επιθυμεί να ταυτοποιήσει τον κόμβο B. Η διαδικασία με την οποία γίνεται αυτό στην προτεινόμενη πλατφόρμα, περιγράφεται από τα παρακάτω 6 βήματα:

1. Ο κόμβος A αποστέλλει ένα τυχαίο μήνυμα (τύπου challenge) στον κόμβο B.

2. Ο κόμβος B απαντά σε αυτό με μια υπογεγραμμένη έκδοση του challenge (τύπου response). Επίσης, αποστέλλει το πιστοποιητικό του $Cert_B$, στον A.

3. Ο κόμβος A επιβεβαιώνει ότι η υπογραφή στην απάντηση προήλθε από το κλειδί που περιγράφεται στο $Cert_B$. Αν κάτι τέτοιο δεν ισχύει, το στάδιο της ταυτοποίησης τερματίζει πρόωρα και η σύνδεση του κόμβου B με τα στοιχεία του πιστοποιητικού $Cert_B$ δεν ολοκληρώνεται. Διαφορετικά, η διαδικασία συνεχίζει στο επόμενο βήμα.

4. Ο κόμβος A εξετάζει τα χαρακτηριστικά του κόμβου B με μια σειρά μετρήσεων. Σε περίπτωση που κάποιο χαρακτηριστικό δε μπορεί να μετρηθεί άμεσα, αυτό εξετάζεται σε συνεργασία με τρίτες έμπιστες οντότητες, όπως περιγράφθηκε στην ενότητα 7.3.

5. Μόλις καταγραφούν τα χαρακτηριστικά του κόμβου B, αυτά συγκρίνονται με τις τιμές που περιγράφονται στο πιστοποιητικό $Cert_B$. Από τη σύγκριση αυτή προκύπτει μια τιμή p_j η οποία περιγράφει την ομοιότητα της μορφής ενός καταγεγραμμένου χαρακτηριστικού j, με τη μορφή που αυτό κατείχε στο πιστοποιητικό $Cert_B$. Η τιμή p_j δε θα χρησιμοποιηθεί ως έχειν, αλλά θα επηρεαστεί από το βαθμό αξιοπιστίας (στάθμιση) που συνδέεται με το συγκεκριμένο χαρακτηριστικό.

6. Η διαδικασία ταυτοποίησης θα παράξει τελικά μια τιμή (μεταξύ 0 και 1) η οποία περιγράφει το βαθμό αξιοπιστίας προς την κρυπτογραφική σύνδεση που έγινε στο βήμα 3. Η τιμή αυτή θα προκύψει από το συγκερασμό των σταθμισμένων τιμών p_j.

Έτσι, το επιβεβαιωμένο κλειδί που χρησιμοποιήθηκε για την υπογραφή του μηνύματος αποτελεί τον πρώτο παράγοντα ταυτοποίησης, ενώ τα χαρακτηριστικά που εξετάστηκαν αποτελούν ένα περαιτέρω παράγοντα ταυτοποίησης, που ενισχύει (ή όχι) τη σύνδεση ενός κόμβου με μια ταυτότητα.

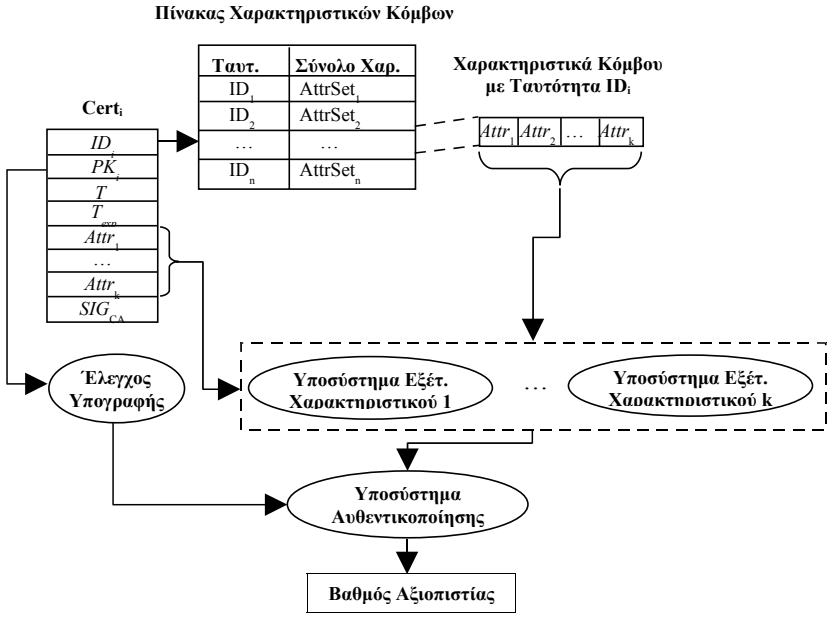
7.5 Θέματα υλοποίησης

Το πιστοποιητικό ενός κόμβου μπορεί να έχει τη μορφή ενός πιστοποιητικού τύπου X.509[2], όπου τα χαρακτηριστικά του κόμβου περιγράφονται σαν *περαιτέρω ιδιότητες* (additional attributes) αυτού. Γενικότερα, το υπογεγραμμένο (από την αρχή πιστοποίησης – CA) πιστοποιητικό ενός κόμβου i έχει την εξής μορφή:

$$Cert_i = (ID_i, PK_i, T, T_{exp}, Attr_1, ..., Attr_k)_{CA}$$

όπου ID_i είναι η μοναδική ταυτότητα του κόμβου i, PK_i είναι το δημόσιο κλειδί του κόμβου, T, T_{exp} είναι οι χρόνοι έκδοσης και λήξης του πιστοποιητικού και $Attr_x$ είναι μια τιμή που περιγράφει τη μορφή του χαρακτηριστικού x. Προφανώς, στο πιστοποιητικό ενός κόμβου μπορούν να περιγραφούν μονάχα τα χαρακτηριστικά που παραμένουν σταθερά για έναν κόμβο, όπως π.χ. είναι η επικοινωνιακή εμβέλεια αυτού. Προκειμένου να μπορούν να εξεταστούν χαρακτηριστικά τα οποία δε παραμένουν σταθερά κατά τη συμμετοχή του κόμβου

[2] http://www.itu.int/rec/T-REC-X.509



Σχήμα 7.5: Διαχείριση χαρακτηριστικών των κόμβων.

στο δίκτυο (π.χ. γεωγραφική τοποθεσία ενός κόμβου) απαιτείται η παροχή πληροφοριών σχετικά με αυτά από έμπιστες πηγές (υπογεγραμμένη τοποθεσία από υπηρεσία GPS κλπ.).

Κάθε κόμβος που συμμετέχει στην πλατφόρμα ταυτοποίησης διατηρεί μια συλλογή από καταγεγραμμένες τιμές χαρακτηριστικών (attributes). Οι τιμές αυτές είναι είτε *Διακριτές*, (π.χ. ένας ακέραιος, μια συμβολοσειρά, μια διεύθυνση IP), είτε έχουν τη μορφή *Συνόλου* (π.χ. "ακρότατες" τιμές σήματος), είτε περιγράφουν ένα *Εύρος* Τιμών (π.χ. ισχύς σήματος). Οι τιμές αυτές αποθηκεύονται σε έναν *Πίνακα Χαρακτηριστικών Κόμβων*, όπως φαίνεται στο σχήμα 7.5. Ο πίνακας αποθηκεύει για κάθε κόμβο ένα *Σύνολο Χαρακτηριστικών*, από το οποίο ένα κοινά αποδεκτό υποσύνολο θα εξεταστεί στα πλαίσια της διαδικασίας ταυτοποίησης. Είναι αρκετά χρήσιμο να διατηρείται ένα ιστορικό των παλαιών τιμών ενός χαρακτηριστικού ενός κόμβου, καθώς το λογισμικό μπορεί να ανατρέξει σε αυτές αν π.χ. κάποια μέτρηση έγινε υπό κακές συνθήκες.

Η τιμή κάθε χαρακτηριστικού καταγράφεται από ένα ειδικό υποσύστημα που ονομάζεται *Υποσύστημα Εξέτασης Χαρακτηριστικού* (Attribute Module). Έτσι, για τον κόμβο με ταυτότητα ID_i, το Υποσύστημα Εξέτασης Χαρακτηριστικού που εξετάζει το χαρακτηριστικό j, θα προσθέσει μια τιμή $Attr_j$ στο Σύνολο Χαρακτηριστικών του κόμβου αυτού. Στη συνέχεια θα συγκρίνει την τιμή του χαρακτηριστικού με αυτή που περιγράφεται στο πιστοποιητικό του κόμβου i και θα παραγάγει μια τιμή p_j η οποία περιγράφει την πιθανότητα το καταγεγραμμένο χαρακτηριστικό να ανήκει στον κόμβο που περιγράφεται στο πιστοποιητικό.

Το *Υποσύστημα Αυθεντικοποίησης* (Authentication Module) αναλαμβάνει να συνδέσει σε κάθε πιθανότητα p_j, ένα βάρος w_j, το οποίο περιγράφει το βαθμό αξιοπιστίας προς τη συγκεκριμένη καταγραφή του χαρακτηριστικού. Από το συγκερασμό των σταθμισμένων (εξαιτίας του βαθμού αξιοπιστίας) πιθανοτήτων p_j, προκύπτει ο τελικός *Βαθμός Αξιοπιστίας* $C(ID_i)$ προς τη φαινόμενη ως ταυτότητα του κόμβου i. Ο συγκερασμός των πιθανοτήτων p_j μπορεί να επιτευχθεί με ένα σταθμισμένο μέσο όρο σαν τον παρακάτω:

$$C(ID_i) = \frac{\sum_{j=1}^{n} w_j p_j}{\sum_{j=1}^{n} w_j},$$

όπου n είναι ο αριθμός των χαρακτηριστικών που λήφθησαν υπόψιν. Μια πιο "ευφυής" μέθοδος συγκερασμού, μπορεί να λάβει υπόψιν της τις σχέσεις μεταξύ των διαφόρων χαρακτηριστικών και να υπολογίσει την τιμή $C(ID_i)$ βάσει ενός δικτύου τύπου Bayesian [213].

Η ταυτότητα ενός κόμβου θεωρείται γνήσια όταν ο τελικός βαθμός αξιοπιστίας που έχει προκύψει για αυτή είναι μεγαλύτερος από κάποιο όριο θ. Το όριο αυτό εξαρτάται από το είδος της εφαρμογής που χρησιμοποιεί την πλατφόρμα ταυτοποίησης. Έτσι, σε στρατιωτικές εφαρμογές το όριο θ θα λάβει υψηλότερη τιμή απ'ότι σε μια πλατφόρμα ταυτοποίησης που χρησιμοποιείται σε ένα οικιακό δίκτυο (καθώς το ρίσκο μιας λανθασμένης ταυτοποίησης στο δεύτερο είναι σαφώς μικρότερο). Όσο αυξάνει η τιμή του θ τόσοι περισσότεροι έλεγχοι θα πρέπει να γίνουν σε κάθε χαρακτηριστικό (ώστε να αυξηθεί ο βαθμός αξιοπιστίας του αποτελέσματος). Επίσης, σε περιπτώσεις όπου γίνονται μετρήσεις υπό κακές περιβαλλοντικές συνθήκες, η τιμή θ θα πρέπει να προσαρμοστεί κατάλληλα ώστε η διαδικσία ταυτοποίησης να φέρει το επιθυμητό αποτέλεσμα.

Ένα Υποσύστημα Χαρακτηριστικού απαιτεί άμεση πρόσβαση σε χαμηλού επιπέδου πληροφορία (π.χ. στοιχεία του οδηγού μιας δικτυακής διεπαφής) και έτσι προτείνεται η υλοποίηση αυτού ως μέρος του πυρήνα του λειτουργικού συστήματος ενός κόμβου. Κάθε υποσύστημα χαρακτηριστικού προσφέρει στον προγραμματιστή μια διεπαφή (API) μέσω της οποίας: α) οι εφαρμογές μπορούν να λάβουν τις τιμές που υπολόγισε το υποσύστημα, β) οι εφαρμογές μπορούν να παραμετροποιήσουν το υποσύστημα, γ) τα δύο υποσυστήματα μπορούν να ανταλλάξουν πληροφορίες μεταξύ τους και δ) οι διαδικασίες του πυρήνα μπορούν να περάσουν δεδομένα ή συναρτήσεις (callbacks) σε αυτό. Τέλος, τα υποσυστήματα χαρακτηριστικών μπορούν να υλοποιηθούν ως υπηρεσίες ενος Ενεργού Κόμβου. Η προγραμματιζόμενη φύση των Ενεργών Κόμβων επιτρέπει μεταξύ άλλων την αποστολή λογισμικού στους εξεταζόμενους κόμβους και την καταγραφή χαρακτηριστικών με τοπικό τρόπο.

7.6 Ποιοτική αξιολόγηση

Στην εργασία [212] των Zhou και Haas παρουσιάζεται μια μέθοδος ταυτοποίησης για κόμβους που συμμετέχουν σε MANET, που στηρίζεται στη χρήση κρυπτογραφίας (συγκεκριμένα, στηρίζεται στην κρυπτογραφία κατωφλίου – *threshold cryptography*) και στην ύπαρξη μιας κατανεμημένης Αρχής Πιστοποίησης. Παρομοίως, οι Hubaux et al. στην εργασία [214] συνδυάζουν ένα δίκτυο εμπιστοσύνης με ψηφιακά πιστοποιητικά ώστε να δημιουργήσουν ένα ιστό εμπιστοσύνης (web of trust), από τον οποίο θα εξάγουν συμπεράσματα για την ταυτότητα ενός κόμβου. Παρόλο που και οι δύο τεχνικές είναι κατανεμημένες και μπορούν να εφαρμοστούν σε δίκτυα μεγάλης κλίμακας, δε λαμβάνουν υπόψιν τους την ευκολία με την οποία μπορεί κάποιος να εκμεταλλευτεί το κοινό μέσο μετάδοσης και έτσι παραμένουν και οι δύο ευάλωτες σε επιθέσεις πλαστοπροσωπείας, όπως αυτή του «Αόρατου Κόμβου». Επίσης, εφόσον οι παραπάνω τεχνικές συνδέουν την ταυτότητα ενός κόμβου μονάχα με ένα πιστοποιητικο, είναι επιπρόσθετα ευάλωτες και σε επιθέσεις Κλοπής Πιστοποιητικών.

Η πλατφόρμα ταυτοποίησης που παρουσιάστηκε στο κεφάλαιο αυτό λύνει τα παραπάνω προβλήματα συνδέοντας τη φαινόμενη ως ταυτότητα ενός κόμβου, με τη φυσική του

υπόσταση. Η σύνδεση αυτή γίνεται με την εξέταση συγκεκριμένων χαρακτηριστικών του κόμβου, ο τύπος των οποίων ορίζεται ανάλογα με την εφαρμογή. Κάθε χαρακτηριστικό εξετάζεται από ένα ειδικό υποσύστημα που βρίσκεται εγκατεστημένο στον κόμβο. Η ύπαρξη αυτού του υποσυστήματος μπορεί να αυξήσει το κόστος κατασκευής ενός κόμβου (π.χ. σύστημα αναγνώρισης αποτυπώματος στην συχνότητα εκπομπής ενός κόμβου) και, για το λόγο αυτό, η επιλογή των εξεταζόμενων χαρακτηριστικών αποτελεί συχνά ένα συμβιβασμό μεταξύ του επιθυμητού επιπέδου ασφάλειας (π.χ. υψηλό επίπεδο ασφάλειας για στρατιωτικές εφαρμογές) και του κόστους κατασκευής.

Η αρχιτεκτονική της προτεινόμενης πλατφόρμας επιτρέπει την εύκολη ενσωμάτωση νέων μεθόδων εξέτασης χαρακτηριστικών, καθώς κάθε Υποσύστημα Εξέτασης Χαρακτηριστικού λειτουργεί ανεξάρτητα από τα άλλα και πάντοτε σύμφωνα με το πρωτόκολλο μιας προκαθορισμένης διεπαφής (API). Μάλιστα, μετά τη δημοσίευση της σχετικής εργασίας [163] που παρουσιάζει την αρχικτεκτονική της προτεινόμενης πλατφόρμας, επακολούθησαν ερευνητικές εργασίες όπως αυτή του B. Sieka [206], οι οποίες εισήγαγαν νέες μεθόδους εξέτασης χαρακτηριστικών για τους κόμβους που συμμετέχουν στην παραπάνω πλατφόρμα ταυτοποίησης.

Τα χαρακτηριστικά που θα εξετάσουν οι κόμβοι κατά τη συμμετοχή τους στην πλατφόρμα ταυτοποίησης είναι συγκεκριμένα και έχουν οριστεί και συμφωνηθεί από όλους τους κόμβους προτού αυτοί χρησιμοποιήσουν την πλατφόρμα. Επίσης, η καταγραφή των χαρακτηριστικών ενός κόμβου γίνεται από μια τρίτη έμπιστη οντότητα, πρίν την είσοδο του κόμβου στο ad hoc δίκτυο. Αυτές οι δύο παραδοχές περιορίζουν αρκετά το είδος των δικτύων στα οποία μπορεί να εφαρμοστεί η πλατφόρμα, καθώς θεωρούν ότι το δίκτυο θα απαρτίζεται από συγκεκριμένους κόμβους οι οποίοι θα είναι γνωστοί εξ αρχής και ότι οι κόμβοι αυτοί θα είναι πιστοποιημένοι από (και θα εμπιστεύονται) μια συγκεκριμενη οντότητα. Στην εργασία [215] των Wishart et al. προτείνονται δύο τεχνικές οι οποίες επιτρέπουν την άρση των παραπάνω περιορισμών και τη χρήση της πλατφόρμας σε δυναμικά περιβάλλοντα από ομότιμους κόμβους. Συγκεκριμένα, οι συγγραφείς της εργασίας προτείνουν την ταυτοποίηση των κόμβων ανά ζευγάρια όπου κάθε ζευγάρι συμφωνεί σε ένα κοινό υποσύνολο χαρακτηριστικών που θα εξεταστούν. Έτσι, επιτρέπεται η ύπαρξη κόμβων στο δίκτυο με διαφορετικά Υποσυστήματα Εξέτασης Χαρακτηριστικών. Επίσης, χρησιμοποιείται η τεχνική κρυπτογράφησης *threshold cryptography* σύμφωνα με την οποία τουλάχιστον k κόμβοι θα πρέπει να συμφωνήσουν ώστε να δημιουργηθεί μια υπογραφή για έναν κόμβο. Με τον τρόπο αυτό δεν απαιτείται πλέον η ύπαρξη μιας τρίτης έμπιστης οντότητας για την έκδοση των πιστοποιητικών των κόμβων, καθώς οι κόμβοι του ίδιου του δικτύου μπορούν να πιστοποιήσουν ανά πάσα στιγμή τα χαρακτηριστικά ενός κόμβου.

Θα πρέπει να σημειωθεί ότι η προτεινόμενη πλατφόρμα έχει ένα ιδιαίτερα ευρύ πεδίο εφαρμογών. Εφαρμογές που διαχειρίζονται τις συσκευές ενός ad hoc δικτύου (asset management) μπορούν να τη χρησιμοποιήσουν για να λάβουν ασφαλείς περιγραφές για τους κόμβους που απαρτίζονν το δίκτυο. Επίσης, πέραν της λειτουργίας της ως εγγυήτρια σε διαδικασίες ταυτοποίησης, μπορεί επίσης να παρέχει υπηρεσίες ασφαλούς ταυτοποίησης σε δικτυακά συστήματα ανίχνευσης επιθέσεων. Τέλος, είναι ιδιαίτερα χρήσιμη και σε "δημοκρατικές" διαδικασίες (π.χ. ηλεκτρονική ψηφοφορία, αξιολόγηση εναλλακτικών μονοπατιών δρομολόγησης σε MANET δίκτυα) όπου μπορεί να διασφαλίσει τη μοναδική συμμετοχή ενός κόμβου σε μια διαδικασία.

Κεφάλαιο 8

Συμπεράσματα και Μελλοντικές Κατευθύνσεις

Οι προγραμματιζόμενες δικτυακές υποδομές επιτρέπουν τη δυναμική προσαρμογή του δικτύου ως προς τις παρεχόμενες υπηρεσίες, εξασφαλίζοντας στους χρήστες μια πιο αποδοτική χρήση του δικτυακού μέσου. Ο προγραμματισμός των δικτυακών υποδομών μπορεί να γίνει είτε μέσω λογισμικού είτε μέσω ρυθμίσεων που οι χρήστες παρέχουν σε αυτές. Σε κάθε περίπτωση, δίνεται η δυνατότητα στους χρήστες να προσαρμόσουν σε ένα βαθμό τη λειτουργία του μέσου ως προς τις δικές τους ανάγκες.

Η δυναμική φόρτωση λογισμικού στους κόμβους του δικτύου επιτρέπει τη δημιουργία νέων καινοτόμων υπηρεσιών που δρουν συνεργατικά με τις υπάρχουσες. Μάλιστα, γίνεται δυνατή και η σύνθεση υπηρεσιών από ήδη υπάρχουσες, οι οποίες μπορεί να εκτελούνται στον ίδιο ή σε διαφορετικούς κόμβους. Η δυνατότητα αυτή ανοίγει νέους ορίζοντες για τη δημιουργία κατανεμημένων εφαρμογών που μπορούν να εξυπηρετήσουν τις ανάγκες μεγάλου αριθμού χρηστών. Επίσης, θέτει τις βάσεις για τη δημιουργία μιας ανοιχτής και ελεύθερα προγραμματιζόμενης υποδομής από την οποία θα μπορούσε να επωφεληθεί μεταξύ άλλων και η παγκόσμια ερευνητική κοινότητα.

Κύριος εκφραστής της φιλοσοφίας των προγραμματιζόμενων δικτυακών υποδομών είναι τα Ενεργά Δίκτυα. Οι κόμβοι των Ενεργών Δικτύων είναι εξοπλισμένοι με ειδικά περιβάλλοντα εκτέλεσης, όπου μπορεί να φορτωθεί λογισμικό για την εξυπηρέτηση διαφόρων αναγκών των χρηστών του δικτύου. Μάλιστα, στα Ισχυρά Ενεργά Δίκτυα, οι υπηρεσίες αυτές είναι εφήμερες και υλοποιούνται μέσω ειδικού λογισμικού που εμπεριέχεται στα πακέτα της δικτυακής κίνησης. Παρόλες τις προτάσεις της ερευνητικής κοινότητας σχετικά με την αξιοποίηση τέτοιων υποδομών για την παροχή ευφυών υπηρεσιών, τα προβλήματα ασφάλειας και απόδοσης που συνδέονται με αυτές δεν έχουν επιτρέψει την ευρύτερη υιοθέτησή τους. Σήμερα, πλατφόρμες για τον προγραμματισμό κατανεμητών όπως η OpenFlow [216], επιλέγουν την υποστήριξη περιορισμένων δυνατοτήτων προγραμματισμού, ώστε μαζί με τα οφέλη της προγραμματιζόμενης αρχιτεκτονικής να μπορούν να προσφέρουν και την απαραίτητη ποιότητα υπηρεσιών.

Στο πλαίσιο της παρούσας διατριβής διερευνήθηκαν δύο είδη προγραμματιζόμενων αρχιτεκτονικών για την υλοποίηση δικτυακών υπηρεσιών. Το πρώτο είδος αρχιτεκτονικής ανήκει στην οικογένεια των Ασθενών Ενεργών Δικτύων και εκμεταλλεύεται κόμβους που βρίσκονται στον κορμό ενός δικτύου. Παράδειγμα αυτής της αρχιτεκτονικής είναι η προταθείσα πλατφόρμα Mobile Active Mail, η οποία προσφέρει τη δυνατότητα δυναμικής δρομολόγησης ηλεκτρονικών μηνυμάτων (e-mail) στο δίκτυο υποδοχής του παραλήπτη. Το δεύτερο είδος αρχιτεκτονικής που διερευνήθηκε ανήκει στην οικογένεια των Ισχυρών Ενερ-

γών Δικτύων και βασίζεται σε Ενεργούς Κόμβους που βρίσκονται στα άκρα του δικτύου. Η προταθείσα πλατφόρμα SEDUCE αξιοποιεί τη δυνατότητα ελεύθερου προγραμματισμού των Ενεργών Κόμβων αυτής της αρχιτεκτονικής για να προσφέρει υψηλού επιπέδου υπηρεσίες ανίχνευσης πολυμορφικού κακόβουλου λογισμικού. Και οι δύο εφαρμογές αποτελούν εξαιρετικά παραδείγματα των εξειδικευμένων εκείνων υπηρεσιών που μπορούν να προσφερθούν από προγραμματιζόμενες δικτυακές υποδομές. Λειτουργούν συνεργατικά με τις ήδη υπάρχουσες υπηρεσίες και συνεισφέρουν στη συνολική αναβάθμιση των υπηρεσιών ενός δικτύου. Επίσης, αποτελούν πεδίο εφαρμογής για μια σειρά προτεινόμενων μέτρων τα οποία καθιστούν δυνατή την παροχή αξιόπιστων και ασφαλών υπηρεσιών πάνω από προγραμματιζόμενες δικτυακές υποδομές.

Συγκεκριμένα, η πλατφόρμα Mobile Active Mail συνεργάζεται με την τεχνολογία Mobile IP για να προωθήσει τα μηνύματα που προορίζονται για ένα χρήστη στο δίκτυο υποδοχής αυτού, αποφεύγοντας πλήρως την προβληματική τριγωνική δρομολόγηση μέσω του οικείου δικτύου. Ο Ενεργός Κόμβος που βρίσκεται στο δίκτυο υποδοχής μπορεί να προβεί σε περαιτέρω επεξεργασία των μηνυμάτων ώστε αυτά να μετασχηματιστούν σε μια μορφή πιο κατάλληλη για τη συσκευή του χρήστη. Επίσης, μέσω των Ενεργών Κόμβων που βρίσκονται σε άλλα σημεία του κορμού του δικτύου, μπορούν να εφαρμοστούν φίλτρα στην εισερχόμενη αλληλογραφία ώστε να μειωθούν σημαντικά τα ανεπιθύμητα μηνύματα. Οι υπηρεσίες των Ενεργών Κόμβων εκτελούνται ως νήματα σε ένα προστατευμένο περιβάλλον εκτέλεσης και ακολουθούν ένα συγκεκριμένο πρότυπο κατά την επικοινωνία με τον Ενεργό Κόμβο και τις υπόλοιπες Ενεργές Υπηρεσίες. Στην περίπτωση όπου απαιτείται η εκτέλεση μιας υπηρεσίας που δεν υπάρχει διαθέσιμη σε ένα Κόμβο, αυτή μπορεί να μεταφορτωθεί από σχετικό πάροχο υπηρεσιών. Η πλατφόρμα δημιουργεί ένα κανάλι ασφαλούς επικοινωνίας κατά τη μεταφορά λογισμικού ή ρυθμίσεων χρηστών και ελέγχει επίσης τη γνησιότητα του περιεχομένου αυτών. Η λιτή αρχιτεκτονική της πλατφόρμας καθώς και η πλήρη συμβατότητα αυτής με τα ήδη υπάρχοντα πρωτόκολλα αποστολής και παραλαβής μηνυμάτων ηλ. ταχυδρομείου συνέβαλαν σε μεγάλο βαθμό στην επιτυχή έκβαση του πειράματος αξιολόγησης της πλατφόρμας, για τις ανάγκες του οποίου εγκαταστάθηκαν κόμβοι σε δίκτυα τεσσάρων ευρωπαϊκών οργανισμών.

Η πλατφόρμα Mobile Active Mail θα μπορούσε να συνδυαστεί με μια σειρά από νέες τεχνολογίες ώστε να παρέχει στους χρήστες αυτής αυξημένες δυνατότητες. Για παράδειγμα, το Mobile Active Overlay τμήμα της πλατφόρμας θα μπορούσε να αναβαθμιστεί κατάλληλα ώστε να υποστηρίζει το πρωτόκολλο HMIPv6 [217] που προσφέρει υπηρεσίες Mobile IP σε συστήματα που χρησιμοποιούν το πρωτόκολλο IPv6. Επίσης, η επιλογή έμπιστου παρόχου υπηρεσιών θα μπορούσε να γίνεται μέσω του δικτύου εμπιστοσύνης TwoHop, που προτάθηκε στο κεφάλαιο 6, ώστε να λαμβάνεται υπόψιν και το επίπεδο της υπηρεσίας που παρέχει ο πάροχος. Η υπηρεσία που ασχολείται με την εφαρμογή φίλτρων στα ηλεκτρονικά μηνύματα θα μπορούσε να συνδυαστεί με συνεργατικά δίκτυα αξιολόγησης ηλεκτρονικών μηνυμάτων [218]. Η πληροφορία από αυτά αυτά τα δίκτυα θα μπορούσε να γίνεται διαθέσιμη στον Ενεργό Κόμβο μέσω μιας Ενεργής Υπηρεσίας. Χρήσιμη, επίσης, θα ήταν η δημιουργία μιας γλώσσας για την περιγραφή σύνθετων φίλτρων καθώς και η υλοποίηση αποδοτικών μέτρων προστασίας του περιβάλλοντος εκτέλεσης από επιθέσεις τύπου άρνησης εξυπηρέτησης, που θα μπορούσαν να προκληθούν από την εφαρμογή συγκεκριμένων φίλτρων.

Η δεύτερη αρχιτεκτονική προγραμματιζόμενων δικτυακών υποδομών που εξετάστηκε ήταν αυτή της πλατφόρμας SEDUCE. Η πλατφόρμα SEDUCE αποτελεί μια κατανεμημένη πλατφόρμα που εντοπίζει πολυμορφικό κακόβουλο λογισμικό (shellcode) σε πακέτα δικτυακής κίνησης. Η πιο σημαντική δυνατότητα της πλατφόρμας είναι αυτή της δυναμικής ανάλυσης δικτυακών πακέτων, η οποία επιτρέπει τον εντοπισμό κακόβουλου λογισμικού

δίχως τη χρήση υπογραφών ή στατιστικών μοντέλων. Κατά την πειραματική αξιολόγηση, η πλατφόρμα εντόπισε επιτυχώς κάθε κακόβουλο payload που εξέτασε, ενώ εντόπισε επίσης και payload που είχαν μεταλλαχθεί με διάφορες μεθόδους. Η μέθοδος δυναμικής ανάλυσης της πλατφόρμας στηρίζεται στον εντοπισμό επικίνδυνων κλήσεων συστήματος. Η τεχνική αυτή συνοδεύεται από μικρή πιθανότητα ψευδών ανιχνεύσεων (false positives), ενώ εξασφαλίζει παράλληλα την εγγυημένη ανίχνευση ενός shellcode όταν αυτό έχει εκτελεστεί σωστά. Κατά τη σύγκριση με άλλες τεχνικές που έχουν προταθεί στη βιβλιογραφία [94], βρέθηκε ότι είναι πιο ακριβής κατά την ανίχνευση κακόβουλου λογισμικού αλλά και (σε αρκετές περιπτώσεις) πιο χρονοβόρα.

Η δεύτερη σημαντική δυνατότητα της πλατφόρμας SEDUCE είναι αυτή του καταμερισμού εργασίας. Για τις χρονοβόρες μορφές ανάλυσης (όπως η δυναμική ανάλυση), η αρχιτεκτονική της προγραμματιζόμενης πλατφόρμας SEDUCE επιτρέπει το διαμοιρασμό της σχετικής εργασίας σε περισσότερα από ένα συστήματα, αξιοποιώντας ακόμη και συστοιχίες υπολογιστών όταν αυτές είναι διαθέσιμες. Επίσης, καταμερισμός εργασίας μπορεί να συμβεί και κατά τη διαδικασία συλλογής πακέτων, επιτρέποντας έτσι την εξέταση δεδομένων ακόμη και σε δίκτυα υψηλών ταχυτήτων.

Οι Ενεργοί Κόμβοι της πλατφόρμας SEDUCE χρησιμοποιούν ένα περιβάλλον εικονικής εκτέλεσης εντολών επεξεργαστή για τη δυναμική ανάλυση και εξέταση των ύποπτων δεδομένων. Το περιβάλλον αυτό απομονώνει πλήρως το ύποπτο λογισμικό από το υπόλοιπο σύστημα, προστατεύοντας τόσο τη λειτουργία του Ενεργού Κόμβου όσο και τις αναλύσεις άλλων δεδομένων που συμβαίνουν παράλληλα. Επιθέσεις τύπου άρνησης εξυπηρέτησης δεν είναι δυνατές κατά την εικονική εκτέλεση καθώς το περιβάλλον τερματίζει αυτόματα μια ανάλυση όταν αυτή δεν έχει ολοκληρωθεί εντός ενός συγκεκριμένου χρονικού διαστήματος.

Η πλατφόρμα SEDUCE είναι εύκολα επεκτάσιμη και μπορεί να προστεθούν σε αυτή νέοι μηχανισμοί ανίχνευσης κακόβουλων εντολών. Μάλιστα, η πλατφόρμα δίνει τη δυνατότητα εξέτασης κίνησης και υψηλότερου επιπέδου (π.χ. πρωτοκόλλων εφαρμογής) οπότε θα μπορούσε κάποιος να την αξιοποιήσει και για την ανίχνευση άλλου τύπου επιθέσεων, όπως "SQL injection" [109]. Για την αύξηση της απόδοσης της πλατφόρμας θα μπορούσαν να διαμορφωθούν οι συλλέκτες κίνησης με τον τρόπο που περιγράφεται στην εργασία [100], ενώ οι Ενεργοί Κόμβοι θα μπορούσαν να αξιοποιήσουν τις δυνατότητες ιδεατής εκτέλεσης (virtualization) που διαθέτουν σύγχρονοι επεξεργαστές [219]. Άλλη μια τεχνική που θα μπορούσε να αυξήσει την απόδοση της πλατφόρμας είναι αυτή της άμεσης αναγνώρισης μη κακόβουλων πακέτων (εξετάζοντας τα αποτελέσματα προηγούμενων αναλύσεων ή άλλα στοιχεία). Για τη χρήση της πλατφόρμας σε ερευνητικές δραστηριότητες, θα ήταν επίσης χρήσιμη η ύπαρξη μιας γλώσσας μέσω της οποίας θα μπορούσαν να συνδυαστούν συγκεκριμένοι μηχανισμοί ανίχνευσης κακόβουλου λογισμικού όταν τα περιεχόμενα ενός πακέτου πληρούν κάποιες προϋποθέσεις. Τέλος, ανοιχτό πρόβλημα παραμένει η ανίχνευση shellcode που χρησιμοποιεί πληροφορίες από το περιβάλλον εκτέλεσης για να φέρει εις πέρας το έργο του.

Στο πεδίο της οργάνωσης και διαχείρισης δικτυακών προγραμματιζόμενων υποδομών παρουσιάστηκαν δύο αλγόριθμοι που έχουν ως στόχο το χρονοπρογραμματισμό των συσκευών σε δίκτυα αισθητήρων κατά τέτοιο τρόπο ώστε να γίνεται η μικρότερη δυνατή κατανάλωση ενέργειας. Ο πρώτος αλγόριθμος, που ονομάζεται B{GOP} παράγει ομάδες αισθητήρων χωρίς κοινά στοιχεία και είναι ιδιαίτερα αποδοτικός. Χαρακτηριστικό του αλγορίθμου είναι η στρατηγική επιλογής κόμβων που ακολουθεί, η οποία αποφεύγει με τρόπο «ελαστικό» την επιλογή συγκεκριμένων κόμβων που πιθανότατα θα δημιουργήσουν προβλήματα κατά τη δημιουργία ομάδων κάλυψης. Τα πειραματικά δεδομένα έδειξαν ότι ο αλγόριθμος αυτός παράγει περισσότερες ομάδες κάλυψης από τον αντίστοιχο της εργασίας [129]. Επίσης, ο χρόνος εκτέλεσης αυτού επηρεάζεται σε μικρότερο βαθμό από αλλαγές

στον αριθμό αισθητήρων ή στόχων.

Ο δεύτερος αλγόριθμος που παρουσιάστηκε ήταν ο CCF, ο οποίος επιτρέπει τη συμμετοχή ενός αισθητήρα σε περισσότερες από μία ομάδες κάλυψης. Η πειραματική αξιολόγηση έδειξε ότι ο CCF μπορεί να παραγάγει περισσότερες ομάδες κάλυψης από τον αντίστοιχο αλγόριθμο της εργασίας [134], χρησιμοποιώντας κάθε αισθητήρα λιγότερες φορές. Η συνολική εξέταση των πειραματικών δεδομένων έδειξε ότι στις περισσότερες περιπτώσεις ο αλγόριθμος B{GOP} παρήγαγε εντός μικρού χρονικού διαστήματος ένα ικανοποιητικό αριθμό ομάδων κάλυψης. Για την παραγωγή ακόμη περισσότερων ομάδων κάλυψης, συνιστάται η χρήση της (πιο αργής) βελτιστοποιημένης έκδοσης του αλγορίθμου CCF.

Στο πλαίσιο εξέτασης τυχαιοποιημένων αλγορίθμων, παρουσιάστηκε και ένας τρίτος αλγόριθμος, ο B{GOP}-random ο οποίος επιλέγει τυχαίους αισθητήρες με πιθανότητα ανάλογη του αποτελέσματος της αντικειμενικής συνάρτησης του αλγορίθμου B{GOP}. Ο αλγόριθμος αυτός έδειξε ιδιαίτερα μειωμένη απόδοση κατά την πειραματική αξιολόγηση και χαρακτηρίζεται από υψηλότερη πολυπλοκότητα από αυτή του αλγορίθμου B{GOP}.

Η υλοποίηση των παραπάνω αλγορίθμων καθώς και το περιβάλλον προσομοίωσης αυτών έχουν γίνει διαθέσιμα ως ελεύθερο λογισμικό, ώστε να μπορέσουν να βασιστούν σε αυτά για το έργο τους και άλλοι ερευνητές. Ένα ενδιαφέρον πρόβλημα που θα μπορούσε να μελετηθεί περαιτέρω, είναι η μοντελοποίηση των συνθηκών που μειώνουν τον αριθμό των ομάδων κάλυψης. Επίσης, θα μπορούσαν να διερευνηθούν περαιτέρω οι κατανεμημένοι αλγόριθμοι του χώρου, καθώς και αλγόριθμοι που εξασφαλίζουν ότι κάθε ομάδα κάλυψης θα έχει συνολική κατανάλωση ενέργειας εντός συγκεκριμένων ορίων.

Το δεύτερο μέρος της διατριβής ασχολήθηκε με θέματα ασφάλειας στις προγραμματιζόμενες δικτυακές υποδομές. Αφού παρουσιάστηκαν μέτρα από τη βιβλιογραφία που προστατεύουν τον κόμβο, τις υπηρεσίες αυτού αλλά και το χρήστη από κακόβουλες ενέργειες, εξετάστηκε το θέμα της εμπιστοσύνης προς ένα πάροχο υπηρεσιών. Συγκεκριμένα, σε προγραμματιζόμενα δικτυακά περιβάλλοντα που βασίζονται σε ανοιχτά πρότυπα (open systems architectures) ο οποιοσδήποτε κόμβος μπορεί να παίξει το ρόλο του παρόχου υπηρεσιών για κάποιο άλλο κόμβο. Επειδή στα συστήματα αυτά εκτελούνται κάποιες αυτοματοποιημένες διαδικασίες μεταξύ των κόμβων (π.χ. μεταφορά λογισμικού), ανά κόμβο ορίζονται κάποιοι τρίτοι κόμβοι ως «έμπιστοι». Στην πραγματικότητα, όμως, η ποιότητα των υπηρεσιών που παρέχουν οι παραπάνω «έμπιστοι» κόμβοι μπορεί να αλλάξει ανά πάσα στιγμή, θέτοντας ενδεχομένως σε κίνδυνο τους παραλήπτες των υπηρεσιών. Για το λόγο αυτό η επιλογή ενός παρόχου υπηρεσιών δε θα πρέπει να εξαρτάται από μια στατική σχέση εμπιστοσύνης.

Τα δίκτυα εμπιστοσύνης είναι ιδιαίτερα χρήσιμα σε τέτοιες περιπτώσεις καθώς επιτρέπουν σε ένα κόμβο να επιλέξει τον κατάλληλο πάροχο υπηρεσιών βάσει προηγούμενων αξιολογήσεων που έχουν γίνει από τον ίδιο ή από άλλους κόμβους. Οι αξιολογήσεις αυτές μαρτυρούν το επίπεδο των υπηρεσιών ενός παρόχου, όπως αυτό διαπιστώθηκε σε προηγούμενες αλληλεπιδράσεις με αυτόν. Στη διατριβή αυτή παρουσιάστηκε το TwoHop, ένα δίκτυο εμπιστοσύνης που έχει σχεδιαστεί σύμφωνα με τις ανάγκες μεγάλων προγραμματιζόμενων δικτύων. Το TwoHop θέτει ένα μέγιστο όριο στο μήκος των μονοπατιών εμπιστοσύνης, κάνοντας έτσι αποδοτικό τον υπολογισμό τιμών εμπιστοσύνης ακόμη και σε δίκτυα με πολλές συμμετοχές. Επίσης υποστηρίζει πολλαπλές τιμές εμπιστοσύνης ανά πάροχο, οι οποίες χαρακτηρίζουν διαφορετικές υπηρεσίες αυτού. Κάθε κόμβος μπορεί να διατηρεί τις δικές του αξιολογήσεις ως προς τρίτους κόμβους. Μπορεί όμως να υπολογίσει και την συνολική τιμή εμπιστοσύνης προς ένα πάροχο, όπως αυτή διαμορφώνεται από τις αξιολογήσεις άλλων κόμβων στο δίκτυο εμπιστοσύνης του. Οι αξιολογήσεις αυτές συνδυάζονται βάσει τοπικών βαρών.

Τα ιδιαίτερα χαρακτηριστικά του δικτύου TwoHop περιγράφηκαν με αλγεβρικό τρόπο

χρησιμοποιώντας τη μέθοδο της εργασίας [185] και συγκρίθηκαν με αυτά άλλων δικτύων εμπιστοσύνης από τη βιβλιογραφία. Κατά την πειραματική αξιολόγηση η μέθοδος υπολογισμού εμπιστοσύνης του TwoHop βρέθηκε ιδιαίτερα αποδοτική ακόμη και όταν χρησιμοποιήθηκε σε δίκτυα εμπιστοσύνης με μεγάλο αριθμό κόμβων. Επίσης, βρέθηκε ταχύτερη αυτής των εργασιών [178] και [182]. Πέραν της αποδοτικής μεθόδου υπολογισμού των τιμών εμπιστοσύνης, το TwoHop διακρίνεται και για την ανθεκτικότητά του ενάντια σε μια σειρά επιθέσεων. Επιθέσεις όπως αυτή της «αυτο-προώθησης», της «δυσφήμισης», των «συνομωτικών κόμβων» και της «αθώωσης» μπορούν να αντιμετωπιστούν επαρκώς μέσω των δυνατοτήτων αξιολόγησης που παρέχει το δίκτυο TwoHop. Οι επιθέσεις «αντιφατικής συμπεριφοράς» και "on-off" μπορεί επίσης να αντιμετωπιστούν με συγκεκριμένα μέτρα στην εφαρμογή που χρησιμοποιεί το TwoHop. Για την αντιμετώπιση της επίθεσης "Sybil" απαιτείται από την εφαρμογή που θα χρησιμοποιήσει το δίκτυο TwoHop η δυνατότητα ελέγχου της ταυτότητας ενός κόμβου, ώστε ένας κακόβουλος κόμβος να μη μπορεί να συμμετέχει στο δίκτυο εμπιστοσύνης με περισσότερες από μία ταυτότητες.

Στο πλαίσιο μελλοντικής έρευνας επί του δικτύου εμπιστοσύνης TwoHop θα μπορούσαν να εξεταστούν στρατηγικές για την αυτόματη ανανέωση των αξιολογήσεων προς άλλους κόμβους. Μια πρώτη έρευνα σε αυτό το χώρο έχει γίνει στην εργασία [188].

Στην ενότητα 7.2 προτάθηκε μια μέθοδος ταυτοποίησης στοιχείων για κόμβους που συμμετέχουν σε ασύρματα δίκτυα επικοινωνιών, η οποία μπορεί να αντιμετωπίσει επιθέσεις όπως τη "Sybil", την επίθεση του «αόρατου κόμβου» αλλά και την επίθεση «κλοπής πιστοποιητικών». Η μέθοδος ταυτοποίησης αυτή στηρίζεται στην εξέταση μιας σειράς χαρακτηριστικών, τα οποία περιλαμβάνουν και φυσικά χαρακτηριστικά των κόμβων (όπως το αποτύπωμα εκπομπής [202, 203, 204]). Κόμβοι που συμμετέχουν σε κρίσιμες εφαρμογές, θα μπορούν πλέον μέσω ενός πιστοποιητικού που περιγράφει αυτά τα χαρακτηριστικά να αποδεικνύουν την ταυτότητά τους σε τρίτους κόμβους. Το πιστοποιητικό αυτό συνδέει με κρυπτογραφικό δεσμό την ψηφιακή ταυτότητα ενός κόμβου με την υλική υπόσταση αυτού, επιτρέποντας έτσι στα συστήματα να επικοινωνούν με μεγαλύτερη ασφάλεια μεταξύ τους.

Η εργασία [206] επέκτεινε την παραπάνω πλατφόρμα ταυτοποίησης εισάγοντας νέους τύπους χαρακτηριστικών. Αντίθετα, στην εργασία [215] διαμορφώθηκε η διαδικασία ταυτοποίησης κατά τέτοιο τρόπο ώστε αυτή να μην εξαρτάται πλέον από μια κεντρική αρχή που εκδίδει πιστοποιητικά. Επίσης, στην ίδια εργασία δόθηκε στους κόμβους η δυνατότητα να ταυτοποιηθούν βάσει χαρακτηριστικών που θα συμφωνούσαν κατά την έναρξη της διαδικασίας ταυτοποίησης. Η έρευνα σε αυτό τον τομέα θα μπορούσε να επεκταθεί με την πειραματική υλοποίηση του παραπάνω συστήματος βάσει χαρακτηριστικών που παρέχονται σήμερα από ασύρματους ελεγκτές δικτύου.

Συνολικά, οι προγραμματιζόμενες δικτυακές υποδομές έφεραν νέες δυνατότητες αλλά και νέες προκλήσεις στο χώρο των δικτύων δεδομένων. Η παρούσα διατριβή προσπάθησε να σκιαγραφήσει τα ιδιαίτερα χαρακτηριστικά αυτών των αρχιτεκτονικών και πρότεινε λύσεις σε συγκεκριμένα προβλήματα που παρουσιάζονται κατά τη λειτουργία τους. Κρίνοντας από τη σημερινή τάση μεταφοράς κρίσιμων συστημάτων προς υποδομές συννέφου, φαίνεται ότι έχει ανοίξει πλέον ο δρόμος για την ανοικοδόμηση ιδεατών υποδομών πάνω σε κοινόχρηστους επεξεργαστικούς πόρους. Με τη χρήση προγραμματιζόμενων δικτυακών υποδομών, οι διαχειριστές των υποδομών συννέφου θα μπορούν να προχωρήσουν στην απελευθέρωση και των κοινόχρηστων δικτυακών πόρων, δίνοντας στους χρήστες τη δυνατότητα να διαχειριστούν το σύνολο πλέον της πληροφοριακής τους υποδομής.

Παράρτημα Α΄

Συμπληρωματικές Αποδείξεις

Α΄.1 Ο ευρετικός αλγόριθμος B{GOP} είναι ικανός να παραγάγει τουλάχιστον μία ομάδα κάλυψης, όταν υπάρχουν οι απαραίτητοι αισθητήρες για τη συγκρότηση αυτής

Έστω ότι T_0 είναι το σύνολο των υπό παρακολούθηση στόχων και S_0 το σύνολο των διαθέσιμων αισθητήρων. Για κάθε αισθητήρα s_j, το σύνολο P_j περιγράφει τους στόχους t_i που αυτός ο αισθητήρας μπορεί να καλύψει.

Κατά τη διάρκεια εκτέλεσης του αλγορίθμου, το σύνολο S_{cur} περιέχει τους αισθητήρες που παραμένουν διαθέσιμοι για χρήση στην υπό κατασκευή ομάδα κάλυψης C_{cur}. Αντίστοιχα, το σύνολο T_{cur} περιέχει τους στόχους που δεν έχουν καλυφθεί ακόμη από την ομάδα κάλυψης C_{cur}.

Αν το σύνολο G είναι μια ομάδα κάλυψης, τότε ισχύει:

$$\forall\ t_i \in T_0,\ \exists\ s_j \in G : t_i \in P_j, \tag{Α΄.1}$$

όπου $G \subseteq S_0$, $G \neq \emptyset$, $P_j \subseteq T_0$ και $P_j \neq \emptyset$.

Έστω ότι σε ένα σενάριο δικτύου αισθητήρων είναι δυνατή η σύνθεση μιας ομάδας κάλυψης G (η οποία περιέχει αισθητήρες από το S_0 που μπορούν να καλύψουν όλους τους στόχους του T_0) αλλά ο ευρετικός αλγόριθμος δεν είναι ικανός να την παράξει. Αυτό σημαίνει ότι κατά την παραγωγή της πρώτης ομάδας κάλυψης, ο αλγόριθμος δεν θα μπορέσει να βρει έναν αισθητήρα s_j, ικανό για να καλύψει ένα υποσύνολο των στόχων του T_{cur}, δηλ.:

$$\forall t_i \in T_{cur}, \nexists s_j \in S_{cur} : t_i \in P_j. \tag{Α΄.2}$$

Εφόσον πρόκειται για την παραγωγή της πρώτης ομάδας κάλυψης, όσοι αισθητήρες δεν περιέχονται στο σύνολο S_{cur} θα αποτελούν μέλη της υπό κατασκευήν ομάδας C_{cur}, δηλ.:

$$S_0 = C_{cur} \cup S_{cur},\ C_{cur} \cap S_{cur} = \emptyset. \tag{Α΄.3}$$

Οι στόχοι που έχουν καλυφθεί ήδη από αισθητήρες του συνόλου C_{cur} δε θα αποτελούν πλέον μέρος του συνόλου T_{cur}. Έτσι, από το (Α΄.3) προκύπτει:

$$\forall t_i \in T_{cur}, \nexists s_j \in C_{cur} : t_i \in P_j. \tag{Α΄.4}$$

Από τις (Α΄.2), (Α΄.3) και (Α΄.4) συνεπάγεται:

$$\forall t_i \in T_{cur}, \nexists s_i \in S_0 : t_i \in P_j. \tag{Α΄.5}$$

Όμως, η πρόταση (Α΄.5) είναι ψευδής, καθώς η πρόταση (Α΄.1) μπορεί να ξαναγραφεί για $T_{cur} \subseteq T_0$ και $G \subseteq S_0$, ως εξής:

$$\forall t_i \in T_{cur}, \exists s_j \in S_0 : t_i \in P_j. \tag{Α΄.6}$$

Επομένως, η αρχική υπόθεση είναι ψευδής· Αν υπάρχουν διαθέσιμοι αισθητήρες για τουλάχιστον μία ομάδα κάλυψης, τότε ο ευρετικός αλγόριθμος θα είναι ικανός να την παραγάγει. □

Α΄.2 Υπολογισμός πιθανότητας εύρεσης εγγραφής σε πορτφόλιο δικτύου εμπιστοσύνης με ακτίνα δύο βημάτων

Έστω δύο τυχαίοι κόμβοι x και y σε ένα δίκτυο εμπιστοσύνης με ακτίνα δύο βημάτων. Για να υπολογιστεί η πιθανότητα εύρεσης μιας εγγραφής που αφορά στον πάροχο y στο δίκτυο εμπιστοσύνης του κόμβου x, θα πρέπει πρώτα να υπολογιστεί η πιθανότητα $P_0(x, y)$, δηλαδή η πιθανότητα ύπαρξης μιας τέτοιας εγγραφής στο πορτφόλιο του κόμβου x.

Α΄.2.1 Πιθανότητα εύρεσης μιας εγγραφής στο Βασικό Πορτφόλιο

Έστω N κόμβοι ενός δικτύου εμπιστοσύνης τύπου TwoHop. Το πορτφόλιο κάθε κόμβου αποτελείται από k εγγραφές οι οποίες μπορεί να περιέχουν και στοιχεία για τον ίδιο τον κόμβο. Η σειρά με την οποία αποθηκεύονται οι εγγραφές σε ένα πορτφόλιο δεν είναι σημαντική, οπότε το πλήθος των συνδυασμών από κόμβους που μπορεί να βρεθούν ως εγγραφές σε ένα πορτφόλιο είναι:

$$S_0 = \binom{N}{k}.$$

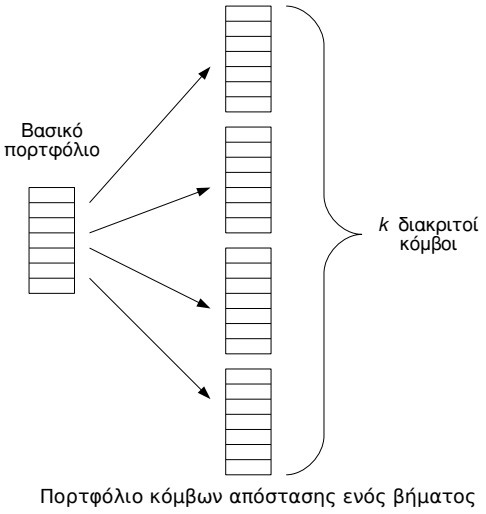
Χάριν συντομίας, οι συνδυασμοί αυτοί θα ονομαστούν *στιγμιότυπα* ενός πορτφόλιο.

Έστω ότι ο κόμβος y έχει την ονομασία `A`. Αν από κάθε εγγραφή ενός πορτφόλιο εξαχθεί η ονομασία του κόμβου στον οποίο αναφέρεται η εγγραφή, τότε θα προκύψει μια μη ταξινομημένη λίστα της μορφής `[ A, C, D, B, E, ...]` , όπου τα A, B, C, D, E θα είναι ονομασίες κόμβων. Αν σε κάθε τέτοια λίστα που περιλαμβάνει τον κόμβο `A`, τοποθετηθεί ο κόμβος αυτός πρώτος, τότε η λίστα θα αποκτήσει τη μορφή: $[A, \underbrace{X_1, X_2, X_3, X_4, \ldots}_{k-1 \text{ κόμβοι}}]$. Επειδή ο κόμβος `A` μπορεί να εμφανιστεί μόνο μια φορά σε ένα πορτφόλιο, οι υπόλοιπες $k-1$ εγγραφές του πορτφόλιο θα πρέπει να αναφέρονται σε μέλη του συνόλου των υπόλοιπων $N-1$ κόμβων. Επομένως, το πλήθος των δυνατών στιγμιότυπων ενός πορτφόλιο που περιέχουν τον κόμβο `A` είναι:

$$\binom{N-1}{k-1},$$

και, άρα η πιθανότητα εύρεσης μιας εγγραφής για ένα συγκεκριμένο κόμβο στο βασικό πορτφόλιο είναι:

$$P_0(x, y) = \frac{\binom{N-1}{k-1}}{\binom{N}{k}} = \frac{\binom{N-1}{k-1}}{\frac{N}{k}\binom{N-1}{k-1}} = \frac{k}{N}. \tag{Α΄.7}$$



Σχήμα Α΄.1: Αναπαράσταση των πορτφόλιο που βρίσκονται σε απόσταση ενός βήματος από το βασικό πορτφόλιο

Α΄.2.2 Πιθανότητα εύρεσης μιας εγγραφής σε πορτφόλιο απόστασης ενός βήματος από το Βασικό Πορτφόλιο

Όπως φαίνεται και στο σχήμα Α΄.1, τα πορτφόλιο που βρίσκονται σε απόσταση ενός βήματος από το βασικό πορτφόλιο είναι k (όσες και οι εγγραφές του βασικού πορτφόλιο). Κάθε ένα από αυτά τα μοναδικά πορτφόλιο έχει $\binom{N}{k}$ πιθανά στιγμιότυπα. Συνεπώς, το συνολικό πλήθος των δυνατών στιγμιότυπων των k πορτφόλιο του πρώτου βήματος είναι:

$$S_1 = \binom{N}{k}^k.$$

Κάθε πορτφόλιο έχει $\binom{N-1}{k}$ δυνατά στιγμιότυπα τα οποία **δεν** περιέχουν αναφορές προς κάποιο συγκεκριμένο κόμβο. Το συνολικό πλήθος των δυνατών στιγμιότυπων όλων των πορτφόλιο του πρώτου βήματος, που δεν περιέχουν αναφορές προς αυτό τον κόμβο είναι:

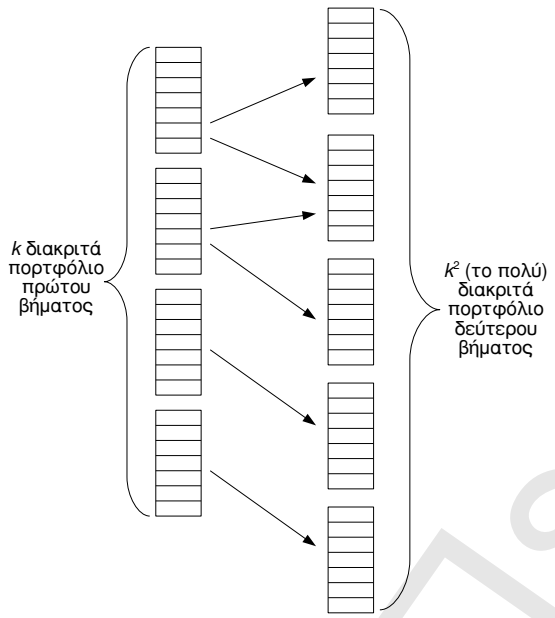
$$M_1 = \binom{N-1}{k}^k,$$

και, άρα, η πιθανότητα να μη βρεθεί κάποια εγγραφή που να περιγράφει αυτό τον κόμβο στα πορτφόλιο του πρώτου βήματος θα είναι:

$$\frac{\binom{N-1}{k}^k}{\binom{N}{k}^k}.$$

Συνεπώς η πιθανότητα να βρεθεί **τουλάχιστον μία** εγγραφή σχετική με αυτό τον κόμβο στα πορτφόλιο του πρώτου βήματος είναι:

$$P_1(x,y) = 1 - \frac{\binom{N-1}{k}^k}{\binom{N}{k}^k} = 1 - \left(\frac{\frac{(N-1)!}{k!(N-1-k)!}}{\frac{N!}{k!(N-k)!}}\right)^k = 1 - (\frac{N-k}{N})^k. \qquad \text{(Α΄.8)}$$



Σχήμα Α΄.2: Αναπαράσταση των πορτφόλιο που βρίσκονται σε απόσταση ενός και δύο βημάτων από το βασικό πορτφόλιο

Α΄.2.3 Πιθανότητα εύρεσης μιας εγγραφής σε πορτφόλιο απόστασης δύο βημάτων από το Βασικό Πορτφόλιο

Δύο πορτφόλιο που βρίσκονται ένα βήμα μακρυά από το βασικό πορτφόλιο ενδέχεται να περιλαμβάνουν εγγραφές που δείχνουν σε κάποιο κοινό κόμβο. Ο κόμβος αυτός εξαιτίας των προηγούμενων πορτφόλιο βρίσκεται δύο βήματα μακρυά από το βασικό πορτφόλιο. Το πλήθος των μοναδικών κόμβων που βρίσκονται δύο βήματα μακρυά από το βασικό κόμβο κυμαίνεται μεταξύ k και k^2, όπως φαίνεται και στο σχήμα Α΄.2.

Η περίπτωση όπου δύο πορτφόλιο πρώτου βήματος αναφέρονται στο ίδιο πορτφόλιο δεύτερου βήματος είναι ισοδύναμη με την περίπτωση όπου τα δύο παραπάνω πορτφόλιο αναφέρονται σε δύο ξεχωριστά πορτφόλιο που έχουν τα ίδια περιεχόμενα. Χρησιμοποιώντας αυτή την ισοδυναμία, μπορεί κανείς εύκολα να υπολογίσει το συνολικό πλήθος από δυνατά στιγμιότυπα των πορτφόλιο δεύτερου βήματος, το οποίο είναι:

$$S_2 = \binom{N}{k}^{k^2}.$$

Για την περίπτωση όπου αυτά τα στιγμιότυπα δεν περιλαμβάνουν κάποια αναφορά για ένα συγκεκριμένο κόμβο, το συνολικό πλήθος δυνατών στιγμιότυπων είναι:

$$M_2 = \binom{N-1}{k}^{k^2}.$$

Συνεπώς, η πιθανότητα εύρεσης έστω και μίας αναφοράς προς τον κόμβο αυτό, στα πορτφόλιο που βρίσκονται σε απόσταση δύο βημάτων από το βασικό πορτφόλιο, θα είναι:

$$P_2(x,y) = 1 - \frac{\binom{N-1}{k}^{k^2}}{\binom{N}{k}^{k^2}} = 1 - (\frac{N-k}{N})^{k^2} \qquad (\text{Α΄.9})$$

Α΄.2.4 Πιθανότητα εύρεσης μιας εγγραφής σε πορτφόλιο που βρίσκεται σε ακτίνα δύο το πολύ βημάτων από το Βασικό Πορτφόλιο

Το συνολικό πλήθος των δυνατών στιγμιοτύπων των πορτφόλιο που βρίσκονται σε απόσταση ενός ή δύο βημάτων από το βασικό πορτφόλιο του κόμβου x, είναι:

$$S_{1,2} = S_1 \cdot S_2 = \binom{N}{k}^k \cdot \binom{N}{k}^{k^2} = \binom{N}{k}^{k+k^2}.$$

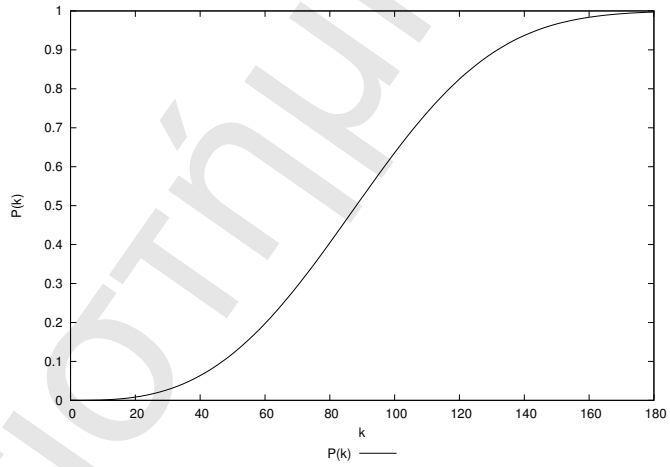
Το πλήθος των στιγμιοτύπων που **δεν περιέχουν** καμμία αναφορά προς τον κόμβο y είναι:

$$M_{1,2} = M_1 \cdot M_2 = \binom{N-1}{k}^k \cdot \binom{N-1}{k}^{k^2} = \binom{N-1}{k}^{k+k^2}.$$

Συνεπώς, η πιθανότητα $P_{1,2}(x,y)$ να υπάρχει τουλάχιστον μία αναφορά για τον κόμβο y σε δίκτυο ακτίνας δύο βημάτων από το βασικό πορτφόλιο του κόμβου x, είναι:

$$P_{1,2}(x,y) = 1 - \frac{\binom{N-1}{k}^{k+k^2}}{\binom{N}{k}^{k+k^2}} = 1 - (\frac{N-k}{N})^{k+k^2}. \quad \text{(Α΄.10)}$$

Στο σχήμα Α΄.3 παρουσιάζεται η γραφική παράσταση της παραπάνω πιθανότητας $P_{1,2}(x,y)$ για ένα δίκτυο με 1.000.000 κόμβους, όπου κάθε κόμβος διατηρεί πορτφόλιο εμπιστοσύνης μεταβλητού μεγέθους k.



Σχήμα Α΄.3: Γραφική παράσταση της $P_{1,2}(x,y)$ για $N = 1.000.000$ κόμβους και μεταβλητό πλήθος k εγγραφών ανά πορτφόλιο.

Γλωσσάρι

a priori (λατ.) εκ των προτέρων.

Ad Hoc (λατ.) Λειτουργία κατ'απαίτηση, ειδικού σκοπού.

API (αγγλ. ακρ. Application Programming Interface) Διεπαφή λογισμικού που απευθύνεται σε προγραμματιστές.

best–effort (αγγλ.) Η καλύτερη δυνατή παροχή υπηρεσίας που δε συνοδεύεται από εγγυήσεις για την ποιότητα αυτής.

challenge-response (αγγλ.) Μέθοδος ταυτοποίησης κατά την οποία η οντότητα A (που παρέχει την ταυτοποίηση) αποστέλλει στην οντότητα B (που θα ταυτοποιηθεί) μια σειρά από τυχαία bytes γνωστά ως "challenge". Η οντότητα B θα απαντήσει σε αυτά τα bytes με ένα μήνυμα γνωστό ως "response" το οποίο περιέχει είτε την τιμή μιας γνωστής συνάρτησης κατακερματισμού που εφαρμόστηκε στο "challenge" και σε ένα κοινό μυστικό, είτε το αποτέλεσμα μιας ψηφιακής υπογραφής στο "challenge" την οποία θα μπορεί να επαληθεύσει η Α μέσω του δημοσίου κλειδιού της B.

DDoS attack (αγγλ. ακρ. Distributed Denial of Service attack) Κατανεμημένη επίθεση τύπου άρνησης εξυπηρέτησης.

DNS (αγγλ. ακρ. Domain Name System) Διαδικτυακή υπηρεσία για τον εντοπισμό εξυπηρετητών που παρέχουν συγκεκριμένες υπηρεσίες. Χρησιμοποιείται μεταξύ άλλων και για την αντιστοίχιση του ονόματος ενός κόμβου με τη διαδικτυακή του διεύθυνση.

GPS (αγγλ. ακρ. Global Positioning System) Παγκόσμιο σύστημα πλοήγησης που χρησιμοποιεί στοιχεία από δορυφόρους.

honeynet (αγγλ.) Δίκτυο συστημάτων που έχει ως στόχο τη συλλογή πληροφοριών και λογισμικού σχετικών με επιθέσεις σε πληροφοριακές υποδομές. Τυπικά παρουσιάζεται ως ένα δίκτυο με ευπαθή συστήματα ώστε να προσελκύσει επιθέσεις από κακόβουλους χρήστες και λογισμικό.

HTTP (αγγλ. ακρ. Hypertext Transfer Protocol) Πρωτόκολλο επικοινωνίας των εξυπηρετητών ιστοσελίδων.

If-Modified-Since (αγγλ.) Ειδική επικεφαλίδα του πρωτοκόλλου `HTTP` έκδοσης 1.1 (`RFC 2068`) η οποία ζητά από τον εξυπηρετητή να αποστείλει ένα έγγραφο μονάχα όταν αυτό είναι νεότερο από κάποιο τοπικό αντίγραφο.

IMAP (αγγλ. ακρ. Internet Message Access Protocol) Πρωτόκολλο για τη μεταφορά μηνυμάτων ηλ. ταχυδρομείου από το διακομιστή στον προσωπικό υπολογιστή ενός χρήστη, με αυξημένες δυνατότητες, όπως η διατήρηση πολλαπλών αρχείων αλληλογραφίας (mail folders) ανά χρήστη.

IP (αγγλ. ακρ. Internet Protocol) Το πρωτόκολλο δρομολόγησης της σουίτας πρωτοκόλλων TCP/IP που χρησιμοποιείται για την επικοινωνία συσκευών στο Διαδίκτυο.

IP Tunnel (αγγλ.) Εικονική δικτυακή ζεύξη που επιτυγχάνεται με την ενθυλάκωση πακέτων κίνησης σε πακέτα του πρωτοκόλλου IP.

karma suicide (αγγλ.) Είδος επίθεσης σε δίκτυα εμπιστοσύνης όπου ένα κόμβος εκμεταλλεύεται την καλή φήμη που έχει για να παρέχει για σύντομο χρονικό διάστημα υπηρεσίες χαμηλής ποιότητας (και συνήθως υψηλού κέρδους). Μόλις ολοκληρωθεί η παροχή των υπηρεσιών αυτών ο κόμβος αλλάζει ταυτότητα και συμμετέχει στο δίκτυο με νέα.

LRU (αγγλ. ακρ. Least Recently Used) Πολιτική αντικατάστασης αντικειμένων που βρίσκονται σε προσωρινή μνήμη, σύμφωνα με την οποία αντικαθιστώνται πρώτα τα αντικείμενα που δεν έτυχαν πρόσφατης χρήσης.

MAC (αγγλ. ακρ. Medium Access Control) Διαχείριση πρόσβασης στο δικτυακό μέσο, όπως αυτή περιγράφεται από το Επίπεδο Ζεύξης Δεδομένων του OSI.

MANET (αγγλ. ακρ. Mobile Ad Hoc Networks) Ad Hoc Δίκτυα Κινητών Κόμβων.

MX (αγγλ. συντ. Mail Exchange) Διακομιστής μεταφοράς μηνυμάτων ηλ. αλληλογραφίας.

online (αγγλ.) Τύπος υπηρεσίας που λειτουργεί σε σύνδεση με το λογισμικό του χρήστη και συνήθως παρέχει άμεση απάντηση στις αιτήσεις του χρήστη στα πλαίσια της ίδιας συνεδρίας.

OpenPGP (αγγλ. ακρ. Open standard for PGP) Ανοιχτό πρότυπο για τη λειτουργία του συστήματος PGP (`RFC 4880`).

OSI (αγγλ. ακρ. Open Systems Interconnection) Μοντέλο αναφοράς Ανοικτής Διασύνδεσης Συστημάτων.

PGP (αγγλ. ακρ. Pretty Good Privacy) Σύστημα που επιτρέπει την κρυπτογραφημένη αποστολή πληροφορίας καθώς και τον έλεγχο της αυθεντικότητας των κρυπτογραφημένων μηνυμάτων, χρησιμοποιώντας τεχνικές ασύμμετρης και συμμετρικής κρυπτογράφησης. Χρησιμοποιείται ευρύτατα για την αποστολή κρυπτογραφημένων μηνυμάτων ηλεκτρονικού ταχυδρομείου.

PKI (αγγλ. ακρ. Public Key Infrastructure) Υποδομή Δημοσίου Κλειδιού.

POP (αγγλ. ακρ. Post Office Protocol) Πρωτόκολλο για τη μεταφορά μηνυμάτων ηλ. ταχυδρομείου από το διακομιστή στον προσωπικό υπολογιστή ενός χρήστη.

RAM (αγγλ. ακρ. Random Access Memory) Μνήμη τυχαίας προσπέλασης.

SHA-1 (αγγλ. ακρ. Secure Hash Algorithm 1) Κρυπτογραφική συνάρτηση κατακερματισμού της οικογένειας SHA που σχεδιάσθηκε από την Εθνική Υπηρεσία Ασφάλειας της Αμερικής (National Security Agency) και κοινοποιήθηκε από το Εθνικό Ίδρυμα Προτύπων και Τεχνολογίας της Αμερικής (National Institute of Standards and Technology) το 1995. Δέχεται είσοδο μεταβλητού μεγέθους και παράγει μια ακολουθία από 160 bits.

SMTP (αγγλ. ακρ. Simple Mail Transfer Protocol) Πρωτόκολλο αποστολής μηνυμάτων ηλεκτρονικού ταχυδρομείου (βλ. `RFC 2821`).

SNMP (αγγλ. ακρ. Simple Network Management Protocol) Πρωτόκολλο διαχείρισης δικτυακών συσκευών (βλ. `RFC 1157`).

socket (αγγλ.) Προγραμματιστική διεπαφή του προτύπου BSD Sockets η οποία επιτρέπει τη μεταφορά δεδομένων μεταξύ διεργασιών που βρίσκονται στο ίδιο ή σε διαφορετικά συστήματα ενός δικτύου.

spam (αγγλ.) Ανεπιθύμητη ηλεκτρονική αλληλογραφία η οποία συνήθως εξυπηρετεί διαφημιστικούς σκοπούς.

TCP (αγγλ. ακρ. Transmission Control Protocol) Πρωτόκολλο για την εγγυημένη μεταφορά δεδομένων πάνω από δίκτυα TCP/IP (βλ. RFC 793).

threshold cryptography (αγγλ.) Μέθοδος κρυπτογραφίας η οποία απαιτεί τη συνδρομή τουλάχιστον k συμμετεχόντων προκειμένου να αποκρυπτογραφηθεί ένα μήνυμα. Με παρόμοιο τρόπο, δημιουργούνται υπογραφές που απαιτούν τη συνδρομή τουλάχιστον k συμμετεχόντων.

UDP (αγγλ. ακρ. User Datagram Protocol) Πρωτόκολλο (μη εγγυημένης) μεταφοράς μηνυμάτων πάνω από δίκτυα TCP/IP (βλ. RFC 768).

Βιβλιογραφία

[1] J. Saltzer, D. Reed, and D. Clark, "End-to-End Arguments in System Design," *ACM Transactions on Computer Systems*, vol. 2, no. 4, pp. 277–288, 1984.

[2] "Oracle Technology Network for Java Developers." `http://www.oracle.com/technetwork/java/index.html`, last accessed May 2012.

[3] D. Chess, C. Harrison, and A. Kershenbaum, "Mobile agents: Are they a good idea?," in *Mobile Object Systems Towards the Programmable Internet*, vol. 1222 of *Lecture Notes in Computer Science*, pp. 25–45, Springer Berlin / Heidelberg, 1997.

[4] J. M. O'Connor and M. Tremblay, "picoJava-I: The Java Virtual Machine in Hardware," *IEEE Micro*, vol. 17, pp. 45–53, 1997.

[5] A. T. Campbell, H. G. D. Meer, M. E. Kounavis, K. Miki, J. B. Vicente, and D. Villela, "A survey of programmable networks," *SIGCOMM Comput. Commun. Rev.*, vol. 29, pp. 7–23, Apr. 1999.

[6] D. L. Tennenhouse and D. J. Wetherall, "Towards an active network architecture," *SIGCOMM Computer Communication Review*, vol. 26, pp. 5–17, Apr. 1996.

[7] "DARPA Active Networks." `http://www.sds.lcs.mit.edu/darpa-activenet/`, last accessed May 2012.

[8] D. Tennenhouse, J. Smith, W. Sincoskie, D. Wetherall, and G. Minden, "A survey of active network research," *IEEE Communications Magazine*, vol. 35, pp. 80 –86, Jan. 1997.

[9] D. Tennenhouse, S. Garland, L. Shrira, and M. Kaashoek, "From Internet to ActiveNet," tech. rep., Laboratory for Computer Science, MIT, 1996.

[10] D. J. Wetherall and D. L. Tennenhouse, "The ACTIVE IP option," in *Proceedings of the 7th workshop on ACM SIGOPS European workshop: Systems support for worldwide applications*, EW 7, (New York, NY, USA), pp. 33–40, ACM, 1996.

[11] D. S. Alexander, B. Braden, C. A. Gunter, A. W. Jackson, A. D. Keromytis, G. J. Minden, and D. Wetherall, "Active Network Encapsulation Protocol (ANEP)." `http://www.cis.upenn.edu/~switchware/ANEP/docs/ANEP.txt`, July 1997. RFC Draft.

[12] M. Hicks, P. Kakkar, J. T. Moore, C. A. Gunter, and S. Nettles, "PLAN: a packet language for active networks," in *ICFP '98: Proceedings of the third ACM SIGPLAN International Conference on Functional programming*, (New York, NY, USA), pp. 86–93, ACM, 1998.

[13] M. Sanders, M. Keaton, S. Bhattacharjee, K. Calvert, S. Zabele, and E. Zegura, "Active reliable multicast on CANEs: a case study," in *Open Architectures and Network Programming Proceedings*, pp. 49–60, IEEE, 2001.

[14] D. J. Wetherall, J. V. Guttag, and D. D. L. Tennenhouse, "Ants: a toolkit for building and dynamically deploying network protocols," in *Proc. of Open Architectures and Network Programming Conference*, pp. 117–129, IEEE, Apr. 1998.

[15] M. Fry and A. Ghosh, "Application level active networking," *Computer Networks*, vol. 31, no. 7, pp. 655–667, 1999.

[16] K. Psounis, "Active networks: Applications, security, safety, and architectures," *IEEE Communications Surveys & Tutorials*, vol. 2, no. 1, pp. 2–16, 1999.

[17] A. Ghosh, M. Fry, and G. MacLarty, "An infrastructure for application level active networking," *Computer Networks*, vol. 36, no. 1, pp. 5–20, 2001.

[18] B. Braden and L. Ricciulli, "A Plan for a Scalable ABone – A modest proposal," tech. rep., USC – Information Science Institute, Jan. 1999.

[19] J. A. Kornblum, D. Raz, and Y. Shavitt, "The active process interaction with its environment," *Computer Networks*, vol. 36, no. 1, pp. 21–34, 2001.

[20] I. W. Marshall and C. Roadknight, "Provision of quality of service for active services," *Computer Networks*, vol. 36, no. 1, pp. 75–85, 2001.

[21] M. Bagnulo, B. Alarcos, M. Calderon, and M. Sedano, "ROSA: Realistic Open Security Architecture for Active Networks," in *Active Networks*, vol. 2546 of *Lecture Notes in Computer Science*, pp. 204–215, Springer Berlin / Heidelberg, 2002.

[22] M. Blaze, J. Feigenbaum, J. Ioannidis, and A. Keromytis, "The Role of Trust Management in Distributed Systems Security," in *Secure Internet Programming*, vol. 1603 of *Lecture Notes in Computer Science*, pp. 185–210, Springer Berlin / Heidelberg, 1999.

[23] D. Wetherall, "Active network vision and reality: lessions from a capsule-based system," *SIGOPS Operating Systems Review*, vol. 33, pp. 64–79, Dec. 1999.

[24] S. Ardon, P. Gunningberg, B. Landfeldt, Y. Ismailov, M. Portmann, and A. Seneviratne, "MARCH: A distributed content adaptation architecture," *International Journal of Communication Systems*, vol. 16, no. 1, pp. 97–115, 2003.

[25] A. Mandal, K. Kennedy, C. Koelbel, G. Marin, J. Mellor-Crummey, B. Liu, and L. Johnsson, "Scheduling strategies for mapping application workflows onto the grid," in *Proc. of 14th IEEE International Symposium on High-Performance Distributed Computing*, vol. 1, pp. 125–134, IEEE Computer Society, 2005.

[26] L. Rudolph, "A Virtualization Infrastructure that Supports Pervasive Computing," *Pervasive Computing*, vol. 8, pp. 8–13, Oct. 2009.

[27] "Amazon Elastic Compute Cloud (Amazon EC2)." `http://aws.amazon.com/ec2`, last accessed June 2011.

[28] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2002.

[29] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson, "Wireless sensor networks for habitat monitoring," in *Proceedings of the International Workshop on Wireless Sensor Networks and Applications*, pp. 88–97, ACM, Sept. 2002.

[30] L. Hai, P. Wan, C.-W. Yi, J. Xiaohua, S. Makki, and N. Pissinou, "Maximal Lifetime Scheduling in Sensor Surveillance Networks," in *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 05)*, vol. 4, pp. 2482–2491, IEEE, Mar. 2005.

[31] A. Goldsmith and S. Wicker, "Design challenges for energy-constrained ad hoc wireless networks," *Wireless Communications*, vol. 9, pp. 8–27, Aug. 2002.

[32] G. Werner-Allen, J. Johnson, M. Ruiz, J. Lees, and M. Welsh, "Monitoring volcanic eruptions with a wireless sensor network," in *Proceeedings of the Second European Workshop on Wireless Sensor Networks*, pp. 108–120, Jan. 2005.

[33] P. Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, and D. Culler, "TinyOS: An operating system for wireless sensor networks," in *Ambient Intelligence*, Springer-Verlag, 2004.

[34] R. Sugihara and R. K. Gupta, "Programming models for sensor networks: A survey," *ACM Transactions on Sensor Networks*, vol. 4, pp. 8:1–8:29, Apr. 2008.

[35] Q. Cao, T. Abdelzaher, J. Stankovic, and T. He, "The LiteOS Operating System: Towards Unix-Like Abstractions for Wireless Sensor Networks," in *Proceedings of the 7th international conference on Information processing in sensor networks (IPSN '08)*, (Washington, DC, USA), pp. 233–244, IEEE Computer Society, 2008.

[36] Y. He, C. Raghavendra, S. Berson, and B. Braden, "A programmable routing framework for autonomic sensor networks," in *Proceedings of the 2003 Autonomic Computing Workshop*, pp. 60–68, June 2003.

[37] A. Boulis, C.-C. Han, and M. B. Srivastava, "Design and implementation of a framework for efficient and programmable sensor networks," in *Proceedings of the 1st international conference on Mobile systems, applications and services (MobiSys '03)*, (New York, NY, USA), pp. 187–200, ACM, 2003.

[38] P. Levis, D. Gay, and D. Culler, "Active sensor networks," in *Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation*, vol. 2 of *NSDI'05*, (Berkeley, CA, USA), pp. 343–356, USENIX Association, 2005.

[39] L. E. Keong, J. Crowcroft, M. Pias, R. Sharma, and S. Lim, "A survey and comparison of peer-to-peer overlay network schemes," *IEEE Communications Surveys & Tutorials*, vol. 7, no. 2, pp. 72–93, 2005.

[40] A. Oram, *Peer-to-Peer: Harnessing the Power of Disruptive Technologies*. Sebastopol, CA, USA: O'Reilly & Associates, Inc., 2001.

[41] D. Benza, M. Cosnard, L. Liquori, and M. Vesin, "Arigatoni: A Simple Programmable Overlay Network," in *John Vincent Atanasoff 2006 International Symposium on Modern Computing (JVA '06)*, pp. 82–91, IEEE, Oct. 2006.

[42] D. Anderson, "BOINC: a system for public-resource computing and storage," in *Proceedings of the Fifth IEEE/ACM International Workshop on Grid Computing*, pp. 4–10, Nov. 2004.

[43] M. Portmann, S. Ardon, and P. Senac, "Programmable Structured Peer-to-Peer Overlay," in *Active and Programmable Networks*, vol. 4388 of *Lecture Notes in Computer Science*, pp. 145–155, Springer Berlin / Heidelberg, 2009.

[44] F. Lau, S. Rubin, M. Smith, and L. Trajkovic, "Distributed denial of service attacks," in *Proceedings of the 2000 IEEE International Conference on Systems, Man, and Cybernetics*, vol. 3, pp. 2275–2280, 2000.

[45] Z. Zhu, G. Lu, Y. Chen, Z. Fu, P. Roberts, and K. Han, "Botnet Research Survey," in *Proceedings of the 32nd Annual International Conference on Computer Software and Applications (COMPSAC '08)*, pp. 967–972, IEEE, Aug. 2008.

[46] A. Gazet, "Comparative analysis of various ransomware virii," *Journal in Computer Virology*, vol. 6, pp. 77–90, 2010.

[47] J. Liu, Y. Xiao, K. Ghaboosi, H. Deng, and J. Zhang, "Botnet: classification, attacks, detection, tracing, and preventive measures," in *Proceedings of the Fourth International Conference on Innovative Computing, Information and Control*, ICICIC '09, (Washington, DC, USA), pp. 1184–1187, IEEE Computer Society, 2009.

[48] N. Weaver, V. Paxson, S. Staniford, and R. Cunningham, "A taxonomy of computer worms," in *Proceedings of the 2003 ACM workshop on Rapid malcode*, WORM '03, (New York, NY, USA), pp. 11–18, ACM, 2003.

[49] J. Nazario and T. Holz, "As the net churns: Fast-flux botnet observations," in *Proceedings of the 3rd International Conference on Malicious and Unwanted Software (MALWARE 2008)*, pp. 24–31, Oct. 2008.

[50] G. Lawton, "On the Trail of the Conficker Worm," *IEEE Computer*, vol. 42, pp. 19–22, June 2009.

[51] "RFC 5944 - IP Mobility Support for IPv4, Revised." `http://tools.ietf.org/rfc/rfc5944.txt`, Nov. 2010.

[52] D. Sterne, K. Djahandari, R. Balupari, W. L. Cholter, B. Babson, B. Wilson, P. Narasimhan, A. Purtell, D. Schnackenberg, and S. Linden, "Active network based DDoS defense," in *Proceedings of DARPA Active Networks Conference and Exposition*, pp. 193–203, 2002.

[53] E. Chen, "AEGIS: An Active-Network-Powered Defense Mechanism against DDoS Attacks," in *Active Networks* (I. Marshall, S. Nettles, and N. Wakamiya, eds.), vol. 2207 of *Lecture Notes in Computer Science*, pp. 1–15, Springer Berlin / Heidelberg, 2001.

[54] X. Fu, W. Shi, A. Akkerman, and V. Karamcheti, "CANS: Composable, Adaptive Network Services Infrastructure," in *Proc. of 3rd USENIX Symposium on Internet Technologies and Systems*, (San Francisco, California), USENIX, Mar. 2001.

[55] D. Glynos, B. Meneklis, T. Berdejoglou, C. Douligeris, C. Boukouvalas, and P. Bosdogianni, "Deployment and Evaluation of Active Mail Services," in *Proceedings of 8th international conference on advances in communication and control (COMCON 8)*, pp. 357–366, 2001.

[56] "RFC 2821 - Simple Mail Transfer Protocol." `http://tools.ietf.org/rfc/rfc2821.txt`, Apr. 2001.

[57] "RFC 1034 - Domain Names - Concepts and Facilities." http://www.ietf.org/rfc/rfc1034.txt, Nov. 1987.

[58] "RFC 3501 - Internet Message Access Protocol - Version 4rev1." http://www.ietf.org/rfc/rfc3501.txt, Mar. 2003.

[59] "RFC 1939 - Post Office Protocol - Version 3." http://www.ietf.org/rfc/rfc1939.txt, May 1996.

[60] A. Juhola, K. Ahola, J. M ols a, and S. Lehtonen, "Mobile active overlay," in *Active Networks* (H. Yasuda, ed.), vol. 1942 of *Lecture Notes in Computer Science*, pp. 416–422, Springer Berlin / Heidelberg, 2000.

[61] T. S. E. Ng and H. Zhang, "Predicting internet network distance with coordinates-based approaches," in *Proc. of Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2002)*, vol. 1, pp. 170–179, IEEE, 2002.

[62] "CASPIAN Project Homepage." http://archive.eurescom.eu/public/projects/P900-series/P926/default.asp, last accessed May 2012.

[63] "Universal tun/tap driver." http://vtun.sourceforge.net/tun/, last accessed May 2012.

[64] "netfilter/iptables project homepage." http://www.netfilter.org/, last accessed May 2012.

[65] "Extensible Markup Language (XML)." http://www.w3.org/XML/, last accessed May 2012.

[66] P. Zimmermann, *The Official PGP User's Guide*. MIT Press, 1995.

[67] C. Cowan, F. Wagle, P. Calton, S. Beattie, and J. Walpole, "Buffer overflows: attacks and defenses for the vulnerability of the decade," in *Proceedings of DARPA Information Survivability Conference and Exposition (DISCEX '00)*, vol. 2, pp. 119–129, 2000.

[68] C. Cowan, M. Barringer, S. Beattie, and G. Kroah-Hartman, "FormatGuard: automatic protection from printf format string vulnerabilities," in *Proceedings of the 10th USENIX Security Symposium*, (Washington, DC), Aug. 2001.

[69] "Using the GNU Compiler Collection - Format Warning Options." http://gcc.gnu.org/onlinedocs/gcc/Warning-Options.html, last accessed May 2012.

[70] A. One, "Smashing the stack for fun and profit," *Phrack Magazine*, vol. 7, no. 49, 1996.

[71] Anonymous, "Once upon a free()...," *Phrack Magazine*, vol. 11, no. 57, 2001.

[72] J. Pincus and B. Baker, "Beyond stack smashing: Recent advances in exploiting buffer overruns," *IEEE Security and Privacy*, vol. 2, no. 4, pp. 20–27, 2004.

[73] S. Bhatkar, D. C. DuVarney, and R. Sekar, "Address obfuscation: an efficient approach to combat a broad range of memory error exploits," in *SSYM'03: Proceedings of the 12th conference on USENIX Security Symposium*, (Berkeley, CA, USA), pp. 8–8, USENIX Association, 2003.

[74] A. Sotirov, "Heap Feng Shui in JavaScript." Black Hat Europe 2007, Amsterdam, the Netherlands, Mar. 2007.

[75] U. Drepper, "Security enhancements in Red Hat Enterprise Linux (beside SElinux)," tech. rep., Dec. 2005. http://people.redhat.com/drepper/nonselsec.pdf.

[76] H. Shacham, M. Page, B. Pfaff, E.-J. Goh, N. dra Modadugu, and D. Boneh, "On the effectiveness of address-space randomization," in *Proceedings of the 11th ACM Conference on Computer and Communications Security*, CCS '04, (New York, NY, USA), pp. 298–307, ACM, 2004.

[77] E. Buchanan, R. Roemer, H. Shacham, and S. Savage, "When good instructions go bad: generalizing return-oriented programming to RISC," in *Proceedings of the 15th ACM conference on Computer and Communications Security*, CCS '08, (New York, NY, USA), pp. 27–38, ACM, 2008.

[78] T. Bletsch, X. Jiang, V. W. Freeh, and Z. Liang, "Jump-oriented programming: a new class of code-reuse attack," in *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security*, ASIACCS '11, (New York, NY, USA), pp. 30–40, ACM, 2011.

[79] C. Cowan, C. Pu, D. Maier, J. Walpole, P. Bakke, S. Beattie, A. Grier, P. Wagle, Q. Zhang, and H. Hinton, "StackGuard: Automatic adaptive detection and prevention of buffer-overflow attacks," in *Proceedings of the 7th USENIX Security Conference*, (San Antonio, Texas), pp. 63–78, Jan. 1998.

[80] W. Robertson, Christopher, D. Mutz, and F. Valeur, "Run-time Detection of Heap-based Overflows," in *Proceedings of the 17th USENIX conference on System administration*, (Berkeley, CA, USA), pp. 51–60, USENIX Association, 2003.

[81] P. Argyroudis and D. Glynos, "Protecting the Core: Kernel Exploitation Mitigations." Black Hat Europe 2011, Barcelona, Spain, Mar. 2011.

[82] M. Abadi, M. Budiu, Úlfar Erlingsson, and J. Ligatti, "Control-flow integrity principles, implementations, and applications," *ACM Transactions on Information and System Security (TISSEC)*, vol. 13, pp. 4:1–4:40, Nov. 2009.

[83] F. Mayer, K. MacMillan, and D. Caplan, *SELinux by Example: Using Security Enhanced Linux*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2006.

[84] Z. C. Schreuders, T. McGill, and C. Payne, "Empowering End Users to Confine Their Own Applications: The Results of a Usability Study Comparing SELinux, AppArmor, and FBAC-LSM," *ACM Transactions on Information and System Security (TISSEC)*, vol. 14, pp. 19:1–19:28, Sept. 2011.

[85] A. Karasaridis, B. Rexroad, and D. Hoeflin, "Wide-scale botnet detection and characterization," in *Proceedings of the first conference on First Workshop on Hot Topics in Understanding Botnets*, (Berkeley, CA, USA), pp. 7–7, USENIX Association, 2007.

[86] A. Pasupulati, J. Coit, K. N. Levitt, S. F. Wu, S. H. Li, J. C. Kuo, and K. P. Fan, "Buttercup: on network-based detection of polymorphic buffer overflow vulnerabilities," in *Network Operations and Management Symposium (NOMS 2004)*, vol. 1, pp. 235–248, Apr. 2004.

[87] M. Roesch, "Snort: Lightweight Intrusion Detection for Networks," in *Proceedings of USENIX LISA'99*, (Seattle, Washington, USA), Nov. 1999.

[88] D. Spinellis, "Reliable identification of bounded-length viruses is NP-complete," *IEEE Transactions on Information Theory*, vol. 49, pp. 280–284, Jan. 2003.

[89] E. Filiol, "Metamorphism, formal grammars and undecidable code mutation," *International Journal of Computer Science*, vol. 2, no. 1, pp. 70–75, 2007.

[90] S. Biles, "Detecting the Unknown with Snort and the Statistical Packet Anomaly Detection Engine (SPADE)," tech. rep., Computer Security Online Ltd., 2003. `http://webpages.cs.luc.edu/~pld/courses/447/sum08/class6/biles.spade.pdf`, last accessed May 2012.

[91] J. Gomez, C. Gil, N. Padilla, R. Banos, and C. Jimenez, "Design of a Snort-Based Hybrid Intrusion Detection System," *Lecture Notes in Computer Science*, vol. 5518, pp. 515–522, 2009.

[92] B. Burns, *Security Power Tools*, ch. 10.5 "Alpha2 alphanumeric shellcode encoder", pp. 302–304. O'Reilly, 2007.

[93] J. Mason, S. Small, F. Monrose, and G. MacManus, "English shellcode," in *Proceedings of the 16th ACM conference on Computer and communications security*, CCS '09, (New York, NY, USA), pp. 524–533, ACM, 2009.

[94] M. Polychronakis, K. G. Anagnostakis, and E. P. Markatos, "Network-level Polymorphic Shellcode Detection using Emulation," *Journal in Computer Virology*, vol. 2, no. 4, pp. 257–274, 2007.

[95] G. Portokalidis, A. Slowinska, and H. Bos, "Argos: an emulator for fingerprinting zero-day attacks for advertised honeypots with automatic signature generation," in *Proceedings of the 1st ACM SIGOPS/EuroSys European Conference on Computer Systems (EuroSys '06)*, (New York, NY, USA), pp. 15–27, ACM, 2006.

[96] G. Portokalidis and H. Bos, "Eudaemon: involuntary and on-demand emulation against zero-day exploits," *SIGOPS Operating Systems Review*, vol. 42, no. 4, pp. 287–299, 2008.

[97] R. Lottiaux, P. Gallard, G. Vallee, C. Morin, and B. Boissinot, "OpenMosix, OpenSSI and Kerrighed: a comparative study," in *Fifth IEEE Internation Symposium on Cluster Computing and the Grid (CCGrid'05)*, vol. 2, pp. 1016–1023, May 2005.

[98] "tcpdump/libpcap public repository." `http://www.tcpdump.org`, last accessed May 2012.

[99] F. Fusco and L. Deri, "High speed network traffic analysis with commodity multi-core systems," in *Proceedings of the 10th annual conference on Internet measurement (IMC '10)*, (New York, USA), pp. 218–224, ACM, 2010.

[100] N. Bonelli, A. D. Pietro, S. Giordano, and G. Procissi, "On Multi-gigabit Packet Capturing with Multi-core Commodity Hardware," in *Passive and Active Measurement* (N. Taft and F. Ricciato, eds.), vol. 7192 of *Lecture Notes in Computer Science*, pp. 64–73, Springer Berlin / Heidelberg, 2012.

[101] "libnids library." `http://libnids.sourceforge.net/`, last accessed May 2012.

[102] S. McCanne and V. Jacobson, "The bsd packet filter: a new architecture for user-level packet capture," in *Proceedings of the USENIX Winter 1993 Conference Proceedings on USENIX Winter 1993 Conference Proceedings (USENIX '93)*, (Berkeley, CA, USA), pp. 2–2, USENIX Association, 1993.

[103] L. Deri, "High-Speed Dynamic Packet Filtering," *Journal of Network and Systems Management*, vol. 15, pp. 401–415, 2007.

[104] K. Zaraska, "Prelude IDS: current state and development perspectives," tech. rep., The Prelude IDS project, Mar. 2003.

[105] "RFC 4765 - The Intrusion Detection Message Exchange Format (IDMEF)." `http://www.ietf.org/rfc/rfc4765.txt`, Mar. 2007.

[106] "Metasploit Penetration Testing Framework." `http://www.metasploit.com`, last accessed May 2012.

[107] P. Baecher and M. Koetter, "x86 Shellcode Emulation." `http://libemu.carnivore.it`, last accessed May 2012.

[108] F. Bellard, "QEMU, a fast and portable dynamic translator," in *Proceedings of the annual conference on USENIX Annual Technical Conference (ATEC '05)*, (Berkeley, CA, USA), pp. 41–41, USENIX Association, 2005.

[109] G. Buehrer, B. W. Weide, and P. A. G. Sivilotti, "Using parse tree validation to prevent SQL injection attacks," in *Proc. of 5th international workshop on Software engineering and middleware (SEM '05)*, (New York, NY, USA), pp. 106–113, ACM, 2005.

[110] R. Jones, "netperf homepage." `http://www.netperf.org`, last accessed May 2012.

[111] G. Wicherski, "Placing a low-interaction honeypot in-the-wild: A review of mwcollectd," *Network Security*, vol. 2010, no. 3, pp. 7–8, 2010.

[112] S. V. R. Team, "Razorback homepage." `http://labs.snort.org/razorback`, last accessed May 2012.

[113] D. Glynos, "Context-keyed Payload Encoding: Fighting the Next Generation of IDS," in *Proceedings of Athens IT Security Conference (AthCon 2010)*, 2010.

[114] B. Schwartz, A. W. Jackson, W. T. Strayer, W. Zhou, R. D. Rockwell, and C. Partridge, "Smart packets: applying active networks to network management," *ACM Transactions on Computer Systems*, vol. 18, no. 1, pp. 67–88, 2000.

[115] "European Grid Infrastructure." `http://www.egi.eu`, last accessed June 2011.

[116] "Intel Virtualization Technologies." `http://www.intel.com/technology/virtualization`, last accessed June 2011.

[117] S. Alexander, W. Arbaugh, A. Keromytis, and J. Smith, "Safety and security of programmable network infrastructures," *IEEE Communications Magazine*, vol. 36, pp. 84–92, Oct. 1998.

[118] B. Remick and R. Kessler, "Managing Agent Platforms with AgentSNMP," in *Proc. of First International Workshop on Challenges in Open Agent Systems*, 2002.

[119] A. Litke, D. Skoutas, K. Tserpes, and T. Varvarigou, "Efficient task replication and management for adaptive fault tolerance in mobile grid environments," *Future Generation Computer Systems*, vol. 23, no. 2, pp. 163–178, 2007.

[120] X. Jiang, J. Polastre, and D. Culler, "Perpetual environmentally powered sensor networks," in *Proc. of Fourth International Symposium on Information Processing in Sensor Networks (IPSN 2005)*, pp. 463–468, IEEE Press, Apr. 2005.

[121] G. P. Halkes and K. G. Langendoen, "Comparing energy-saving MAC protocols for wireless sensor networks," *Mobile Networks and Applications*, vol. 10, pp. 783–791, Oct. 2005.

[122] X. Wu, G. Chen, and S. K. Das, "On the Energy Hole Problem of Nonuniform Node Distribution in Wireless Sensor Networks," in *Proc. of 2006 IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS)*, pp. 180–187, IEEE press, Oct. 2006.

[123] H. Gupta, V. Navda, S. Das, and V. Chowdhary, "Efficient gathering of correlated data in sensor networks," *ACM Transactions on Sensor Networks*, vol. 4, no. 1, pp. 1–31, 2008.

[124] H. Cha, S. Choi, I. Jung, H. Kim, H. Shin, J. Yoo, and C. Yoon, "RETOS: Resilient, Expandable, and Threaded Operating System for Wireless Sensor Networks," in *Proc. of 6th International Conference on Information Processing in Sensor Networks*, pp. 148–157, ACM, 2007.

[125] M. Cardei and J. Wu, "Energy Efficient Coverage Problems in Wireless Ad Hoc Sensor Networks," *Computer Communications*, vol. 29, no. 4, pp. 413–420, 2006.

[126] M. Cardei and D.-Z. Du, "Improving wireless sensor network lifetime through power aware organization," *ACM Wireless Networks*, vol. 11, no. 3, pp. 333–340, 2005.

[127] D. Tian and N. D. Georganas, "A coverage-preserving node scheduling scheme for large wireless sensor networks," in *Proc. of the 1st ACM international workshop on Wireless sensor networks and applications (WSNA '02)*, pp. 32–41, ACM Press, 2002.

[128] F. Ye, G. Zhong, S. Lu, and L. Zhang, "PEAS: A Robust Energy Conserving Protocol for Long-lived Sensor Networks," in *Proc. of the 10th IEEE International Conference on Network Protocols*, pp. 200–201, IEEE Computer Society, Sept. 2002.

[129] S. Slijepcevic and M. Potkonjak, "Power efficient organization of wireless sensor networks," in *Proc. of International Conference on Communications (ICC'01)*, pp. 472–476, IEEE press, June 2001.

[130] M. Cardei, D. MacCallum, M. X. Cheng, M. Min, X. Jia, D. Li, and D.-Z. Du, "Wireless sensor networks with energy efficient organization," *Journal of Interconnection Networks*, vol. 3, no. 3-4, pp. 213–229, 2002.

[131] M. T. Thai, F. Wang, H. Du, and X. Jia, "Coverage problems in wireless sensor networks: Designs and analysis," *International Journal of Sensor Networks, Special issue on coverage problems*, vol. 3, pp. 191–200, 2008.

[132] M. R. Garey and D. S. Johnson, *Computers and Intractability. A Guide to the Theory of NP-Completeness*. New York: Freeman, 1979.

[133] Z. Abrams, A. Goel, and S. Plotkin, "Set k-cover algorithms for energy efficient monitoring in wireless sensor networks," in *Proc. of Third International Symposium on Information Processing in Sensor Networks*, pp. 424–432, ACM, 2004.

[134] M. Cardei, M. Thai, Y. Li, and W. Wu, "Energy-efficient target coverage in wireless sensor networks," in *Proc. of INFOCOM 05*, vol. 3, pp. 1976–1984, IEEE, Mar. 2005.

[135] P. Berman, G. Calinescu, C. Shah, and A. Zelikovsky, "Power efficient monitoring management in sensor networks," in *Proc. of Wireless Communications and Networking Conference*, vol. 4, pp. 2329–2334, IEEE, Mar. 2004.

[136] N. Garg and J. Könemann, "Faster and simpler algorithms for multicommodity flow and other fractional packing problems," in *Proc. of 39th Annual IEEE Symposium on Foundations of Computer Science*, pp. 300–309, IEEE, Nov. 1998.

[137] D. Zorbas, D. Glynos, P. Kotzanikolaou, and C. Douligeris, "B{GOP}: An Adaptive Algorithm for Coverage Problems in Wireless Sensor Networks," in *Online Proc. of the 13th European Wireless Conference (EW2007)*, Apr. 2007.

[138] D. Zorbas, D. Glynos, P. Kotzanikolaou, and C. Douligeris, "Solving coverage problems in wireless sensor networks using cover sets," *Ad Hoc Networks*, vol. 8, no. 4, pp. 400–415, 2010.

[139] D. Zorbas, D. Glynos, and C. Douligeris, "Connected partial target coverage and network lifetime in wireless sensor networks," in *Proceedings of the 2nd IFIP conference Wireless Days*, WD'09, (Piscataway, NJ, USA), pp. 150–154, IEEE Press, 2009.

[140] D. S. Alexander, W. A. Arbaugh, A. D. Keromytis, and J. M. Smith, *Security in Active Networks*, vol. 1603 of *Lecture Notes in Computer Science*, pp. 433–451. Springer–Verlag, 1999.

[141] W. Jansen and T. Karygiannis, "Mobile agents and security," *NIST Special Publication 800–19*, Sept. 1999.

[142] C. Lin, V. Varadharajan, Y. Wang, and V. Pruthi, "Enhancing grid security with trust management," pp. 303–310, Sept. 2004.

[143] N. Karnik, *Security in Mobile Agent Systems*. PhD thesis, Department of Computer Science, University of Minnesota, Oct. 1998.

[144] J. Feigenbaum and P. Lee, "Trust management and proof-carrying code in secure mobile-code applications," in *Proc. of DARPA Workshop on Foundations for Secure Mobile Code*, (Monterey, CA, USA), Mar. 1997. Position paper.

[145] I. Biehl, B. Meyer, and S. Wetzel, *Ensuring the Integrity of Agent-Based Computations by Short Proofs*, vol. 1477 of *Lecture Notes in Computer Science*, pp. 183–194. Springer-Verlag, 1998.

[146] R. Harper, F. Honsell, and G. Plotkin, "A framework for defining logics," *Journal of the Association for Computing Machinery*, vol. 80, pp. 143–184, Jan. 1993.

[147] "Java Security Architecture." `http://java.sun.com/javase/6/docs/technotes/guides/security/spec/security-spec.doc.html`, last accessed July 2009.

[148] P. Menage, *RCANE: A resource controlled framework for active network services*, vol. 1653 of *Lecture Notes in Computer Science*. Springer–Verlag, June 1999.

[149] G. Vigna, "Protecting mobile agents through tracing," in *Proc. of the 3rd ECOOP Workshop on Mobile Object Systems*, June 1997.

[150] U. Wilhelm, L. ButtyΓ n, and S. Staamann, "On the Problem of Trust in Mobile Agent Systems," in *Symposium on Network and Distributed System Security*, Internet Society, 1998.

[151] T. Sander and C. F. Tschudin, *Protecting Mobile Agents Against Malicious Hosts*, vol. 1419 of *Lecture Notes in Computer Science*, pp. 44–60. London, UK: Springer–Verlag, 1998.

[152] P. Kotzanikolaou, M. Burmester, and V. Chrissikopoulos, *Secure Transactions with Mobile Agents in Hostile Environments*, vol. 1841 of *Lecture Notes in Computer Science*, pp. 289–297. Springer–Verlag, 2000.

[153] F. Hohl, *Time Limited Blackbox Security: Protecting Mobile Agents From Malicious Hosts*, vol. 1419 of *Lecture Notes in Computer Science*, pp. 92–113. Springer–Verlag, 1998.

[154] N. S. Larry, L. Peterson, A. Bavier, Y. Gottlieb, S. Karlin, A. Nakao, X. Qie, T. Spalink, and M. Wawrzoniak, "Extensible routers for active networks," in *Proc. of DARPA Active Networks Conference and Exposition (DANCE'02)*, pp. 92–116, IEEE Computer Society, 2002.

[155] K. G. Anagnostakis, M. B. Greenwald, S. Ioannidis, D. Li, and J. M. Smith, "Flexible network monitoring with FLAME," *Computer Networks*, vol. 50, no. 14, pp. 2548–2563, 2006.

[156] H. Bos and B. Samwel, "Safe kernel programming in the OKE," in *Open Architectures and Network Programming Proceedings*, pp. 141–152, IEEE, 2002.

[157] V. Ramachandran, R. Pandey, and S.-H. Chan, "Fair resource allocation in active networks," in *Proc. of Ninth International Conference on Computer Communications and Networks*, pp. 468–475, IEEE, 2000.

[158] W. Eaves, L. Cheng, A. Galis, T. Becker, T. Suzuki, S. Denazis, and C. Kitahara, "SNAP based resource control for active networks," in *Proc. of Global Telecommunications Conference 2002 (GLOBECOM '02)*, vol. 3, pp. 2098–2102, IEEE, Nov. 2002.

[159] W. A. Arbaugh, D. J. Farber, and J. M. Smith, "A secure and reliable bootstrap architecture," in *Proc. of 1997 IEEE Symposium on Security and Privacy*, pp. 65–71, May 1997.

[160] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 6, no. 22, 1976.

[161] National Institute of Standards, "Digital Signature Standard," Tech. Rep. FIPS-186, U.S. Department of Commerce, May 1994.

[162] R. Mavropodi, P. Kotzanikolaou, and C. Douligeris, "SecMR - a secure multipath routing protocol for ad hoc networks," *Ad Hoc Networks*, vol. 5, no. 1, pp. 87–99, 2007.

[163] D. Glynos, P. Kotzanikolaou, and C. Douligeris, "Preventing Impersonation Attacks in MANET with Multi-Factor Authentication," in *Proc. of 3rd International Symposium on Modeling and Optimization in Mobile, Ad-Hoc and Wireless Networks (WiOpt 2005)*, pp. 59–64, IEEE, 2005.

[164] T. Jim, G. Morrisett, D. Grossman, M. Hicks, J. Cheney, and Y. Wang, "Cyclone: A Safe Dialect of C," in *Proc. of 2002 USENIX Annual Technical Conference*, pp. 275–288, USENIX Association, June 2002.

[165] L. Sarmenta, "Sabotage-tolerance mechanisms for volunteer computing systems," in *Proc. of First IEEE/ACM International Symposium on Cluster Computing and the Grid, 2001.*, pp. 337–346, 2001.

[166] S. Marsh, *Formalising Trust as a Computational Concept*. PhD thesis, Department of Mathematics and Computer Science, University of Stirling, 1994.

[167] F. Azzedin and M. Maheswaran, "Evolving and managing trust in grid computing systems," in *Proc. of Canadian Conference on Electrical and Computer Engineering (IEEE CCECE 2002)*, vol. 3, pp. 1424–1429, IEEE, 2002.

[168] D. Glynos, P. Argyroudis, C. Douligeris, and D. O'Mahony, "Twohop: Metric-based Trust Evaluation for Peer-to-Peer Collaboration Environments," in *Proceedings of 51st Annual IEEE Global Telecommunications Conference (Globecom'08)*, (New Orleans, LA, USA), p. 6, IEEE Xplore, Nov. 30 - Dec. 4 2008.

[169] eBay Inc., "Online Auction and Shopping Website." `http://www.ebay.com/`, 2009.

[170] R. Chen and W. Yeager, "Poblano: a Distributed Trust Model for Peer-to-Peer Networks," tech. rep., Sun Microsystems, Inc., 2003.

[171] D. Kugler, "An Analysis of GNUnet and the Implications for Anonymous, Censorship-Resistant Networks," in *Proceedings of 3rd Workshop on Privacy Enhancing Technologies (PET'03)*, pp. 161–176, 2003.

[172] R. Levien and A. Aiken, "Attack-resistant Trust Metrics for Public Key Certification," in *Proceedings of 7th USENIX Security Symposium*, pp. 229–242, 1998.

[173] P. Crowley, "TrustFlow." `http://trustflow.lshift.net/`, last accessed Dec. 2009.

[174] F. Labalme and K. Burton, "Enhancing the Internet with Reputations: an OpenPrivacy White Paper." `http://www.openprivacy.org/papers/200103-white.html`, 2001.

[175] R. Dingledine, M. Freedman, and D. Molnar, "The Free Haven Project: Distributed Anonymous Storage Service," in *Proceedings of 2000 Workshop on Design Issues in Anonymity and Unobservability*, pp. 67–95, 2000.

[176] F. Cornelli, E. Damiani, S. Vimercati, S. Paraboschi, and P. Samarati, "Choosing Reputable Servents in a P2P Network," in *Proceedings of 11th International World Wide Web Conference (W3C'02)*, (Honolulu, Hawaii), May 2002.

[177] E. Damiani, S. Vimercati, S. Paraboschi, and P. Samarati, "Managing and Sharing Servents' Reputations in P2P Systems," *IEEE Transactions on Knowledge and Data Engineering*, vol. 15, no. 4, pp. 840–854, 2003.

[178] S. Kamvar, M. Schlosser, and H. Garcia-Molina, "EigenTrust Algorithm for Reputation Management in P2P Networks," in *Proceedings of 12th International World Wide Web Conference (WWW'03)*, (Budapest, Hungary), pp. 640–651, 2003.

[179] T. Jiang and J. Baras, "Trust Evaluation in Anarchy: A Case Study on Autonomous Networks," in *Proceedings of 25th IEEE International Conference on Computer Communications (Infocom'06)*, (Barcelona, Spain), pp. 1–12, 2006.

[180] M. Raya, P. Papadimitratos, V. Gligor, and J.-P. Hubaux, "On Data–Centric Trust Establishment in Ephemeral Ad hoc Networks," in *Proceedings of 27th IEEE International Conference on Computer Communications (Infocom'08)*, (Phoenix, AZ, USA), pp. 1238–1246, 2008.

[181] G. Shafer, *A Mathematical Theory of Evidence*. Princeton, New Jersey, USA: Princeton University Press, 1976.

[182] R. Prasad, V. Srinivas, V. Kumari, and K. Raju, "An Effective Calculation of Reputation in P2P Networks," *Journal of Networks*, vol. 4, no. 5, pp. 332–342, 2009.

[183] L. Xiong and L. Liu, "PeerTrust: Supporting reputation based trust of peer-to-peer electronic communities," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, pp. 843–857, July 2004.

[184] G. Theodorakopoulos and J. Baras, "On Trust Models and Trust Evaluation Metrics for Ad hoc Networks," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 2, pp. 318–328, 2006.

[185] G. Theodorakopoulos and J. Baras, "A Testbed for Comparing Trust Computation Algorithms," in *Proceedings of 25th Army Science Conference (ASC'06)*, (Orlando, FL, USA), Nov. 2006.

[186] A. Abdul-Rahman and S. Hailes, "Supporting Trust in Virtual Communities," in *Proceedings of the 33rd Hawaii International Conference on System Sciences*, vol. 6 of *HICSS '00*, (Washington, DC, USA), pp. 6007–, IEEE Computer Society, 2000.

[187] B. Christianson and W. Harbison, "Why isn't trust transitive?," in *Security Protocols*, vol. 1189 of *Lecture Notes in Computer Science*, pp. 171–176, Springer Berlin / Heidelberg, 1997.

[188] M. G. Uddin, M. Zulkernine, and S. I. Ahamed, "CAT: a context-aware trust model for open and dynamic systems," in *SAC '08: Proceedings of the 2008 ACM symposium on Applied computing*, (New York, NY, USA), pp. 2024–2029, ACM, 2008.

[189] R. Hill and R. Dunbar, "Social Network Size in Humans," *Human Nature*, vol. 14, no. 1, pp. 53–72, 2003.

[190] D. Glynos, P. Argyroudis, and C. Douligeris, "Collaborative service evaluation with the TwoHop trust framework," *Security and Communication Networks*, vol. 5, pp. 594–613, June 2012.

[191] I. Stoica, R. Morris, D. Karger, F. Kaashoek, and H. Balakrishnan, "Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications," in *Proceedings of 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, pp. 149–160, 2001.

[192] B. Zhao, L. Huang, J. Stribling, S. Rhea, A. Joseph, and J. Kubiatowicz, "Tapestry: a Resilient Global-scale Overlay for Service Deployment," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 1, pp. 41–53, 2004.

[193] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Schenker, "A scalable content-addressable network," in *SIGCOMM '01: Proceedings of the 2001 conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, (New York, NY, USA), pp. 161–172, ACM Press, 2001.

[194] F. Dabek, F. Kaashoek, D. Karger, R. Morris, and I. Stoica, "Wide-area cooperative storage with CFS," *SIGOPS Operating Systems Review*, vol. 35, no. 5, pp. 202–215, 2001.

[195] J. Wang, "A survey of web caching schemes for the Internet," *ACM SIGCOMM Computer Communication Review*, vol. 29, no. 5, pp. 36–46, 1999.

[196] T. Jiang, , and J. Baras, "Trust Credential Distribution in Autonomic Networks," in *Proceedings of 51st Annual IEEE Global Telecommunications Conference (Globecom'08)*, (New Orleans, LA, USA), IEEE Xplore, Nov. 30 - Dec. 4 2008.

[197] K. Hoffman, D. Zage, and C. Nita-Rotaru, "A survey of attack and defense techniques for reputation systems," *ACM Computing Surveys*, vol. 42, pp. 1:1–1:31, Dec. 2009.

[198] Y. Sun, Z. Han, and K. Liu, "Defense of trust management vulnerabilities in distributed networks," *IEEE Communications Magazine*, vol. 46, pp. 112–119, Feb. 2008.

[199] J. Douceur, "The Sybil Attack," in *Proc. of 1st International Workshop on Peer-to-Peer Systems (IPTPS'02)*, pp. 251–260, 2002.

[200] A. Back, "The hashcash proof-of-work function." `http://www.hashcash.org/papers/draft-hashcash.txt`, June 2003.

[201] D. Ingram, *An Evidence Based Architecture for Efficient, Attack-Resistant Computational Trust Dissemination in Peer-to-Peer Networks*, vol. 3477 of *Lecture Notes in Computer Science*, pp. 377–386. Springer Berlin / Heidelberg, 2005.

[202] D. Shaw and W. Kinsner, "Multifractal modelling of radio transmitter transients for classification," in *Proceedings of Conference on Communications, Power and Computing*, (Winnipeg, Manitoba, Canada), pp. 306–312, IEEE, May 1997.

[203] O. Ureten and N. Serinken, "Bayesian detection of radio transmitter turn-on transients," in *Proceedings of NSIP99*, pp. 830–834, 1999.

[204] J. Hall, M. Barbeau, and E. Kranakis, "Detection of transient in radio frequency fingerprinting using phase characteristics of signals," in *Proceedings of 3rd IASTED International Conference on Wireless and Optical Communications (WOC 2003)*, (Alberta, Canada), pp. 13–18, ACTA Press, July 2003.

[205] J. Kleider, S. Gifford, S. Chuprun, and B. Fette, "Radio frequency watermarking for OFDM wireless networks," in *Proceedings of Acoustics, Speech, and Signal Processing (ICASSP'04)*, vol. 5, pp. 397–400, IEEE, May 2004.

[206] B. Sieka, "Active Fingerprinting of 802.11 Devices by Timing Analysis," in *Proceedings of 3rd Consumer Communications and Networking Conference (CCNC 2006)*, vol. 1, pp. 15–19, IEEE, 2006.

[207] J.-H. Song, V. W. Wong, and V. C. Leung, "A framework of secure location service for position-based ad hoc routing," in *Proceedings of 1st International Conference on Mobile Computing and Networking archive*, (Venezia, Italy), pp. 99–106, ACM, 2004.

[208] "p0f: passive OS fingerprinting tool." `http://lcamtuf.coredump.cx/p0f.shtml`, last accessed July 2009.

[209] "DACOS: active OS fingerprinting library." `http://rainbow.cs.unipi.gr/projects/dacos`, last accessed July 2009.

[210] "nmap: Network Mapper." `http://www.insecure.org/nmap`, last accessed July 2009.

[211] N. Koblitz, "Elliptic curve cryptosystems," *Mathematics of Computation*, vol. 48, pp. 203–209, July 1997.

[212] L. Zhou and Z. Haas, "Securing Ad hoc Networks," *IEEE Network*, vol. 6, no. 13, pp. 24–30, 1999.

[213] F. V. Jensen, *Bayesian Networks and Decision Graphs*. Information Science and Statistics, Springer, 2001.

[214] J. Hubaux, L. Buttyan, and S. Capkun, "The quest for security in mobile ad hoc networks," in *Proceedings of the 2nd MobiHoc Conference*, (BA, Massachusetts), Aug. 2001.

[215] R. Wishart, J. Indulska, M. Portmann, and P. Sutton, *Context-Enhanced Authentication for Infrastructureless Network Environments*, vol. 4159 of *Lecture Notes in Computer Science*, pp. 924–935. Springer, 2006.

[216] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: enabling innovation in campus networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, pp. 69–74, Apr. 2008.

[217] H. Soliman, C. Castelluccia, K. ElMalki, and L. Bellier, "RFC 5380 - Hierarchical Mobile IPv6 (HMIPv6) Mobility Management." `http://www.ietf.org/rfc/rfc5380.txt`, Oct. 2008.

[218] J. Kong, B. Rezaei, N. Sarshar, V. Roychowdhury, and P. Boykin, "Collaborative Spam Filtering Using E-Mail Networks," *Computer*, vol. 39, pp. 67–73, Aug. 2006.

[219] K. Z. Snow, S. Krishnan, F. Monrose, and N. Provos, "ShellOS: enabling fast detection and forensic analysis of code injection attacks," in *Proceedings of the 20th USENIX Conference on Security (SEC'11)*, (Berkeley, CA, USA), pp. 9–9, USENIX Association, 2011.

Ευρετήριο