ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

Μέθοδοι εύρεσης πληροφοριών σε ασφαλή περιβάλλοντα δικτύων μεγάλης κλίμακας

Αγάπιος Αβραμίδης

Διδακτορική Διατριβή

Τμήμα Πληροφορικής

Νοέμβριος 2009

Αγάπιος Αβραμίδης

Τμήμα Πληροφορικής,Πανεπιστήμιο Πειραιώς

Η παρούσα διδακτορική διατριβή εντάσσεται στο υποέργο Προηγμένα Συστήματα Ασφάλειας και Αντιμετώπισης Επιθέσεων κωδικός έργου 03ΕΔ546 Το έργο συγχρηματοδοτείται:

- 80% της Δημόσιας Δαπάνης από την Ευρωπαϊκή Ένωση – Ευρωπαϊκό Κοινωνικό Ταμείο
- 20% της Δημόσιας Δαπάνης από το Ελληνικό Δημόσιο – Υπουργείο Ανάπτυξης – Γενική Γραμματεία Έρευνας και Τεχνολογίας
- και από τον Ιδιωτικό Τομέα

στο πλαίσιο του Μέτρου 8.3 του Ε.Π. Ανταγωνιστικότητα – Γ΄ Κοινοτικό Πλαίσιο Στήριξης.

ΔΕΣΜΕΥΤΙΚΗ ΔΗΛΩΣΗ

Οι διδακτορικές μου σπουδές διεξήχθησαν υπό την εποπτεία του Καθηγητή Χρήστου Δουληγέρη, μεταξύ Φεβρουαρίου 2003 και Νοεμβρίου 2009 στο τμήμα Πληροφορικής του Πανεπιστημίου Πειραιώς.

Η παρούσα διδακτορική διατριβή είναι το αποτέλεσμα πρωτότυπης ερευνητικής εργασίας που διεξήχθει από εμένα σε συνεργασία με άλλους, ενώ ήμουν υποψήφιος διδάκτωρ του Τμήματος Πληροφορικής στο Πανεπιστήμιο Πειραιώς και δεν έχει κατατεθεί για τίτλο σπουδών σε κανένα άλλο Πανεπιστήμιο ή εκπαιδευτικό ίδρυμα.

Αγάπιος Αβραμίδης

Νοέμβριος 2009

Αφιέρωση

Στην οικογένεια μου για την αμέριστη συμπαράστασή της.

Ευχαριστίες

Κατά τη διάρκεια εκπόνησης της διδακτορικής διατριβής μου δόθηκε η ευκαιρία να συναναστραφώ και να συνεργαστώ με ένα πλήθος ανθρώπων. Θα ήθελα να εκφράσω τις ειλικρινείς ευχαριστίες μου προς αυτούς.

Ιδιαιτέρως θα ήθελα να ευχαριστήσω τον επιβλέποντα Καθηγητή Χρήστο Δουληγέρη. Η συνεχής υποστήριξη, η υπομονή του καθώς και οι εύστοχες παρατηρήσεις του όποτε χρειαζόντουσαν έδρασαν καταλυτικά και με βοήθησαν στην ολοκλήρωση της διατριβής. Επίσης, θα ήθελα να ευχαριστήσω όλα τα μέλη της επταμελούς εξεταστικής επιτροπής.

Επίσης, θα ήθελα να ευχαριστήσω την Γενική Γραμματεία Έρευνας Τεχνολογίας (ΓΓΕΤ) για την χρηματοδότηση που μου προσέφερε καθόλη την διάρκεια εκπόνησης της διατριβής (στο πλαίσιο του έργου Επιχειρησιακού Προγράμματος Ενίσχυσης Ερευνητικού Δυναμικού (ΠΕΝΕΔ) 2003 03ΕΔ/546-ΠΕΝΕΔ 2003, Αντιμετώπιση Επιθέσεων Άρνησης Υπηρεσίας και Νέες Τεχνολογίες Πλέγματος).

Τέλος, θα ήθελα να ευχαριστήσω όλους τους συναδέλφους από το εργαστήριο Τηλεπικοινωνιακών Εφαρμογών (εργαστήριο 208), για την βοήθεια, την συμπαράσταση και για τις ευχάριστες ώρες που περάσαμε μαζί. Φίλοι και συνάδελφοι σας ευχαριστώ.

ΠΕΡΙΛΗΨΗ

ΜΕΘΟΔΟΙ ΕΥΡΕΣΗΣ ΠΛΗΡΟΦΟΡΙΩΝ ΣΕ ΑΣΦΑΛΗ ΠΕΡΙΒΑΛΟΝΤΑ ΔΙΚΤΥΩΝ ΜΕΓΑΛΗΣ ΚΛΙΜΑΚΑΣ

Αγάπιος Αβραμίδης

Στη διατριβή αυτή εξετάζεται το πρόβλημα της εύρεσης πληροφοριών σε δίκτυα ομότιμων κόμβων. Η προσέγγιση που ακολουθείται δραστηριοποιείται σε δύο κύριους άξονες. Στον έναν άξονα μελετώνται οι αδυναμίες και οι ελλείψεις που υπάρχουν στις διαδικασίες εύρεσης σε δίκτυα ομότιμων κόμβων και προτείνονται λύσεις. Στον άλλο άξονα εξετάζεται το πρόβλημα της ασφάλειας σε δίκτυα ομότιμων κόμβων με στόχο την βελτίωση των υποδομών των δικτύων ομότιμων κόμβων. Μέσω συνεισφορών και στους δύο άξονες παρέχονται τεχνικές που απαιτούνται από αξιόπιστους μηχανισμούς εύρεσης.

Αρχικά περιγράφονται τα βασικά χαρακτηριστικά των δικτύων ομότιμων κόμβων, ταξινομούνται τα πρωτόκολλα βάσει των οποίων οργανώνονται οι κόμβοι και οι πληροφορίες σε αυτά, και στην συνέχεια εστιάζουμε στα δίκτυα δομημένης επικάλυψης αναλύοντας τις ιδιότητές τους αλλά και τους βασικούς τους εκπρόσωπους.

Στην συνέχεια ορίζονται οι διαδικασίες εύρεσης που πρέπει να υποστηρίζονται από τα συστήματα των δικτύων δομημένης επικάλυψης. Αναλύονται τα ερωτήματα πολλαπλών χαρακτηριστικών και τα ερωτήματα εύρους, και περιγράφονται οι αδυναμίες των δικτύων δομημένης επικάλυψης ως προς την εφαρμογή των παραπάνω ερωτημάτων και προτείνονται μεθοδολογίες μέσω των οποίων επέρχεται τελικά η υποστήριξη των ερωτημάτων. Δίνεται ιδιαίτερη βάση στο πρόβλημα των ερωτημάτων εύρους, για το οποίο προτείνεται λύση που βασίζεται στην κατανομή ενός δυαδικού δένδρου αναζήτησης στους κόμβους του δικτύου επικάλυψης.

Επίσης, προτείνεται ένας μηχανισμός ασφάλειας για δίκτυα δομημένης επικάλυψης μέσω του οποίου οι κόμβοι του συστήματος πιστοποιούνται από άλλους κόμβους του δικτύου επικάλυψης. Με αυτόν τον τρόπο δημιουργούνται σχέσεις εμπιστοσύνης μεταξύ των κόμβων και παρέχεται ένα υπόστρωμα στο οποίο μπορεί να βασιστούν επιπρόσθετοι μηχανισμοί ασφάλειας για τις εφαρμογές στα δίκτυα επικάλυψης.

Στην συνέχεια μελετάται μια εναλλακτική κατανεμημένη τεχνολογία, τα συστήματα υπολογιστικού πλέγματος. Με βάση τα πρότυπα που ορίζονται στην ανοιχτή αρχιτεκτονική OGSA για υπηρεσίες πλέγματος ορίζονται οι βασικές λειτουργίες εύρεσης και ανάκτησης εγγράφων για την ψηφιακή βιβλιοθήκη LU-Grid. Τα πρότυπα, οι μηχανισμοί και οι βιβλιοθήκες που παρέχονται είναι επαρκή για την δημιουργία κατανεμημένων εφαρμογών στο πλαίσιο των οποίων διαμοιράζονται πόροι.

ABSTRACT

INFORMATION DISCOVERY IN LARGE SCALE NETWORKS

Agapios Avramidis

This dissertation addresses the problem of searching for data organized in a large, highly transient network of nodes. We propose techniques that conform to the requirements for scalability of peer-to-peer (P2P) networks and contribute to the searchability and the security of state of the art P2P architectures, such as structured overlay networks (SON). This is accomplished by extending the search operation of SON to support range queries and by proposing a security infrastructure for SON implemented on top of the peers of the overlay network.

In particular, we study current P2P systems and related technologies in terms of their search and routing mechanisms. We provide a brief history of the evolution of P2P systems that lead to the current generation of P2P systems that meet the requirements of decentralization, reliability, scalability and describe the most representative protocols of SON.

We describe the requirements for searching in SON. In particular, we argue that the queries must support multiple attributes over a range of data values. We analyze the weaknesses of SON that prevent the direct implementation of the aforementioned search requirements and briefly survey some solutions. Then, we focus on the range query problem and provide a solution by proposing a replication scheme for the keys of the data items being stored in the system. All the key replicas are organized into a binary tree structure, named Replication Tree (RT). The novelty of RT over similar mechanisms lies on the fact that RT does not enforce a limit on the number of key replicas per tree node. Instead, RT provides the replica management policy and let the peer nodes to enforce it whenever is necessary. Thus, RT efficiently exploits the capabilities of the peer nodes.

Furthermore, we survey the security issues in structured overlay networks and provide a distributed Public Key Infrastructure (PKI) infrastructure which is built upon the Chord overlay network, in order to provide security services for the peer nodes. The solution distributes the functionality of a PKI across the peer nodes, by using threshold cryptography and proactive updating. The security of the proposed infrastructure is evaluated through extensive simulation experiments. Also, we measure the performance of the system using several scenarios of untrusted nodes and threshold sets, that illustrate that realistic implementation can be conducted for various distributions of untrusted nodes.

Finally, we explore the potential of another distributed technology, Grid Computing. Despite their differences, both P2P and Grid Computing have similar goal, the sharing of computing resources. By designing a digital library according to the service-oriented architecture, which is based on the OGSA specifications, we explore and evaluate the capabilities of grid technologies and investigate common problems both in P2P and Grid.

Περιεχόμενα

Κατάλογος σχημάτων

Κατάλογος πινάκων

Συντομεύσεις

DHT	**D**istributed **H**ash **T**able
RA	**R**egistration **A**gency
GIIS	**G**rid **I**ndex **I**nformation **S**ervers
GIIS	**G**rid **I**nformation **S**ervers
OGSA	**O**pen **G**rid **S**ervices **A**rchitecture
P2P	**P**eer-to-**P**eer
ΔΔΕ	**Δίκτυα Δ**ομημένης **Ε**πικάλυψης
ΥΔΚ	**Υποδομής Δ**ημοσίου **Κ**λειδιού
ΚΚ	**Κ**ρυπτογραφία **Κ**ατωφλιού
ΚΠ	**Κ**ρυπτογραφία **Π**ρόληψης

Γλωσσάρι

Adversary Model	Πλάνο επίθεσης
Cache	(τεχνική) προσωρινή(ς) αποθήκευση(ς)
Eclipse Attack	Επίθεση Έκλειψης
Redundancy	Πλεονασμός
Replication Techniques	Τεχνικές αντιγραφής
Overlay	Επικάλυψη
Structured Overlay Network	Δίκτυο Δομημένης Επικάλυψης
Hashtable	Πίνακας Κατακερματισμού
Distributed Hashtable	Κατανεμημένος Πίνακας Κατακερματισμού
Internet Address	Διεύθυνση Διαδικτύου
Hash function	Συνάρτηση Κατακερματισμού (τυχαία συνάρτηση)
Port (in tcp/ip)	Θύρα
Computational puzzles	Υπολογιστικοί γρίφοι
Network Proximity	Δικτυακή Εγγύτητα Γειτονία, Γειτνίαση
Iterative Routing	Επαναληπτική Δρομολόγηση
Offline Certification Authority	(εξωτερική) Αρχή Πιστοποίησης
Self-certifying data	Πιστοποίηση επί των δεδομένων
Non-repudiation	Μη αποποίηση
Reputation mechanism	Μηχανισμοί Φήμης
Proactive security	Προληπτική(Προνοητική) Ασφάλεια
Load Balance	Εξισορόπιση Φόρτου
Theshold Cryptography	Κρυπτογραφία Κατωφλιού
Proactive Cryptography	Κρυπογραφία Πρόληψης
Adversary	Αντίπαλος
Peer-to-Peer	Ομότιμα Δίκτυα - Ομότιμα Συστήματα

Grid Computing System	Σύστημα Υπολογιστικού Πλέγματος
Cluster	Συστοιχία υπολογιστών
Bandwidth	Εύρος ζώνης
Replication factor	Βαθμός αντιγραφής
Peer-to-Peer network	Δίκτυο ομότιμων κόμβων
Content Delivery Networks	Δίκτυα παράδοσης περιεχομένου
Consistent Hashing	Αποδοτικός κατακερματισμός
Web Service	Υπηρεσία ιστού
Registry	Μητρώο

Κεφάλαιο 1

Εισαγωγή

Περίληψη

Στο εισαγωγικό αυτό κεφάλαιο αναφέρονται και αναλύονται οι προκλήσεις που παρουσιάζονται κατά την εύρεση πληροφοριών στα σύγχρονα υπολογιστικά περιβάλλοντα. Παράλληλα αναδεικνύεται το πρόβλημα της εύρεσης πληροφοριών σε περιβάλλοντα δικτύων μεγάλης κλίμακας, στα οποία ο όγκος των δεδομένων είναι τεράστιος και βρίσκεται κατανεμημένος σε προσωρινούς υπολογιστικούς πόρους που ανήκουν σε διαφορετικές αρχές διαχείρισης. Με βάση τις διαπιστώσεις αυτές, ορίζονται οι στόχοι της παρούσας διατριβής που είναι η δημιουργία τεχνικών μέσω των οποίων μπορεί να κατασκευαστεί ένας μηχανισμός εύρεσης πληροφοριών, και αναλύεται η συνεισφορά της διατριβής.

1.1 Εύρεση πληροφοριών σε συστήματα μεγάλης κλίμακας

Στην σημερινή εποχή οι άνθρωποι έχουν άμεση πρόσβαση, με χαμηλό κόστος, σε ισχυρά υπολογιστικά συστήματα, μέσω των οποίων επικοινωνούν άμεσα με μεγάλη ταχύτητα. Παράλληλα, μέσω του διαδικτύου παρέχεται τεράστιος όγκος πληροφορίας προς *κατανάλωση* και επεξεργασία. Επίσης, παρέχεται η δυνατότητα σε κάθε χρήστη του διαδικτύου, που βρίσκεται σε διαφορετικά γεωγραφικά μέρη, να παράγει πληροφορίες, οι οποίες διατίθενται προς κάθε ενδιαφερόμενο. Ο όγκος της πληροφορίας που βρίσκεται στο διαδίκτυο είναι τεράστιος και αυξάνεται με γρήγορους ρυθμούς. Η εύρεση της κατάλληλης πληροφορίας όταν ο συνολικός όγκος των ψηφιακών δεδομένων είναι τεράστιος είναι μια δύσκολη και απαιτητική διαδικασία.

Ο τρόπος με τον οποίο ανακαλύπτονται πληροφορίες στο διαδίκτυο σήμερα γίνεται με την βοήθεια των *μηχανών αναζήτησης* [1, 2, 3]. Οι μηχανές αναζήτησης έχουν ως στόχο την γρήγορη εύρεση ενός συνόλου πληροφοριών που ικανοποιεί σε μεγάλο βαθμό τα κριτήρια αναζήτησης που θέτει ο χρήστης. Αυτός ο τρόπος αναζήτησης της πληροφορίας δημιουργεί *εξαρτήσεις* των χρηστών του διαδικτύου γύρω από μια κεντρική οντότητα, την εταιρεία που διαχειρίζεται την μηχανή αναζήτησης. Παρά το γεγονός ότι οι μηχανές αναζήτησης χρησιμοποιούν τεχνικές με τις οποίες κατανέμουν τα ευρετήρια σε πολλαπλές συστοιχίες υπολογιστικών συστημάτων, οπότε ουσιαστικά δεν υπάρχει κάποιο κεντρικό σημείο αποτυχίας του συστήματος, οι πληροφορίες στο διαδίκτυο, οι προτιμήσεις των χρηστών για αυτές είναι προσβάσιμες και διαχειρίσιμες από μια μόνον οντότητα, την συγκεκριμένη εταιρεία. Το γεγονός αυτό μπορεί να μην είναι επιθυμητό σε όλες τις περιπτώσεις.

Παράλληλα, την τελευταία δεκαετία ξεκίνησε η εξάπλωση των προγραμμάτων διαμοίρασης αρχείων [4, 5, 6], κυρίως μουσικού-ψυχαγωγικού περιεχομένου, τα οποία βασίζονται σε τεχνολογίες *δικτύων ομότιμων κόμβων* (peer-to-peer networks). Ο μεγάλος αριθμός των χρηστών τους αναζητά και ανακτά πληροφορίες και αρχεία βασισμένος σε δικά του υπολογιστικά συστήματα. Οι ιδιαιτερότητες στην συμπεριφορά των χρηστών (π.χ. εγωιστική συμπεριφορά), η μειωμένη αξιοπιστία των επιμέρους υπολογιστικών συστημάτων και των συνδέσεων των χρηστών, και η

έλλειψη κεντρικού ελέγχου ωθούν την έρευνα και την τεχνολογία των δικτύων ομότιμων κόμβων στην αντιμετώπιση αυτών των προβλημάτων και στην δημιουργία υποδομών μέσω των οποίων οι χρήστες του διαδικτύου έχουν τη δυνατότητα να διαμοιράζονται περιεχόμενο (π.χ. μουσικά αρχεία) και υπολογιστικούς πόρους (π.χ. αποθηκευτικό χώρο, χρόνο επεξεργασίας).

Παρόμοιες δυνατότητες διαμοίρασης υπολογιστικών πόρων παρέχουν και οι *τεχνολογίες υπολογιστικού πλέγματος* [7, 8]. Παρά το γεγονός ότι αρχικά οι τεχνολογίες πλέγματος απευθύνονταν στον επιστημονικό χώρο, παρέχοντας υπολογιστικές υποδομές (συστοιχίες υπολογιστικών συστημάτων, τηλεσκόπια) μέσω των οποίων ερευνητικές ομάδες σε διαφορετικές γεωγραφικές τοποθεσίες αντάλλαζαν δεδομένα και εκτελούσαν χρονοβόρα (επεξεργαστικά) πειράματα, έχουν εξελιχθεί σε γενικού σκοπού πλατφόρμες διαμοίρασης υπολογιστικών πόρων. Με την ωρίμανση των τεχνολογιών πλέγματος, την έλευση του προτύπου της *ανοιχτής αρχιτεκτονικής υπηρεσιών πλέγματος* (Open Grid Services Architecture-OGSA) [9], και την ελεύθερη διάθεση προγραμματιστικών βιβλιοθηκών, όπως το Globus Toolkit [10], παρέχεται η δυνατότητα σε οποιονδήποτε χρήστη του διαδικτύου να χρησιμοποιεί τις υπηρεσίες πλέγματος.

Όλες οι παραπάνω τεχνολογίες δημιουργούν μεγάλο όγκο πληροφοριών που είναι διαθέσιμες μέσω του διαδικτύου. Απαιτούνται, λοιπόν, μηχανισμοί εύρεσης μέσω των οποίων ανακαλύπτονται αρχεία, υπολογιστικοί πόροι, υπηρεσίες πλέγματος και γενικότερα οποιαδήποτε πληροφορία είναι διαθέσιμη σε αυτόνομα υπολογιστικά συστήματα που επικοινωνούν μέσω πρωτοκόλλων του διαδικτύου.

1.2 Περιγραφή προβλήματος

Η εύρεση πληροφοριών σε υπολογιστικά συστήματα που είναι κατανεμημένα σε όλο το εύρος του διαδικτύου παρουσιάζει προκλήσεις που οφείλονται στην τεράστια κλίμακα, στην ετερογένεια αλλά και στην δυναμικότητα του διαδικτύου.

Συστήματα που λειτουργούν στο πλαίσιο του διαδικτύου, όπως τα δίκτυα ομότιμων κόμβων περιλαμβάνουν μεγάλο αριθμό από κόμβους[1] καθένας από τους οποίους έχει διαφορετικές υπολογιστικές δυνατότητες. Η πληροφορία που διαμοιράζεται μεταξύ των κόμβων είναι επίσης μεγάλη. Επιπλέον, οι κόμβοι των δικτύων ομότιμων κόμβων δεν είναι συνεχώς διαθέσιμοι. Αυτό σημαίνει ότι το σύνολο της πληροφορίας δεν είναι διαθέσιμο σε κάποιο κεντρικό σημείο. Επιπλέον, εφόσον ο πληθυσμός των κόμβων είναι μεγάλος είναι απίθανο κάθε κόμβος στο σύστημα να *γνωρίζει* όλους τους άλλους κόμβους. Το γεγονός αυτό δημιουργεί έλλειψη εμπιστοσύνης μεταξύ των κόμβων.

Σε μεγάλο βαθμό οι διαδικασίες εύρεσης των δεδομένων εξαρτώνται από τον τρόπο με τον οποίο οργανώνονται οι πληροφορίες. Στο πλαίσιο των συστημάτων μεγάλης κλίμακας η οργάνωση της πληροφορίας λαμβάνει χώρα σε μια κατανεμημένη πλατφόρμα που από την φύση της είναι δυναμική, με κόμβους να έρχονται και να φεύγουν από το σύστημα, και αναξιόπιστη καθώς οι κόμβοι ή οι συνδέσεις μεταξύ τους αποτυγχάνουν. Η κατασκευή ενός αξιόπιστου μηχανισμού εύρεσης πληροφοριών με βάση αυτήν την πλατφόρμα παρουσιάζει πολλές προκλήσεις.

1.3 Στόχοι διατριβής

Οι στόχοι της διατριβής επικεντρώνονται στην οργάνωση και στην εύρεση πληροφοριών στο κατανεμημένο περιβάλλον αυτόνομων υπολογιστικών συστημάτων που οργανώνονται σε δίκτυα ομότιμων κόμβων. Συνοπτικά οι στόχοι της διατριβής είναι:

- Η μελέτη των ιδιαίτερων χαρακτηριστικών των δικτύων ομότιμων κόμβων που δημιουργούν δυσκολίες στην οργάνωση και συνεπώς στην εύρεση πληροφοριών.
- Η αναζήτηση τεχνικών μέσω των οποίων εξασφαλίζεται κλιμάκωση, αξιοπιστία και ασφάλεια σε συστήματα που οργανώνονται σε δίκτυα ομότιμων κόμβων.
- Ο ορισμός των λειτουργιών αναζήτησης πληροφοριών, λαμβάνοντας ταυτόχρονα υπόψιν τις παραπάνω τεχνικές.

[1] Στο πλαίσιο των δικτύων ομότιμων κόμβων κάθε υπολογιστικό σύστημα με δυνατότητες διασύνδεσης με το διαδίκτυο ονομάζεται κόμβος

- Ο σχεδιασμός ενός μηχανισμού εύρεσης πληροφοριών για δίκτυα ομότιμων κόμβων.
- Η μελέτη της ασφάλειας των δικτύων ομότιμων κόμβων και εντοπισμός των σημαντικότερων αδυναμιών τους.
- Ο σχεδιασμός μηχανισμών ασφάλειας μέσω του οποίου εξασφαλίζεται η αξιοπιστία των δικτύων ομότιμων κόμβων στους οποίους βασίζονται οι προτεινόμενοι μέθοδοι εύρεσης.

1.4 Συνεισφορά

Προκειμένου να επιλυθεί το πρόβλημα που παρουσιάστηκε στην ενότητα 1.2 και να επιτευχθούν οι στόχοι της προηγούμενης ενότητας μελετήθηκαν τα συστήματα των δικτύων ομότιμων κόμβων, αλλά και τα συστήματα υπολογιστικού πλέγματος ώστε να αναδειχτούν οι απαιτήσεις και οι ιδιαιτερότητες που έχουν τα συστήματα μεγάλης κλίμακας, ως προς το εξεταζόμενο πρόβλημα. Συγκεκριμένα οι συνεισφορές της διατριβής είναι οι ακόλουθες:

- *Μελέτη συστημάτων ομότιμων κόμβων*. Μελετήθηκαν ερευνητικές εργασίες, τεχνολογίες και συστήματα που υπάρχουν στην θεματική περιοχή των συστημάτων ομότιμων κόμβων. Καθώς το ερευνητικό πεδίο των δικτύων ομότιμων κόμβων αποτελεί έναν από τους πιο *δραστήριους* τομείς έρευνας στην ευρύτερη περιοχή των δικτύων υπολογιστών, έχουν δημοσιευτεί πολλές εργασίες, οι οποίες έχουν μελετηθεί στο πλαίσιο της διατριβής. Επίσης, γίνεται ταξινόμηση όλων αυτών των εργασιών και παρουσιάζονται οι σημαντικότεροι εκπρόσωποι δικτύων ομότιμων κόμβων, αναδεικνύοντας τα προτερήματά τους και ανακαλύπτοντας τις αδυναμίες τους.
- *Αξιολόγηση των συστημάτων ομότιμων κόμβων*. Με βάση την τεχνογνωσία που αποκτήθηκε και αναλύοντας τα προτερήματα και τα μειονεκτήματα των δικτύων ομότιμων κόμβων επιλέχτηκαν τα δίκτυα δομημένης επικάλυψης ως η βασική *πλατφόρμα*, πάνω στην οποία οικοδομούνται οι μεθοδολογίες εύρεσης που προτείνονται στη διατριβή. Η επεκτασιμότητα των δικτύων δομημένης επικάλυψης σε συνδυασμό με πρωτόκολλα που εξασφαλίζουν την αξιόπιστη

λειτουργία του συστήματος υπό το καθεστώς συνεχών αλλαγών, διατηρώντας ταυτόχρονα την δικτυακή κίνηση σε χαμηλά επίπεδα, είναι σημαντικές απαιτήσεις για συστήματα μεγάλης κλίμακας. Επιπλέον, οι εγγυήσεις που παρέχουν στην εύρεση ενός αντικειμένου οδήγησαν στην υιοθέτηση των συστημάτων αυτών και στην αναζήτηση τρόπων με τον οποίο υλοποιούνται πολύπλοκα ερωτήματα στα δίκτυα δομημένης επικάλυψης.

- *Ορισμός της εύρεσης πληροφοριών σε δίκτυα δομημένης επικάλυψης*. Τα συστήματα δομημένης επικάλυψης παρέχουν δύο βασικές λειτουργίες μέσω των οποίων καταχωρούνται και αναζητούνται στο σύστημα ζεύγη κλειδιών και αντικειμένων $\langle a, v \rangle$. Το κλειδί είναι η τιμή ενός χαρακτηριστικού (attribute) του αντικειμένου. Ορίζεται ότι η διαδικασία εύρεσης περιλαμβάνει την αναζήτηση αντικειμένων με βάση ένα ή περισσότερα *χαρακτηριστικά* (attributes). Η διαδικασία της αναζήτησης επεκτείνεται ώστε να περιλαμβάνει ένα ή περισσότερα χαρακτηριστικά που έχουν συγκεκριμένο εύρος τιμών.

- *Υλοποίηση του πρωτοκόλλου αναζήτησης Chord*. Υλοποιήθηκε ένα σύστημα δομημένης επικάλυψης βασισμένο στο πρωτόκολλο αναζήτησης του Chord [11]. Κατά την διάρκεια της υλοποίησης και της πιλοτικής λειτουργίας του συστήματος σε πειραματικό περιβάλλον επιβεβαιώθηκε πρακτικά η θεωρητική απόδοση του Chord. Επιπλέον, η υλοποίηση του Chord έχει χρησιμοποιηθεί ως βάση για την δημιουργία πειραματικών εφαρμογών αποθήκευσης δεδομένων. Η κυριότερη χρησιμότητα της υλοποίησης είναι ότι οι βιβλιοθήκες της χρησιμοποιούνται για την πειραματική αξιολόγηση των λύσεων που προτείνονται στο πλαίσιο της διατριβής.

- *Υποστήριξη ερωτημάτων εύρους σε δίκτυα δομημένης επικάλυψης*. Με βάση τα κριτήρια που ορίστηκαν για την εύρεση πληροφοριών γίνεται ανασκόπηση της βιβλιογραφίας και εντοπίζονται λύσεις οι οποίες συνδυαζόμενες παρέχουν λύση στο πρόβλημα της εύρεσης. Ορίζεται ένας μηχανισμός μέσω του οποίου ένα σύστημα ομότιμων κόμβων υποστηρίζει ερωτήματα εύρους για ένα χαρακτηριστικό. Ο μηχανισμός αυτός βασίζεται στην λογική οργάνωση αντιγράφων από ζεύγη $\langle a, v \rangle$ σε ένα δυαδικό δένδρο, με βάση το εύρος τιμών τους. Η ρίζα του δυαδικού δένδρου είναι υπεύθυνη για όλα τα κλειδιά και κάθε παιδί διχοτομεί το πεδίο τιμών, το οποίο καθορίζει τα κλειδιά που του αντιστοιχούν.

Οι κόμβοι του δυαδικού δένδρου στους οποίους αντιστοιχίζονται τα κλειδιά, αποθηκεύονται στους κόμβους του δικτύου δομημένης επικάλυψης. Για να μην επιβαρύνονται οι κόμβοι του δικτύου επικάλυψης με πολλαπλά αντίγραφα προτείνεται ένας αλγόριθμος ο οποίος λαμβάνοντας υπόψιν το φόρτο του κάθε κόμβου διαγράφει αντίγραφα των κλειδιών, με στόχο την μείωση του φόρτου. Η συγκεκριμένη τεχνική συμβάλλει επίσης και στην αξιοπιστία του συστήματος αφού μέσω του πλεονασμού των κλειδιών επιτυγχάνεται ανοχή σε σφάλματα και αποτυχίες κόμβων. Η απόδοση της προτεινόμενης μεθόδου αξιολογείται πειραματικά βάσει του αριθμού των μηνυμάτων που πρέπει να μεταδοθούν στο δίκτυο.

- *Επισκόπηση των ζητημάτων ασφάλειας σε δίκτυα ομότιμων κόμβων*. Η αξιολόγηση της ασφάλειας των συστημάτων ομότιμων κόμβων είναι επιτακτική καθώς η αξιοπιστία των δικτύων δομημένης επικάλυψης, των εφαρμογών και των τεχνικών που βασίζονται σε αυτά, εξαρτάται σε μεγάλο βαθμό από την ασφάλεια που προσφέρουν στους χρήστες τους. Η έλλειψη μηχανισμών ασφάλειας επιτρέπει σε κακόβουλες οντότητες να εκμεταλλεύονται τις αδυναμίες του συστήματος και να υπονομεύουν τις προσφερόμενες υπηρεσίες. Η εξέταση της ασφάλειας παρουσιάζει ιδιαίτερες προκλήσεις στα δίκτυα δομημένης επικάλυψης καθώς δεν υπάρχει ένα κεντρικό σημείο το οποίο πρέπει να προστατευτεί. Επιπλέον, λόγω του τρόπου λειτουργίας τους τα δίκτυα επικάλυψης βασίζονται σε άγνωστους κόμβους για την αναζήτηση δεδομένων και κόμβων. Ένας κόμβος δεν μπορεί να βασίζεται σε ένα μικρό σύνολο κόμβων που *εμπιστεύεται*, λόγω του μεγάλου πληθυσμού των κόμβων και της συχνότητας με την οποία εισέρχονται και αποχωρούν από το σύστημα. Εξετάζουμε όλες τις αδυναμίες των δικτύων δομημένης επικάλυψης, τις επιθέσεις που εκμεταλλεύονται τις αδυναμίες τους και παρουσιάζουμε λύσεις που έχουν προταθεί. Ένας από τους παράγοντες των αδυναμιών τους αποτελεί η αδυναμία πιστοποίησης της ταυτότητας κάποιου κόμβου. Το γεγονός αυτό επιτρέπει σε επιτιθέμενες οντότητες να *πλαστογραφούν* τα αναγνωριστικά των κόμβων, να τοποθετούν πλαστούς κόμβους σε καίρια σημεία του δικτύου, και τελικά να διαβάλουν τις λειτουργίες της αναζήτησης.

- *Σχεδιασμός ενός μηχανισμού ασφάλειας για δίκτυα επικάλυψης*. Προτείνεται μια λύση για την πιστοποίηση των κόμβων μέσω μίας *υποδομής δημοσίου*

κλειδιού. Ο κατανεμημένος τρόπος λειτουργίας των συστημάτων ομότιμων κόμβων και η σχεδιαστική απαίτηση που θέτουν για την μη εξάρτησή τους από κεντρικοποιημένες οντότητες, όπως μια αρχή πιστοποίησης, δεν επιτρέπουν την χρήση παραδοσιακών κεντρικών υποδομών δημοσίου κλειδιού. Για αυτόν τον λόγο σχεδιάζεται και προτείνεται μια κατανεμημένη υποδομή δημοσίου κλειδιού, το Chord-PKI [12]. Οι λειτουργίες της πιστοποίησης και αποθήκευσης των πιστοποιητικών κατανέμονται στους κόμβους του δικτύου δομημένης επικάλυψης. Ορίζουμε τα πρωτόκολλα αρχικοποίησης του συστήματος και τα πρωτόκολλα ανταλλαγής των κλειδιών και αναλύουμε τα πρωτόκολλα πιστοποίησης και ανάκλησης μέσω των οποίων οι κόμβοι πιστοποιούνται μέσω ενός συνόλου έμπιστων κόμβων. Το σύστημα λαμβάνει υπόψιν την ύπαρξη κακόβουλων κόμβων στο σύστημα και παρέχει εγγυήσεις για την ορθή λειτουργία της υπηρεσίας πιστοποίησης, δοθέντος ότι ο αριθμός των κακόβουλων οντοτήτων δεν ξεπερνά ένα όριο t, το οποίο είναι παράμετρος του συστήματος. Αυξάνοντας το όριο t αυξάνεται η ανοχή του συστήματος σε κακόβουλους κόμβους αλλά ταυτόχρονα αυξάνεται και το δικτυακό κόστος του συστήματος. Μέσω πειραμάτων αναλύεται η επίδοση του συστήματος για διαφορετικά όρια ανοχής κακόβουλων κόμβων. Ο προσομοιωτής του Chord, επεκτάθηκε ώστε να υποστηρίζει και τις λειτουργίες ενός συστήματος κρυπτογραφίας κατωφλιού RSA, και μετρήθηκε το επεξεργαστικό (χρονικό) κόστος της πιστοποίησης και παρουσιάστηκε η συμπεριφορά του ChordPKI συναρτήσει της αύξησης του πληθυσμού των κακόβουλων κόμβων.

- *Σχεδιασμός ψηφιακής βιβλιοθήκης πανεπιστημιακών δεδομένων*. Σχεδιάζονται οι βασικές λειτουργίες της δημοσίευσης, της αναζήτησης και της ανάκτησης των εγγράφων βάσει της αρχιτεκτονικής OGSA [9] για συστήματα υπολογιστικού πλέγματος. Οι λειτουργίες της βιβλιοθήκης υλοποιούνται από τους διαθέσιμους υπολογιστικούς πόρους του πλέγματος. Τόσο οι τεχνολογίες υπολογιστικού πλέγματος όσο και τα συστήματα ομότιμων κόμβων έχουν ώς στόχο την διαμοίραση υπολογιστικών πόρων. Η μελέτη των συστημάτων υπολογιστικού πλέγματος βοήθησε στον εντοπισμό των κοινών στοιχείων με τα συστήματα των ομότιμων κόμβων και κυρίως στον εντοπισμό των ελλείψεων που υπάρχουν και στις δύο τεχνολογίες όπως στις μεθόδους εύρεσης υπολογιστικών πόρων.

1.5 Δομή διατριβής

Στο κεφάλαιο αυτό παρουσιάζονται σημαντικά ζητήματα που αφορούν συστήματα που λειτουργούν σε δίκτυα ευρείας κλίμακας. Αναδεικνύεται το πρόβλημα της εύρεσης πληροφοριών σε περιβάλλοντα μεγάλης κλίμακας, ορίζονται οι στόχοι της διατριβής που είναι η δημιουργία τεχνικών μέσω των οποίων μπορεί να κατασκευαστεί ένας μηχανισμός εύρεσης πληροφοριών και εξηγείται η συνεισφορά της διατριβής.

Στο κεφάλαιο 2 περιγράφονται τα δίκτυα ομότιμων κόμβων (peer-to-peer networks), αναλύεται η έννοια του δικτύου επικάλυψης στο πλαίσιο των ομότιμων κόμβων και αναφέρονται οι υποκατηγορίες δικτύων επικάλυψης, δομημένη και αδόμητη. Στην συνέχεια εστιάζουμε στα δίκτυα δομημένης επικάλυψης, αναλύουμε πώς καλύπτουν τις απαιτήσεις της επεκτασιμότητας, της δυναμικής συμπεριφοράς και της συμμετρίας στις λειτουργίες των κόμβων, και παρουσιάζουμε τους βασικότερους εκπροσώπους τους.

Στο κεφάλαιο 3 ορίζεται η διαδικασία της εύρεσης στα συστήματα ομότιμων κόμβων. Συγκεκριμένα αναλύεται πώς ανακαλύπτονται πόροι με βάση πολλαπλά κριτήρια επί των χαρακτηριστικών ενός πόρου. Στην συνέχεια αναφέρονται οι αδυναμίες των δικτύων δομημένης επικάλυψης ως προς την εφαρμογή των ερωτημάτων και προτείνονται μεθοδολογίες μέσω των οποίων επέρχεται τελικά η υποστήριξη των ερωτημάτων. Δίνεται ιδιαίτερη βάση στο πρόβλημα των ερωτημάτων εύρους, για το οποίο προτείνεται λύση που βασίζεται στην *κατανομή* ενός λογικού δυαδικού δένδρου αναζήτησης στους κόμβους του δικτύου επικάλυψης και τέλος αξιολογείται η προτεινόμενη λύση.

Στο κεφάλαιο 4 γίνεται επισκόπηση των προβλημάτων ασφαλείας που παρουσιάζονται στα δίκτυα δομημένης επικάλυψης. Αναλύονται οι αδυναμίες στην ασφάλεια των δικτύων δομημένης επικάλυψης και αναφέρονται λύσεις για κάθε κατηγορία επιθέσεων που έχουν προταθεί στην βιβλιογραφία. Στην συνέχεια αναλύεται το πρόβλημα της *πιστοποίησης της ταυτότητας* μιας οντότητας (χρήστης, κόμβος) σε δίκτυα δομημένης επικάλυψης και προτείνεται μια πρότυπη λύση, το Chord-PKI [12]. Περιγράφονται οι κρυπτογραφικές τεχνικές στις οποίες βασίζεται

η λύση αυτή και αναλύονται τα πρωτόκολλα λειτουργίας της. Τέλος, αξιολογείται η απόδοσή του με την χρήση προσομοιώσεων και πειραμάτων.

Στο κεφάλαιο 5 περιγράφεται μια πρότυπη αρχιτεκτονική ψηφιακής βιβλιοθήκης προσανατολισμένη σε πανεπιστημιακά τεκμήρια. Η ψηφιακή βιβλιοθήκη χρησιμοποιεί τεχνολογίες υπολογιστικού πλέγματος με την βοήθεια των οποίων οι λειτουργίες της κατανέμονται στις διαθέσιμες υπολογιστικές υποδομές. Η κατανομή των λειτουργιών πραγματοποιείται σχεδιάζοντας την αρχιτεκτονική της ψηφιακής βιβλιοθήκης στα πρότυπα της αρχιτεκτονικής OGSA.

Τέλος, στο κεφάλαιο 6 αναφέρονται τα συμπεράσματα τις διατριβής και θέματα μελλοντικής έρευνας.

Κεφάλαιο 2

Εννοιολογική θεμελίωση δικτύων ομότιμων κόμβων

Περίληψη

Στο κεφάλαιο αυτό περιγράφονται τα δίκτυα ομότιμων κόμβων (Peer-to-Peer Networks), αναλύεται η έννοια του δικτύου επικάλυψης στο πλαίσιο των δικτύων ομότιμων κόμβων και αναφέρονται οι υποκατηγορίες δικτύων επικάλυψης, δομημένη και αδόμητη. Στην συνέχεια εστιάζουμε στα δομημένα δίκτυα επικάλυψης και παρουσιάζουμε τους βασικότερους εκπροσώπους τους.

2.1 Εισαγωγή

Την τελευταία δεκαετία παρατηρείται ραγδαία εξάπλωση εφαρμογών και συστημάτων που βασίζονται σε *δίκτυα ομότιμων κόμβων* (peer-to-peer networks) μέσω των οποίων ένας μεγάλος αριθμός χρηστών του διαδικτύου *ανταλλάσσει* αρχεία (ως επί το πλείστον μουσικού-ψυχαγωγικού περιεχομένου). Τα συστήματα αυτά δημιουργούν ένα *δίκτυο επικάλυψης* που τους επιτρέπει να διαμοιράζονται πόρους (π.χ. πολυμεσικά αρχεία) που είναι αποθηκευμένα σε έναν ή περισσότερους υπολογιστές του ιδεατού δικτύου επικάλυψης.

Οι εφαρμογές διαμοίρασης αρχείων [4, 13, 14, 5, 6] και οι εφαρμογές εθελοντικού υπολογισμού [15, 16] αποδεικνύουν ότι οι συνδυασμένες αποθηκευτικές και επεξεργαστικές δυνατότητες των κόμβων στα ομότιμα δίκτυα υπερκαλύπτουν τις ανάγκες ακόμη και των πιο απαιτητικών εφαρμογών. Το μοντέλο λειτουργίας των ομότιμων κόμβων βασίζεται στην φιλοσοφία του *δούναι και λαβείν*. Συγκεκριμένα αν ένας κόμβος επιθυμεί να λαμβάνει τις υπηρεσίες που παρέχουν οι άλλοι κόμβοι τότε πρέπει και αυτός να *συνεισφέρει* πόρους για την υπηρεσία αυτή. Ο μεγάλος αριθμός των χρηστών των εφαρμογών δικτύων ομότιμων κόμβων δείχνει ότι το μοντέλο αυτό έχει απήχηση στους χρήστες του διαδικτύου. Μελέτες [17, 18] δείχνουν ότι σημαντικό ποσοστό της δικτυακής κίνησης του διαδικτύου οφείλεται στα προγράμματα που βασίζονται σε δίκτυα ομότιμων κόμβων μέσω των οποίων διαμοιράζονται πληροφορίες. Παράλληλα, όμως, ο μεγάλος αριθμός των χρηστών, σε συνδυασμό με την συμπεριφορά τους ορίζει τις προδιαγραφές και τις απαιτήσεις των συστημάτων ομότιμων κόμβων. Για παράδειγμα ο συχνός ρυθμός εισαγωγής και αναχώρησης των κόμβων από το σύστημα, δημιουργεί επιπλέον δικτυακή κίνηση που οφείλεται σε μηνύματα μέσω των οποίων ενημερώνονται οι κόμβοι για την καινούρια κατάσταση του δικτύου.

Αντικείμενο του κεφαλαίου αυτού είναι η παρουσίαση των σημαντικότερων πρωτοκόλλων δικτύων ομότιμων κόμβων μέσω των οποίων ικανοποιούνται οι προδιαγραφές της αξιοπιστίας, της επεκτασιμότητας και της δυναμικής συμπεριφοράς των κόμβων. Παράλληλα, αναδεικνύονται προβλήματα που σχετίζονται με την εύρεση πληροφοριών στα περιβάλλοντα των δικτύων ομότιμων κόμβων. Οι μεθοδολογίες

για εύρεση πληροφοριών που προτείνονται στη διατριβή λειτουργούν σε περιβάλλοντα ευρείας κλίμακας που έχουν τα ίδια χαρακτηριστικά και τις ίδιες απαιτήσεις με τις εφαρμογές των δικτύων ομότιμων κόμβων.

Στις επόμενες ενότητες παραθέτουμε συνοπτικά την εξέλιξη των δικτύων ομότιμων κόμβων, αναλύουμε τις βασικές έννοιες των δικτύων επικάλυψης και αναφέρουμε τους βασικούς εκπροσώπους τους. Το κεφάλαιο ξεκινά με την περιγραφή των δικτύων ομότιμων κόμβων κατά την οποία καθορίζονται οι ιδιότητες τους. Παράλληλα μέσα από την ιστορική αναδρομή στις γενιές των δικτύων ομότιμων κόμβων εξηγούνται οι στόχοι και οι προκλήσεις που αντιμετώπισαν οι ερευνητές και οι κατασκευαστές των συστημάτων αυτών. Στην συνέχεια, εξηγούμε την βασική έννοια των δικτύων επικάλυψης βάσει της οποίας ορίζονται στα δίκτυα αυτά κανόνες δρομολόγησης και εύρεσης δεδομένων σε επίπεδο εφαρμογής. Επικεντρωνόμαστε στα *δομημένα δίκτυα επικάλυψης* αναλύοντας τον τρόπο λειτουργίας τους. Επίσης, αναφέρουμε τους βασικούς εκπροσώπους των δικτύων δομημένης επικάλυψης και περιγράφουμε αναλυτικά το δίκτυο επικάλυψης του πρωτοκόλλου Chord [11], στο οποίο βασίζονται οι μηχανισμοί, οι μεθοδολογίες και τα πειράματα που προτείνονται στη διατριβή. Τέλος, περιγράφονται προβλήματα στην εύρεση και στην ασφάλεια των δικτύων επικάλυψης, τα οποία αντιμετωπίζονται στο πλαίσιο της διατριβής προκειμένου να δημιουργηθεί ένας αξιόπιστος μηχανισμός εύρεσης πληροφοριών σε περιβάλλοντα δικτύων ομότιμων κόμβων.

2.2 Δίκτυα ομότιμων κόμβων

Ο Risson [19] αναφέρει ότι τα δίκτυα ομότιμων κόμβων χαρακτηρίζονται από τις τρεις παρακάτω ιδιότητες:

- Αυτο-οργάνωση,
- Συμμετρική επικοινωνία,
- Κατανεμημένος έλεγχος.

Τα συστήματα αυτά είναι αυτο-οργανούμενα καθώς το δίκτυο προσαρμόζεται στις (συχνές) αφίξεις, αναχωρήσεις και αποτυχίες των κόμβων. Η επικοινωνία μεταξύ

των κόμβων είναι συμμετρική αφού οι κόμβοι είναι *πελάτες* και *εξυπηρετητές* ταυτόχρονα σε αντίθεση με κλασικό μοντέλο *πελάτη-εξυπηρετητή*. Επίσης, στα συστήματα αυτά δεν υπάρχει κάποιος κεντρικός έλεγχος.

Επίσης, οι Androutsellis-Theotokis,Spinellis στην βιβλιογραφική τους μελέτη [20] περιγράφουν τα συστήματα ομότιμων κόμβων ως κατανεμημένα συστήματα που αποτελούνται από διασυνδεδεμένους κόμβους ικανούς να αυτο- οργανώνονται στην τοπολογία του δικτύου με στόχο την διαμοίραση πόρων όπως ψηφιακά δεδομένα, επεξεργαστική ισχύ, αποθηκευτικό χώρο και δικτυακό εύρος ζώνης. Επίσης, τα συστήματα αυτά πρέπει να προσαρμόζονται στις αποτυχίες των κόμβων, και στους μεγάλους μεταβλητούς πληθυσμούς των κόμβων ενώ ταυτόχρονα πρέπει να διατηρούν την συνδεσιμότητα και την απόδοση του συστήματος σε ανεκτά όρια, χωρίς να απαιτείται η μεσολάβηση κάποιας κεντρικής αρχής ελέγχου.

Ουσιαστικά, πρόκειται για κατανεμημένα συστήματα που έχουν βασικό στόχο τη διαμοίραση πόρων. Βέβαια, όπως θα δούμε και στις επόμενες ενότητες, ο *βαθμός της αποκέντρωσης* του εκάστοτε συστήματος ποικίλει. Υπάρχουν συστήματα ομότιμων κόμβων, οι λειτουργίες των οποίων κατανέμονται πλήρως σε όλους τους κόμβους. Άλλα συστήματα αναθέτουν κάποιες λειτουργίες σε ορισμένους κόμβους με ιδιαίτερες δυνατότητες [21], ενώ κάποια άλλα διατηρούν ορισμένες βασικές λειτουργίες σε κάποιον κεντρικό εξυπηρετητή [4]. Παρόλο που ο βαθμός της αποκέντρωσης αλλάζει από σύστημα σε σύστημα ο βασικός τους στόχος είναι η εύρεση των πόρων που διαμοιράζονται.

2.2.1 Ιστορική εξέλιξη

Το πρόγραμμα διαμοίρασης μουσικών αρχείων Napster [4] ήταν η πρώτη εφαρμογή διαμοίρασης αρχείων που είχε μαζική απήχηση στους χρήστες του διαδικτύου, και η οποία χρησιμοποίησε ευρέως τον όρο των *ομότιμων κόμβων* (peer nodes). Στην πρώτη έκδοση του Napster υπήρχε ένας κεντρικός εξυπηρετητής που διαχειριζόταν το κεντρικό ευρετήριο του συστήματος. Το ευρετήριο αυτό περιείχε τις εξής πληροφορίες για κάθε *πελάτη* που είναι εγγεγραμμένος στο σύστημα:

- Τη διεύθυνση διαδικτύου (IP addess), την θύρα λειτουργίας και τον προσφερόμενο ρυθμό μετάδοσης της πληροφορίας στην εφαρμογή του πελάτη,

- Τη λίστα με τα αρχεία που διαμοιράζει κάθε πελάτης.

Κάθε νέος πελάτης επικοινωνεί με τον κεντρικό εξυπηρετητή και τον ενημερώνει για τα αρχεία που είναι διατεθειμένος να διαμοιράζει. Αντίστοιχα, όταν κάποιος πελάτης επιθυμεί να προσπελάσει κάποιο αρχείο, προωθεί το ερώτημα του στον κεντρικό εξυπηρετητή, ο οποίος του επιστρέφει μια λίστα με πελάτες που έχουν το ζητούμενο αρχείο. Με αυτόν τον τρόπο, στην εφαρμογή Napster, η αποθήκευση και η μετάδοση των αρχείων κατανέμεται στους πελάτες του συστήματος. Βέβαια η αναζήτηση των αρχείων είναι μια υπηρεσία που υλοποιείται κεντρικά. Αυτό αποτελούσε το σημαντικότερο τεχνικό πρόβλημα[1] του Napster και αντιμετωπίστηκε κατανέμοντας το κεντρικό ευρετήριο σε μια *συστοιχία από εξυπηρετητές*.

Η επόμενη γενιά εφαρμογών δικτύων ομότιμων κόμβων δεν διατηρεί κάποιο κεντρικό ευρετήριο αλλά κατανέμει την διαδικασία εύρεσης στους κόμβους του συστήματος. Ο βασικός εκπρόσωπος της δεύτερης γενιάς είναι η εφαρμογή Gnutella [5]. Τα πρωτόκολλα της δεύτερης γενιάς βασίζονται στο γεγονός ότι κάθε κόμβος-πελάτης διατηρεί τοπικό ευρετήριο των αρχείων-πόρων που διαμοιράζει. Τα πρωτόκολλα αυτά παρέχουν τεχνικές μέσω των οποίων ένας κόμβος μαθαίνει για τους πόρους κάποιου άλλου. Οι πιο διαδεδομένες τεχνικές είναι της *πλημμύρας* (flooding) [22] και του *τυχαίου περιπάτου* [22]. Στις τεχνικές πλημμύρας ένας κόμβος προωθεί κάποιο ερώτημα προς όλους τους κόμβους που γνωρίζει. Οι κόμβοι αυτοί επιστρέφουν αποτελέσματα στον αρχικό κόμβο (αν έχουν) και προωθούν το ερώτημα προς τους κόμβους που γνωρίζουν. Για να περιοριστεί ο αριθμός των μηνυμάτων που μεταδίδονται στο δίκτυο και για να μην αναμεταδίδονται συνεχώς τα ερωτήματα υπάρχει ένα μέγιστο όριο στον αριθμό των κόμβων προς τους οποίους μπορεί να αναμεταδοθεί ένα ερώτημα. Συγκεκριμένα, στο ερώτημα εμπεριέχεται και μια μεταβλητή που δηλώνει πόσες φορές μπορεί να προωθηθεί το ερώτημα (πεδίο *time-to-live*). Κάθε κόμβος που λαμβάνει ένα τέτοιο ερώτημα μειώνει τον μετρητή και αν αυτός δεν έχει μηδενιστεί ο κόμβος προωθεί το ερώτημα στους γνωστούς του κόμβους.

Το βασικό πρόβλημα της δεύτερης γενιάς των δικτύων ομότιμων κόμβων είναι ότι δεν παρέχουν εγγυήσεις εύρεσης. Υπάρχει περίπτωση κάποιο ερώτημα να μην επιστρέψει το επιθυμητό αποτέλεσμα παρόλο που μέσα στο σύστημα υπάρχει πόρος

[1] εκτός από το νομικό πρόβλημα που οδήγησε στην διακοπή της λειτουργίας του

που ικανοποιεί το κριτήριο του ερωτήματος. Η τρίτη γενιά των δικτύων ομότιμων κόμβων, που αναφέρονται αναλυτικότερα στην ενότητα 2.3, παρέχει εγγυήσεις στην εύρεση. Αν οι πόροι που ικανοποιούν τα κριτήρια της εύρεσης είναι διαθέσιμοι τότε το ερώτημα θα επιστρέψει τους κόμβους που αντιστοιχούν στους πόρους.

2.2.2 Δίκτυα επικάλυψης

Τα δίκτυα ομότιμων κόμβων οργανώνονται στο λεγόμενο *δίκτυο επικάλυψης*. Η έννοια του δικτύου επικάλυψης δεν είναι καινούργια και έχει χρησιμοποιηθεί κατά κόρον για να διευκολύνει ερευνητές που μελετούν καινούργια πρωτόκολλα και μηχανισμούς δικτύων που δεν υποστηρίζονται από το υφιστάμενο δίκτυο. Συγκεκριμένα, το δίκτυο επικάλυψης είναι ένα λογικό δίκτυο που δημιουργείται *πάνω* από ένα φυσικό δίκτυο. Τα δίκτυα αυτά υλοποιούνται στο επίπεδο εφαρμογής του μοντέλου αναφοράς OSI. Οι συνδέσεις μεταξύ των κόμβων του δικτύου επικάλυψης υλοποιούνται μέσω του υφισταμένου φυσικού δικτύου.

Στο πλαίσιο του διαδικτύου έχουν δημιουργηθεί πολλά δίκτυα επικάλυψης καθένα από τα οποία εξυπηρετούσε διαφορετικούς σκοπούς. Στο MBone [23] δημιουργούνται *λογικά* κανάλια επικοινωνίας πάνω από το διαδίκτυο που συνδέουν δίκτυα με υποστήριξη πολλαπλής εκπομπής (multicast) στο επίπεδο του πρωτοκόλλου IP. Στο δίκτυο επικάλυψης RON [24] στόχος είναι η αξιόπιστη επικοινωνία μεταξύ των κόμβων του συστήματος, η οποία επιτυγχάνεται ανακαλύπτοντας εναλλακτικά μονοπάτια μεταξύ των κόμβων που βρίσκονται σε διαφορετικά αυτόνομα συστήματα, τα οποία δεν μπορεί να βρεθούν μέσω του πρωτοκόλλου BGP του διαδικτύου. Επίσης *δίκτυα παράδοσης περιεχομένου* (Content Delivery Networks) όπως το Akamai [25] χρησιμοποιούν τεχνικές δικτύων επικάλυψης για να προσφέρουν ταχύτερη μετάδοση περιεχομένου σε εφαρμογές του διαδικτύου όπως το HTTP, και γενικότερα σε εφαρμογές μετάδοσης δεδομένων σε πραγματικό χρόνο (π.χ. video streaming). Τέλος, οι εφαρμογές δικτύων ομότιμων κόμβων οργανώνονται σε δίκτυο επικάλυψης προκειμένου να διευκολύνονται οι μεταφορές των αρχείων και η αναζήτηση των δεδομένων.

Συγκεκριμένα, η εφαρμογή Gnutella δημιουργεί δυναμικά το δίκτυο επικάλυψης καθώς οι κόμβοι συνδέονται μεταξύ τους. Το συγκεκριμένο δίκτυο δεν έχει κάποια δομή καθώς οι κόμβοι ανακαλύπτουν ο ένας τον άλλο με τυχαίο τρόπο. Όπως

αναφέρθηκε στην προηγούμενη ενότητα αυτό δημιουργεί πρόβλημα στην αναζήτηση αφού δεν μπορεί να δοθούν εγγυήσεις ότι θα βρεθεί κάποιο αρχείο. Για να αντιμετωπιστεί αυτό το πρόβλημα πρέπει οι κόμβοι και τα δεδομένα που αντιστοιχούν σε αυτούς να οργανώνονται με προκαθορισμένο τρόπο στο δίκτυο επικάλυψης. Η οργάνωση των κόμβων βασίζεται σε μια *λογική* τοπολογία στον γεωμετρικό χώρο. Επειδή η τοπολογία έχει κάποια συγκεκριμένη γεωμετρική δομή (π.χ. κύκλος, υπερκύβος) το δίκτυο επικάλυψης ονομάζεται *δομημένη επικάλυψη* (structured overlay) ενώ σε διαφορετική περίπτωση ονομάζεται *αδόμητη επικάλυψη* (unstructured overlay)[19]. Τα πρωτόκολλα της δεύτερης γενιάς των δικτύων ομότιμων κόμβων είναι αδόμητης επικάλυψης. Αυτό έχει ως συνέπεια να μην υπάρχει ένας σαφής τρόπος οργάνωσης των δεδομένων με άμεσο αποτέλεσμα τη δημιουργία μεθόδων εύρεσης που δημιουργούν μεγάλη δικτυακή κίνηση (π.χ τεχνικές πλημμύρας) χωρίς να παρέχονται εγγυήσεις για την εύρεση δεδομένων. Αντίθετα, τα πρωτόκολλα της τρίτης γενιάς των δικτύων ομότιμων κόμβων, τα δίκτυα δομημένης επικάλυψης, βασίζονται σε μια προκαθορισμένη τοπολογία δικτύου που είναι οργανωμένη πάνω σε κάποιο συγκεκριμένο γεωμετρικό σχήμα (π.χ κύκλος). Τα δεδομένα και οι κόμβοι *αντιστοιχίζονται* σε προκαθορισμένα *σημεία* στην τοπολογία με αποτέλεσμα οι διαδικασίες της εύρεσης να βρίσκουν πάντοτε ένα αντικείμενο αν αυτό υπάρχει διαθέσιμο στο δίκτυο. Οι εγγυήσεις εύρεσης που παρέχονται, σε συνδυασμό με άλλες επιθυμητές ιδιότητες που διαθέτουν όπως αξιοπιστία και επεκτασιμότητα, οδήγησαν στην υιοθέτηση των δικτύων δομημένης επικάλυψης για τους μηχανισμούς που προτείνονται στην διατριβή.

2.3 Συστήματα δικτύων δομημένης επικάλυψης

Τα δίκτυα δομημένης επικάλυψης έχουν συγκεκριμένη δομή και τα δεδομένα (αρχεία, ή κομμάτια αρχείων (chunks)) αποθηκεύονται σε προκαθορισμένες τοποθεσίες που ορίζονται μέσω των αλγορίθμων [11, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36] του κάθε συστήματος. Η αποθήκευση ενός αντικειμένου (δεδομένου) σε κάποια τοποθεσία ονομάζεται *αντιστοίχηση*[2] (mapping) του αντικειμένου. Στόχος της αντιστοίχησης είναι η δημιουργία γρήγορων μεθόδων αναζήτησης. Για τον λόγο αυτό

[2]Στο πλαίσιο των δικτύων δομημένης επικάλυψης οι λέξεις αντιστοίχηση, ανάθεση, διαχείριση έχουν ταυτόσημη έννοια

οι αλγόριθμοι αναζήτησης των δικτύων δομημένης επικάλυψης ονομάζονται και πρωτόκολλα αναζήτησης (lookup systems), και αντίστοιχα, τα δίκτυα επικάλυψης ονομάζονται συστήματα αναζήτησης.

Τα συστήματα αναζήτησης παρέχουν παρόμοιες μεθόδους με αυτές που υπάρχουν στους *πίνακες κατακερματισμού* (hash tables). Συγκεκριμένα, παρέχουν δυο βασικές μεθόδους μέσω οποίων:

- Αποθηκεύονται ζεύγη κλειδιών-αντικειμένων $\langle a, v \rangle$ (key-value pairs),
- Ανακτώνται τα αντικείμενα με βάση το κλειδί που αντιστοιχεί στο ζεύγος.

Στο παραπάνω ζεύγος το a μπορεί να είναι το όνομα ενός αρχείου και το v είτε το ίδιο το αρχείο είτε κάποιος σύνδεσμος URI [37] στο αρχείο. Στα συστήματα δομημένης επικάλυψης χρησιμοποιείται η συνάρτηση κατακερματισμού (hash function) H, μέσω της οποίας οποίας προκύπτουν τα αναγνωριστικά για τους κόμβους και τα δεδομένα του συστήματος. Παράλληλα, ορίζονται κανόνες αντιστοίχησης των δεδομένων στους κόμβους, οι οποίοι βασίζονται στην ελαχιστοποίηση της απόστασης μεταξύ των αντίστοιχων αναγνωριστικών (κόμβου και δεδομένων). Η συγκεκριμένη μεθοδολογία αντιστοίχησης είναι αρκετά ευέλικτη αφού τα αναγνωριστικά για το a μπορεί να αντιστοιχούν σε οποιοδήποτε χαρακτηριστικό του αρχείου (π.χ. μέγεθος, όνομα, λέξεις κλειδιά). Επιπλέον, λόγω του τυχαίου τρόπου με τον οποίο παράγονται τα αναγνωριστικά των δεδομένων, δηλαδή μέσω της συνάρτησης κατακερματισμού, επιτυγχάνεται ομοιόμορφη κατανομή των δεδομένων στο πεδίο τιμών της συνάρτησης H. Ταυτόχρονα όμως καταστρέφεται οποιαδήποτε σχέση προϋπήρχε μεταξύ των τιμών του χαρακτηριστικού a. Το γεγονός αυτό δυσκολεύει την δημιουργία πολύπλοκων ερωτημάτων επί του χαρακτηριστικού a, όπως αναζήτηση με βάση συγκεκριμένο εύρος τιμών για το a. Τα συστήματα δομημένης επικάλυψης ονομάζονται και συστήματα DHT (Distributed Hash Tables) λόγω της συνάρτησης κατακερματισμού και λόγω των μεθόδων `get` και `put` μέσω των οποίων αναζητώνται τα αναγνωριστικά και αποθηκεύονται τα ζεύγη $\langle a, v \rangle$. Καθόλη την διάρκεια της διατριβής οι όροι συστήματα δομημένης επικάλυψης και DHT έχουν ταυτόσημη έννοια.

Στην συνέχεια περιγράφουμε βασικές των συστημάτων αυτών και αναλύουμε τους τρόπους με τους οποίους επιτυγχάνεται η επεκτασιμότητα και η αξιοπιστία τους.

2.3.1 Περιγραφή βασικών εννοιών

Τα συστήματα DHT παρέχουν την δυνατότητα εύρεσης αναγνωριστικών σε ένα κοινό πεδίο τιμών που ορίζεται από την συνάρτηση κατακερματισμού που χρησιμοποιείται. Τα αναγνωριστικά αντιστοιχούν σε ένα μεγάλο σύνολο δεδομένων και σε κόμβους οι οποίοι παρουσιάζουν δυναμική συμπεριφορά καθώς εισάγονται και αποχωρούν από το σύστημα αρκετά συχνά. Οι βασικοί στόχοι που έχουν τα συστήματα αυτά είναι:

1. *Μικρός βαθμός κόμβων*. Βαθμός ενός κόμβου ονομάζεται ο αριθμός των συνδέσεων που διατηρεί προς άλλους κόμβους. Αν κάθε κόμβος διατηρεί μόνο λίγες *συνδέσεις* προς άλλους κόμβους, τότε περιορίζεται ο αντίκτυπος του υψηλού ρυθμού αφίξεων και αναχωρήσεων κόμβων.

2. *Μικρή διάμετρος δικτύου*. Η διάμετρος του δικτύου εξαρτάται από τον αριθμό των κόμβων που πρέπει να προσπελαστούν προκειμένου να επικοινωνήσουν δυο κόμβοι στο σύστημα. Ο αριθμός των κόμβων που μεσολαβούν μεταξύ δυο κόμβων πρέπει να είναι όσο το δυνατόν μικρότερος ώστε να ελαχιστοποιείται η καθυστέρηση στην επικοινωνία τους.

3. *Ατομική δρομολόγηση*. Οι κόμβοι πρέπει να ανακαλύπτουν μόνοι τους το συντομότερο μονοπάτι. Σε κάθε ενδιάμεσο κόμβο (σε κάθε hop) πρέπει το ερώτημα να *πλησιάζει* προς τον προορισμό.

4. *Ευρωστία*. Πρέπει να υπάρχει ένα μονοπάτι προς τον προορισμό ακόμη και αν κάποιοι ενδιάμεσοι κόμβοι (συνδέσεις) αποτυγχάνουν.

Τα συστήματα DHT που χρησιμοποιούνται σήμερα, έχουν επηρεαστεί από τρεις βασικές εργασίες που έγιναν στην δεκαετία του 1990 [38, 39, 40]. Η χρήση πινάκων δρομολόγησης συγκεκριμένου μεγέθους [39], μέσω των οποίων εντοπίζονται αντικείμενα, και η χρήση του *αποδοτικού κατακερματισμού* (consistent hashing) για την ελαχιστοποίηση της αναδιανομής των κλειδιών, αποτελούν βασικούς μηχανισμούς των συστημάτων DHT.

Τα δεδομένα που αποθηκεύονται στα δίκτυα δομημένης επικάλυψης ονομάζονται *αντικείμενα*. Αντικείμενα και κόμβοι αποκτούν αναγνωριστικά (identifiers) με κοινό

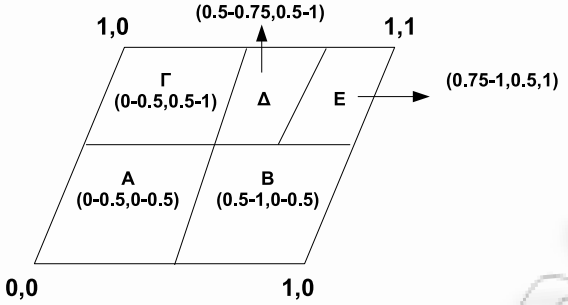
πεδίο τιμών της συνάρτησης κατακερματισμού. Η εύρεση του κόμβου στον οποίο είναι αποθηκευμένο (αντιστοιχεί) το αντικείμενο Α, βασίζεται στον εντοπισμό του αναγνωριστικού του κόμβου που είναι πλησιέστερο προς το αναγνωριστικό του κλειδιού του Α. Για το λόγο αυτό η δρομολόγηση των συστημάτων αυτών ονομάζεται και *δρομολόγηση με βάση το κλειδί* (key-based routing). Ο όρος *πλησιέστερος* ορίζει την μικρότερη απόσταση μεταξύ δυο αναγνωριστικών. Η συνάρτηση υπολογισμού της απόστασης δυο αναγνωριστικών ορίζεται ξεχωριστά για κάθε αλγόριθμο αναζήτησης.

2.3.2 Βασικοί εκπρόσωποι δικτύων δομημένης επικάλυψης

Η έρευνα στην θεματική περιοχή των δικτύων δομημένης επικάλυψης οδήγησε στην δημιουργία αρκετών πρωτοκόλλων δρομολόγησης. Παρακάτω περιγράφουμε τους βασικούς εκπρόσωπους δικτύων δομημένων επικάλυψης και στην συνέχεια αναλύουμε σε μεγαλύτερη λεπτομέρεια τον αλγόριθμο αναζήτησης του πρωτοκόλλου Chord [11] στο οποίο βασίζονται οι μηχανισμοί που έχουν προταθεί στην διατριβή. Τα συστήματα αυτά παρέχουν τις δύο βασικές μεθόδους μέσω των οποίων εγγράφονται ζεύγη κλειδιών-δεδομένων και αναζητούνται τα δεδομένα με βάση το αναγνωριστικό (κλειδί). Όλα τα πρωτόκολλα χρησιμοποιούν κάποια συγκεκριμένη δομή (υπερκύβος για το CAN [26], δένδρα για το Tapestry [27] και το Kademlia [28], και δακτύλιος για το Chord [11]), προτείνουν αλγόριθμους αναζήτησης αναγνωριστικών και παρέχουν μηχανισμούς *ανανέωσης* των πινάκων δρομολόγησής τους υπό το καθεστώς διαρκών αλλαγών στην κατάσταση του δικτύου.

2.3.2.1 Content addressable Network-CAN

Το σύστημα CAN [26] βασίζεται στην τοπολογία του υπερκύβου δ διαστάσεων, όπου δ παράμετρος του συστήματος. Ο υπερκύβος διαμερίζεται δυναμικά μεταξύ των κόμβων του συστήματος ώστε κάθε κόμβος που είναι *υπεύθυνος για μια περιοχή του υπερκύβου* να έχει ως γείτονές του κόμβους (στους πίνακες δρομολόγησης του) οι οποίοι είναι υπεύθυνοι για γειτονικές περιοχές στον υπερκύβο. Για παράδειγμα, το σχήμα 2.1 παρουσιάζει ένα σύστημα CAN δύο ($\delta = 2$) διαστάσεων ($[0,1]x[0,1]$) με 5 κόμβους στο σύστημα.



Σχήμα 2.1: Επικάλυψη δυο διαστάσεων του CAN με 5 κόμβους στο σύστημα.

Σε αυτόν το 2-διάστατο χώρο αποθηκεύονται ζεύγη κλειδιών-αντικειμένων (έστω K_1, V_1). Χρησιμοποιώντας την συνάρτηση κατακερματισμού του συστήματος, η οποία έχει πεδίο τιμών το $[0, 1]$, το K_1 αντιστοιχεί σε ένα σημείο P (0.2,0.1). Για το σημείο αυτό υπεύθυνος είναι ο κόμβος Α (σχήμα 2.1). Η διαδικασία της δρομολόγησης κάποιου ερωτήματος προς τον προορισμό του γίνεται μέσω της προώθησης του ερωτήματος μέσω των γειτονικών κόμβων. Στο CAN οι πίνακες δρομολόγησης κάθε κόμβου αποθηκεύουν μέχρι δ ($O(\delta)$) γείτονες και το κόστος της αναζήτησης είναι $O(\delta\ N^{1/\delta})$, όπου N ο συνολικός αριθμός των κόμβων στο σύστημα.

2.3.2.2 Tapestry

Το δίκτυο δομημένης επικάλυψης του Tapestry [27] παρέχει στους κόμβους του συστήματος μεθόδους εύρεσης αντικειμένων που είναι αποθηκευμένα στο σύστημα. Ο σχεδιασμός του στηρίζεται στην δενδρική δομή του Plaxton [39]. Σε αυτήν την δομή η δρομολόγηση (η εύρεση των αντικειμένων) προς τον κόμβο που είναι αποθηκευμένο κάποιο αντικείμενο γίνεται ακολουθώντας πάντοτε κόμβους των οποίων τα αναγνωριστικά έχουν κοινό πρόθεμα με το αναγνωριστικό του αντικειμένου. Τα αντικείμενα αποθηκεύονται στους κόμβους με τους οποίους έχουν κοινά προθέματα στα αναγνωριστικά τους. Ο κόμβος ο οποίος έχει το μεγαλύτερο κοινό πρόθεμα με ένα αντικείμενο ονομάζεται *ρίζα*.

Κάθε κόμβος διατηρεί έναν πίνακα δρομολόγησης και μια λίστα δεικτών που περιέχει τοποθεσίες αντιγράφων των αντικειμένων. Η λίστα δεικτών ανανεώνεται κάθε φορά που εισάγεται ή διαγράφεται ένα αντικείμενο. Όταν εισάγεται ένα αντικείμενο Α στο σύστημα ανανεώνονται όλες οι λίστες δεικτών των κόμβων που έχουν κοινό πρόθεμα (αναγνωριστικού) με το Α.

Η δενδρική δομή του Plaxton *θεωρεί* ότι το σύνολο των κόμβων είναι στατικό, δηλαδή ότι δεν υπάρχουν συχνές εισαγωγές και αναχωρήσεις κόμβων. Αυτό βέβαια δεν ισχύει στα περιβάλλοντα των ομότιμων κόμβων. Το Tapestry περιλαμβάνει μηχανισμούς μέσω των οποίων παρέχει στους κόμβους *ανοχή σε σφάλματα* (fault tolerance). Συγκεκριμένα κάθε αντικείμενο έχει πολλαπλές *ρίζες*, οι δείκτες στα αντικείμενα δεν είναι μόνιμοι άλλα λήγουν μετά από κάποιο χρονικό διάστημα και ανανεώνονται περιοδικά. Αυτό σημαίνει θεωρητικά ότι δεν υπάρχουν δείκτες σε αντικείμενα που δεν υπάρχουν.

2.3.2.3 Kademlia

Το Kademlia [28] είναι το πιο δημοφιλές σύστημα και χρησιμοποιείται από πολλές εφαρμογές διαμοίρασης αρχείων [14, 13, 6]. Κόμβοι και δεδομένα (κλειδιά των δεδομένων) αντιστοιχίζονται σε αναγνωριστικά, τα οποία αναπαρίστανται ως δυαδικές λέξεις μήκους b bit ($b = 160$). Τα αντικείμενα στο Kademlia αποθηκεύονται σε ζεύγη κλειδί-αντικείμενο σε κατάλληλους κόμβους. Ένα αντικείμενο με κλειδί K αποθηκεύεται στον κόμβο του οποίου το αναγνωριστικό έχει την μικρότερη *απόσταση* από το αναγνωριστικό του K. Η απόσταση μεταξύ των αναγνωριστικών καθορίζεται μέσω της συνάρτησης XOR. Στο σύστημα οι κόμβοι αντιστοιχούν στα φύλλα ενός δυαδικού δένδρου ύψους b. Η θέση κάθε κόμβου στο δένδρο καθορίζεται από το ελάχιστο κοινό πρόθεμα του αναγνωριστικού του. Για να επικοινωνούν οι κόμβοι μεταξύ τους πρέπει κάθε κόμβος να γνωρίζει κάποιους άλλους κόμβους που βρίσκονται σε διαφορετικά υποδένδρα από το δικό του.

Κάθε κόμβος N διατηρεί έναν πίνακα δρομολόγησης που περιέχει b εγγραφές. Οι εγγραφές αυτές ονομάζονται k-bucket, όπου k είναι η παράμετρος του συστήματος που καθορίζει το βαθμό του πλεονασμού της πληροφορίας που θα υπάρχει στους πίνακες δρομολόγησης. Συγκεκριμένα η εγγραφή i, με $0 \leq i \leq b$, περιέχει μια λίστα με k κόμβους οι οποίοι απέχουν από τον κόμβο N απόσταση d, $2^i \leq d \leq 2^{i+1}$. Οι κόμβοι στο k-bucket είναι διατεταγμένοι σύμφωνα με την τελευταία φορά που επικοινώνησε ο N μαζί τους. Οι κόμβοι με τους οποίους επικοινώνησε πιο πρόσφατα ο N βρίσκονται στο τέλος του k-bucket.

Το ζεύγος (κλειδί,αντικείμενο) αποθηκεύεται στους k κόμβους του συστήματος που έχουν την μικρότερη απόσταση από το αναγνωριστικό του κλειδιού. Για την εύρεση

ενός αντικειμένου, ένας κόμβος εντοπίζει τους k πλησιέστερους κόμβους από το κλειδί του αντικειμένου. Η διαδικασία αυτή είναι αναδρομική. Ο αρχικός κόμβος που ξεκινά την διαδικασία εύρεσης βρίσκει στον πίνακα δρομολόγησης του το k-bucket που αντιστοιχεί στην μικρότερη απόσταση (από το αναγνωριστικό) και προωθεί το ερώτημα σε a από συνολικούς k κόμβους. Οι κόμβοι αυτοί με την σειρά τους επιστρέφουν στον αρχικό κόμβο το πολύ k κόμβους που είναι πλησιέστερα στον προορισμό. Ο αρχικός κόμβος επιλέγει κάποιους από αυτούς και τους προωθεί το ερώτημα. Η διαδικασία τερματίζει όταν βρεθεί ο κόμβος που περιέχει το ζητούμενο αντικείμενο.

2.3.2.4 Chord

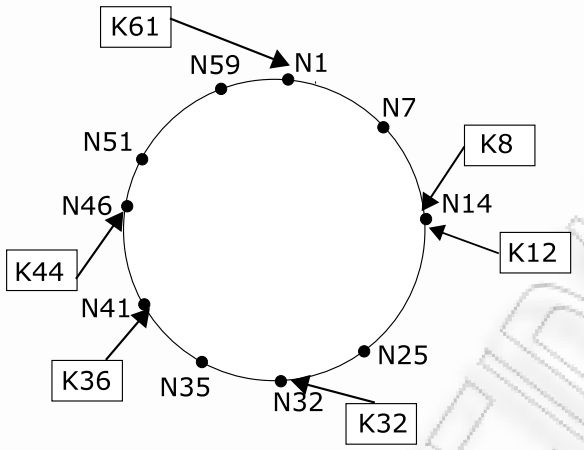
Το σύστημα Chord [11] αποτελεί τον βασικό εκπρόσωπο των συστημάτων DHT. Το πρωτόκολλο του Chord ορίζει ένα κοινό κυκλικό πεδίο τιμών για τα αναγνωριστικά των κόμβων και των αντικειμένων. Το εύρος του πεδίου τιμών καθορίζεται από την συνάρτηση κατακερματισμού H που χρησιμοποιεί το σύστημα για να παράγει τα αναγνωριστικά. Οι συγγραφείς του Chord προτείνουν την χρήση της συνάρτησης SHA-1 από την οποία προκύπτουν αναγνωριστικά μεγέθους 160 bit.

Οι κόμβοι και τα δεδομένα οργανώνονται στο κυκλικό πεδίο τιμών. Στο σχήμα 2.2 παρουσιάζεται το δίκτυο του Chord για αναγνωριστικά μεγέθους 6 (πεδίο τιμών $[0, 2^6 - 1)$). Για το διαχωρισμό των αναγνωριστικών των κόμβων και των κλειδιών χρησιμοποιείται το πρόθεμα N για τους κόμβους και το K για τα κλειδιά. Το Chord παρέχει μια βασική λειτουργία αναζήτησης μέσω της οποίας εντοπίζονται αναγνωριστικά[3].

Το σύστημα Chord προσφέρει αρκετές επιθυμητές ιδιότητες που απαιτούνται σε κατανεμημένα συστήματα που αποτελούνται από μεγάλο αριθμό κόμβων, με μεταβλητούς χρόνους παραμονής στο σύστημα. Συγκεκριμένα, για ένα σύστημα με N κόμβους η διαδικασία της αναζήτησης ενός αναγνωριστικού έχει κόστος[4] $O(logN)$ μηνυμάτων. Επίσης, κατά την εισαγωγή και αναχώρηση κάποιου κόμβου το ποσοστό των κλειδιών που πρέπει να μεταφερθούν (από τον κόμβο που φεύγει σε

[3] Η διαδικασία εύρεσης εφαρμόζεται και για τα αναγνωριστικά των κόμβων και για τα αναγνωριστικά των κλειδιών. Όταν αναφέρεται μέσα στο κείμενο εύρεση κόμβου ή εύρεση κλειδιού υπονοείται ότι αναζητείται κάποιο αναγνωριστικό που αντιστοιχεί σε κόμβο ή σε κλειδί (δεδομένων)

[4] Όλοι οι λογάριθμοι που χρησιμοποιούνται σε αυτήν την ενότητα έχουν για βάση το 2 είναι με βάση το 2, δηλ. $log_2 N$



Σχήμα 2.2: Επικάλυψη Chord, πεδίο τιμών $[0, 2^6 - 1]$

κάποιον άλλο) είναι της τάξης $O(1/N)$. Τέλος, για την ορθή και αποδοτική αναζήτηση αναγνωριστικών κάθε κόμβος του Chord διατηρεί πληροφορίες (δρομολόγησης) για $O(logN)$ άλλους κόμβους.

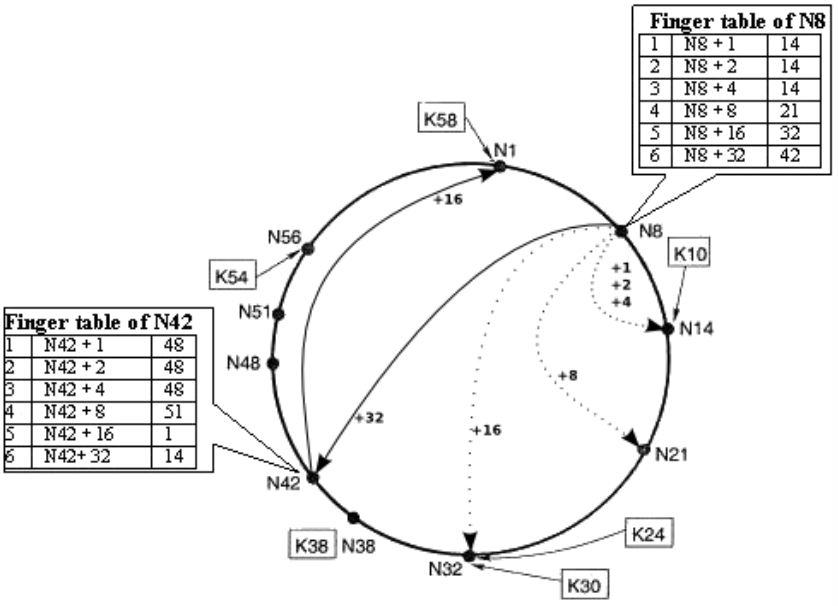
Το πρωτόκολλο του Chord ορίζει ότι τα αναγνωριστικά των κόμβων προκύπτουν από την συνάρτηση κατακερματισμού H δίνοντας ως είσοδο την διεύθυνση δικτύου τους και ενδεχομένως την θύρα λειτουργίας της σύνδεσης, δηλαδή $id = H(IP, port\ number)$. Τα αναγνωριστικά των κλειδιών προκύπτουν από την συνάρτηση κατακερματισμού με είσοδο το κλειδί. Τα αναγνωριστικά έχουν μήκος m bit, π.χ. αν $H = SHA - 1,\ m = 160$, και διατάσσονται σε έναν λογικό δακτύλιο (επικάλυψη) μήκους 2^m. Το πεδίο τιμών είναι κυκλικό και όλες οι αριθμητικές πράξεις μεταξύ των αναγνωριστικών είναι modulo 2^m.

Ένα κλειδί k αντιστοιχεί στον πρώτο κόμβο του δακτυλίου του οποίου το αναγνωριστικό είναι ίσο ή μεγαλύτερο από το αναγνωριστικό του κλειδιού. Ο κόμβος αυτός ονομάζεται ο *διαχειριστής* (successor) του κλειδιού, δηλαδή $succ(k)$. Στο σχήμα 2.2, για τα κλειδιά $K8, K12, K32, K36, K44, K61$ έχουμε
$succ(8) = 14,\ succ(12) = 14,\ succ(32) = 32,\ succ(36) = 41,\ succ(44) = 46,\ succ(61) = 1$.

Κάθε κόμβος στο σύστημα διατηρεί έναν πίνακα δρομολόγησης (finger table) που περιέχει m εγγραφές. Κάθε εγγραφή (finger) περιέχει το αναγνωριστικό, την διεύθυνση δικτύου και την θύρα λειτουργίας κάποιου κόμβου. Συγκεκριμένα, η i-οστή εγγραφή στον πίνακα δρομολόγησης του κόμβου x, περιέχει τον διαχειριστή του αναγνωριστικού $(x+2^{i-1}) mod\ 2^m$. Η διαδικασία της αναζήτησης ενός αναγνωριστικού

ξεκινά από τον τοπικό κόμβο, ο οποίος συμβουλεύεται τον πίνακα δρομολόγησης του και βρίσκει το πλησιέστερο αναγνωριστικό κόμβου. Επικοινωνεί με αυτόν τον κόμβο και η διαδικασία επαναλαμβάνεται μέχρι να βρεθεί ο διαχειριστής του αναγνωριστικού. Για παράδειγμα στο σχήμα 2.3, αν ο κόμβος $N8$ αναζητά το κλειδί $K58$, οδηγείται από τον πίνακα δρομολόγησή του στον κόμβο $N42$, ο οποίος με την σειρά του προωθεί το ερώτημα στον κόμβο $N1$ που είναι και ο διαχειριστής του κλειδιού $K58$.

Εκτός από τον αλγόριθμο της αναζήτησης, στο Chord ορίζεται ο τρόπος με τον οποίο καινούριοι κόμβοι εισάγονται στο σύστημα και ένας αλγόριθμος σταθεροποίησης ώστε οι κόμβοι του συστήματος να ενημερώνονται για τις συχνές αλλαγές στην κατάσταση του συστήματος.



Σχήμα 2.3: Διαδικασία αναζήτησης στο πρωτόκολλο Chord

2.4 Ανακεφαλαίωση

Τα δίκτυα ομότιμων κόμβων αποτελούν ένα σχετικά καινούριο μοντέλο για το σχεδιασμό και την οργάνωση κατανεμημένων συστημάτων. Τα δεδομένα και οι λειτουργίες της εφαρμογής, που χρησιμοποιεί υποδομές δικτύων ομότιμων κόμβων, κατανέμονται σε τυπικά υπολογιστικά συστήματα που βρίσκονται σε διαφορετικές

γεωγραφικές τοποθεσίες και είναι προσβάσιμα μέσω του διαδικτύου. Τα κυριότερα χαρακτηριστικά τους είναι ο μεγάλος πληθυσμός των κόμβων και το γεγονός ότι αυτοί αυτοί οι κόμβοι δεν είναι αξιόπιστοι. Ο σχεδιασμός των ομότιμων δικτύων λαμβάνει υπόψιν του το γεγονός της αξιοπιστίας των κόμβων και παρέχει τεχνικές (π.χ. τεχνικές πλεονασμού σε επίπεδο δεδομένων και αποθήκευσης) που εγγυούνται την ορθή λειτουργία τους.

Επιπλέον, η τρίτη γενιά των δικτύων δομημένης επικάλυψης παρέχει εγγυημένη απόδοση καθώς το κόστος λειτουργίας (σε αριθμό μηνυμάτων) και το πλήθος των συνδέσεων που έχουν οι κόμβοι του συστήματος δεν αυξάνεται γραμμικά σε σχέση με τον συνολικό αριθμό των κόμβων του συστήματος. Στα περισσότερα δίκτυα δομημένης επικάλυψης η εξάρτηση αυτή είναι λογαριθμική. Το γεγονός αυτό καθιστά τα συστήματα επεκτάσιμα, και για τον λόγο αυτό μπορεί να χρησιμοποιηθούν για την κατασκευή εφαρμογών μεγάλης κλίμακας που περιλαμβάνουν θεωρητικά απεριόριστο αριθμό υπολογιστικών συστημάτων. Για παράδειγμα ο αριθμός των υπολογιστικών συστημάτων στο διαδίκτυο είναι $\approx 6.8x10^8$. Εφόσον, στα συστήματα DHT το κόστος της αναζήτησης είναι λογαριθμικό σε σχέση με τον αριθμό των κόμβων του συστήματος, τότε η αναζήτηση περιλαμβάνει την *επίσκεψη* $log(6.81x10^8) = 29$ κόμβων (χωρίς να χρησιμοποιηθούν τεχνικές βελτιστοποίησης όπως caching). Επίσης, λόγω της οργάνωσης των δεδομένων στην επικάλυψη η διαδικασία της αναζήτησης εγγυάται ότι αν υπάρχει διαθέσιμο κάποιο δεδομένο στο σύστημα τότε αυτό θα βρεθεί (π.χ. η διαδικασία αναζήτησης του Chord επιστρέφει πάντοτε το διαχειριστή του αναγνωριστικού).

Λόγω της εγγυημένης απόδοσής τους τα δίκτυα ομότιμων κόμβων αποτελούν ελκυστική λύση για την δημιουργία εφαρμογών μεγάλης κλίμακας. Οι απαιτήσεις αναζήτησης που έχουν οι εφαρμογές αυτές ποικίλουν ανάλογα με το είδος τους. Έτσι έχουν δημιουργηθεί εναλλακτικές μέθοδοι εύρεσης για δίκτυα δομημένης επικάλυψης. Στο κεφάλαιο 3 αναλύονται οι απαιτήσεις αναζήτησης διαφόρων εφαρμογών και παρουσιάζονται τρόποι μέσω των οποίων υλοποιούνται σε δίκτυα δομημένης επικάλυψης. Τέλος, άλλος ένας σημαντικός παράγοντας που αφορά τις εφαρμογές που χρησιμοποιούν τις διαδικασίες αναζήτησης των δικτύων δομημένης επικάλυψης είναι η ασφάλεια. Στο κεφάλαιο 4 αναφέρονται οι αδυναμίες στην ασφάλεια των δικτύων επικάλυψης και λύσεις που έχουν προταθεί στην βιβλιογραφία. Επίσης αναλύεται το πρόβλημα της *ταυτοποίησης* οντοτήτων στα δίκτυα ομότιμων

κόμβων και προτείνεται ένας κατανεμημένος μηχανισμός πιστοποίησης [12] που βασίζεται στο δίκτυο δομημένης επικάλυψης του Chord.

Κεφάλαιο 3

Μέθοδοι εύρεσης πληροφοριών σε δίκτυα δομημένης επικάλυψης

Περίληψη

Στο κεφάλαιο αυτό αναφέρονται οι βασικές απαιτήσεις που έχει ένα κατανεμημένο σύστημα εύρεσης πληροφοριών. Αναλύεται το πρόβλημα της εύρεσης πληροφοριών σε δίκτυα δομημένης επικάλυψης και αναφέρονται οι απαιτήσεις εύρεσης πόρων με βάση κριτήρια που αφορούν πολλαπλά χαρακτηριστικά των πόρων. Επιπλέον, περιγράφονται οι αδυναμίες υποστήριξης ερωτημάτων εύρους σε συστήματα δομημένης επικάλυψης και ταξινομούνται λύσεις που έχουν δοθεί για το συγκεκριμένο πρόβλημα. Στην συνέχεια, προτείνεται μια λύση, το δυαδικό δένδρο αντιγράφων, μέσω της οποίας υποστηρίζονται ερωτήματα εύρους σε δίκτυα δομημένης επικάλυψης αυξάνοντας ταυτόχρονα την διαθεσιμότητα των πόρων μέσω της αντιγραφής τους σε πολλαπλούς κόμβους. Τέλος, η απόδοση του δυαδικού δένδρου αντιγράφων αξιολογείται βάσει πειραμάτων στα οποία αναλύεται πώς επηρεάζεται το κόστος αναζήτησης, μετρούμενο σε αριθμό μηνυμάτων που πρέπει να μεταδοθούν στο δίκτυο, σε σχέση με τον αριθμό των πόρων που υπάρχουν στο σύστημα και με τις δυνατότητες των κόμβων που διαχειρίζονται τους πόρους.

3.1 Εισαγωγή

Οι έντονες εξελίξεις της τελευταίας δεκαετίας στην θεματική περιοχή των κατανεμημένων συστημάτων, ιδίως των συστημάτων ομότιμων κόμβων, σε επίπεδο έρευνας όσο και σε επίπεδο υλοποίησης συστημάτων δημιούργησε το πλαίσιο για τη λειτουργία συστημάτων μεγάλης κλίμακας. Οι δυνατότητες των συστημάτων αυτών είναι τεράστιες αν ληφθούν υπόψιν (αθροιστικά) οι υπολογιστικές δυνατότητες του μεγάλου αριθμού των υποσυστημάτων που τα απαρτίζουν. Βέβαια, η παροχή αξιόπιστων υπηρεσιών στο δυναμικό περιβάλλον των συστημάτων αυτών, στα οποία κόμβοι εισάγονται, αναχωρούν και αποτυγχάνουν, παρουσιάζει προκλήσεις. Παράλληλα, τεχνολογίες όπως οι υπηρεσίες υπολογιστικού πλέγματος [8] προσφέρουν την συσσωρευμένη επεξεργαστική ισχύ πολλαπλών συστημάτων τα οποία βρίσκονται διάσπαρτα σε διαφορετικές γεωγραφικά και διαχειριστικά περιοχές, για την επίλυση (κυρίως επιστημονικών) υπολογιστικά χρονοβόρων προβλημάτων. Από αυτήν την άποψη οι τεχνολογίες πλέγματος και συστημάτων ομότιμων κόμβων έχουν αρκετά κοινά χαρακτηριστικά οδηγώντας στην σύγκλιση αυτών των τεχνολογιών [41].

Ένα από τα μεγαλύτερα προβλήματα που αντιμετωπίζουν οι δύο παραπάνω τεχνολογίες είναι η εύρεση πληροφοριών (όπως αρχείων στα δίκτυα ομότιμων κόμβων και υπολογιστικών πόρων όπως επεξεργαστική ισχύ, αποθηκευτικό χώρο και υπηρεσίες πλέγματος στην περίπτωση των συστημάτων πλέγματος). Το παρόν κεφάλαιο ασχολείται με αυτό το πρόβλημα εύρεσης δίνοντας ιδιαίτερη έμφαση στα ιδιαίτερα χαρακτηριστικά της πλατφόρμας των δικτύων ομότιμων κόμβων, για την οποία και αντιμετωπίζεται το πρόβλημα. Συγκεκριμένα, οι εγγυήσεις που παρέχουν ως προς την εύρεση δεδομένων, η επεκτασιμότητά τους σε μεγάλο αριθμό κόμβων διατηρώντας το δικτυακό κόστος σε χαμηλά επίπεδα και οι σχεδιαστικές απαιτήσεις που έχουν ως προς την συμμετρία των κόμβων (τόσο στην επικοινωνία όσο και οποιαδήποτε άλλη λειτουργία παρέχεται) πρέπει να μην παραβιάζονται από τους προτεινόμενους μηχανισμούς εύρεσης.

Οι λειτουργίες της εύρεσης εξαρτώνται σε μεγάλο βαθμό από την *οργάνωση* της πληροφορίας και κατ᾽επέκταση από την οργάνωση των κόμβων στους οποίους αποθηκεύεται η πληροφορία. Τα πρωτόκολλα της προηγούμενης γενιάς δικτύων ομότιμων κόμβων [19] που οδήγησαν στην δημιουργία συστημάτων διαμοίρασης

περιεχομένου, όπως τα Gnutella [5] και Kazaa [42], παρέχουν εκτεταμένες δυνατότητες εύρεσης μέσα από την δημιουργία εκφραστικών ερωτημάτων αλλά παρουσιάζουν τις παρακάτω αδυναμίες:

- Δεν παρέχουν εγγυήσεις στην εύρεση των αποτελεσμάτων ακόμη και αν η πληροφορία είναι διαθέσιμη σε κάποιον κόμβο,
- Οι τεχνικές εύρεσης πληροφοριών που χρησιμοποιούν, (τυχαίος περίπατος και πλημμύρα (βλ. 2.2.1), δημιουργούν μεγάλη (και πιθανόν περιττή) κίνηση στο δίκτυο, χωρίς να υπάρχουν εγγυήσεις για την εύρεση της πληροφορίας.

Οι παραπάνω λόγοι οδήγησαν στην δημιουργία της τρέχουσας γενιάς συστημάτων που βασίζονται σε δίκτυα δομημένης επικάλυψης για τα οποία και αναλύεται το πρόβλημα της εύρεσης σε αυτό το κεφάλαιο. Συγκεκριμένα στα δίκτυα δομημένης επικάλυψης [11, 26, 43, 27] οι κόμβοι και τα δεδομένα αποκτούν αναγνωριστικά σε ένα κοινό πεδίο τιμών βάσει των οποίων *διατάσσονται* στην λογική τοπολογία (δακτύλιος, υπερκύβος) του δικτύου επικάλυψης. Ο τρόπος εύρεσης των κόμβων και των δεδομένων για τα οποία είναι υπεύθυνοι βασίζεται σε προκαθορισμένους κανόνες που εκμεταλλεύονται κάποιο γεωμετρικό χαρακτηριστικό της τοπολογίας. Με αυτόν τον τρόπο παρέχονται εγγυήσεις ότι αν κάποιος κόμβος είναι διαθέσιμος τότε αυτός βρίσκεται σε προκαθορισμένο σημείο στο δίκτυο. Επιπλέον, οι δυνατότητες επεκτασιμότητας και ανοχής σε σφάλματα που παρέχονται στα συστήματα δομημένης επικάλυψης αποτελούν έναν ακόμη λόγο για τον οποίο μπορεί να χρησιμοποιηθούν ως βασική πλατφόρμα για την υλοποίηση κατανεμημένων συστημάτων μεγάλης κλίμακας. Βέβαια, η εγγυημένη δυνατότητα εύρεσης που προσφέρεται μέσω της λειτουργίας `get` δεν ικανοποιεί τις απαιτήσεις εύρεσης.

Σε συστήματα εύρεσης πληροφοριών όπως μηχανές αναζήτησης [1], συστήματα εύρεσης περιεχομένου, διαμοίρασης αρχείων [6, 13, 5], συστήματα εύρεσης υπολογιστικών πόρων [44] και γενικότερα οποιασδήποτε εφαρμογής επιθυμεί λειτουργίες εύρεσης στις οποίες ο χρήστης δεν γνωρίζει εκ των προτέρων (το κλειδί) τι ακριβώς ψάχνει. Για παράδειγμα ερωτήματα προς μία ψηφιακή βιβλιοθήκη αναζήτησης επιστημονικών εργασιών (π.χ. citeseer [45]) μπορεί να είναι:

Εύρεση εργασιών του συγγραφέα με επώνυμο Παπαδόπουλος που έχουν δημοσιευτεί μεταξύ 2004 και 2007.

Αντίστοιχο ερώτημα σε έναν μηχανισμό εύρεσης υπολογιστικών πόρων [44] είναι της μορφής:

Arch=='SGI' Opsys=='IRIX6' Memory >= 1GB

με το οποίο αναζητούνται όλα τα μηχανήματα αρχιτεκτονικής SGI με λειτουργικό σύστημα IRIX6 και με μνήμη RAM μεγαλύτερη από 1GB.

Αντίστοιχα, σε ένα σύστημα διαμοίρασης μουσικών αρχείων ένα τυπικό ερώτημα είναι της μορφής:

Βρες μουσικά αρχεία με τίτλο δίσκου 'Master of Reality' και ποιότητα (bps) μεγαλύτερη από 192Kb/sec

Ανεξάρτητα από την φύση του μηχανισμού αναζήτησης (εύρεση μουσικών αρχείων ή εύρεση υπολογιστικών πόρων), τα ερωτήματα *υποθέτουν* ότι οι πόροι (αρχεία, υπολογιστές, κ.α.) έχουν κάποια χαρακτηριστικά (attributes), βάσει των οποίων ζητείται ένα σύνολο πόρων ώστε ένα χαρακτηριστικό να έχει κάποια συγκεκριμένη τιμή ή/και κάποιο χαρακτηριστικό να έχει κάποιο εύρος τιμών.

Στην επόμενη ενότητα περιγράφονται τεχνικές με τις οποίες υποστηρίζονται οι παραπάνω υπηρεσίες εύρεσης σε δίκτυα δομημένης επικάλυψης.

3.2 Εύρεση σε συστήματα δικτύων δομημένης επικάλυψης

Τα δίκτυα δομημένης επικάλυψης παρέχουν την μέθοδο put μέσω της οποίας εγγράφονται στο σύστημα ζεύγη χαρακτηριστικού και τιμών (attribute-value pairs)

$\langle a, v\rangle$, όπου το a αντιστοιχεί συνήθως σε μια λέξη κλειδί που περιγράφει κάποιον πόρο και το v σε έναν δείκτη (π.χ. ένα URL [46]) προς τον πόρο. Εργασίες στην ερευνητική περιοχή των συστημάτων υπολογιστικού πλέγματος [44, 47, 48, 49] όπως και εργασίες στην περιοχή των *συστημάτων διανομής περιεχομένου* [50, 51, 52], που βασίζονται σε δίκτυα ομότιμων κόμβων ασχολούνται με την οργάνωση των *λέξεων κλειδιών* σε ευρετήριο το οποίο είναι κατανεμημένο μεταξύ των κόμβων του δικτύου επικάλυψης. Η *κατανομή* του ευρετηρίου εξαρτάται από τις λέξεις κλειδιά. Συνήθως χρησιμοποιείται η συνάρτηση κατακερματισμού H του δικτύου δομημένης επικάλυψης για να προσδιοριστεί ο κόμβος στον οποίο αντιστοιχεί η λέξη (για την ακρίβεια το ζεύγος $\langle a, v\rangle$). Συγκεκριμένα, αν $H(a) \rightarrow id$, τότε το ζεύγος $\langle a, v\rangle$ θα αποθηκευτεί στον κόμβο του δικτύου δομημένης επικάλυψης που είναι υπεύθυνος για το αναγνωριστικό id.

Τα συστήματα εύρεσης πρέπει να υποστηρίζουν πολλαπλά χαρακτηριστικά a_1, a_2, $\cdots$, a_n που χαρακτηρίζουν έναν πόρο και κατ' επέκταση ορίζουν τα κριτήρια αναζήτησης για αυτόν τον πόρο. Αυτό υλοποιείται μέσω δύο εναλλακτικών τεχνικών: είτε ορίζονται πολλαπλά δίκτυα επικάλυψης (ανά χαρακτηριστικό) [53], είτε το πεδίο τιμών των n χαρακτηριστικών μετατρέπεται σε ένα μονοδιάστατο πεδίο τιμών μέσω ειδικών τεχνικών, όπως η Hilbert SFC [44, 49], ή μέσω κάποιας ειδικής συνάρτησης κατακερματισμού [48]. Οι τεχνικές αυτές μπορεί να συνδυαστούν με τις τεχνικές που περιγράφονται στην ενότητα 3.3 ώστε οι μηχανισμοί εύρεσης να υποστηρίζουν ερωτήματα για πόρους βάσει πολλών χαρακτηριστικών, οι τιμές των οποίων ανήκουν σε κάποιο εύρος.

3.3 Ερωτήματα εύρους σε δίκτυα δομημένης επικάλυψης

Τα ερωτήματα εύρους αφορούν την εύρεση πόρων που αντιστοιχούν σε ένα χαρακτηριστικό A με πεδίο τιμών D_A, για κάποιο υποσύνολο τιμών (εύρος) του D_A. Το βασικό πρόβλημα στα δίκτυα δομημένης επικάλυψης στα οποία οι πόροι εγγράφονται στο σύστημα με βάση ζεύγη $\langle a, v\rangle$, είναι ότι η κλασσική συνάρτηση κατακερματισμού που χρησιμοποιείται για την μετατροπή στο κοινό πεδίο τιμών των αναγνωριστικών στο δίκτυο επικάλυψης (π.χ [11]) *καταστρέφει* οποιαδήποτε σχέση διάταξης υπήρχε μεταξύ των τιμών του χαρακτηριστικού $a \in D_A$.

Στην σχετική βιβλιογραφία έχουν προταθεί αρκετές λύσεις που επιχειρούν να λύσουν το παραπάνω πρόβλημα. Μία κατηγορία λύσεων βασίζεται σε κάποιες ιδιότητες που προσφέρει το δίκτυο δομημένης επικάλυψης [26, 35, 33]. Αλλά με αυτόν τον τρόπο η εφαρμογή της λύσης περιορίζεται σε συγκεκριμένα συστήματα.

Μία άλλη κατηγορία λύσεων [53, 54] δεν χρησιμοποιεί την συνάρτηση κατακερματισμού για να αντιστοιχιστούν τα δεδομένα στο κοινό πεδίο τιμών των αναγνωριστικών στο δίκτυο επικάλυψης. Αντίθετα, χρησιμοποιεί το αρχικό πεδίο τιμών των δεδομένων. Με αυτόν τον τρόπο όμως δημιουργούνται προβλήματα υπερφόρτωσης κάποιων κόμβων του δικτύου επικάλυψης. Αν η κατανομή των δεδομένων είναι πιο *πυκνή* σε κάποιο διάστημα του πεδίου ορισμού της, τότε ορισμένοι κόμβοι θα διαχειρίζονται περισσότερα δεδομένα. Το ίδιο πρόβλημα παρατηρείται και στις λύσεις [55, 44, 48, 49] που αντικαθιστούν την συνάρτηση κατακερματισμού με μια συνάρτηση κατακερματισμού H_l [56] που όμως διατηρεί οποιαδήποτε σχέση διάταξης προϋπήρχε μεταξύ των τιμών του χαρακτηριστικού a (δηλαδή αν $a_1 < a_2 \Rightarrow H_l(a_1) < H_l(a_2)$). Με αυτόν τον τρόπο η αναζήτηση ενός χαρακτηριστικού a σε εύρος τιμών μεταξύ $l \leq a \leq h, \quad l, a, h \in D_A$, *μεταφέρεται* στο κοινό πεδίο αναγνωριστικών του δικτύου επικάλυψης. Λόγω της υπερφόρτωσης των κόμβων οι συγκεκριμένες λύσεις προτείνουν επιπλέον αλγορίθμους εξισορρόπησης του φόρτου εργασίας για τους κόμβους.

Πιστεύουμε ότι μία γενική λύση δεν πρέπει να βασίζεται σε κάποιο συγκεκριμένο πρωτόκολλο δικτύου δομημένης επικάλυψης, ούτε να τροποποιεί τον τρόπο λειτουργίας τους καθώς δεν μπορεί να εφαρμοστεί σε συστήματα επικάλυψης που ήδη λειτουργούν. Αντίθετα, η γενική λύση πρέπει να βασίζεται στα κοινά χαρακτηριστικά που υπάρχουν σε όλα τα δίκτυα επικάλυψης, δηλαδή στις μεθόδους get και put, χωρίς να δημιουργεί επιπλέον εξαρτήσεις στο σύστημα. Ένας τρόπος με τον οποίο μπορεί να επιτευχθεί αυτό είναι με την ιεραρχική οργάνωση των δεδομένων σε μια δενδρική δομή δεδομένων.

Ο Chawathe [57] προτείνει την οργάνωση των τιμών ενός χαρακτηριστικού a σε ένα δένδρο που βασίζεται στο κοινό πρόθεμα των (δυαδικών) τιμών του a, το Prefix Hash Tree – PHT. Κάθε κόμβος του PHT μπορεί να αποθηκεύσει συγκεκριμένο αριθμό από ζεύγη $\langle a, v \rangle$. Τα ζεύγη αποθηκεύονται μόνο στους κόμβους - φύλλα του PHT. Αν κατά την διαδικασία εισαγωγής σε κάποιο κόμβο-φύλλο ο αριθμός

από ζεύγη είναι στο μέγιστο όριο τότε το συγκεκριμένο φύλλο θα διασπαστεί σε δύο νέους κόμβους και τα ζεύγη του θα μοιραστούν μεταξύ αυτών των κόμβων. Επιπλέον, η διαγραφή κλειδιών από κάποιον κόμβο μπορεί να προκαλέσει την συγχώνευση δύο κόμβων. Η εύρεση ενός ερωτήματος εύρους επιτυγχάνεται υπολογίζοντας το κοινό πρόθεμα των ορίων του ερωτήματος εύρους. Στην συνέχεια, εντοπίζεται ο κόμβος του δικτύου επικάλυψης που αντιστοιχεί στον κόμβο του δένδρου με το κοινό πρόθεμα. Η απόδοση της συγκεκριμένης μεθόδου εξαρτάται σε μεγάλο βαθμό από τον τρόπο εισαγωγής των δεδομένων στο σύστημα. Συγκεκριμένα, ορισμένες κατανομές των δεδομένων οδηγούν στην διαδοχική διάσπαση κάποιου κόμβου με άμεσο αποτέλεσμα την μεταφορά των δεδομένων του στους άλλους κόμβους του δικτύου. Επιπλέον, λόγω του δυναμικού τρόπου ανάπτυξης του δένδρου δημιουργούνται ζητήματα συγχρονισμού (π.χ. δύο διαδοχικές αιτήσεις εισαγωγής προς έναν κόμβο-φύλλο με μέγιστο αριθμό ζευγών).

Επίσης, ο Gao [50] προτείνει έναν αλγόριθμο με τον οποίο υποστηρίζονται ερωτήματα εύρους μέσω ενός δυαδικού δένδρου, του Range Search Tree – RST. Τα ζεύγη $\langle a, v\rangle$ αποθηκεύονται στους κόμβους του RST. Το ερώτημα εύρους διασπάται σε υποερωτήματα για μικρότερα εύρη τιμών ώστε να διαμοιραστεί ο φόρτος του ερωτήματος μεταξύ των κόμβων του δικτύου. Επίσης, προτείνεται ένας πίνακας εξισορρόπισης φόρτου (Load Balancing Matrix) με τον οποίο αντιμετωπίζεται το πρόβλημα ετεροκλινών κατανομών (skew distribution) δεδομένων. Ο συγκεκριμένος μηχανισμός εξασφαλίζει εξισορρόπηση στον φόρτο των κόμβων του συστήματος έναντι ερωτημάτων.

Τέλος, το Distributed Segment Tree – DST [58] είναι ένα δυαδικό δένδρο στο οποίο αποθηκεύονται αντίγραφα από τα ζεύγη $\langle a, v\rangle$ σε όλους τους κόμβους του δένδρου που περιλαμβάνουν το εύρος του a. Συγκεκριμένα, τα ζεύγη $\langle a, v\rangle$ αποθηκεύονται σε κάθε κόμβο μεταξύ του κόμβου φύλλου και της ρίζας του δυαδικού δένδρου. Ολόκληρο το πεδίο τιμών D_a διαμοιράζεται μεταξύ των κόμβων του DST. Για να περιοριστεί ο αριθμός των αντιγράφων που αποθηκεύονται στους υψηλότερους κόμβους (κοντά στο κόμβο ρίζα) του DST προτείνεται ένα κοινό μέγιστο όριο για τον αριθμό αντιγράφων που αποθηκεύονται σε κάθε κόμβο (ανεξαρτήτως επιπέδου).

Όλες οι παραπάνω εργασίες βασίζονται στην ιεραρχική οργάνωση ζευγών $\langle a, v\rangle$ ή αντιγράφων τους σε ένα δυαδικό δένδρο το οποίο είτε έχει σταθερό ύψος (RST,

DST) είτε μεταβλητό ύψος (PHT) υποστηρίζοντας με αυτόν τον τρόπο χαρακτηριστικά με μεταβλητό πεδίο τιμών. Σε κάθε περίπτωση οι κόμβοι του *λογικού* δυαδικού δένδρου αποθηκεύονται στους *φυσικούς* κόμβους του δικτύου δομημένης επικάλυψης, οι οποίοι πρέπει να παρέχουν απαντήσεις σε ερωτήματα σε εύρη τιμών που τους αντιστοιχούν. Πρωταρχικό στόχο αποτελεί ο ομοιογενής καταμερισμός των ερωτημάτων σε όλους τους κόμβους του δικτύου επικάλυψης. Στο PHT ορίζεται ένα κατώφλι B που καθορίζει τον μέγιστο αριθμό από ζεύγη $\langle a, v\rangle$ που αποθηκεύονται σε κάθε κόμβο. Μια αίτηση εισαγωγής σε έναν κόμβο του PHT με B ζεύγη προκαλεί την διάσπασή του, και ακολούθως την αντιστοίχηση των απογόνων του, και κατ'επέκταση την μεταφορά των ζευγών $\langle a, v\rangle$ που του αντιστοιχούν, σε δύο νέους φυσικούς κόμβους του δικτύου επικάλυψης. Με παρόμοιο τρόπο στο DST περιορίζεται ο αριθμός των αντιγράφων στο DST. Όλες αυτές οι τεχνικές θέτουν ένα προκαθορισμένο όριο για το σύστημα, και επιπλέον, το όριο τίθεται στο επίπεδο του δυαδικού δένδρου.

Στην επόμενη ενότητα προτείνεται ένας μηχανισμός, μέσω του οποίου υποστηρίζονται ερωτήματα εύρους για ένα χαρακτηριστικό a, ο οποίος λαμβάνει υπόψιν τις πραγματικές δυνατότητες των φυσικών κόμβων του δικτύου δομημένης επικάλυψης και προτείνει έναν αλγόριθμο μέσω του οποίου ορίζεται δυναμικά ο αριθμός των αντιγράφων στους κόμβους του δυαδικού δένδρου αντιγράφων.

3.4 Δυαδικό δέντρο αντιγράφων

Το *δυαδικό δένδρο αντιγράφων* – ΔΔΑ, είναι ένας μηχανισμός μέσω του οποίου υποστηρίζονται ερωτήματα εύρους σε δίκτυα δομημένης επικάλυψης. Το ΔΔΑ προτείνει την ιεραρχική οργάνωση αντιγράφων των κλειδιών σε μια λογική δομή δεδομένων όμοια με ένα δυαδικό δένδρο αναζήτησης [59]. Τα ζεύγη κλειδιών-δεδομένων αποθηκεύονται σε προκαθορισμένους εξυπηρετητές[1] του δικτύου επικάλυψης. Η προτεινόμενη λύση είναι ανεξάρτητη από το δίκτυο επικάλυψης, καθώς η υλοποίησή της βασίζεται μόνο στις δύο βασικές λειτουργίες `get` και `put` των DHT. Το γεγονός αυτό επιτρέπει την εφαρμογή της λύσης σε υπάρχοντα συστήματα δομημένης

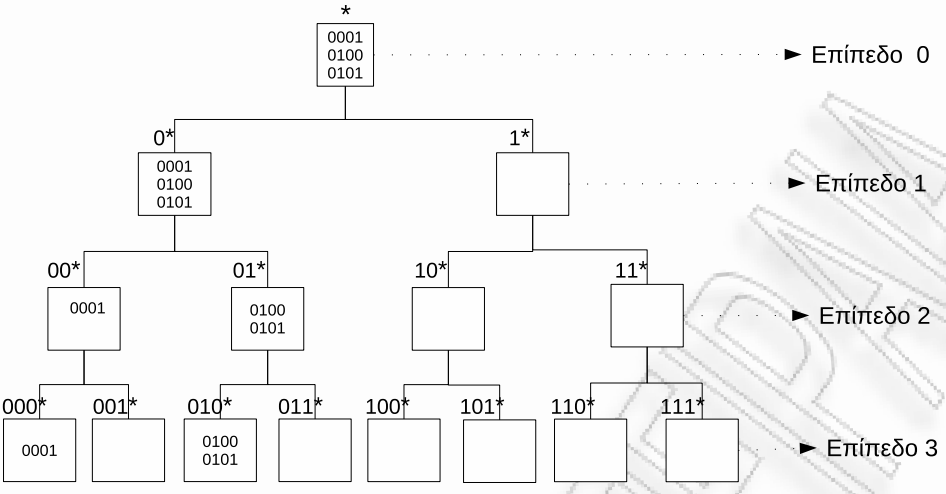
[1]Για τον διαχωρισμό των κόμβων του δικτύου δομημένης επικάλυψης και των κόμβων του ΔΔΑ χρησιμοποιούμε τον όρο εξυπηρετητής για την αναφορά σε κόμβους του δικτύου επικάλυψης.

επικάλυψης χωρίς να απαιτείται κάποια αλλαγή στους εσωτερικούς μηχανισμούς λειτουργίας τους (π.χ αλλαγή στην συνάρτηση κατακερματισμού).

Στις επόμενες ενότητες περιγράφεται λεπτομερώς η δομή του δυαδικού δένδρου αντιγράφων και ορίζεται η αντιστοιχία μεταξύ των *λογικών κόμβων* του ΔΔΑ και των φυσικών κόμβων του δικτύου δομημένης επικάλυψης. Στην συνέχεια ορίζονται οι διαδικασίες εισαγωγής και διαγραφής ζευγαριών κλειδιών - δεδομένων και η διαδικασία εύρεσης κλειδιών για κάποιο εύρος τιμών. Τέλος, προτείνεται ένας αλγόριθμος μέσω του οποίου μειώνεται ο φόρτος που υφίσταται κάποιος κόμβος που είναι υπεύθυνος για μεγάλο εύρος τιμών.

3.4.1 Περιγραφή της δομής δεδομένων

Το ΔΔΑ είναι ένα δυαδικό δένδρο ύψους $h = \lfloor log_2 M \rfloor + 1$. Κάθε επίπεδο $l \in [0, h-1]$ του ΔΔΑ περιέχει 2^l κόμβους καθένας από τους οποίους είναι υπεύθυνος για έναν ορισμένο αριθμό κλειδιών. Ο μέγιστος αριθμός αντιγράφων ενός κόμβου, N_l, που ανήκει στο επίπεδο l είναι 2^{h-l+1}. Το πεδίο τιμών των κλειδιών που αποθηκεύονται στους κόμβους του δένδρου είναι $[0, M], \mathrm{M} \in \mathbb{N}$. Κάθε κλειδί αναπαρίσταται ως μια δυαδική λέξη με μέγεθος h ψηφία. Κάθε κόμβος του ΔΔΑ χαρακτηρίζεται από ένα *όνομα*. Η ρίζα του δένδρου έχει όνομα $*$. Τα ονόματα των άλλων κόμβων παράγονται αναδρομικά. Συγκεκριμένα το αριστερό *παιδί* ενός κόμβου με όνομα $1*$ έχει όνομα $10*$, ενώ το όνομα του δεξιού παιδιού είναι $11*$. Δηλαδή στο όνομα του πατέρα κόμβου τοποθετείται το ψηφίο 0 (αριστερό παιδί) ή 1 (δεξί παιδί). Αν το όνομα του κόμβου αποτελεί πρόθεμα ενός κλειδιού K τότε ο αντίστοιχος κόμβος είναι *υπεύθυνος* κόμβος για το κλειδί K. Ο κόμβος ρίζα του ΔΔΑ είναι υπεύθυνος για όλα τα κλειδιά. Στο σχήμα 3.1 παρουσιάζεται ένα δυαδικό δένδρο αντιγράφων για κλειδιά με πεδίο τιμών $[0, 15]$. Η δυαδική αναπαράσταση των κλειδιών είναι δυαδικές λέξεις με 4 ψηφία. Για παράδειγμα υπεύθυνοι για το κλειδί 0001 είναι οι κόμβοι $*, 0*, 00*, 000*$. Η συγκεκριμένη ονοματολογία των κόμβων χρησιμοποιήθηκε για πρώτη φορά στο πλαίσιο των δικτύων δομημένης επικάλυψης στην εργασία Prefix Hash Tree [57].



Σχήμα 3.1: Δυαδικό δένδρο αντιγράφων με πεδίο τιμών κλειδιών $[0, 15]$

3.4.2 Κατανομή του δένδρου στο δίκτυο δομημένης επικάλυψης

Οι εξυπηρετητές του δικτύου δομημένης επικάλυψης *διαχειρίζονται* τους κόμβους και τα κλειδιά του ΔΔΑ. Στο πλαίσιο των δικτύων δομημένης επικάλυψης όλα τα *αναγνωριστικά* έχουν κοινό πεδίο τιμών που ορίζεται από την συνάρτηση κατακερματισμού H που χρησιμοποιείται (π.χ. $SHA-1$ στο Chord [11]). Το κοινό πεδίο τιμών των αναγνωριστικών των κλειδιών και των εξυπηρετητών είναι $[0, 2^m - 1]$, όπου m το πλήθος των ψηφίων της συνάρτησης H. Όπως αναφέρθηκε στο κεφάλαιο 2 ο τρόπος με τον οποίο αντιστοιχίζονται τα κλειδιά στους εξυπηρετητές του δικτύου επικάλυψης βασίζεται στην ελαχιστοποίηση της απόστασης μεταξύ των αναγνωριστικών των εξυπηρετητών και των κλειδιών. Αυτό προϋποθέτει ένα κοινό πεδίο αναγνωριστικών που εξασφαλίζεται μέσω της συνάρτησης H.

Με βάση τη συνάρτηση H, οι κόμβοι και τα κλειδιά του ΔΔΑ *μετατρέπονται* σε αναγνωριστικά στο επίπεδο του δικτύου επικάλυψης. Συγκεκριμένα στο παράδειγμα του σχήματος 3.1 οι κόμβοι $*,\ 0*,\ 1*,\ ...$ αντιστοιχούν στα αναγνωριστικά $H(*),\ H(0*),\ H(1*),\ ...$ αντίστοιχα. Για το δίκτυο του Chord, που αναλύθηκε στην ενότητα 2.3.2.4, οι εξυπηρετητές στους οποίους αντιστοιχούν τα αναγνωριστικά αυτά είναι $succ(H(*)),\ succ(H(0*)),\ succ(H(1*))$.

3.4.3 Διαδικασία εισαγωγής και διαγραφής αντιγράφων

Η διαδικασία εισαγωγής σε συστήματα που βασίζονται σε δίκτυα δομημένης επικάλυψης περιλαμβάνει μια αίτηση put με την οποία ένα ζεύγος κλειδιού-δεδομένου $\langle a, v\rangle$ αποθηκεύεται σε κάποιον εξυπηρετητή του συστήματος. Αν το σύστημα χρησιμοποιεί τεχνικές πλεονασμού τότε το ζεύγος $\langle a, v\rangle$ αποθηκεύεται και σε άλλους εξυπηρετητές. Για παράδειγμα, ο Dabek [60] προτείνει την αποθήκευση σε t συνεχόμενους εξυπηρετητές (ο πρώτος εξυπηρετητής είναι ο διαχειριστής του αναγνωριστικού που αντιστοιχεί στο a). Επίσης, στην ενότητα 4.6.2.1 αναφέρεται ένας τρόπος αποθήκευσης σε t εξυπηρετητές που βασίζεται σε έναν *προσυμφωνημένο* τρόπο παραγωγής των αναγνωριστικών στα οποία αντιστοιχίζονται οι εξυπηρετητές.

Παρόμοια μέθοδος ακολουθείται και κατά την διαδικασία εισαγωγής ζευγών $\langle a, v\rangle$ στο ΔΔΑ. Το σύνολο των κόμβων που ανήκουν στο *μονοπάτι* του ΔΔΑ, μεταξύ της ρίζας και του a ορίζουν το σύνολο των κόμβου που είναι υπεύθυνοι για το ζεύγος $\langle a, v\rangle$. Με βάση αυτό το σύνολο ορίζονται και οι εξυπηρετητές του δικτύου στους οποίους θα αποθηκευτούν τελικά τα ζεύγη. Η εισαγωγή ενός κλειδιού a αποτελείται από $h+1$ put λειτουργίες στο σύστημα δομημένης επικάλυψης. Μια λειτουργία για την αποθήκευση του πραγματικού κλειδιού και h λειτουργίες για την αποθήκευση αντιγράφων του κλειδιού a στους εξυπηρετητές που διαχειρίζονται τους κόμβους του ΔΔΑ που έχουν κοινό πρόθεμα με το a (τη δυαδική αναπαράσταση της τιμής του a). Για παράδειγμα, αν το πεδίο τιμών του κλειδιού είναι $[0, 15]$, τότε το κλειδί 0001 αποθηκεύεται στον διαχειριστή (εξυπηρετητή) του αναγνωριστικού $H(0001)$ και επιπλέον στους διαχειριστές των αναγνωριστικών $H(000*), H(00*), H(0*), H(*)$.

Η διαδικασία εισαγωγής που προτείνεται δημιουργεί επιπλέον δικτυακή κίνηση, λόγω των h αιτήσεων για την αποθήκευση των αντιγράφων των κλειδιών. Βέβαια στα δίκτυα δομημένης επικάλυψης και γενικότερα στα δίκτυα ομότιμων κόμβων *τεχνικές πλεονασμού* [11, 60] χρησιμοποιούνται κατά κόρον για να αυξηθεί η διαθεσιμότητα των δεδομένων. Όλες οι τεχνικές πλεονασμού ορίζουν τον *βαθμό αντιγραφής* r του συστήματος που δηλώνει τον αριθμό των αντιγράφων των ζευγών. Όταν εισάγεται ή διαγράφεται ένα δεδομένο δημιουργείται επιπλέον δικτυακή κίνηση για την ενημέρωση των αντιγράφων του. Το ΔΔΑ μπορεί να θεωρηθεί και ως

μια τεχνική αντιγραφής κλειδιών με τον βαθμό αντιγραφής h να εξαρτάται (λογαριθμικά) από το πεδίο ορισμού του κλειδιού ($h = \lfloor log_2 M \rfloor + 1$, όπου M η μέγιστη τιμή για το a).

Στο μηχανισμό του ΔΔΑ δεν ορίζεται άμεση λειτουργία για την διαγραφή των αντιγράφων των κλειδιών. Στην ενότητα 3.4.5 περιγράφεται ένας αλγόριθμος βάσει του οποίου κάθε κόμβος μπορεί και διαγράφει αντίγραφα των κλειδιών. Η διαγραφή γίνεται βάσει του *φόρτου* που υφίσταται ο αντίστοιχος εξυπηρετητής λόγω των ερωτημάτων που αντιστοιχούν σε κάποιο από τα αντίγραφα που διατηρεί. Αν παρόλα αυτά χρειάζεται κάποια άμεση λειτουργία διαγραφής τότε όταν θα διαγραφεί κάποιο κλειδί K πρέπει να σταλούν άλλες h αιτήσεις διαγραφής στους διαχειριστές των αντιγράφων του a.

3.4.4 Διαδικασία αναζήτησης εύρους

Η διαδικασία αναζήτησης εύρους τιμών είναι απλή. Δοθείσης μιας ελάχιστης και μιας μέγιστης τιμής για το κλειδί επιστρέφονται όλα τα κλειδιά σε αυτό το εύρος. Συγκεκριμένα, αν έχουμε ένα ερώτημα εύρους $R_q = [x, y],\ x, y \in \{0,1\}^h$, δηλαδή x, y είναι δυαδικές λέξεις με h ψηφία, υπολογίζεται το κοινό πρόθεμα των x, y το οποίο αντιστοιχεί σε κάποιον κόμβο N του ΔΔΑ. Στην συνέχεια το ερώτημα R_q αποστέλλεται στον εξυπηρετητή που διαχειρίζεται τον κόμβο N.

Η ιδανική περίπτωση είναι ο εξυπηρετητής αυτός να διαχειρίζεται όλο το ζητούμενο εύρος τιμών του N. Πρακτικά βέβαια αυτό δεν είναι δυνατόν καθώς οι κόμβοι των υψηλότερων επιπέδων του ΔΔΑ (κοντά στην ρίζα) είναι υπεύθυνοι για μεγάλο αριθμό κλειδιών (μεγάλο εύρος). Αυτό θα δημιουργούσε στους αντίστοιχους εξυπηρετητές των κόμβων αυτών μεγάλο φορτίο καθώς θα *κατακλυζόντουσαν* με ερωτήματα στα οποία θα έπρεπε να απαντήσουν. Έτσι το διαθέσιμο εύρος ζώνης καθώς και οι επεξεργαστικές δυνατότητες του αντίστοιχου εξυπηρετητή δεν θα επαρκούσαν για την εξυπηρέτηση του ερωτήματος. Για να μην *εξαντλούνται* οι διαθέσιμοι πόροι των εξυπηρετητών, εφαρμόζεται ένας αλγόριθμος διαχείρισης των αντιγράφων των κλειδιών (βλ. 3.4.5). Όταν το φορτίο των ερωτημάτων που υφίσταται ένας εξυπηρετητής προκαλεί την εξάντληση των διαθέσιμων πόρων, τότε ο εξυπηρετητής καθορίζει ένα σύνολο από τους κόμβους του ΔΔΑ που διαχειρίζεται ως *κορεσμένους* και διαγράφει τα αντίγραφα των κλειδιών τους. Όταν κάποιος

κόμβος N του ΔΔΑ είναι κορεσμένος δεν μπορεί να απαντήσει σε κάποιο ερώτημα R_q για το εύρος που είναι υπεύθυνος και πρέπει το ερώτημα να προωθηθεί στους απογόνους του. Το ερώτημα R_q μπορεί να απαντηθεί επιτυχώς όταν βρεθούν όλοι οι μη κορεσμένοι απόγονοι του N.

Επιπλέον, η παραπάνω *απλή διαδικασία αναζήτησης εύρους τιμών* (simple search) βασισμένη στο κοινό πρόθεμα μεταξύ της ελάχιστης και μέγιστης τιμής εύρους παρουσιάζει το εξής πρόβλημα. Ο κόμβος-ρίζα είναι υπεύθυνος ακόμα και για ερωτήματα με μικρό εύρος τιμών. Αυτό προκαλεί συχνές επισκέψεις του κόμβου-ρίζα που οδηγεί στην εξάντληση των πόρων του αντίστοιχου εξυπηρετητή. Για παράδειγμα στο δένδρο του σχήματος 3.1 για το ερώτημα εύρους $[0111, 1000]$ (αναζήτηση ζευγών με τιμές $7 \leq a \leq 8$) πρέπει να επισκεφτούμε την ρίζα του δένδρου παρόλο που το εύρος του ερωτήματος είναι μόλις 2.

Για αυτόν τον λόγο ακολουθούμε και μια εναλλακτική στρατηγική για τα ερωτήματα εύρους κατά την οποία το εύρος του ερωτήματος διασπάται σε υπο-εύρη. Η στρατηγική αυτή προτείνεται από τον Gao [50]. Το ερώτημα R_q διασπάται σε $log_2|R_q|$ υποερωτήματα, όπου $|R_q|$ το εύρος του ερωτήματος. Η συγκεκριμένη στρατηγική διάσπασης εγγυάται ότι κανένα υποερώτημα δεν θα σταλεί σε κόμβο του ΔΔΑ που βρίσκεται σε επίπεδο υψηλότερο από $h - log_2|R_q|$ [61]. Στο ΔΔΑ του σχήματος 3.1 το ερώτημα εύρους $[0010,\ 1101]$ διασπάται σε υποερωτήματα που απευθύνονται στους κόμβους $001*,\ 01*,\ 1*$. Στο συγκεκριμένο παράδειγμα το ύψος του δένδρου είναι $h = 4$ και $|R_q| = 13 - 2 = 11$. Παρατηρούμαι ότι ο κόμβος στο υψηλότερο επίπεδο είναι ο $1*$ (επίπεδο $1 = 4 - \lfloor log_2 11 \rfloor$).

3.4.5 Εξισορρόπηση φόρτου - διαχείριση αντιγράφων

Η απόδοση αλλά και το κόστος της διαδικασίας αναζήτησης εύρους εξαρτάται από τον κορεσμό των κόμβων του ΔΔΑ. Ο όρος κορεσμός καθορίζει αν ένας κόμβος N μπορεί να απαντήσει σε ερωτήματα για το εύρος για το οποίο είναι υπεύθυνος. Αν ο N δεν είναι κορεσμένος τότε το κόστος ενός ερωτήματος για το εύρος του N είναι σταθερό. Δυστυχώς, δεν είναι δυνατόν όλοι οι κόμβοι του ΔΔΑ να μην είναι κορεσμένοι, ιδιαιτέρως όταν το εύρος τιμών των κλειδιών που διαχειρίζονται είναι αρκετά μεγάλο. Οι διαχειριστές των μη κορεσμένων κόμβων του ΔΔΑ (που

βρίσκονται στα υψηλότερα επίπεδα κοντά στην ρίζα) θα αναγκάζονται να εξυπηρετούν πολλά ερωτήματα καθώς είναι υπεύθυνοι για μεγάλο εύρος τιμών των κλειδιών. Με αυτόν τον τρόπο επέρχεται και ο φυσικός κορεσμός των διαθέσιμων υπολογιστικών πόρων του διαχειριστή των κόμβων.

Είναι σαφές λοιπόν ότι χρειάζεται μια ορθή διαχείριση των αντιγράφων των κλειδιών ώστε να περιορίζεται ο φόρτος των διαχειριστών των κόμβων. Στην βιβλιογραφία υπάρχουν διάφορες στρατηγικές που μπορούν να εφαρμοστούν. Ο Zheng [58] προτείνει ένα σταθερό όριο στον αριθμό των αντιγράφων που αποθηκεύονται σε κάθε κόμβο του δένδρου DST.

Η φιλοσοφία που ακολουθείται στο ΔΔΑ είναι διαφορετική. Αντί να επιβάλλεται κάποιο προκαθορισμένο (και προφανώς προϋπολογισμένο όριο που βασίζεται σε κάποια συγκεκριμένη κατανομή κλειδιών στο σύστημα) όριο αντιγράφων για τους κόμβους του δένδρου ακολουθείται μια διαδραστική στρατηγική που λαμβάνει υπόψιν το φόρτο (σε αριθμό αντιγράφων) που υφίστανται οι διαχειριστές των κόμβων σε συνδυασμό με τις δυνατότητες τους. Η χρήση ενός προκαθορισμένου ορίου αντιγράφων σε κάθε κόμβο δεν λαμβάνει υπόψιν τον τις υπολογιστικές δυνατότητες που έχει το υπολογιστικό σύστημα του διαχειριστή. Ένα πιθανό σενάριο είναι ο κόμβος του δένδρου να έχει φτάσει στο όριο των αντιγράφων αλλά το πραγματικό φορτίο του διαχειριστή να είναι χαμηλό. Σε αυτήν την περίπτωση ο κόμβος είναι *κορεσμένος* και δεν μπορεί να εξυπηρετήσει ερωτήματα εύρους παρόλο που ο διαχειριστής δεν υφίσταται υψηλό φόρτο για τις δυνατότητες που έχει.

Ο μηχανισμός του ΔΔΑ ορίζει ότι κάθε εξυπηρετητής S έχει μια (μέγιστη) χωρητικότητα c_{max} που δηλώνει τον μέγιστο αριθμό αντιγράφων που μπορεί να εξυπηρετήσει. Επίσης με βάση των αριθμό των εξυπηρετητών στο δίκτυο δομημένης επικάλυψης, την κατανομή τους στο δίκτυο επικάλυψης και το πεδίο τιμών του κλειδιού, σε κάθε εξυπηρετητή S αντιστοιχεί ένα σύνολο U_S από κόμβους του ΔΔΑ. Όσο το πλήθος των αντιγράφων c στον εξυπηρετητή είναι λιγότερο από το όριο c_{max}, ο εξυπηρετητής εξακολουθεί και αποθηκεύει αντίγραφα των κλειδιών. Όταν ο αριθμός των αντιγράφων *ξεπεράσει* το c_{max} τότε διαγράφονται κάποια αντίγραφα με βάση τον αλγόριθμο 1.

Όταν ο αριθμός των αντιγράφων c φτάσει το c_{max} ο αλγόριθμος βρίσκει το σύνολο των κόμβων U_i, που βρίσκεται στο υψηλότερο επίπεδο (κοντά στην ρίζα),

Αλγόριθμος 1 Διαχείριση Αντιγράφων

Input: $U_S = \{U_i : |U_i| \neq 0\}, L$: επίπεδα στο ΔΔΑ, c_{max} : χωρητικότητα εξυπηρετητή, c αριθμός αντιγράφων (φόρτος)

forall U_i *in* U_S **do** i from 0 to L
  forall N_j *in* U_i **do**
    if $|N_j| > 0$ *and* N_j *is not saturated* **then**
      delete all replicas of N_j $N_j \leftarrow$ saturated c $-= |N_j|$ return
    end
  end
end

που περιέχει μη κορεσμένους κόμβους N_j του δένδρου ΔΔΑ και διαγράφει τα κλειδιά τους. Η διαδικασία επαναλαμβάνεται και σε χαμηλότερα επίπεδα έως ότου η χωρητικότητα $c \leq c_{max}$.

Συμπερασματικά, ο αλγόριθμος διαχείρισης αντιγράφων επιτρέπει στους εξυπηρετητές να καθορίζουν τον φόρτο εργασίας που υφίστανται διαγράφοντας αντίγραφα. Το γεγονός αυτό επιτρέπει στους εξυπηρετητές να διατηρούν το φόρτο εργασίας τους σε επιθυμητά επίπεδα, μειώνοντας βέβαια την αποδοτικότητα των αντίστοιχων ερωτημάτων. Αν ο αριθμός των ερωτημάτων για ένα συγκεκριμένο εύρος είναι μεγάλος (π.χ. δημοφιλές ερώτημα), τότε ο αντίστοιχος εξυπηρετητής ορίζει τον κόμβο του ΔΔΑ, έστω N, ο οποίος είναι υπεύθυνος για το εύρος ως κορεσμένο και για το συγκεκριμένο ερώτημα εύρους επιβαρύνονται ισότιμα οι εξυπηρετητές που αντιστοιχούν στους απογόνους του N. Στην επόμενη ενότητα παρουσιάζονται πειράματα που αναλύουν την σταδιακή μείωση στην αποδοτικότητα των ερωτημάτων εύρους και παράλληλα αναλύουν πώς αξιοποιείται η αυξημένη χωρητικότητα των εξυπηρετητών προς όφελος της απόδοσης των ερωτημάτων εύρους.

3.5 Αξιολόγηση

Στο δυαδικό δένδρο αντιγράφων ΔΔΑ ο υπολογισμός της απόδοσης του μηχανισμού εύρεσης ερωτημάτων εύρους βασίζεται στον εντοπισμό των κόμβων που είναι υπεύθυνοι για το ζητούμενο εύρος. Για την ακρίβεια αναζητούνται μη-κορεσμένοι κόμβοι ώστε να διαθέτουν τα αντίγραφα των κλειδιών. Επειδή οι κόμβοι του ΔΔΑ

αντιστοιχούν σε εξυπηρετητές, καθένας από τους οποίους έχει συγκεκριμένη χωρητικότητα, δεν είναι δυνατόν κάθε κόμβος να μπορεί να αποθηκεύει όλα τα αντίγραφα κλειδιών για το εύρος που είναι υπεύθυνος με άμεσο αποτέλεσμα να μην μπορεί να απαντήσει σε ένα ερώτημα για το εύρος για το οποίο είναι υπεύθυνος. Ο κόμβος αυτός ονομάζεται κορεσμένος. Για παράδειγμα όταν ο εξυπηρετητής που διαχειρίζεται τον κόμβο ρίζα του ΔΔΑ φτάσει το μέγιστο όριο αντιγράφων κλειδιών (χωρητικότητα) τότε *θέτει* την ρίζα ως κορεσμένη και διαγράφει τα αντίγραφα των κλειδιών για τα οποία είναι υπεύθυνη η ρίζα. Για να βρεθούν όλα τα κλειδιά με εύρος που περιλαμβάνει ολόκληρο το πεδίο τιμών (ουσιαστικά όλα τα κλειδιά του συστήματος) πρέπει τουλάχιστον να βρεθούν οι διαχειριστές των δυο παιδιών του κόμβου ρίζα.

Ο κορεσμός ενός κόμβου N του ΔΔΑ που βρίσκεται σε επίπεδο l του δένδρου και είναι υπεύθυνος για εύρος τιμών 2^{h-l+1} προκαλεί τον διπλασιασμό των μηνυμάτων που χρειάζονται για να απαντηθεί ένα ερώτημα για το παραπάνω εύρος. Στόχος της ενότητας αυτής είναι η αξιολόγηση της απόδοσης των ερωτημάτων εύρους. *Ως απόδοση ενός ερωτήματος εύρους ορίζεται ο αριθμός των μηνυμάτων που απαιτούνται για να ευρεθούν οι διαχειριστές μη κορεσμένων κόμβων που είναι υπεύθυνοι για το ζητούμενο εύρος.*

Συγκεκριμένα μέσω πειραμάτων που θα αναλυθούν στην συνέχεια εξετάζονται δύο παράμετροι του συστήματος:

- Ο κορεσμός στους κόμβους του ΔΔΑ σε συνάρτηση με τον αριθμό των κλειδιών που εισάγονται στο σύστημα,
- Η επίδραση του κορεσμού των κόμβων στην απόδοση των ερωτημάτων εύρους.

Στις επόμενες υποενότητες παρουσιάζονται οι λεπτομέρειες της υλοποίησης των πειραμάτων, οι παράμετροι που τα επηρεάζουν και στην συνέχεια αναλύονται τα αποτελέσματα των πειραμάτων που διεξήχθησαν.

3.5.1 Προσομοιωτής

Όλα τα πειράματα που παρουσιάζονται στην συνέχεια έχουν διεξαχθεί σε έναν προσομοιωτή που κατασκευάστηκε κατά την διάρκεια εκπόνησης της διατριβής. Ο προσομοιωτής υλοποιήθηκε χρησιμοποιώντας την γλώσσα προγραμματισμού $JAVA^{\text{TM}}$, στην οποία μοντελοποιήθηκε και υλοποιήθηκε το δένδρο ΔΔΑ αλλά και το DST [58]. Το DST είναι επίσης ένας μηχανισμός μέσω του οποίου υποστηρίζονται ερωτήματα εύρους σε δίκτυα δομημένης επικάλυψης και χρησιμοποιείται ως μέτρο σύγκρισης με μεθόδους αντιγραφής που χρησιμοποιεί προκαθορισμένο όριο αντιγράφων για κάθε κόμβο του δένδρου. Όπως έχει περιγραφεί το ΔΔΑ βασίζεται στις δύο βασικές λειτουργίες get και put του DHT, με τη βοήθεια των οποίων μπορεί και αποθηκεύει αντίγραφα των ζευγών $\langle a, v \rangle$.

Ο προσομοιωτής αποτελείται από τρεις βασικές έννοιες που μοντελοποιούνται ως *κλάσεις* στην JAVA:

- **Label** Αντιπροσωπεύει το κλειδί του ζεύγους $\langle a, v \rangle$ και αναπαριστά μια δυαδική λέξη μήκους h. Υποστηρίζει λειτουργίες ερωτημάτων προθέματος (π.χ. κοινό πρόθεμα μεταξύ δυο Label) και λειτουργίες πλοήγησης (εύρεση προγόνου, αριστερού,δεξιού παιδιού) στο δυαδικό δένδρο αντιγράφων.

- **DhtNode**. Αντιπροσωπεύει έναν εξυπηρετητή του δικτύου δομημένης επικάλυψης. Χαρακτηρίζεται από ένα αναγνωριστικό $ID \in [0, 2^{160} - 1]$ (που παράγεται από την υλοποίηση της συνάρτησης SHA-1 στην JAVA) και παρέχει τις δύο βασικές μεθόδους get και put μέσω των οποίων αποθηκεύονται στην αποθήκη ReplicaStore των εξυπηρετητών τα αντίγραφα. Συγκεκριμένα η λειτουργία αποθήκευσης είναι της μορφής $put(Label, Label)$.

- **ReplicaStore**. Στην ενότητα 3.4.2 αναφέρεται ο τρόπος με τον οποίο *κατανέμονται* οι κόμβοι του ΔΔΑ στους κόμβους του δικτύου επικάλυψης. Κάθε DhtNode χρησιμοποιεί την κλάση ReplicaStore ώστε να οργανώνει τα αντίγραφα ζεύγη $< Label, Label >$ στους κόμβους που αντιστοιχούν σε έναν εξυπηρετητή.

Το πεδίο τιμών του $Label$ είναι το διάστημα $[0, 2^{20}]$. Το σύστημα αποτελείται από 1000 εξυπηρετητές (DhtNode), τα αναγνωριστικά των οποίων κατανέμονται

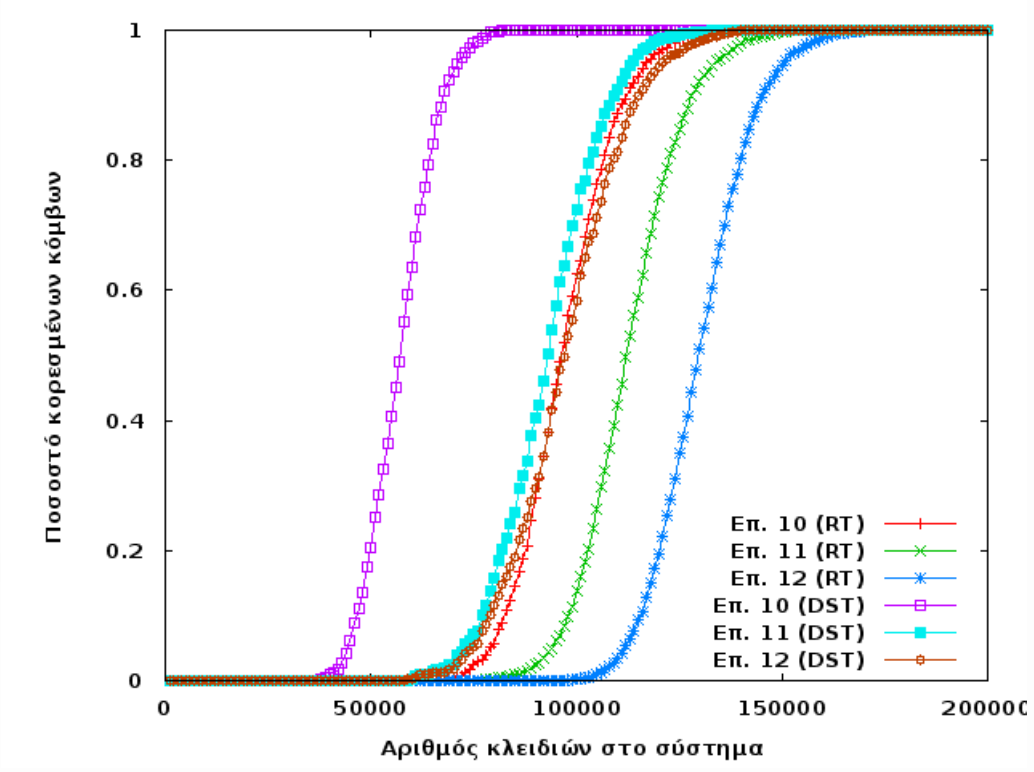
ομοιόμορφα στο πεδίο πεδίο τιμών των αναγνωριστικών με την βοήθεια της συνάρτησης κατακερματισμού αλλά και με την βοήθεια αλγορίθμων [62, 63, 64] που εξασφαλίζουν ότι ο λόγος των αποστάσεων μεταξύ δυο διαδοχικών αναγνωριστικών DhtNode είναι ίσος με 1 (συγκεκριμένα ο λόγος μεταξύ της μικρότερης προς τη μεγαλύτερη απόσταση). Για κάθε λειτουργία $put(Label, Label)$ εκτελούνται άλλες 20 λειτουργίες put ώστε να αποθηκευτούν αντίγραφα του $Label$ στους DHTNode που είναι υπεύθυνοι για όλα τα πιθανά προθέματα του Label. Συγκεκριμένα αν το Label αντιστοιχεί στο ακόλουθο id_0:

$$
\begin{aligned}
0010\ 0010\ 0010\ 0010\ 0010 &\xrightarrow{SHA-1} id_0\\
0010\ 0010\ 0010\ 0010\ 001* &\xrightarrow{SHA-1} id_1\\
0010\ 0010\ 0010\ 0010\ 00* &\xrightarrow{SHA-1} id_2\\
&\cdots\\
&\cdots\\
&\cdots\\
0* &\xrightarrow{SHA-1} id_{19}\\
* &\xrightarrow{SHA-1} id_{20}
\end{aligned}
$$

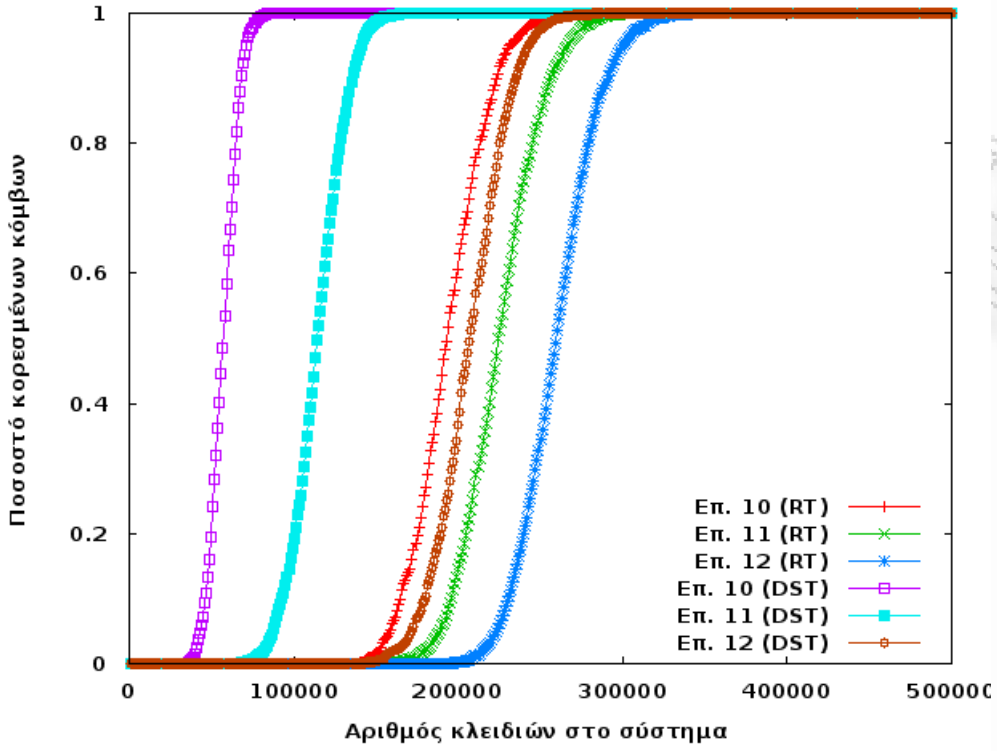
παράγονται άλλα 20 αναγνωριστικά $id_1, \cdots, id_{20}$ για όλα τα πιθανά προθέματα και αποστέλνονται αιτήσεις put στους διαχειριστές DhtNode των αναγνωριστικών αυτών. Για το ΔΔΑ ορίζεται η χωρητικότητα c_{max} που δηλώνει τον μέγιστο αριθμό αντιγράφων που μπορούν να αποθηκευτούν στην ReplicaStore, και για το DST, ορίζεται η παράμετρος γ που δηλώνει τον μέγιστο αριθμό αντιγράφων που αποθηκεύονται σε κάθε (λογικό) κόμβο του δυαδικού δένδρου DST. Τα πειράματα που έχουν διεξαχθεί έχουν επαναληφθεί 10^4 φορές. Στα αποτελέσματα παρουσιάζεται ο στατιστικός μέσος των αποτελεσμάτων. Επίσης στο σύστημα υποθέτουμε ότι ο αριθμός των DhtNode, N, είναι αρκετά μικρότερος από τον αριθμό των κόμβων του ΔΔΑ M, δηλαδή N $<<$ M, με άμεσο αποτέλεσμα σε κάθε DhtNode να αντιστοιχούν περισσότεροι του ενός κόμβοι του δυαδικού δένδρου. Η συγκεκριμένη υπόθεση ισχύει στα περισσότερα συστήματα δικτύων ομότιμων κόμβων όπου τα δεδομένα είναι αρκετά περισσότερα από τους κόμβους που τα διαχειρίζονται.

3.5.2 Πείραμα 1: Κορεσμός στο ΔΔΑ

Στο συγκεκριμένο πείραμα αξιολογείται η συμπεριφορά της προτεινόμενης μεθοδολογίας ως προς τον κορεσμό των κόμβων του δυαδικού δένδρου. Συγκεκριμένα, εξετάζεται το φαινόμενο του κορεσμού των κόμβων στα επίπεδα του δένδρου καθώς εισάγονται κλειδιά (και αντίγραφα αυτών) στο σύστημα. Στο συγκεκριμένο πείραμα θεωρούμε ότι το δίκτυο δομημένης επικάλυψης αποτελείται από ομογενείς κόμβους με ίδια χωρητικότητα c_{max}. Ο αλγόριθμος του ΔΔΑ λαμβάνει υπόψιν του την χωρητικότητα του κάθε DhtNode. Όταν ξεπεραστεί το ανώτατο όριο αντιγράφων εφαρμόζεται στον DhtNode ο αλγόριθμος 3.4.5, και διαγράφονται κάποια αντίγραφα δίνοντας προτεραιότητα (για διαγραφή) στα αντίγραφα που αντιστοιχούν σε κόμβους κοντά στην ρίζα του δένδρου. Ο μηχανισμός αυτός επιτυγχάνει καλύτερη διαχείριση των αντιγράφων σε σχέση με προληπτικές μεθόδους που εκ των προτέρων θέτουν ένα ανώτατο όριο αντιγράφων ανά κόμβο του δένδρου [58]. Στο DST, υπάρχει το όριο γ το οποίο ορίζει τον μέγιστο αριθμό αντιγράφων που έχει ένας κόμβος.



Σχήμα 3.2: Κορεσμός κόμβων του ΔΔΑ και DST,$c_{max} = 1000, \gamma = 30$

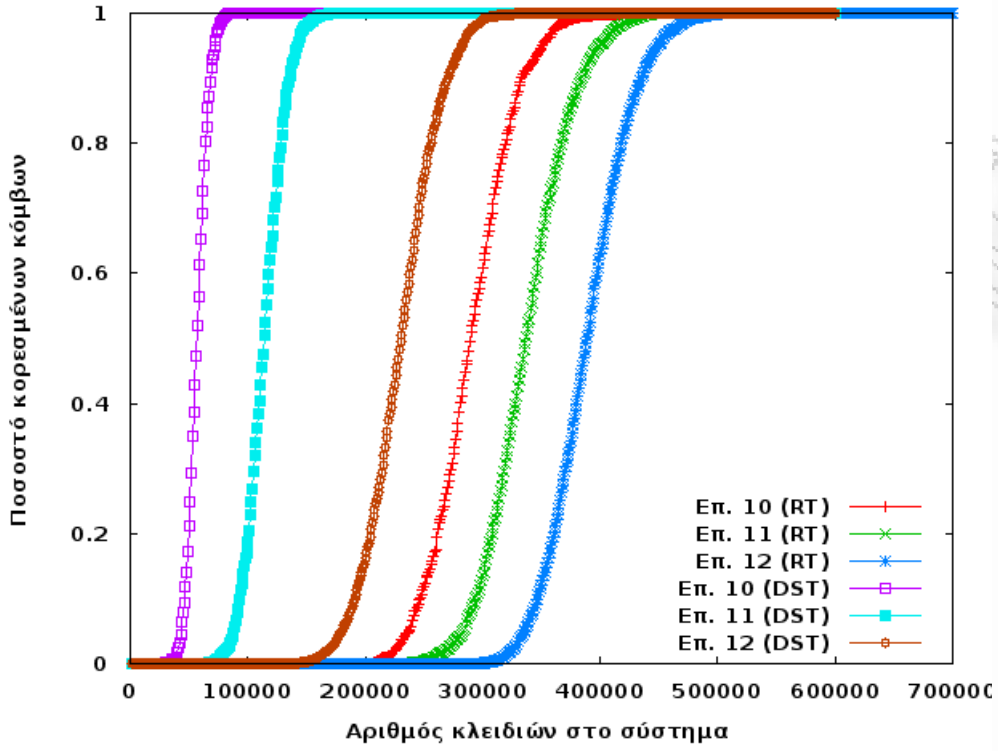


Σχήμα 3.3: Κορεσμός κόμβων του ΔΔΑ και DST, $c_{max} = 2000, \gamma = 30$

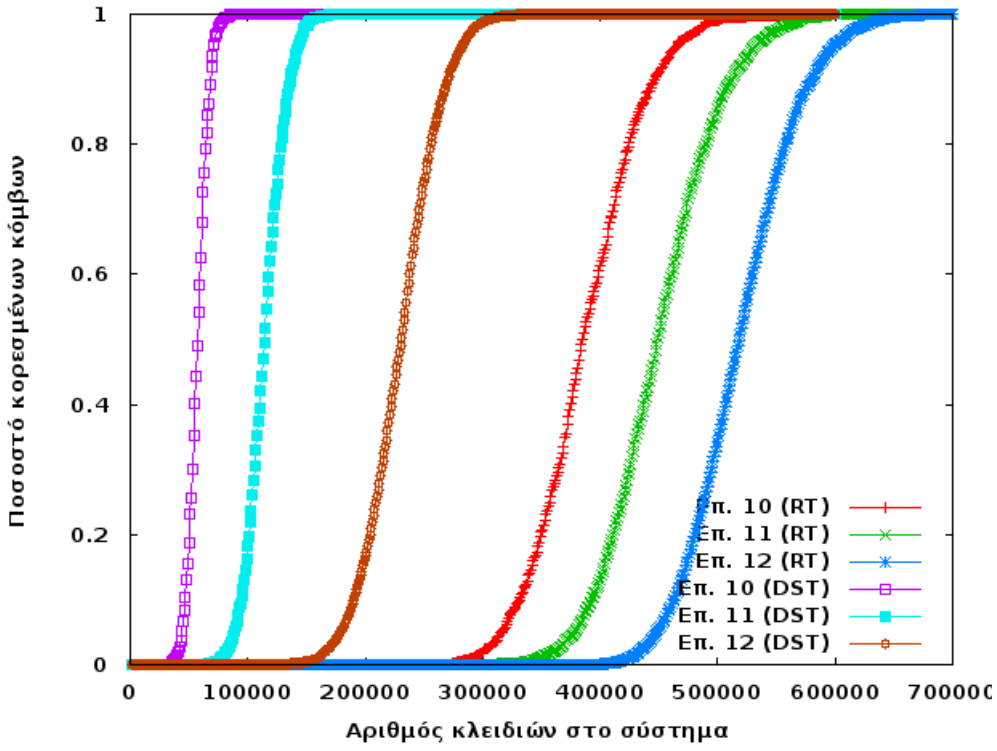
Το όριο $\gamma = 30$ σημαίνει ότι οι κόμβοι του DST επιτρέπουν μέγιστο αριθμό αντιγράφων ανά κόμβο (ανεξαρτήτως επιπέδου) ίσο με 60. Για το ΔΔΑ και το DST εισάγουμε 10^3 κλειδιά[2] και καταγράφουμε τον κορεσμό των κόμβων στα επίπεδα των δένδρων. Η διαδικασία συνεχίζεται μέχρι να εισαχθούν συνολικά 2^{20} κλειδιά (ή μέχρι να επέλθει κορεσμός σε όλους τους κόμβους του δυαδικού δένδρου). Το πείραμα αυτό επαναλαμβάνεται για χωρητικότητα DhtNode $c_{max} = 1000, 2000, 3000, 4000$ και καταγράφουμε τον κορεσμό των κόμβων στα επίπεδα $10, 11, 12$ των ΔΔΑ και DST.

Στο σχήμα 3.2 παρατηρούμε ότι ο κορεσμός στο επίπεδο 10 DST ξεκινά[3] μετά την εισαγωγή 38×10^3 κλειδιών ενώ στο ΔΔΑ μετά την εισαγωγή 67×10^3 κλειδιών. Επίσης

[2] Η κατανομή των κλειδιών κατά την διαδικασία της εισαγωγής είναι τυχαία (ομοιόμορφη κατανομή)
[3] Όταν παρουσιαστεί στο επίπεδο 10 ένας κορεσμένος κόμβος



Σχήμα 3.4: Κορεσμός κόμβων του ΔΔΑ και DST,$c_{max} = 3000, \gamma = 30$



Σχήμα 3.5: Κορεσμός κόμβων του ΔΔΑ και DST, $c_{max} = 4000, \gamma = 30$

ο κορεσμός στο επίπεδο 10 επέρχεται[4] για το DST μετά την εισαγωγή 86×10^3 ενώ για το ΔΔΑ 138×10^3. Παρόμοια συμπεριφορά παρατηρείται και για τα επίπεδα $11, 12$ τόσο για το DST όσο και για το ΔΔΑ αλλά ο κορεσμός ξεκινά αργότερα. Παρατηρείται ότι ο αλγόριθμος του ΔΔΑ διαχειρίζεται καλύτερα την χωρητικότητα των εξυπηρετητών στους οποίους αντιστοιχούν οι κόμβοι. Αντίθετα, η προσέγγιση του DST παραμένει δέσμια της αρχικής εκτίμησης για τον πληθυσμό των κλειδιών αλλά και για το εύρος των ερωτημάτων που μπορεί να υποστηρίζει το σύστημα χωρίς να λαμβάνει υπόψιν του τις δυνατότητες των DhtNodes.

Στα σχήματα 3.3, 3.4 και 3.5 φαίνεται ξεκάθαρα ότι ο αλγόριθμος 1 *διαχειρίζεται* καλύτερα την χωρητικότητα των εξυπηρετητών επιτρέποντας επιπλέον εισαγωγές κλειδιών στο σύστημα. Καθώς αυξάνεται η χωρητικότητα των εξυπηρετητών, ο κορεσμός των κόμβων κατά την εισαγωγή κλειδιών *καθυστερείται*. Όταν η χωρητικότητα είναι $c_{max} = 1000$ ο κορεσμός των κόμβων στο επίπεδο 12 του ΔΔΑ ξεκινά μετά την εισαγωγή 100×10^3 κλειδιών και ολοκληρώνεται μετά την εισαγωγή 167×10^3. Όταν διπλασιαστεί η χωρητικότητα του κάθε εξυπηρετητή τότε ο κορεσμός στο επίπεδο 12 ξεκινά αφού έχουν εισαχθεί στο σύστημα 200×10^3 κλειδιά και ολοκληρώνεται όταν έχουν εισαχθεί 300×10^3 κλειδιά.

Αντίθετα, παρά την αύξηση στην χωρητικότητα ($c_{max} = 2000,\ 3000,\ 4000$) των εξυπηρετητών ο κορεσμός των κόμβων του DST στα επίπεδα $10, 11, 12$ παρουσιάζει την ίδια συμπεριφορά. Για παράδειγμα, ο κορεσμός των κόμβων στο επίπεδο 10 του DST ξεκινά μετά την εισαγωγή 38×10^3 κλειδιών και ολοκληρώνεται με την εισαγωγή 86×10^3 κλειδιών ανεξάρτητα από την χωρητικότητα των εξυπηρετητών (σχήματα 3.2, 3.3, 3.4 και 3.5)

3.5.3 Πείραμα 2: Απόδοση ερωτημάτων εύρους

Η απόδοση ενός ερωτήματος εύρους R_q καθορίζεται από τον αριθμό των μηνυμάτων που πρέπει να αποσταλούν στο δίκτυο προκειμένου να εντοπιστούν μη κορεσμένοι κόμβοι του ΔΔΑ που είναι υπεύθυνοι για το εύρος $|R_q|$. Στα δίκτυα δομημένης επικάλυψης η βασική διαδικασία αναζήτησης `get` εντοπίζει αναγνωριστικά που αντιστοιχούν σε εξυπηρετητές, δεδομένα και σε κόμβους του ΔΔΑ. Το κόστος των ερωτημάτων εύρους μετρείται σε `get` λειτουργίες. Εφόσον ο μηχανισμός

[4] Όταν όλοι οι κόμβοι του επιπέδου 10 έχουν κορεστεί

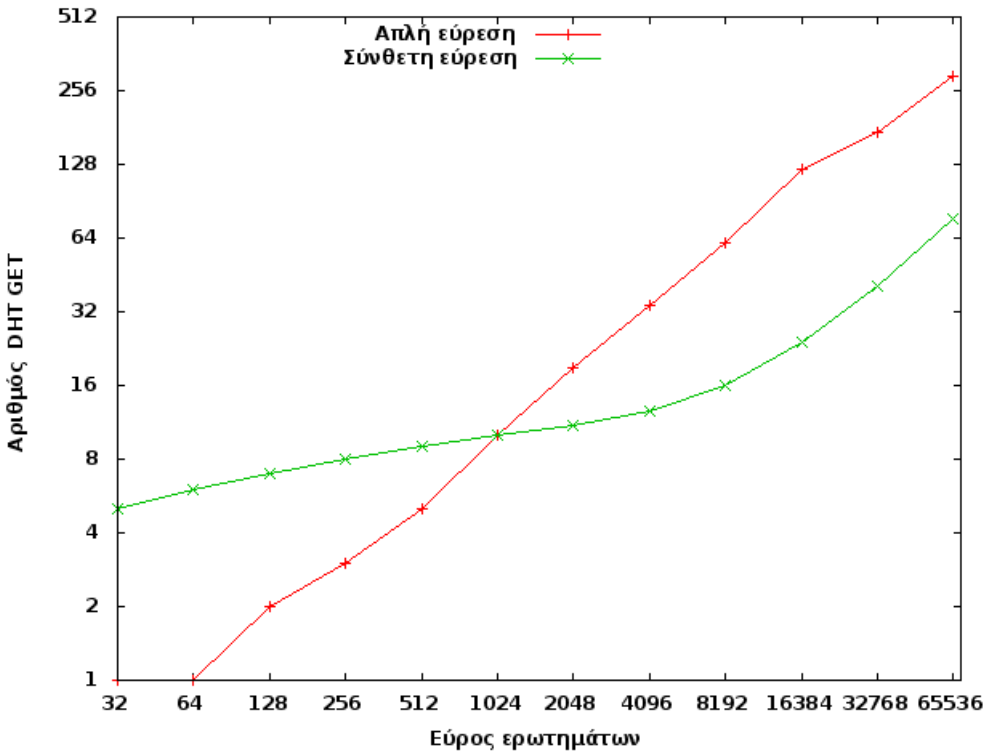
του ΔΔΑ μπορεί να εφαρμοστεί σε οποιοδήποτε δίκτυο δομημένης επικάλυψης υποστηρίζει τις λειτουργίες get και put, το κόστος της προτεινόμενης λύσης δεν πρέπει να επηρεάζεται από τις λεπτομέρειες του κάθε δικτύου επικάλυψης όπως το χρησιμοποιούμενο πρωτόκολλο αναζήτησης ή ο αριθμός των εξυπηρετητών στο σύστημα. Βέβαια, στα περισσότερα πρωτόκολλα αναζήτησης δικτύων δομημένης επικάλυψης [11, 27, 28] το κόστος αναζήτησης εξαρτάται λογαριθμικά $O(log_2 N)$ από τον αριθμό N των εξυπηρετητών στο σύστημα, αν και ο μέσος αριθμός μηνυμάτων κατά την αναζήτηση είναι $1/2 log_2 N$. Για παράδειγμα το σύστημα που διεξάγονται τα πειράματα αποτελείται από 1000 DhtNode και το κόστος μιας λειτουργίας get σε κάθε κόμβο δημιουργεί στο δίκτυο $log_2(1000) \approx 10$ μηνύματα (μέγιστος αριθμός).

Στο συγκεκριμένο πείραμα δημιουργήθηκαν 10^5 ερωτήματα εύρους για τα οποία μετρήθηκε ο αριθμός get μηνυμάτων[5] που χρειάζονται για να απαντηθούν. Το εύρος των ερωτημάτων κυμαίνεται μεταξύ $2^5 - 2^{16}$. Στο σχήμα 3.6 παρουσιάζεται ο μέσος αριθμός μηνυμάτων σε συνάρτηση με το εύρος των ερωτημάτων. Προφανώς αν στο ΔΔΑ δεν υπάρχουν κορεσμένοι κόμβοι τότε η απλή μέθοδος εύρεσης (βλ. 3.4.4) έχει κόστος ενός μηνύματος και το κόστος της σύνθετης μεθόδου εύρεσης εξαρτάται (λογαριθμικά) από το εύρος του ερωτήματος R_q, δηλαδή έχει κόστος $O(log_2|R_q|)$ μηνύματα. Σε αυτό το πείραμα όλοι οι κόμβοι του ΔΔΑ στα επίπεδα $0 - 7$ είναι κορεσμένοι. Επίσης, το 80% στο επίπεδο 8 και το 25% στο επίπεδο 9 είναι κορεσμένοι.

Παρατηρείται ότι για την απλή μέθοδο εύρεσης για εύρη τιμών μεγαλύτερα από 2^6 το κόστος διπλασιάζεται καθώς αυξάνονται τα εύρη. Για την σύνθετη μέθοδο αναζήτησης παρατηρείται ότι έως το εύρος 4096 (2^{12}) το κόστος της μεθόδου ορίζεται από τον δυαδικό λογάριθμο του εύρους του ερωτήματος. Από το σημείο αυτό η σύνθετη μέθοδος εύρεσης επηρεάζεται από τον κορεσμό των κόμβων και ο αριθμός των μηνυμάτων αυξάνεται καθώς αναζητούνται οι εξυπηρετητές των παιδιών των κορεσμένων κόμβων.

Στα σχήματα 3.7 και 3.8 αναλύεται περισσότερο η επίπτωση του κορεσμού των κόμβων στην απόδοση των ερωτημάτων. Συγκεκριμένα παρουσιάζεται πως οι αποδόσεις της απλής (σχήμα 3.7) και της σύνθετης (σχήμα 3.8) αναζήτησης φθίνουν καθώς ο κορεσμός προχωρεί *βαθύτερα* στα επίπεδα του δένδρου. Συγκεκριμένα, το

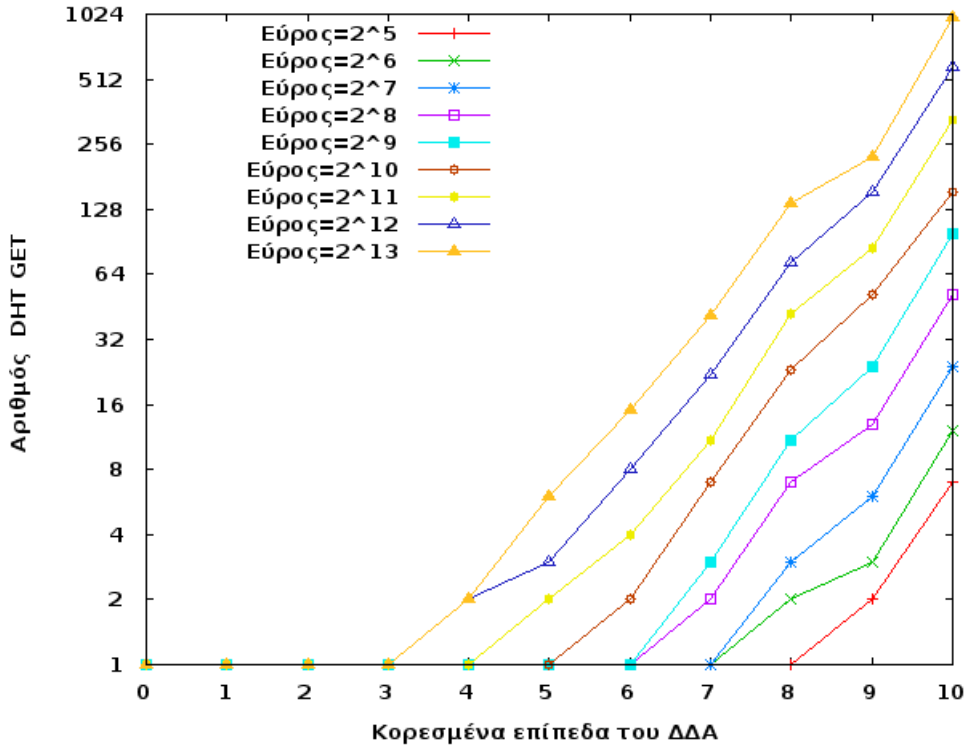
[5]Στην ενότητα αυτή ο όρος μήνυμα χρησιμοποιείται καταχρηστικά για να δηλώσει τον αριθμό get μηνυμάτων



Σχήμα 3.6: Απόδοση απλής και σύνθετης αναζήτησης σε σύστημα με κορεσμένους κόμβους

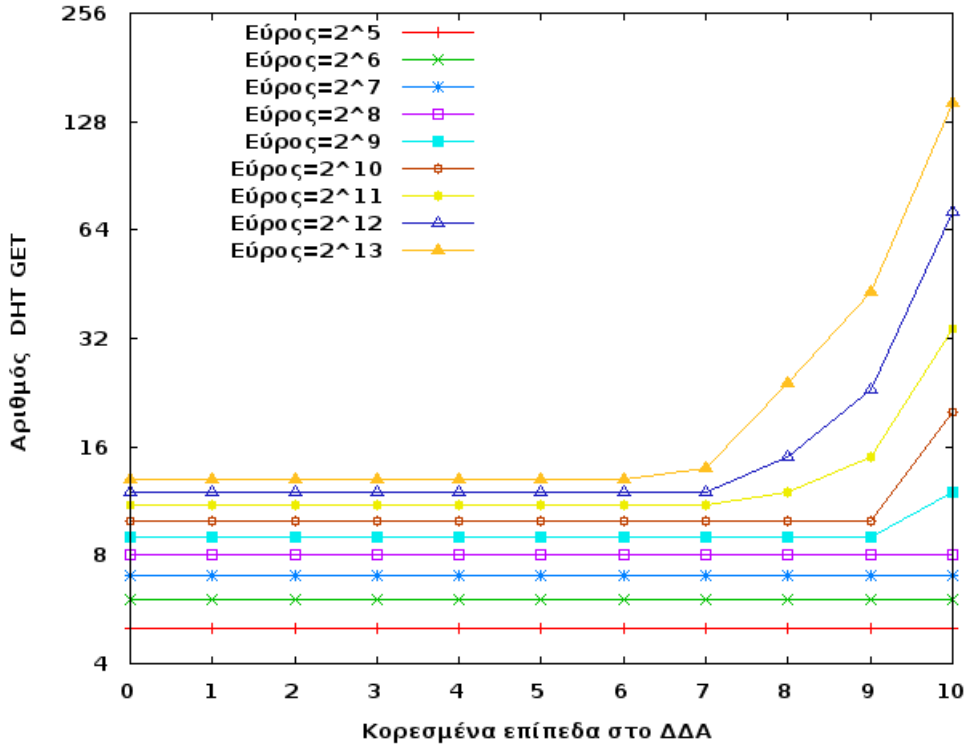
εύρος των ερωτημάτων κυμαίνεται μεταξύ 2^5 ως 2^{13}. Για κάθε εύρος δημιουργούνται 10^5 ερωτήματα με το εύρος τους σε κάθε περίπτωση να κατανέμεται ομοιόμορφα στο πεδίο τιμών του κλειδιού $[0, 2^{20}]$. Η επίδραση του κορεσμού είναι εμφανής και για τις δυο μεθόδους εύρεσης. Βέβαια, η απλή μέθοδος εύρεσης επηρεάζεται περισσότερο. Για παράδειγμα στο σχήμα 3.7 όταν ο κορεσμός έχει επέλθει στο επίπεδο 3 του ΔΔΑ είναι αναμενόμενο ότι το κόστος ερωτημάτων εύρους 2^{17} να μην μπορούν να απαντηθούν με κόστος ενός μόνο μηνύματος. Παρατηρείται όμως, ότι ακόμα και τα ερωτήματα εύρους 2^{13} δεν έχουν κόστος ενός μόνο μηνύματος. Το γεγονός αυτό οφείλεται σε δύο λόγους: λόγω της δενδρικής δομής του ΔΔΑ και της ομοιόμορφης κατανομής των κλειδιών όταν επέλθει ο κορεσμός σε ένα επίπεδο l του δένδρου αυτό σημαίνει ότι και σε υψηλότερα επίπεδα του δένδρου ($\leq l$) υπάρχουν κορεσμένοι κόμβοι. Αυτό σημαίνει ότι ερωτήματα που αντιστοιχούν σε αυτούς τους κορεσμένους κόμβους θα πρέπει να προωθηθούν στους αντίστοιχους απογόνους στο δένδρο αυξάνοντας τον αριθμό μηνυμάτων. Επιπλέον, ο τρόπος λειτουργίας της απλής μεθόδους εύρεσης που βασίζεται στον εντοπισμό του κόμβου ο οποίος έχει κοινό πρόθεμα μεταξύ των δυο *τιμών* του εύρους μπορεί να οδηγήσει

στην επιλογή κάποιου κόμβου που βρίσκεται κοντά στην ρίζα. Τα αποτελέσματα του σχήματος 3.7 δείχνουν ότι κατά μέσο όρο τα ερωτήματα (που κατανέμονται ομοιόμορφα στο πεδίο τιμών του κλειδιού) στέλνονται σε κόμβους που βρίσκονται σε επίπεδο $l-3$, με l . Για αυτόν τον λόγο το κόστος των ερωτημάτων εύρους 2^{13} διπλασιάζεται όταν έχουν κορεστεί οι κόμβοι στο επίπεδο 4 και όχι όταν κορεστούν οι κόμβοι στο επίπεδο 7.



Σχήμα 3.7: Απλή εύρεση: Κόστος εύρεσης συναρτήσει του κορεσμού στο ΔΔΑ (επίπεδα $0-10$)

Όταν το ερώτημα R_q, διασπαστεί σε υπο-ερωτήματα τότε το μεγαλύτερο εύρος που έχει κάποιο υποερώτημα είναι $log|R_q|$ [61]. Το γεγονός αυτό εξασφαλίζει ότι εφόσον στο επίπεδο l του ΔΔΑ δεν υπάρχουν κορεσμένοι κόμβοι τότε τα ερωτήματα εύρους 2^{h-l} έχουν κόστος $h-l$ μηνυμάτων. Στο σχήμα 3.8 παρατηρούμε ότι το ερωτήματα εύρους 2^{13} έχουν κόστος 13 μηνυμάτων εφόσον ο κορεσμός δεν έχει φτάσει στο επίπεδο 7. Από το σημείο αυτό και καθώς ο κορεσμός σταδιακά προχωρά στα επίπεδα του ΔΔΑ, το κόστος ερωτημάτων με εύρος 13 παρουσιάζει τάσεις διπλασιασμού.



Σχήμα 3.8: Σύνθετη εύρεση: Κόστος εύρεσης συναρτήσει του κορεσμού στο ΔΔΑ (επίπεδα $0-10$)

3.6 Συμπεράσματα

Στο κεφάλαιο αυτό εξετάζεται το πρόβλημα της εύρεσης σε δίκτυα δομημένης επικάλυψης. Τα δίκτυα δομημένης επικάλυψης παρέχουν μια αποδοτική μέθοδο εύρεσης αναγνωριστικών που παρέχει εγγυήσεις εύρεσης αναγνωριστικού. Βέβαια, λόγω της συνάρτησης κατακερματισμού που χρησιμοποιείται δεν μπορούν άμεσα να υλοποιηθούν εκφραστικότερα ερωτήματα αναζήτησης, όπως ερωτήματα για ένα εύρος τιμών ενός κλειδιού αναζήτησης. Το συγκεκριμένο πρόβλημα αντιμετωπίστηκε με την χρήση του δυαδικού δένδρου αντιγράφων, στους κόμβους του οποίου διασπάται το πεδίο τιμών του κλειδιού αναζήτησης. Η απόδοση της συγκεκριμένης μεθοδολογίας αξιολογήθηκε με την χρήση προσομοιώσεων.

Παρέχοντας εκφραστικότητα ερωτήματα αναζήτησης, όπως ερωτήματα εύρους, παρέχεται η δυνατότητα να κατασκευαστούν κατανεμημένες εφαρμογές που αποθηκεύουν τα δεδομένα σε δίκτυα δομημένης επικάλυψης, όπως μια ψηφιακή βιβλιοθήκη ερευνητικών εργασιών. Οι απαιτήσεις εύρεσης που έχουν αυτές οι εφαρμογές καλύπτονται μέσω της διαδικασίας αναζήτησης που προσφέρουν τα

δίκτυα δομημένης επικάλυψης σε συνδυασμό με το δυαδικό δένδρο αντιγράφων.

Συγκεκριμένα, αναλύθηκε πως μπορεί να επεκταθεί η βασική μέθοδος αναζήτησης που παρέχουν τα δίκτυα δομημένης επικάλυψης για να υποστηρίζονται ερωτήματα εύρους τιμών επί κάποιου χαρακτηριστικού αναλύθηκαν τρόποι με τους οποίους μπορεί να χρησιμοποιηθεί η βασική μέθοδος get που παρέχουν τα δίκτυα δομημένης επικάλυψης Στο κεφάλαιο αυτό παρουσιάστηκαν μηχανισμοί βάση των οποίων ανακαλύπτονται πόροι σε δίκτυα δομημένης επικάλυψης

Κεφάλαιο 4

Θέματα ασφάλειας σε δίκτυα δομημένης επικάλυψης

Περίληψη

Στο παρόν κεφάλαιο γίνεται επισκόπηση των προβλημάτων ασφαλείας που παρουσιάζονται στα δίκτυα δομημένης επικάλυψης. Αναλύονται οι αδυναμίες στην ασφάλεια των δικτύων επικάλυψης και αναφέρονται λύσεις για κάθε κατηγορία επιθέσεων που έχουν προταθεί στην βιβλιογραφία. Στην συνέχεια αναλύεται το πρόβλημα της πιστοποίησης της ταυτότητας μιας οντότητας σε δίκτυα δομημένης επικάλυψης και παρουσιάζεται μια πρότυπη λύση, το Chord-PKI [12]. Περιγράφονται οι κρυπτογραφικές τεχνικές στις οποίες βασίζεται η λύση αυτή και αναλύονται τα πρωτόκολλα λειτουργίας της. Τέλος, αξιολογείται η απόδοσή του με την χρήση προσομοιώσεων και πειραμάτων.

4.1 Εισαγωγή

Η ασφάλεια αποτελεί σημαντικό παράγοντα για την ομαλή λειτουργία κάθε υπολογιστικού συστήματος. Καθώς η διατριβή αυτή βασίζεται σε μεγάλο βαθμό στα δίκτυα δομημένης επικάλυψης κρίνεται επιτακτική η ενδελεχής μελέτη της ασφάλειάς τους. Τα προβλήματα στην ασφάλεια των δικτύων δομημένης επικάλυψης αφορούν γενικές επιθέσεις που μπορεί να εφαρμοστούν σε οποιοδήποτε σύστημα και επιθέσεις που εκμεταλλεύονται συγκεκριμένες αδυναμίες τους. Το κεφάλαιο αυτό ασχολείται κυρίως με τις δεύτερες επιθέσεις, και προτείνει τρόπους αντιμετώπισής τους.

Τα δίκτυα ομότιμων κόμβων και γενικότερα τα κατανεμημένα συστήματα παρουσιάζουν ιδιαιτερότητες και προκλήσεις στην ασφάλειά τους σε σχέση με κεντρικά συστήματα καθώς δεν υπάρχει ένα και μοναδικό κεντρικό σύστημα για να προστατευτεί. Επιπλέον, οι κόμβοι των ομότιμων συστημάτων λειτουργούν σε άγνωστα περιβάλλοντα και επικοινωνούν με κόμβους που ανήκουν σε διαφορετικές διαχειριστικές αρχές. Όπως θα αναλυθεί στις επόμενες ενότητες το εγγενές χαρακτηριστικό των συστημάτων, να επικοινωνούν δηλαδή με άγνωστους κόμβους, αποτελεί την βασική αδυναμία τους, την οποία εκμεταλλεύονται οι περισσότερες επιθέσεις.

Στις ενότητες που ακολουθούν αναλύεται το θέμα της ασφάλειας των δικτύων δομημένης επικάλυψης, αναφέρονται οι γνωστότερες επιθέσεις στον χώρο και παρουσιάζονται συνοπτικά λύσεις που έχουν προταθεί στην βιβλιογραφία. Παράλληλα αναδεικνύεται το πρόβλημα της *εμπιστοσύνης* μεταξύ των κόμβων (έμπιστων και μη) των δικτύων επικάλυψης και προτείνεται ένας κατανεμημένος μηχανισμός ασφάλειας βάσει του οποίου υλοποιούνται υπηρεσίες εμπιστοσύνης αλλά και διάφορες άλλες υπηρεσίες ασφάλειας όπως αυθεντικοποίηση, δυνατότητα μη-αποποίησης (non-repudiation) κ.α. Ο προτεινόμενος μηχανισμός παρέχει ισχυρές εγγυήσεις ασφάλειας καθώς βασίζεται σε γνωστές κρυπτογραφικές τεχνικές ασύμμετρης κρυπτογράφησης, οι οποίες έχουν προσαρμοστεί στο κατανεμημένο, μη έμπιστο περιβάλλον των δικτύων ομότιμων κόμβων.

Αντιμετωπίζοντας τα ζητήματα ασφάλειας και προτείνοντας τον κατανεμημένο μηχανισμό πιστοποίησης συνεισφέρουμε στην αξιοπιστία των δικτύων δομημένης επικάλυψης, των εφαρμογών που βασίζονται σε αυτά και κατ᾽επέκτασιν στην αξιοπιστία τεχνικών εύρεσης όπως αυτές που αναπτύχθηκαν στο κεφάλαιο 3.

4.2 Ανάλυση προβλήματος ασφάλειας σε δίκτυα δομημένης επικάλυψης

Το πρόβλημα της ασφάλειας στα δίκτυα επικάλυψης επηρεάζεται και περιπλέκεται από πολλούς παράγοντες. Η κατανεμημένη φύση των δικτύων επικάλυψης προσφέρει στους επιτιθέμενους όχι μόνο ένα αλλά πολλαπλά σημεία επίθεσης. Επιπλέον, η βασικότερη λειτουργία των συστημάτων επικάλυψης, η δρομολόγηση, γίνεται συνεργατικά μεταξύ των κόμβων του συστήματος. Κάθε κόμβος προκειμένου να βρει κάποιο αναγνωριστικό βασίζεται στις πληροφορίες άγνωστων μη έμπιστων κόμβων. Ένας από τους βασικούς στόχους στα δίκτυα επικάλυψης είναι η επεκτασιμότητα. Το σύστημα πρέπει να λειτουργεί με λογικό κόστος ακόμα και όταν ο αριθμός των κόμβων που συμμετέχουν στο σύστημα είναι τεράστιος. Οι κόμβοι έρχονται και φεύγουν από το σύστημα συχνά, οπότε είναι απαραίτητο το σύστημα να μην υπερφορτώνεται με πληροφορίες για όλους τους κόμβους του συστήματος. Η κλιμάκωση στα δίκτυα επικάλυψης επιτυγχάνεται με τον κατάλληλο σχεδιασμό των πρωτοκόλλων ώστε κάθε κόμβος να διατηρεί περιορισμένες πληροφορίες για τους άλλους κόμβους του δικτύου, ώστε να μην σπαταλά πολλά μηνύματα κατά την διάρκεια του πρωτοκόλλου ανανέωσης. Επίσης, κατά την εισαγωγή και εξαγωγή κάθε κόμβου ενημερώνονται ελάχιστοι κόμβοι. Η περιορισμένη *αντίληψη* για το δίκτυο που έχει ο κάθε κόμβος επιτρέπει την κλιμάκωση του συστήματος αλλά ταυτόχρονα δίνει την ευκαιρία σε κακόβουλες οντότητες να εκμεταλλευτούν την άγνοια των κόμβων προς δικό τους όφελος.

Οι παραπάνω ιδιαιτερότητες και αδυναμίες των δικτύων επικάλυψης μπορούν να υπονομεύσουν την ασφάλειά τους. Στην παρακάτω ενότητα κατηγοριοποιούνται οι επιθέσεις που μπορούν να λαμβάνουν χώρα στα δίκτυα δομημένης επικάλυψης και στην συνέχεια περιγράφονται οι γνωστότερες από αυτές.

4.2.1 Κατηγοριοποίηση επιθέσεων

Οι επιθέσεις στα συστήματα δομημένης επικάλυψης επηρεάζουν είτε το ίδιο το δίκτυο επικάλυψης είτε το προϊστάμενο επίπεδο εφαρμογής που χρησιμοποιεί το δίκτυο (π.χ. μια εφαρμογή διαμοίρασης αρχείων που χρησιμοποιεί το δίκτυο επικάλυψης). Η πρώτη βασική κατηγοριοποίηση στις επιθέσεις είναι μεταξύ των *επιθέσεων πρωτοκόλλου* και των *επιθέσεων επιπέδου εφαρμογής* του δικτύου δομημένης επικάλυψης.

Όπως έχει αναφερθεί στο κεφάλαιο 2 το πρωτόκολλο των δικτύων επικάλυψης περιλαμβάνει τα εξής βασικά στοιχεία:

- Το πεδίο τιμών (αναγνωριστικών) των κλειδιών
- Το πεδίο τιμών (αναγνωριστικών) των κόμβων
- Κανόνες συσχέτισης-αντιστοίχησης κλειδιών με κόμβους
- Πίνακες δρομολόγησης ανά κόμβο
- Κανόνες ανανέωσης των πινάκων δρομολόγησης κατά την εισαγωγή-αποχώρηση κόμβων.

Οι επιθέσεις που πλήττουν το πρωτόκολλο του δικτύου επικάλυψης στοχεύουν σε ένα ή σε περισσότερα από τα παραπάνω στοιχεία. Μια κακόβουλη οντότητα μπορεί να διαβάλλει το σύστημα δημιουργώντας ένα μεγάλο αριθμό από αναγνωριστικά. Τα αναγνωριστικά αυτά αντιστοιχούν σε πλασματικούς-ψεύτικους κόμβους (κλώνους) οι οποίοι συνεργάζονται υπό τις οδηγίες της κακόβουλης οντότητας προκειμένου να βλάψουν το σύστημα. Η επίθεση αυτή είναι γνωστή με το όνομα επίθεση τύπου Sybil [65].

Επιπλέον, όταν η επιτιθέμενη οντότητα ελέγχει έναν αριθμό από κόμβους στο δίκτυο επικάλυψης μπορεί να πραγματοποιήσει επιθέσεις που βλάπτουν την λειτουργία του συστήματος. Η πιο διαδεδομένη επίθεση αυτού του είδους προσπαθεί να τοποθετήσει τους κόμβους σε *κατάλληλα σημεία* στο δίκτυο επικάλυψης ώστε να απομονώσουν τους τίμιους κόμβους. Η επίθεση αυτή ονομάζεται *έκλειψη* (eclipse attack) επειδή οι κακόβουλοι κόμβοι προσπαθούν να τοποθετηθούν σε κατάλληλα

σημεία, στους πίνακες δρομολόγησης των άλλων κόμβων, ώστε να τους κρύψουν την πραγματική δομή του δικτύου.

Οι επιθέσεις στο επίπεδο εφαρμογής των δικτύων επικάλυψης εξαρτώνται σε μεγάλο βαθμό από την φύση της εφαρμογής. Πολλές εφαρμογές που βασίζονται σε δίκτυα επικάλυψης ασχολούνται με την διαμοίραση δεδομένων [6, 14, 13]. Επιθέσεις σε τέτοιες εφαρμογές σχετίζονται με την άρνηση της παροχής δεδομένων (άρνηση εξυπηρέτησης) ή την επιστροφή στον αιτούντα μη έγκυρων δεδομένων. Οι επιθέσεις αυτού του είδους μπορεί να αντιμετωπιστούν χρησιμοποιώντας τεχνικές ασύμμετρης κρυπτογράφησης [66]. Η δυσκολία στην εφαρμογή των παραπάνω τεχνικών προκύπτει από την γενικότερη φύση των ομότιμων δικτύων, στα οποία υπάρχει απαίτηση για συμμετρία στις λειτουργίες ανάμεσα στους κόμβους του συστήματος και παράλληλα απαίτηση για την αποφυγή εξαρτήσεων από κεντρικές οντότητες, όπως μια κεντρική *αρχή πιστοποίησης*. Η σχεδιαστική αυτή απαίτηση προσπαθεί να διασφαλίσει την επεκτασιμότητα των συστημάτων ομότιμων κόμβων και την αποφυγή κεντρικών σημείων αποτυχίας. Επιπλέον, εφόσον εντοπιστεί ανορθόδοξη συμπεριφορά από κάποιον κόμβο θα πρέπει να παρέχονται μέθοδοι ώστε οι άλλοι κόμβοι να ενημερώνονται για την ύπαρξη του κακόβουλου κόμβου.

Στις επόμενες ενότητες αναλύονται οι επιθέσεις σύμφωνα με την παραπάνω κατηγοριοποίηση και αναφέρονται τρόποι αντιμετώπισής τους.

4.2.2 Επιθέσεις Sybil

Οι επιθέσεις Sybil εκμεταλλεύονται το γεγονός ότι στα συστήματα μεγάλης κλίμακας δεν είναι δυνατόν κάθε κόμβος να *γνωρίζει* όλους τους απομακρυσμένους κόμβους του συστήματος. Οι κόμβοι των δικτύων επικάλυψης δεν είναι πάντοτε σε θέση να επαληθεύουν την ορθότητα των αναγνωριστικών των απομακρυσμένων κόμβων, ιδιαίτερα όταν το δίκτυο είναι αρκετά μεγάλο. Το γεγονός αυτό παρέχει στους επιτιθέμενους την δυνατότητα να εισάγουν πολλαπλά πλαστά αναγνωριστικά στο δίκτυο που θα αντιστοιχούν σε εικονικούς κόμβους. Η επίθεση αυτή δεν καταστρέφει άμεσα το δίκτυο αλλά είναι το μέσο βάση του οποίου μπορεί να δημιουργηθεί μια συνεργαζόμενη ομάδα από έναν μεγάλο αριθμό (πλασματικών) κόμβων. Η ομάδα αυτή δρα υπό της οδηγίες μιας κακόβουλης οντότητας προκειμένου να προκαλέσει ζημιά στο δίκτυο.

Ο Douceur [65] ήταν ο πρώτος που ανάδειξε το παραπάνω πρόβλημα. Ισχυρίζεται ότι σε ένα κατανεμημένο περιβάλλον, όπως είναι τα δίκτυα επικάλυψης, δεν είναι δυνατόν να υπάρξει επαλήθευση των αναγνωριστικών (ταυτοποίηση) των απομακρυσμένων οντοτήτων.

Γενικότερα, σε οποιοδήποτε κατανεμημένο σύστημα οι απομακρυσμένες λογικές οντότητες αντιπροσωπεύουν τους φυσικούς πόρους, οι οποίοι μπορούν να χρησιμοποιηθούν από το σύστημα. Για παράδειγμα, στα δίκτυα επικάλυψης οι απομακρυσμένες οντότητες αντιπροσωπεύονται από τα αναγνωριστικά των κόμβων που αντιστοιχούν σε υπολογιστές. Αν το σύστημα δεν μπορεί να εγγυηθεί ότι κάθε λογική οντότητα αντιστοιχίζεται σε μια μόνο φυσική οντότητα, τότε κάποιος επιτιθέμενος μπορεί να δημιουργήσει αρκετές πλασματικές οντότητες (κλώνους) και να καταστρέψει το δίκτυο επικάλυψης ξεγελώντας τα πρωτόκολλα που βασίζονται σε τεχνικές πλεονασμού.

Ο μόνος πρακτικός τρόπος με τον οποίο παρέχεται ένα προς ένα αντιστοιχία μεταξύ των φυσικών (υπολογιστών) και των λογικών (αναγνωριστικών των κόμβων) οντοτήτων είναι η χρησιμοποίηση μιας *λογικά κεντρικοποιημένης* αρχής πιστοποίησης, η οποία θα εκδίδει πιστοποιητικά που αντιστοιχούν τα αναγνωριστικά με τις φυσικές οντότητες που διαχειρίζονται τους κόμβους του δικτύου.

Παρόλα αυτά στην βιβλιογραφία έχουν προταθεί αρκετοί τρόποι που επιχειρούν να λύσουν στο πρόβλημα. Όλες οι λύσεις προσπαθούν να κάνουν σχετικά δύσκολη την απόκτηση ενός αναγνωριστικού.

Ο Sit [67] συσχετίζει τα αναγνωριστικά με την *διεύθυνση* διαδικτύου (IP address). Αυτή η μέθοδος χρησιμοποιείται στο πρωτόκολλο Chord όπου το αναγνωριστικό κάθε κόμβου παράγεται από την συνάρτηση κατακερματισμού με παραμέτρους την *διεύθυνση* διαδικτύου και τη *θύρα* (port) λειτουργίας της εφαρμογής. Οποιοσδήποτε κόμβος μπορεί να επαληθεύσει το αναγνωριστικό του απομακρυσμένου κόμβου, καθώς η διεύθυνση διαδικτύου του απομακρυσμένου κόμβου μαζί με την θύρα πρέπει να επαληθεύουν το αναγνωριστικό. Βέβαια κάποιος επιτιθέμενος θα μπορούσε να υποκλέψει αρκετές διευθύνσεις διαδικτύου αλλά αυτό είναι δυσκολότερο από την παραγωγή τυχαίων αναγνωριστικών. Από την στιγμή που παρέχεται δυνατότητα για επαλήθευση των αναγνωριστικών των κόμβων η επιτιθέμενη οντότητα πρέπει να καταβάλει μεγάλη προσπάθεια για να υποκλέψει αναγνωριστικά.

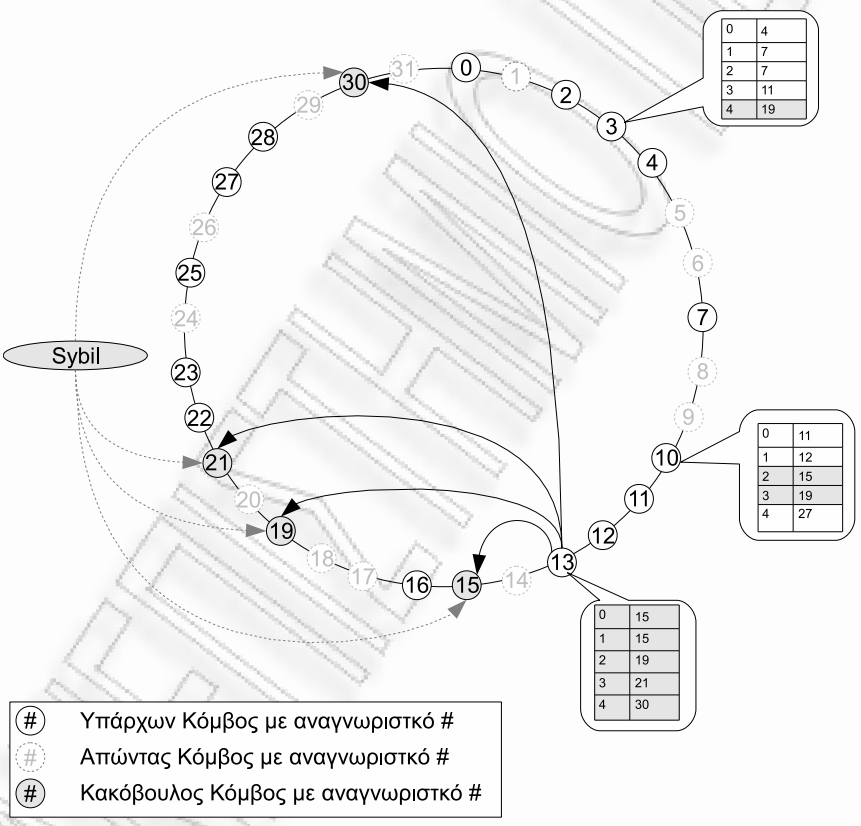
Ο Castro [68] προτείνει τα αναγνωριστικά στα δίκτυα επικάλυψης να πιστοποιούνται από μια *έμπιστη οντότητα* και να συνδέονται με την πραγματική *διεύθυνση* διαδικτύου του εκάστοτε κόμβου. Για να δυσκολέψει την απόκτηση των πιστοποιητικών προτείνει την κοστολόγησή τους με κάποιο χρηματικό ποσό ώστε να μην είναι εύκολη η απόκτησή τους από την επιτιθέμενη οντότητα. Η χρήση των πιστοποιητικών δίνει την δυνατότητα στο σύστημα να ελέγχει ποιος εισέρχεται στο σύστημα. Βέβαια, η χρήση της αρχής πιστοποίησης εισάγει επεξεργαστικό και διαχειριστικό κόστος. Επιπλέον, η κοστολόγηση των πιστοποιητικών είναι σίγουρο ότι θα αποθαρρύνει τους περισσότερους χρήστες καθώς η κοινότητα των χρηστών των ομότιμων δικτύων δεν είναι συνηθισμένη σε τέτοιου είδους πρακτικές πληρωμής των υπηρεσιών αλλά βασίζονται στην εθελοντική διάθεση πόρων τους.

Ένας εναλλακτικός τρόπος με τον οποίο δυσκολεύεται η διαδικασία απόκτησης πολλών αναγνωριστικών είναι με την χρήση υπολογιστικών (κρυπτογραφικών) γρίφων [69]. Κατά την εκτέλεση του πρωτοκόλλου ανανέωσης ο κάθε κόμβος στέλνει σε κάθε γείτονά του έναν αριθμό και μια πρόκληση (challenge). Ανά τακτά χρονικά διαστήματα κάθε κόμβος λύνει ένα γρίφο που αφορά την επίλυση κάποιου προβλήματος. Με αυτόν τον τρόπο ο επιτιθέμενος αν έχει πολλούς κόμβους στην διάθεσή του θα πρέπει να σπαταλά αρκετή υπολογιστική ισχύ για να υπολογίζει τους γρίφους. Προφανώς αν οι υπολογιστικοί πόροι που διαθέτει ο επιτιθέμενος είναι αρκετοί για να επιλύσουν τους γρίφους τότε η συγκεκριμένη επίθεση δεν μπορεί να αποτραπεί.

4.2.3 Επιθέσεις έκλειψης (Eclipse attacks)

Οι κόμβοι στα δίκτυα δομημένης επικάλυψης διατηρούν *συνδέσμους* για έναν μικρό αριθμό κόμβων του συστήματος, τους γείτονές τους. Αν ο επιτιθέμενος διαβάλλει τους πίνακες δρομολόγησης των τίμιων κόμβων και παρεμβάλει κακόβουλους κόμβους τότε μπορεί να τους απομονώσει από το υπόλοιπο δίκτυο. Αυτός ο τρόπος επίθεσης είναι γνωστός ως *επίθεση έκλειψης*. Στο σχήμα 4.1 παρουσιάζεται ένα δίκτυο επικάλυψης βασισμένο στο πρωτόκολλο Chord. Η συνάρτηση κατακερματισμού έχει μέγεθος 5 bit και το εύρος τιμών των αναγνωριστικών των κόμβων είναι $[0, 31]$. Στο σύστημα η κακόβουλη οντότητα ελέγχει τέσσερα αναγνωριστικά (επίθεση τύπου Sybil). Κάθε τίμιος κόμβος που έχει στον πίνακα δρομολόγησής

του κακόβουλους κόμβους (χρωματισμένοι με γκρι) πιθανότατα να μην μπορεί να εκτελέσει το πρωτόκολλο δρομολόγησης σωστά, αν η κακόβουλη οντότητα παρέχει λανθασμένες πληροφορίες δρομολόγησης. Ο κόμβος 10 για την δρομολόγηση, εύρεση αναγνωριστικών στο διάστημα $[14, 18]$ βασίζεται στους κόμβους 15 ως 19. Οι κόμβοι αυτοί είναι κακόβουλοι και έχουν την δυνατότητα να *κρύψουν* το συγκεκριμένο τμήμα του δικτύου $(15 - 19)$ από τον κόμβο 10 (δηλαδή να έχουν προκαλέσει έκλειψη στο συγκεκριμένο τμήμα). Σε ακόμη χειρότερη κατάσταση βρίσκεται ο κόμβος 13 που περιέχει στον πίνακα δρομολόγησης μόνο κακόβουλους κόμβους. Ουσιαστικά δεν έχει γνώση του πραγματικού δικτύου καθώς κακόβουλοι κόμβοι καθορίζουν πλήρως όλους τους κανόνες της δρομολόγησης.



Σχήμα 4.1: Δίκτυο επικάλυψης υπό επίθεση τύπου sybil και έκλειψης

Τα δίκτυα δομημένης επικάλυψης που δεν παρέχουν επαληθεύσιμους κανόνες δρομολόγησης είναι περισσότερο ευάλωτα στις επιθέσεις που αλλοιώνουν τους πίνακες δρομολόγησης των τίμιων κόμβων [67], και για τον λόγο αυτό πρέπει να παρέχουν εναλλακτικούς μηχανισμούς αντιμετώπισης ενάντια σε κακόβουλους κόμβους που παρέχουν λάθος πληροφορίες δρομολόγησης. Αντίθετα, σε πρωτόκολλα

επικάλυψης όπως το Chord, που παρέχουν μηχανισμούς επαλήθευσης της δρομολόγησης, οι κόμβοι μπορούν να εντοπίζουν λανθασμένες δρομολογήσεις. Στο παράδειγμα του σχήματος 4.1 έστω ότι ο κόμβος 10 ψάχνει για το κλειδί 16. Το ερώτημα θα δρομολογηθεί στον κακόβουλο κόμβο 19, ο οποίος βέβαια δεν μπορεί να τον δρομολογήσει προς τον 12 ή τον 15 καθώς στο πρωτόκολλο του Chord πρέπει να να αυξάνονται τα αναγνωριστικά σε κάθε βήμα δρομολόγησης.

Επιπλέον, δίκτυα δομημένης επικάλυψης που δεν παρέχουν εναλλακτικά μονοπάτια δρομολόγησης επηρεάζονται σε μεγάλο βαθμό από τις επιθέσεις τύπου έκλειψης, καθώς μειώνεται το ποσοστό επιτυχίας της δρομολόγησης. Αν σε ένα δίκτυο το ποσοστό των κακόβουλων κόμβων είναι 20% του συνόλου και ο μέσος αριθμός βημάτων είναι 5 (για ένα δίκτυο με περίπου 1000 κόμβους) τότε το ποσοστό επιτυχούς δρομολόγησης είναι $(1 - 0.2)^5 \approx 0.327$. Οι περισσότερες εφαρμογές δικτύων επικάλυψης δεν μπορούν να λειτουργήσουν με τόσο χαμηλά ποσοστά επιτυχούς δρομολόγησης.

Στην βιβλιογραφία υπάρχουν αρκετοί τρόποι με τους οποίους αντιμετωπίζεται η συγκεκριμένη επίθεση [68, 70, 71]. Οι προτεινόμενοι μηχανισμοί προσπαθούν να εγγυηθούν ότι το ποσοστό των κακόβουλων κόμβων στους πίνακες δρομολόγησης δεν ξεπερνά το ποσοστό των έντιμων κόμβων στο δίκτυο. Αυτό ισχύει στην περίπτωση που τα αναγνωριστικά των κόμβων προκύπτουν με τυχαίο τρόπο.

Ιδιαίτερα ευάλωτα στις επιθέσεις τύπου έκλειψης είναι τα δίκτυα επικάλυψης στα οποία η επιλογή των γειτονικών κόμβων δεν γίνεται με βάση το πρωτόκολλο λειτουργίας τους αλλά βάσει κάποιου μηχανισμού βελτιστοποίησης κάποιου δικτυακού μεγέθους. Δηλαδή τα αναγνωριστικά που τοποθετούνται στους πίνακες δρομολόγησης δεν βασίζονται στην σχέση διάταξης που ορίζεται από πρωτόκολλο (αύξουσα αριθμητική διάταξη [11] ή μέγιστο κοινό πρόθεμα αναγνωριστικού [43]) αλλά στην ελαχιστοποίηση της δικτυακής καθυστέρησης (latency) ή στην ελαχιστοποίηση του αριθμού των βημάτων μεταξύ δύο κόμβων. Οι κακόβουλοι κόμβοι εκμεταλλεύονται το γεγονός αυτό παρέχοντας λανθασμένες πληροφορίες σχετικά με την δικτυακή απόσταση των κόμβων. Στο παράδειγμα του σχήματος 4.1 όταν ο τίμιος κόμβος 10 ρωτήσει για την απόσταση του 15 η κακόβουλη οντότητα (Sybil στην συγκεκριμένη περίπτωση) μπορεί να επιλέξει μεταξύ των $15, 21, 30$, αναλόγως με το ποιος κόμβος βρίσκεται στην εγγύτητα του 10. Μια λύση [68] αντιμετωπίζει

το πρόβλημα των επιθέσεων έκλειψης με την χρήση δυο πινάκων δρομολόγησης. Ο βασικός πίνακας δρομολόγησης κάθε κόμβου περιέχει εγγραφές με βάση την δικτυακή εγγύτητα των κόμβων. Ο άλλος πίνακας περιέχει εγγραφές η ορθότητα των οποίων μπορεί να επαληθευτεί καθώς βασίζονται στην σχέση διάταξης που ορίζεται από το πρωτόκολλο του δικτύου επικάλυψης. Η συγκεκριμένη λύση προϋποθέτει ότι το δίκτυο επικάλυψης παρέχει κανόνες επαλήθευσης της δρομολόγησης.

4.2.4 Ασφάλεια στο επίπεδο δρομολόγησης-αποθήκευσης

Οι προαναφερθείσες επιθέσεις δεν βλάπτουν άμεσα το δίκτυο δομημένης επικάλυψης αλλά αποτελούν τη βάση για την πραγματοποίηση άλλων επιθέσεων που βλάπτουν τις εφαρμογές που βασίζονται σε δίκτυα δομημένης επικάλυψης. Οι περισσότερες επιθέσεις προσπαθούν να βλάψουν το σύστημα υπονομεύοντας την λειτουργία της δρομολόγησης που είναι η κύρια λειτουργία των δικτύων επικάλυψης. Όπως είναι γνωστό στα δίκτυα δομημένης επικάλυψης η δρομολόγηση γίνεται συνεργατικά μεταξύ των κόμβων του συστήματος. Κακόβουλοι κόμβοι μπορεί να αρνηθούν την προώθηση κάποιας αίτησης εύρεσης ή ακόμη χειρότερα να την προωθήσουν σε λάθος κόμβους. Επίσης, τα συστήματα επικάλυψης είναι συστήματα δρομολόγησης κλειδιών που αντιστοιχούν σε κάποια δεδομένα. Οι κακόβουλοι κόμβοι μπορεί να αρνηθούν την ύπαρξη κάποιου έγκυρου κλειδιού η να επιστρέψουν μη έγκυρα δεδομένα.

Οι επιθέσεις που διαβάλουν την διαδικασία της δρομολόγησης του συστήματος όπως και αυτές που επιστρέφουν λανθασμένες απαντήσεις για δεδομένα αντιμετωπίζονται με τεχνικές *πλεονασμού* στα αντίστοιχα επίπεδα. Δηλαδή, χρησιμοποιούν εναλλακτικά μονοπάτια δρομολόγησης ή πολλαπλά αντίγραφα των δεδομένων. Στην συνέχεια αναλύουμε συγκεκριμένες επιθέσεις στο επίπεδο δρομολόγησης και αποθήκευσης και παρουσιάζουμε τεχνικές με τις οποίες μπορεί να αντιμετωπιστούν.

Ο Sit [67] αντιμετωπίζει τις επιθέσεις στο επίπεδο της δρομολόγησης υιοθετώντας μηχανισμούς *επαναληπτικής δρομολόγησης*. Στην επαναληπτική δρομολόγηση ο κόμβος που ψάχνει κάποιο αναγνωριστικό ελέγχει την διαδικασία της δρομολόγησης σε κάθε βήμα. Σε κάθε βήμα οι κόμβοι επιστρέφουν αναγνωριστικά κόμβων

πιο κοντά προς το κλειδί και στην συνέχεια ο αρχικός κόμβος προωθεί την αίτηση (εύρεσης του κλειδιού) στο αντίστοιχο κόμβο. Συγκεκριμένα, ο κόμβος που ξεκινάει την διαδικασία δρομολόγησης προωθεί το μήνυμα προς κάποιον κόμβο, έστω N_1, και περιμένει από τον N_1 τον επόμενο κόμβο στον οποίο πρέπει να προωθηθεί το μήνυμα, έστω ο κόμβος N_2. Ο κόμβος επαναλαμβάνει την προηγούμενη διαδικασία για τον N_2. Η διαδικασία αυτή επαναλαμβάνεται μέχρι να βρεθεί ο ζητούμενος κόμβος. Σε δίκτυα επικάλυψης που παρέχουν τρόπους επαλήθευσης της δρομολόγησης, όπως το Chord, ο αρχικός κόμβος ελέγχει σε κάθε ενδιάμεσο κόμβο την πρόοδο της δρομολόγησης. Συγκεκριμένα, ο αρχικός κόμβος ελέγχει ότι σε κάθε βήμα της δρομολόγησης το επιστρεφόμενο αναγνωριστικό αυξάνεται. Αν υποθέσουμε ότι μια αίτηση δρομολόγησης (επαναληπτική) αφορά τους κόμβους $N, N_1, N_2, N_3, \cdots, N_n$ τότε ο N γνωρίζει ότι τα αναγνωριστικά $N_1, \cdots, N_n$ πρέπει να είναι σε αύξουσα σειρά. Το κόστος που έχει η επαναληπτική δρομολόγηση είναι ο διπλασιασμός της καθυστέρησης αλλά και του αριθμού μηνυμάτων στο δίκτυο σε σχέση με την αναδρομική δρομολόγηση.

Τα δίκτυα επικάλυψης μπορούν να προστατευτούν από επιθέσεις στο επίπεδο της δρομολόγησης με την χρήση επιπλέον εγγραφών στον πίνακα δρομολόγησης τους. Συγκεκριμένα, με βάση την την τεχνική των *πλατιών μονοπατιών*-(wide paths) [72] κάθε κόμβος διατηρεί επιπλέον εγγραφές στους πίνακες δρομολόγησης για κάθε προορισμό. Με αυτήν την τεχνική για να αποτύχει η δρομολόγηση σε κάποιον κόμβο θα πρέπει να αποτύχουν όλοι οι επιπλέον κόμβοι στους πίνακες δρομολόγησης. Επίσης, τα μονοπάτια μπορεί να είναι ανεξάρτητα μεταξύ τους [73] ώστε η αποτυχία της δρομολόγησης του ενός μονοπατιού να μην εγγυάται και την αποτυχία του άλλου. Οι τεχνικές αυτές αυξάνουν το δικτυακό κόστος του πρωτοκόλλου ανανέωσης καθώς πρέπει να ελέγχονται και να ανανεώνονται οι επιπλέον εγγραφές.

Ένας εναλλακτικός τρόπος με τον οποίο εξασφαλίζεται η επιτυχής δρομολόγηση είναι με την χρήση υπογεγραμμένων πιστοποιητικών [74], μέσω των οποίων κάθε κόμβος αποδεικνύει ότι είναι *υπεύθυνος* για το εύρος τιμών. Για να αποδείξει κάθε κόμβος ότι είναι υπεύθυνος για το υποτιθέμενο εύρος τιμών θα πρέπει να διαθέτει ζευγάρια ιδιωτικού και δημόσιου κλειδιού για κάθε πρόθεμα του αναγνωριστικού του. Ο κόμβος με αναγνωριστικό $ABCD$ θα λάβει ζευγάρια για τα προθέματα A, AB, ABC. Ανά τακτά χρονικά διαστήματα ο κόμβος θα στέλνει σε

συγκεκριμένους κόμβους (τους διαχειριστές των αναγνωριστικών $H(A), H(AB)$) αποδείξεις ότι είναι υπεύθυνος για τα συγκεκριμένα προθέματα (υπογεγραμμένες με το ιδιωτικό κλειδί του κάθε προθέματος).

Ο πιο διαδεδομένος τρόπος με τον οποίο εξασφαλίζεται η ασφάλεια των δεδομένων ενάντια σε αντίστοιχες επιθέσεις είναι με την αντιγραφή τους σε πολλαπλούς κόμβους. Τα περισσότερα DHT [11], [43], [28] αποθηκεύουν αντίγραφα των δεδομένων σε κοντινούς κόμβους για την ευκολότερη διαχείριση των αντιγράφων (π.χ. διατήρηση αριθμού αντιγράφων, ανανέωση αντιγράφων).

Ο Castro [68] χρησιμοποιεί μια συνάρτηση βάσει της οποίας αντιστοιχίζονται τα αντίγραφα σε πολλούς κόμβους στο δίκτυο. Εναλλακτικά τα αντίγραφα μπορούν να αποθηκευτούν σε τυχαίους κόμβους του δικτύου. Η συγκεκριμένη μέθοδος χρησιμοποιείται στο Tapestry [27]. Παρόμοια, ο Harverst [75] προτείνει την τοποθέτηση των αντιγράφων σε κόμβους που ισαπέχουν στο κυκλικό πεδίο τιμών του Chord. Με τον πλεονασμό τον δεδομένων δημιουργούνται πολλαπλά ανεξάρτητα μονοπάτια δρομολόγησης.

Τέλος, προβλήματα ασφάλειας που σχετίζονται με την ακεραιότητα των δεδομένων μπορεί να λυθούν χρησιμοποιώντας τεχνικές και μηχανισμούς κρυπτογραφίας. Επίσης, η χρήση ψηφιακών υπογραφών επιτρέπει την επαλήθευση της ταυτότητας του αποστολέα κάποιου μηνύματος δοθέντος ότι υπάρχει κάποια αρχή πιστοποίησης την οποία εμπιστεύονται όλοι οι κόμβοι του συστήματος. Γενικότερα, οι τεχνικές κρυπτογραφίας εισάγουν επιπλέον κόστος στο δίκτυο επικάλυψης (κόστος διατήρησης της υποδομής δημοσίου κλειδιού – ΥΔΚ) αλλά ταυτόχρονα επιτρέπουν στα δίκτυα επικάλυψης να μην χρησιμοποιούν τις τεχνικές πολλαπλής δρομολόγησης που αναφέρθηκαν παραπάνω και οι οποίες αυξάνουν το δικτυακό κόστος, παρά μόνο στην περίπτωση που ο μηχανισμός πιστοποίησης των δεδομένων διαπιστώσει λάθος στα δεδομένα.

4.3 Πρόβλημα ταυτοποίησης

Η πληθώρα των παραπάνω προβλημάτων ασφάλειας καταδεικνύουν ότι το περιβάλλον λειτουργίας των δικτύων δομημένης επικάλυψης είναι αρκετά επισφαλές, και επίσης, ότι οι άγνωστοι κόμβοι δεν πρέπει να θεωρούνται αυτόματα έμπιστοι.

Το γεγονός ότι η δρομολόγηση στα δίκτυα επικάλυψης γίνεται συνεργατικά επιτρέπει σε κακόβουλους κόμβους να παρεμβληθούν στην επικοινωνία δυο κόμβων. Είναι απαραίτητη λοιπόν για τις προϊστάμενες εφαρμογές των δικτύων δομημένης επικάλυψης μια υποδομή ασφάλειας με την οποία να εξασφαλίζεται η ασφαλής επικοινωνία μεταξύ δύο έμπιστων κόμβων. Επιπλέον, απαιτούνται και άλλες υπηρεσίες ασφάλειας που προστατεύουν από την παραποίηση των δεδομένων. Πολλές από τις λύσεις που παρουσιάστηκαν παραπάνω βασίζονται σε μία αρχή πιστοποίησης, μέσω της οποίας πιστοποιούνται τα αναγνωριστικά των κόμβων, δηλαδή η ταυτότητα των κόμβων. Χρησιμοποιώντας μια υποδομή δημοσίου κλειδιού – ΥΔΚ εξασφαλίζεται μερικώς η ασφαλής επικοινωνία μεταξύ των έμπιστων κόμβων του δικτύου και παράλληλα παρέχονται επιπλέον υπηρεσίες ασφάλειας με την χρήση ψηφιακών υπογραφών. Παραδοσιακά, η αρχή πιστοποίησης είναι μια οντότητα την οποία εμπιστεύονται άλλες οντότητες προκειμένου να συνδιαλλαγούν μεταξύ τους. Η χρησιμοποίηση μιας κεντρικής αρχής πιστοποίησης από τα συστήματα δικτύων επικάλυψης που είναι πλήρως κατανεμημένα συστήματα παρουσιάζει προβλήματα καθώς δημιουργούνται ανεπιθύμητες εξαρτήσεις με την κεντρική αρχή. Οι εξαρτήσεις αυτές είναι ανεπιθύμητες καθώς παραβιάζουν έναν από τους βασικότερους κανόνες των συστημάτων επικάλυψης, όπου κανένας κόμβος δεν είναι σημαντικότερος από κάποιον άλλον. Επιπλέον, η υιοθέτηση της κεντρικής αρχής θα δημιουργούσε για το σύστημα ένα κεντρικό σημείο αποτυχίας. Εκτός από τον κίνδυνο της αποτυχίας, το κεντρικό σημείο πλήττει και την επεκτασιμότητα των δικτύων δομημένης επικάλυψης καθώς με τις λειτουργίες της πιστοποίησης θα επιβαρύνεται αποκλειστικά η αρχή πιστοποίησης. Για αυτόν τον λόγο προτείνεται μια κατανεμημένη υποδομή πιστοποίησης. Συγκεκριμένα, οι λειτουργίες της πιστοποίησης κατανέμονται στους κόμβους του δικτύου επικάλυψης και δεν δημιουργούνται εξαρτήσεις του συστήματος από μια εξωτερική κεντρική αρχή πιστοποίησης.

Στην ενότητα 4.4 ορίζονται οι απαιτήσεις και οι προδιαγραφές που πρέπει να έχει ένας κατανεμημένος μηχανισμός πιστοποίησης και στην συνέχεια στην ενότητα 4.5 προτείνεται μια κατανεμημένη υποδομή δημοσίου κλειδιού, που υποστηρίζει τις προδιαγραφές και βασίζεται στο δίκτυο επικάλυψης του πρωτοκόλλου Chord.

4.4 Κατανεμημένος μηχανισμός πιστοποίησης

Οι υποδομές δημοσίου κλειδιού–ΥΔΚ [76] μπορούν να δώσουν λύσεις σε πολλά από τα προβλήματα ασφάλειας που αντιμετωπίζουν οι εφαρμογές των δικτύων επικάλυψης μέσω των υπηρεσιών ασφάλειας που παρέχουν όπως αυθεντικοποίηση, ψηφιακές υπογραφές και κρυπτογράφηση δεδομένων. Βέβαια λόγω της κατανεμημένης φύσης των δομημένων δικτύων επικάλυψης, του σύντομου χρόνου παραμονής των κόμβων στο σύστημα αλλά και λόγω του επισφαλούς περιβάλλοντος στο οποίο βασίζεται η δρομολόγηση των μηνυμάτων στο σύστημα η δημιουργία, η εγκατάσταση αλλά και η λειτουργία ενός μηχανισμού δημοσίου κλειδιού περιπλέκεται. Για αυτόν τον λόγο αναπτύξαμε τα απαραίτητα πρωτόκολλα για την δημιουργία μιας κατανεμημένης ΥΔΚ [12] και αξιολογήσαμε την επίδοσή τους με την χρήση προσομοιώσεων. Η κατανεμημένη ΥΔΚ χρησιμοποιεί το πρωτόκολλο αναζήτησης του Chord [11] ως δίκτυο δομημένης επικάλυψης, στους κόμβους του οποίου κατανέμονται οι λειτουργίες της πιστοποίησης. Θεωρητικά θα μπορούσε να χρησιμοποιηθεί (με ορισμένες μετατροπές) οποιοδήποτε άλλο πρωτόκολλο δρομολόγησης δομημένων δικτύων επικάλυψης. Στην υποενότητα 4.4.1 ορίζονται οι λειτουργίες της ΥΔΚ και τίθενται οι προδιαγραφές στην ασφάλεια και στην απόδοση της ΥΔΚ που κατανέμεται στους κόμβους του δικτύου επικάλυψης. Στην συνέχεια, στην ενότητα 4.5 περιγράφεται μια πρότυπη κατανεμημένη ΥΔΚ, το Chord-PKI, που υλοποιεί τις λειτουργίες με βάση τις προδιαγραφές (ασφάλειας και απόδοσης) της υποενότητας 4.4.1.

4.4.1 Βασικές λειτουργίες της κατανεμημένης ΥΔΚ

Οι βασικές υπηρεσίες της ΥΔΚ αφορούν την έκδοση πιστοποιητικών βάσει των οποίων οι κόμβοι του συστήματος εξασφαλίζουν την ασφάλεια στις υπηρεσίες τους (κρυπτογράφηση επικοινωνίας, επιβεβαίωση δεδομένων, ψηφιακές υπογραφές). Συγκεκριμένα, οι παρεχόμενες λειτουργίες είναι:

- **Έκδοση πιστοποιητικών**. Οι κόμβοι του συστήματος έχουν την δυνατότητα πιστοποίησης από την ΥΔΚ εφόσον πληρούν κάποιες προϋποθέσεις. (π.χ. παρατηρούμενη αξιοπιστία κόμβου)

- **Ανάκληση πιστοποιητικών**. Τα πιστοποιητικά των κόμβων μπορεί να ανακληθούν εφόσον ο αντίστοιχος κόμβος παύει να πληρεί τις προϋποθέσεις βάσει των οποίων πιστοποιήθηκε αρχικά.

- **Αποθήκευση πιστοποιητικών**. Τα πιστοποιητικά και οι λίστες ανάκλησης πρέπει να αποθηκεύονται μεταξύ των κόμβων του συστήματος.

- **Αναζήτηση/ανάκτηση πιστοποιητικών**. Οι κόμβοι στο σύστημα πρέπει να αναζητούν και να ανακτούν τα πιστοποιητικά και τις λίστες ανάκλησης (πιστοποιητικών) των άλλων κόμβων.

4.4.2 Προδιαγραφές ασφάλειας

Το επισφαλές περιβάλλον των δικτύων δομημένης επικάλυψης θέτει κάποιες απαιτήσεις στο επίπεδο της ασφάλειας των υπηρεσιών που βασίζονται σε αυτά. Η κατανεμημένη ΥΔΚ πρέπει να λάβει υπόψιν της αυτές τις απαιτήσεις ώστε οι υπηρεσίες πιστοποίησης να είναι ασφαλείς και πρακτικά εφαρμόσιμες στα δίκτυα επικάλυψης. Οι απαιτήσεις είναι:

- **Διαθεσιμότητα**. Η υπηρεσία ΥΔΚ πρέπει να είναι συνεχώς διαθέσιμη προς τους κόμβους. Στα δίκτυα δομημένης επικάλυψης δεν είναι διαθέσιμοι συνεχώς όλοι οι κόμβοι του συστήματος. Οι κόμβοι έρχονται και φεύγουν από το σύστημα συχνά. Το γεγονός αυτό δεν πρέπει να επηρεάζει τις υπηρεσίες της ΥΔΚ οι οποίες πρέπει να παρέχονται αδιάλειπτα.

- **Ανοχή σε σφάλματα** Η ΥΔΚ πρέπει να ανέχεται εκούσιες και ακούσιες ενέργειες που οδηγούν σε σφάλματα του συστήματος, δοθέντος βέβαια ότι ο αριθμός των σφαλμάτων είναι κάτω από κάποιο όριο. Με αυτόν τον τρόπο το σύστημα δεν θα έχει κάποιο κεντρικό σημείο κατάρρευσης ούτε θα επηρεάζεται από την αποτυχία (εκούσια ή ακούσια) ενός (περιορισμένου) αριθμού κόμβων.

- **Μη πλαστογράφηση (Unforgeability)**. Οι κακόβουλοι κόμβοι δεν μπορούν να πλαστογραφήσουν τα πιστοποιητικά των κόμβων δοθέντος ότι το πλήθος τους είναι κάτω από ένα όριο.

- **Προληπτική ασφάλεια**. Το σύστημα πρέπει να παρέχει προστασία ενάντια σε κάποιον επιτιθέμενο που μπορεί να επιτίθεται και να διαβάλλει κόμβους του συστήματος κατά την διάρκεια μεγάλων χρονικών περιόδων. Συγκεκριμένα, το σύστημα δεν πρέπει να επιτρέπει στην επιτιθέμενη οντότητα να συνεργαστεί με κόμβους τους οποίους έχει διαβάλλει σε διαφορετικές, προγενέστερες χρονικές περιόδους.

- **Ασφαλής επικοινωνία**. Το σύστημα πρέπει να παρέχει στους κόμβους ασφαλή επικοινωνία. Συγκεκριμένα πρέπει να υποστηρίζεται εμπιστευτικότητα, ακεραιότητα, αυθεντικοποίηση και ταυτοποίηση της προέλευσης των μηνυμάτων μεταξύ των κόμβων του δικτύου.

4.4.3 Προδιαγραφές απόδοσης

Η ΥΔΚ πρέπει να αξιοποιεί όσο το δυνατόν περισσότερο τις δυνατότητες που παρέχουν τα δομημένα δίκτυα επικάλυψης και φυσικά να προσαρμόζεται στον τρόπο λειτουργίας τους. Ο τρόπος με τον οποίο επιτυγχάνεται αυτό είναι ικανοποιώντας τις παρακάτω προδιαγραφές:

- **Επεκτασιμότητα**. Το κόστος για τις υπηρεσίες πιστοποίησης-ανάκλησης και αποθήκευσης-αναζήτησης πρέπει να κλιμακώνεται ανάλογα με τον αριθμό των κόμβων στο σύστημα.

- **Κατανομή της λειτουργικότητας**. Όλες οι υπηρεσίες της ΥΔΚ πρέπει να υλοποιούνται από τους κόμβους του συστήματος. Επιπροσθέτως, οι λειτουργίες αυτές πρέπει να κατανέμονται με όσο το δυνατόν δικαιότερο τρόπο στους κόμβους ώστε όλοι να επιφορτίζονται ισότιμα με τις λειτουργίες. Η κατανομή των λειτουργιών μπορεί να διασπαστεί σε

  - **Εξισορρόπησή φόρτου της πιστοποίησης και της ανάκλησης πιστοποιητικών**. Στην ιδανική περίπτωση το κόστος της πιστοποίησης και της ανάκλησης των πιστοποιητικών πρέπει να διαμοιράζεται ισομερώς μεταξύ των κόμβων του συστήματος. Μια πιο χαλαρή, και πρακτικά υλοποιήσιμη, προδιαγραφή είναι η λειτουργία της πιστοποίησης των κόμβων να εναλλάσσεται μεταξύ ορισμένων κόμβων. Οι κόμβοι αυτοί, οι κόμβοι

πιστοποίησης, θα έχουν πρόσθετη λειτουργικότητα και ανά χρονικές περιόδους μια ομάδα από αυτούς θα εκτελεί την λειτουργία της πιστοποίησης. Με αυτόν τον τρόπο κατά την διάρκεια ζωής του συστήματος όλοι οι κόμβοι έχουν την δυνατότητα να εκτελέσουν τις λειτουργίες της πιστοποίησης (και της ανάκλησης)

  - **Εξισορρόπηση φόρτου της αποθήκευσης των πιστοποιητικών**. Τα πιστοποιητικά,και οι λίστες ανάκλησης, πρέπει να αποθηκεύονται ισότιμα σε όλους τους κόμβους του συστήματος.

- **Ανοχή στις συχνές εισαγωγές/αποχωρήσεις κόμβων**. Η υπηρεσία ΥΔΚ πρέπει να λειτουργεί ομαλά ακόμα και όταν οι κόμβοι του συστήματος έρχονται και φεύγουν από αυτό συχνά.

4.5 Chord-PKI: Υποδομή δημοσίου κλειδιού βασισμένη σε δίκτυα δομημένης επικάλυψης

Το Chord-PKI βασίζεται στο πρωτόκολλο δρομολόγησης του Chord [11], το οποίο και επεκτείνει για να υποστηρίζονται οι λειτουργίες της ΥΔΚ. Οι υπηρεσίες πιστοποίησης βασίζονται σε γνωστούς κρυπτογραφικούς μηχανισμούς οι οποίοι εξασφαλίζουν ισχυρές εγγυήσεις για την ασφάλεια του συστήματος και την ανοχή ενός συγκεκριμένου αριθμού κακόβουλων κόμβων και παράλληλα λαμβάνουν υπόψιν τους περιορισμούς που θέτει το κατανεμημένο περιβάλλον των δικτύων δομημένης επικάλυψης (πχ. το γεγονός ότι το μυστικό κλειδί πρέπει να δημιουργείται από ένα σύνολο κόμβων χωρίς κανένας από αυτούς να το γνωρίζει). Στην συνέχεια, αναλύονται οι κρυπτογραφικοί μηχανισμοί στους οποίους βασίζεται το Chord-PKI και περιγράφονται τα βασικά πρωτόκολλα με τα οποία ορίζονται οι λειτουργίες της ΥΔΚ που περιγράφηκαν στο 4.4.1 και ικανοποιούνται οι προδιαγραφές ασφάλειας (βλ. 4.4.2) και απόδοσης (βλ. 4.4.3).

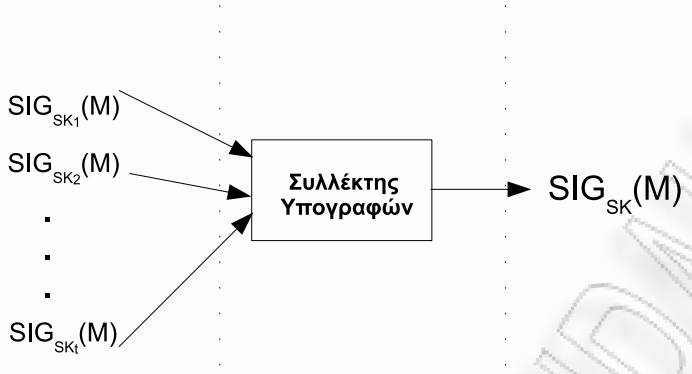
4.5.1 Κρυπτογραφικοί μηχανισμοί

Η βασική λειτουργία του Chord-PKI είναι η ασύμμετρη κρυπτογράφηση ή κρυπτογράφηση δημοσίου κλειδιού [77]. Το προτεινόμενο κρυπτοσύστημα βασίζεται

σε δύο βασικούς μηχανισμούς ασφαλείας, την *κρυπτογραφία κατωφλιού*–ΚΚ και στην *κρυπτογραφία πρόληψης*–ΚΠ. Η κρυπτογραφία κατωφλιού εξασφαλίζει ανοχή σε κακόβουλους κόμβους δοθέντος ότι ο αριθμός τους δε ξεπερνά το κατώφλι (ανώτερο όριο κακόβουλων κόμβων). Παράλληλα, λόγω του τρόπου με τον οποίο παράγεται το πιστοποιητικό στην ΚΚ υπάρχει κατανομή (τουλάχιστον της επεξεργαστικής ισχύος) της λειτουργίας της πιστοποίησης στους κόμβους που κατέχουν τα μέρη του μυστικού κλειδιού. Η κρυπτογραφία πρόληψης ανανεώνει προληπτικά τα μέρη του μυστικού κλειδιού, ώστε οι κακόβουλες οντότητες να μην έχουν την δυνατότητα να αποκτήσουν/διαβάλουν σταδιακά όλα τα μέρη του μυστικού κλειδιού.

4.5.1.1 Κρυπτογραφία κατωφλιού

Η κρυπτογραφία κατωφλιού (threshold cryptography) [78, 79, 80] παρέχει την δυνατότητα κατανομής μιας κρυπτογραφικής λειτουργίας, όπως η ψηφιακή υπογραφή, σε ένα σύνολο από έμπιστες οντότητες. Χρησιμοποιώντας ένα μηχανισμό κατωφλιού (t, n), το μυστικό κλειδί SK ενός ζευγαριού ιδιωτικού/δημοσίου κλειδιού PK, SK διαμοιράζεται μεταξύ n οντοτήτων. Οποιοδήποτε σύνολο αποτελούμενο από t ή περισσότερες οντότητες μπορούν να χρησιμοποιούν τα μερίδια τους από το μυστικό κλειδί SK προκειμένου να δημιουργήσουν την ψηφιακή υπογραφή, η επαλήθευση της οποίας γίνεται με το δημόσιο κλειδί PK. Αντίθετα, αν το σύνολο περιέχει λιγότερες από t οντότητες δεν είναι δυνατή η δημιουργία έγκυρης υπογραφής. Προκειμένου να υπογραφεί ένα μήνυμα M, τουλάχιστον t οντότητες χρησιμοποιούν τα μερίδια του μυστικού κλειδιού SK_j για να δημιουργήσουν τις *μερικές* υπογραφές $SIG_{SK_j}(M)$. Ο συμβολισμός $SK_j, j \in [1, n]$ περιγράφει το j-οστό μερίδιο του μυστικού κλειδιού. Οι μερικές υπογραφές του μηνύματος αποστέλλονται σε μια από τις t οντότητες, η οποία τις *συνδυάζει* με την δική της υπογραφή και παράγει την υπογραφή του μηνύματος M, $SIG_{SK}(M)$. Οποιαδήποτε από τις t οντότητες συλλέγει τις μερικές υπογραφές και τελικά παράγει το υπογεγραμμένο μήνυμα. Χρησιμοποιώντας το δημόσιο κλειδί PK οποιαδήποτε οντότητα, στην συγκεκριμένη περίπτωση η οντότητα που συλλέγει τις μερικές υπογραφές, μπορεί να επιβεβαιώσει την ορθότητα των παραγομένων υπογραφών. Στο σχήμα 4.2 παρουσιάζεται η παραπάνω διαδικασία.



Σχήμα 4.2: Κρυπτογραφία κατωφλιού (t, n): Διαδικασία υπογραφής μηνύματος M από t οντότητες

Αν η διαδικασία της υπογραφής αποτύχει, λόγω κάποιας λανθασμένης μερικής υπογραφής, επιλέγεται ένα καινούριο σύνολο από t οντότητες για να παραχθεί η υπογραφή και η διαδικασία επαναλαμβάνεται. Ο περιγραφόμενος βασικός μηχανισμός κρυπτογράφησης είναι ιδανικός για να εφαρμοστεί σε κατανεμημένα συστήματα καθώς οι αρμοδιότητες (π.χ. η υπογραφή ενός μηνύματος) διαμοιράζονται μεταξύ των κόμβων του συστήματος. Βέβαια, ο βασικός μηχανισμός δεν παρέχει αποδοτική λύση όταν υπάρχουν λανθασμένες (εκούσιες ή ακούσιες) μερικές υπογραφές ενός μηνύματος. Παραπάνω αναφέρθηκε ότι σε περίπτωση λανθασμένης μερικής υπογραφής όλη η διαδικασία επαναλαμβάνεται. Η επίπτωση σε ένα κατανεμημένο σύστημα είναι η αναμετάδοση κάποιων μηνυμάτων, δηλαδή η επιβάρυνση του δικτύου. Για την αποφυγή τέτοιων καταστάσεων μπορούν να χρησιμοποιηθούν *αξιόπιστοι* μηχανισμοί κρυπτογραφίας κατωφλιού [81], [82], [83]. Οι αξιόπιστοι μηχανισμοί κρυπτογραφίας κατωφλιού παρέχουν την δυνατότητα επαλήθευσης της ορθότητας των μερικών υπογραφών καθώς εμπεριέχουν λειτουργίες με τις οποίες εντοπίζονται οι λανθασμένες μερικές υπογραφές και ξεχωρίζονται από τις έγκυρες. Σε περίπτωση αποτυχίας της υπογραφής δεν χρειάζεται να επαναληφθεί η διαδικασία από την αρχή. Απλά πρέπει να δημιουργηθούν μόνο οι λανθασμένες μερικές υπογραφές με άμεση συνέπεια την αποστολή μηνυμάτων που αντιστοιχούν στην δημιουργία και συλλογή αυτών των μερικών υπογραφών. Οι μηχανισμοί που επαληθεύουν τις επιμέρους υπογραφές έχουν μεγαλύτερο υπολογιστικό κόστος σε σχέση με τους βασικούς μηχανισμούς αλλά στην περίπτωση των δικτυακών συστημάτων το επικοινωνιακό κόστος (αριθμός μηνυμάτων, καθυστέρηση επικοινωνίας) είναι σημαντικότερο.

Ο Perersen [84] επεκτείνει τον βασικό μηχανισμό κατωφλιού, ώστε ο υπολογισμός και ο διαμοιρασμός των μεριδίων του μυστικού κλειδιού να πραγματοποιείται χωρίς να υπάρχει μία έμπιστη οντότητα που να γνωρίζει το μυστικό κλειδί SK. Σε αυτήν την περίπτωση, όλα τα εμπλεκόμενα μέρη συμμετέχουν στην δημιουργία του μυστικού κλειδιού με τέτοιο τρόπο ώστε κανένας δεν γνωρίζει το SK.

4.5.1.2 Κρυπτογραφία πρόληψης

Η ασφάλεια που παρέχεται από την κρυπτογραφία κατωφλιού μπορεί να βελτιωθεί ανανεώνοντας προληπτικά τα μέρη μυστικού κλειδιού SK. Αν δεν υπάρχει ανανέωση των επιμέρους τμημάτων του μυστικού κλειδιού τότε ο επιτιθέμενος έχει την δυνατότητα να διαβάλλει το μυστικό κλειδί, αφού έχει άπειρο χρόνο στην διάθεσή του. Οι μηχανισμοί *προληπτικής ανανέωσης* [85], [86], [87] επιτρέπουν στις οντότητες που διατηρούν μέρη του μυστικού κλειδιού να τα ανανεώνουν χωρίς να αλλάξει το μυστικό κλειδί SK. Μετά την ανανέωση των μερών του κλειδιού SK, οι οντότητες ακυρώνουν τα προηγούμενα μέρη του κλειδιού και χρησιμοποιούν τα καινούργια μέρη του κλειδιού SK για να παράγουν τις μερικές υπογραφές. Επισημαίνεται ότι δεν είναι δυνατός ο συνδυασμός μερικών υπογραφών που έχουν δημιουργηθεί από μέρη του μυστικού κλειδιού που έχουν παραχθεί σε διαφορετικές χρονικές περιόδους. Συνοπτικά, η διαδικασία της ανανέωσης των τμημάτων του κλειδιού βασίζεται στο γεγονός ότι αν $SK_1, SK_2, \cdots, SK_n$ είναι τα μέρη του μυστικού κλειδιού SK και ότι $SK_1^{'}, SK_2^{'}, \cdots, SK_n^{'}$ είναι τα μέρη του μυστικού κλειδιού $SK^{'}$, τότε $SK_1 + SK_1^{'}, SK_2 + SK_2^{'}, \cdots, SK_n + SK_1^{'}$ είναι μέρη του μυστικού κλειδιού $SK + SK^{'}$. Αν $SK^{'} = 0$ τότε προκύπτει η καινούργια διαμοίραση του κλειδιού.

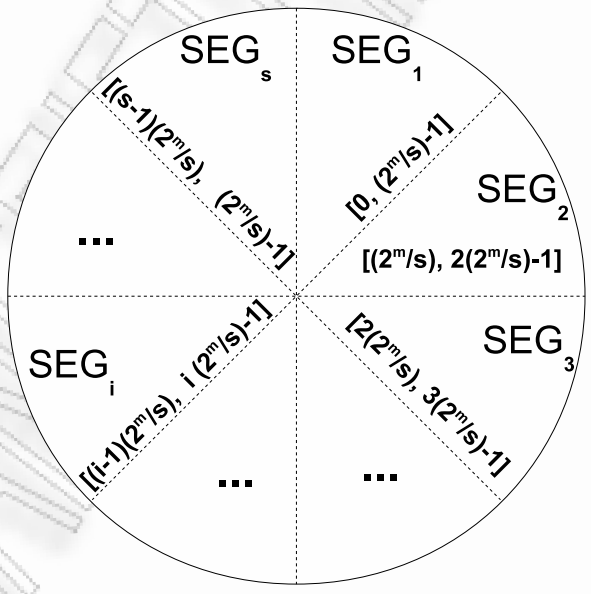
4.5.2 Προαπαιτούμενα και παραδοχές

Πριν από την έναρξη της λειτουργίας του Chord-PKI, εκτελείται ένα πρωτόκολλο αρχικοποίησης κατά το οποίο το κυκλικό πεδίο τιμών του Chord χωρίζεται σε s ίσα τμήματα. Ο αριθμός των τμημάτων-τομέων είναι γνωστός σε κάθε κόμβο του συστήματος. Επίσης, σε κάθε έναν από τους τομείς υπάρχουν τουλάχιστον n κόμβοι, δηλαδή υπάρχουν συνολικά στο σύστημα $n \times s$ κόμβοι. Τέλος κατά την διάρκεια του πρωτοκόλλου αρχικοποίησης όλοι οι συμμετέχοντες κόμβοι είναι έμπιστοι.

Το γεγονός ότι απαιτείται από τους αρχικούς κόμβους να είναι έμπιστοι είναι μία αρκετά αυστηρή παραδοχή, η οποία ενδεχομένως να περιορίζει το προτεινόμενο σύστημα. Βέβαια δεν είναι δυνατόν να *κατανεμηθεί* και να *διαχειριστεί* εμπιστοσύνη σε ένα σύστημα, χωρίς να υπάρχει εμπιστοσύνη. Στόχος του προτεινόμενου συστήματος είναι να διατηρήσει και ενδεχομένως να διαδώσει την εμπιστοσύνη που υπάρχει αρχικά στο σύστημα και σε νέους κόμβους, υπό την παρουσία κακόβουλων οντοτήτων.

4.5.3 Αρχικοποίηση συστήματος

Οι κόμβοι που βρίσκονται αρχικά στο σύστημα χωρίζουν το ιδεατό πεδίο τιμών του δικτύου επικάλυψης σε s *τομείς*. Αν το ιδεατό πεδίο τιμών είναι $[0, 2^m - 1]$ με m το μέγεθος της συνάρτησης κατακερματισμού (hash) του συστήματος, τότε το μέγεθος του κάθε τομέα είναι $2^m/s$. Συγκεκριμένα οι τομείς του συστήματος είναι $SEG_i = [(i-1)\cdot(2^m/s), i\cdot(2^m/s) - 1]$, $i \in [1, s]$. Η τιμή της παραμέτρου s καθορίζει τον αριθμό των τομέων στους οποίους χωρίζεται το (ιδεατό) κυκλικό πεδίο τιμών. Στο σχήμα 4.3 παρουσιάζεται η παραπάνω διαμέριση.



Σχήμα 4.3: Διαμέριση του κυκλικού πεδίου τιμών του Chord $[0, 2^m - 1]$ σε s περιοχές

Αν η συνάρτηση κατακερματισμού ήταν η $SHA-256$ τότε ενδεικτικές τιμές για το s είναι της τάξης $o(m)$. Για την ορθή λειτουργία του κάθε τομέα SEG_i πρέπει να υπάρχουν τουλάχιστον $n \geq 2t + 1$ κόμβοι, όπου t το όριο (κατώφλι) των έμπιστων

κόμβων. Οι κόμβοι αυτοί είναι οι αρχικοί *κόμβοι πιστοποίησης* του κάθε τομέα, οι οποίοι πιστοποιούν οποιοδήποτε άλλον κόμβο του τομέα για μια συγκεκριμένη χρονική περίοδο. Μετά την πάροδο της χρονικής αυτής περιόδου θα επιλεγούν νέοι κόμβοι πιστοποίησης.

Εφόσον διαμεριστεί σε s τομείς το πεδίο τιμών του δικτύου δομημένης επικάλυψης εκτελείται σε καθέναν τομέα SEG_i το *πρωτόκολλο αρχικοποίησης*, με το οποίο οι n κόμβοι πιστοποίησης δημιουργούν το ζευγάρι του δημοσίου/ιδιωτικού κλειδιού PK_i, SK_i, χρησιμοποιώντας ένα κρυπτοσύστημα δημοσίου κλειδιού (όπως το RSA ή το DSS). Το μυστικό κλειδί SK_i του κάθε τομέα χρησιμοποιείται για την πιστοποίηση των υπαρχόντων αλλά και των μελλοντικών κόμβων του τομέα SEG_i. Η διαδικασία αυτή περιγράφεται στο κεφάλαιο 4.5.4.1. Το δημόσιο κλειδί PK_i χρησιμοποιείται για την επαλήθευση του πιστοποιητικού (της υπογραφής) οποιουδήποτε κόμβου που έχει υπογραφεί με το SK_i. Το πρωτόκολλο αρχικοποίησης μπορεί να ενεργοποιηθεί ξεχωριστά για κάθε τομέα.

Το μυστικό κλειδί SK_i του τομέα SEG_i δημιουργείται από τους n κόμβους πιστοποίησης, οι οποίοι εφαρμόζουν έναν επαληθεύσιμο αλγόριθμο διαμοίρασης μυστικού κλειδιού, όπως το, Joint-Exp-RSS [88], κατά τον οποίο χρησιμοποιείται ένας διαμοιρασμός (t, n), με $n \geq 2t + 1$. Εφαρμόζοντας αυτόν τον αλγόριθμο το μυστικό κλειδί δεν είναι γνωστό από κανέναν κόμβο, ενώ χρειάζεται ένα σύνολο από τουλάχιστον t κόμβους για να δημιουργηθεί η υπογραφή με το SK_i. Λιγότεροι από t κόμβοι δεν είναι σε θέση να παράγουν έγκυρη υπογραφή.

Εναλλακτικά, και εφόσον οι αρχικοί κόμβοι πιστοποίησης του κάθε τομέα είναι έμπιστοι, ένας από αυτούς μπορεί να παίξει τον ρόλο του διαμοιραστή, να δημιουργήσει και να διανείμει τα μέρη του μυστικού κλειδιού καθώς και την απαραίτητη πληροφορία για την επαλήθευση και στην συνέχεια να διαγράψει το μυστικό κλειδί.

Στο τέλος του πρωτοκόλλου αρχικοποίησης, κάθε κόμβος πιστοποίησης $C_{i,j}, \in SEG_i, 1 \leq j \leq n$ θα έχει ένα μέρος $SK_{i,j}$ του μυστικού κλειδιού SK_i. Εφόσον ο διαμοιρασμός των μερών του μυστικού κλειδιού είναι επιτυχής, οποιοδήποτε σύνολο από t κόμβους θα δημιουργήσει το πιστοποιητικό του τομέα $CERT_i = SIG_{SK_i}(i, PK_i)$.

Μετά από την δημιουργία του δημόσιου/ιδιωτικού κλειδιού, οι κόμβοι πιστοποίησης του κάθε τομέα SEG_i πρέπει να διανείμουν το δημόσιο κλειδί PK_i στους κόμβους πιστοποίησης των άλλων τομέων. Αυτό πραγματοποιείται μέσω ενός *εξωτερικού καναλιού επικοινωνίας* μέσω του οποίου οι ενδιαφερόμενοι κόμβοι πιστοποίησης οποιουδήποτε τομέα μπορούν να αποκτήσουν το δημόσιο κλειδί PK_i του τομέα SEG_i. Προφανώς για την απόκτηση οποιουδήποτε άλλου δημοσίου κλειδιού (κάποιου τομέα) πρέπει να είναι γνωστό το αντίστοιχο κανάλι επικοινωνίας. Η λύση είναι εφικτή καθώς μόνο ένα ποσοστό των κόμβων, το οποίο φράσσεται ασυμπτωτικά από το m ($o(m)$), χρειάζεται να ανταλλάξει κλειδιά. Η χρήση κάποιου εξωτερικού καναλιού επικοινωνίας είναι ένας συνηθισμένος τρόπος που χρησιμοποιείται για την αρχικοποίηση (bootstrapping) συστημάτων ομότιμων κόμβων. Στο πρωτόκολλο του Chord [11] η εισαγωγή ενός νέου κόμβου στο σύστημα προϋποθέτει ότι αυτός γνωρίζει (από κάποιο εξωτερικό κανάλι επικοινωνίας) κάποιον κόμβο του συστήματος. Παρόμοια στο $eMule$ [13] ο κόμβος πρέπει να γνωρίζει την λίστα με τους εξυπηρετητές που μπορεί να συνδεθεί.

Εναλλακτικά, κάποιος κόμβος του συστήματος, έστω ο I μπορεί να θεωρηθεί ως το σημείο σύνδεσης του συστήματος με όλους τους άλλους κόμβους κατά την διάρκεια της εκκίνησης. Όλοι οι κόμβοι πιστοποίησης αποκτούν τα δημόσια κλειδιά των άλλων τομέων μέσου του κόμβου I.

Εφόσον όλοι οι κόμβοι πιστοποίησης έχουν την λίστα με τα δημόσια κλειδιά όλων των τομέων, $PK_1, PK_2, \cdots, PK_s$, την υπογράφουν, δημιουργώντας με αυτόν τον τρόπο την πιστοποιημένη λίστα με τα δημόσια κλειδιά των τομέων του συστήματος, $CL_i = SIG_{SK_i}(PK_1, PK_2, \cdots, PK_s)$. Η CL_i αποθηκεύεται σε κάθε κόμβο πιστοποίησης του τομέα $SEG_i, 1 \leq i \leq s$ καθώς και στον κόμβο του δικτύου επικάλυψης που διαχειρίζεται το αναγνωριστικό $H(CERT_i)$, με H την συνάρτηση κατακερματισμού του δικτύου επικάλυψης.

Τέλος σε κάθε τομέα του συστήματος, οι κόμβοι πιστοποίησης διατηρούν μια λίστα $List_i$ που περιέχει:

- Τις διευθύνσεις δικτύου (IP) των κόμβων πιστοποίησης.
- Την ημερομηνία και τον χρόνο λήξης της περιόδου διαμοίρασης του κλειδιού SK_i.

- Τα όρια του τομέα. Δηλαδή το αναγνωριστικό του δικτύου επικάλυψης από το οποίο ξεκινά ο τομέας και το αναγνωριστικό στο οποίο τελειώνει.

Η λίστα έχει την μορφή
$List_i = [ID(Ci,j), IP(Ci,j), ThreshExpTime, ID_{start-i}, ID_{end-i}], \forall j \in [1,n]$.

4.5.4 Λειτουργίες συστήματος

Εφόσον αρχικοποιηθεί το σύστημα και μοιραστούν τα ζεύγη κλειδιών ανά τομέα το Chord-PKI παρέχει τις παρακάτω λειτουργίες:

4.5.4.1 Πιστοποίηση κόμβων

Για να πιστοποιηθεί ένας κόμβος του δικτύου δομημένης επικάλυψης, έστω N, εκτελείται το παρακάτω πρωτόκολλο:

1. **Αναγνώριση τομέα**. Αρχικά, ο N υπολογίζει σε ποιόν τομέα SEG_i ανήκει. Με βάση το αναγνωριστικό του N, και τον αριθμό των τομέων s στο σύστημα ο κόμβος εύκολα υπολογίζει τον τομέα που ανήκει.

2. **Δημιουργία δημόσιου/ιδιωτικού κλειδιού** . Ο N δημιουργεί το ζευγάρι των κλειδιών του (pk_N, sk_N) χρησιμοποιώντας κάποιον αλγόριθμο δημοσίου κλειδιού όπως ο RSA ή ο DSS.

3. **Εύρεση κόμβου πιστοποίησης**. Ο N βρίσκει κάποιον από τους κόμβους πιστοποίησης του τομέα SEG_i σύμφωνα με το πρωτόκολλο που περιγράφεται στο 4.5.4.5. Έστω ότι αυτός είναι ο $C_{i,1}$, δηλαδή ο κόμβος πιστοποίησης που είναι υπεύθυνος για το μέρος $SK_{i,1}$ του μυστικού κλειδιού SK_i.

4. **Έκδοση πιστοποιητικού** Ο N στέλνει στον $C_{i,1}$ το δημόσιο κλειδί του, μια απόδειξη γνώσης του μυστικού κλειδιού sk_N, και μια αίτηση για την δημιουργία του πιστοποιητικού (Η σύνταξη της αίτησης βασίζεται στο πρότυπο PKCS#10 [89]). Ο $C_{i,1}$ δέχεται ή απορρίπτει την αίτηση του N. Σε περίπτωση θετικής απάντησης ο $C_{i,1}$ λειτουργεί ως συλλέκτης του μηχανισμού κατωφλιού και εκδίδει το πιστοποιητικό για τον N. Για αυτόν τον λόγο ο

συλλέκτης αρχικά υπογράφει την αίτηση με δικό του τμήμα του μυστικού κλειδιού, $SIG_{SK_{i,1}}(i, N, pk_N)$. Επιπλέον, επικοινωνεί με $t-1$ κόμβους πιστοποίησης του τομέα του, οι επιλέγονται τυχαία από την λίστα $List_i$, και ζητά από κάθε κόμβο να υπογράψει το μήνυμα (i, N, pk_N) με το αντίστοιχο μέρος του μυστικού κλειδιού. Εφόσον υπογράψουν το μήνυμα και οι υπόλοιποι $t-1$ κόμβοι, ο $C_{i,1}$ συλλέγει τα υπογεγραμμένα μηνύματα και δημιουργεί ένα πιστοποιητικό $cert_j = SIG_{SK_i}(i, N, pk_N)$ για τον κόμβο N.

5. **Επαλήθευση πιστοποιητικού** Ο συλλέκτης $C_{i,1}$ ελέγχει την εγκυρότητα του πιστοποιητικού. Σε περίπτωση που το πιστοποιητικό δεν είναι έγκυρο επαναλαμβάνεται το προηγούμενο βήμα με άλλο υποσύνολο κόμβων πιστοποίησης. Σε περίπτωση που χρησιμοποιηθεί κάποιος αξιόπιστος μηχανισμός κατωφλιού τότε οποιοδήποτε σφάλμα στην δημιουργία του πιστοποιητικού μπορεί να ανιχνευθεί και να βρεθεί το μέρος της υπογραφής που έχει το σφάλμα (το οποίο μπορεί να έχει προέλθει από κακόβουλη ενέργεια).

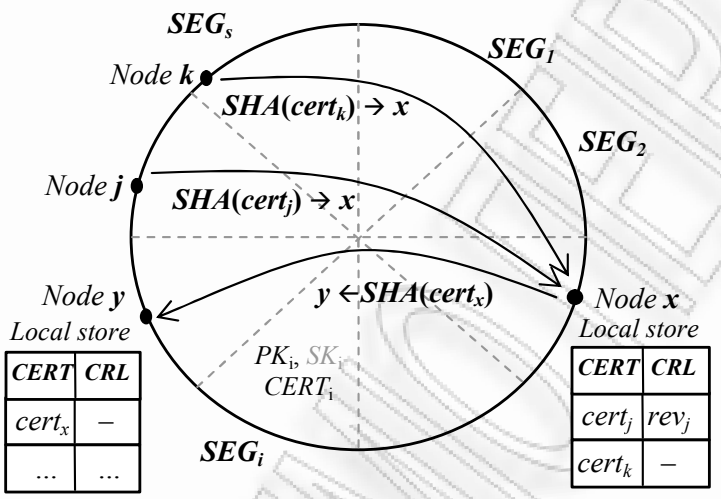
6. **Παράδοση πιστοποιητικού** Ο συλλέκτης στέλνει το πιστοποιητικό στον κόμβο N. Ο N αποθηκεύει το πιστοποιητικό $cert_N$ και τοποθετεί στην προσωρινή μνήμη του (cache) τον κόμβο πιστοποίησης $C_{i,1}$ για μελλοντική επικοινωνία.

4.5.4.2 Ανάκληση πιστοποιητικών κόμβων

Οι κόμβοι πιστοποίησης ενός τομέα i δέχονται αιτήσεις ανάκλησης για κάποιο πιστοποιητικό που αφορά κόμβους που ανήκουν στον τομέα ευθύνης τους. Μία αίτηση ανάκλησης ενός πιστοποιητικού είναι έγκυρη μόνο αν είναι υπογεγραμμένη από κάποιον πιστοποιημένο κόμβο, ο οποίος μπορεί να ανήκει σε οποιοδήποτε τομέα. Οι κόμβοι πιστοποίησης περιοδικά εξετάζουν τις αιτήσεις ανάκλησης και αποφασίζουν για τα πιστοποιητικά που θα ανακληθούν. Παρακάτω αναλύονται τα βήματα του πρωτοκόλλου ανάκλησης των πιστοποιητικών.

1. **Επιλογή συνόλου ανάκλησης.** Ο κόμβος πιστοποίησης $C_{i,j}$ λειτουργεί ως συλλέκτης για την διαδικασία της ανάκλησης. Επιλέγει $t-1$ κόμβους πιστοποίησης από την λίστα $List_i$

2. **Υπογραφή ανάκλησης.** Το σύνολο των κόμβων ανάκλησης θα υπογράψουν το μήνυμα ανάκλησης $rev_N = SIG_{SK_i}(cert_N, RevTime)$, όπου $RevTime$ είναι η ώρα της ανάκλησης. Το μήνυμα της ανάκλησης θα αποθηκευτεί στη *λίστα ανάκλησης πιστοποιητικών* βάσει του πρωτοκόλλου που αναλύεται στην ενότητα 4.5.4.3.



Σχήμα 4.4: Σύστημα Chord-PKI: αποθήκευση και αναζήτηση πιστοποιητικών και λιστών ανάκλησης κόμβων

4.5.4.3 Αποθήκευση πιστοποιητικών και λιστών ανάκλησης

Η διαδικασία της αποθήκευσης των πιστοποιητικών και των λιστών ανάκλησης κατανέμεται σε όλους τους κόμβους του Chord-PKI όπως φαίνεται στο σχήμα 4.4. Κάθε κόμβος στο σύστημα διαθέτει αποθηκευτικό χώρο ώστε να αποθηκεύει πιστοποιητικά και λίστες ανάκλησης πιστοποιητικών. Οι κόμβοι στους οποίους αποθηκεύονται τα πιστοποιητικά και οι λίστες ανάκλησης βασίζεται στον τρόπο που αντιστοιχίζονται τα κλειδιά στους κόμβους του δικτύου επικάλυψης του Chord. Συγκεκριμένα το πρωτόκολλο της αποθήκευσης περιλαμβάνει από τα παρακάτω βήματα:

1. **Δημιουργία πιστοποιητικού**. Ο κόμβος N αποκτά ένα πιστοποιητικό $cert_N$ (βλ. 4.5.4.1).

2. **Αποθήκευση πιστοποιητικού**. Ο κόμβος N στέλνει στον κόμβο Y_1 το πιστοποιητικό του $cert_N$. Έστω ότι ο Y_1 είναι ο διαχειριστής του αναγνωριστικού

Η$(cert_N, 1)$. Σύμφωνα με το πρωτόκολλο του Chord η διαδικασία αυτή αποτελείται από δύο επί μέρους βήματα, δηλαδή την αναγνώριση του διαχειριστή Y_1 (DHT get) και στην συνέχεια την αποθήκευση του πιστοποιητικού (DHT put). Με αυτόν τον τρόπο κάθε κόμβος που θέλει να επαληθεύσει το πιστοποιητικό κάποιου άλλου κόμβου, γνωρίζει εκ των προτέρων την τοποθεσία αποθήκευσής του.

3. **Αποθήκευση λίστας ανάκλησης**. Σε περίπτωση που το πιστοποιητικό $cert_N$ ανακληθεί από τους κόμβους πιστοποίησης το $cert_N$ θα αποθηκευτεί στην αποθήκη του κόμβου Y_1, το διαχειριστή του αναγνωριστικού Η$(cert_N, 1)$. Οποιοσδήποτε κόμβος θέλει να ελέγξει την εγκυρότητα του πιστοποιητικού θα πρέπει να ελέγξει αν υπάρχει το $cert_N$ στη λίστα ανάκλησης του κόμβου Y_1.

4. **Πλεονασμός**. Το Chord-PKI είναι ένα κατανεμημένο σύστημα στο οποίο κόμβοι έρχονται και φεύγουν από το σύστημα. Για να αυξηθεί η *διαθεσιμότητα* των δεδομένων το πιστοποιητικό $cert_N$ ή η ανάκληση του πιστοποιητικού rev_N αποθηκεύεται σε ένα πλήθος από t κόμβους. Τα αντίγραφα των πιστοποιητικών αποθηκεύονται στους κόμβους του συστήματος που διαχειρίζονται τα αναγνωριστικά που προκύπτουν από τον τύπο Η$(cert_N, i) \rightarrow Y_i$, $i = 1, \cdots, t$, όπου t το όριο του σχήματος κατωφλιού.

4.5.4.4 Ανάκτηση και Επαλήθευση πιστοποιητικών και λιστών ανάκλησης πιστοποιητικών

Η ανάκτηση ενός πιστοποιητικού ή μια λίστας ανάκλησης είναι παρόμοια με την λειτουργία της εύρεσης αναγνωριστικών (get) του πρωτοκόλλου Chord. Κάθε κόμβος του συστήματος μπορεί να ανακτήσει και να ελέγξει την ορθότητα οποιουδήποτε πιστοποιητικού (οποιουδήποτε τομέα), όπως φαίνεται στο σχήμα 4.4.

1. **Εύρεση του κόμβου αποθήκευσης**. Ο τυχαίος κόμβος X βρίσκει το αναγνωριστικό id που αντιστοιχεί στο πιστοποιητικό $cert_N$, δηλ. $id = H(cert_N, 1)$ και βρίσκει τον διαχειριστή κόμβο του αναγνωριστικού id, έστω Y_1.

2. **Εύρεση του πιστοποιητικού στην τοπική αποθήκη**. Ο X αρχικά ρωτά τον Y_1 αν περιέχεται στην αποθήκη ανάκλησής του το πιστοποιητικό. Σε περίπτωση θετικής απάντησης η διαδικασία τερματίζεται καθώς το πιστοποιητικό

έχει ανακληθεί. Διαφορετικά, ζητείται από τον Y_1 το πιστοποιητικό. Σε περίπτωση που το πιστοποιητικό υπάρχει εκτελείται το βήμα 4, διαφορετικά το 3.

3. **Εύρεση του πιστοποιητικού**. Τη θέση του Y_1 του προηγούμενου βήματος παίρνουν διαδοχικά οι κόμβοι $Y_i \rightarrow H(cert_N, i), i = 2, \cdots, t$ και εκτελείται η ίδια διαδικασία ώσπου να βρεθεί το πιστοποιητικό (ή να βρεθεί κάποια ανάκληση για το πιστοποιητικό αυτό). Σε αυτή την περίπτωση η διαδικασία συνεχίζεται στο βήμα 4. Αν μετά από τον έλεγχο των t κόμβων δεν έχει βρεθεί το πιστοποιητικό ο X απορρίπτει το πιστοποιητικό $cert_N$.

4. **Επαλήθευση πιστοποιητικού**. Ο X ελέγχει την ορθότητα του πιστοποιητικού $cert_N$ χρησιμοποιώντας το πιστοποιητικό του τομέα που ανήκει ο N. Παράλληλα ελέγχει την ημερομηνία λήψης του πιστοποιητικού. Αν οι έλεγχοι είναι επιτυχείς ο X αποδέχεται το πιστοποιητικό $cert_N$.

5. **Έλεγχος ανάκλησης πιστοποιητικού**. Ο κόμβος X ελέγχει τις αποθήκες των λιστών ανάκτησης των υπολοίπων t κόμβων αν υπάρχει κάποια ανάκληση για το πιστοποιητικό $cert_N$. Το $cert_N$ θεωρείται έγκυρο αν δεν υπάρχει ανάκληση σε κανέναν από τους t κόμβους.

4.5.4.5 Εύρεση κόμβου πιστοποίησης σε έναν τομέα

Η πιστοποίηση του δημοσίου κλειδιού (ή η ανάκληση ενός εκδοθέντος πιστοποιητικού) προϋποθέτει την επικοινωνία μεταξύ ενός κόμβου του συστήματος και ενός κόμβου πιστοποίησης $C_{i,j}$. Η επικοινωνία περιγράφεται από το παρακάτω πρωτόκολλο, στο οποίο ένας κόμβος N του Chord-PKI θέλει να επικοινωνήσει με έναν κόμβο πιστοποίησης του τομέα SEG_i

1. **Υπολογισμός Τομέα**. Ο κόμβος N υπολογίζει τα όρια του τομέα ευθύνης του $SEG_i : [(i-1)(2^m/s), i(2^m/s) - 1]$

2. **Τυχαία επιλογή**. Ο κόμβος N επιλέγει ένα τυχαίο αναγνωριστικό $[(i-1)(2^m/s) \leq id \leq i(2^m/s) - 1]$ και επιλέγει τον διαχειριστή του κόμβου $X \rightarrow successor(id)$.

3. **Αίτηση μετάδοσης**. Ο κόμβος N στέλνει ένα μήνυμα στον X, ζητώντας κάποιον κόμβο πιστοποίησης που ανήκει στον τομέα του.

4. **Εύρεση κόμβου πιστοποίησης** Αν κόμβος X είναι:

   (α′) Κόμβος πιστοποίησης. Σε αυτήν την περίπτωση ο X επιστρέφει τη διεύθυνση του στον N.

   (β′) Ένας πιστοποιημένος κόμβος του συστήματος. Σε αυτήν την περίπτωση ο X ελέγχει αν λειτουργούν οι κόμβοι πιστοποίησης που γνωρίζει. Ο πρώτος λειτουργικός κόμβος πιστοποίησης επιστρέφεται στον N. Σε διαφορετική περίπτωση εκτελείται το βήμα 4γ′

   (γ′) Ένας κόμβος που δεν έχει πιστοποιηθεί και ο οποίος δεν γνωρίζει την ύπαρξη κάποιου κόμβου πιστοποίησης. Σε αυτήν την περίπτωση ο X προωθεί την αίτηση στον άμεσο γείτονά του (σύμφωνα με τον αλγόριθμο δρομολόγησης του Chord) και το πρωτόκολλο συνεχίζει με το βήμα 3

5. **Αίτηση για πιστοποίηση** Ο N στέλνει μια αίτηση για πιστοποίηση στον κόμβο του προηγούμενου βήματος.

4.5.4.6 Επικοινωνία των κόμβων πιστοποίησης ενός τομέα

Όλοι οι κόμβοι πιστοποίησης ενός τομέα επικοινωνούν μεταξύ τους μέσω της λίστας $List_i$ που έχει δημιουργηθεί στην αρχικοποίηση του συστήματος (βλ. 4.5.3).

4.5.4.7 Επικοινωνία των κόμβων πιστοποίησης διαφορετικών τομέων

Ο κόμβος πιστοποίησης $C_{i,j}$ (του τομέα SEG_i) επικοινωνεί με τον κόμβο πιστοποίησης $C_{k,j}$ (του τομέα SEG_k) με βάση το πρωτόκολλο 4.5.4.5.

4.5.4.8 Ανανέωση μυστικού κλειδιού ανά τομέα

Κάθε ένας από τους κόμβους πιστοποίησης ενός τομέα SEG_i διατηρεί κάποιο μέρος του μυστικού κλειδιού του τομέα για μια συγκεκριμένη χρονική περίοδο. Στο τέλος της περιόδου αυτής οι κόμβοι πιστοποίησης ακολουθούν ένα πρωτόκολλο ανανέωσης με το οποίο αλλάζουν οι κόμβοι πιστοποίησης και ανανεώνονται τα τμήματα του μυστικού κλειδιού.

Έλεγχος διαθεσιμότητας κόμβων: Για να υλοποιηθεί το πρωτόκολλο της ανανέωσης του (μυστικού) κλειδιού ανά τομέα θα πρέπει να υπάρχουν τουλάχιστον t διαθέσιμοι κόμβοι πιστοποίησης σε έναν τομέα SEG_i. Αν οι διαθέσιμοι κόμβοι είναι λιγότεροι από n αλλά πάντοτε περισσότεροι από t (t, n είναι οι παράμετροι του μηχανισμού κατωφλιού) οι υπάρχοντες κόμβοι πιστοποίησης δημιουργούν τα μέρη του μυστικού κλειδιού που λείπουν. Παρόμοια διαδικασία ακολουθείται όταν κάποιοι κόμβοι εγκαταλείψουν το δίκτυο κατά την διάρκεια της ανανέωσης. Δηλαδή οι κόμβοι που εγκαταλείπουν θεωρούνται απόντες και οι υπόλοιποι κόμβοι δημιουργούν τα μέρη του κλειδιού που τους αντιστοιχούσαν. Στο τέλος του ελέγχου θα υπάρχουν διαθέσιμα, σε τουλάχιστον t υπολογιστές, τα n μέρη του μυστικού κλειδιού.

Ανανέωση των κόμβων πιστοποίησης: Οι κόμβοι πιστοποίησης κάθε τομέα επιλέγουν το σύνολο των νέων n κόμβων πιστοποίησης για την επόμενη περίοδο με τον ακόλουθο τρόπο. Κάθε κόμβος πιστοποίησης παρέχει ένα τυχαίο δεδομένο $R_j,\ j = 1, \cdots, t$. Όλα τα τυχαία δεδομένα χρησιμοποιούνται από την συνάρτηση κατακερματισμού H για την παραγωγή του αναγνωριστικού του υποψηφίου κόμβου. ($ID = H(R_1, \cdots, R_n)$). Αν το αναγνωριστικό δεν ανήκει στο τομέα των κόμβων πιστοποίησης η διαδικασία επαναλαμβάνεται μέχρι να βρεθεί το κατάλληλο αναγνωριστικό.

Ο διαχειριστής κόμβος του αναγνωριστικού ID είναι ένας υποψήφιος κόμβος πιστοποίησης. Αν ο κόμβος αυτός είναι πιστοποιημένος και αποδεχτεί τον ρόλο του τότε έχει βρεθεί ένας από τους ζητούμενους n. Η ίδια διαδικασία επαναλαμβάνεται μέχρι να βρεθούν n κόμβοι.

Κάθε κόμβος της προηγούμενης προηγούμενης περιόδου αντιστοιχίζεται σε έναν νέο κόμβο. Ο παλιός κόμβος πιστοποίησης κρυπτογραφεί το μέρος του μυστικού του κλειδιού SK_i με το δημόσιο κλειδί του νέου κόμβου, το οποίο αποστέλλει.

Προληπτική ανανέωση: Οι νέοι κόμβοι πιστοποίησης ανανεώνουν τα μέρη του μυστικού κλειδιού με βάση κάποιον αλγόριθμο ανανέωσης [85].

4.6 Ανάλυση ασφάλειας

Το Chord-PKI παρέχει υπηρεσίες ασφάλειας στους κόμβους του Chord. Η ασφάλεια του Chord-PKI είναι αρκετά σημαντική καθώς υπάρχουσες αδυναμίες θα οδηγήσουν στην αποτυχία των παρεχόμενων υπηρεσιών ασφάλειας. Στην συνέχεια ορίζεται το *πλάνο επίθεσης* (Adversary Model), όπου περιγράφονται οι δυνατότητες και οι περιορισμοί της επιτιθέμενης οντότητας και στην συνέχεια αναλύεται πώς επιτυγχάνονται οι απαιτήσεις ασφάλειας του συστήματος (βλ.4.4.2)

4.6.1 Πλάνο επίθεσης

Το πλάνο επίθεσης (adversary model)επιτρέπει στην κακόβουλη οντότητα να διαβάλλει τα μηνύματα επικοινωνίας (των πρωτοκόλλων) (Byzantine Adversary). Επιπλέον, όλες οι οντότητες στο σύστημα, συμπεριλαμβανομένης και της κακόβουλης οντότητας έχουν περιορισμένους πόρους στην διάθεσή τους (polynomially bounded). Στο Chord-PKI υπάρχουν δύο ειδών οντότητες, οι *έμπιστες* και οι *μη–έμπιστες*. Οι έμπιστες οντότητες συμπεριφέρονται όπως ορίζεται στα πρωτόκολλα που έχουν περιγραφεί παραπάνω. Η συμπεριφορά των μη-έμπιστων οντοτήτων αποκλίνει από τα πρωτόκολλα. Μια μη-έμπιστη οντότητα που προσπαθεί να υπονομεύσει την ασφάλεια του Chord-PKI ονομάζεται *κακόβουλη*. Η κακόβουλη οντότητα μπορεί να εμποδίσει την παράδοση ενός μηνύματος ή να διαβάλλει την διαδικασία πιστοποίησης. Οι μη–έμπιστες οντότητες των οποίων τα μυστικά κλειδιά είναι στην διάθεση της επιτιθέμενης οντότητας ονομάζονται *εκτεθειμένες*.

Οι μη–έμπιστες οντότητες ελέγχονται από τον αντίπαλο (adversary) ο οποίος μπορεί να είναι είτε ενεργητικός είτε παθητικός. Ο παθητικός μπορεί να υποκλέπτει μηνύματα που ανταλλάσσονται μεταξύ των οντοτήτων του συστήματος, ενώ αντίθετα ο ενεργητικός αντίπαλος μπορεί να διαβάλλει την επικοινωνία μεταξύ των οντοτήτων είτε αλλοιώνοντας τα μηνύματα είτε εισάγοντας δικά του μηνύματα. Υποθέτουμε ότι ο αριθμός των μη-έμπιστων οντοτήτων στο σύστημα είναι λιγότερος από t, το κατώφλι που ορίζεται στο 4.5.1.1, και είναι παράμετρος που ρυθμίζεται στην αρχή του συστήματος για κάθε τομέα.

4.6.2 Ασφάλεια του Chord PKI

Οι προδιαγραφές ασφάλειας που ορίστηκαν στο κεφάλαιο 4.4.2 επιτυγχάνονται με την χρήση της κρυπτογραφίας κατωφλιού και με την χρήση τεχνικών πλεονασμού. Συγκεκριμένα

4.6.2.1 Διαθεσιμότητα

Το πλάνο επίθεσης ορίζει ότι ο αντίπαλος μπορεί να εμποδίσει την παράδοση των μηνυμάτων για έναν περιορισμένο αριθμό κόμβων. Ωστόσο με τεχνικές πλεονασμού, τόσο στο επίπεδο της δρομολόγησης μέσω της λίστας με επιπλέον γείτονες [11], όσο και στο επίπεδο δεδομένων, το σύστημα επιτρέπει σε όλους τους έμπιστους κόμβους να ανταλλάξουν και να ανακτήσουν πιστοποιητικά.

Τα πιστοποιητικά, όπως επίσης και οι λίστες ανάκλησης πιστοποιητικών, αποθηκεύονται σε t διαφορετικούς κόμβους. Συγκεκριμένα, το πιστοποιητικό $cert_N$ αποθηκεύεται στους κόμβους που διαχειρίζονται τα αναγνωριστικά $ID_i = H(cert_N, i), i = 1, \cdots, t$. Σε περίπτωση που κάποιος κόμβος αρνηθεί να επιστρέψει κάποιο πιστοποιητικό από την τοπική αποθήκη του τότε ο αιτούμενος κόμβος μπορεί να δοκιμάσει τους εναλλακτικούς κόμβους. Ο αντίπαλος δεν μπορεί εύκολα να ελέγχει όλους τους εναλλακτικούς κόμβους αφού αντιστοιχούν σε αναγνωριστικά που παράγονται από την τυχαία συνάρτηση κατακερματισμού. Επίσης, το πρωτόκολλο δρομολόγησης του Chord παρέχει ανοχή σε σφάλματα (κακόβουλα και μη) μέσα σε κάποια λογικά πλαίσια.

4.6.2.2 Ανοχή σε μη-έμπιστους κόμβους

Θεώρημα 1: Ανοχή κατά την διάρκεια ανανέωσης των κόμβων πιστοποίησης

Αν έστω και ένας κόμβος πιστοποίησης είναι έμπιστος κατά την διάρκεια ανανέωσης του συνόλου τους τότε ο αντίπαλος δεν μπορεί να καθορίσει το νέο σύνολο με τους κόμβους πιστοποίησης

Απόδειξη:

Κατά το πρωτόκολλο ανανέωσης του κλειδιού, που περιγράφηκε στο 4.5.4.8, οι κόμβοι πιστοποίησης της προηγούμενης περιόδου συμμετέχουν στην επιλογή των νέων κόμβων. Κάθε κόμβος της προηγούμενης περιόδου συνεισφέρει τυχαία δεδομένα R_j τα οποία αποτελούν είσοδο στην συνάρτηση κατακερματισμού H, η οποία παράγει το αναγνωριστικό ID. Ο αντίπαλος δεν μπορεί να επιλέξει υποψήφιο κόμβο καθώς το αναγνωριστικό ID είναι τυχαίο και έχει παραχθεί από στατιστικά ανεξάρτητα τυχαία δεδομένα.

Θεώρημα 2: Ανοχή κατά την διάρκεια διαφορετικών περιόδων ανανέωσης

Αν ο αριθμός των μη-έμπιστων κόμβων είναι λιγότερος από t τότε η διαδικασία της πιστοποίησης (και της ανάκλησης των πιστοποιητικών) δεν μπορεί να ελεγχθεί από τον αντίπαλο.

Απόδειξη:

Αυτό προκύπτει από την χρήση της κρυπτογραφίας κατωφλιού (t, n). Βάσει σχεδιασμού στην κρυπτογραφία κατωφλιού λιγότεροι από t κόμβοι δεν μπορούν να παράγουν έγκυρα πιστοποιητικά.

4.6.2.3 Μη πλαστογράφηση

Η πλαστογράφηση ενός πιστοποιητικού προϋποθέτει ότι ο αντίπαλος ελέγχει τουλάχιστον t κόμβους πιστοποίησης σε έναν τομέα κατά την τρέχουσα περίοδο διαμοίρασης του κλειδιού. Ωστόσο στο πλάνο επίθεσης υποθέτουμε ότι ο αντίπαλος δεν ελέγχει περισσότερους από t μη-έμπιστους κόμβους. Δηλαδή ο αντίπαλος δεν μπορεί να δημιουργήσει υπογραφές με το μυστικό κλειδί (του τομέα) και να πλαστογραφήσει κάποιο πιστοποιητικό.

4.6.2.4 Προληπτική ασφάλεια

Το Chord-PKI παρέχει μηχανισμούς προληπτικής ασφάλειας αφού δεν επιτρέπει σε έναν ενεργό αντίπαλο να συνδυάσει μέρη του μυστικού κλειδιού (ενός τομέα) που έχουν εκτεθεί σε διαφορετικές περιόδους διαμοίρασης του κλειδιού.

Θεώρημα 3: Περιορισμένος χρόνος επίθεσης

Ο αντίπαλος μπορεί να πλαστογραφήσει ένα πιστοποιητικό μόνο αν καταφέρει και ελέγξει περισσότερους από t κόμβους πιστοποίησης στην ίδια περίοδο.

Απόδειξη:

Κατά την διαδικασία ανανέωσης του μυστικού κλειδιού ενός τομέα (βλ. 4.5.4.8) οι παλιοί κόμβοι ανανεώνουν τα τμήματα του μυστικού κλειδιού του τομέα. Τα καινούρια τμήματα του κλειδιού μεταφέρονται με ασφάλεια στους καινούργιους κόμβους πιστοποίησης. Σε κάθε νέο κόμβο αντιστοιχεί ένα μόνο μέρος του μυστικού κλειδιού. Με αυτόν τον τρόπο, κάποια κακόβουλη οντότητα που έχει διαβάλλει το σύστημα, δηλαδή έχει ανακαλύψει τα μέρη του μυστικού κλειδιού για την περίοδο i, δεν μπορεί να συνδυάσει τα μέρη αυτά σε μια άλλη περίοδο $j > i$ για να υπογράψει ένα μήνυμα (δημιουργία πιστοποιητικού). Ο μοναδικός τρόπος με τον οποίο μπορεί μια κακόβουλη οντότητα να πλήξει την ασφάλεια του συστήματος είναι αποκτώντας t μερικές υπογραφές (σε έναν τομέα) στην διάρκεια μιας περιόδου.

4.7 Αξιολόγηση απόδοσης Chord-PKI

Στις προηγούμενες ενότητες αναφέρθηκαν οι βασικές προϋποθέσεις για την λειτουργία του Chord-PKI, αναλύθηκαν τα πρωτόκολλα λειτουργίας του και περιγράφηκε πώς επιτυγχάνεται η ασφάλεια με βάση το πλάνο επίθεσης της ενότητας 4.6.1. Σε αυτήν την ενότητα αξιολογούμε την επίδοση του συστήματος. Συγκεκριμένα, εξετάζονται:

1. Η ανοχή του συστήματος σε κακόβουλους κόμβους.
2. Το υπολογιστικό κόστος που έχουν οι κρυπτογραφικές πράξεις του Chord-PKI
3. Το επικοινωνιακό κόστος (σε αριθμό μηνυμάτων) που εισάγουν οι κρυπτογραφικές υπηρεσίες.

Για τα τρία αυτά χαρακτηριστικά αναπτύσσουμε στις τρεις επόμενες υποενότητες μεθοδολογίες και πειράματα με τα οποία παρουσιάζεται η απόδοση (κόστος) του Chord-PKI.

4.7.1 Ανοχή σε μη-έμπιστους κόμβους

Η βασικότερη λειτουργία που προσφέρει το Chord-PKI είναι η έκδοση πιστοποιητικών. Με βάση τα πιστοποιητικά οι κόμβοι μπορούν να έχουν πρόσβαση σε άλλες υπηρεσίες ασφαλείας υψηλότερου επιπέδου. Είναι σημαντικό λοιπόν να αναλυθεί ο τρόπος με τον οποίο οι μη-έμπιστοι κόμβοι επηρεάζουν την διαδικασία έκδοσης των πιστοποιητικών.

Η μοντελοποίηση του προβλήματος βασίζεται στην θεωρία πιθανοτήτων. Συγκεκριμένα, κάθε κόμβος κατά την διάρκεια μιας περιόδου πιστοποίησης ανήκει είτε στο σύνολο των έμπιστων κόμβων είτε στο σύνολο των μη-έμπιστων. Όλοι οι κόμβοι κατά την διάρκεια της ζωής τους (διάρκεια λειτουργίας του συστήματος) μπορεί να αλλάζουν κατάσταση από το ένα σύνολο στο άλλο, αλλά για μια δεδομένη περίοδο πιστοποίησης ανήκουν σε ένα μόνο σύνολο. Έστω p η πιθανότητα να ανήκει κάποιος από τους κόμβους του συστήματος στο σύνολο των μη-έμπιστων και $q = 1 - p$ στο σύνολο των έμπιστων. Σε κάθε περίοδο ανανέωσης των κόμβων πιστοποίησης, επιλέγονται n τυχαίοι κόμβοι σε κάθε τομέα. Εφόσον οι κόμβοι μπορούν να επανεκλεγούν και στο μέλλον ως κόμβοι πιστοποίησης, η πιθανότητα ότι x από τους n συνολικά κόμβους πιστοποίησης ανήκουν στο σύνολο των μη-έμπιστων δίνεται από τον τύπο

$$Pr(X = x) = \binom{n}{x} p^x q^{n-x}$$

Αντίστοιχα, η αθροιστική συνάρτηση πιθανότητας ότι το πολύ $t - 1$ κόμβοι είναι μη-έμπιστοι είναι:

$$Pr(X < t) = \sum_{x=1}^{t-1} \binom{n}{x} p^x q^{n-x}$$

Χρησιμοποιώντας την παραπάνω εξίσωση υπολογίζεται η ανοχή του συστήματος σε μη-έμπιστους κόμβους. Για συγκεκριμένες τιμές των p, n, t η πιθανότητα ότι η διαδικασία της πιστοποίησης θα γίνει σωστά είναι $Pr(X < t) \geq 1 - \epsilon,\ \epsilon \simeq 0$. Στον πίνακα 4.1 παρουσιάζεται η πιθανότητα λιγότεροι από t κόμβοι να είναι κακόβουλοι αν κατά την διαδικασία πιστοποίησης η πιθανότητα να επιλεγεί κακόβουλος κόμβος είναι 0.3. Για παράδειγμα στο σχήμα κατωφλιού $(24, 49)$ βλέπουμε ότι η

Πίνακας 4.1: Πιθανότητα επιτυχίας πιστοποίησης για διάφορα σχήματα κατωφλιού και μεταβλητό πληθυσμό μη-έμπιστων κόμβων

Κατώφλι (t, n)	$p = 0$	$p = 0.1$	$p = 0.2$	$p = 0.3$
$(6, 13)$	1	0.999	0.969	0.834
$(12, 25)$	1	$1 - 1.49 \cdot 10^{-6}$	0.998	0.955
$(24, 49)$	1	$1 - 5.1 \cdot 10^{-12}$	$1 - 5.3 \cdot 10^{-6}$	0.955
$(48, 97)$	1	$1 - 8.3 \cdot 10^{-23}$	$1 - 8.5 \cdot 10^{-11}$	$1 - 4.5 \cdot 10^{-5}$
$(96, 193)$	1	$1 - 2.95 \cdot 10^{-44}$	$1 - 3.02 \cdot 10^{-20}$	$1 - 7.5 \cdot 10^{-9}$

πιθανότητα να επιλεγούν το πολύ 23 μη έμπιστοι κόμβοι, δοθέντος ότι στον τομέα το 20% των κόμβων είναι μη έμπιστοι είναι σχεδόν 1. Για την ακρίβεια είναι $Pr(X < t) \geq 1 - \epsilon$ όπου $\epsilon = 5.3 \times 10^{-6}$.

Στο πλαίσιο της ενότητας αυτής έγινε η υπόθεση ότι οι μη-έμπιστοι κόμβοι είναι κατανεμημένοι τυχαία. Αυτή η υπόθεση είναι αληθής καθώς μετά το πρωτόκολλο αρχικοποίησης οι κόμβοι σε κάθε περίοδο ανανέωσης επιλέγονται με τυχαίο τρόπο (βλ. 4.5.4.8).

4.7.1.1 Υπολογιστικό κόστος

Η απόδοση του Chord-PKI εξαρτάται σε μεγάλο βαθμό από την απόδοση του συστήματος κατωφλιού. Για τον λόγο αυτό υλοποιήθηκε ένα σύστημα κατωφλιού και μελετήθηκε η υπολογιστική επιβάρυνση που έχουν οι κρυπτογραφικές πράξεις που αφορούν την πιστοποίηση των κόμβων, με δυο βασικούς στόχους:

- Να αποδειχθεί αν οι συγκεκριμένες κρυπτογραφικές λειτουργίες μπορεί να εκτελεστούν από τυπικά υπολογιστικά συστήματα, τα οποία αποτελούν και την συντριπτική πλειοψηφία σε δίκτυα ομότιμων κόμβων.
- Να μελετηθεί η επίδραση που έχουν στο υπολογιστικό κόστος των κρυπτογραφικών λειτουργιών οι παράμετροι του συστήματος (αριθμός κατωφλιού, μέγεθος μηνύματος)

Η λειτουργία της πιστοποίησης των κόμβων υλοποιήθηκε σε προσομοιωτή. Ο προσομοιωτής υλοποιήθηκε σε γλώσσα προγραμματισμού $JAVA^{TM}$, ο οποίος χρησιμοποιεί δύο παραλλαγές συστημάτων κρυπτογραφίας κατωφλιού RSA: το βασικό σύστημα κατωφλιού και το αξιόπιστο που παρέχει την δυνατότητα να ελεγχθεί

Πίνακας 4.2: Υπολογιστικό κόστος μηχανισμού κατωφλιού: κόστος δημιουργίας κλειδιού, κόστος μερικών υπογραφών και κόστος επιβεβαίωσης υπογραφών

	Key Generation (sec)				**Partial Signature Generation (sec)**				**Signature Verification (sec)**			
Threshold set	msg 1024	msg 2048	msg 4096	msg 8192	msg 1024	msg 2048	msg 4096	msg 8192	msg 1024	msg 2048	msg 4096	msg 8192
(6, 13)	3.38	2.60	2.52	3.62	1.17	1.12	1.13	1.18	1.07	1.02	1.05	1.12
(12, 25)	3.66	3.65	3.45	2.99	2.30	2.30	2.30	2.35	2.17	2.06	2.05	2.13
(24, 49)	4.43	4.33	4.64	4.76	5.03	4.93	4.93	4.92	4.60	4.46	4.37	4.35
(48, 97)	6.98	7.29	7.10	7.68	11.92	12.33	12.67	12.14	9.72	10.17	9.98	10.02
(96, 193)	15.56	17.50	16.58	15.54	38.02	41.68	40.55	36.74	28.45	30.44	29.79	27.18

η ορθότητα κάθε μέρους της υπογραφής. Στο αξιόπιστο σύστημα οι κακόβουλοι κόμβοι μπορεί να ανιχνευτούν και να απομακρυνθούν από το σύνολο των κόμβων πιστοποίησης.

Τα πειράματα διεξήχθησαν σε ένα τυπικό υπολογιστικό σύστημα με επεξεργαστή τύπου Pentium IV με συχνότητα 2.8 GHz για τα παρακάτω σχήματα κατωφλιού:

- $(6, 13)$
- $(12, 25)$
- $(24, 49)$
- $(48, 97)$
- $(96, 193)$

Στον πίνακα 4.2 παρουσιάζονται οι χρόνοι εκτέλεσης των πειραμάτων για σύστημα βασικού κατωφλιού, όπου κάθε μέτρηση είναι ο μέσος όρος 1000 εκτελέσεων του πειράματος. Συγκεκριμένα, μετρήθηκαν οι παρακάτω χρόνοι:

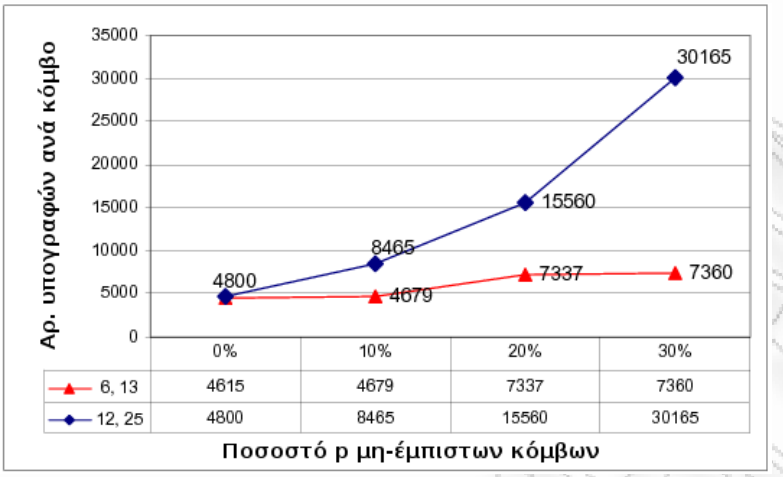
- Δημιουργίας των κλειδιών
- Δημιουργίας υπογραφής (πιστοποίησης) μηνυμάτων μεταβλητού μεγέθους $(1K, 2K, 4K, 8K)$
- Επαλήθευσης των μηνυμάτων.

Στην περίπτωση ενός αξιόπιστου σχήματος κατωφλιού το υπολογιστικό κόστος για την λειτουργία της υπογραφής και της επαλήθευσης είναι διπλάσια από αυτά στον πίνακα 4.2. Βέβαια όταν υπάρχουν στο σύστημα κακόβουλοι κόμβοι να υπονομεύουν την διαδικασία της πιστοποίησης παρέχοντας μη έγκυρες μερικές υπογραφές τότε το συνολικό κόστος (σε αριθμό μερικών υπογραφών) του αξιόπιστου σχήματος είναι μικρότερο από το βασικό σχήμα.

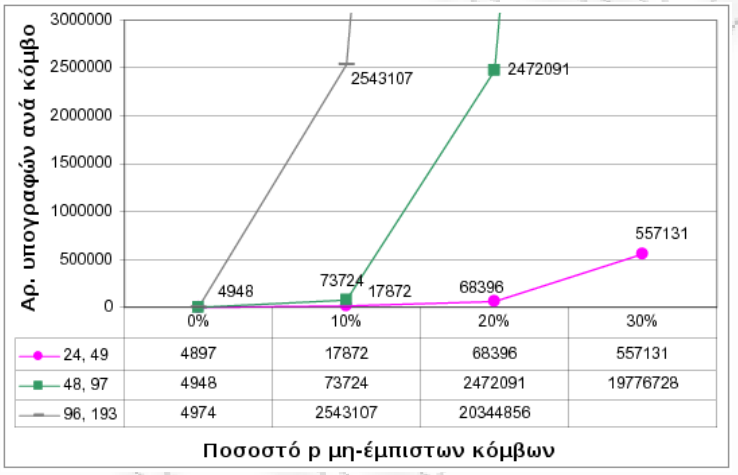
Για να μετρηθεί η επίδοση των δύο σχημάτων κατωφλιού εκτελέστηκε πείραμα κατά το οποίο πιστοποιήθηκαν 10000 κόμβοι. Το πρωτόκολλο πιστοποίησης (4.5.4.1) ορίζει ότι οι κόμβοι πιστοποίησης επιλέγονται τυχαία. Στα πειράματα η επιλογή των κόμβων πιστοποίησης ακολουθεί την ομοιόμορφη κατανομή. Για να πιστοποιηθεί κάποιος από τους 10000 κόμβους πρέπει να υπογραφεί το πιστοποιητικό του από t κόμβους πιστοποίησης, το συλλέκτη και άλλους $t-1$ κόμβους πιστοποίησης που επιλέγονται τυχαία. Για 10000 πιστοποιήσεις απαιτούνται $t \times 10000$ μερικές υπογραφές. Στα σχήματα 4.5 και 4.6 παρατηρούμε τον μέσο αριθμό πιστοποιήσεων ανά κόμβο όταν δεν υπάρουν κακόβουλοι κόμβοι πιστοποίησης ($p = 0\%$) για τα σχήματα κατωφλιού $(6, 13)...(96, 193)$.

Στην περίπτωση που υπάρχουν κακόβουλοι κόμβοι, στον πληθυσμό των κόμβων πιστοποίησης, ο βασικός μηχανισμός κατωφλιού λειτουργεί ως εξής: όταν σε κάποια από τις 10000 πιστοποιήσεις επιλεγεί είτε κακόβουλος συλλέκτης είτε κάποιος κακόβουλος κόμβος πιστοποίησης (από τους $t-1$) τότε η διαδικασία επαναλαμβάνεται από την αρχή μέχρι το σύνολο πιστοποίησης να μην περιέχει κακόβουλους κόμβους. Αυτό έχει ως αποτέλεσμα την μεγάλη αύξηση των μερικών υπογραφών (σε σχέση με τις υπογραφές όταν δεν υπάρχουν κακόβουλοι κόμβοι) καθώς αυξάνεται και το σχήμα κατωφλιού (από $(6, 13)$ ως $(96, 193)$) αλλά και καθώς αυξάνεται ο πληθυσμός των κακόβουλων κόμβων. Παρατηρούμε στο σχήμα 4.5, ιδιαίτερα για τα *μεγάλα* σχήματα κατωφλιού ($(24, 49)...(96, 193)$) ότι ακόμα και για μικρό ποσοστό κακόβουλων κόμβων ($p = 10\%$) ο μέρος αριθμός των πιστοποιήσεων αυξάνεται εκθετικά.

Αντίθετα με τον αξιόπιστο μηχανισμό κατωφλιού η διαδικασία της πιστοποίησης δεν επηρεάζεται σε τόσο μεγάλο βαθμό από την αύξηση των κακόβουλων κόμβων (σχήμα 4.6). Αν κάποιος από τους κόμβους πιστοποίησης είναι κακόβουλος και



(α΄) (6, 13) και (12, 25)



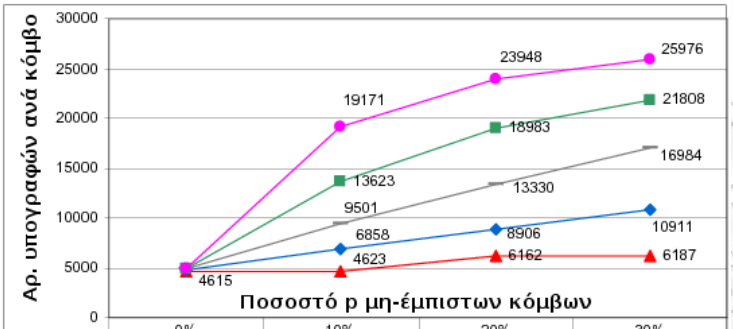
(β΄) (24, 49), (48, 97) και (96, 193)

Σχήμα 4.5: Βασικό σχήμα κατωφλιού: αριθμός μερικών υπογραφών ανά κόμβο πιστοποίησης καθώς αυξάνεται ο πληθυσμός των κακόβουλων κόμβων

δημιουργήσει *λανθασμένη* μερική υπογραφή εντοπίζεται από τον μηχανισμό κατωφλιού και στην θέση του επιλέγεται κάποιος άλλος κόμβος. Η διαδικασία αυτή επαναλαμβάνεται αν χρειαστεί μέχρι να βρεθεί κάποιος μη κακόβουλος κόμβος.

4.7.1.2 Δικτυακό κόστος

Στην συνέχεια μελετήθηκε το δικτυακό κόστος της διαδικασίας της πιστοποίησης των κόμβων και η επίδραση που έχει η αύξηση του συνόλου των κακόβουλων κόμβων πιστοποίησης. Συγκεκριμένα, στο πλαίσιο ενός τομέα του Chord-PKI εκτελέστηκε πείραμα κατά το οποίο μετρήθηκε ο αριθμών των μηνυμάτων που

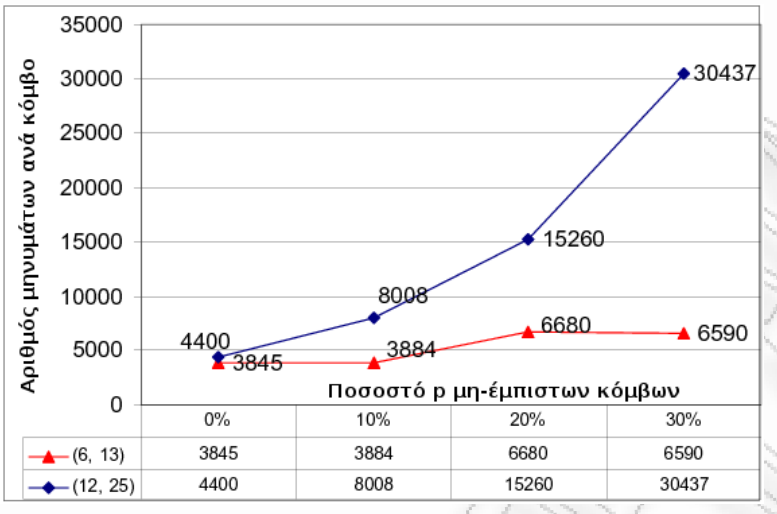


Σχήμα 4.6: Αξιόπιστο σχήμα κατωφλιού:αριθμός μερικών υπογραφών ανά κόμβο πιστοποίησης καθώς αυξάνεται ο πληθυσμός των κακόβουλων κόμβων

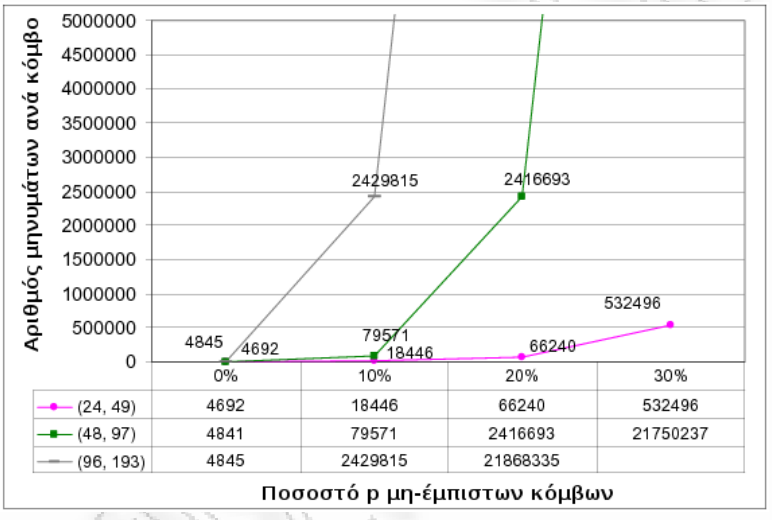
στάλθηκαν στο δίκτυο κατά τις πιστοποιήσεις 10000 κόμβων. Τα μηνύματα δημιουργούνται από τους κόμβους πιστοποίησης κατά την διαδικασία πιστοποίησης κάποιου κόμβου. Στο σχήμα 4.7 παρατηρούμε τον αριθμό μηνυμάτων για το βασικό μηχανισμό ενώ στο σχήμα 4.8 τον αριθμό μηνυμάτων για τον αξιόπιστο μηχανισμό κρυπτογράφησης κατωφλιού. Για το βασικό σχήμα παρατηρούμε ότι ακόμα και αν έχουμε μικρό ποσοστό κακόβουλων κόμβων (π.χ 10% του συνολικού πληθυσμού) ο αριθμός των μηνυμάτων είναι τεράστιος καθώς κάθε φορά που επιλέγεται έστω και ένας κακόβουλος κόμβος πιστοποίησης η διαδικασία της πιστοποίησης επαναλαμβάνεται από την αρχή. Αντίθετα, στην περίπτωση του αξιόπιστου μηχανισμού κατωφλιού δεν επαναλαμβάνεται η διαδικασία από την αρχή αλλά επιλέγεται ένας καινούργιος κόμβος πιστοποίησης από τον συλλέκτη με επιπλέον κόστος ενός μηνύματος.

4.8 Συμπεράσματα

Στο κεφάλαιο αυτό περιγράφηκαν τα σημαντικότερα ζητήματα ασφάλειας των δικτύων δομημένης επικάλυψης. Αναδείχτηκε το πρόβλημα της ταυτοποίησης των κόμβων και προτάθηκε μια κατανεμημένη υποδομή δημοσίου κλειδιού, μέσω της οποίας οι κόμβοι των δικτύων επικάλυψης αποκτούν πιστοποιητικά. Η υποδομή αυτή, το Chord-PKI, βασίζεται στο πρωτόκολλο δρομολόγησης του Chord και το επεκτείνει ορίζοντας τα πρωτόκολλα μέσω των οποίων πιστοποιούνται οι κόμβοι,



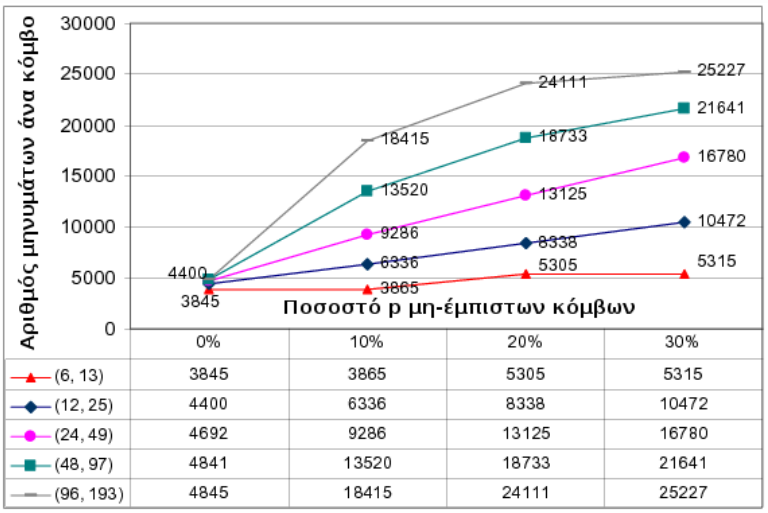
(α΄) (6, 13) και (12, 25)



(β΄) (24, 49), (48, 97) και (96, 193)

Σχήμα 4.7: Βασικό σχήμα κατωφλιού: αριθμός μηνυμάτων ανά κόμβο πιστοποίησης καθώς αυξάνεται ο πληθυσμός των κακόβουλων κόμβων

αποθηκεύονται τα πιστοποιητικά και οι λίστες ανάκλησής τους. Όλες οι λειτουργίες που προτείνονται μπορεί να πραγματοποιηθούν από οποιονδήποτε κόμβο του συστήματος διατηρώντας με αυτόν τον τρόπο τις ιδιότητες της επεκτασιμότητας και της απόδοσης που υπάρχουν σε συστήματα όπως το Chord. Εφόσον οι κόμβοι που συμμετέχουν σε συστήματα δικτύων δομημένης επικάλυψης δεν μπορούν να θεωρούνται έμπιστοι, η ασφάλεια του Chord-PKI αναλύθηκε υπό την παρουσία κακόβουλων κόμβων και ορίστηκε η ανοχή του συστήματος σε κακόβουλους κόμβους. Τέλος μέσω προσομοιώσεων παρουσιάστηκε η απόδοση του συστήματος που



Σχήμα 4.8: Αξιόπιστο σχήμα κατωφλιού: αριθμός μηνυμάτων ανά κόμβο πιστοποίησης καθώς αυξάνεται ο πληθυσμός των κακόβουλων κόμβων

αποδεικνύει ότι μια υποδομή όπως το Chord-PKI μπορεί να υλοποιηθεί από τυπικά υπολογιστικά συστήματα που αποτελούν την συντριπτική πλειοψηφία των κόμβων σε ομότιμα δίκτυα.

Κεφάλαιο 5

Κατανεμημένη ψηφιακή βιβλιοθήκη

Περίληψη

Στο κεφάλαιο αυτό περιγράφεται μια πρότυπη αρχιτεκτονική ψηφιακής βιβλιοθήκης προσανατολισμένη σε πανεπιστημιακά τεκμήρια. Η ψηφιακή βιβλιοθήκη χρησιμοποιεί τεχνολογίες υπολογιστικού πλέγματος με την βοήθεια των οποίων οι λειτουργίες της κατανέμονται στις διαθέσιμες υπολογιστικές υποδομές. Η κατανομή των λειτουργιών πραγματοποιείται σχεδιάζοντας την αρχιτεκτονική της ψηφιακής βιβλιοθήκης στα πρότυπα της ανοιχτής αρχιτεκτονικής OGSA [9].

5.1 Εισαγωγή

Στο κεφάλαιο αυτό περιγράφεται μια πρότυπη *ψηφιακή βιβλιοθήκη* [90] προσανατολισμένη σε δεδομένα του πανεπιστημιακού χώρου (LU-Grid). Η βιβλιοθήκη LU-Grid βασίζεται στα πρότυπα των *υπηρεσιών ιστού* (Web Services) [91] και στην αρχιτεκτονική OGSA [9] για συστήματα υπολογιστικού πλέγματος. Ο σχεδιασμός της προτεινόμενης βιβλιοθήκης περιλαμβάνει τον ορισμό των βασικών υπηρεσιών της ψηφιακής βιβλιοθήκης, τις αλληλεπιδράσεις μεταξύ τους καθώς και με άλλες υπηρεσίες πλέγματος. Οι υπηρεσίες της LU-Grid ορίζουν τις λειτουργίες επί του βασικού δομικού στοιχείου της βιβλιοθήκης του *έγγραφου*, για το οποίο αναλύονται τα συστατικά του και ορίζονται οι τρόποι περιγραφής του ώστε να είναι δυνατή η σημασιολογική του επεξεργασία.

Η σημασιολογική αξιοποίηση της πληροφορίας αποτελεί έναν από τους πρωτογενείς στόχους της βιβλιοθήκης και επιτυγχάνεται με την ενσωμάτωση κατάλληλων τεχνολογιών όπως της RDF [92]. Μέσω αυτών των τεχνολογιών τα δεδομένα και γενικότερα οι πόροι και τα τεκμήρια που φυλάσσονται στην ψηφιακή συλλογή, ορίζονται και περιγράφονται με κατάλληλο τρόπο ώστε να είναι εφικτή η σημασιολογική τους αξιοποίηση.

Οι λόγοι για τους οποίους επιλέχτηκε η χρήση των συστημάτων υπολογιστικού πλέγματος ως βάση για την δημιουργία της βιβλιοθήκης LU-Grid βασίζονται στις μεγάλες απαιτήσεις που υπάρχουν στον χώρο των ψηφιακών βιβλιοθηκών για την αποθήκευση και την επεξεργασία των ψηφιακών συλλογών. Οι ψηφιακές βιβλιοθήκες πανεπιστημιακών δεδομένων περιλαμβάνουν ένα ευρύ σύνολο δεδομένων που είναι αποθηκευμένο σε διαφορετικές (φυσικές) τοποθεσίες. Κάθε πανεπιστημιακό ίδρυμα διαχειρίζεται (δημιουργεί, ανανεώνει, διαγράφει) δεδομένα, που *φιλοξενούνται* σε εξυπηρετητές διαφορετικών τμημάτων. Επιπλέον, κάποιες υπηρεσίες εύρεσης της βιβλιοθήκης βασίζονται στην σημασιολογική επεξεργασία των δεδομένων τα οποία έχουν μεγάλες απαιτήσεις σε επεξεργασία και σε αποθηκευτικούς χώρους. Τα συστήματα υπολογιστικού πλέγματος παρέχουν τεχνολογίες μέσω των οποίων είναι δυνατή η χρησιμοποίηση υπολογιστικών υποδομών που ανήκουν σε διαφορετικές διαχειριστικές αρχές, για την εφαρμογή υπολογιστικά χρονοβόρων εργασιών.

Στα πρώτα στάδια της εξέλιξής τους τα συστήματα υπολογιστικού πλέγματος επικεντρώνονταν σε επιστημονικές εφαρμογές που αφορούσαν υπολογιστικά χρονοβόρες εργασίες που έπρεπε να εκτελεστούν σε υπολογιστικά συστήματα πολλαπλών οργανισμών οργανισμών-ιδρυμάτων (στο πλαίσιο των οποίων διεξαγόταν το αντίστοιχο πείραμα). Η τεχνογνωσία που αποκτήθηκε κατά την ανάπτυξη πολλών εφαρμογών οδήγησε στην δημιουργία (και στην εξέλιξη) της εργαλειοθήκης Globus [93] που αποτελεί το *de facto* εργαλείο για την ανάπτυξη εφαρμογών πλέγματος. Κατά την ωρίμανση της εργαλειοθήκης κατασκευάστηκαν προγραμματιστικές βιβλιοθήκες γενικού σκοπού που μπορούσαν να χρησιμοποιηθούν από οποιαδήποτε εφαρμογή επιθυμούσε να χρησιμοποιήσει τεχνολογίες πλέγματος. Σε μεγάλο βαθμό η δημιουργία αυτών των βιβλιοθηκών βασίζεται στην διαδικασία προτυποποίησης των τεχνολογιών πλέγματος που αποτυπώνεται στο πρότυπο *ανοιχτής αρχιτεκτονικής υπηρεσιών πλέγματος* (Open Grid Services Architecture – OGSA) [9]. Η βιβλιοθήκη LU-Grid είναι από τις πρώτες γενικού σκοπού εφαρμογές υπολογιστικού πλέγματος που βασιζόταν στην αρχιτεκτονική της OGSA.

Στις ενότητες που ακολουθούν γίνεται επισκόπηση των τεχνολογιών πλέγματος και ορίζονται τα βασικά στοιχεία της αρχιτεκτονικής OGSA που χρησιμοποιούνται στη βιβλιοθήκη LU-Grid. Στην συνέχεια παρουσιάζεται η αρχιτεκτονική της βιβλιοθήκης LU-Grid, και ορίζεται το έγγραφο και οι υπηρεσίες που παρέχονται επί του εγγράφου.

5.2 Συστήματα υπολογιστικού πλέγματος

Τα συστήματα υπολογιστικού πλέγματος – Grid Computing Systems είναι κατανεμημένα υπολογιστικά συστήματα. Ο όρος πλέγμα, στο πλαίσιο τον υπολογιστικών συστημάτων, εμφανίστηκε για πρώτη φορά στα μέσα της δεκαετίας του 1990 και αναφερόταν σε μια κατανεμημένη υπολογιστική υποδομή ικανή να επιλύσει πολύπλοκα, χρονοβόρα επιστημονικά πειράματα [7, 8]. Σύμφωνα με τους Foster και Kasselman [8] ο ρόλος των *συστημάτων πλέγματος* έχει εξελιχθεί σε ένα σύστημα στο οποίο διαμοιράζονται *υπολογιστικοί πόροι* μεταξύ διάφορων οντοτήτων προκειμένου να επιλυθεί κάποιο πρόβλημα. Ο όρος υπολογιστικός πόρος περιλαμβάνει οποιονδήποτε πόρο βρίσκεται σε κάποιον οργανισμό και ο οποίος μπορεί να συνεισφέρει στην επίλυση του προβλήματος. Παραδείγματα υπολογιστικών

πόρων περιλαμβάνουν μεμονωμένους επεξεργαστές, αποθηκευτικούς δίσκους αλλά και συστοιχίες υπολογιστών (clusters), εξειδικευμένα συστήματα αποθήκευσης αλλά και πιο εξωτικούς υπολογιστικούς πόρους όπως τηλεσκόπια. Οι πόροι αυτοί παρότι ανήκουν σε ξεχωριστούς οργανισμούς και βρίσκονται σε διαφορετικά γεωγραφικά σημεία μπορεί να χρησιμοποιηθούν όλοι μαζί συγχρονισμένα στο πλαίσιο ενός προκειμένου να επιλυθεί κάποιο πρόβλημα.

Ο όρος του ιδεατού οργανισμού αναφέρεται για πρώτη φορά στην εργασία του Foster, *'η ανατομία του πλέγματος'* [94] στην οποία αναλύεται το πρόβλημα της διαμοίρασης υπολογιστικών πόρων στο πλαίσιο ενός ιδεατού οργανισμού. Επίσης, περιγράφεται η αρχιτεκτονική των συστημάτων πλέγματος και αναφέρονται οι λειτουργίες που υποστηρίζονται από ένα τέτοιο σύστημα. Στην αρχιτεκτονική αυτή οποιοσδήποτε πόρος παρέχει τις λειτουργίες του (π.χ αποθήκευση δεδομένων αν ο πόρος είναι χώρος αποθήκευσης) μέσω μίας ή περισσοτέρων υπηρεσιών. Ένας από τους βασικότερους στόχους της αρχιτεκτονικής των συστημάτων πλέγματος είναι η προτυποποίηση των βασικών υπηρεσιών πλέγματος καθώς και του τρόπου που αλληλεπιδρούν μεταξύ τους. Η διαδικασία αυτή περιγράφεται στο πρότυπο της αρχιτεκτονικής OGSA [9], το οποίο το καθορίζει ο διεθνής οργανισμός Open Grid Forum [95] (παλιότερα γνωστός ως Global Grid Forum). Ο οργανισμός αυτός αποτελείται από ερευνητές, χρήστες και προγραμματιστές του υπολογιστικού πλέγματος καθώς επίσης και από περίπου 400 οργανισμούς που βρίσκονται σε περισσότερες από 50 χώρες. Ο βασικός στόχος του οργανισμού είναι η ταχύτατη εξέλιξη και υιοθέτηση των τεχνολογιών κατανεμημένου υπολογισμού μέσα από τα πρότυπα που δημιουργεί.

Στις επόμενες υποενότητες περιγράφουμε τις απαιτήσεις της αρχιτεκτονικής πλέγματος και παραθέτουμε τα επίπεδα στα οποία οι απαιτήσεις μετουσιώνονται σε λειτουργίες. Στην συνέχεια στην υποενότητα 5.2.3 ορίζεται η υπηρεσία πλέγματος και περιγράφονται οι μηχανισμοί με τους οποίους οι υπηρεσίες αυτές αλληλεπιδρούν στα πλαίσια της ανοιχτής αρχιτεκτονικής πλέγματος.

5.2.1 Αρχιτεκτονική υπολογιστικού πλέγματος

Βασικός στόχος της αρχιτεκτονικής του υπολογιστικού πλέγματος είναι ο ορισμός των βασικών δομικών στοιχείων-συστατικών μέσω των οποίων μπορούν να

διαμοιράζονται υπολογιστικοί πόροι που ανήκουν σε διαφορετικούς οργανισμούς. Συγκεκριμένα, για κάθε δομικό συστατικό ορίζεται ο σκοπός και η λειτουργία του. Επίσης καθορίζονται οι αλληλεπιδράσεις μεταξύ των συστατικών. Οι οργανισμοί, που διαχειρίζονται τους υπολογιστικούς πόρους, στο σύνολό τους απαρτίζουν τον ιδεατό οργανισμό στο πλαίσιο του οποίου ορίζονται οι σχετικοί κανόνες διαμοίρασης των υπολογιστικών πόρων.

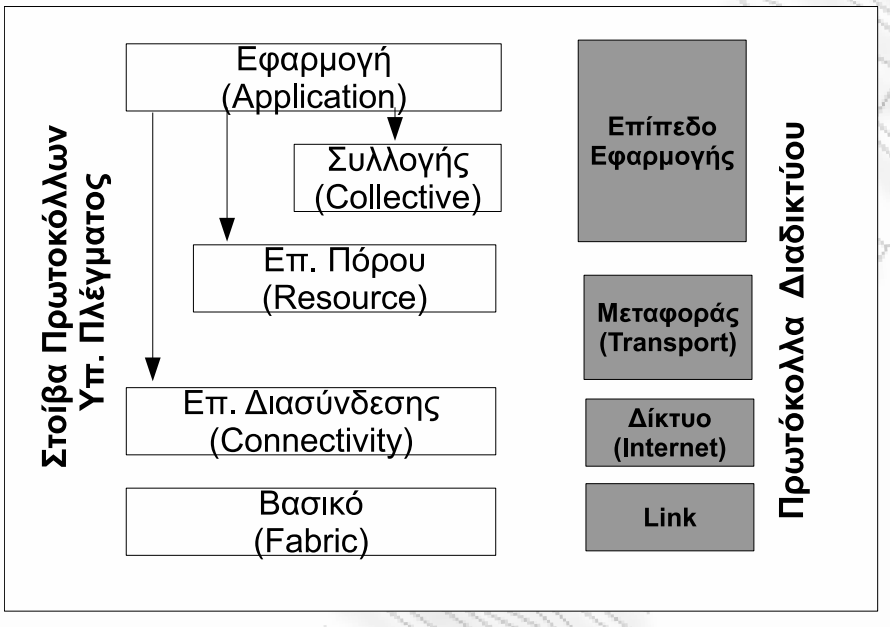
Ο ιδεατός οργανισμός παρουσιάζει μεγάλες διαφοροποιήσεις ως προς τον αριθμό και το είδος των συμμετεχόντων οντοτήτων, τις δραστηριότητές του, την διάρκειά του, την κλίμακα των οργανισμών οι οποίοι αλληλεπιδρούν και φυσικά ως προς τους πόρους που διαμοιράζονται. Το γεγονός αυτό παρουσιάζει αρκετές δυσκολίες στην μοντελοποίηση του συστήματος.

Επιπροσθέτως, στον ιδεατό οργανισμό υπάρχουν κάποιοι συμμετέχοντες οι οποίοι ενδεχομένως δεν έχουν καμία σχέση εμπιστοσύνης μεταξύ τους και οι οποίοι επιθυμούν να μοιραστούν υπολογιστικούς πόρους για να επιλύσουν κάποιο πρόβλημα. Για να γίνει ο διαμοιρασμός ο κάθε ιδιοκτήτης των πόρων πρέπει να ορίζει ποιοι πόροι είναι διαθέσιμοι, για ποιο χρονικό διάστημα και με ποιον τρόπο (κανόνες λειτουργίας).

Στην θεματική περιοχή των κατανεμημένων υπολογιστικών συστημάτων υπάρχουν και άλλες τεχνολογίες με τις οποίες γίνεται δυνατή η διαμοίραση υπολογιστικών πόρων. Η κυριότερη διαφοροποίηση της αρχιτεκτονικής υπολογιστικού πλέγματος είναι η κλίμακα στην οποία συμβαίνει η διαμοίραση. Για παράδειγμα μέσω τεχνολογιών όπως η CORBA και η Enterprise Java (J2EE) είναι δυνατή η διαμοίραση πόρων στα όρια ενός οργανισμού. Επίσης, οι παροχείς υπηρεσιών αποθήκευσης (Storage Service Provider–SSP) και γενικότερα οι παροχείς υπηρεσιών εφαρμογών (Application Service Provider) επιτρέπουν σε εξωτερικές οντότητες-οργανισμούς να χρησιμοποιούν, με περιορισμένο τρόπο, αποθηκευτικούς χώρους και γενικότερα υπηρεσίες που βρίσκονται εκτός του οργανισμού μέσω του ιδιωτικού δικτύου του παροχέα (Virtual Private Network).

Η αρχιτεκτονική του πλέγματος έρχεται να καλύψει όλες τις απαιτήσεις για διαμοίραση πόρων μεταξύ διαφορετικών οργανισμών μέσω δημόσιων δικτύων όπως το διαδίκτυο. Την τελευταία δεκαετία έρευνα και ανάπτυξη στην κοινότητα του υπολογιστικού πλέγματος δημιούργησε πρωτόκολλα, υπηρεσίες και εργαλεία μέσω

των οποίων γίνεται πραγματικότητα η ασφαλής διαμοίραση των πόρων. Σύμφωνα με τους Foster και Kasselman [9] οι λειτουργίες και τα πρωτόκολλα των συστημάτων πλέγματος μπορούν να οργανωθούν σύμφωνα με την αρχιτεκτονική που παρουσιάζεται στο σχήμα 5.1.



Σχήμα 5.1: Η στοίβα των πρωτοκόλλων του Υπολογιστικού Πλέγματος και ο συσχετισμός τους με τα πρωτόκολλα του Διαδικτύου

Στο ίδιο σχήμα παρατίθεται και η στοίβα των πρωτοκόλλων του διαδικτύου ώστε γίνεται κατανοητό σε ποιο επίπεδο του διαδικτύου ανήκουν τα επίπεδα του πλέγματος. Αυτό είναι απαραίτητο καθώς τα πρωτόκολλα ή τα εργαλεία που αναπτύσσονται για κάθε επίπεδο του πλέγματος βασίζονται και χρησιμοποιούν τα πρωτόκολλα του διαδικτύου. Η στρωματοποίηση των πρωτοκόλλων του σχήματος 5.1 ως ένα βαθμό έχει διαμορφωθεί από τις εργαλειοθήκες, τα προγραμματιστικά εργαλεία και γενικότερα το λογισμικό που δημιουργείται διεθνώς για τα υπολογιστικά πλέγματα. (Globus Toolkit [10], National Technology Grid [96], EGEE Grid [97]). Στις επόμενες υποενότητες παρουσιάζονται συνοπτικά κάθε ένα από τα επίπεδα του πλέγματος.

5.2.1.1 Βασικό επίπεδο (Fabric)

Το επίπεδο αυτό περιλαμβάνει τους πόρους μέσω των οποίων είναι δυνατή η επικοινωνία των πρωτοκόλλων του υπολογιστικού πλέγματος. Τα συστατικά στο

βασικό επίπεδο υλοποιούν τις πρωτογενείς λειτουργίες που υποστηρίζονται από τον πόρο. Για παράδειγμα για υπολογιστικούς πόρους χρειάζονται μηχανισμοί μέσω των οποίων θα ενεργοποιούνται προγράμματα αλλά και θα παρακολουθείται η εξέλιξη τους. Οι πόροι αποθήκευσης χρειάζονται μηχανισμούς μέσω των οποίων θα αποθηκεύονται και θα ανακτώνται τα δεδομένα από αυτούς.

5.2.1.2 Επίπεδο συνδεσιμότητας

Στο *επίπεδο της συνδεσιμότητας* ορίζονται τα βασικά πρωτόκολλα επικοινωνίας και αυθεντικοποίησης μέσω των οποίων διεξάγεται οποιαδήποτε ενέργεια στο υπολογιστικό πλέγμα. Συγκεκριμένα, τα πρωτόκολλα επικοινωνίας ασχολούνται με την μεταφορά δεδομένων από και προς τους πόρους του βασικού επιπέδου. Τα πρωτόκολλα αυθεντικοποίησης παρέχουν στις υπηρεσίες επικοινωνίας κρυπτογραφικές τεχνικές μέσω των οποίων επιτυγχάνεται η επαλήθευση της ταυτότητας οποιασδήποτε οντότητας του συστήματος. Θέματα που αφορούν την ασφάλεια στο επίπεδο της συνδεσιμότητας βασίζονται σε υπάρχουσες λύσεις ασφαλείας όπου αυτό είναι εφικτό.

5.2.1.3 Επίπεδο διαμοίρασης ατομικών πόρων

Στο συγκεκριμένο επίπεδο ορίζονται τα πρωτόκολλα και οι υπηρεσίες μέσω των οποίων ενεργοποιούνται, παρακολουθούνται, ελέγχονται και ενδεχομένως κοστολογούνται οι λειτουργίες ανά πόρο του υπολογιστικού πλέγματος. Τα πρωτόκολλα αυτά βασίζονται στις υπηρεσίες του επιπέδου συνδεσιμότητας για την συνεργασία τους και χρησιμοποιούν τις υπηρεσίες του βασικού επιπέδου για να αποκτήσουν πρόσβαση και για να ελέγξουν κάποιον πόρο. Τα πρωτόκολλα σε αυτό το επίπεδο χωρίζονται σε πρωτόκολλα πληροφοριών, μέσω των οποίων γνωστοποιούνται οι πληροφορίες για τους πόρους και σε πρωτόκολλα διαχείρισης μέσω των οποίων ορίζονται και πραγματοποιούνται οι ενέργειες που αφορούν την χρήση ενός διαμοιραζόμενου πόρου.

5.2.1.4 Επίπεδο συντονισμού (Collective)

Ενώ το προηγούμενο επίπεδο επικεντρώνεται στις λειτουργίες ενός μόνο πόρου, το επίπεδο συντονισμού ασχολείται με την οργάνωση και το συντονισμό πολλών πόρων που αλληλεπιδρούν στο πλαίσιο μιας υπηρεσίας.

5.2.1.5 Επίπεδο εφαρμογής

Το επίπεδο εφαρμογής περιλαμβάνει τις εφαρμογές των χρηστών του υπολογιστικού πλέγματος. Οι κατασκευαστές (προγραμματιστές) των εφαρμογών χρησιμοποιούν τις υπηρεσίες, μέσω προγραμματιστικών βιβλιοθηκών, των τριών υφιστάμενων επιπέδων (βλ. 5.4). Σε αυτό το επίπεδο δεν βρίσκονται μόνο εφαρμογές που χρησιμοποιούν τους πόρους του πλέγματος αλλά και πολύπλοκα συστήματα, βιβλιοθήκες που μπορεί να χρησιμοποιηθούν από άλλες εφαρμογές. Η προτεινόμενη ψηφιακή βιβλιοθήκη ανήκει σε αυτήν την κατηγορία.

5.2.2 Υπηρεσίες ιστού

Για κάθε ένα από τα παραπάνω επίπεδα ορίζονται οι υπηρεσίες μέσω της *ανοιχτής αρχιτεκτονικής υπηρεσιών πλέγματος* (OGSA). Η αρχιτεκτονική OGSA βασίζεται στις *υπηρεσίες ιστού* [91] τις οποίες επεκτείνει προκειμένου να κατασκευαστούν οι *υπηρεσίες πλέγματος*. Για αυτόν τον λόγο παραθέτουμε στην ενότητα αυτή τα βασικότερα συστατικά των υπηρεσιών ιστού.

Οι υπηρεσίες ιστού (Web Services) αποτελούν μια από τις σημαντικότερες κατανεμημένες τεχνολογίες που έχουν αναπτυχθεί τα τελευταία χρόνια. Η κύρια διαφορά της από κατανεμημένες τεχνολογίες όπως η CORBA, η DCE και το RMI της Java είναι ότι βασίζεται σε διαδεδομένα πρότυπα του διαδικτύου όπως η XML [98] ώστε να παρέχει δυνατότητες κατανεμημένης επεξεργασίας μεταξύ ετερογενών περιβάλλοντων. Όπως σε όλες τις κατανεμημένες τεχνολογίες έτσι και στις υπηρεσίες ιστού ο βασικός στόχος είναι η απομακρυσμένη χρησιμοποίηση ενός λογισμικού μέσω του οποίου πραγματοποιείται μια ή περισσότερες λειτουργίες. Στην τεχνολογία των υπηρεσιών ιστού ορίζεται ο τρόπος περιγραφής του λογισμικού, ορίζονται μέθοδοι

για την χρησιμοποίησή του καθώς και ευρεστικές μέθοδοι μέσω των οποίων ανακαλύπτονται οι υπηρεσίες. Τα πρότυπα των υπηρεσιών ιστού αναπτύσσονται στο πλαίσιο του οργανισμού W3C καθώς και με την συνεργασία σημαντικών εταιρειών όπως η Microsoft, IBM, Sun, οι οποίες δημιουργούν πολλά από τα προγραμματιστικά εργαλεία.

Οι υπηρεσίες ιστού περιγράφονται από πολλά πρότυπα αλλά θα περιοριστούμε εδώ στην περιγραφή των προτύπων SOAP, WSDL, WS-Inspection καθώς οι υπηρεσίες πλέγματος βασίζονται σε αυτά.

- Το πρωτόκολλο Simple Object Access Protocol (SOAP) [99] καθορίζει πώς ανταλλάσσονται τα μηνύματα μεταξύ της οντότητας που έχει δημιουργήσει μια υπηρεσία και της οντότητας που θέλει να την χρησιμοποιήσει. Στο πρωτόκολλο SOAP ορίζεται ένας μηχανισμός ενθυλάκωσης δεδομένων XML, μέσω των οποίων περιγράφονται οι απομακρυσμένες κλήσεις που γίνονται στο πλαίσιο της επικοινωνίας μεταξύ του παροχέα και του χρήστη της υπηρεσίας. Το πρωτόκολλο αυτό δεν ορίζει και δεν βασίζεται σε κάποιο συγκεκριμένο πρωτόκολλο μεταφοράς δεδομένων αλλά μπορεί να χρησιμοποιεί οποιοδήποτε πρωτόκολλο του διαδικτύου (π.χ. HTTP, TCP, FTP, JMS)
- Η γλώσσα περιγραφής των υπηρεσιών ιστού (Web Services Description Language–WSDL) [100] μέσω της οποίας ορίζονται οι λειτουργίες (endpoints) των υπηρεσιών ιστού.
- Το πρωτόκολλο WS-Inspection ορίζει μια γλώσσα XML μέσω της οποίας οι χρήστες των υπηρεσιών μπορούν να εντοπίζουν τις υπηρεσίες που τους ενδιαφέρουν. Το έγγραφο που είναι γραμμένο σε αυτήν την γλώσσα (WSIL) περιέχει είτε περιγραφές υπηρεσιών ιστού είτε δείκτες σε πηγές που περιέχουν υπηρεσίες ιστού. Συνήθως οι περιγραφές των υπηρεσιών είναι δείκτες URL [46] στο αντίστοιχο WSDL έγγραφο της υπηρεσίας.

5.2.3 Ανοιχτή αρχιτεκτονική υπηρεσιών πλέγματος

Η ανοιχτή αρχιτεκτονική υπηρεσιών πλέγματος [9] (Open Grid Services Architecture–OGSA) είναι η απόρροια της συγχώνευσης όλων των τεχνολογιών και πρωτοκόλλων που είχαν αναπτυχθεί από διαφορετικούς φορείς ώστε να είναι δυνατή

η αξιοποίηση τους κάτω από μια κοινή πλατφόρμα που αναπτύσσεται από το διεθνή οργανισμό Open Grid Forum [95]. Στην αρχιτεκτονική της OGSA κυρίαρχο ρόλο παίζει η *υπηρεσία* πλέγματος μέσω της οποίας οι υπολογιστικές υποδομές παρέχουν τις λειτουργίες τους στους χρήστες του πλέγματος. Η αρχιτεκτονική OGSA ορίζει την υπηρεσία πλέγματος, τους πρότυπους τρόπους με τους οποίους δημιουργούνται οι υπηρεσίες και τις διεπαφές μέσω των οποίων αλληλεπιδρούν.

Στις επόμενες ενότητες θα αναλύσουμε την υπηρεσία πλέγματος περιγράφοντας τις τεχνολογίες στις οποίες βασίζεται και στην συνέχεια θα αναλύσουμε τους βασικούς μηχανισμούς της αρχιτεκτονικής OGSA που χρησιμοποιούνται στο σχεδιασμό της βιβλιοθήκης LU-Grid. Περισσότερες λεπτομέρειες με την αρχιτεκτονική OGSA υπάρχουν στα έγγραφα που διατηρεί ο οργανισμός Open Grid Forum [95].

5.2.3.1 Υπηρεσία υπολογιστικού πλέγματος

Όλοι οι διαμοιραζόμενοι πόροι στο πλέγμα (επεξεργαστές, δίσκοι, βάσεις δεδομένων, δίκτυα κ.λ.π) αντιστοιχούν σε υπηρεσίες μέσω των οποίων παρέχεται η λειτουργία των πόρων στους χρήστες του πλέγματος. Σε ένα κατανεμημένο περιβάλλον με τα χαρακτηριστικά του πλέγματος σημαντικό ρόλο παίζουν οι μηχανισμοί μέσω των οποίων εξασφαλίζεται η διαλειτουργικότητα. Στην αρχιτεκτονική της OGSA το πρόβλημα της διαλειτουργικότητας αντιμετωπίζεται με δύο τρόπους:

- Ορίζοντας πρότυπες διεπαφές για τις υπηρεσίες πλέγματος.
- Καθορίζοντας ποιο πρωτόκολλο μπορεί να καλέσει ποια διεπαφή.

Ορίζοντας πρότυπες διεπαφές καθορίζεται ο τρόπος αλληλεπίδρασης των υπηρεσιών. Η αρχιτεκτονική OGSA βασίζεται σε μεγάλο βαθμό στις υπηρεσίες ιστού (Web Services) τις οποίες επεκτείνει, παρέχοντας διεπαφές που ακολουθούν συγκεκριμένους κανόνες. Μέσω των προτεινόμενων επεκτάσεων στις υπηρεσίες ιστού προσφέρονται οι λειτουργίες ανακάλυψης, δημιουργίας και διαχείρισης των υπηρεσιών πλέγματος.

Όπως αναφέρθηκε στην ενότητα 5.2.2 μέσω του εγγράφου WSDL ορίζονται οι λειτουργίες (portTypes) μιας υπηρεσίας ιστού. Η αρχιτεκτονική OGSA έχει επεκτείνει την γλώσσα WSDL ώστε να υποστηρίζονται οι βασικές λειτουργίες των υπηρεσιών

πλέγματος. Συγκεκριμένα οι λειτουργίες αυτές επιτρέπουν την ανακάλυψη, την δημιουργία, την διαχείριση και την ειδοποίηση των υπηρεσιών.

Στην συνέχεια περιγράφεται το μοντέλο των υπηρεσιών πλέγματος στο οποίο ορίζονται οι διεπαφές και οι λειτουργίες που πρέπει να έχουν οι υπηρεσίες και επισημαίνονται οι μηχανισμοί μέσω των οποίων υλοποιείται το μοντέλο των υπηρεσιών οι οποίες είναι απαραίτητες για κατανόηση της λειτουργίας της αρχιτεκτονικής OGSA αλλά και της αρχιτεκτονικής της κατανεμημένης βιβλιοθήκης LU-Grid που αναπτύσσεται στην ενότητα 5.3.

5.2.3.2 Διεπαφές και λειτουργίες υπηρεσίας πλέγματος

Οι υπηρεσίες πλέγματος διατηρούν *πληροφορίες εσωτερικής κατάστασης* καθόλη την διάρκεια της ζωής τους. Η πληροφορία κατάστασης ουσιαστικά διαχωρίζει διαφορετικά στιγμιότυπα (instances) της ίδιας υπηρεσίας. Επίσης, οι υπηρεσίες αλληλεπιδρούν μεταξύ τους ανταλλάσσοντας μηνύματα. Στα κατανεμημένα συστήματα δεν μπορεί να υπάρξουν εγγυήσεις ότι ένα μήνυμα που έχει αποσταλεί έχει φτάσει στον προορισμό του. Διατηρώντας πληροφορίες σχετικά με την εσωτερική κατάσταση μπορεί να επιβεβαιωθεί ότι μια υπηρεσία έχει λάβει κάποιο μήνυμα. Τέλος οι υπηρεσίες της αρχιτεκτονικής OGSA δημιουργούνται και τερματίζονται δυναμικά μέσω κατάλληλων διεπαφών. Επειδή οι υπηρεσίες δημιουργούνται δυναμικά και έχουν εσωτερική κατάσταση πρέπει να υπάρχει ένας τρόπος με τον οποίο θα ξεχωρίζεται ένα στιγμιότυπο υπηρεσίας από ένα άλλο. Για αυτόν τον λόγο σε κάθε στιγμιότυπο υπηρεσίας αντιστοιχεί ένα μοναδικό όνομα, το GSH (Grid Service Handle).

Οι υπηρεσίες πλέγματος κατά την διάρκεια της ζωής τους μπορεί να ανανεωθούν (π.χ. να δημιουργηθεί καινούρια έκδοση της υπηρεσίας) ή να υποστηρίζουν καινούρια λειτουργικότητα. Για αυτόν τον λόγο το GSH δεν πρέπει να περιέχει πληροφορίες που έχουν σχέση με το στιγμιότυπο της υπηρεσίας όπως με το πρωτόκολλο επικοινωνίας (π.χ. τη διεύθυνση δικτύου). Οι σχετικές πληροφορίες με το στιγμιότυπο της κάθε υπηρεσίας περιέχεται στην οντότητα Grid Service Reference (GSR). Σε αντίθεση με το GSH το GSR μπορεί να αλλάξει κατά την διάρκεια ζωής μιας υπηρεσίας.

Στο πλαίσιο της αρχιτεκτονικής OGSA ορίζονται οι διεπαφές που πρέπει να υποστηρίζονται από την *υπηρεσία πλέγματος* ώστε να είναι δυνατή η δημιουργία τους (μέσω των GSH και GSR), η διαχείριση του χρόνου ζωής τους, η ανακάλυψή τους και επίσης μηχανισμοί ειδοποίησης υπηρεσιών.

Για την δημιουργία των υπηρεσιών η αρχιτεκτονική OGSA ορίζει την διεπαφή Factory. Οποιαδήποτε υπηρεσία πλέγματος υλοποιεί αυτήν την διεπαφή παρέχει την λειτουργία CreateService, μέσω της οποίας δημιουργούνται το GSH και το αρχικό GSR για την υπηρεσία. Ενώ το GSH θα αντιστοιχεί στο στιγμιότυπο της υπηρεσίας που δημιουργείται μέσω της διεπαφής Factory, το GSR μπορεί να αλλάξει κατά την διάρκεια ζωής του στιγμιοτύπου. Για αυτόν τον λόγο η αρχιτεκτονική OGSA παρέχει την διεπαφή HandleMap μέσω της οποίας ορίζονται οι αντιστοιχίες GSH με GSR. Η διεπαφή παρέχει λειτουργία μέσω της οποίας δοθέντος ενός GSH επιστρέφεται ένα έγκυρο GSR.

Για την διαχείριση των υπηρεσιών και του χρόνου ζωής τους η αρχιτεκτονική OGSA ορίζει ότι οι υπηρεσίες πλέγματος δημιουργούνται με προκαθορισμένο, αρχικό χρόνο ζωής. Αυτός ο χρόνος μπορεί να επεκταθεί για συγκεκριμένο χρονικό διάστημα μέσω κάποιας αίτησης (η αίτηση αυτή μπορεί να μην γίνει αποδεκτή). Αν η περίοδος ζωής του στιγμιοτύπου της υπηρεσίας λήξει, η πλατφόρμα εκτέλεσης της υπηρεσίας θα την τερματίσει και θα απελευθερώσει τους πόρους που χρησιμοποιούσε. Τόσο ο αρχικός χρόνος ζωής όσο και η επέκταση του χρόνου ζωής της υπηρεσίας καθορίζονται μέσω της λειτουργίας SetTerminationTime της διεπαφής GridService.

Κάθε στιγμιότυπο υπηρεσίας περιέχει πληροφορίες εσωτερικής κατάστασης που περιγράφουν τα χαρακτηριστικά του συγκεκριμένου στιγμιοτύπου (π.χ. ημερομηνία έκδοσης υπηρεσίας, περιγραφή υπηρεσίας κ.λ.π.). Τα δεδομένα των υπηρεσιών ορίζονται σε ένα έγγραφο XML το οποίο αποτελείται από ένα ή περισσότερα *στοιχεία δεδομένων* (Service Data Elements). Το έγγραφο αυτό παρέχεται στους ενδιαφερόμενους μέσω της λειτουργίας FindServiceData της διεπαφής GridService. Στην αρχιτεκτονική OGSA η υπηρεσία πλέγματος που παρέχει ανακάλυψης άλλων υπηρεσιών πλέγματος ονομάζεται *μητρώο* (Registry). Τέτοιου είδους υπηρεσίες ορίζονται από:

Πίνακας 5.1: Υποστηριζόμενες διεπαφές της υπηρεσίας πλέγματος σύμφωνα με τις προδιαγραφές OGSA

Διεπαφή($portType$)	Λειτουργία(ες)
$GridService$	$FindServiceData$
	$SetTerminationTime$
	$Destroy$
$NotificationSource$	$SubscribeTo$
$NotificationSink$	$DeliverNotification$
$Registry$	$RegistrerService$
	$UnregisterService$
$Factory$	$CreateService$
$HandleMap$	$FindByHandle$

- Την διεπαφή Registry που παρέχει τις λειτουργίες ώστε τα GSH των υπηρεσιών να εγγράφονται στο μητρώο.
- Τα στοιχεία δεδομένων των υπηρεσιών που αντιστοιχούν στα εγγεγραμμένα GSH.

Δηλαδή με την διεπαφή Registry εγγράφεται το GSH στο μητρώο και με την λειτουργία FindServiceData της διεπαφής GridService ανακτάται η πληροφορία για κάποιο GSH.

Τέλος, η αρχιτεκτονική OGSA ορίζει μηχανισμούς ειδοποίησης μέσω των οποίων οι ενδιαφερόμενες οντότητες (π.χ. υπηρεσίες) δηλώνουν ότι θέλουν να λαμβάνουν μηνύματα που αφορούν κάποιο γεγονός (event). Αν κάποια υπηρεσία θέλει να στέλνει ειδοποιήσεις σε άλλες υπηρεσίες πρέπει να υλοποιεί την διεπαφή NotificationSource. Αν μία υπηρεσία επιθυμεί να λαμβάνει μηνύματα πρέπει να υλοποιεί τη διεπαφή NotificationSink.

Ο πίνακας 5.1 παρουσιάζει συνοπτικά τις βασικές διεπαφές της αρχιτεκτονικής OGSA και τις (βασικές) λειτουργίες τους.

5.3 Πρότυπη ψηφιακή βιβλιοθήκη πανεπιστημιακών δεδομένων

Η πληροφορία που δημιουργείται και διανέμεται από την πανεπιστημιακή κοινότητα, τόσο σε διδακτικό όσο και σε ερευνητικό αλλά και διαχειριστικό επίπεδο παρουσιάζει τεράστια ποικιλία. Το γεγονός αυτό έχει παρακινήσει πολλούς ερευνητές, οι οποίοι εξέτασαν την εφαρμογή των τεχνολογιών εξειδικευμένων ψηφιακών βιβλιοθηκών για την πανεπιστημιακή κοινότητα.

Σε αυτή την ενότητα παρουσιάζεται η αρχιτεκτονική της ψηφιακής βιβλιοθήκης LU-Grid η οποία ορίζει τις βασικές υπηρεσίες της βιβλιοθήκης. Οι υπηρεσίες υλοποιούνται με βάση τους μηχανισμούς υπηρεσιών πλέγματος που περιγράφηκαν στην προηγούμενη ενότητα. Με αυτό τον τρόπο διασφαλίζεται ότι η βιβλιοθήκη LU-Grid εκμεταλλεύεται τις υποδομές και τους πόρους του υπολογιστικού πλέγματος.

Η προτεινόμενη αρχιτεκτονική εστιάζει στην αποδοτική διαχείριση των δεδομένων που παράγονται και διακινούνται στο πανεπιστήμιο και τα οποία είναι οργανωμένα σε έγγραφα. Η ιδιαίτερη φύση της πληροφορίας των πανεπιστημιακών δεδομένων, η οποία αφορά μεγάλες ποσότητες δεδομένων από πολλαπλές επιστημονικές περιοχές, καθιστά απαραίτητη τη σημασιολογική μελέτη της παραπάνω πληροφορίας ώστε οι λειτουργίες της ψηφιακής βιβλιοθήκης (π.χ. Αναζήτηση) να λαμβάνουν υπόψιν την σημασιολογία των δεδομένων. Για τον λόγο αυτό έχουν υιοθετηθεί διεθνή πρότυπα όπως το RDF [92] και OWL [101], με τα οποία είναι δυνατή η σημασιολογική αναπαράσταση των δεδομένων.

5.4 Αρχιτεκτονική LU-Grid

Στο πλαίσιο της αρχιτεκτονικής της βιβλιοθήκης ορίζεται η δομή του εγγράφου μέσω των οποίου αναπαρίσταται η πληροφορία στον πανεπιστημιακό χώρο και στην συνέχεια περιγράφονται οι παρεχόμενες λειτουργίες επί του εγγράφου. Οι λειτουργίες αυτές ακολουθούν τα πρότυπα των υπηρεσιών πλέγματος οπότε ορίζονται οι αλληλεπιδράσεις τους στο πλαίσιο της αρχιτεκτονικής OGSA.

5.4.1 Έγγραφο

Μια από τις βασικότερες απαιτήσεις σε περιβάλλοντα με ευρύ γνωστικό αντικείμενο, όπως το πανεπιστημιακό, είναι η ομαδοποίηση πόρων που έχουν την ίδια *σημασία*. Επιπλέον, η σημασιολογία των πόρων πρέπει να περιγράφεται με εύκολο τρόπο ώστε να είναι δυνατή η συνάθροιση, η μοντελοποίηση και η επαναχρησιμοποίηση της γνώσης [102].

Στις ψηφιακές βιβλιοθήκες, τα έγγραφα αποτελούνται από πολλαπλούς πόρους (π.χ κείμενο, εικόνα, ήχος). Οι συγγενείς σημασιολογικά πόροι περιλαμβάνουν ψηφιακά αρχεία σε διάφορες μορφές (π.χ αρχεία pdf, doc, ppt, jpeg) αλλά και ένα ειδικό αρχείο που περιέχει πληροφορίες σχετικές με τη δομή του εγγράφου. Οι πληροφορίες των *μετα-δεδομένων* παρέχονται από τους συγγραφείς των εγγράφων μέσω υπηρεσιών της βιβλιοθήκης LU-Grid που θα αναλυθούν στις επόμενες ενότητες.

Το σύνολο των πόρων μαζί με το ειδικό αρχείο μεταδεδομένων αποτελούν ένα *έγγραφο* στο πλαίσιο της βιβλιοθήκης LU-Grid. Το έγγραφο έχει δομή που βασίζεται στο πρότυπο RDF για την περιγραφή των πόρων που περιέχονται στο έγγραφο και αποτελείται από μετα-δεδομένα που ακολουθούν το πρότυπο Dublin Core [103].

Επιπλέον, σε κάθε έγγραφο της βιβλιοθήκης αντιστοιχεί ένα αναγνωριστικό DOI- Document Object Identifier [104], το οποίο προσδιορίζει το έγγραφο όχι μόνο στα πλαίσια της συλλογής εγγράφων του LU-Grid, αλλά σε ολόκληρο τον παγκόσμιο ιστό. Με την χρησιμοποίηση τεχνολογιών όπως το αναγνωριστικό DOI, η προτεινόμενη αρχιτεκτονική μπορεί να είναι συμβατή με παρόμοιες ψηφιακές συλλογές.

5.4.2 Υπηρεσίες

Η προτεινόμενη αρχιτεκτονική επεκτείνει τις βασικές υπηρεσίες του υπολογιστικού πλέγματος, όπως αυτές περιγράφονται πρότυπο OGSI- Open Grid Services Infrastructure [105], με εξειδικευμένες υπηρεσίες ψηφιακών βιβλιοθηκών, μέσω των οποίων ανακαλύπτονται, ανακτώνται και διαχειρίζονται τα δεδομένα που υπάρχουν στην βιβλιοθήκη και είναι οργανωμένα ως έγγραφα της βιβλιοθήκης LU-Grid.

Συγκεκριμένα οι υπηρεσίες αυτές είναι:

- Υπηρεσία δημοσίευσης εγγράφων
- Υπηρεσία εύρεσης εγγράφων
- Υπηρεσία ανάκτησης εγγράφων

Ο χρήστης των υπηρεσιών αποκτά πρόσβαση στις παραπάνω υπηρεσίες μέσω της διεπαφής Factory όπως περιγράφεται στην αρχιτεκτονική της OGSA [106]. Κατά την αρχικοποίηση της υπηρεσίας εκτελείται η λειτουργία 'create' στο Factory που έχει ώς αποτέλεσμα την δημιουργία ενός στιγμιοτύπου υπηρεσίας. Ταυτόχρονα, δίνεται στον χρήστη της υπηρεσίας ένα αντικείμενο μέσω του οποίου θα έχει πρόσβαση στην υπηρεσία (GSH) και το στιγμιότυπο της υπηρεσίας εγγράφεται στο μητρώο (Registry) της βιβλιοθήκης LU-Grid.

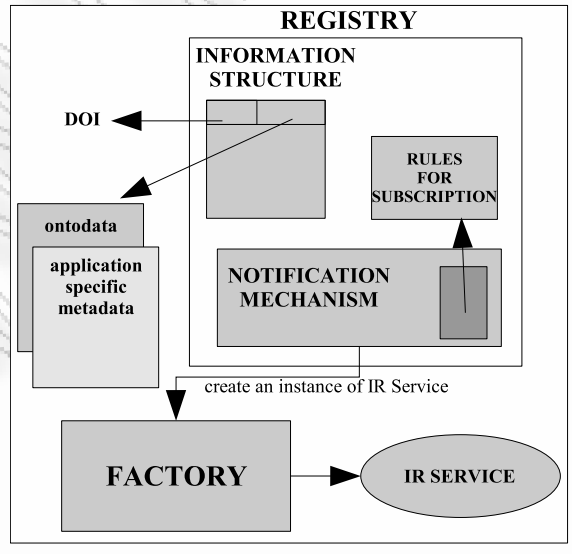
5.4.2.1 Μητρώο βιβλιοθήκης

Βασικό συστατικό στην υπηρεσιοστραφή αρχιτεκτονική της βιβλιοθήκης LU-Grid αποτελεί το *μητρώο* στο οποίο αποθηκεύονται πληροφορίες σχετικά με τα έγγραφα της βιβλιοθήκης αλλά και πληροφορίες σχετικές με τις παρεχόμενες υπηρεσίες. Ουσιαστικά το μητρώο αποτελεί την βάση δεδομένων της βιβλιοθήκης LU-Grid και αποτελείται από δύο λογικές οντότητες, μια δομή δεδομένων στην οποία αποθηκεύονται οι πληροφορίες και ένας μηχανισμός ενημέρωσης με την βοήθεια του οποίου οι χρήστες και οι υπηρεσίες ενημερώνονται για συγκεκριμένα γεγονότα. Οι προδιαγραφές του μηχανισμού περιγράφονται ορίζονται στο πρότυπο OGSI [105]. Το σχήμα 5.2 συνοψίζει την αρχιτεκτονική του μητρώου. Η *δομή πληροφοριών* αποθηκεύει δυο ειδών μετα-δεδομένα:

- Μετα-δεδομένα υπηρεσιών.
- Μετα-δεδομένα εγγράφου.

Τα μετα-δεδομένα υπηρεσιών παρέχουν πληροφορίες σχετικές με τους τρόπους ανάκτησης των δεδομένων και τα υποστηριζόμενα πρωτόκολλα μέσω των οποίων μεταδίδονται τα δεδομένα (π.χ έγγραφα). Επιπλέον, μέσω των μετα-δεδομένων των υπηρεσιών παρέχονται πληροφορίες σχετικά με την τοποθεσία των κατανεμημένων πόρων που αποτελούν το έγγραφο. Οι πληροφορίες αυτές κωδικοποιούνται

με βάση το πρότυπο του URI [37]. Από την άλλη μεριά, πληροφορίες σχετικά με την δομή και την σημασιολογία των εγγράφων παρέχονται από τα μετα-δεδομένα του εγγράφου. Κατάλληλες υπηρεσίες μπορούν να επεξεργαστούν την πληροφορία των μετα-δεδομένων του εγγράφου παρέχοντας σημασιολογικού χαρακτήρα υπηρεσίες στην βιβλιοθήκη. Τα δυο είδη μετα-δεδομένων αποθηκεύονται σε ένα πίνακα κατακερματισμού (Hash table) με το κλειδί της κάθε εγγραφής στον πίνακα να είναι ένα αναγνωριστικό τύπου doi, το οποίο όπως θα περιγραφεί στην συνέχεια δημιουργείται από την υπηρεσία δημοσίευσης εγγράφου. Ουσιαστικά, ο πίνακας κατακερματισμού είναι η *δομή πληροφορίας* του LU-Grid, στην οποία αποθηκεύονται τα μεταδεδομένα τα οποία είναι κωδικοποιημένα σύμφωνα με το πρότυπο RDF. Οι υπηρεσίες του LU-Grid χρησιμοποιούν τις πρωτογενείς πληροφορίες που βρίσκονται στον πίνακα κατακερματισμού ώστε να παρέχουν σύνθετες υπηρεσίες. Τονίζεται ότι ο πίνακας κατακερματισμού αποτελεί έναν λογικό τρόπο οργάνωσης της πληροφορίας παρέχοντας πρότυπες διεπαφές για την ανάκτηση και αναζήτηση της πληροφορίας (get(doi), put(doi, metadata)). Επίσης, ο πίνακας κατακερματισμού μπορεί να μην είναι αποθηκευμένος σε ένα υπολογιστικό σύστημα αλλά να είναι κατανεμημένος σε πολλαπλούς υπολογιστικούς πόρους. Στην βιβλιογραφία υπάρχουν αρκετά πρωτόκολλα κατανεμημένων πινάκων κατακερματισμού [11, 28, 26] μέσω τον οποίων επιτυγχάνεται πλήρης κατανομή των περιεχομένων του πίνακα στους διαθέσιμους υπολογιστικούς πόρους με το κόστος της αναζήτησης να είναι λογαριθμικό σε σχέση με τον αριθμό των πόρων στους οποίους κατανέμεται ο πίνακας.



Σχήμα 5.2: Μητρώο βιβλιοθήκης LU-grid

Ο *μηχανισμός ειδοποίησης* επιτρέπει στις οντότητες του συστήματος (συγκεκριμένα στις υπηρεσίες του συστήματος που δρουν εκ μέρους των οντοτήτων) να ενημερώνονται για γεγονότα που τους ενδιαφέρουν καθώς έχουν την δυνατότητα να εγγράφονται στο μηχανισμό ειδοποίησης για κάθε *γεγονός* (event) που τους αφορά.

5.4.2.2 Υπηρεσία δημοσίευσης εγγράφου

Στην υπηρεσία δημοσίευσης εγγράφου ορίζονται οι ενέργειες βάσει των οποίων οι συγγραφείς των εγγράφων μπορούν να εισάγουν πληροφορίες στην βιβλιοθήκη LU-Grid. Συγκεκριμένα οι συγγραφείς μπορούν να εισάγουν-ανανεώνουν-διαγράφουν τους επί μέρους πόρους που αποτελούν το *έγγραφο*. Επίσης, τις ίδιες ενέργειες μπορεί να τις εφαρμόσουν επί των μετα-δεδομένων του εγγράφου.

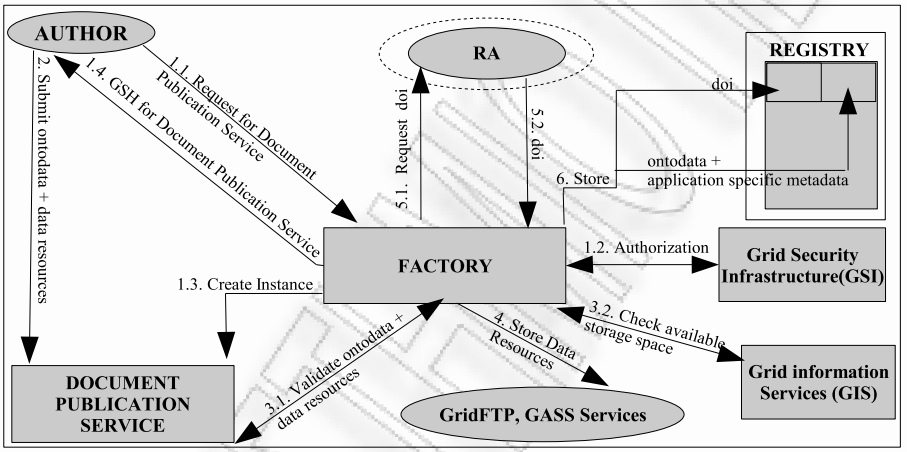
Όταν κάποιος συγγραφέας (ή κάποια υπηρεσία εκ μέρους του συγγραφέα) αποστείλει τους πόρους του εγγράφου (κείμενο, εικόνες, κ.λ.π) και τα μετα-δεδομένα του η πλατφόρμα του υπολογιστικού πλέγματος αναλαμβάνει δράση και διαπεραιώνει όλες τις απαραίτητες ενέργειες που παρουσιάζονται στο σχήμα 5.3.

- Αρχικά ο συγγραφέας αποκτά ένα στιγμιότυπο της υπηρεσίας δημοσίευσης μέσω του Factory. Αν ο συγγραφέας έχει τα απαραίτητα πιστοποιητικά για να χρησιμοποιήσει την υποδομή του υπολογιστικού πλέγματος, το Factory ελέγχει τα πιστοποιητικά που έδωσε ο συγγραφέας και σε περίπτωση που δικαιούται πρόσβαση στην υποδομή επιστρέφεται στον συγγραφέα το στιγμιότυπο της υπηρεσίας μέσω του GSH.
- Ο συγγραφέας δίνει τους πόρους του εγγράφου και τα μεταδεδομένα στην υπηρεσία
- Το Factory ελέγχει την ορθότητα της πληροφορίας που έδωσε ο συγγραφέας και στην συνέχεια επικοινωνεί με την υπηρεσία πληροφοριών του πλέγματος για να ελέγξει για τον διαθέσιμο ελεύθερο χώρο που χρειάζεται για την αποθήκευση της πληροφορίας.

- Στην συνέχεια οι υπηρεσίες του πλέγματος που είναι υπεύθυνες για την μεταφορά των δεδομένων (π.χ. GridFTP [107]) αναλαμβάνουν δράση και αποθηκεύουν το έγγραφο.

- Το Factory επικοινωνεί με την *αρχή διαμοίρασης* (Registration Agency–RA [104]) μέσω της οποίας αποδίδονται τα προθέματα doi. Με βάση το πρόθεμα αυτό αντιστοιχίζεται ένα doi αναγνωριστικό στο έγγραφο

- Τελικά μέσω του Factory, αποθηκεύεται στο μητρώο της βιβλιοθήκης μια εγγραφή με κλειδί το αναγνωριστικό doi του προηγούμενου βήματος που περιέχει τα μετα-δεδομένα του εγγράφου και της υπηρεσίας.



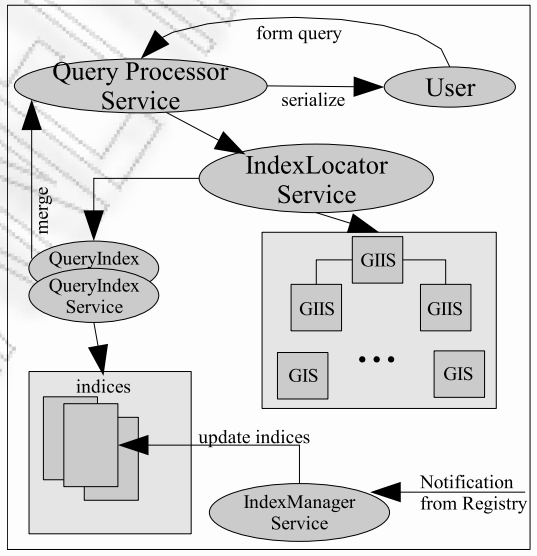
Σχήμα 5.3: Υπηρεσία δημοσίευσης εγγράφου

5.4.3 Υπηρεσία εύρεσης

Η υπηρεσία εύρεσης επιτρέπει στους χρήστες να βρίσκουν έγγραφα της βιβλιοθήκης LU-Grid βάσει κριτηρίων αναζήτησης. Σε αυτήν την ενότητα περιγράφουμε την υπηρεσία εύρεσης καθώς και τις αλληλεπιδράσεις της με τις υπηρεσίες της πλατφόρμας του υπολογιστικού πλέγματος. Στο σχήμα 5.4 παρουσιάζονται οι αλληλεπιδράσεις των συστατικών της υπηρεσίας καθώς ένας χρήστης θέτει το ερώτημά του στην υπηρεσία εύρεσης. Συγκεκριμένα, η υπηρεσία QueryProcessor επεξεργάζεται τα κριτήρια της αναζήτησης που έθεσε ο χρήστης και δημιουργείται ένα ερώτημα το οποίο μεταδίδεται στους κόμβους του υπολογιστικού πλέγματος οι οποίοι περιέχουν ευρετήρια.

Η μετάδοση γίνεται με την βοήθεια της υπηρεσίας IndexLocator, η οποία συνεργάζεται και χρησιμοποιεί τις υπηρεσίες πληροφόρησης του υπολογιστικού πλέγματος. Οι υπηρεσίες πληροφόρησης του πλέγματος χρησιμοποιούν μια ιεραρχική υποδομή από εξυπηρετητές καθένας από τους οποίους περιέχει πληροφορίες σχετικά με την τοποθεσία των πόρων του συστήματος αλλά και δείκτες προς άλλους εξυπηρετητές ώστε να είναι δυνατή η εύρεση οποιουδήποτε πόρου στο σύστημα. Ο μηχανισμός αυτός, ο οποίος ονομάζεται Grid Information Index Servers – GIIS [108], οργανώνει ιεραρχικά τους εξυπηρετητές GIS που περιέχουν τις τοποθεσίες των πόρων. Στο πλαίσιο της υπηρεσίας εύρεσης της βιβλιοθήκης LU-Grid υποθέτουμε ότι οι εξυπηρετητές GIS παρέχουν πληροφορίες σχετικά με την τοποθεσία των ευρετηρίων αλλά και με είδος των ευρετηρίων (π.χ. το ευρετήριο Α περιέχει συγγραφείς).

Τα ευρετήρια που διαχειρίζονται οι κόμβοι του υπολογιστικού πλέγματος κατασκευάζονται και ανανεώνονται από την υπηρεσία IndexManager. Η υπηρεσία αυτή συλλέγει τα μετα-δεδομένα του εγγράφου και δημιουργεί ή ανανεώνει κάποια από τα ευρετήρια που βρίσκονται κατανεμημένα στους κόμβους του συστήματος. Ο μηχανισμός ειδοποίησης που περιγράφηκε στην ενότητα 5.4.2.1 ειδοποιεί την υπηρεσία IndexManager για οποιαδήποτε αλλαγή στα μετα-δεδομένα ώστε να ανανεωθούν τα ευρετήρια. Επίσης, η υπηρεσία QueryIndex παρέχει στην υπηρεσία εύρεσης όλες τις απαραίτητες λειτουργίες εύρεσης επί των κατανεμημένων ευρετηρίων.



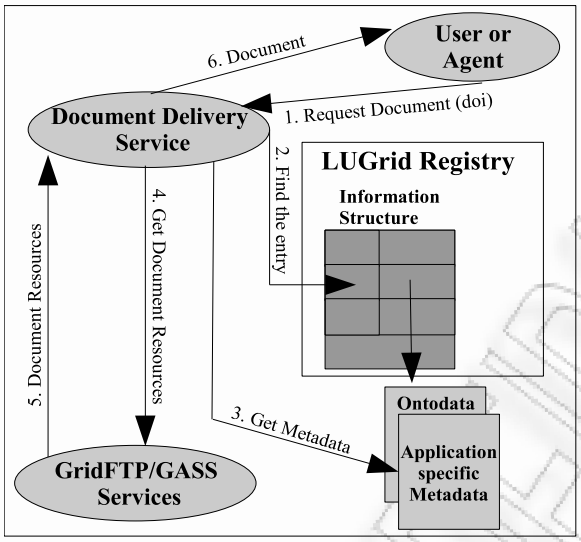
Σχήμα 5.4: Υπηρεσία εύρεσης εγγράφων

Στο σχήμα 5.4 παρουσιάζεται η αλληλεπίδραση ανάμεσα στα συστατικά της υπηρεσίας εύρεσης καθώς εξυπηρετείται το ερώτημα ενός χρήστη. Αρχικά ο IndexLocator βρίσκει τα ευρετήρια που ικανοποιούν το ερώτημα του χρήστη. Κάθε κόμβος του πλέγματος που περιέχει ευρετήρια επιστρέφει το υποσύνολο των εγγράφων των μετα-δεδομένων που διαχειρίζεται και ικανοποιούν το παραπάνω ερώτημα. Το υποσύνολο των μεταδεδομένων αποτελείται από το αναγνωριστικό doi καθώς και από τον τίτλο του εγγράφου, την λίστα με τους συγγραφής, ημερομηνία δημιουργίας κ.λ.π. Οι πληροφορίες αυτές αποστέλνονται στην υπηρεσία QueryProcessor η οποία τις ενσωματώνει σε μια δομή τύπου XML. Η δομή αυτή αποτελεί την *λίστα αποτελεσμάτων* του ερωτήματος και χρησιμοποιείται από την *υπηρεσία ανάκτησης* που θα περιγραφεί στην επόμενη ενότητα ώστε να επιλεχθούν ένα ή περισσότερα έγγραφα από την βιβλιοθήκη.

5.4.4 Υπηρεσία ανάκτησης εγγράφου

Μέσω της υπηρεσίας εύρεσης η οποία αναλύθηκε στην προηγούμενη ενότητα ο χρήστης της βιβλιοθήκης βρίσκει ένα ή περισσότερα έγγραφα. Η υπηρεσία ανάκτησης παραδίδει τα έγγραφα στα υπολογιστικά συστήματα των χρηστών. Το σχήμα 5.5 παρουσιάζει τις αλληλεπιδράσεις της υπηρεσίας ανάκτησης με τα υπόλοιπα συστατικά της βιβλιοθήκης.

- Ο χρήστης παρέχει το αναγνωριστικό doi του εγγράφου που επιθυμεί να ανακτήσει. (Οι χρήστες είναι σε θέση να γνωρίζουν τα αναγνωριστικά doi των εγγράφων μέσω της υπηρεσίας εύρεσης που αναλύθηκε στην προηγούμενη ενότητα.)
- Η υπηρεσία ανάκτησης επικοινωνεί με την δομή πληροφοριών του μητρώου και βρίσκει τα μεταδεδομένα (υπηρεσιών και εγγράφου) που αντιστοιχούν στο ζητούμενο έγγραφο.
- Τελικά με βάση τις πληροφορίες των μετα-δεδομένων συγκεντρώνονται τα δεδομένα του εγγράφου και μέσω των υπηρεσιών μεταφοράς δεδομένων (GridFTP, GASS) που παρέχονται από την πλατφόρμα υπολογιστικού πλέγματος επιστρέφεται το έγγραφο στον χρήστη.



Σχήμα 5.5: Υπηρεσία ανάκτησης εγγράφου

5.5 Συμπεράσματα

Στο κεφάλαιο αυτό εξερευνήθηκε πώς ένα σύστημα ψηφιακών βιβλιοθηκών μπορεί να *κατανεμηθεί* σε υπολογιστικά συστήματα πλέγματος. Ο τρόπος με τον οποίο οργανώνεται η πληροφορία στον ακαδημαϊκό χώρο, τόσο σε λογικό όσο και φυσικό επίπεδο ευνοεί την κατανεμημένη βιβλιοθήκη αφού η πληροφορία δημιουργείται και *καταναλώνεται* από πολλά τμήματα και βρίσκεται αποθηκευμένη στις υποδομές του κάθε ακαδημαϊκού τμήματος. Οι τεχνολογίες του υπολογιστικού πλέγματος αποτελούν σημαντικό αρωγό στην δημιουργία της κατανεμημένης βιβλιοθήκης καθώς παρέχουν μηχανισμούς μέσω των οποίων διάσπαρτοι υπολογιστικοί πόροι (π.χ. πληροφορίες, μηχανήματα που υπάρχουν σε κάθε τμήμα) μπορούν να συνδυαστούν ώστε να παρέχουν υπηρεσίες. Συγκεκριμένα, με την βοήθεια της ανοιχτής αρχιτεκτονικής συστημάτων πλέγματος [9] ορίζονται εφαρμογές οι λειτουργίες των οποίων κατανέμονται στις υποδομές του υπολογιστικού πλέγματος. Ακολουθώντας τα πρότυπα της ανοιχτής αρχιτεκτονικής υπηρεσιών πλέγματος περιγράψαμε την βιβλιοθήκη πανεπιστημιακών δεδομένων LU-Grid. Στα πλαίσια της βιβλιοθήκης ορίστηκε το *έγγραφο*, μέσω του οποίου συγγενείς σημασιολογικά πόροι οργανώνονται λογικά σε μια οντότητα και περιγράφηκαν οι βασικές υπηρεσίες της βιβλιοθήκης ως λειτουργίες επί του εγγράφου. Οι υπηρεσίες αυτές αλληλεπιδρούν μεταξύ τους, αλλά και με άλλες υπάρχουσες υπηρεσίες πλέγματος, μέσω διεπαφών (interfaces) που ορίζει η αρχιτεκτονική OGSA.

Μέσω της ανοιχτής αρχιτεκτονικής υπηρεσιών πλέγματος αξιοποιούνται υπολογιστικές υποδομές που ανήκουν σε διαφορετικές διαχειριστικές αρχές (π.χ. ακαδημαϊκά τμήματα) για την εκτέλεση υπολογιστικά χρονοβόρων εργασιών που αφορούν την σημασιολογική επεξεργασία των δεδομένων της βιβλιοθήκης LU-Grid.

Κεφάλαιο 6

Συμπεράσματα και μελλοντική έρευνα

Περίληψη

Στο κεφάλαιο αυτό αναφέρονται τα συμπεράσματα τις διατριβής και προτάσεις για μελλοντική έρευνα. Οι τεχνικές που παρουσιάστηκαν στην διατριβή βελτιώνουν τις λειτουργίες εύρεσης αλλά και την ασφάλεια των δικτύων δομημένης επικάλυψης. Με βάση τις συνεισφορές αυτές, αυξάνεται το πεδίο εφαρμογής των δικτύων δομημένης επικάλυψης και σε συστήματα που έχουν μεγαλύτερες απαιτήσεις τόσο στην εύρεση πληροφοριών όσο και στην ασφάλεια. Για την καλύτερη προστασία των δικτύων δομημένης επικάλυψης προτείνεται ο συνδυασμός του κατανεμημένου μηχανισμού πιστοποίησης με μηχανισμούς φήμης που έχουν αναπτυχθεί στο πλαίσιο των δικτύων ομότιμων κόμβων.

6.1 Συμπεράσματα

Η μελέτη της διατριβής εντάσσεται στην ερευνητική περιοχή των δικτύων ομότιμων κόμβων. Βασικός στόχος είναι η δημιουργία μηχανισμών αναζήτησης πληροφοριών σε συστήματα που λειτουργούν σε δικτυακά περιβάλλοντα μεγάλης κλίμακας. Τα δίκτυα δομημένης επικάλυψης μπορούν να αποτελέσουν το βασικό δομικό στοιχείο πάνω στο οποίο μπορεί να οικοδομηθεί ένας μηχανισμός εύρεσης πληροφοριών, καθώς είναι αποκεντρωμένα συστήματα με σχεδόν απεριόριστες δυνατότητες κλιμάκωσης. Επιπλέον, η εγγυημένη αναζήτηση που προσφέρει ο βασικός μηχανισμός εύρεσης που παρέχουν αποτελεί έναν ακόμα λόγο για την επιλογή τους ως βασική πλατφόρμα για εφαρμογές που λειτουργούν στο διαδίκτυο και έχουν πολλούς χρήστες. Βέβαια, τα δίκτυα επικάλυψης παρουσιάζουν και κάποιες αδυναμίες της οποίες έρχεται να καλύψει η μελέτη που γίνεται στην διατριβή. Συγκεκριμένα, οι δυνατότητες αναζήτησης που προσφέρουν είναι περιορισμένες και, επιπλέον, λόγω των συνθηκών του περιβάλλοντος στο οποίο σχεδιάστηκαν και λειτουργούν τίθενται σημαντικά ζητήματα ασφάλειας που υπονομεύουν την ορθή λειτουργία οποιασδήποτε εφαρμογής βασίζεται στα δίκτυα δομημένης επικάλυψης. Για αυτόν τον λόγο η διατριβή κινείται σε δυο κύριους άξονες. Αρχικά, εντοπίζονται οι ελλείψεις στις λειτουργίες εύρεσης των δικτύων δομημένης επικάλυψης και προτείνονται μέθοδοι μέσω των οποίων καλύπτονται οι ελλείψεις και στην συνέχεια αντιμετωπίζονται βασικά προβλήματα ασφάλειας των δικτύων επικάλυψης.

Συγκεκριμένα, στο κεφάλαιο 1 οριοθετήθηκε η θεματική περιοχή των συστημάτων μεγάλης κλίμακας παρουσιάζοντας τις απαιτήσεις και τις ιδιαίτερες συνθήκες λειτουργίας τους, παραθέτοντας παραδείγματα εφαρμογών [4, 5, 6] που λειτουργούν σε τέτοια περιβάλλοντα με τεράστια απήχηση στο ευρύ κοινό χρηστών του διαδικτύου. Παραθέτοντας παράλληλα και τις απαιτήσεις καινούργιων τεχνολογιών, όπως τα συστήματα υπολογιστικού πλέγματος [8], και προγραμματιστικών μοντέλων (π.χ. υπηρεσίες ιστού [91]) μέσω των οποίων δημιουργούνται εφαρμογές στη κλίμακα του διαδικτύου, δείχνουμε την *δυναμική* του διαδικτύου που οδηγεί στην συγκέντρωση τεράστιου όγκου πληροφοριών. Απαιτούνται λοιπόν μηχανισμοί εύρεσης ώστε να ανακαλύπτεται ο τεράστιος όγκος της πληροφορίας που είναι κατανεμημένη σε διάσπαρτα, γεωγραφικά και διαχειριστικά, συστήματα. Η

επιτυχία των εφαρμογών διαμοίρασης αρχείων [6] δείχνει ότι το κατανεμημένο υπολογιστικό σύστημα, μέσω του οποίου διαμοιράζονται υπολογιστικοί πόροι, έχει τεράστια απήχηση στους χρήστες του διαδικτύου. Καταδεικνύεται λοιπόν, ότι ένα κατανεμημένο σύστημα που οργανώνεται με βάση τα πρωτόκολλα των δικτύων ομότιμων κόμβων αποτελεί λύση για την εύρεση πληροφοριών σε μεγάλη κλίμακα, όπως είναι η κλίμακα του διαδικτύου.

Παραθέτοντας στο κεφάλαιο 2 την εξέλιξη των δικτύων ομότιμων κόμβων, και αναλύοντας τον τρόπο λειτουργίας τους, αναλύουμε τις εγγυήσεις σε επεκτασιμότητα, δυναμική συμπεριφορά και συμμετρία στην συμπεριφορά των κόμβων που παρέχονται από τα τελευταίας γενιάς δίκτυα ομότιμων κόμβων, τα δομημένα δίκτυα επικάλυψης. Επίσης, η παρουσίαση των βασικότερων εκπροσώπων δικτύων δομημένης επικάλυψης επιτρέπει την κατανόηση του τρόπου λειτουργίας τους αλλά ταυτόχρονα φανερώνονται οι αδυναμίες και οι περιορισμοί που θέτουν τα δίκτυα αυτά σε μηχανισμούς εύρεσης, που θέλουν να υποστηρίζουν πολύπλοκα ερωτήματα με βάση πολλαπλά κλειδιά αναζήτησης, αλλά και ερωτήματα σε εύρος τιμών. Εφόσον η οργάνωση της πληροφορίας, η οποία αποθηκεύεται σε αυτά τα συστήματα, βασίζεται σε συγκεκριμένους κανόνες ώστε να διασφαλίζεται η αποδοτική λειτουργία τους, περιορίζονται οι λειτουργίες εύρεσης που παρέχονται.

Στο κεφάλαιο 3 αντιμετωπίστηκε το συγκεκριμένο πρόβλημα, με κύριο γνώμονα την διαφύλαξη της επεκτασιμότητας του συστήματος και την συμμετρία στην επικοινωνία ανάμεσα στους κόμβους που το απαρτίζουν. Το δυαδικό δένδρο αντιγράφων ΔΔΑ αποτελεί μια γενική λύση η οποία μπορεί να εφαρμοστεί για την πλειοψηφία των δικτύων δομημένης επικάλυψης, μέσω των βασικών λειτουργιών `get` και `put` που παρέχουν. Στο δυαδικό δένδρο αντιγράφων ΔΔΑ, μέσω της ιεραρχικής οργάνωσης αντιγράφων της πληροφορίας υποστηρίζονται ερωτήματα εύρους τιμών για ορισμένο εύρος τιμών ενός κριτηρίου αναζήτησης της πληροφορίας. Επιπλέον, ο πλεονασμός της πληροφορίας μέσω του οποίου υποστηρίζονται τα ερωτήματα εύρους συμβάλλει και στην διαθεσιμότητα των δεδομένων. Με αυτόν τον τρόπο εξασφαλίζεται η ευρωστία του συστήματος υπό το καθεστώς συχνών σφαλμάτων στην λειτουργία των κόμβων, τα οποία αποτελούν τον κανόνα και όχι την εξαίρεση στην περίπτωση των δικτύων ομότιμων κόμβων.

Παρέχοντας λύσεις στο πρόβλημα της εύρεσης ασχοληθήκαμε με την αξιοπιστία των δικτύων δομημένης επικάλυψης, η οποία αποτελεί σημαντικό παράγοντα για την ορθή λειτουργία όχι μόνο των μεθόδων εύρεσης αλλά και γενικότερα οποιασδήποτε εφαρμογής βασίζεται σε δίκτυα δομημένης επικάλυψης. Για αυτόν τον λόγο αναλύονται στην ενότητα 4.2 οι ιδιαίτερες συνθήκες που επηρεάζουν την ασφάλεια των δικτύων δομημένης επικάλυψης και στην ενότητα 4.2.1 κατηγοριοποιούνται οι επιθέσεις και στην συνέχεια παρουσιάζονται λύσεις μέσω των οποίων αντιμετωπίζονται ορισμένα από τα προβλήματα. Μία από τις βασικές αδυναμίες στα δίκτυα δομημένης επικάλυψης, την οποία εκμεταλλεύονται πολλές επιθέσεις, είναι η αδυναμία των κόμβων για την *ταυτοποίηση* των απομακρυσμένων κόμβων. Λαμβάνοντας υπόψιν το μεγάλο αριθμό των κόμβων που συμμετέχουν στο δίκτυο επικάλυψης, τις απαιτήσεις επεκτασιμότητας, οι οποίες επιβάλλουν σε κάθε κόμβο να κρατά ορισμένο (προκαθορισμένο) πλήθος από κόμβους στους πίνακες δρομολόγησης του, η εμπιστοσύνη σε κάποιον *άγνωστο* κόμβο μπορεί να οδηγήσει στην *υπονόμευση* του συστήματος. Το πρόβλημα είναι σοβαρό αν λάβουμε υπόψιν ότι η βασικότερη λειτουργία των δικτύων δομημένης επικάλυψης, η δρομολόγηση προς κόμβους και δεδομένα υλοποιείται συνεργατικά μεταξύ μη-έμπιστων κόμβων. Βέβαια παρά το γεγονός ότι σε κατανεμημένα συστήματα δεν μπορεί να δοθεί ολοκληρωμένη λύση ώστε να επαληθεύεται η ταυτότητα απομακρυσμένων οντοτήτων [65], πιστεύουμε ότι μέσω τεχνικών κρυπτογραφίας όπως οι ψηφιακές υπογραφές τίθενται οι βάσεις για την ταυτοποίηση των κόμβων. Κόμβοι οι οποίοι πληρούν ορισμένες προϋποθέσεις, όπως για παράδειγμα ορθή λειτουργία στο δίκτυο δομημένης επικάλυψης, αποκτούν πιστοποιητικά, τα οποία και χρησιμοποιούν στην συνέχεια ως ταυτότητα στο δίκτυο επικάλυψης. Κόμβοι που παραβιάζουν τους κανόνες λειτουργίας του συστήματος και *βλάπτουν* το σύστημα, απομονώνονται και ανακαλούνται τα πιστοποιητικά τους. Η χρήση μιας κεντρικής αρχής πιστοποίησης θα παραβίαζε τις προδιαγραφές των δικτύων δομημένης επικάλυψης που επιβάλλουν την συμμετρική λειτουργία μεταξύ των κόμβων και τη μη-εξάρτηση τους από κεντρικές οντότητες. Έτσι ορίζουμε πρωτόκολλα βάσει των οποίων μπορεί να κατανεμηθούν οι λειτουργίες της υποδομής δημοσίου κλειδιού στους κόμβους του δικτύου δομημένης επικάλυψης.

Τέλος, στο κεφάλαιο 5 παρουσιάζεται η αρχιτεκτονική μιας ψηφιακής βιβλιοθήκης πανεπιστημιακών δεδομένων. Οι βασικές της λειτουργίες υλοποιούνται μέσω της

αρχιτεκτονικής OGSA [9] για συστήματα υπολογιστικού πλέγματος. Μέσω της αρχιτεκτονικής OGSA οι υπηρεσίες της βιβλιοθήκης κατανέμονται στους διαθέσιμους υπολογιστικούς πόρους της πλατφόρμας υπολογιστικού πλέγματος. Παράλληλα, εξετάζοντας τα συστήματα υπολογιστικού πλέγματος εντοπίζονται ελλείψεις, οι οποίες καλύπτονται από τα συστήματα ομότιμων κόμβων με την βοήθεια των μηχανισμών εύρεσης και ασφάλειας που αναπτύχθηκαν στην πτυχιακή. Συγκεκριμένα, ο μηχανισμός εύρεσης του μητρώου που αναπτύχθηκε στην ενότητα 5.4.2.1, μπορεί να υλοποιηθεί μέσω ενός συστήματος δικτύου δομημένης επικάλυψης. Ερωτήματα εύρους επί των πόρων (π.χ. εύρεση εγγράφων με έτος δημοσίευσης μεταξύ 2005 και 2006) μπορούν να απαντηθούν μέσω των τεχνικών που αναπτύχθηκαν στο κεφάλαιο 3.

6.2 Προτάσεις για μελλοντική έρευνα

Η αξιοπιστία και γενικότερα η ορθή λειτουργία των συστημάτων ομότιμων κόμβων βασίζεται σε μεγάλο βαθμό στην ασφάλεια αλλά και στην ανοχή του συστήματος σε κακόβουλους κόμβους. Προκειμένου να ενθαρρύνεται η διαμοίραση των πόρων (πληροφορίες, αποθηκευτικός χώρος, επεξεργαστική ισχύ, εύρος ζώνης) μεταξύ των κόμβων και να αποθαρρύνεται η κακόβουλη, συνήθως εγωιστική, συμπεριφορά έχουν προταθεί *μηχανισμοί φήμης* [109, 110], με τη βοήθεια των οποίων οι κόμβοι αξιολογούν την *αξιοπιστία* των άλλων κόμβων. Συγκεκριμένα κάθε κόμβος βαθμολογεί τις αλληλεπιδράσεις που έχει με τους άλλους κόμβους του συστήματος και μέσω των μηχανισμών φήμης, η γνώση διαχέεται στο σύστημα ώστε οι άλλοι κόμβοι να αλληλεπιδρούν με τους πιο αξιόπιστους.

Ένας από τους πρώτους μηχανισμός φήμης είναι ο μηχανισμός που χρησιμοποιείται στην ηλεκτρονική πλατφόρμα αγορών eBay, βάσει του οποίου οι αγοραστές αξιολογούν τους πωλητές. Βέβαια στα κατανεμημένα συστήματα, όπως αυτά που βασίζονται σε δίκτυα ομότιμων κόμβων, η χρήση ενός κεντρικού μηχανισμού φήμης αποτελεί τροχοπέδη τόσο για την επεκτασιμότητα του συστήματος όσο και για την ασφάλειά του (κεντρικό σημείο αποτυχίας). Το γεγονός αυτό οδήγησε στην δημιουργία κατανεμημένων μηχανισμών φήμης [111, 112, 113] που βασίζονται σε δίκτυα δομημένης επικάλυψης για την συλλογή και την διανομή πληροφορίας σχετικά

με την φήμη των κόμβων. Στους κατανεμημένους μηχανισμούς φήμης προκειμένου να καταγράφεται το ιστορικό των *συναλλαγών* μιας συγκεκριμένης οντότητας απαιτείται ένα μόνιμο αναγνωριστικό με το οποίο να ταυτοποιείται η οντότητα. Για παράδειγμα στον μηχανισμό EigenTrust [111] προτείνεται η χρήση των αναγνωριστικών που παράγονται από την συνάρτηση κατακερματισμού, έστω H,του δικτύου δομημένης επικάλυψης (Chord ή CAN). Όλοι οι κόμβοι έχουν ένα αναγνωριστικό και η πληροφορία για την αξιοπιστία ενός κόμβου βρίσκεται επικοινωνώντας με τον διαχειριστή (Score Manager) του αναγνωριστικού που παράγεται από την συνάρτηση H. Βέβαια, όπως αναφέρθηκε και στην ενότητα 4.2.2 τα δίκτυα δομημένης επικάλυψης είναι ευάλωτα ενάντια σε μια κακόβουλη οντότητα, η οποία έχει στην κατοχή της πολλαπλά πλαστά αναγνωριστικά.

Πιστεύουμε ότι ο συνδυασμός του Chord-PKI που αναλύθηκε στην ενότητα 4 με τις τεχνικές φήμης [111, 112, 113] που βασίζονται σε δίκτυα δομημένης επικάλυψης μπορεί να βελτιώσει τα αδύναμα σημεία των μηχανισμών φήμης που απορρέουν από το πρόβλημα της ταυτοποίησης των κόμβων. Παράλληλα, ο μηχανισμός του Chord-PKI μπορεί να βασίζει τις αποφάσεις για πιστοποίηση των κόμβων σε μηχανισμούς φήμης. Συγκεκριμένα για να εκδοθεί ένα πιστοποιητικό κόμβου θα πρέπει η αξιοπιστία του κόμβου να είναι πάνω από ένα όριο το οποίο θα το ορίζουν οι κόμβοι πιστοποίησης. Από την στιγμή που οι κόμβοι θα έχουν στην κατοχή τους το πιστοποιητικό, αυτό θα χρησιμοποιείται από τους μηχανισμούς φήμης για τον χαρακτηρισμό των συναλλαγών των κόμβων. Ο σχεδιασμός και η αξιολόγηση της απόδοσης ενός υβριδικού συστήματος που θα συνδυάζει τους μηχανισμούς φήμης με κρυπτογραφικές τεχνικές του Chord-PKI αποτελεί μια υποσχόμενη προσέγγιση που αξίζει να μελετηθεί μελλοντικά.

Επιπλέον, στο μέλλον πρέπει να εξεταστούν βελτιστοποιήσεις που βασίζονται σε τεχνικές *προσωρινής αποθήκευσης* (caching). Συγκεκριμένα στο ΔΔΑ, οι εξυπηρετητές που μαθαίνουν, μέσω κάποιου ερωτήματος εύρους, ποιοι εξυπηρετητές είναι υπεύθυνοι για ένα συγκεκριμένο εύρος, τους αποθηκεύουν προσωρινά (η διάρκεια διατήρησης της πληροφορίας είναι μεταβλητή). Μελλοντικά ερωτήματα για συγκεκριμένα εύρη μπορούν να επωφεληθούν από την πληροφορία που είναι προσωρινά αποθηκευμένη στους εξυπηρετητές. Με αυτόν τον τρόπο μειώνεται ο αριθμός των μηνυμάτων που πρέπει να διαδοθούν στο δίκτυο και κατ᾽επέκτασιν αυξάνεται η

απόδοση των ερωτημάτων εύρους. Τέλος, αναφορικά με τις τεχνικές προσωρινής αποθήκευσης ενδιαφέρον έχει να βρεθεί το ελάχιστο χρονικό όριο αποθήκευσης των πληροφοριών σχετικά με τα ερωτήματα εύρους που βελτιστοποιούν την απόδοση των ερωτημάτων.

Βιβλιογραφία

[1] "Google," available at http://www.google.com/.

[2] "Yahoo," available at http://search.yahoo.com/.

[3] "Bing," available at http://www.bing.com/.

[4] "Napster," available at http://www.napster.com/.

[5] "Gnutella," available at http://www.gnutella.com.

[6] "BitTorrent," available at http://www.bittorent.com/.

[7] I. Foster and K. C., eds., *The Grid: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann, 1999.

[8] I. Foster and K. C., eds., *The Grid2: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann, 2004.

[9] "Open Grid Services Architecture," available at http://www.ogf.org/.

[10] I. Foster, "Globus Toolkit Version 4: Software for Service-Oriented Systems," *Computer Science and Technology*, vol. 21, pp. 513–520, July 2006.

[11] I. Stoica, R. Morris, D. Liben-Nowell, D. Karger, F. Dabek, and H. Balakrishnan, "Chord: a scalable peer-to-peer lookup protocol for internet applications," *IEEE/ACM Transactions on Networking*, vol. 11, no. 1, pp. 17–32, 2003.

[12] A. Avramidis, P. Kotzanikolaou, and C. Douligeris, "Chord-pki: Embedding a public key infrastructure into the chord overlay network," in *Public Key Infrastructure - EuroPKI'07 Proceedings*, vol. 4582 of *LNCS*, (Mallorca, Spain), pp. 354–361, Springer Berlin / Heidelberg, June 2007.

[13] "eMule," available at http://www.emule-project.net.

[14] "eDonkey 2000," available at http://www.edonkey2000.com.

[15] "seti@home," available at http://setiathome.ssl.berkeley.edu/.

[16] "folding@home," available at http://folding.stanford.edu/.

[17] S. Saroiu, K. P. Gummadi, R. J. Dunn, S. D. Gribble, and H. M. Levy, "An analysis of internet content delivery systems," *SIGOPS Operating Systems Review*, vol. 36, no. SI, pp. 315–327, 2002.

[18] K. P. Gummadi, R. J. Dunn, S. Saroiu, S. D. Gribble, H. M. Levy, and J. Zahorjan, "Measurement, modeling, and analysis of a peer-to-peer file-sharing workload," in *SOSP '03: Proceedings of the nineteenth ACM Symposium on Operating Systems Principles*, (Bolton Landing, NY, USA), pp. 314–329, ACM, 2003.

[19] J. Risson and T. Moors, "Survey of research towards robust peer-to-peer networks: search methods," *Computer Netwoks*, vol. 50, no. 17, pp. 3485–3521, 2006.

[20] S. Androutsellis-Theotokis and D. Spinellis, "A survey of peer-to-peer content distribution technologies," *ACM Computing Surveys*, vol. 36, no. 4, pp. 335–371, 2004.

[21] H. Beverly Yang, B. Garcia-Molina, "Designing a super-peer network," in *ICDE'03: Proceedings of the 19th International Conference on Data Engineering*, (Bangalore, India), pp. 49–60, IEEE, March 2003.

[22] Q. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker, "Search and replication in unstructured peer-to-peer networks," in *ICS '02: Proceedings of the 16th International Conference on Supercomputing*, (New York, NY, USA), pp. 84–95, ACM, 2002.

[23] H. Eriksson, "Mbone: the multicast backbone," *Communications of the ACM*, vol. 37, no. 8, pp. 54–60, 1994.

[24] D. Andersen, H. Balakrishnan, F. Kaashoek, and R. Morris, "Resilient overlay networks," in *SOSP '01: Proceedings of the eighteenth ACM Symposium on Operating Systems Principles*, (Banff, Alberta, Canada), pp. 131–145, ACM, 2001.

[25] "Akamai," available at http://www.akamai.com.

[26] 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*, (San Diego, CA, USA), pp. 161–172, ACM, 2001.

[27] B. Y. Zhao, L. Huang, J. Stribling, S. C. Rhea, A. D. Joseph, and J. D. Kubiatowicz, "Tapestry: A resilient global-scale overlay for service deployment," *IEEE Journal on Selected Areas in Communications*, vol. 22, pp. 41–53, 2004.

[28] P. Maymounkov and D. Mazières, "Kademlia: A peer-to-peer information system based on the xor metric," in *IPTPS'02: Proceedings for the 1st International Workshop on Peer-to-Peer Systems*, (Cambridge, MA, USA), pp. 53–65, March 2002.

[29] I. Gupta, K. Birman, P. Linga, A. Demers, and R. van Renesse, "Kelips: Building an efficient and stable p2p dht through increased memory and background overhead," in *IPTPS '03: Proceedings of the 2th International Workshop on Peer-to-Peer Systems*, (Berkeley, CA, USA), pp. 160–169, Springer-Verlag, February 2003.

[30] M. F. Kaashoek and D. R. Karger, "Koorde: A simple degree-optimal distributed hash table," in *IPTPS '03: Proceedings of the 2th International Workshop on Peer-to-Peer Systems*, (Berkeley, CA, USA), pp. 98–107, Springer-Verlag, February 2003.

[31] R. van Renesse and A. Bozdog, "Willow: Dht, aggregation, and publish subscribe in one protocol," in *IPTPS '04: Proceedings of the 3rd International Workshop on Peer-to-Peer Systems*, (San Diego, CA, USA), pp. 173–183, Springer-Verlag, February 2004.

[32] P. Ganesan, K. Gummadi, and H. Garcia-Molina, "Canon in g major: designing dhts with hierarchical structure," in *ICDCS '04: Proceedings of the 24th International Conference on Distributed Computing Systems*, (Tokyo, Japan), pp. 263–272, IEEE, March 2004.

[33] N. J. A. Harvey, M. B. Jones, S. Saroiu, M. Theimer, and A. Wolman, "Skipnet: a scalable overlay network with practical locality properties," in *USITS'03: Proceedings of the 4th conference on USENIX Symposium on Internet Technologies and Systems*, (Berkeley, CA, USA), pp. 9–9, USENIX Association, 2003.

[34] J. Aspnes and G. Shah, "Skip graphs," *ACM Transactions on Algorithms*, vol. 3, no. 4, p. 37, 2007.

[35] K. Aberer, P. Cudre-Mauroux, A. Datta, Z. Despotovic, M. Hauswirth, M. Punceva, and R. Schmidt, "P-grid: a self-organizing structured p2p system," *SIGMOD Record*, vol. 32, no. 3, pp. 29–33, 2003.

[36] D. Malkhi, M. Naor, and D. Ratajczak, "Viceroy: a scalable and dynamic emulation of the butterfly," in *PODC '02: Proceedings of the twenty-first annual symposium on Principles of distributed computing*, (New York, NY, USA), pp. 183–192, ACM, 2002.

[37] "Uniform Resource Identifier," available at http://tools.ietf.org/html/rfc3986.

[38] D. Karger, E. Lehman, T. Leighton, R. Panigrahy, M. Levine, and D. Lewin, "Consistent hashing and random trees: distributed caching protocols for relieving hot spots on the world wide web," in *STOC '97: Proceedings of the twenty-ninth annual ACM Symposium on Theory of Computing*, (El Paso, Texas, USA), pp. 654–663, ACM, 1997.

[39] C. G. Plaxton, R. Rajaraman, and A. W. Richa, "Accessing nearby copies of replicated objects in a distributed environment," in *SPAA '97: Proceedings of the ninth annual ACM Symposium on Parallel Algorithms and Architectures*, (Newport, Rhode Island, USA), pp. 311–320, ACM, 1997.

[40] W. Litwin, M.-A. Neimat, and D. A. Schneider, "Lh: Linear hashing for distributed files," in *SIGMOD '93: Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, (Washington, D.C., USA), pp. 327–336, ACM, 1993.

[41] I. Foster and A. Iamnitchi, "On death, taxes, and the convergence of peer-to-peer and grid computing," in *IPTPS 2003: Proceedings of the second*

International workshop on Peer-To-Peer Systems, (Berkeley, CA, USA), pp. 118–128, Springer, February 2003.

[42] J. Liang, R. Kumar, and K. W. Ross, "The fasttrack overlay: A measurement study," *Computer Networks*, vol. 50, no. 6, pp. 842 – 858, 2006.

[43] A. I. T. Rowstron and P. Druschel, "Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems," in *Middleware '01: Proceedings of the IFIP/ACM International Conference on Distributed Systems Platforms Heidelberg*, (London, UK), pp. 329–350, Springer-Verlag, 2001.

[44] A. Andrzejak and Z. Xu, "Scalable, efficient range queries for grid information services," in *P2P '02: Proceedings of the Second International Conference on Peer-to-Peer Computing*, (Washington, DC, USA), p. 33, IEEE Computer Society, 2002.

[45] "CiteSeerX," available at http://citeseerx.ist.psu.edu/.

[46] "Uniform Resource Locators (URL)," available at http://www.w3.org/Addressing/URL/url-spec.txt.

[47] A. S. Cheema, M. Muhammad, and I. Gupta, "Peer-to-peer discovery of computational resources for grid applications," in *GRID '05: Proceedings of the 6th IEEE/ACM International Workshop on Grid Computing*, (Washington, DC, USA), pp. 179–185, IEEE Computer Society, 2005.

[48] M. Gai, M. Frank, J. Chen, and P. Szekely, "Maan: A multi-attribute addressable network for grid information services," *Journal of Grid Computing*, vol. 2, no. 1, pp. 3–14, 2005.

[49] C. Schmidt and M. Parashar, "Enabling flexible queries with guarantees in p2p systems," *IEEE Internet Computing*, vol. 8, no. 3, pp. 19–26, 2004.

[50] J. Gao and P. Steenkiste, "An adaptive protocol for efficient support of range queries in dht-based systems," in *ICNP '04: Proceedings of the 12th IEEE International Conference on Network Protocols*, (Washington, DC, USA), pp. 239–250, IEEE Computer Society, 2004.

[51] D. Oppenheimer, J. Albrecht, D. Patterson, and A. Vahdat, "Design and Implementation Tradeoffs for Wide-Area Resource Discovery," in *HPDC*

05: Proceedings of the Fourteenth IEEE Symposium on High Performance Distributed Computing, (Triangle Park, North Carolina, USA), pp. 113–124, July 2005.

[52] P. Reynolds and A. Vahdat, "Efficient peer-to-peer keyword searching," in *Middleware '03: Proceedings of the ACM/IFIP/USENIX 2003 International Conference on Middleware*, (Rio de Janeiro, Brazil), pp. 21–40, Springer-Verlag New York, Inc., 2003.

[53] A. R. Bharambe, M. Agrawal, and S. Seshan, "Mercury: supporting scalable multi-attribute range queries," in *SIGCOMM '04: Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*, (Portland, Oregon, USA), pp. 353–366, ACM, 2004.

[54] A. Crainiceanu, P. Linga, A. Machanavajjhala, J. Gehrke, and J. Shanmugasundaram, "P-ring: an efficient and robust p2p range index structure," in *SIGMOD '07: Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, (Beijing, China), pp. 223–234, ACM, 2007.

[55] P. Ganesan, B. Yang, and H. Garcia-Molina, "One torus to rule them all: multi-dimensional queries in p2p systems," in *WebDB '04: Proceedings of the 7th International Workshop on the Web and Databases*, (Paris, France), pp. 19–24, ACM, 2004.

[56] P. Indyk, R. Motwani, P. Raghavan, and S. Vempala, "Locality-preserving hashing in multidimensional spaces," in *STOC '97: Proceedings of the twenty-ninth annual ACM Symposium on Theory of Computing*, (El Paso, Texas, USA), pp. 618–625, ACM, 1997.

[57] Y. Chawathe, S. Ramabhadran, S. Ratnasamy, A. LaMarca, S. Shenker, and J. Hellerstein, "A case study in building layered dht applications," in *SIGCOMM '05: Proceedings of the 2005 conference on applications, technologies, architectures, and protocols for computer communications*, (Philadelphia, Pennsylvania, USA), pp. 97–108, ACM, 2005.

[58] C. Zheng, G. Shen, S. Li, and S. Shenker, "Distributed Segment Tree: Support of Range Query and Cover Query over DHT," in *IPTPS06: Proceedings of*

the 5th International workshop on Peer-To-Peer Systems, (Santa Barbara, CA, USA), February 2006.

[59] D. Comer, "Ubiquitous b-tree," *ACM Computing Surveys*, vol. 11, no. 2, pp. 121–137, 1979.

[60] F. Dabek, M. F. Kaashoek, D. Karger, R. Morris, and I. Stoica, "Wide-area cooperative storage with cfs," in *SOSP '01: Proceedings of the eighteenth ACM Symposium on Operating Systems Principles*, (Banff, Alberta, Canada), pp. 202–215, ACM, 2001.

[61] J. Gao and P. Steenkiste, "Efficient support for range queries in dht-based systems," tech. rep., Carnegie Mellon University, 2003.

[62] G. S. Manku, "Balanced binary trees for id management and load balance in distributed hash tables," in *PODC '04: Proceedings of the twenty-third annual ACM Symposium on Principles of Distributed Computing*, (St. John's, Newfoundland, Canada), pp. 197–205, ACM, 2004.

[63] M. Naor and U. Wieder, "Novel architectures for p2p applications: The continuous-discrete approach," *ACM Trans. Algorithms*, vol. 3, no. 3, p. 34, 2007.

[64] G. S. Manku, M. Naor, and U. Wieder, "Know thy neighbor's neighbor: the power of lookahead in randomized p2p networks," in *STOC '04: Proceedings of the thirty-sixth annual ACM Symposium on Theory of Computing*, (Chicago, IL, USA), pp. 54–63, ACM, 2004.

[65] J. R. Douceur, "The sybil attack," in *IPTPS '01: Revised Papers from the First International Workshop on Peer-to-Peer Systems*, (London, UK), pp. 251–260, Springer-Verlag, 2002.

[66] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*. CRC Press, 2001.

[67] E. Sit and R. Morris, "Security considerations for peer-to-peer distributed hash tables," in *IPTPS 2002: Proceedings of the First International Workshop on Peer-to-Peer Systems*, (Cambridge, MA, USA), pp. 261–269, Springer Berlin / Heidelberg, March 2002.

[68] M. Castro, P. Druschel, A. Ganesh, A. Rowstron, and D. S. Wallach, "Secure routing for structured peer-to-peer overlay networks," *SIGOPS Oper. Syst. Rev.*, vol. 36, pp. 299–314, 2002.

[69] N. Borisov, "Computational puzzles as sybil defenses," in *P2P '06: Proceedings of the Sixth IEEE International Conference on Peer-to-Peer Computing*, (Washington, DC, USA), pp. 171–176, IEEE Computer Society, 2006.

[70] A. Singh, M. Castro, P. Druschel, and A. Rowstron, "Defending against eclipse attacks on overlay networks," in *EW11: Proceedings of the 11th workshop on ACM SIGOPS European workshop*, (Leuven, Belgium), p. 21, ACM, 2004.

[71] T. Condie, V. Kacholia, S. Sankararaman, J. M. Hellerstein, and P. Maniatis, "Induced churn as shelter from routing-table poisoning," in *NDSS: In Proceedings of the 13th Annual Network and Distributed System Security Symposium*, 2006.

[72] H. Kirsten and K. John, "Asymptotically efficient approaches to fault-tolerance in peer-to-peer networks," in *Distributed Computing*, vol. 2848, (Sorrento, Italy), pp. 321–336, Springer Berlin / Heidelberg, 2003.

[73] M. Artigas, P. Lopez, and A. Skarmeta, "A novel methodology for constructing secure multipath overlays," *Internet Computing, IEEE*, vol. 9, no. 6, pp. 50–57, 2005.

[74] L. Ganesh and B. Zhao, "Identity theft protection in structured overlays," in *Secure Network Protocols, 2005. (NPSec). 1st IEEE ICNP Workshop on*, (Boston, Massachusetts, USA), pp. 49–54, November 2005.

[75] C. Harvesf and D. M. Blough, "The effect of replica placement on routing robustness in distributed hash tables," in *P2P '06: Proceedings of the Sixth IEEE International Conference on Peer-to-Peer Computing*, (Washington, DC, USA), pp. 57–6, IEEE Computer Society, 2006.

[76] R. Housley, W. Polk, W. Ford, and D. Solo, "Internet x.509 public key infrastructure certificate and certificate revocation list (crl) profile." RFC, 2002.

[77] B. Schneier, *Applied Cryptography*. John Wiley @ Sons, 1994.

[78] Y. Desmedt and Y. Frankel, "Threshold cryptosystems," in *CRYPTO '89: Proceedings of the 9th Annual International Cryptology Conference on Advances in Cryptology*, (London, UK), pp. 307–315, Springer-Verlag, 1990.

[79] V. Shoup and R. Gennaro, "Securing threshold cryptosystems against chosen ciphertext attack," *Journal of Cryptology*, vol. 15, no. 2, pp. 75–96, 2002.

[80] M.-S. Hwang, E. J.-L. Lu, and I.-C. Lin, "A practical (t, n) threshold proxy signature scheme based on the rsa cryptosystem," *IEEE Trans. on Knowl. and Data Eng.*, vol. 15, no. 6, pp. 1552–1560, 2003.

[81] Y. Frankel, P. Gemmell, and M. Yung, "Witness-based cryptographic program checking and robust function sharing," in *STOC '96: Proceedings of the twenty-eighth annual ACM Symposium on Theory of Computing*, (Philadelphia, Pennsylvania, USA), pp. 499–508, ACM, 1996.

[82] R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin, "Robust and efficient sharing of rsa functions," in *CRYPTO '96: Proceedings of the 16th Annual International Cryptology Conference on Advances in Cryptology*, (Santa Barbara, California, USA), pp. 157–172, Springer-Verlag, August 1996.

[83] T. Rabin, "A simplified approach to threshold and proactive rsa," in *CRYPTO '98: Proceedings of the 18th Annual International Cryptology Conference on Advances in Cryptology*, (Santa Barbara, California, USA), pp. 89–104, Springer-Verlag, August 1998.

[84] T. P. Pedersen, "A threshold cryptosystem without a trusted party," in *Advances in Cryptology — EUROCRYPT '91*, vol. 547, pp. 522–526, Springer-Verlag, 1991.

[85] A. Herzberg, M. Jakobsson, S. Jarecki, H. Krawczyk, and M. Yung, "Proactive public key and signature systems," in *CCS '97: Proceedings of the 4th ACM conference on Computer and communications security*, (Zurich, Switzerland), pp. 100–110, ACM, 1997.

[86] A. Herzberg, S. Jarecki, H. Krawczyk, and M. Yung, "Proactive secret sharing or: How to cope with perpetual leakage," in *CRYPTO '95: Proceedings of the 15th Annual International Cryptology Conference on Advances in Cryptology*, (London, UK), pp. 339–352, Springer-Verlag, 1995.

[87] Y. Frankel, P. Gemmell, P. D. MacKenzie, and M. Yung, "Proactive rsa," in *CRYPTO '97: Proceedings of the 17th Annual International Cryptology Conference on Advances in Cryptology*, (London, UK), pp. 440–454, Springer-Verlag, 1997.

[88] R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin, "Robust threshold dss signatures," in *EUROCRYPT 96: Proceedings on the Theory and Applications of Cryptographic Techniques*, (Saragossa, Spain), pp. 354–371, May 1996.

[89] M. Nystrom and B. Kaliski, "PKCS #10: Certification request syntax specification version 1.7." RFC, 2000.

[90] E. Fox, "Digital libraries," *Computer*, vol. 26, no. 11, pp. 79–81, 1993.

[91] "Web Services," available at http://www.w3.org/TR/ws-arch/.

[92] "Resource Description Framework," available at http://www.w3.org/RDF/.

[93] "Globus Toolkit," available at http://www.globus.org.

[94] I. Foster, C. Kesselman, and S. Tuecke, "The anatomy of the grid: Enabling scalable virtual organizations," *International Journal of High Performance Computing Applications*, vol. 15, no. 3, pp. 200–222, 2001.

[95] "Global Grid Forum," available at http://www.ggf.org.

[96] R. Stevens, P. Woodward, T. DeFanti, and C. Catlett, "From the i-way to the national technology grid," *Communications of the ACM*, vol. 40, no. 11, pp. 50–60, 1997.

[97] "gLite Middleware," available at http://glite.web.cern.ch/glite/.

[98] "Extensible Markup Language (XML) 1.0 (Fifth Edition)," available at http://www.w3.org/TR/xml/.

[99] "Simple Object Access Protocol (SOAP)," available at http://www.w3.org/TR/soap/.

[100] "Web Services Description Language (WSDL) 1.1," available at http://www.w3.org/TR/wsdl/.

[101] "Web Ontology Language," available at http://www.w3.org/2007/OWL/.

[102] F. Tao, S. Cox, L. Chen, N. Shadbolt, F. Xu, C. Puleston, C. Goble, and W. Song, "Towards the semantic grid: Enriching content for management and reuse," in *Proceedings of Delivering e-Science, UK e-Science All-hand Conference 2003*, pp. 695–702, 2003.

[103] "The Dublin Core Metadata Initiative," available at http://dublincore.org/documents/dcq-rdf-xml/.

[104] "The Digital Object Identifier System," available at http://www.doi.org.

[105] S. Tuecke, K. Czajkowski, I. Foster, J. Frey, S. Graham, C. Kesselman, T. Maquire, T. Sandholm, D. Snelling, and P. Vanderbilt, "Open grid services infrastructure (ogsi) version 1.0," tech. rep., Global Grid Forum, http://www.globus.org, June 2003.

[106] I. Foster, C. Kesselman, J. Nick, and S. Tuecke, "The physiology of the grid: An open grid services architecture for distributed systems integration," tech. rep., OGSI-WG, Global Grid Forum, http://www.globus.org/, June 2002.

[107] B. Allcock, J. Bester, J. Bresnahan, A. Chervenak, I. Foster, C. Kesselman, S. Meder, V. Nefedova, D. Quesnel, and S. Tuecke, "Data Management and Transfer in High Performance Computational Grid Environments," *Parallel Computing Journal*, vol. 28, no. 5, pp. 749–771, 2002.

[108] K. Czajkowski, S. Fitzgerald, I. Foster, and C. Kesselman, "Grid information services for distributed resource sharing," in *HPDC-10: Proceedings of the Tenth IEEE International Symposium on High-Performance Distributed Computing*, (San Francisco CA), pp. 181–184, IEEE Press, August 2001.

[109] S. Marti and H. Garcia-Molina, "Taxonomy of trust: categorizing p2p reputation systems," *Computer Networks*, vol. 50, no. 4, pp. 472–484, 2006.

[110] K. Hoffman, D. Zage, and C. Nita-rotaru, "A survey of attack and defense techniques for reputation systems." To appear in ACM Computing Surveys, vol. 42, no. 1, March 2010.

[111] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, "The eigentrust algorithm for reputation management in p2p networks," in *WWW '03: Proceedings of*

the 12th international conference on World Wide Web, (Budapest, Hungary), pp. 640–651, ACM, 2003.

[112] L. Xiong and L. Liu, "Peertrust: supporting reputation-based trust for peer-to-peer electronic communities," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 7, pp. 843–857, 2004.

[113] R. Zhou and K. Hwang, "Powertrust: A robust and scalable reputation system for trusted peer-to-peer computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 18, no. 4, pp. 460–473, 2007.