



**«Βέλτιστη διαχείριση υποδομών κινητού
υπολογιστικού νέφους»
Optimal management of mobile edge
computing infrastructures**

Διπλωματική εργασία

Κεφαλόπουλος Παναγιώτης
ΜΨΕ1701

Επιβλέπων Καθηγητής

Καθ. Άγγελος Ρούσκας,

Τμήμα Ψηφιακών Συστημάτων Πανεπιστήμιο Πειραιώς

Περιεχόμενα

Περίληψη	4
Ευχαριστίες	6
Κεφάλαιο 1 ^ο Υπολογιστικό νέφος (Cloud Computing).....	7
1.1 Ορισμός υπολογιστικού νέφους	7
1.2 Βασικά χαρακτηριστικά υπολογιστικού νέφους.....	8
1.3 Πλεονεκτήματα υπολογιστικού νέφους	9
1.4 Μειονεκτήματα υπολογιστικού νέφους	10
1.5 Μοντέλα παροχής υπηρεσιών υπολογιστικού νέφους	10
1.5.1 Υποδομή ως Υπηρεσία (IaaS)	10
1.5.2 Πλατφόρμα ως Υπηρεσία (PaaS).....	11
1.5.3 Λογισμικό ως Υπηρεσία (SaaS).....	12
1.5.4 Διαφορές μεταξύ των μοντέλων	12
1.6 Μοντέλα ανάπτυξης υπολογιστικών νεφών	13
1.6.1 Public Cloud (Δημόσια νέφη)	14
1.6.2 Private Cloud (Ιδιωτικά νέφη)	14
1.6.3 Community Cloud (Κοινοτικά νέφη).....	15
1.6.4 Hybrid Cloud (Υβριδικά νέφη).....	15
Κεφάλαιο 2 ^ο Κινητή υπολογιστική στα άκρα του δικτύου (Mobile Edge Computing)	17
2.1 Ορισμός κινητού υπολογιστικού νέφους (Mobile Cloud Computing).....	17
2.2 Πλεονεκτήματα κινητού υπολογιστικού νέφους.....	18
2.3 Cloudlet	19
2.3.1 Ορισμός του Cloudlet	19
2.3.2 Βασικά χαρακτηριστικά ενός Cloudlet	20
2.3.3 Διαφορές μεταξύ Cloud και Cloudlet	21
2.4 Κινητή υπολογιστική στα άκρα του δικτύου (Mobile Edge Computing)	21
2.5 Πλεονεκτήματα Mobile Edge Computing.....	22
2.6 Προκλήσεις σχετικά με το Mobile Edge Computing	23
Κεφάλαιο 3 ^ο Τοπολογίες δικτύου και αλγόριθμοι.....	25
3.1 Τοπολογίες δικτύου	25
3.1.1 Τοπολογία scale free	25
3.1.2 Τοπολογία lattice.....	26
3.2 Ανάλυση των τοπολογιών που σχεδιάστηκαν	27
3.3 Ευριστικοί αλγόριθμοι	29

3.4 Μεθοδολογία αλγορίθμων	30
Κεφάλαιο 4 ^ο Μελέτη αποτελεσμάτων	40
4.1 Σενάριο 1 ^ο Uniform κατανομή.....	41
4.1.1 Τοπολογία scale free	41
4.1.2 Τοπολογία lattice.....	48
4.2 Σενάριο 2 ^ο Κατανομή στους κεντρικούς κόμβους	54
4.2.1 Τοπολογία scale free	54
4.2.2 Τοπολογία lattice.....	60
4.3 Σενάριο 3 ^ο Κατανομή στους περιφερειακούς κόμβους.....	66
4.3.1 Τοπολογία scale free	66
4.3.2 Τοπολογία lattice.....	71
4.4 Συμπεράσματα	78
Βιβλιογραφία	81

Περίληψη

Δεδομένης της ανάπτυξης της τεχνολογίας των έξυπνων τερματικών συσκευών (smartphones), του υψηλού εύρους ζώνης (bandwidth) και της πολύ χαμηλής καθυστέρησης από άκρο σε άκρο, που προσφέρουν τεχνολογίες όπως τα δίκτυα πέμπτης γενιάς (5G), αλλά και του μεγάλου αριθμού των συνδεδεμένων συσκευών, παρατηρείται ραγδαία αύξηση των αναγκών που καλείται να εξυπηρετήσει ένα δίκτυο παρόχου κινητών επικοινωνιών. Συνεπώς κρίνεται απαραίτητη η υιοθέτηση τεχνολογιών, οι οποίες θα μπορέσουν να ανταποκριθούν στις ολοένα και αυξανόμενες ανάγκες των χρηστών. Τεχνολογίες όπως η κινητή υπολογιστική στα άκρα του δικτύου (mobile edge computing), θα αποτελέσουν ακρογωνιαίους λίθους στην προσπάθεια να έρθουν πιο κοντά στον τελικό χρήστη, οι πόροι και οι υπηρεσίες του δικτύου. Συγκεκριμένα, η υπολογιστική στα άκρα των δικτύων κινητής, προσφέρει τα πλεονεκτήματα του υπολογιστικού νέφους στους κινητούς χρήστες και συνεισφέρει στην άρση των περιορισμών της επεξεργαστικής ισχύος και της χωρητικότητας των κινητών συσκευών, ενώ βοηθάει στην μείωση της καθυστέρησης της απόκρισης των εφαρμογών και στην επιμήκυνση της διάρκειας ζωής της μπαταρίας των συσκευών, μεταφέροντας την εκτέλεση απαιτητικών εφαρμογών στο νέφος.

Σκοπός της παρούσας διπλωματικής, είναι η μελέτη της απόδοσης ευριστικών αλγορίθμων που αποσκοπούν στην ενεργοποίηση κατάλληλων κόμβων πρόσβασης (access nodes) ώστε να λειτουργούν ταυτόχρονα και ως κόμβοι παροχής υπηρεσιών κινητής υπολογιστικής (service nodes) στα άκρα ενός δικτύου παρόχου. Η μελέτη και σύγκριση γίνεται ως προς την καθυστέρηση του δικτύου (latency), καθώς και την εξισορρόπηση της κατανομής του φορτίου των χρηστών (variation), στους κόμβους υπηρεσιών του ασύρματου δικτύου (wireless edge network).

Στο πρώτο κεφάλαιο περιγράφεται η τεχνολογία της υπολογιστικής νέφους (cloud computing), παρατίθενται τα βασικά χαρακτηριστικά της εν λόγω τεχνολογίας, τα πλεονεκτήματα και τα μειονεκτήματα που απορρέουν από την χρήση της. Αναλύονται ακόμα τα μοντέλα παροχής υπηρεσιών υπολογιστικού νέφους (SPI model) και τα είδη των υπολογιστικών νεφών.

Στο δεύτερο κεφάλαιο αναλύεται ο όρος του κινητού υπολογιστικού νέφους (mobile cloud computing) και γίνεται ιδιαίτερη μνεία στην κινητή υπολογιστική στα άκρα του δικτύου (mobile edge computing) και συγκεκριμένα στα βασικά

χαρακτηριστικά της, στην λειτουργία των cloudlets καθώς και στις προκλήσεις που καλείται να ανταπεξέλθει.

Στο τρίτο κεφάλαιο γίνεται ανάλυση τόσο των δύο τοπολογιών δικτύου που σχεδιάστηκαν για τις ανάγκες της μελέτης, όσο και των δύο ευριστικών αλγορίθμων που μελετήθηκαν. Πραγματοποιείται αναλυτική περιγραφή των ευριστικών αλγορίθμων και αναφέρονται τα σενάρια στα οποία θα βασιστεί η μελέτη.

Στο τέταρτο κεφάλαιο παρουσιάζονται τα συμπεράσματα που έχουν εξαχθεί, από την μελέτη των αποτελεσμάτων, για κάθε ένα από τα σενάρια που τέθηκαν. Παραθέτονται πίνακες και γραφήματα με την σύγκριση των αλγορίθμων.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω θερμά τον Καθηγητή μου κ. Άγγελο Ρούσκα, για την πολύτιμη συνεισφορά και την καθοδήγησή του, στην διεκπεραίωση της διπλωματικής μου εργασίας.

Επιπλέον, ιδιαίτερες ευχαριστίες θα ήθελα να απευθύνω στον υποψήφιο διδάκτορα κ. Ευθύμιο Οικονόμου, για την άριστη συνεργασία καθώς για τη συνεχή υποστήριξη και βοήθειά καθ' όλη τη διάρκεια της διπλωματικής.

Κεφάλαιο 1^ο Υπολογιστικό νέφος (Cloud Computing)

1.1 Ορισμός υπολογιστικού νέφους

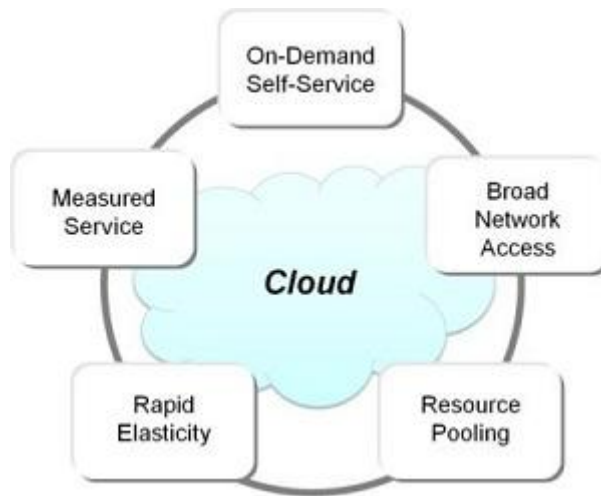
Το υπολογιστικό νέφος (cloud computing) είναι μια νέα τάση για τα υπολογιστικά συστήματα, όπου δυναμικά κλιμακούμενοι (dynamically scalable) και εικονοποιημένοι (virtualized) πόροι, προσφέρονται σαν υπηρεσίες μέσω του διαδικτύου [1].

Μέσω αυτής της τάσης πραγματοποιείται μια στροφή από το υλικό (hardware) προς το λογισμικό (software), προκειμένου να μειωθούν τα σχετικά κόστη που αφορούν την διαχείριση και την συντήρηση εξοπλισμού, καθώς και να προσαρμοστεί η προσφορά των υπολογιστικών πόρων ανάλογα με τις ανάγκες, οι οποίες είναι μεταβαλλόμενες.

Μια βασική αρχή του cloud computing, είναι ο διαμοιρασμός των υπολογιστικών πόρων μεταξύ των χρηστών, προσφέροντας πολλαπλά οφέλη τόσο για τους τελικούς χρήστες, οι οποίοι δεν επιβαρύνονται με έξοδα συντήρησης πάγιου εξοπλισμού, όσο και περιβαλλοντικά από την διαστασιοποίηση των αναγκών σε υπολογιστική ισχύ. Αυτό έχει ως αποτέλεσμα, την μεγιστοποίηση της υπολογιστικής ισχύος, ενώ παράλληλα μειώνεται το ενεργειακό κόστος [2].

Η άνοδος του cloud computing στηρίχτηκε στην πρόοδο συγκριμένων τεχνολογιών, οι οποίες αποτελούν βασικά/δομικά στοιχεία του. Η ανάπτυξη τεχνολογιών, σε επίπεδο hardware, όπως η εικονικοποίηση (virtualization) [3] και η δυνατότητα για αυξημένη επεξεργαστική ισχύ των τερματικών συσκευών (multi-core chips) [4], η διαδεδομένη χρήση του internet καθώς και η πληθώρα εφαρμογών σε αυτό (web services), η χρήση συστημάτων καταμεμημένης υπολογιστικής (clusters, grids) καθώς και η ραγδαία ανάπτυξη των συστημάτων διαχείρισης υπολογιστικών πόρων (autonomic computing, data centers), αποτέλεσαν τις βάσεις για την ανάπτυξη και την εδραίωση του cloud computing ως την βασική προσέγγιση που ακολουθείται στα πληροφοριακά συστήματα παγκοσμίως.

1.2 Βασικά χαρακτηριστικά υπολογιστικού νέφους



Εικόνα 1. Βασικά χαρακτηριστικά υπολογιστικού νέφους

Τα βασικά χαρακτηριστικά ενός υπολογιστικού νέφους [5],[6] αναλύονται ως εξής.

- **Χρήση κατά απαίτηση (On demand self-service):** Ο τελικός χρήστης μπορεί να έχει πρόσβαση στους υπολογιστικούς πόρους του υπολογιστικού νέφους, χωρίς να απαιτείται καμία άμεση αλληλεπίδραση με τον πάροχο της εκάστοτε υπηρεσίας.
- **Ευρεία πρόσβαση στο δίκτυο:** Οι δυνατότητες του υπολογιστικού νέφους είναι προσβάσιμες στους χρήστες μέσω του διαδικτύου, από διαφορετικές πλατφόρμες και συσκευές.
- **Διάθεση των υπολογιστικών πόρων (resource pooling):** Οι διαθέσιμοι υπολογιστικοί πόροι του παρόχου, βρίσκονται συγκεντρωμένοι έτσι ώστε να μπορούν να εξυπηρετήσουν πολλούς χρήστες, οι οποίοι προσαρμόζονται ανάλογα με τις απαιτήσεις του κάθε χρήστη. Κάποια παραδείγματα τέτοιων πόρων είναι ο αποθηκευτικός χώρος, η επεξεργαστική ισχύ (cpu), η μνήμη (ram) και το εύρος ζώνης δικτύου (bandwidth).
- **Ταχεία ελαστικότητα δέσμευσης και αποδέσμευσης των πόρων:** Οι υπολογιστικοί πόροι μπορούν να δεσμευθούν και να αποδεσμευτούν αυτόματα ανάλογα με την ζήτηση του κάθε χρήστη.
- **Τιμολόγηση βάσει χρήσης:** Τα συστήματα υπολογιστικού νέφους έχουν την δυνατότητα να ελέγχουν, να καταγράφουν και να βελτιστοποιούν την χρήση

των πόρων χρησιμοποιώντας κάποια εξειδικευμένα συστήματα μέτρησης και τιμολόγησης, ανάλογα με το είδος της υπηρεσίας.

1.3 Πλεονεκτήματα υπολογιστικού νέφους

Τα κυριότερα πλεονεκτήματα από την χρήση υπολογιστικού νέφους [9],[10] είναι:

Η κλιμάκωση (scalability): Το Cloud παρέχει την δυνατότητα για δυναμικό καταμερισμό των πόρων, ανάλογα με τις ανάγκες προς εξυπηρέτηση. Υπάρχουν δύο είδη κλιμάκωσης, η οριζόντια (horizontal scalability) στην οποία αυξάνεται ο αριθμός των πόρων προς διάθεση (π.χ. ο αριθμός των virtual machines) και η κάθετη (vertical scalability) στην οποία αυξάνεται ο διαθέσιμος όγκος των ήδη διαθέσιμων πόρων (π.χ. το μέγεθος του υπάρχοντος virtual machine).

Η ελαστικότητα (elasticity): Απορρέει βάσει της επεκτασιμότητας και αναφέρεται στην δυνατότητα για άμεση προσαρμογή των πόρων ανάλογα με τις ζητούμενες ανάγκες. Η ελαστικότητα αναφέρεται στην ικανότητα της αναδιάταξης των προς παροχή υπολογιστικών πόρων, είτε με αύξηση είτε με μείωση, για να επέλθει το βέλτιστο αποτέλεσμα στην σχέση αναγκών και υπολογιστικών πόρων.

Η κοινή χρήση των πόρων (multitenancy): Γίνεται διαμοιρασμός των υπολογιστικών πόρων, είτε σε επίπεδο hardware (cpu,ram) είτε σε επίπεδο software (libraries,applications,OS), σε ένα ευρύ σύνολο χρηστών [7].

Άμεση προσβασιμότητα: Παρέχει την δυνατότητα για πρόσβαση στις υπηρεσίες από οπουδήποτε, ανεξάρτητα από τα χαρακτηριστικά των τελικών χρηστών, είτε σε επίπεδο hardware (device) είτε σε επίπεδο λογισμικού (windows/linux).

Ανθεκτικότητα (Resiliency): Προσφέρεται η δυνατότητα για άμεση μεταγωγή σε εφεδρικό σύστημα, σε περίπτωση που διαπιστωθεί κάποια δυσλειτουργία σε επίπεδο υλικού, εξαλείφοντας τον κίνδυνο για κατάρρευση του συστήματος (single point of failure) [8].

Δυνατότητα για μέτρηση της χρήσης: Αναφέρεται στην δυνατότητα για παρακολούθηση της απόδοσης των πόρων τόσο από την πλευρά των παρόχων, είτε για monitoring είτε για θέματα τιμολόγησης, όσο και από την πλευρά των χρηστών νέφους για παρακολούθηση των απαιτήσεων τους.

1.4 Μειονεκτήματα υπολογιστικού νέφους

Είναι σαφές πως πέρα από τα πλεονεκτήματα, τα υπολογιστικά νέφη έχουν μειονεκτήματα [10],[11], τα κυριότερα εκ των οποίων είναι:

Άμεση εξάρτηση από το internet: Για την προσπέλαση στις υπηρεσίες και στους πόρους του υπολογιστικού νέφους, είναι απαραίτητη η ύπαρξη σύνδεσης στο internet. Ενδεχόμενη απουσία, οδηγεί στην αδυναμία χρήσης των υπηρεσιών. Για τους ίδιους λόγους επηρεάζεται σε μεγάλο βαθμό και από την ποιότητα της υπηρεσίας διασύνδεσης.

Ασφάλεια δεδομένων: Δεδομένου ότι μια επιχείρηση υλοποιεί τις εφαρμογές της στο Cloud, υπάρχει κίνδυνος σχετικά με τα ευαίσθητα δεδομένα, τα οποία διαχειρίζεται και τα οποία μπορεί να είναι εκτεθειμένα, δεδομένου ότι δεν πραγματοποιούνται από την ίδια οι ενέργειες για την προφύλαξη τους.

Έμμεση εξάρτηση χρήστη παρόχου: Είναι πολύ δύσκολο για μια επιχείρηση η αλλαγή παρόχου, δεδομένης της δυσκολίας μετάβασης των κρίσιμων για αυτή εφαρμογών ή λειτουργιών που εκτελούνται στο Cloud. Συντελείται έτσι κάποιου είδους κλείδωμα του χρήστη στον πάροχο, το οποίο ονομάζεται *vendor lock-in*.

1.5 Μοντέλα παροχής υπηρεσιών υπολογιστικού νέφους

1.5.1 Υποδομή ως Υπηρεσία (IaaS)

Το μοντέλο της υποδομής ως υπηρεσίας-IaaS (Infrastructure as a service) αποτελείται από το σύνολο των IT πόρων, οι οποίοι είναι προς διάθεση προς τους τελικούς χρήστες υπηρεσιών cloud, μέσω δικτύου δεδομένων. Πρόκειται ουσιαστικά για υλικοτεχνική υποδομή (hardware), όπως διακομιστές (servers), αποθηκευτικά μέσα (storage), υποδομές δικτύου (network) και σχετικού με αυτά λογισμικό, όπως τεχνολογίες εικονικοποίησης (virtualization technology) και συστήματα διαχείρισης αρχείων (file systems). Πρόκειται για μια εξέλιξη του μοντέλου της παραδοσιακής φιλοξενίας αρχείων, η οποία επιτρέπει στους τελικούς χρήστες να δεσμεύουν και να χρησιμοποιούν δυναμικά τις υποδομές, ανάλογα με την προς κάλυψη ζήτηση. Στο μοντέλο αυτό ο πάροχος IaaS δεν συμμετέχει ενεργά στις εργασίες υποστήριξης, για τις οποίες υπεύθυνοι είναι χρήστες, στην βάση της λογικής της διαχείρισης των υποδομών σε δικό τους περιβάλλον (data center) [12].

Τα βασικά χαρακτηριστικά του μοντέλου IaaS είναι:

- Η πληθώρα των χρηστών οι οποίοι εξυπηρετούνται από τους ίδιους πόρους.
- Η δυναμική κατανομή πόρων.
- Τιμολόγηση ανάλογα με την χρήση.

Το μοντέλο IaaS ενδείκνυται κυρίως για εταιρείες μικρού μεγέθους ή startups, οι οποίες δεν έχουν τα απαιτούμενα κεφάλαια για αγορά πάγιου εξοπλισμού. Επίσης για εταιρείες με κλιμακούμενη ανάγκη για υπολογιστικούς πόρους.

Οι κυριότερες εφαρμογές IaaS είναι:

- Amazon Web Services (AWS)
- Cisco Metapod
- Microsoft Azure
- Google Compute Engine (GCE)
- DigitalOcean
- Linode
- Rackspace

1.5.2 Πλατφόρμα ως Υπηρεσία (PaaS)

Το μοντέλο της πλατφόρμας ως υπηρεσίας-PaaS (Platform as a Service) περιλαμβάνει το σύνολο του προσφερόμενου λογισμικού, όπως ολοκληρωμένα περιβάλλοντα ανάπτυξης λογισμικού, γλώσσες προγραμματισμού, βάσεις δεδομένων και λοιπά εργαλεία ανάπτυξης, με την βοήθεια των οποίων ο τελικός χρήστης μπορεί να αναπτύξει το δικό του λογισμικό.

Το μοντέλο Platform as a Service (PaaS), είναι ένα περιβάλλον ανάπτυξης, εγκατάστασης και διανομής λογισμικού. Διευκολύνει την ανάπτυξη του λογισμικού και τον διαμοιρασμό του, απαλλαγμένο από το κόστος και την πολυπλοκότητα της αγοράς και της διαχείρισης της αναγκαίας υποδομής, ενώ ταυτόχρονα παρέχει όλες τις διευκολύνσεις που απαιτούνται για την υποστήριξη του κύκλου ζωής του λογισμικού. Με αυτόν τον τρόπο μειώνεται η δυσκολία για την δημιουργία λογισμικού, λόγω της πληθώρας των παρεχόμενων εργαλείων, συνάμα διευκολύνεται η δυνατότητα διανομής του εν λόγω λογισμικού [13].

Τα βασικά χαρακτηριστικά του μοντέλου PaaS είναι:

- Βασίζεται στην εικονικοποίηση (virtualization).
- Υπάρχει ένα κοινό περιβάλλον ανάπτυξης για τους τελικούς χρήστες.

- Παρέχει ολοκληρωμένες υπηρεσίες δικτύου και βάσεις δεδομένων.
- Δυνατότητα πέρα από ανάπτυξη, για δοκιμή και υλοποίηση.

Οι κυριότερες εφαρμογές PaaS είναι:

- AWS Elastic Beanstalk
- Windows Azure
- Heroku
- Force.com
- Google App Engine
- Apache Stratos
- OpenShift

1.5.3 Λογισμικό ως Υπηρεσία (SaaS)

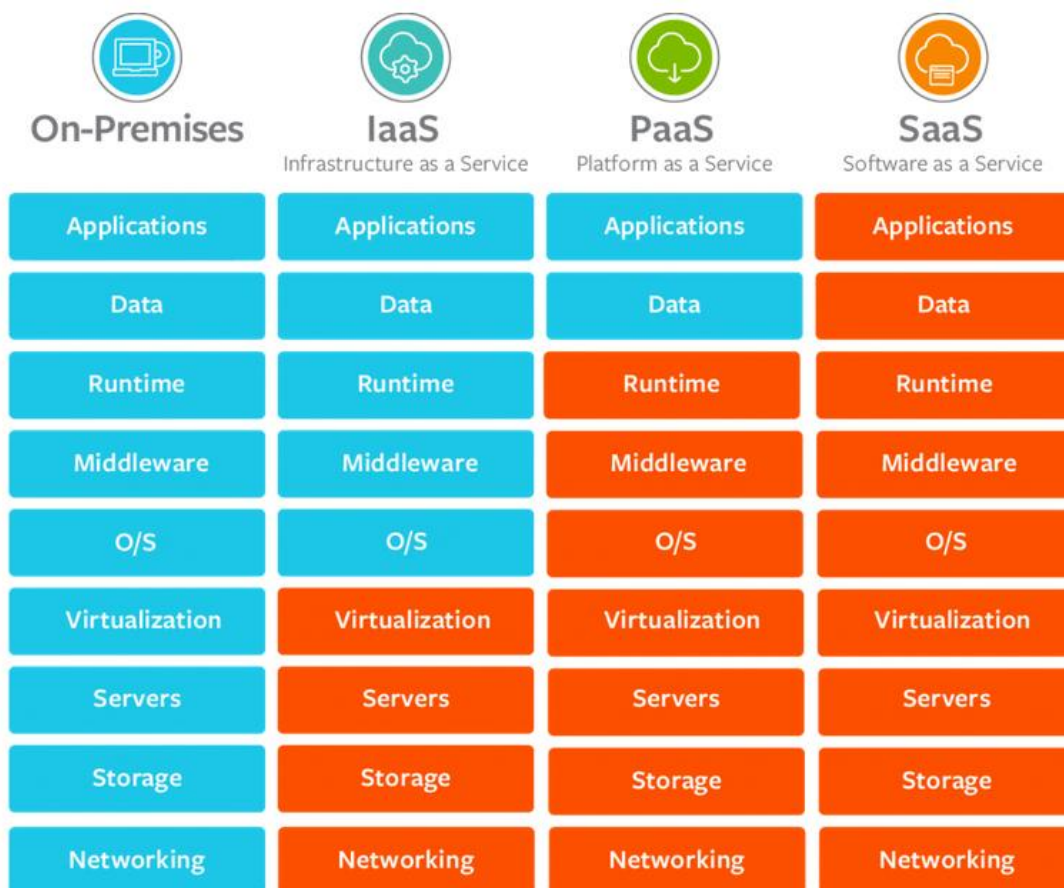
Το μοντέλο της εφαρμογής ως υπηρεσίας-SaaS (Software as a Service), παρέχει στους τελικούς χρήστες την δυνατότητα για χρήση των εφαρμογών του παρόχου, εν αντιθέσει με το PaaS στο οποίο εκτελούσαν τις δικές τους εφαρμογές. Συνεπώς οι τελικοί χρήστες δεν έχουν τον έλεγχο ή την δικαιοδοσία να προβούν σε αλλαγές και να χειρίζονται τους υπολογιστικούς πόρους ή τις επιμέρους εφαρμογές. Σε ένα μοντέλο SaaS ο πελάτης δεν αγοράζει το λογισμικό, αλλά το νοικιάζει προς χρήση [14].

Οι κυριότερες εφαρμογές SaaS είναι:

- Google Apps
- Dropbox
- Salesforce
- Cisco WebEx
- Concur
- GoToMeeting

1.5.4 Διαφορές μεταξύ των μοντέλων

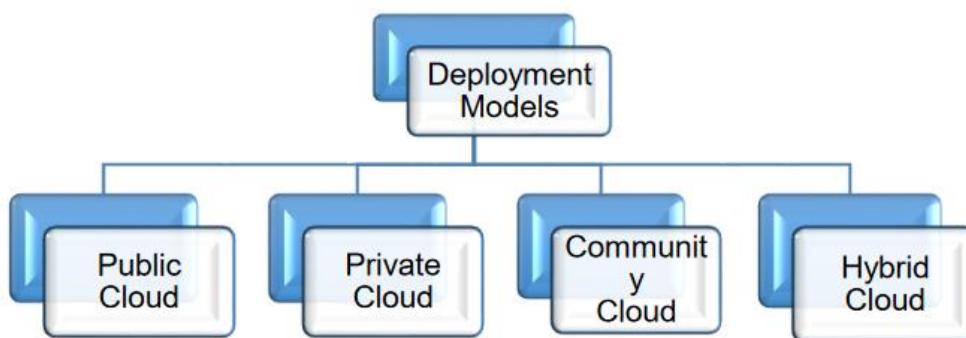
Στον παρακάτω πίνακα απεικονίζονται οι διαφορές μεταξύ των μοντέλων, ως προς την διαχείριση των αναγραφόμενων λειτουργιών. Με μπλε χρώμα απεικονίζονται οι λειτουργίες οι οποίες εκτελούνται ή διαχειρίζονται από τον τελικό χρήστη. Στην πρώτη περίπτωση δεν ακολουθείται κάποια από τα αναφερόμενα μοντέλα (SPI), αλλά όλες οι διεργασίες υλοποιούνται στον τελικό χρήστη [15].



Εικόνα 2. Διαφορές μοντέλων παροχής υπολογιστικών νεφών

1.6 Μοντέλα ανάπτυξης υπολογιστικών νεφών

Υπάρχουν τέσσερα μοντέλα ανάπτυξης υπολογιστικών νεφών [16][17].



Εικόνα 3. Μοντέλα ανάπτυξης υπολογιστικών νεφών

1.6.1 Public Cloud (Δημόσια νέφη)

Αναφέρεται στα νέφη τα οποία έχουν αναπτυχθεί και είναι προσπελάσιμα για ελεύθερη χρήση. Τα δημόσια νέφη είναι ευρέως γνωστά και χρησιμοποιούνται από έναν μεγάλο αριθμό χρηστών, οι οποίοι διαφέρουν μεταξύ τους τόσο ως προς το αντικείμενο που επιτελούν όσο και ως προς τις ανάγκες προς κάλυψη. Ο πάροχος νέφους είναι υπεύθυνος για την δημιουργία και συνεχή συντήρηση του νέφους και των υποδομών του, είτε αυτό αφορά υλικό είτε λογισμικό.

Ως πλεονεκτήματα από την χρήση δημόσιων νεφών μπορούν να θεωρηθούν τα κάτωθι:

- Χαμηλό κόστος
- Αυξημένη αποδοτικότητα

Αντίστοιχα τα μειονεκτήματα από την χρήση δημόσιων νεφών:

- Μειωμένος έλεγχος των υπηρεσιών από τους χρήστες.
- Μη σχεδίαση προσαρμοσμένων υπηρεσιών στις ανάγκες κάθε χρήστη (non-dedicated solutions)

Οι κυριότεροι προμηθευτές δημόσιων νεφών είναι:

- Microsoft (Microsoft Windows Azure)
- Amazon (Amazon EC2)
- Salesforce
- Google (Google App engine)
- IBM (IBM start cloud)

1.6.2 Private Cloud (Ιδιωτικά νέφη)

Πρόκειται για υπηρεσίες Cloud, οι οποίες είναι σχεδιασμένες ανάλογα με τις απαιτήσεις μεμονωμένων πελατών, όπως μια επιχείρηση ή ένας οργανισμός. Ο αριθμός των χρηστών είναι μικρότερος, κατά αντιστοιχία με τα δημόσια νέφη. Συνηθίζεται σε ορισμένες περιπτώσεις ένα ιδιωτικό νέφος να αναπτύσσεται εντός του data center των επιχειρήσεων. Ιδιαίτερως επιχειρήσεις ή οργανισμοί, οι οποίοι διαχειρίζονται ευαίσθητα δεδομένα, επιλέγουν την συγκεκριμένη τακτική καθώς έχουν την δυνατότητα να θωρακίσουν επιπλέον την ασφάλεια με χρήση δικών τους μέσων (firewalls).

Παρακάτω συνοψίζονται τα πλεονεκτήματα από την χρήση ιδιωτικών νεφών:

- Μεγαλύτερος βαθμός ασφάλειας.
- Μεγαλύτερη επιδραστικότητα και αμεσότητα στην λειτουργία και ανάπτυξη των υπηρεσιών του ιδιωτικού νέφους.
- Σχεδιασμός βασισμένος στις ανάγκες κάθε πελάτη.

Αντίστοιχα τα μειονεκτήματα από την χρήση δημόσιων νεφών:

- Υψηλά κόστη λειτουργίας και συντήρησης.
- Πεπερασμένος όγκος IT πόρων.

1.6.3 Community Cloud (Κοινοτικά νέφη)

Ως κοινοτικό ορίζεται το νέφος, το οποίο είναι προσπελάσιμο από ένα συγκεκριμένο σύνολο χρηστών ή οργανισμών. Μια βασική διαφορά των κοινοτικών σε σχέση με τα δημόσια, είναι η κοινή σχέση των χρηστών μεταξύ τους. Επίσης στα κοινοτικά νέφη, την ευθύνη για την ανάπτυξη του νέφους την έχουν τα μέλη της κοινότητας που το αποτελούν.

Συνηθίζεται κατά κύριο λόγο η χρήση κοινοτικών νεφών από κυβερνητικούς οργανισμούς ή φορείς δημοσίου, για την αποτελεσματικότερη διασύνδεση τους.

Οι κυριότερες εφαρμογές κοινοτικών νεφών είναι:

- Google Apps for Government
- Microsoft Government Community Cloud

1.6.4 Hybrid Cloud (Υβριδικά νέφη)

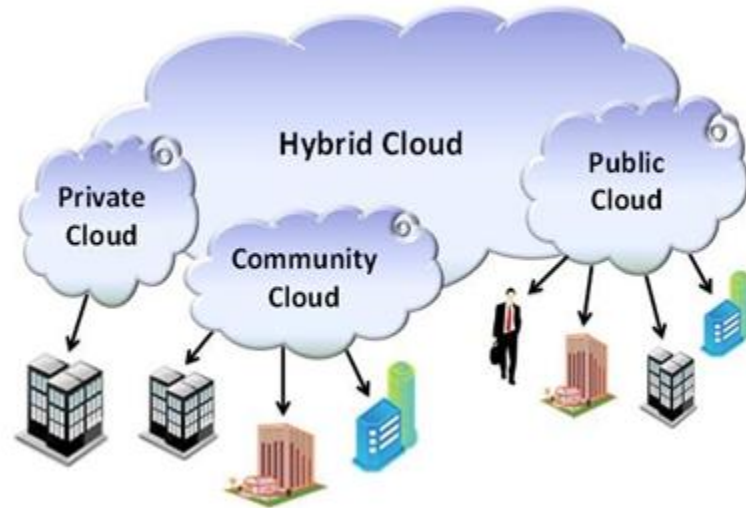
Τα υβριδικά νέφη αποτελούν το συνδυασμό τουλάχιστον δύο μοντέλων ανάπτυξης cloud. Ο συνηθέστερος συνδυασμός είναι μεταξύ δημόσιων και ιδιωτικών νεφών. Ορισμένες επιχειρήσεις προχωρούν σε κατάτμηση των εργασιών τους ανάλογα με τη φύση τους. Έτσι περιπτώσεις προσωπικών δεδομένων, όπως η λειτουργία ενός συστήματος ERP, υλοποιείται σε ένα ιδιωτικό νέφος και οι υπόλοιπες λειτουργίες υλοποιούνται σε ένα δημόσιο νέφος.

Η λογική του υβριδικού νέφους μπορεί να χρησιμοποιηθεί και στην περίπτωση που οι IT πόροι ενός ιδιωτικού δικτύου μια επιχείρησης εξαντληθούν, η πλεονάζουσα από αυτό το υπάρχον ιδιωτικό νέφος, να καλυφθεί μέσω ενός δημόσιου νέφους. Αυτό έγκειται στην διαφορά υλοποίησης των δημοσίων και ιδιωτικών νεφών, προσδίδει μεγαλύτερη ευελιξία και μειώνει ένα σαφές μειονέκτημα των ιδιωτικών νεφών.

Οι κυριότερες εφαρμογές υβριδικών νεφών είναι:

- Windows Azure
- VMware vCloud

Με την παρακάτω εικόνα αποσαφηνίζεται τόσο η σχέση μεταξύ των νεφών, όσο και η χρήση της κάθε κατηγορίας νέφους από τους χρήστες.



Εικόνα 4. Διαφορές υπολογιστικών νεφών

Κεφάλαιο 2^ο Κινητή υπολογιστική στα άκρα του δικτύου (Mobile Edge Computing)

2.1 Ορισμός κινητού υπολογιστικού νέφους (Mobile Cloud Computing)

Τα τελευταία χρόνια, λόγω της ραγδαίας ανάπτυξης και υιοθέτησης των έξυπνων κινητών τερματικών συσκευών (smartphones), παρατηρείται μεγάλη ζήτηση για υπηρεσίες νέφους σε αυτές. Το κινητό υπολογιστικό νέφος δίνει την δυνατότητα στους χρήστες να έχουν πρόσβαση στους πόρους ενός υπολογιστικού νέφους. Με αυτό τον τρόπο απαλείφεται η σχέση εξάρτησης με τους περιορισμένους πόρους των τερματικών συσκευών, αυξάνοντας τις ικανότητες των κινητών συσκευών, μέσω της αδιάλειπτης πρόσβασης στους υπολογιστικούς πόρους και χώρους αποθήκευσης του υπολογιστικού νέφους [18],[19].

Ουσιαστικά αναφέρεται σε μια υποδομή, στην οποία τόσο η επεξεργασία δεδομένων (data processing) όσο και η αποθήκευση δεδομένων (data storage) πραγματοποιείται εκτός της κινητής τερματικής συσκευής [20]. Στόχος λοιπόν του κινητού υπολογιστικού νέφους, είναι η αδιάλειπτη πρόσβαση των χρηστών στο νέφος, ανεξάρτητα από την τοποθεσία τους. Για την επίτευξη αυτού του στόχου, συνδυάζονται οι τεχνολογίες της υπολογιστικής νέφους (cloud computing) και της κινητής υπολογιστικής (mobile computing). Αυτό επιτυγχάνεται μέσω δυναμικής ρύθμισης και συσχετισμού των διαθέσιμων υπολογιστικών πόρων και αλληλεπίδρασης των κινητών συσκευών.

Καθίσταται προφανές ότι υπάρχουν παράγοντες οι οποίοι κρίνουν την επιτυχία της συγκεκριμένης αρχιτεκτονικής [21],[22]. Αρχικά ο σχεδιασμός τερματικών συσκευών, οι οποίες να στοχεύουν στην όσο το δυνατόν χαμηλότερη κατανάλωση ισχύος. Ένας τρόπος επίτευξης είναι η διενέργεια των εργασιών στο Cloud, η οποία επιτυγχάνει χαμηλότερη κατανάλωση ενέργειας και δίνει μεγαλύτερες δυνατότητες από πλευράς υπολογιστικής ισχύος και όγκου αποθήκευσης.

Κρίσιμοι παράγοντες πρέπει επίσης να θεωρηθούν η καθυστέρηση του δικτύου (latency), το διαθέσιμο εύρος ζώνης (bandwidth) καθώς και η διαθεσιμότητα του δικτύου. Συνεπώς ένα πολύ μεγάλο σύνολο, ευρέως διαδεδομένων εφαρμογών, είναι άρρηκτα συνδεδεμένες με τους παραπάνω παράγοντες και ορισμένες από αυτές απαιτούν το συνδυασμό περισσότερων από έναν, για την λειτουργία τους. Εφαρμογές εικονικής πραγματικότητας (virtual reality), εφαρμογές επαυξημένης πραγματικότητας (augmented reality), υπηρεσίες streaming, απαιτούν πέραν από πολύ μικρές τιμές σε latency, πολύ μεγάλη επεξεργαστική ισχύ.

Στον παρακάτω πίνακα απεικονίζεται η σχέση εφαρμογών με τις τιμές για τα κάτωθι στοιχεία δικτύου.

Πίνακας 1. Απαιτήσεις εφαρμογών σε υπολογιστικούς πόρους

Εφαρμογές	Επεξεργαστική ισχύς	Εύρος Bandwidth	Latency
Social networks	Χαμηλή	Μεσαίο	Μεσαία
Augmented reality	Υψηλή	Μεσαίο	Χαμηλή
Web services	Χαμηλή	χαμηλό	Υψηλή
Face recognition	Υψηλή	Μεσαίο	Χαμηλή
Web mail	Χαμηλή	Χαμηλή	Υψηλή
HD video streaming	Υψηλή	Υψηλή	Χαμηλή
Online gaming	Υψηλή	Μεσαίο	χαμηλή

Για την όσο το δυνατόν καλύτερη κάλυψη των ανωτέρω παραγόντων, έχει δημιουργηθεί η αρχιτεκτονική της κινητής υπολογιστικής στα άκρα του δικτύου (Mobile Edge Computing) [23], η οποία θα αναλυθεί παρακάτω.

2.2 Πλεονεκτήματα κινητού υπολογιστικού νέφους

Συνοπτικά, παρουσιάζονται τα κύρια πλεονεκτήματα από την χρήση κινητών υπολογιστικών νεφών [18],[22].

Ανθεκτικότητα: Εξαλείφεται ο κίνδυνος απώλειας δεδομένων σε περίπτωση βλάβης της τερματικής συσκευής. Υπάρχει αποθήκευση των δεδομένων σε απομακρυσμένους χώρους αποθήκευσης, τα οποία μπορεί να προσπελαστούν από οποιαδήποτε συσκευή.

Μειωμένο κόστος συσκευών: Δεδομένου ότι το σύνολο των εφαρμογών εκτελείται απομακρυσμένα, χρησιμοποιώντας τους υπολογιστικούς πόρους του νέφους, δεν απαιτούνται συσκευές υψηλών προδιαγραφών, από άποψη υπολογιστικής ισχύος. Στόχευση των συσκευών είναι η αδιάλειπτη σύνδεση με το υπολογιστικό νέφος καθώς και η χαμηλή κατανάλωση ισχύος.

Βελτίωση των παρεχόμενων δυνατοτήτων των συσκευών: Οι τερματικές συσκευές έχουν την δυνατότητα για πρόσβαση σε ένα εύρος υπολογιστικών πόρων, σαφώς μεγαλύτερο από τους πεπερασμένους πόρους που διαθέτει μια συσκευή. Με αυτό τον τρόπο αυξάνεται η δυνατότητα τόσο σε υπολογιστική ισχύ όσο και σε αποθήκευση.

Κλιμακούμενη χρήση υπολογιστικών πόρων : Υπάρχουν οφέλη τόσο για την συσκευή, στην οποία αυξάνεται ο κύκλος ζωής της μπαταρίας, όσο και περιβαλλοντικά με την βέλτιστη κατανομή των υπολογιστικών πόρων ανάλογα με τις προς κάλυψη ανάγκες. Με αυτόν τον τρόπο εξασφαλίζεται ότι δεν συντελείται σπατάλη των υπολογιστικών πόρων.

Επιτρέπει την κινητικότητα: Παρέχεται η δυνατότητα στους χρήστες να έχουν πρόσβαση σε πληθώρα υπολογιστικών πόρων, ανεξάρτητα από την γεωγραφική τους τοποθεσία.

2.3 Cloudlet

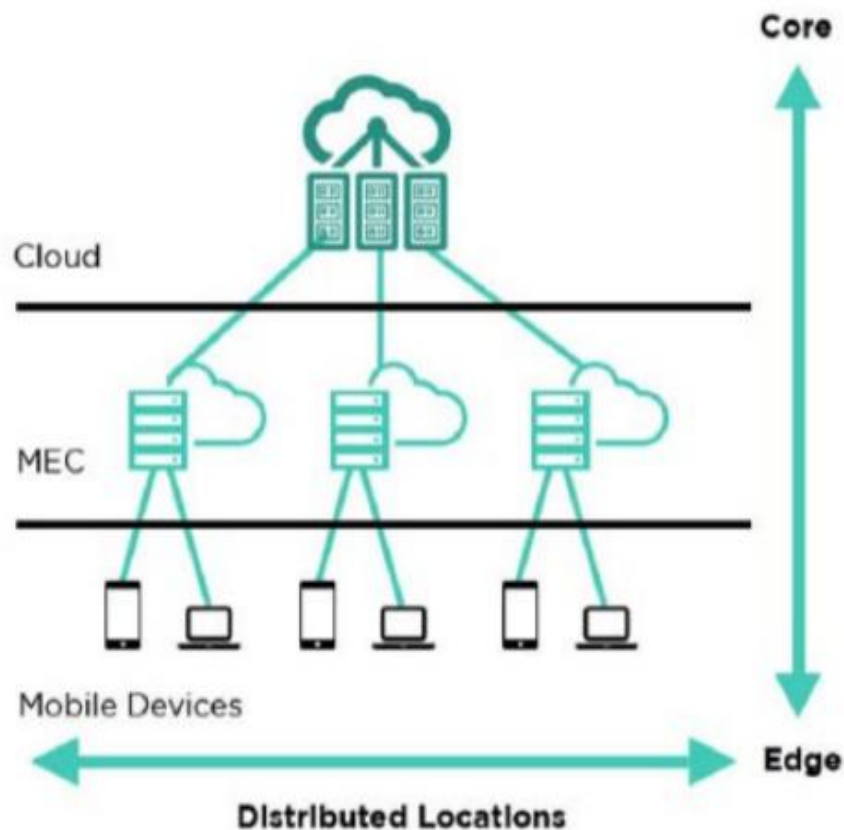
2.3.1 Ορισμός του Cloudlet

Ως cloudlet ορίζεται ένα μικρής κλίμακας data center, το οποίο παρέχει πρόσβαση σε υπολογιστικούς πόρους και είναι τοποθετημένο στις άκρες ενός δικτύου [24]. Βασικός στόχος ενός cloudlet είναι να παρέχει τις υπηρεσίες ενός μεγαλύτερου σε μέγεθος Cloud, στο οποίο διασυνδέεται, όσο το δυνατόν πιο κοντά στους τελικούς χρήστες στα άκρα του δικτύου [25],[26].

Οι τερματικές συσκευές μέσω της αλληλεπίδρασης με το πιο κοντινό cloudlet, μετακυλίνουν σε αυτό τις διεργασίες οι οποίες θα πρέπει να εκτελεστούν. Ανάλογα με την περίπτωση οι συγκεκριμένες διαδικασίες μπορούν είτε να εκτελεστούν στο cloudlet είτε να υπάρξει μεταβίβαση αυτών στο Cloud [20].

Θα μπορούσε να γίνει παρομοίωση ενός cloudlet με ένα access point, όσον αφορά τον τρόπο διασύνδεσης με ένα μεγαλύτερο δίκτυο και ως προς την μεταφορά των δεδομένων από και προς αυτό το δίκτυο. Χωροταξικά ένα cloudlet είναι εφικτό να τοποθετηθεί είτε σε κάποια σταθερή τοποθεσία, όπως ένα στάδιο ή σε μια κινητή συσκευή, όπως ένα μέσο μεταφοράς.

Το cloudlet ιδεατά μπορεί να θεωρηθεί στο ενδιάμεσο μιας ιεραρχίας, η οποία περιλαμβάνει την τερματική συσκευή, το cloudlet (MEC server) και το ευρύτερο cloud.



Εικόνα 5. Ιεραρχική θέση cloudlet

2.3.2 Βασικά χαρακτηριστικά ενός Cloudlet

Προκειμένου να μπορεί να ανταπεξέλθει στον ρόλο του, κάθε cloudlet θα πρέπει να τηρεί τις προδιαγραφές, όσον αφορά τα παρακάτω χαρακτηριστικά [20].

Επάρκεια σε υπολογιστικούς πόρους: Ένα cloudlet θα πρέπει να έχει τους απαραίτητους υπολογιστικούς πόρους τόσο από άποψη υπολογιστικής ισχύος όσο και άποψη αποθηκευτικού χώρου, προκειμένου να είναι σε θέση να καλύψει τις ανάγκες των χρηστών, οι οποίοι αλληλεπιδρούν με αυτό. Επίσης θα πρέπει να εξασφαλίζεται η κάλυψη των ενεργειακών αναγκών και να υπάρχει επάρκεια σε επίπεδο τροφοδοσίας.

Εγγύτητα στους τελικούς χρήστες (proximity): Θα πρέπει να είναι όσο το δυνατόν πιο κοντά τοποθετημένα στους τελικούς χρήστες, προκειμένου να εξασφαλίζεται με τον πιο αποτελεσματικό τρόπο η βέλτιστη απόκριση του τελικού χρήστη είτε με το ίδιο το cloudlet είτε με το cloud, από άποψη καθυστέρησης δικτύου (low end to end latency).

Γρήγορη και αδιάλειπτη διασύνδεση με το Cloud: Θα πρέπει να εξασφαλίζεται η βέλτιστη διασύνδεση με το Cloud, προκειμένου να παρέχεται υπηρεσία με την μικρότερη καθυστέρηση και το υψηλότερο bandwidth.

Αποτελεί παράγοντα soft state: Η λειτουργία ενός cloudlet θεωρείται αυξημένης αποδοτικότητας για την λειτουργία ενός δικτύου, παρόλα αυτά δεν είναι κρίσιμη για την λειτουργία του δικτύου, καθώς πιθανή δυσλειτουργία του cloudlet δεν επιφέρει κατάρρευση του δικτύου.

2.3.3 Διαφορές μεταξύ Cloud και Cloudlet

Οι βασικές διαφορές μεταξύ ενός Cloud και ενός cloudlet, έγκεινται στους παρακάτω παραμέτρους, οι οποίοι σχετίζονται με την λειτουργία τους [24].

Διαχείριση: Σε ένα cloudlet η υποστήριξη μπορεί να γίνει τοπικά από την υπηρεσία που το διαχειρίζεται, εν αντιθέσει με ένα cloud το οποίο απαιτεί εξειδικευμένη παρακολούθηση από τον πάροχο της υπηρεσίας.

Μέγεθος: Ένα cloudlet μπορεί να είναι εγκατεστημένο στις εγκαταστάσεις ενός τοπικού σημείου, εν αντιθέσει με το cloud, το οποίο έχει μεγαλύτερες απαιτήσεις και κόστη από άποψη εγκατάστασης και συντήρησης.

Κρισιμότητα: Ένα cloudlet μπορεί να θεωρηθεί soft state, εν συγκρίσει με το cloud το οποίο θεωρείται hard state, δεδομένης της επίδρασής του στην συνολική λειτουργία του δικτύου.

Αριθμός χρηστών: Ο αριθμός των τελικών χρηστών ενός cloudlet είναι πολύ μικρότερος από τους χρήστες που εξυπηρετούνται από το cloud.

Διασύνδεση: Το cloudlet είναι βασισμένο στην δικτύωση τοπικού δικτύου (LAN), σε σύγκριση με το Cloud το οποίο βασίζεται στο internet.

2.4 Κινητή υπολογιστική στα άκρα του δικτύου (Mobile Edge Computing)

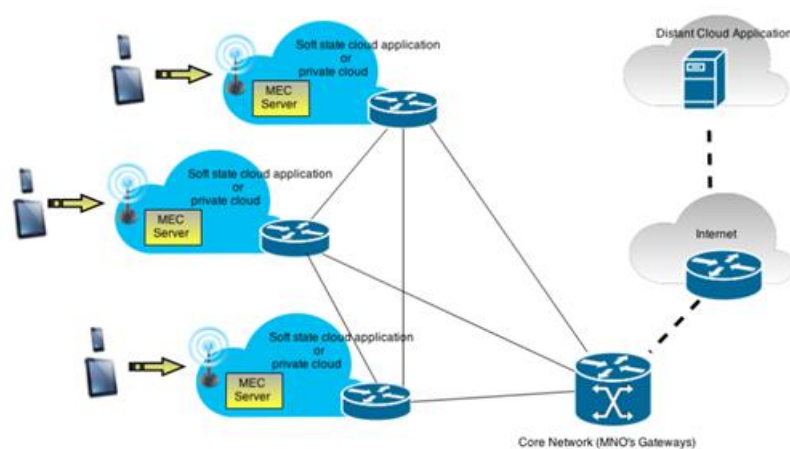
Η κινητή υπολογιστική στα άκρα του δικτύου αποτελεί μια αρχιτεκτονική, η οποία έχει ως κύριο στόχο την παροχή πόρων και υπηρεσιών υπολογιστικού νέφους στα άκρα (edges) ενός κυψελωτού (cellular) δικτύου [28]. Επισημαίνεται πως αποτελεί συνέχεια του Mobile Cloud Computing, όπου με την χρήση των cloudlets [30] ως πρώτο κόμβο (first hop node) για τους τελικούς χρήστες, μειώνει πρακτικά την απόσταση τους από το δίκτυο (last mile), οδηγώντας στην αποφόρτιση του δικτύου (offloading) και στην βελτίωση της ποιότητας της παρεχόμενης υπηρεσίας [25],[27].

Δεδομένης της δομής σχεδίασης των κυψελωτών δικτύων και της θέσης των σταθμών βάσης (RAN Base Stations), έχουν πραγματοποιηθεί αρκετές προσπάθειες από τους παρόχους, για την χρησιμοποίησή τους ως τον πρώτο κόμβο (first hop) για τους χρήστες κινητής στα άκρα του δικτύου. Αξίζει να αναφερθεί πως εταιρείες όπως

η Nokia έχει προβεί στην δημιουργία έξυπνων σταθμών βάσης, οι οποίοι μπορούν να χρησιμοποιηθούν ως πλατφόρμες υπολογιστικού νέφους στην άκρη του δικτύου (Radio Application Cloud Server) [20].

Επίσης το Ευρωπαϊκό Ινστιτούτο Τηλεπικοινωνιακών Προτύπων (ETSI-European Telecommunications Standards Institute), σε συνεργασία με τις Huawei, IBM, Intel, Nokia Networks, NTT DOCOMO και Vodafone, έχουν έρθει σε συμφωνία για την δημιουργία μιας ανοιχτής προτυποποιημένης πλατφόρμας ανάπτυξης λογισμικού, η οποία θα παρέχει υπηρεσίες Cloud πιο κοντά στους τελικούς χρήστες.

Απώτερος σκοπός είναι να μετατραπούν οι σταθμοί βάσης σε MEC servers, και να λειτουργήσει το mobile edge computing σαν ένα σύστημα, στο οποίο θα μπορούν να τρέχουν διάφορες εφαρμογές στις άκρες του δικτύου, κάνοντας χρήση μιας προτυποποιημένης πλατφόρμας, η οποία θα μπορεί με την σειρά της να προσφέρει υπολογιστικούς πόρους (υπολογιστική ισχύ, χωρητικότητα) στην εφαρμογή. Το περιβάλλον αυτό χαρακτηρίζεται από εξαιρετικά χαμηλή καθυστέρηση και υψηλό εύρος ζώνης, καθώς και από πρόσβαση που εκτελείται σε πραγματικό χρόνο σε ασύρματες πληροφορίες που μπορούν να αξιοποιηθούν από τις εφαρμογές.



Εικόνα 6. Αρχιτεκτονική του Mobile Edge Computing

2.5 Πλεονεκτήματα Mobile Edge Computing

Τα πλεονεκτήματα τα οποία απορρέουν από την χρήση του Mobile Edge Computing [29], αναλύονται ως εξής:

Αποφόρτιση του δικτύου (traffic offloading): Με την χρήση των cloudlets μεγάλος όγκος του φορτίου της κίνησης το οποίο θα προορίζονταν για τον πυρήνα

του δικτύου (core network) έχει εξοικονομηθεί, οδηγώντας έτσι σε αποτελεσματικότερη διαχείριση των πόρων του δικτύου [30].

Καλύτερη ποιότητα υπηρεσίας στις άκρες: Βελτιώνεται αισθητά ο χρόνος απόκρισης των εφαρμογών, μέσω μείωσης της καθυστέρησης στην επικοινωνία του δικτύου και της πρόσβασης στους υπολογιστικούς πόρους που παρέχει ένα υπολογιστικό νέφος, αυξάνοντας παράλληλα την ικανοποίηση των χρηστών.

Δυνατότητα μεταγωγής: Παρέχεται η δυνατότητα για άμεση μεταγωγή των λειτουργιών σε περίπτωση βλάβης, είτε σε κάποια εικονική μηχανή (virtual machine) του υπάρχοντος cloudlet, με μεταγωγή σε άλλο διαθέσιμο VM, είτε σε περίπτωση απώλειας επικοινωνίας με το cloudlet σε άλλο στοιχείο του δικτύου, χωρίς να τίθεται εν αμφιβόλω η αδιάλειπτη λειτουργία του δικτύου.

Αυξημένη ασφάλεια: Δεδομένης της δυνατότητας για λειτουργία πληθώρας εφαρμογών, δίνεται η δυνατότητα για εφαρμογή περαιτέρω μέτρων ασφαλείας, όπως firewalls, σε πολλαπλά σημεία του δικτύου, ακόμα και στην περίπτωση που δεν απαιτηθεί η δρομολόγηση και εξυπηρέτηση στο Cloud.

2.6 Προκλήσεις σχετικά με το Mobile Edge Computing

Πέρα από τις αδιαμφισβήτητα αρκετές και καινοτόμες δυνατότητες που παρέχει η κινητή υπολογιστική στα άκρα του δικτύου, προκύπτουν αρκετές παράμετροι οι οποίες θα πρέπει να ληφθούν υπόψη κατά την εφαρμογή της συγκεκριμένης τεχνολογίας, προκειμένου να αντιμετωπισθούν με επιτυχία οι ολοένα αυξανόμενες ανάγκες, από την πρόοδο της τεχνολογίας των επικοινωνιών [29]. Συστήματα όπως το 5G αναμένεται να κάνουν εκτενή χρήση της συγκεκριμένης τεχνολογίας.

Δεδομένου ότι η δομή ενός δικτύου είναι κατά ένα μεγάλο βαθμό σταθερή, προκύπτει ότι και ο αριθμός των κόμβων στα άκρα είναι πεπερασμένος, εν συγκρίσει με τους υπολογιστικούς πόρους. Βάσει αυτού, θα χρειαστεί να υιοθετηθούν ορισμένες πρακτικές κατά την υλοποίηση του.

Είναι ξεκάθαρο πως δεν μπορεί να υπάρχει για κάθε τελικό χρήστη στις άκρες ένα συγκεκριμένο cloudlet, από το οποίο θα εξυπηρετείται. Συνεπώς, κρίνεται απαραίτητη η υλοποίηση κανόνων σχετικά με την βέλτιστη κατανομή των χρηστών στα αντίστοιχα cloudlets, δεδομένων χωροταξικών κριτηρίων καθώς και του συνολικού φόρτου είτε του εκάστοτε cloudlet είτε όλου του δικτύου. Θα πρέπει να δημιουργηθούν οι μηχανισμοί εκείνοι, οι οποίοι θα μπορούν να αποφασίσουν άμεσα, για το ποιοι χρήστες θα εξυπηρετηθούν σε ποιο cloudlet καθώς και πότε είναι απαραίτητο να γίνει δρομολόγηση τους στο Cloud.

Επιπροσθέτως, στην λογική της βέλτιστης επιλογής με το μικρότερο κόστος, θα χρειαστεί να δημιουργηθούν μηχανισμοί, οι οποίοι θα μπορούν να προβλέψουν εγκαίρως, πόσα από τα διαθέσιμα cloudlets του δικτύου θα πρέπει να λειτουργούν και σε ποιο βαθμό, προκειμένου να εξυπηρετήσουν τις μεταβαλλόμενες ανάγκες που υπάρχουν στο δίκτυο. Καθίσταται σαφές ότι η επιλογή να λειτουργούν όλα τα διαθέσιμα cloudlets σε ένα δίκτυο, δεν συνάδει με την βέλτιστη διαχείριση των πόρων και δεν συντελεί στην μέγιστη ενεργειακή απόδοση, την οποία επιθυμούν οι πάροχοι είτε του δικτύου είτε των υπηρεσιών Cloud.

Κρίσιμη επίσης θεωρείται η συμβατότητα με διαφόρους τύπους ετερογενών δικτύων (HetNets), όπως τα συστήματα κινητής επικοινωνίας (2G/3G/4G/5G/femto) καθώς και συστήματα σταθερής, όπως το wifi.

Στα πλαίσια της υπάρχουσα διπλωματικής, εξετάζονται αλγόριθμοι [32] για την δυναμική κατανομή των χρηστών στα διαθέσιμα cloudlet ενός δικτύου, οι οποίοι καλούνται να δώσουν λύση σχετικά με το ποια cloudlets πρέπει να επιλεγούν καθώς και σε ποια από αυτά θα κατανεμηθούν οι χρήστες, με γνώμονα τόσο την χαμηλότερη καθυστέρηση του δικτύου, άρα και την βέλτιστη ικανοποίηση των τελικών χρηστών, όσο και την μικρότερη δυνατή διακύμανση των χρηστών στα επιλεγμένα cloudlets, με άλλα λόγια την εξισορρόπηση του φορτίου των χρηστών, άρα και την βέλτιστη ικανοποίηση των παρόχων, από πλευράς κατανομής πόρων.

Κεφάλαιο 3^ο Τοπολογίες δικτύου και αλγόριθμοι

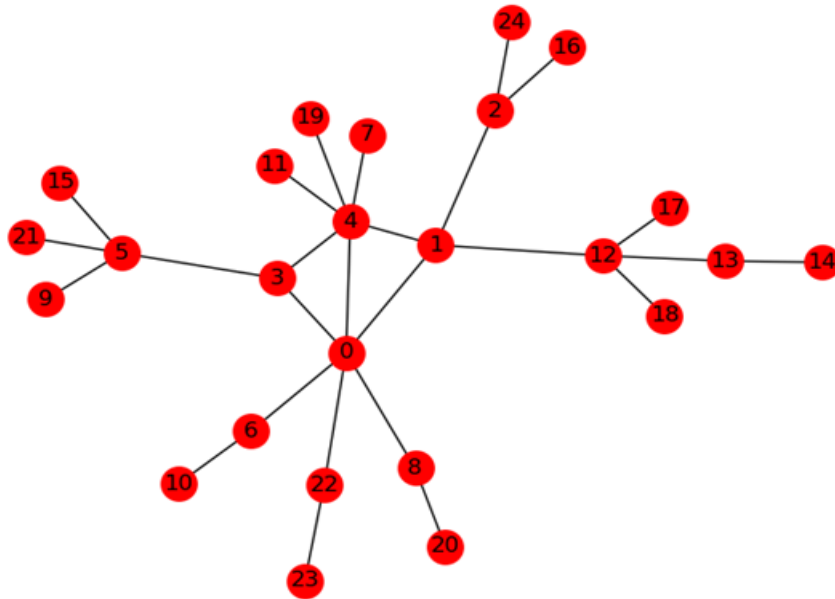
3.1 Τοπολογίες δικτύου

Στα πλαίσια της παρούσας εργασίας, πραγματοποιήθηκε προσαρμογή και μελέτη της απόδοσης δυο συγκεκριμένων ευριστικών αλγορίθμων σε δυο τοπολογίες δικτύου που αναπτύχθηκαν. Η υλοποίηση των αλγορίθμων έγινε σε γλώσσα προγραμματισμού python και οι αλγόριθμοι μπορούν να χρησιμοποιηθούν για την μελέτη οποιασδήποτε τοπολογίας δικτύου, είτε λαμβάνοντας ως παραμέτρους τις διαστάσεις του εν λόγω δικτύου, όσον αφορά τον αριθμό κόμβων και διασυνδέσεων του, είτε με την χρησιμοποίηση κάποιας γνωστής τοπολογίας από τις βιβλιοθήκες της python. Ο σχεδιασμός των τοπολογιών που εξετάστηκαν πραγματοποιήθηκε μέσω της βιβλιοθήκης NetworkX και συγκεκριμένα των γραφημάτων `networkx.generators.random_graphs.barabasi_albert_graph` [33] και `triangular_lattice_graph`.

Και στις δυο τοπολογίες ο αριθμός των κόμβων (nodes) του δικτύου είναι σταθερός και ίσος με εικοσιπέντε, αν και το πλήθος των κόμβων είναι μια παράμετρος που μπορεί να πάρει διαφορετικές τιμές. Επίσης, οι θέσεις των κόμβων στο δίκτυο, αλλά και οι διασυνδέσεις τους παρέμειναν σταθερές, καθ' όλη την διάρκεια της μελέτης.

3.1.1 Τοπολογία scale free

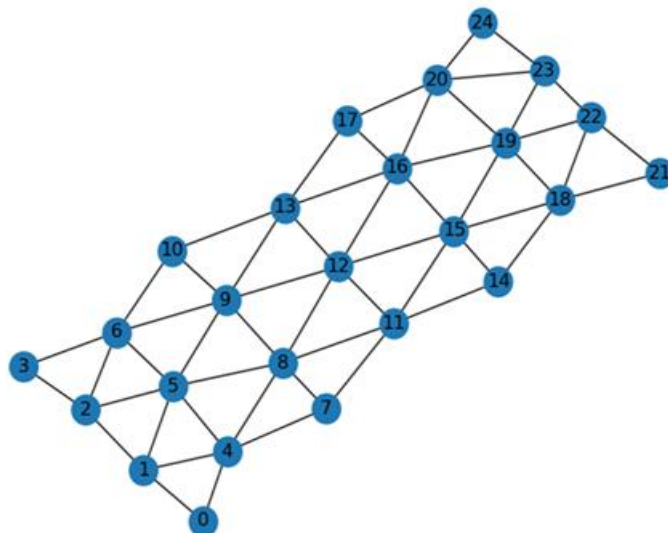
Με την συγκεκριμένη τοπολογία επιχειρείται να γίνει μελέτη ενός τυχαίου δικτύου (random) και συγκεκριμένα ενός scale free network. Η βασική παράμετρος ενός scale free network είναι ότι η δομή του δεν εξαρτάται από το μέγεθος. Συνεπώς παρά την αύξηση των κόμβων στις άκρες, η κύρια δομή του παραμένει σταθερή [34].



Εικόνα 7. Τοπολογία scale free

3.1.2 Τοπολογία lattice

Με την συγκεκριμένη τοπολογία επιχειρείται να γίνει μελέτη ενός πιο στατικού μοντέλου δικτύου, στο οποίο όλοι οι κόμβοι να διασυνδέονται απευθείας με τουλάχιστον δυο γειτονικούς κόμβους, το οποίο όμως να διαφέρει από την τοπολογία grid ($N \times N$) διαστάσεων.

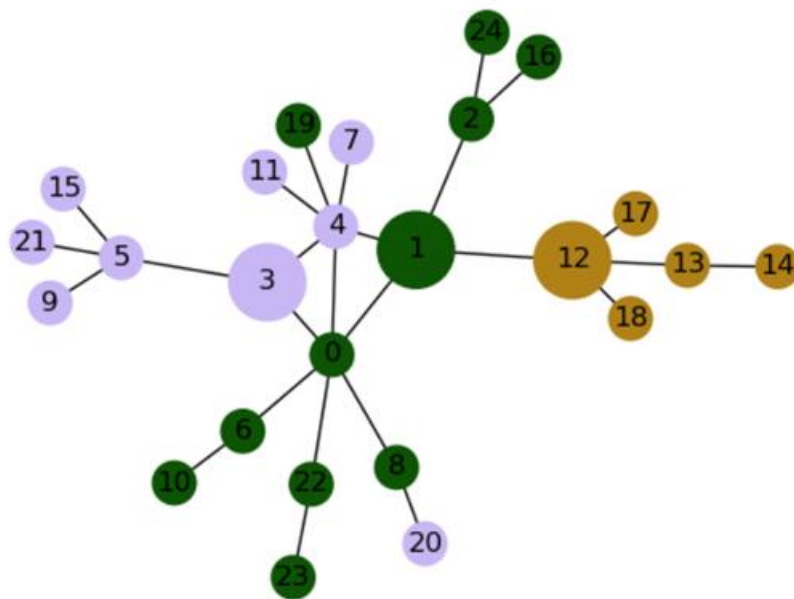


Εικόνα 8. Τοπολογία lattice

3.2 Ανάλυση των τοπολογιών που σχεδιάστηκαν

Ο κάθε κόμβος στα γραφήματα, αναπαριστά έναν access node για το δίκτυο που προσφέρει προσβασιμότητα στους χρήστες του δικτύου και ο οποίος ουσιαστικά μπορεί να λειτουργήσει και ως ένα cloudlet. Ο αριθμός πάνω σε κάθε κόμβο υποδηλώνει τον αύξοντα αριθμό του στο δίκτυο.

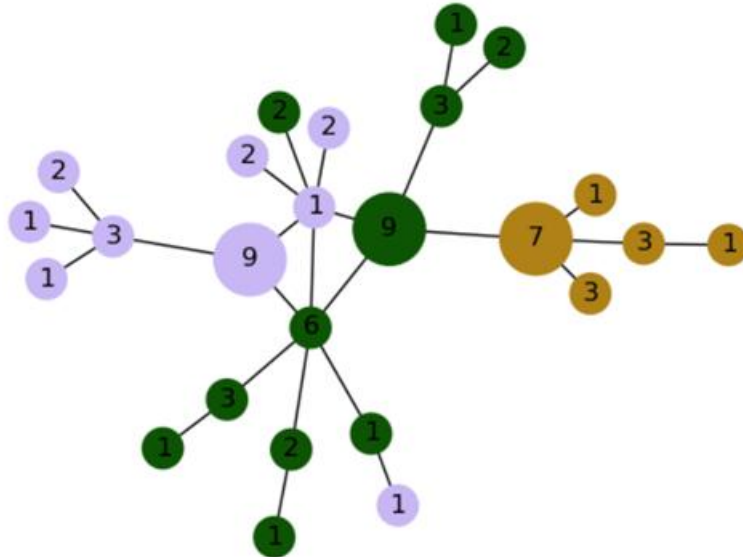
Στην παρακάτω εικόνα οι κόμβοι με μεγαλύτερο μέγεθος (1,3,12) είναι τρεις access nodes που έχουν επιλεγεί να λειτουργούν και ως service nodes ή cloudlets, για την εξυπηρέτηση των απαιτήσεων σε υπολογιστική ισχύ των χρηστών του δικτύου. Οι υπόλοιποι access nodes λειτουργούν ως κόμβοι πρόσβασης μόνο και ανάλογα με τον service node από τον οποίο οι χρήστες κάθε access node λαμβάνουν υπηρεσίες υπολογιστικής, έχουν αντίστοιχα και το χρώμα του service node. Με άλλα λόγια, το χρώμα κάθε access node συμβολίζει σε ποιον service node έχουν ανατεθεί οι χρήστες του για να λαμβάνουν υπηρεσίες κινητής υπολογιστικής.



Εικόνα 8. Τοπολογία scale free με τρεις service nodes

Σε αντίθεση με τα παραπάνω σχήματα, οι αριθμοί που εμφανίζονται στο παρακάτω σχήμα, απεικονίζουν το βάρος των κόμβων και το οποίο είναι ανάλογο του πλήθους των χρηστών που λαμβάνουν πρόσβαση μέσω του κάθε κόμβου πρόσβασης. Ο όρος βάρος αναφέρεται λοιπόν στον αριθμό των χρηστών που ανατίθενται σε κάθε access node.

Στην συγκεκριμένη περίπτωση φαίνεται πως δίνεται μεγαλύτερο βάρος, και συνεπώς μεγαλύτερο πλήθος χρηστών, στους κεντρικούς κόμβους της συγκεκριμένης τοπολογίας (0,1,3,12). Για την τοπολογία lattice κεντρικοί κόμβοι θεωρούνται οι κόμβοι 8,9,12,15,6.



Εικόνα 9. Κατανομή βάρους σε τοπολογία scale free

Η μελέτη βασίστηκε σε διαφορετικά σενάρια, σχετικά με την κατανομή των χρηστών στους κόμβους του δικτύου. Συγκεκριμένα υπάρχουν για κάθε τοπολογία τρία σενάρια, όπως φαίνεται στον παρακάτω πίνακα.

Πίνακας 2. Σενάρια εκτέλεσης αλγορίθμων

Τοπολογία	Κατανομή βάρους	Τιμές βαρών Κεντρικοί κόμβοι	Τιμές βαρών Περιφερειακοί κόμβοι	Αριθμός πειραμάτων
Scale free/Lattice	στατική	1	1	1
Scale free/Lattice	τυχαία	(6,9)	(1,3)	10
Scale free/Lattice	τυχαία	(1,3)	(6,9)	10

Αρχικά γίνεται υπολογισμός έχοντας ως δεδομένο ότι ο αριθμός των χρηστών σε κάθε κόμβο είναι ίδιος και ίσος με την μονάδα (uniform).

Στην συνέχεια πραγματοποιήθηκαν δέκα πειράματα για κάθε τοπολογία, στα οποία σε κάθε έναν από τους κεντρικούς κόμβους κατανέμονται τυχαία από έξι έως

εννέα χρήστες και στους περιφερειακούς από έναν έως τρεις χρήστες. Το σενάριο αντιστοιχεί σε υψηλότερη (χαμηλότερη) ζήτηση στο κέντρο (περιφέρεια) του δικτύου.

Έπειτα, πραγματοποιήθηκαν δέκα πειράματα για κάθε τοπολογία, στα οποία κατανέμονται από έναν έως τρεις χρήστες στους κεντρικούς κόμβους και από έξι έως εννέα στους περιφερειακούς. Το σενάριο αντιστοιχεί σε χαμηλότερη (υψηλότερη) ζήτηση στο κέντρο (περιφέρεια) του δικτύου.

Για κάθε ένα από τα σαράντα δύο πειράματα, έγινε σύγκριση των αποτελεσμάτων των δυο αλγορίθμων, ως προς την καθυστέρηση του δικτύου και την κατανομή των access nodes (variation), στους επιλεγθέντες service nodes, με τα αποτελέσματα ενός εξαντλητικού αλγορίθμου.

3.3 Ευριστικοί αλγόριθμοι

Ως ευριστικός θεωρείται ένας αλγόριθμος, ο οποίος έχει σχεδιαστεί προκειμένου να επιλύει ένα πρόβλημα με έναν πιο γρήγορο και αποτελεσματικό τρόπο, προσαρμόζοντας τις απαιτήσεις για ακρίβεια σε συνάρτηση με την επιδιωκόμενη ταχύτητα επίλυσης.

Ως κύριο πεδίο ανάπτυξης έχουν εφαρμογές, οι οποίες απαιτούν γρήγορες και όσο το δυνατόν πιο ακριβείς αποφάσεις, για την επίλυση σύνθετων προβλημάτων. Ιδιαίτερα σε επιστημονικά πεδία όπως της τεχνητής νοημοσύνης (AI) ή των νευρωνικών δικτύων, οι ευριστικοί αλγόριθμοι, αποτελούν δομικά στοιχεία για την λειτουργία τους και την περαιτέρω ανάπτυξη τους.

Εν αντιθέσει, οι εξαντλητικοί αλγόριθμοι υπολογίζουν όλες τις διαθέσιμες επιλογές για την επίλυση ενός προβλήματος και στην συνέχεια επιλέγουν αυτή που ικανοποιεί καλύτερα τον προς επίτευξη στόχο. Από άποψη ακρίβειας θεωρούνται αρκετά αποτελεσματικοί, έχοντας ως μειονέκτημα την ταχύτητα εκτέλεσης καθώς και την μη βέλτιστη αξιοποίηση των πόρων.

Αρχικά χρειάστηκε να αναπτυχθεί ένα εξαντλητικός αλγόριθμος, ο οποίος θα υπολογίζει όλες τις δυνατές επιλογές, τα αποτελέσματα του οποίου θα αποτελέσουν μέτρο σύγκρισης μετέπειτα για τους αλγορίθμους που αναπτύχθηκαν. Λόγω του μεγάλου όγκου των δεδομένων και της έκτασης του δικτύου, ο εξαντλητικός αλγόριθμος επιστρέφει σαν επιλογές, όλους τους συνδυασμούς, οι οποίοι θα έχουν μέγιστη απόκλιση από την μικρότερη τιμή κόστους καθυστέρησης (latency cost) 2 μονάδες. Στην συνέχεια επιλέγεται ο συνδυασμός με την καλύτερη σχέση κόστους και διακύμανσης.

Οι αλγόριθμοι που έχουν αναπτυχθεί, έχουν ως κεντρική ιδέα την εύρεση της όσο το δυνατόν καλύτερης επιλογής, εν συγκρίσει με την βέλτιστη, σύμφωνα με μια προκαθορισμένη διαδικασία [32].

Ο στόχος τους είναι διπτός, αρχικά η εξυπηρέτηση του δικτύου με την όσο το δυνατόν μικρότερη καθυστέρηση [35], κάτι το οποίο είναι βασική παράμετρος για εξυπηρέτηση των χρηστών και βάση του mobile edge computing.

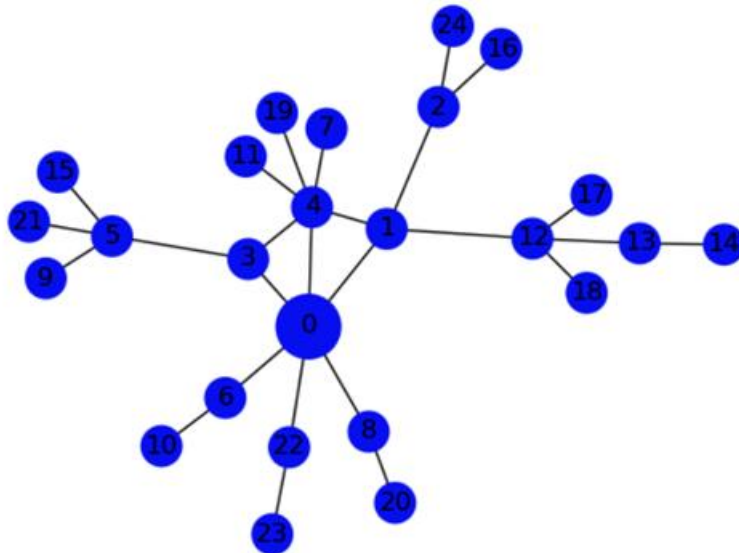
Επίσης, ιδιαίτερη βαρύτητα δίνεται στην βέλτιστη διασπορά των χρηστών στους επιλεχθέντες service nodes, προκειμένου να χρησιμοποιούνται στο βέλτιστο οι υποδομές του δικτύου, το οποίο είναι ζητούμενο από τους παρόχους. Δεδομένης της διασυνδεσιμότητας όλων των cloudlets μεταξύ τους, δεν απαιτείται η λειτουργία όλων των διαθέσιμων cloudlets, καθώς ανάλογα με το φορτίο που πρέπει να εξυπηρετηθεί, μπορεί να υποστηριχθεί με μικρότερο αριθμό cloudlets [36].

Η λειτουργία των αλγορίθμων στηρίζεται στην επιλογή του κάθε κόμβου ευριστικά, χωρίς να πραγματοποιείται υπολογισμός όλων των σεναρίων και μετέπειτα επιλογή της βέλτιστης λύσης [37],[38].

Γίνεται εύκολα αντιληπτό πως οι διεργασίες πραγματοποιούνται σε πολύ μικρότερο διάστημα, σε σχέση με την προαναφερθείσα διαδικασία ενός εξαντλητικού αλγορίθμου.

3.4 Μεθοδολογία αλγορίθμων

Για τον πρώτο ευριστικό αλγόριθμο, πραγματοποιείται υπολογισμός για όλους τους κόμβους, βάσει της συνολικής καθυστέρησης του δικτύου, για την περίπτωση που το σύστημα μπορεί να εξυπηρετηθεί με το άνοιγμα ενός μόνο κόμβου (cloudlet). Κατόπιν γίνεται ταξινόμηση των κόμβων. Στην περίπτωση που χρειάζεται μόνο ένας κόμβος, επιλέγεται αυτός με την μικρότερη τιμή. Στην πλειοψηφία των περιπτώσεων, αποτελεί τον κεντρικό κόμβο του συστήματος, ο οποίος και επιφέρει το χαμηλότερο κόστος καθυστέρησης, έχοντας την μικρότερη απόσταση από τους υπόλοιπους κόμβους.



Εικόνα 10. Παράδειγμα τοπολογία scale free με έναν service node

Στην περίπτωση που απαιτηθούν δυο service nodes για την εξυπηρέτηση των αναγκών του δικτύου, ο αλγόριθμός δεν επιλέγει τον προαναφερθέντα service node, επιλέγεται ο αμέσως επόμενος κόμβος βάσει της λίστας ταξινόμησης.

Για την επιλογή του δεύτερου service node, υπολογίζονται οι διαφορές όλων των κόμβων από τον ήδη επιλεγθέντα. Επιλέγεται ως τιμή αναφοράς (d_{max}) η απόσταση του service node από τον κόμβο, ο οποίος βρίσκεται πιο μακριά.

Πραγματοποιείται έλεγχος για όσους κόμβους απέχουν την μισή απόσταση, από τον πιο μακρινό κόμβο του service node, ίση με $d_{max}/2$. Από αυτούς επιλέγεται ως ο δεύτερος service node, αυτός με το μικρότερο κόστος.

Στην συνέχεια για τον υπολογισμό και των υπόλοιπων σεναρίων, έχοντας ως επιλογή τους δυο πρώτους service nodes, ο αλγόριθμος υπολογίζει αυτούς που απέχουν και από τους δυο απόσταση ίση με $d_{max}/2$. Κατόπιν επιλέγεται αυτός που έχει το μικρότερο κόστος αθροιστικά.

Σε περίπτωση που στην ανωτέρω διαδικασία εύρεσης κόμβων, οι οποίοι να απέχουν απόσταση μεγαλύτερη από $d_{max}/2$ από τους ήδη επιλεγθέντες service nodes, δεν βρεθεί έστω ένας κόμβος, τότε μειώνεται κατά μία μονάδα η τιμή του $d_{max}/2$, μέχρι να εντοπιστεί.

Η διαδικασία συνεχίζεται μέχρι να έχει ληφθεί η απόφαση για όλους τους κόμβους του δικτύου.

Συνοπτικά τα βήματα του αλγόριθμου:

1. Υπολογίζει το κόστος (delay) για όλους τους access nodes (AN).
2. Πραγματοποιείται ταξινόμηση των access nodes βάσει του κόστους.
3. Για έναν service node ($N=1$), επιλέγεται ο access node με το μικρότερο κόστος.
4. Για δυο service nodes ($N=2$), αρχικά επιλέγεται ως πρώτος ο αμέσως καλύτερος, από άποψη κόστους, από τον κόμβο που αναφέρεται στο βήμα 3.
5. Βρίσκει τον κόμβο με μεγαλύτερη απόσταση από τον πρώτο service node (d_{max}).
6. Βρίσκει τους κόμβους με απόσταση από τον πρώτο service node, μεγαλύτερη ή ίση (\geq) από $d_{max}/2$.
7. Επιλέγεται αυτός με την μικρότερη τιμή κόστους ως δεύτερος service node.
8. Μέχρι να υπολογισθούν και οι υπόλοιποι service nodes ($N \leq AN$), υπολογίζει τους κόμβους που απέχουν από όλους τους service nodes $\geq d_{max}/2$.
9. Αν σε κάποιο από τα παραπάνω βήματα δεν βρεθεί κανένας, μειώνεται το $d_{max}/2$ κατά μία μονάδα και γίνεται ίσο με $(\lfloor d_{max}/2 \rfloor - 1)$.
10. Επιλέγεται αυτός με την μικρότερο κόστος.
11. Αν υπάρχουν παραπάνω από μία επιλογές, γίνεται τυχαία επιλογή.

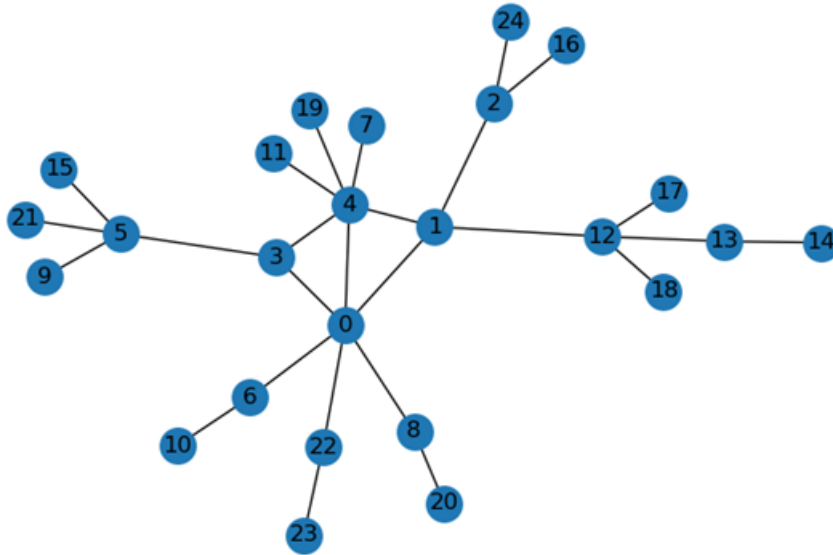
Ο δεύτερος αλγόριθμος ακολουθεί την ίδια διαδικασία με τον πρώτο όσον αφορά την επιλογή των service nodes, διαφέρει όσον αφορά την κατανομή των access nodes στους επιλεχθέντες service nodes. Ο συγκεκριμένος αλγόριθμος προσπαθεί να επιτύχει την χαμηλότερη δυνατή διακύμανση του δικτύου, χρησιμοποιώντας την τεχνική round robin. Σε κάθε γύρο εξετάζεται ένας service node, στον οποίο ανατίθεται ο πιο κοντινός του access node. Η διαδικασία ολοκληρώνεται όταν έχουν ανατεθεί όλοι οι access nodes στους αντίστοιχους service nodes. Με αυτή την τεχνική επιτυγχάνεται load balancing στο δίκτυο.

Σαν αποτέλεσμα έχει οι επιλογές του να έχουν μεγαλύτερο κόστος καθυστέρησης, συγκριτικά με τις αντίστοιχες του πρώτου ευριστικού αλγορίθμου.

Για το παρακάτω δίκτυο, σε τοπολογία scale free, έχουμε την κάτωθι κατανομή χρηστών στους κόμβους. Όπως είναι εύκολα αντιληπτό δίνεται ίδιο βάρος σε όλους του κόμβους.

Πίνακας 3. Κατανομή βάρους (χρηστών) στους κόμβους

#Node	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Users	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1



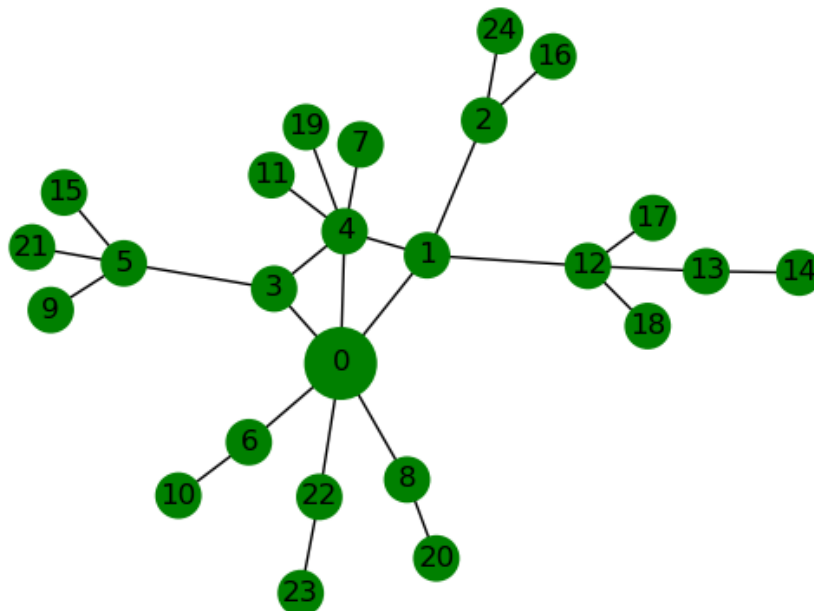
Εικόνα 11. Τοπολογία προς εξέταση

Στον κατωτέρω πίνακα (minimum distance cost matrix) απεικονίζεται τόσο η απόσταση κάθε κόμβου σε σχέση με τους υπόλοιπους μεμονωμένα, όσο και η συνολική απόσταση του από όλους τους κόμβους. Βάσει των υπολογισμών του κόστους στην τελευταία στήλη, η βέλτιστη επιλογή για έναν service node θα ήταν ο access node 0, ο οποίος αποτελεί τον κεντρικό κόμβο του δικτύου.

Πίνακας4. Minimum distance cost matrix για έναν service node

#Node	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	Total Cost
0	0	1	2	1	1	2	1	2	1	3	2	2	2	3	4	3	3	3	3	2	2	3	1	2	3	52
1	1	0	1	2	1	3	2	2	2	4	3	2	1	2	3	4	2	2	2	2	3	4	2	3	2	55
2	2	1	0	3	2	4	3	3	3	5	4	3	2	3	4	5	1	3	3	3	4	5	3	4	1	74
3	1	2	3	0	1	1	2	2	2	2	3	2	3	4	5	2	4	4	4	2	3	2	2	3	4	63
4	1	1	2	1	0	2	2	1	2	3	3	1	2	3	4	3	3	3	3	1	3	3	2	3	3	55
5	2	3	4	1	2	0	3	3	3	1	4	3	4	5	6	1	5	5	5	3	4	1	3	4	5	80
6	1	2	3	2	2	3	0	3	2	4	1	3	3	4	5	4	4	4	4	3	3	4	2	3	4	73
7	2	2	3	2	1	3	3	0	3	4	4	2	3	4	5	4	4	4	4	2	4	4	3	4	4	78
8	1	2	3	2	2	3	2	3	0	4	3	3	3	4	5	4	4	4	4	3	1	4	2	3	4	73
9	3	4	5	2	3	1	4	4	4	0	5	4	5	6	7	2	6	6	6	4	5	2	4	5	6	103
10	2	3	4	3	3	4	1	4	3	5	0	4	4	5	6	5	5	5	5	4	4	5	3	4	5	96
11	2	2	3	2	1	3	3	2	3	4	4	0	3	4	5	4	4	4	4	2	4	4	3	4	4	78
12	2	1	2	3	2	4	3	3	3	5	4	3	0	1	2	5	3	1	1	3	4	5	3	4	3	70
13	3	2	3	4	3	5	4	4	4	6	5	4	1	0	1	6	4	2	2	4	5	6	4	5	4	91
14	4	3	4	5	4	6	5	5	5	7	6	5	2	1	0	7	5	3	3	5	6	7	5	6	5	114
15	3	4	5	2	3	1	4	4	4	2	5	4	5	6	7	0	6	6	6	4	5	2	4	5	6	103
16	3	2	1	4	3	5	4	4	4	6	5	4	3	4	5	6	0	4	4	4	5	6	4	5	2	97
17	3	2	3	4	3	5	4	4	4	6	5	4	1	2	3	6	4	0	2	4	5	6	4	5	4	93
18	3	2	3	4	3	5	4	4	4	6	5	4	1	2	3	6	4	2	0	4	5	6	4	5	4	93
19	2	2	3	2	1	3	3	2	3	4	4	2	3	4	5	4	4	4	4	0	4	4	3	4	4	78
20	2	3	4	3	3	4	3	4	1	5	4	4	4	5	6	5	5	5	5	4	0	5	3	4	5	96
21	3	4	5	2	3	1	4	4	4	2	5	4	5	6	7	2	6	6	6	4	5	0	4	5	6	103
22	1	2	3	2	2	3	2	3	2	4	3	3	3	4	5	4	4	4	4	3	3	4	0	1	4	73
23	2	3	4	3	3	4	3	4	3	5	4	4	4	5	6	5	5	5	5	4	4	5	1	0	5	96
24	3	2	1	4	3	5	4	4	4	6	5	4	3	4	5	6	2	4	4	4	5	6	4	5	0	97

Συνεπώς η επιλογή και η κατανομή των υπόλοιπων access nodes θα ήταν η παρακάτω.



Εικόνα 11. Επιλογή ενός service node

Ακολούθως επιλέγεται ο αμέσως επόμενος καλύτερος κόμβος βάσει ταξινόμησης, ο οποίος είναι ο κόμβος 1, με τιμή κόστους 55. Η μεγαλύτερη απόσταση από αυτόν,

όπως φαίνεται στον πίνακα αποστάσεων, είναι ίση με 4 (d_{max}) και αφορά την απόσταση του κόμβου 1 από τους κόμβους 9,15,21. Η απόσταση αναφοράς θα είναι ίση με 2 ($d_{max}/2$).

Πίνακας 5. Minimum distance cost matrix για τον υπολογισμό d_{max}

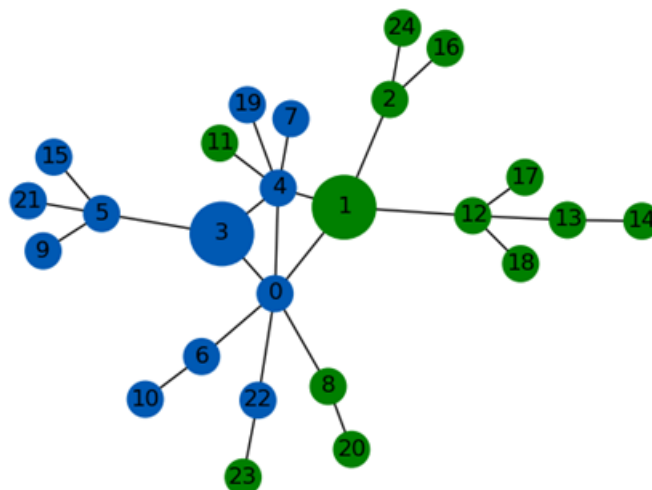
#Node	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	Total Cost
0	0	1	2	1	1	2	1	2	1	3	2	2	2	3	4	3	3	3	3	2	2	3	1	2	3	52
1	1	0	1	2	1	3	2	2	2	4	3	2	1	2	3	4	2	2	2	2	3	4	2	3	2	55
2	2	1	0	3	2	4	3	3	3	5	4	3	2	3	4	5	1	3	3	3	4	5	3	4	1	74
3	1	2	3	0	1	1	2	2	2	2	3	2	3	4	5	2	4	4	4	2	3	2	2	3	4	63
4	1	1	2	1	0	2	2	1	2	3	3	1	2	3	4	3	3	3	3	1	3	3	2	3	3	55
5	2	3	4	1	2	0	3	3	3	1	4	3	4	5	6	1	5	5	5	3	4	1	3	4	5	80
6	1	2	3	2	2	3	0	3	2	4	1	3	3	4	5	4	4	4	4	3	3	4	2	3	4	73
7	2	2	3	2	1	3	3	0	3	4	4	2	3	4	5	4	4	4	4	2	4	4	3	4	4	78
8	1	2	3	2	2	3	2	3	0	4	3	3	3	4	5	4	4	4	4	3	1	4	2	3	4	73
9	3	4	5	2	3	1	4	4	4	0	5	4	5	6	7	2	6	6	6	4	5	2	4	5	6	103
10	2	3	4	3	3	4	1	4	3	5	0	4	4	5	6	5	5	5	5	4	4	5	3	4	5	96
11	2	2	3	2	1	3	3	2	3	4	4	0	3	4	5	4	4	4	4	2	4	4	3	4	4	78
12	2	1	2	3	2	4	3	3	3	5	4	3	0	1	2	5	3	1	1	3	4	5	3	4	3	70
13	3	2	3	4	3	5	4	4	4	6	5	4	1	0	1	6	4	2	2	4	5	6	4	5	4	91
14	4	3	4	5	4	6	5	5	5	7	6	5	2	1	0	7	5	3	3	5	6	7	5	6	5	114
15	3	4	5	2	3	1	4	4	4	2	5	4	5	6	7	0	6	6	6	4	5	2	4	5	6	103
16	3	2	1	4	3	5	4	4	4	6	5	4	3	4	5	6	0	4	4	4	5	6	4	5	2	97
17	3	2	3	4	3	5	4	4	4	6	5	4	1	2	3	6	4	0	2	4	5	6	4	5	4	93
18	3	2	3	4	3	5	4	4	4	6	5	4	1	2	3	6	4	2	0	4	5	6	4	5	4	93
19	2	2	3	2	1	3	3	2	3	4	4	2	3	4	5	4	4	4	4	0	4	4	3	4	4	78
20	2	3	4	3	3	4	3	4	1	5	4	4	4	5	6	5	5	5	5	4	0	5	3	4	5	96
21	3	4	5	2	3	1	4	4	4	2	5	4	5	6	7	2	6	6	6	4	5	0	4	5	6	103
22	1	2	3	2	2	3	2	3	2	4	3	3	3	4	5	4	4	4	4	3	3	4	0	1	4	73
23	2	3	4	3	3	4	3	4	3	5	4	4	4	5	6	5	5	5	5	4	4	5	1	0	5	96
24	3	2	1	4	3	5	4	4	4	6	5	4	3	4	5	6	2	4	4	4	5	6	4	5	0	97

Από τους κόμβους με απόσταση μεγαλύτερη από 2, επιλέγονται αυτοί με την μικρότερη τιμή, οι οποίοι απεικονίζονται στον παρακάτω πίνακα με κόκκινο. Δεδομένου ότι έχουμε παραπάνω από μια επιλογές, η επιλογή θα γίνει τυχαία. Βάσει τυχαίας επιλογής, επιλέχθηκε ως δεύτερος service node ο κόμβος 3.

Πίνακας 6. Minimum distance cost matrix για τον υπολογισμό του δεύτερου service node

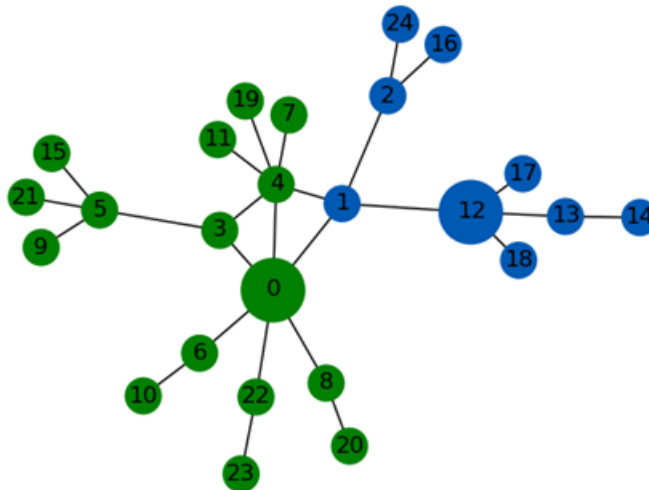
#Node	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	Total Cost
0	0	1	2	1	1	2	1	2	1	3	2	2	2	3	4	3	3	3	3	2	2	3	1	2	3	52
1	1	0	1	2	1	3	2	2	2	4	3	2	1	2	3	4	2	2	2	2	3	4	2	3	2	55
2	2	1	0	3	2	4	3	3	3	5	4	3	2	3	4	5	1	3	3	3	4	5	3	4	1	74
3	1	2	3	0	1	1	2	2	2	2	3	2	3	4	5	2	4	4	4	2	3	2	2	3	4	63
4	1	1	2	1	0	2	2	1	2	3	3	1	2	3	4	3	3	3	3	1	3	3	2	3	3	55
5	2	3	4	1	2	0	3	3	3	1	4	3	4	5	6	1	5	5	5	3	4	1	3	4	5	80
6	1	2	3	2	2	3	0	3	2	4	1	3	3	4	5	4	4	4	4	3	3	4	2	3	4	73
7	2	2	3	2	1	3	3	0	3	4	4	2	3	4	5	4	4	4	4	2	4	4	3	4	4	78
8	1	2	3	2	2	3	2	3	0	4	3	3	3	4	5	4	4	4	4	3	1	4	2	3	4	73
9	3	4	5	2	3	1	4	4	4	0	5	4	5	6	7	2	6	6	6	4	5	2	4	5	6	103
10	2	3	4	3	3	4	1	4	3	5	0	4	4	5	6	5	5	5	5	4	4	5	3	4	5	96
11	2	2	3	2	1	3	3	2	3	4	4	0	3	4	5	4	4	4	4	2	4	4	3	4	4	78
12	2	1	2	3	2	4	3	3	3	5	4	3	0	1	2	5	3	1	1	3	4	5	3	4	3	70
13	3	2	3	4	3	5	4	4	4	6	5	4	1	0	1	6	4	2	2	4	5	6	4	5	4	91
14	4	3	4	5	4	6	5	5	5	7	6	5	2	1	0	7	5	3	3	5	6	7	5	6	5	114
15	3	4	5	2	3	1	4	4	4	2	5	4	5	6	7	0	6	6	6	4	5	2	4	5	6	103
16	3	2	1	4	3	5	4	4	4	6	5	4	3	4	5	6	0	4	4	4	5	6	4	5	2	97
17	3	2	3	4	3	5	4	4	4	6	5	4	1	2	3	6	4	0	2	4	5	6	4	5	4	93
18	3	2	3	4	3	5	4	4	4	6	5	4	1	2	3	6	4	2	0	4	5	6	4	5	4	93
19	2	2	3	2	1	3	3	2	3	4	4	2	3	4	5	4	4	4	4	0	4	4	3	4	4	78
20	2	3	4	3	3	4	3	4	1	5	4	4	4	5	6	5	5	5	5	4	0	5	3	4	5	96
21	3	4	5	2	3	1	4	4	4	2	5	4	5	6	7	2	6	6	6	4	5	0	4	5	6	103
22	1	2	3	2	2	3	2	3	2	4	3	3	3	4	5	4	4	4	4	3	3	4	0	1	4	73
23	2	3	4	3	3	4	3	4	3	5	4	4	4	5	6	5	5	5	5	4	4	5	1	0	5	96
24	3	2	1	4	3	5	4	4	4	6	5	4	3	4	5	6	2	4	4	4	5	6	4	5	0	97

Βάσει των παραπάνω προκύπτει η παρακάτω κατανομή των access nodes στους δυο επιλεχθέντες service nodes (1,3). Έτσι οι κόμβοι 2,8,11,12,13,14,16,17,18,20,23,24 εξυπηρετούνται από τον service node 1 και οι κόμβοι 0,4,5,6,7,9,10,15,19,21,22 εξυπηρετούνται από τον service node 3, όπως αποτυπώνεται και στο κάτωθι διάγραμμα.



Εικόνα 12. Επιλογή δυο service nodes βάσει του πρώτου ευριστικού αλγορίθμου

Η συγκεκριμένη επιλογή επιφέρει το βέλτιστο όσον αφορά την διακύμανση της ανάθεσης των κόμβων πρόσβασης σε κόμβους υπηρεσίας του δικτύου (0,5) και διαφέρει κατά 3 μονάδες όσον αφορά το κόστος καθυστέρησης (45ms) έναντι του αποτελέσματος του εξαντλητικού (42ms), σύμφωνα με τον οποίο η βέλτιστη επιλογή θα ήταν οι κόμβοι 0 και 12, με υποπολλαπλάσια κατανομή (24,50). Δεδομένου ότι από πλευράς διακύμανσης έχουμε το βέλτιστο αποτέλεσμα, δε θα υπάρξει κάποια διαφορά όσον αφορά τα ευρήματα του δεύτερου ευριστικού αλγορίθμου.

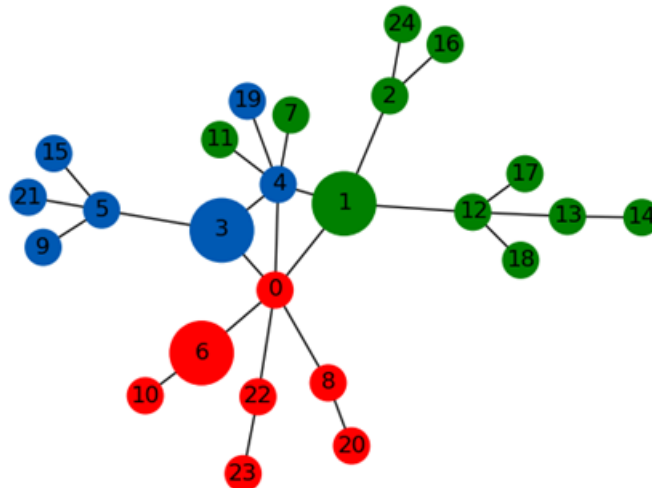


Εικόνα 13. Επιλογή δυο service nodes βάσει του εξαντλητικού αλγορίθμου

Στην συνέχεια υπολογίζονται οι κόμβοι οι οποίοι απέχουν και από τους δυο επιλεγθέντες service nodes, απόσταση μεγαλύτερη από 2. Από αυτούς επιλέγονται αρχικά αυτοί με τις μικρότερες τιμές, άρα όσοι απέχουν απόσταση ίση με 2 και από τις 2 κόμβους. Αυτοί είναι οι κόμβοι 6,7,8,11,19,22, οι οποίοι απεικονίζονται με κόκκινο χρώμα.

Πίνακας 7. Minimum distance cost matrix για τον υπολογισμό του τρίτου service node

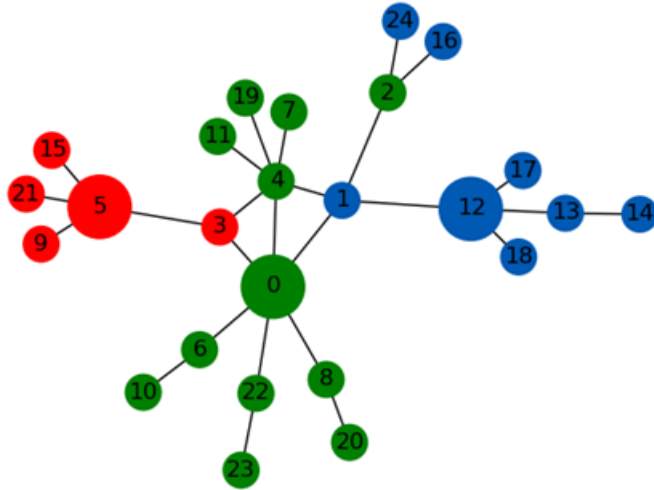
#Node	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	Total Cost
0	0	1	2	1	1	2	1	2	1	3	2	2	2	3	4	3	3	3	3	2	2	3	1	2	3	52
1	1	0	1	2	1	3	2	2	2	4	3	2	1	2	3	4	2	2	2	2	3	4	2	3	2	55
2	2	1	0	3	2	4	3	3	3	5	4	3	2	3	4	5	1	3	3	3	4	5	3	4	1	74
3	1	2	3	0	1	1	2	2	2	2	3	2	3	4	5	2	4	4	4	2	3	2	2	3	4	63
4	1	1	2	1	0	2	2	1	2	3	3	1	2	3	4	3	3	3	3	1	3	3	2	3	3	55
5	2	3	4	1	2	0	3	3	3	1	4	3	4	5	6	1	5	5	5	3	4	1	3	4	5	80
6	1	2	3	2	2	3	0	3	2	4	1	3	3	4	5	4	4	4	4	3	3	4	2	3	4	73
7	2	2	3	2	1	3	3	0	3	4	4	2	3	4	5	4	4	4	4	2	4	4	3	4	4	78
8	1	2	3	2	2	3	2	3	0	4	3	3	3	4	5	4	4	4	4	3	1	4	2	3	4	73
9	3	4	5	2	3	1	4	4	4	0	5	4	5	6	7	2	6	6	6	4	5	2	4	5	6	103
10	2	3	4	3	3	4	1	4	3	5	0	4	4	5	6	5	5	5	5	4	4	5	3	4	5	96
11	2	2	3	2	1	3	3	2	3	4	4	0	3	4	5	4	4	4	4	2	4	4	3	4	4	78
12	2	1	2	3	2	4	3	3	3	5	4	3	0	1	2	5	3	1	1	3	4	5	3	4	3	70
13	3	2	3	4	3	5	4	4	4	6	5	4	1	0	1	6	4	2	2	4	5	6	4	5	4	91
14	4	3	4	5	4	6	5	5	5	7	6	5	2	1	0	7	5	3	3	5	6	7	5	6	5	114
15	3	4	5	2	3	1	4	4	4	2	5	4	5	6	7	0	6	6	6	4	5	2	4	5	6	103
16	3	2	1	4	3	5	4	4	4	6	5	4	3	4	5	6	0	4	4	4	5	6	4	5	2	97
17	3	2	3	4	3	5	4	4	4	6	5	4	1	2	3	6	4	0	2	4	5	6	4	5	4	93
18	3	2	3	4	3	5	4	4	4	6	5	4	1	2	3	6	4	2	0	4	5	6	4	5	4	93
19	2	2	3	2	1	3	3	2	3	4	4	2	3	4	5	4	4	4	4	0	4	4	3	4	4	78
20	2	3	4	3	3	4	3	4	1	5	4	4	4	5	6	5	5	5	5	4	0	5	3	4	5	96
21	3	4	5	2	3	1	4	4	4	2	5	4	5	6	7	2	6	6	6	4	5	0	4	5	6	103
22	1	2	3	2	2	3	2	3	2	4	3	3	3	4	5	4	4	4	4	3	3	4	0	1	4	73
23	2	3	4	3	3	4	3	4	3	5	4	4	4	5	6	5	5	5	5	4	4	5	1	0	5	96
24	3	2	1	4	3	5	4	4	4	6	5	4	3	4	5	6	2	4	4	4	5	6	4	5	0	97



Εικόνα 13. Επιλογή τριών service nodes βάσει του πρώτου ευριστικού αλγορίθμου.

Στην συγκεκριμένη περίπτωση οι κόμβοι 2,7,11,12,13,14,16,17,18,24 θα εξυπηρετηθούν από τον service node 1, οι κόμβοι 4,5,9,15,19,21 θα εξυπηρετηθούν από τον service node 3 και οι κόμβοι 0,8,10,20,22,23 από τον service node 6.

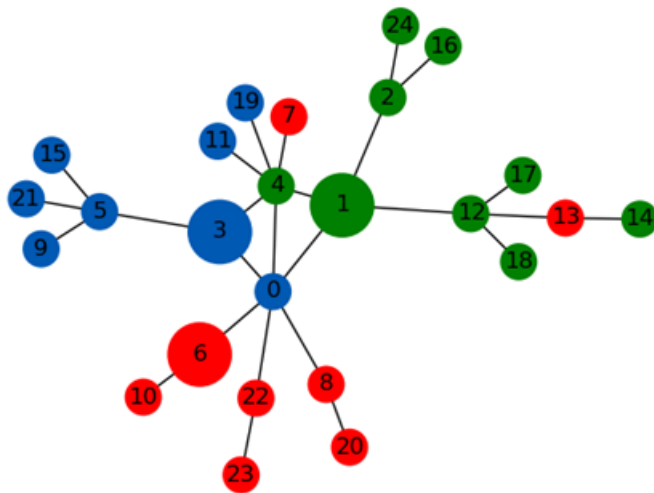
Γίνεται αντιληπτό ότι η συγκεκριμένη επιλογή δεν είναι η βέλτιστη από άποψη κατανομής των χρηστών, παρόλα αυτά είναι κατά πολύ μικρότερη, 5,33 έναντι 12,33, από την αντίστοιχη του εξαντλητικού αλγορίθμου. Σύμφωνα με τον εξαντλητικό αλγόριθμο, ο βέλτιστος συνδυασμός από άποψης κόστους θα ήταν οι κόμβοι 0,5,12 με latency cost 34ms, έναντι 41ms του πρώτου ευριστικού.



Εικόνα 14. Επιλογή τριών service nodes βάσει του εξαντλητικού αλγορίθμου

Επισημαίνεται πως ακόμα και αν ληφθούν υπόψη οι άλλες εναλλακτικές του εξαντλητικού αλγορίθμου, η βέλτιστη επιλογή όσον αφορά την διακύμανση είναι ο συνδυασμός των κόμβων 0,1,5 με latency cost 35ms, αυξημένο κατά μία μονάδα σε σχέση με πριν και διακύμανση ίση με 9,33.

Λόγω του ότι όπως αναφέρθηκε η επιλογή του πρώτου ευριστικού αλγορίθμου δεν είναι η βέλτιστη, υπάρχει διαφορά όσον αφορά την κατανομή των access nodes στους ήδη υπάρχοντες service nodes. Έχουμε λοιπόν την βέλτιστη επιλογή από άποψη διακύμανσης μεταξύ των τριών service nodes, ίση με 0,33 , με αύξηση του latency cost κατά 3 μονάδες (44ms). Με τον ίδιο τρόπο λειτουργεί ο αλγόριθμος και για τις υπόλοιπες περιπτώσεις.



Εικόνα 15. Επιλογή τριών service nodes βάσει του δεύτερου ευριστικού αλγορίθμου.

Κεφάλαιο 4^ο Μελέτη αποτελεσμάτων

Από τις μετρήσεις που διενεργήθηκαν, έχουν εξαχθεί τα αποτελέσματα των αλγορίθμων και έχει πραγματοποιηθεί σύγκριση των αποτελεσμάτων σε επίπεδο κόστους καθυστέρησης δικτύου, κόστους καθυστέρησης ανά χρήστη και διακύμανσης, στα τρία σενάρια που έχουν αναφερθεί, για κάθε μια από τις δυο τοπολογίες.

Βάσει των ορίων κατανομής των βαρών που τέθηκαν, προκύπτει διαφορετικός μέσος αριθμός χρηστών σε κάθε μια περίπτωση.

Στην περίπτωση που θέτουμε σαν τιμή βάρους ίση με την μονάδα για κάθε access node, ο συνολικός αριθμός των χρηστών είναι 25. Στο σενάριο όπου δίνεται βάρος στους κεντρικούς κόμβους, δηλαδή υπάρχει μεγαλύτερη συγκέντρωση χρηστών στο κέντρο του δικτύου, ο αριθμός των χρηστών προκύπτει ως εξής.

Στην τοπολογία scale free έχουμε τέσσερις κεντρικούς κόμβους και είκοσιένα περιφερειακούς. Το εύρος από το οποίο αποδίδεται βάρος στους κεντρικούς είναι από 6 έως 9, συνεπώς η μέση τιμή χρηστών είναι 7,5. Το εύρος από το οποίο αποδίδεται βάρος στους περιφερειακούς είναι από 1 έως 3, η μέση τιμή χρηστών σε κάθε κόμβο είναι ίση με 2.

Πολλαπλασιάζοντας τον αριθμό των κόμβων κάθε κατηγορίας με την αντίστοιχη μέση τιμή, έχουμε τον μέσο αριθμό των χρηστών. Συνεπώς για την τοπολογία scale free έχουμε τους παρακάτω αριθμούς χρηστών.

Πίνακας 8. Υπολογισμός χρηστών σε τοπολογία scale free

Topology	Weights of Center nodes	Weights of Edge nodes	Average users
Scale free	(6,9)	(1,3)	72
Scale free	(1,3)	(6,9)	165

Αντιστοίχως και για την τοπολογία lattice, με την μόνη διαφορά ότι υπάρχουν πέντε κεντρικοί κόμβοι και είκοσι περιφερειακοί, υπολογίζεται ο μέσος αριθμός των χρηστών.

Πίνακας 9. Υπολογισμός χρηστών σε τοπολογία lattice

Topology	Weights of Center nodes	Weights of Edge nodes	Average users
Lattice	(6,9)	(1,3)	77
Lattice	(1,3)	(6,9)	160

4.1 Σενάριο 1^ο Uniform κατανομή

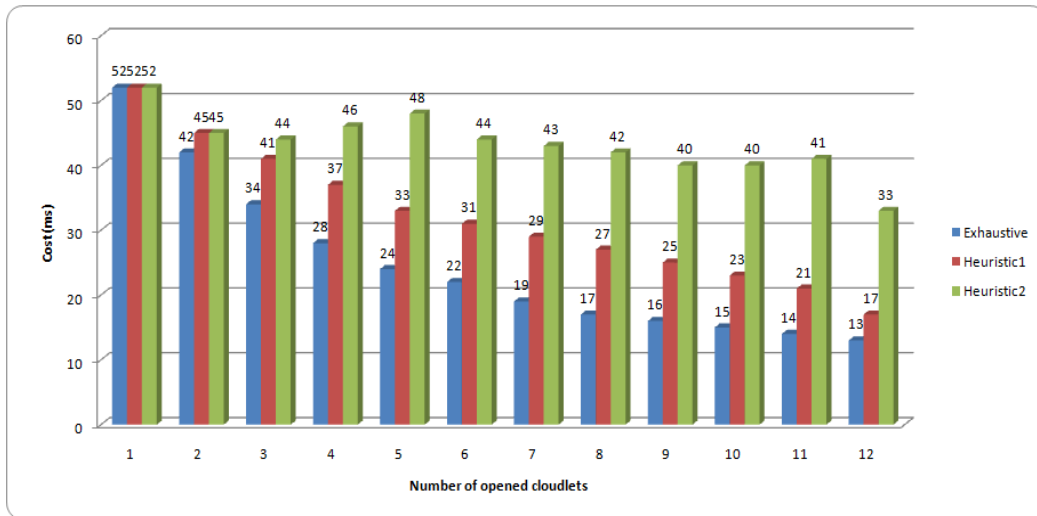
4.1.1 Τοπολογία scale free

Αρχικά για την τοπολογία scale free παρατίθεται πίνακας, στον οποίο αποτυπώνεται η μέση τιμή του κόστους καθυστέρησης, η μέση τιμή του κόστους καθυστέρησης ανά χρήστη και η μέση τιμή διακύμανσης σε κάθε σενάριο κόμβων, για κάθε έναν από τους αλγορίθμους. Επίσης αποτυπώνεται τόσο η διαφορά σε απόλυτες τιμές (ms) του πρώτου ευριστικού αλγορίθμου από τον εξαντλητικό.

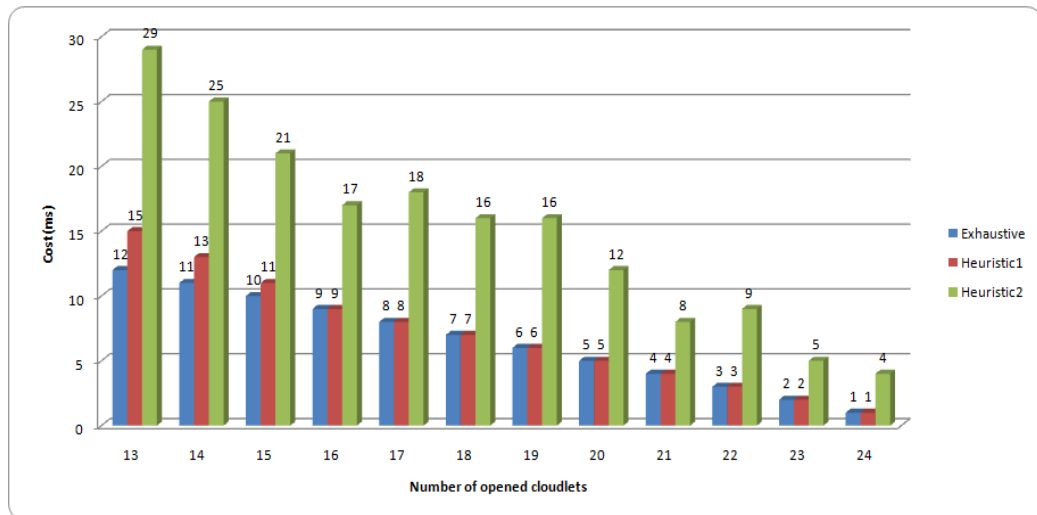
Παρατηρούμε ότι η μέγιστη διαφορά των τιμών όσον αφορά τον πρώτο ευριστικό αλγόριθμό είναι ίση με 10ms ή 0,40ms ανά χρήστη και εμφανίζεται στις περιπτώσεις που απαιτούνται επτά και οκτώ service nodes για την κάλυψη των αναγκών του δικτύου. Αντίστοιχα παρατηρείται αύξηση της διακύμανσης, ιδιαίτερα όταν απαιτούνται πέντε service nodes. Η συνολική αύξηση της καθυστέρησης του δικτύου ανέρχεται σε 24,33% (0,16ms ανά χρήστη) σε σχέση με τον εξαντλητικό αλγόριθμο. Όσον αφορά την διακύμανση παρουσιάζεται αύξηση 50,49%.

Ο δεύτερος ευριστικός αλγόριθμός παρά την δεδομένη αύξηση του κόστους καθυστέρησης που επιφέρει, παρουσιάζει χαμηλότερη διακύμανση από τον εξαντλητικό στην πλειοψηφία των σεναρίων, ιδιαιτέρως στις περιπτώσεις που απαιτούνται δυο (η μεγαλύτερη διαφορά), τρεις και τέσσερις service nodes. Συνολικά επιφέρει μείωση στην διακύμανση του δικτύου κατά 92,48% από τον εξαντλητικό. Αντίστοιχα η αύξηση του κόστους καθυστέρησης είναι 86,83% από τον εξαντλητικό. Εν συγκρίσει με τον πρώτο ευριστικό επιφέρει μείωση της διακύμανσης κατά 95,01%, αυξάνοντας το κόστος κατά 50,11%.

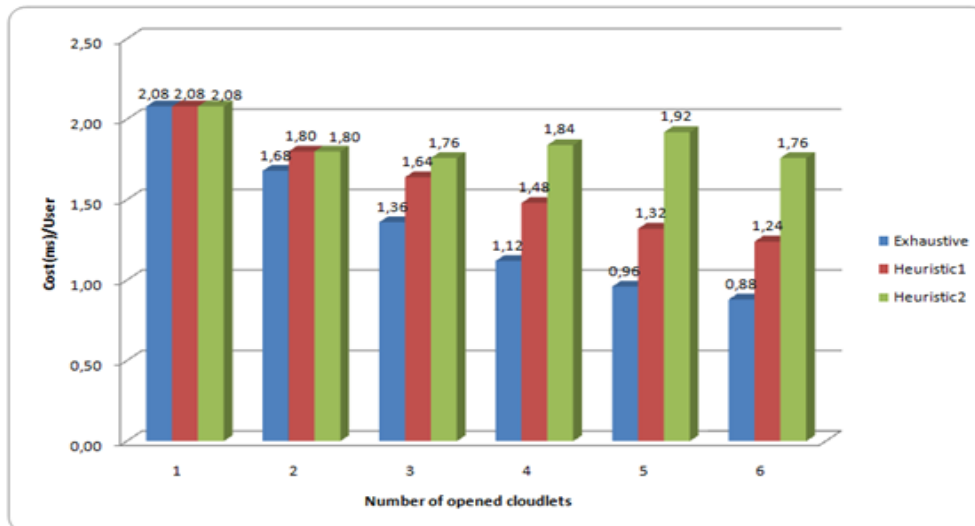
4.1.1.1 Γραφήματα



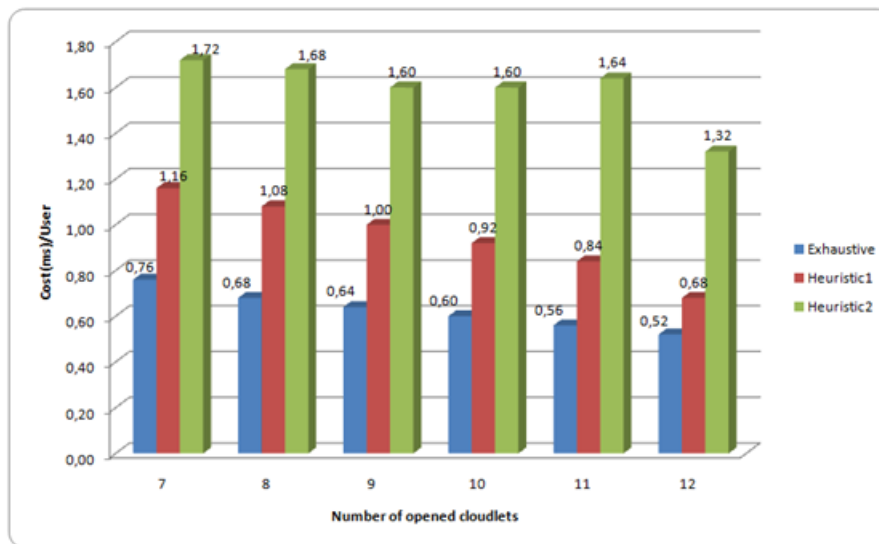
Εικόνα 15. Γράφημα κόστους σε τοπολογία scale free, uniform distribution



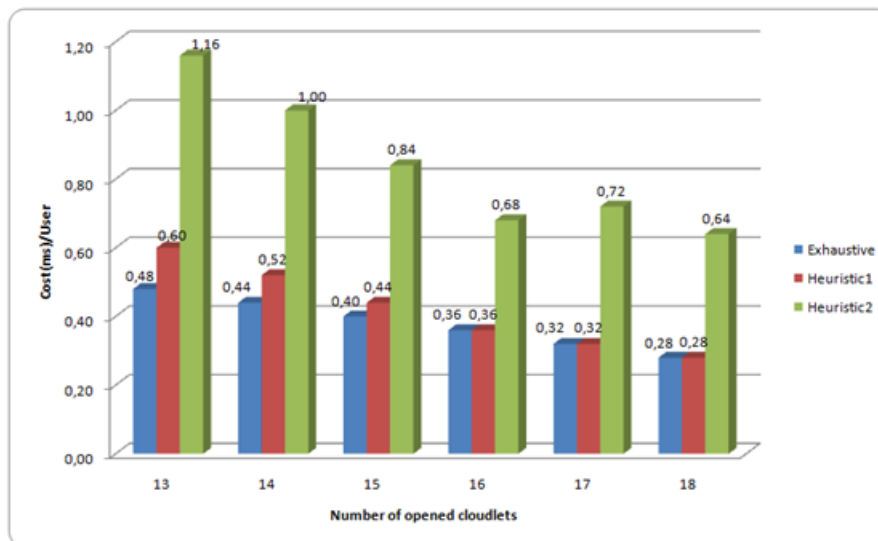
Εικόνα 16. Γράφημα κόστους σε τοπολογία scale free, uniform distribution



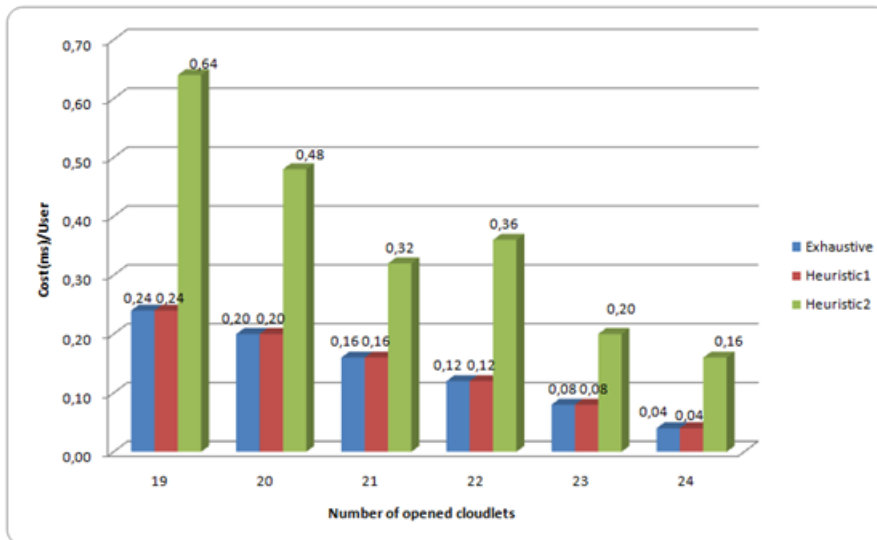
Εικόνα 17. Γράφημα κόστους ανά χρήστη σε τοπολογία scale free, uniform distribution



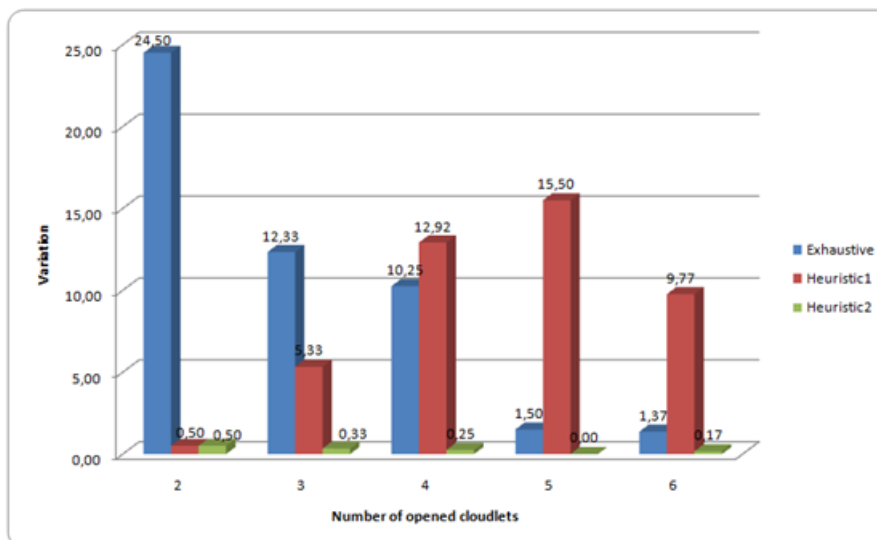
Εικόνα 18. Γράφημα κόστους ανά χρήστη σε τοπολογία scale free, uniform distribution



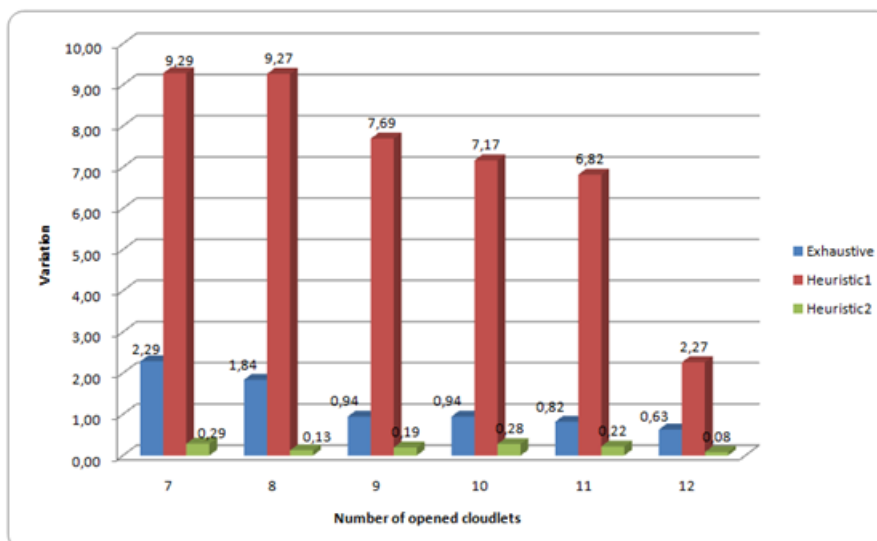
Εικόνα 19. Γράφημα κόστους ανά χρήστη σε τοπολογία scale free, uniform distribution



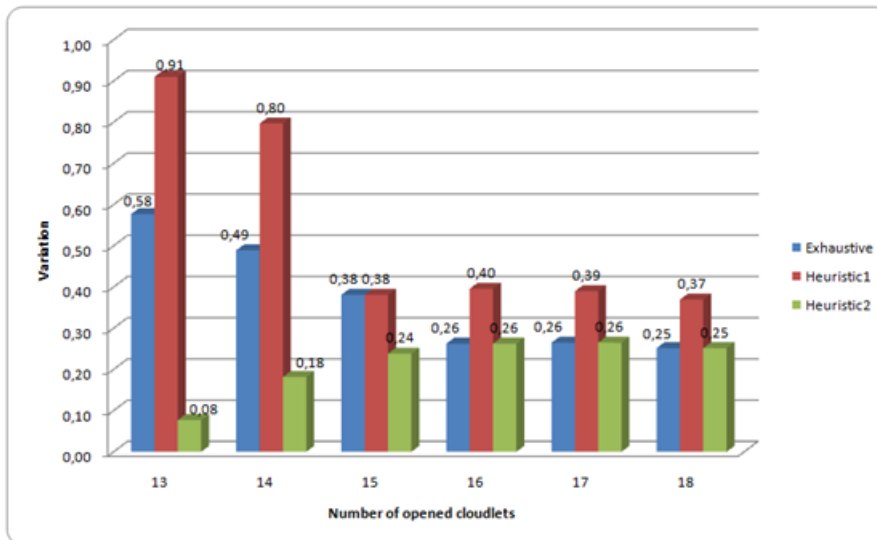
Εικόνα 20. Γράφημα κόστους ανά χρήστη σε τοπολογία scale free, uniform distribution



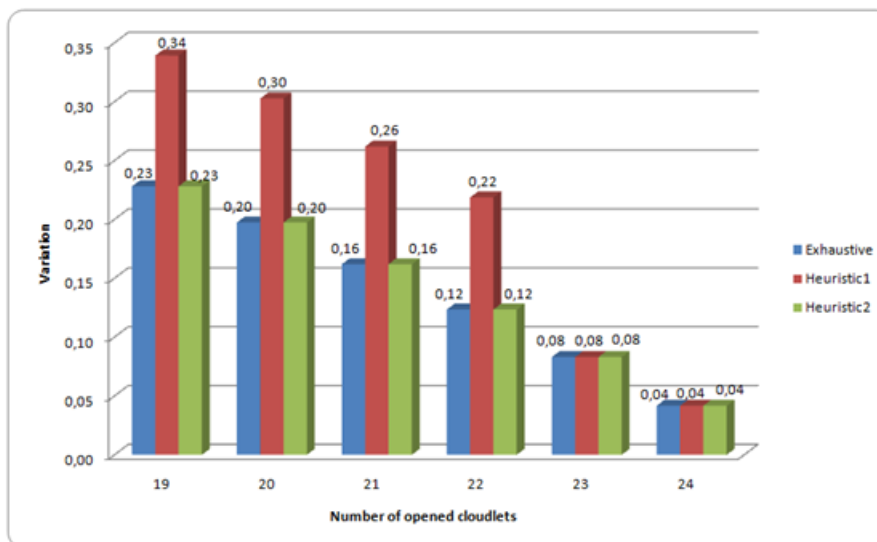
Εικόνα 21. Γράφημα διακύμανσης access node σε τοπολογία scale free, uniform distribution



Εικόνα 22. Γράφημα διακύμανσης access node σε τοπολογία scale free, uniform distribution



Εικόνα 23. Γράφημα διακύμανσης access node σε τοπολογία scale free, uniform distribution



Εικόνα 24. Γράφημα διακύμανσης access node σε τοπολογία scale free, uniform distribution

4.1.1.2 Συγκεντρωτικά αποτελέσματα

Πίνακας 9. Αποτελέσματα κόστους σε τοπολογία scale free, uniform distribution

# Nodes	Exh	Heur1	Heur2	Διαφορά heur1 (ms)	Διαφορά heur2 (ms)
1	52	52	52	0	0
2	42	45	45	3	3
3	34	41	44	7	10
4	28	37	46	9	18
5	24	33	48	9	24
6	22	31	44	9	22
7	19	29	43	10	24
8	17	27	42	10	25
9	16	25	40	9	24
10	15	23	40	8	25
11	14	21	41	7	27
12	13	17	33	4	20
13	12	15	29	3	17
14	11	13	25	2	14
15	10	11	21	1	11
16	9	9	17	0	8
17	8	8	18	0	10
18	7	7	16	0	9
19	6	6	16	0	10
20	5	5	12	0	7
21	4	4	8	0	4
22	3	3	9	0	6
23	2	2	5	0	3
24	1	1	4	0	3

Πίνακας 10. Αποτελέσματα κόστους ανά χρήση σε τοπολογία scale free, uniform distribution

# Nodes	Exh	Heur1	Heur2	Διαφορά heur1 (ms)	Διαφορά heur2 (ms)
1	2,08	2,08	2,08	0,00	0,00
2	1,68	1,80	1,80	0,12	0,12
3	1,36	1,64	1,76	0,28	0,40
4	1,12	1,48	1,84	0,36	0,72
5	0,96	1,32	1,92	0,36	0,96
6	0,88	1,24	1,76	0,36	0,88
7	0,76	1,16	1,72	0,40	0,96
8	0,68	1,08	1,68	0,40	1,00
9	0,64	1,00	1,60	0,36	0,96
10	0,60	0,92	1,60	0,32	1,00
11	0,56	0,84	1,64	0,28	1,08
12	0,52	0,68	1,32	0,16	0,80
13	0,48	0,60	1,16	0,12	0,68
14	0,44	0,52	1,00	0,08	0,56
15	0,40	0,44	0,84	0,04	0,44
16	0,36	0,36	0,68	0,00	0,32
17	0,32	0,32	0,72	0,00	0,40
18	0,28	0,28	0,64	0,00	0,36
19	0,24	0,24	0,64	0,00	0,40
20	0,20	0,20	0,48	0,00	0,28
21	0,16	0,16	0,32	0,00	0,16
22	0,12	0,12	0,36	0,00	0,24
23	0,08	0,08	0,20	0,00	0,12
24	0,04	0,04	0,16	0,00	0,12

Πίνακας 11. Αποτελέσματα διακύμανσης σε τοπολογία scale free, uniform distribution

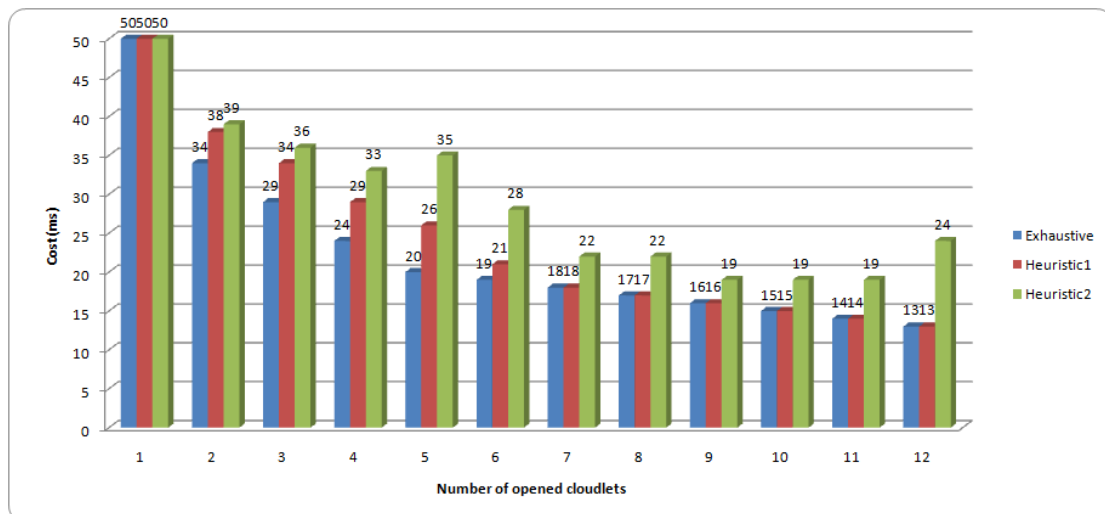
# Nodes	Exh	Heur1	Heur2	Διαφορά heur1	Διαφορά heur2
2	24,50	0,50	0,50	-24,00	-24,00
3	12,33	5,33	0,33	-7,00	-12,00
4	10,25	12,92	0,25	2,67	-10,00
5	1,50	15,50	0,00	14,00	-1,50
6	1,37	9,77	0,17	8,40	-1,20
7	2,29	9,29	0,29	7,00	-2,00
8	1,84	9,27	0,13	7,43	-1,71
9	0,94	7,69	0,19	6,75	-0,75
10	0,94	7,17	0,28	6,22	-0,67
11	0,82	6,82	0,22	6,00	-0,60
12	0,63	2,27	0,08	1,64	-0,55
13	0,58	0,91	0,08	0,33	-0,50
14	0,49	0,80	0,18	0,31	-0,31
15	0,38	0,38	0,24	0,00	-0,14
16	0,26	0,40	0,26	0,13	0,00
17	0,26	0,39	0,26	0,13	0,00
18	0,25	0,37	0,25	0,12	0,00
19	0,23	0,34	0,23	0,11	0,00
20	0,20	0,30	0,20	0,11	0,00
21	0,16	0,26	0,16	0,10	0,00
22	0,12	0,22	0,12	0,10	0,00
23	0,08	0,08	0,08	0,00	0,00
24	0,04	0,04	0,04	0,00	0,00

4.1.2 Τοπολογία lattice

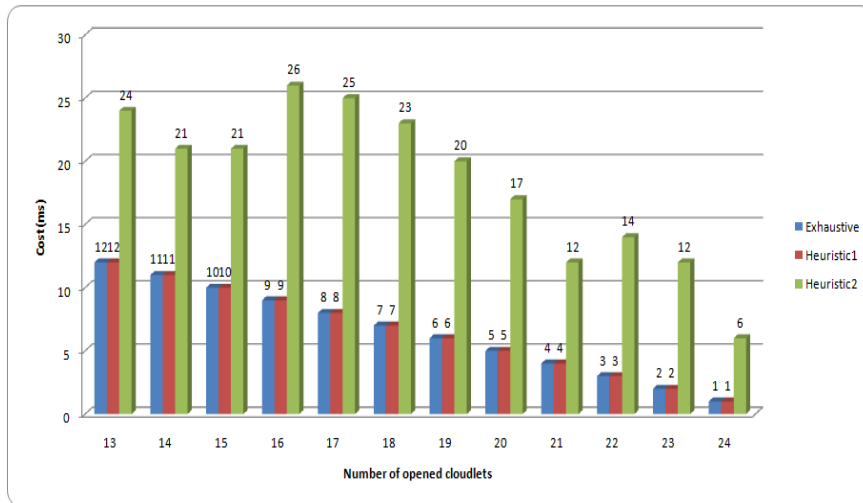
Όσον αφορά τον πρώτο ευριστικό αλγόριθμο, παρατηρείται πολύ μικρή διαφορά όσον αφορά την καθυστέρηση, η οποία εμφανίζεται στις περιπτώσεις που απαιτούνται από δυο έως πέντε service nodes, στις υπόλοιπες περιπτώσεις από επτά έως εικοσιτέσσερις service nodes, επιτυγχάνει ακριβώς τα ίδια αποτελέσματα με τον εξαντλητικό αλγόριθμο. Ομοίως στο κομμάτι της διακύμανσης παρουσιάζει αυξημένα αποτελέσματα, κυρίως για πέντε επιλεχθέντες service nodes. Η συνολική αύξηση που επιφέρει στο κόστος καθυστέρησης, σε σχέση με τον εξαντλητικό, ανέρχεται σε 6,34%, 0,52ms ανά χρήστη έναντι 0,45ms του εξαντλητικού. Η διακύμανση αυξάνεται κατά 2,4 φορές, ανέρχεται σε 1 έναντι 0,29 του εξαντλητικού

Ο δεύτερος ευριστικός αλγόριθμος παρουσιάζει αύξηση ύψους 63,40% (0,37ms ανά χρήστη) στο κόστος καθυστέρησης, ενώ παράλληλα μειώνει την διακύμανση κατά 35,58%. Σε σχέση με τον πρώτο ευριστικό μειώνει την διακύμανση κατά 85,34%, παρουσιάζοντας αύξηση στο κόστος κατά 53,66%.

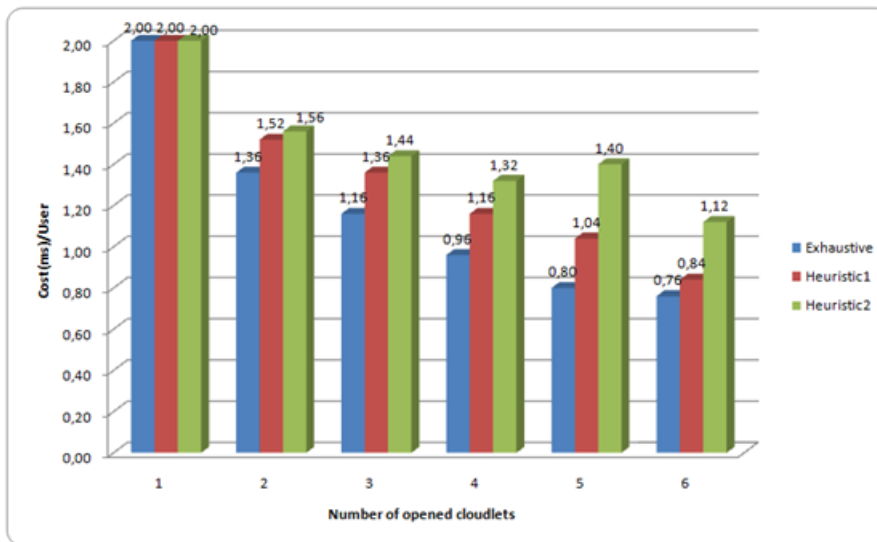
4.1.2.1 Γραφήματα



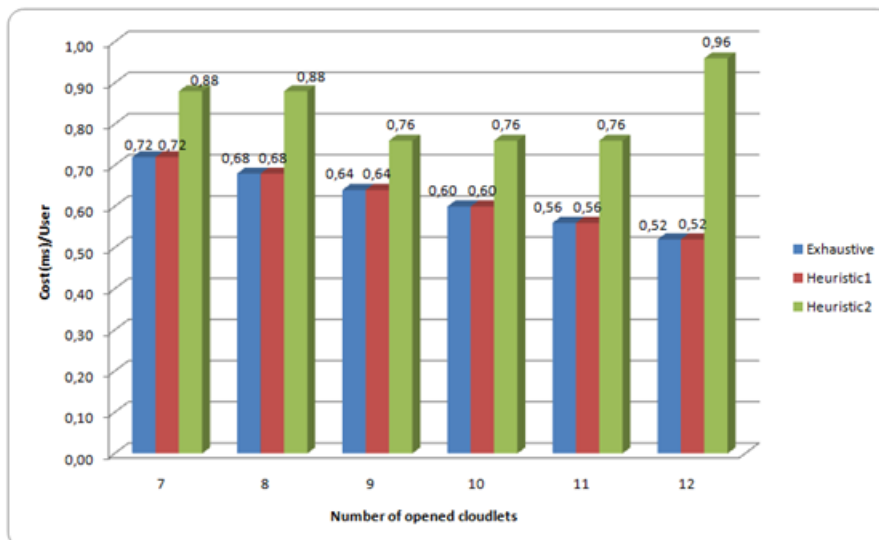
Εικόνα 25. Γράφημα κόστους σε τοπολογία lattice, uniform distribution



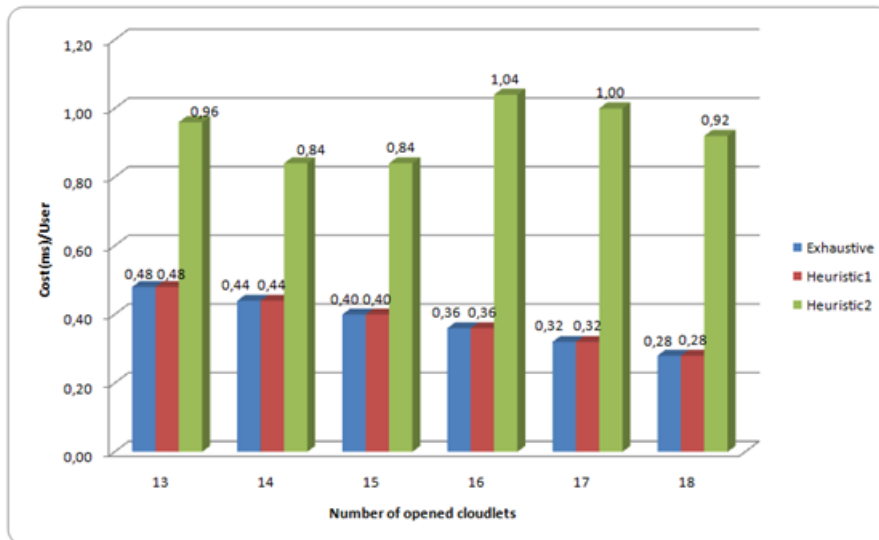
Εικόνα 26. Γράφημα κόστους σε τοπολογία lattice, uniform distribution



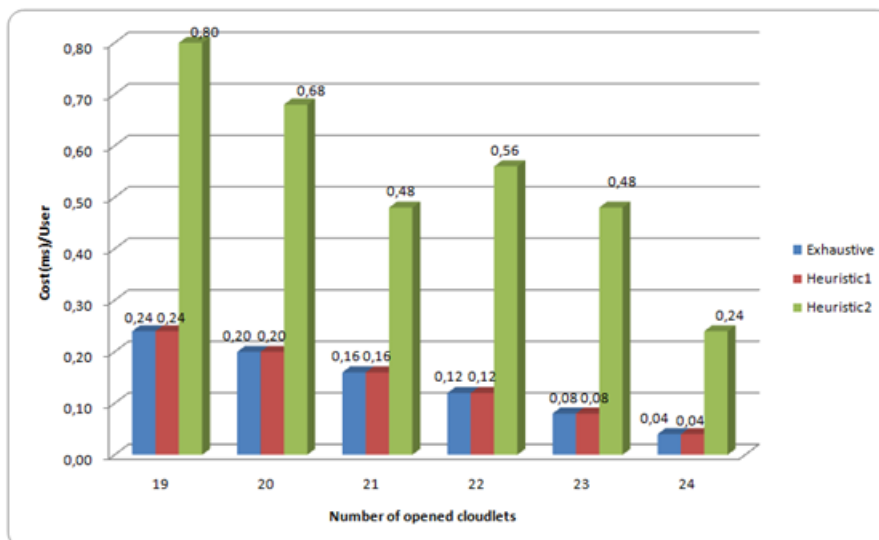
Εικόνα 27. Γράφημα κόστους ανά χρήστη σε τοπολογία lattice, uniform distribution



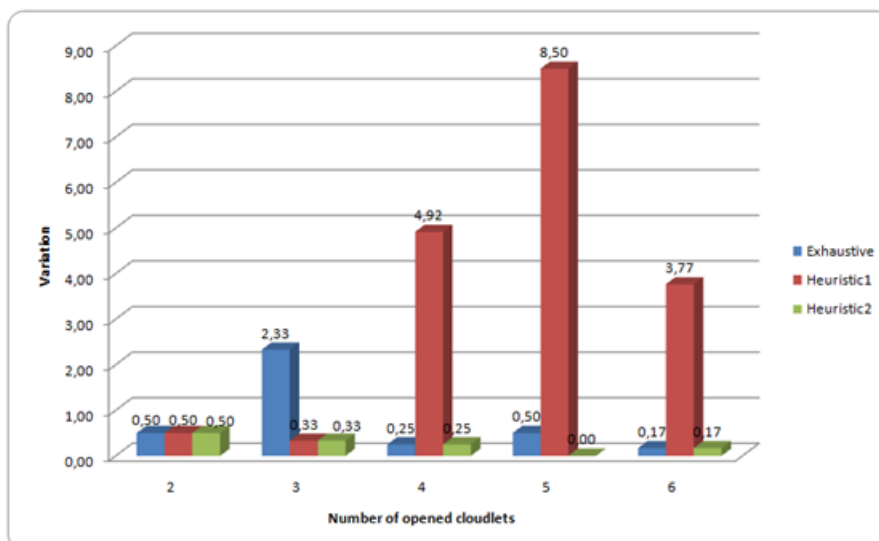
Εικόνα 28. Γράφημα κόστους ανά χρήστη σε τοπολογία lattice, uniform distribution



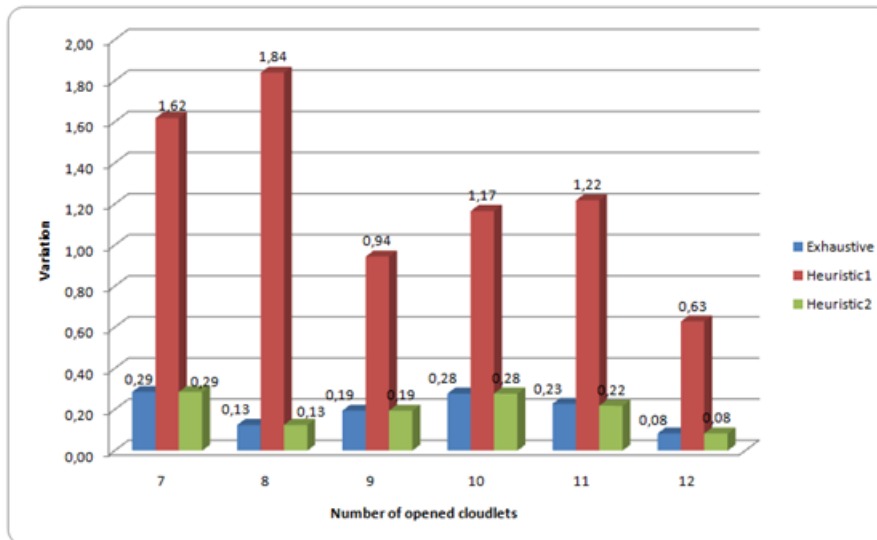
Εικόνα 29. Γράφημα κόστους ανά χρήστη σε τοπολογία lattice, uniform distribution



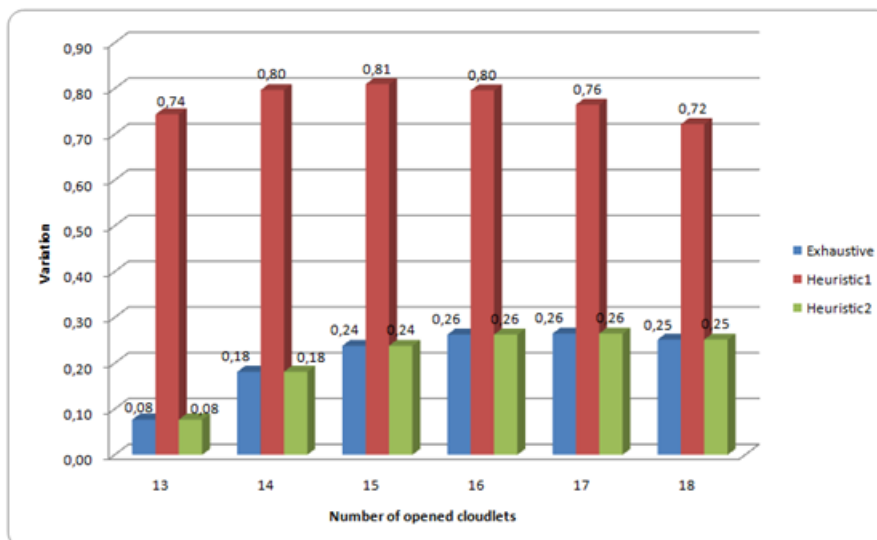
Εικόνα30. Γράφημα κόστους ανά χρήστη σε τοπολογία lattice, uniform distribution



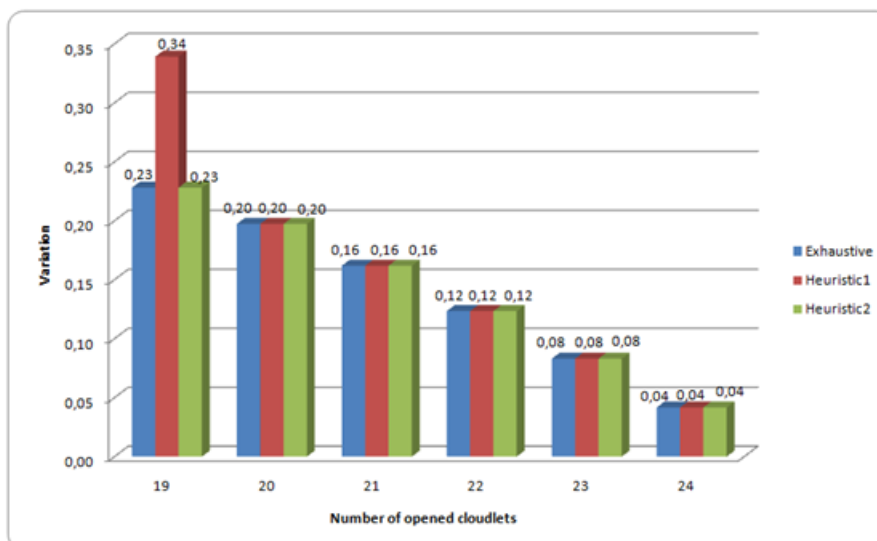
Εικόνα 31. Γράφημα διακύμανσης access node σε τοπολογία lattice, uniform distribution



Εικόνα32. Γράφημα διακύμανσης access node σε τοπολογία lattice, uniform distribution



Εικόνα33. Γράφημα διακύμανσης access node σε τοπολογία lattice, uniform distribution



Εικόνα 34. Γράφημα διακύμανσης access node σε τοπολογία lattice, uniform distribution

4.1.2.2 Συγκεντρωτικά αποτελέσματα

Πίνακας 12. Αποτελέσματα κόστους σε τοπολογία lattice, uniform distribution

# Nodes	Exh	Heur1	Heur2	Διαφορά heur1 (ms)	Διαφορά heur2 (ms)
1	50	50	50	0	0
2	34	38	39	4	5
3	29	34	36	5	7
4	24	29	33	5	9
5	20	26	35	6	15
6	19	21	28	2	9
7	18	18	22	0	4
8	17	17	22	0	5
9	16	16	19	0	3
10	15	15	19	0	4
11	14	14	19	0	5
12	13	13	24	0	11
13	12	12	24	0	12
14	11	11	21	0	10
15	10	10	21	0	11
16	9	9	26	0	17
17	8	8	25	0	17
18	7	7	23	0	16
19	6	6	20	0	14
20	5	5	17	0	12
21	4	4	12	0	8
22	3	3	14	0	11
23	2	2	12	0	10
24	1	1	6	0	5

Πίνακας 13. Αποτελέσματα κόστους ανά χρήστη σε τοπολογία lattice, uniform distribution

# Nodes	Exh	Heur1	Heur2	Διαφορά heur1 (ms)	Διαφορά heur2 (ms)
1	2,00	2,00	2,00	0,00	0,00
2	1,36	1,52	1,56	0,16	0,20
3	1,16	1,36	1,44	0,20	0,28
4	0,96	1,16	1,32	0,20	0,36
5	0,80	1,04	1,40	0,24	0,60
6	0,76	0,84	1,12	0,08	0,36
7	0,72	0,72	0,88	0,00	0,16
8	0,68	0,68	0,88	0,00	0,20
9	0,64	0,64	0,76	0,00	0,12
10	0,60	0,60	0,76	0,00	0,16
11	0,56	0,56	0,76	0,00	0,20
12	0,52	0,52	0,96	0,00	0,44
13	0,48	0,48	0,96	0,00	0,48
14	0,44	0,44	0,84	0,00	0,40
15	0,40	0,40	0,84	0,00	0,44
16	0,36	0,36	1,04	0,00	0,68
17	0,32	0,32	1,00	0,00	0,68
18	0,28	0,28	0,92	0,00	0,64
19	0,24	0,24	0,80	0,00	0,56
20	0,20	0,20	0,68	0,00	0,48
21	0,16	0,16	0,48	0,00	0,32
22	0,12	0,12	0,56	0,00	0,44
23	0,08	0,08	0,48	0,00	0,40
24	0,04	0,04	0,24	0,00	0,20

Πίνακας 14. Αποτελέσματα διακύμανσης σε τοπολογία scale free, uniform distribution

# Nodes	Exh	Heur1	Heur2	Διαφορά heur1	Διαφορά heur2
2	0,50	0,50	0,50	0,00	0,00
3	2,33	0,33	0,33	-2,00	-2,00
4	0,25	4,92	0,25	4,67	0,00
5	0,50	8,50	0,00	8,00	-0,50
6	0,17	3,77	0,17	3,60	0,00
7	0,29	1,62	0,29	1,33	0,00
8	0,13	1,84	0,13	1,71	0,00
9	0,19	0,94	0,19	0,75	0,00
10	0,28	1,17	0,28	0,89	0,00
11	0,23	1,22	0,22	0,99	-0,01
12	0,08	0,63	0,08	0,55	0,00
13	0,08	0,74	0,08	0,67	0,00
14	0,18	0,80	0,18	0,62	0,00
15	0,24	0,81	0,24	0,57	0,00
16	0,26	0,80	0,26	0,53	0,00
17	0,26	0,76	0,26	0,50	0,00
18	0,25	0,72	0,25	0,47	0,00
19	0,23	0,34	0,23	0,11	0,00
20	0,20	0,20	0,20	0,00	0,00
21	0,16	0,16	0,16	0,00	0,00
22	0,12	0,12	0,12	0,00	0,00
23	0,08	0,08	0,08	0,00	0,00
24	0,04	0,04	0,04	0,00	0,00

4.2 Σενάριο 2ο Κατανομή στους κεντρικούς κόμβους

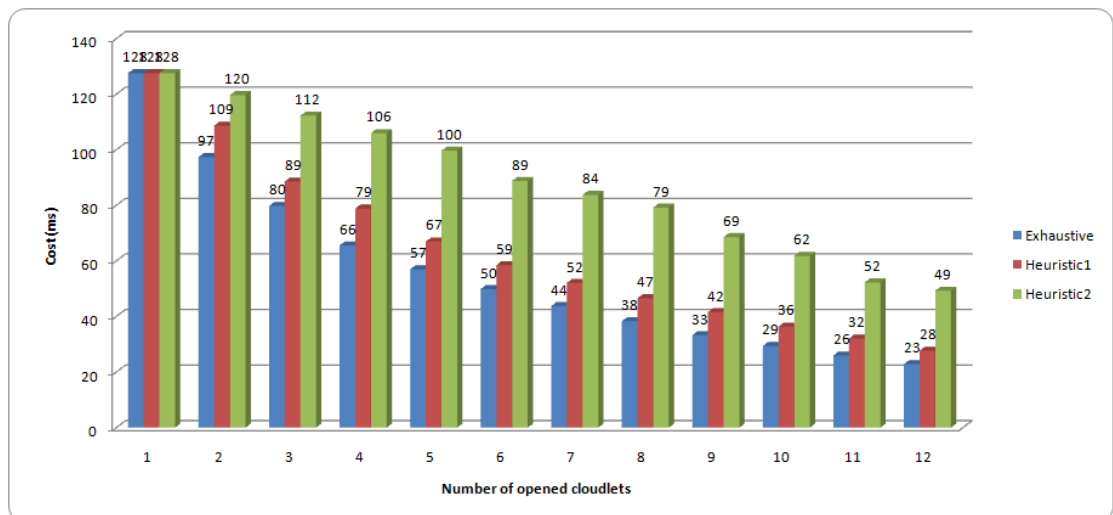
4.2.1 Τοπολογία scale free

Η μεγαλύτερη διαφορά στην απόδοση του πρώτου ευριστικού εντοπίζεται στη περίπτωση που έχουμε παρατηρείται τέσσερις service nodes, συνολικότερα παρατηρείται μια αύξηση ύψους 14,62% (0,7ms ανά χρήστη) στην συνολική καθυστέρηση του δικτύου.

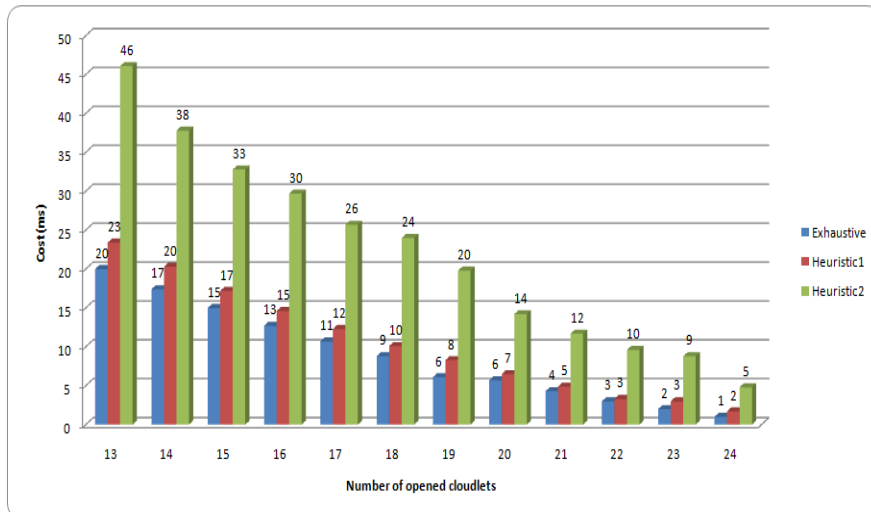
Σχετικά με την διακύμανση παρουσιάζεται αυξημένη εικόνα, η οποία κατά μεγάλο βαθμό εντοπίζεται στην περίπτωση που απαιτούνται δυο service nodes. Από τους έξι service nodes και μετά οι διαφορές μειώνονται κάτω από μία μονάδα. Συγκεκριμένα παρατηρείται αύξηση 124,80%, 4,53 μονάδες έναντι 2,02 μονάδων του εξαντλητικού.

Ο δεύτερος ευριστικός παρουσιάζει σαφώς μεγαλύτερη αύξηση του κόστους (68,97%) στο σύνολο του δικτύου, παρόλα αυτά οδηγεί σε μείωση ύψους 90,61% στην κατανομή των χρηστών. Σε σχέση με τον πρώτο ευριστικό επιτυγχάνεται μείωση της διακύμανσης κατά 95,82%, με παράλληλη αύξηση του κόστους καθυστέρησης κατά 47,42%.

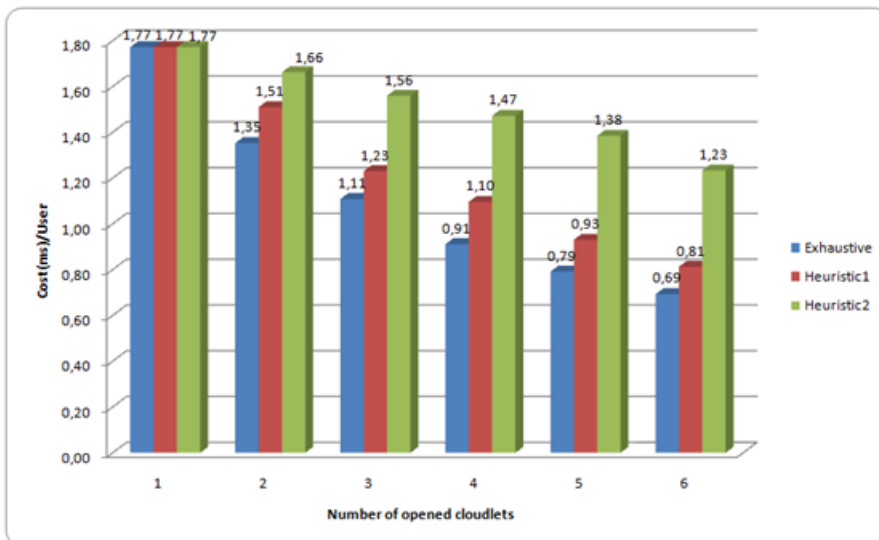
4.2.1.1 Γραφήματα



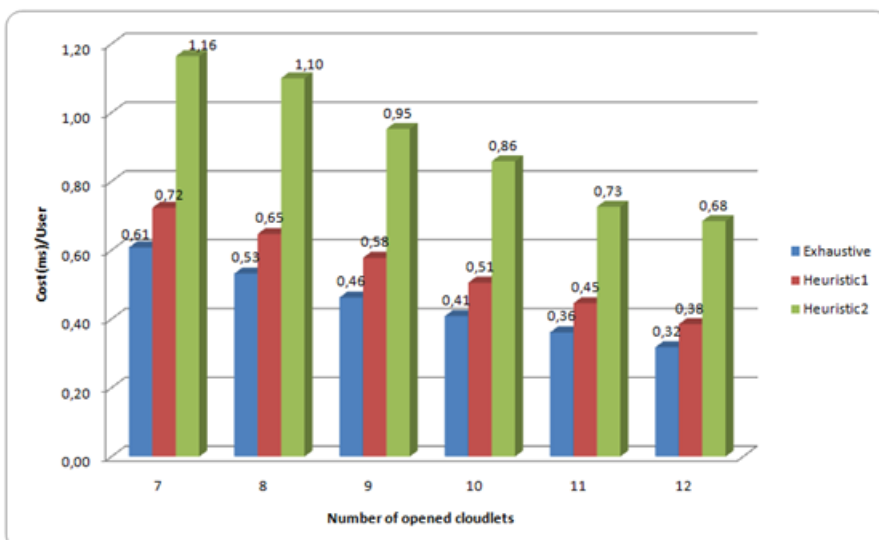
Εικόνα 35. Γράφημα κόστους σε τοπολογία scale free, central distribution



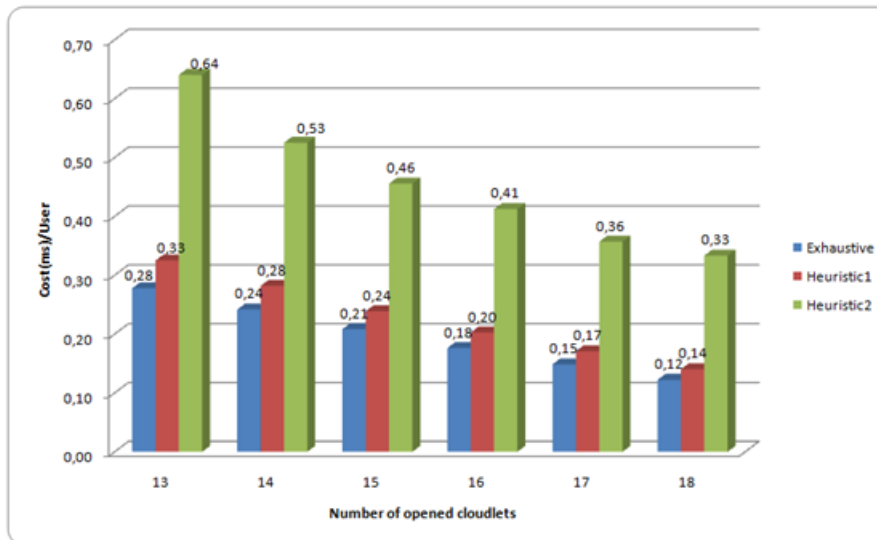
Εικόνα 36. Γράφημα κόστους σε τοπολογία scale free, central distribution



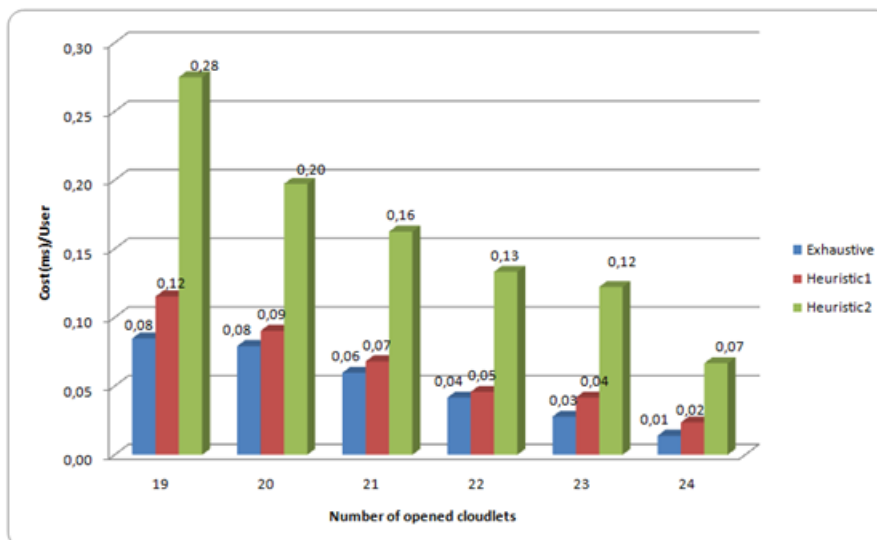
Εικόνα 37. Γράφημα κόστους ανά χρήστη σε τοπολογία scale free, central distribution



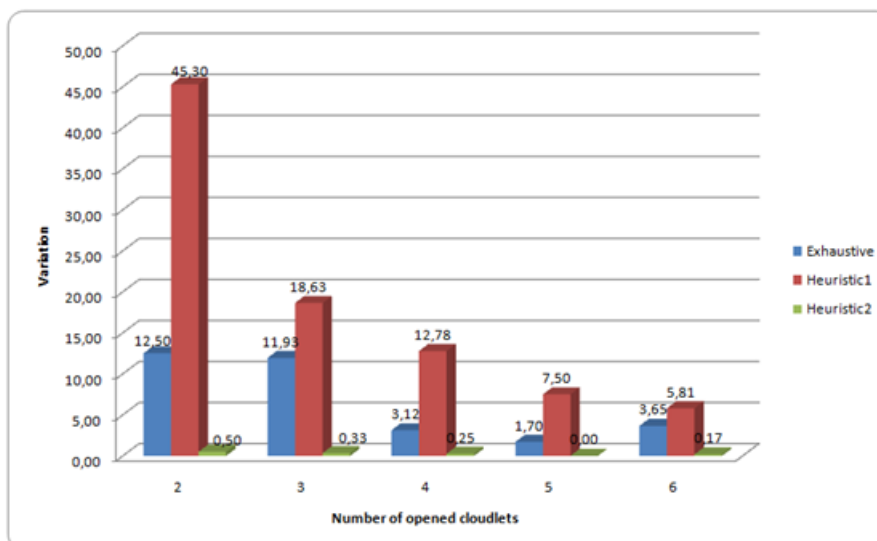
Εικόνα 38. Γράφημα κόστους ανά χρήστη σε τοπολογία scale free, central distribution



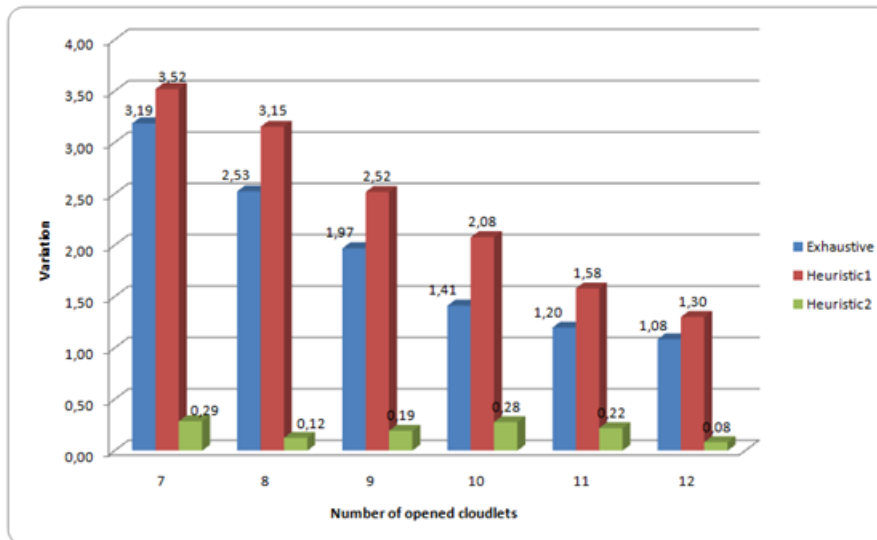
Εικόνα 39. Γράφημα κόστους ανά χρήστη σε τοπολογία scale free, central distribution



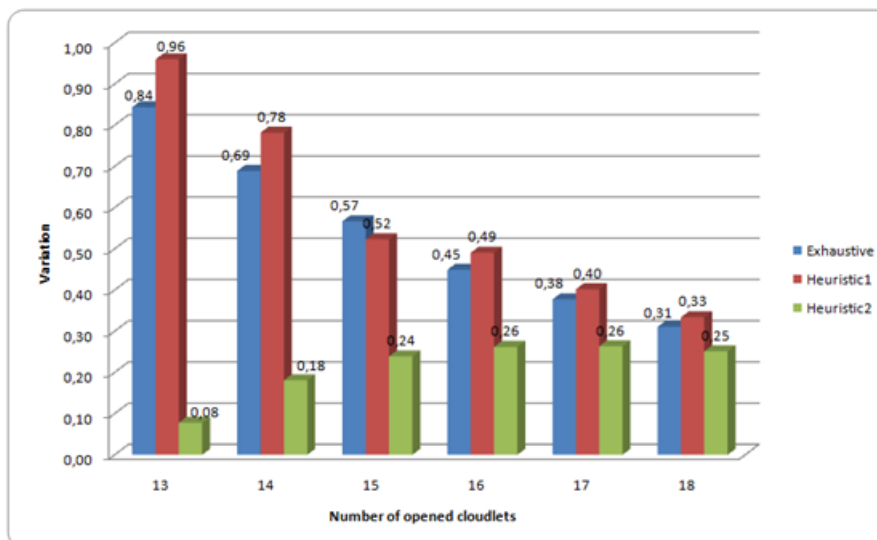
Εικόνα 40. Γράφημα κόστους ανά χρήστη σε τοπολογία scale free, central distribution



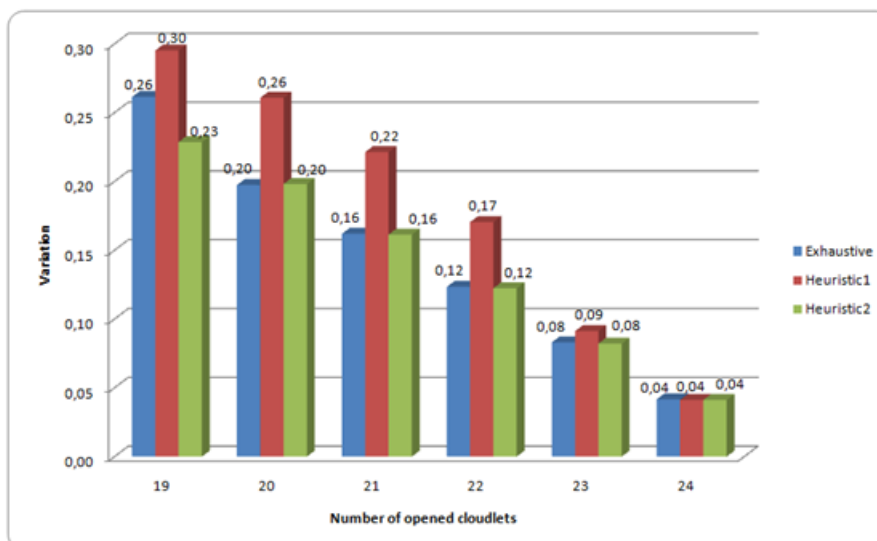
Εικόνα 41. Γράφημα διακύμανσης access node σε τοπολογία scale free, central distribution



Εικόνα 42. Γράφημα διακύμανσης access node σε τοπολογία scale free, central distribution



Εικόνα 43. Γράφημα διακύμανσης access node σε τοπολογία scale free, central distribution



Εικόνα 44. Γράφημα διακύμανσης access node σε τοπολογία scale free, central distribution

4.2.1.2 Συγκεντρωτικά αποτελέσματα

Πίνακας 15. Αποτελέσματα κόστους σε τοπολογία scale free, central distribution

# Nodes	Exh	Heur1	Heur2	Διαφορά heur1 (ms)	Διαφορά heur2 (ms)
1	128	128	128	0	0
2	97	109	120	11,3	22,3
3	80	89	112	8,8	32,5
4	66	79	106	13,3	40,4
5	57	67	100	10	42,7
6	50	59	89	8,6	38,9
7	44	52	84	8,3	40
8	38	47	79	8,3	40,9
9	33	42	69	8,3	35,3
10	29	36	62	7	32,4
11	26	32	52	6,1	26,3
12	23	28	49	4,8	26,4
13	20	23	46	3,4	26,1
14	17	20	38	2,9	20,4
15	15	17	33	2,2	17,8
16	13	15	30	1,9	17
17	11	12	26	1,6	15
18	9	10	24	1,3	15,2
19	6	8	20	2,2	13,7
20	6	7	14	0,8	8,5
21	4	5	12	0,6	7,4
22	3	3	10	0,3	6,6
23	2	3	9	1	6,8
24	1	2	5	0,7	3,8

Πίνακας 16. Αποτελέσματα κόστους ανά χρήστη σε τοπολογία scale free, central distribution

# Nodes	Exh	Heur1	Heur2	Διαφορά heur1 (ms)	Διαφορά heur2 (ms)
1	1,77	1,77	1,77	0,00	0,00
2	1,35	1,51	1,66	0,16	0,31
3	1,11	1,23	1,56	0,12	0,45
4	0,91	1,10	1,47	0,18	0,56
5	0,79	0,93	1,38	0,14	0,59
6	0,69	0,81	1,23	0,12	0,54
7	0,61	0,72	1,16	0,12	0,56
8	0,53	0,65	1,10	0,12	0,57
9	0,46	0,58	0,95	0,12	0,49
10	0,41	0,51	0,86	0,10	0,45
11	0,36	0,45	0,73	0,08	0,37
12	0,32	0,38	0,68	0,07	0,37
13	0,28	0,33	0,64	0,05	0,36
14	0,24	0,28	0,53	0,04	0,28
15	0,21	0,24	0,46	0,03	0,25
16	0,18	0,20	0,41	0,03	0,24
17	0,15	0,17	0,36	0,02	0,21
18	0,12	0,14	0,33	0,02	0,21
19	0,08	0,12	0,28	0,03	0,19
20	0,08	0,09	0,20	0,01	0,12
21	0,06	0,07	0,16	0,01	0,10
22	0,04	0,05	0,13	0,00	0,09
23	0,03	0,04	0,12	0,01	0,09
24	0,01	0,02	0,07	0,01	0,05

Πίνακας 17. Αποτελέσματα διακύμανσης σε τοπολογία scale free, central distribution

# Nodes	Exh	Heur1	Heur2	Διαφορά heur1	Διαφορά heur2
2	12,50	45,30	0,50	32,80	-12,00
3	11,93	18,63	0,33	6,70	-11,60
4	3,12	12,78	0,25	9,67	-2,87
5	1,70	7,50	0,00	5,80	-1,70
6	3,65	5,81	0,17	2,16	-3,48
7	3,19	3,52	0,29	0,33	-2,90
8	2,53	3,15	0,12	0,63	-2,40
9	1,97	2,52	0,19	0,55	-1,78
10	1,41	2,08	0,28	0,67	-1,13
11	1,20	1,58	0,22	0,38	-0,98
12	1,08	1,30	0,08	0,22	-1,00
13	0,84	0,96	0,08	0,12	-0,77
14	0,69	0,78	0,18	0,09	-0,51
15	0,57	0,52	0,24	-0,04	-0,33
16	0,45	0,49	0,26	0,04	-0,19
17	0,38	0,40	0,26	0,02	-0,11
18	0,31	0,33	0,25	0,02	-0,06
19	0,26	0,30	0,23	0,03	-0,03
20	0,20	0,26	0,20	0,06	0,00
21	0,16	0,22	0,16	0,06	0,00
22	0,12	0,17	0,12	0,05	0,00
23	0,08	0,09	0,08	0,01	0,00
24	0,04	0,04	0,04	0,00	0,00

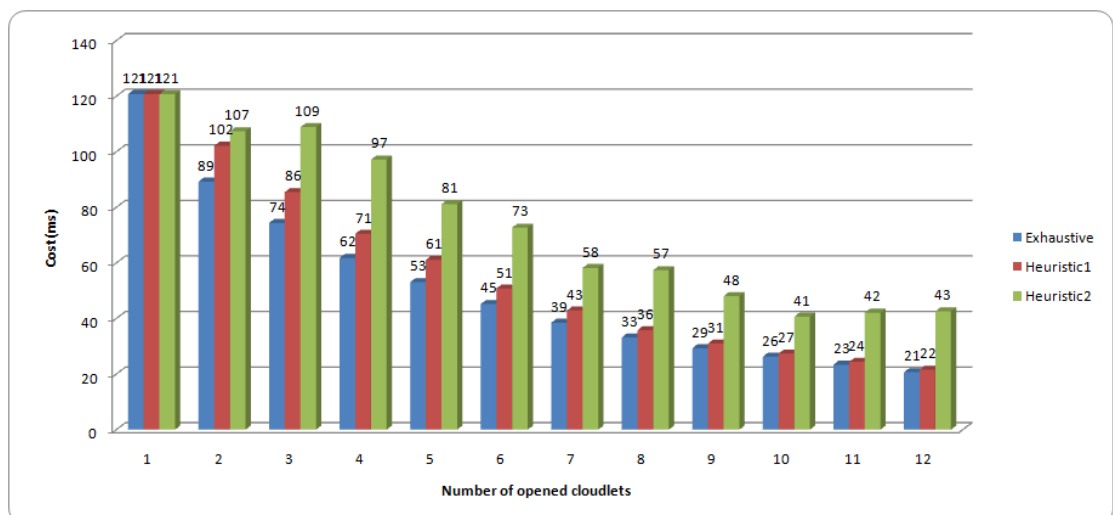
4.2.2 Τοπολογία lattice

Στην συγκεκριμένη τοπολογία ο πρώτος ευριστικός αλγόριθμος παρουσιάζει γενικότερη αύξηση 8,39% όσον αφορά το κόστος καθυστέρησης, η οποία εν πολλοίς οφείλεται στα σενάρια από δύο έως τέσσερις service nodes, έπειτα βελτιώνεται αισθητά η ακρίβεια του αλγορίθμου.

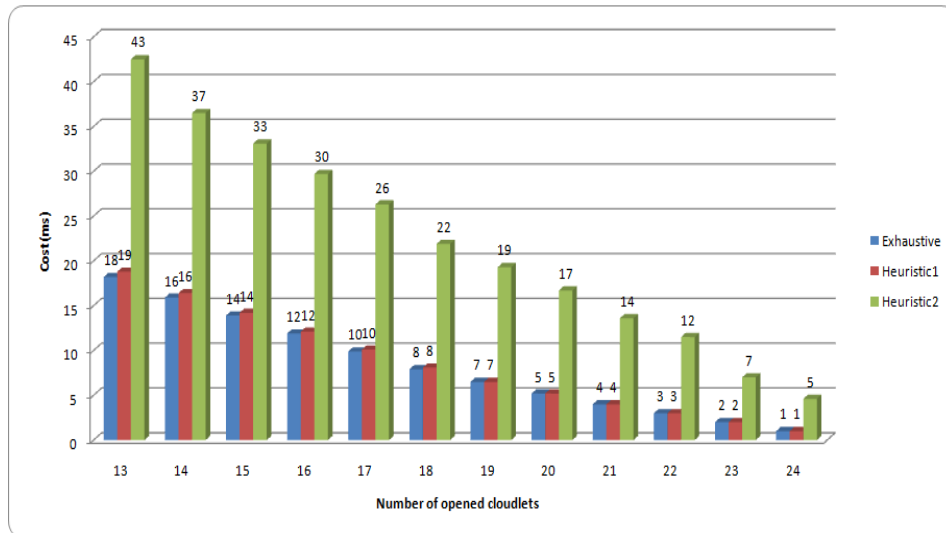
Στα ίδια σενάρια παρατηρείται ιδιαίτερως αυξημένη διακύμανση, η οποία οφείλεται για το 81% της συνολικής αύξησης στην διακύμανση. Η συνολική αύξηση είναι ίση με 182,06%.

Ο δεύτερος ευριστικός αλγόριθμος επιτυγχάνει μείωση της διακύμανσης κατά 73,99% με παράλληλη αύξηση του κόστους καθυστέρησης κατά 59,35%, σε σχέση με τα αποτελέσματα του εξαντλητικού. Επίσης επιτυγχάνεται μείωση 90,78% σε σχέση με την διακύμανση του πρώτου ευριστικού αλγορίθμου, με αντίστοιχη αύξηση του κόστους καθυστέρησης κατά 41,02%.

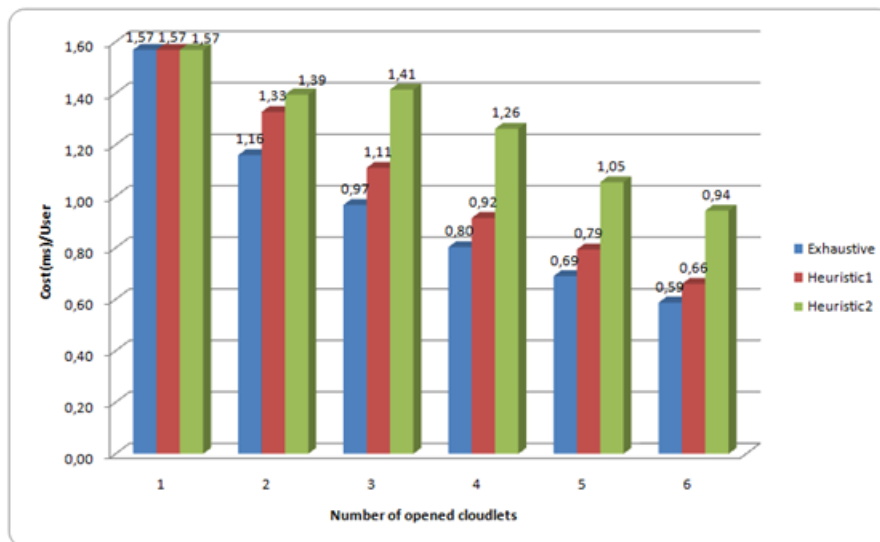
4.2.2.1 Γραφήματα



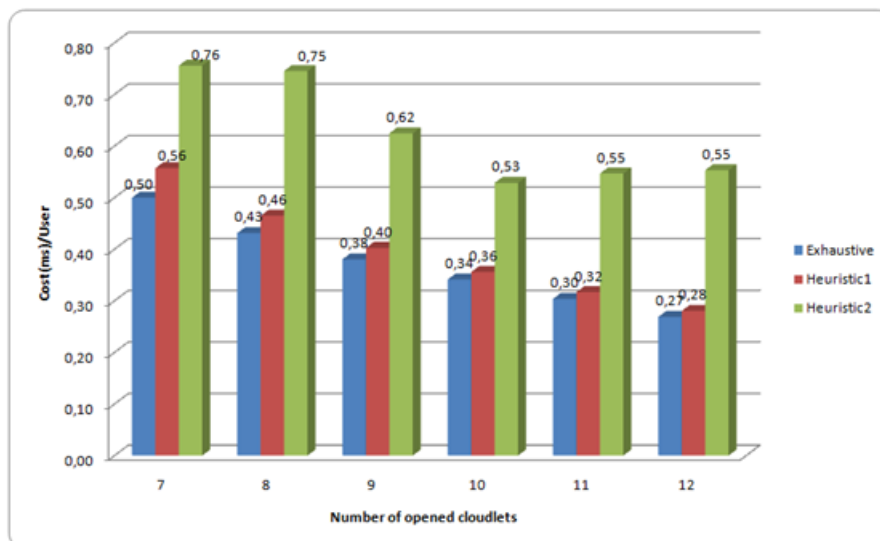
Εικόνα 45. Γράφημα κόστους σε τοπολογία lattice, central distribution



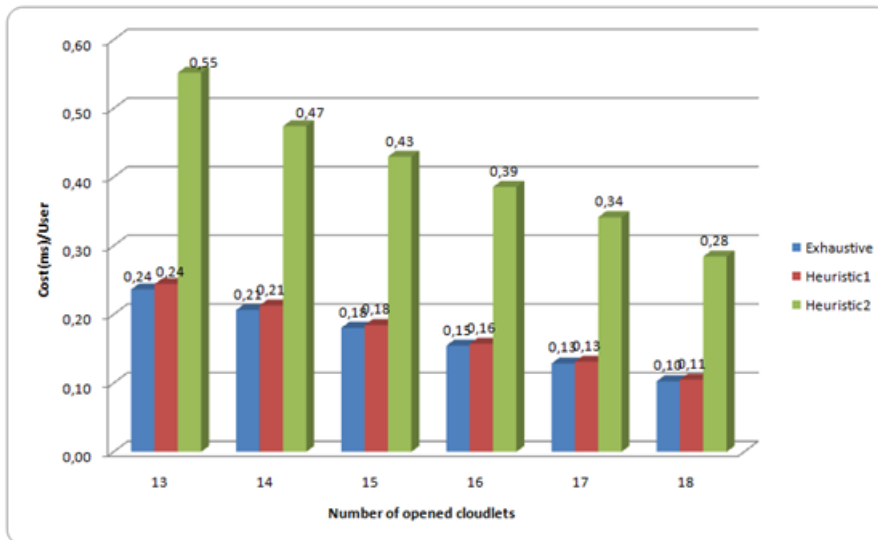
Εικόνα 46. Γράφημα κόστους σε τοπολογία lattice, central distribution



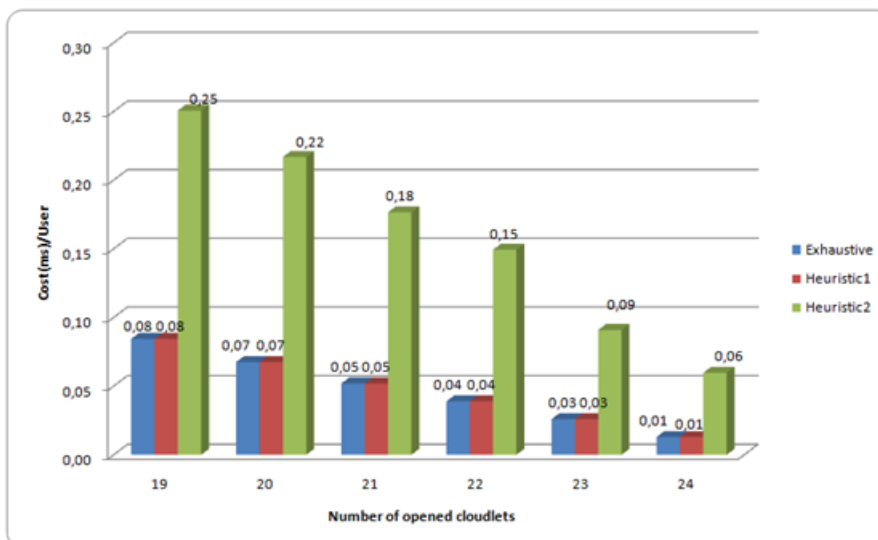
Εικόνα 47. Γράφημα κόστους ανά χρήστη σε τοπολογία lattice, central distribution



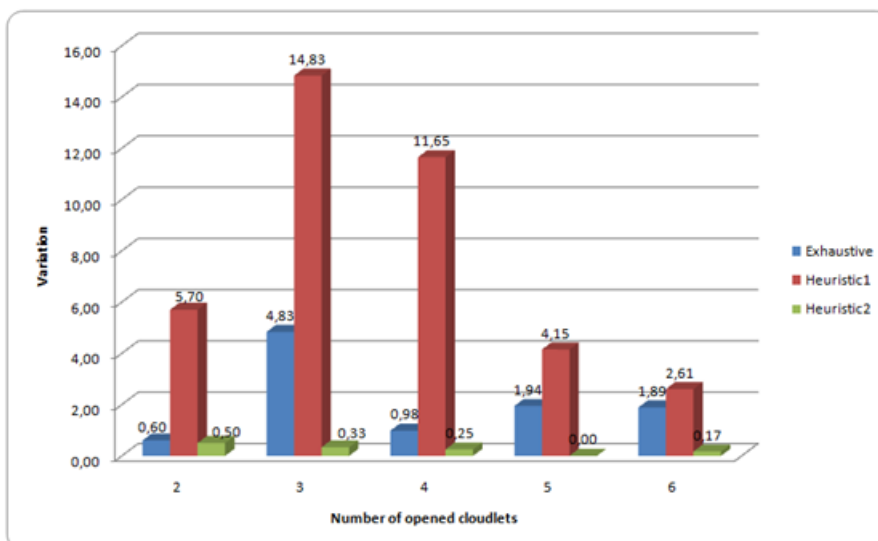
Εικόνα 48. Γράφημα κόστους ανά χρήστη σε τοπολογία lattice, central distribution



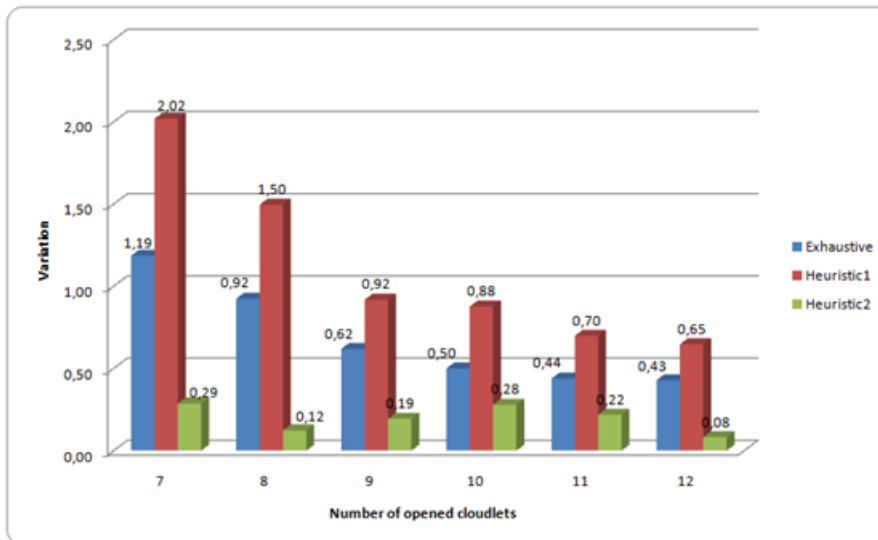
Εικόνα 49. Γράφημα κόστους ανά χρήστη σε τοπολογία lattice, central distribution



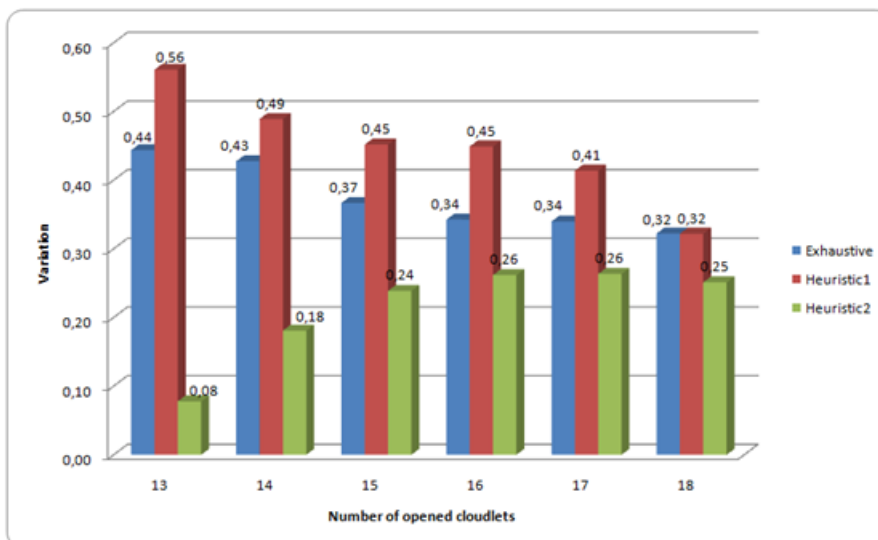
Εικόνα 50. Γράφημα κόστους ανά χρήστη σε τοπολογία lattice, central distribution



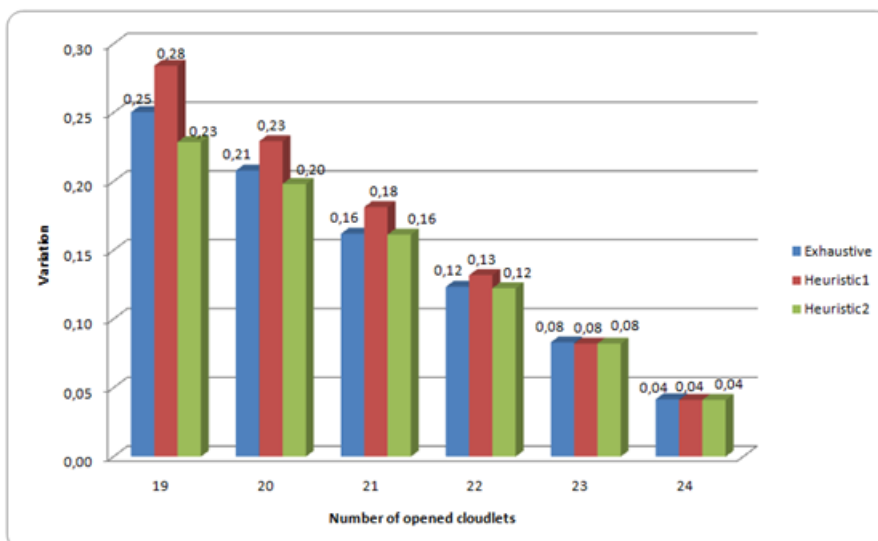
Εικόνα 51. Γράφημα διακύμανσης access node σε τοπολογία lattice, central distribution



Εικόνα 52. Γράφημα διακύμανσης access node σε τοπολογία lattice, central distribution



Εικόνα 53. Γράφημα διακύμανσης access node σε τοπολογία lattice, central distribution



Εικόνα 54. Γράφημα διακύμανσης access node σε τοπολογία lattice, central distribution

4.2.2.2 Συγκεντρωτικά αποτελέσματα

Πίνακας 18. Αποτελέσματα κόστους σε τοπολογία lattice, central distribution

# Nodes	Exh	Heur1	Heur2	Διαφορά heur1 (ms)	Διαφορά heur2 (ms)
1	121	121	121	0	0
2	89	102	107	13	18
3	74	86	109	11	35
4	62	71	97	9	35
5	53	61	81	8	28
6	45	51	73	6	28
7	39	43	58	4	20
8	33	36	57	3	24
9	29	31	48	2	19
10	26	27	41	1	14
11	23	24	42	1	19
12	21	22	43	1	22
13	18	19	43	1	24
14	16	16	37	0	21
15	14	14	33	0	19
16	12	12	30	0	18
17	10	10	26	0	16
18	8	8	22	0	14
19	7	7	19	0	13
20	5	5	17	0	12
21	4	4	14	0	10
22	3	3	12	0	9
23	2	2	7	0	5
24	1	1	5	0	4

Πίνακας 19. Αποτελέσματα κόστους ανά χρήση σε τοπολογία lattice, central distribution

# Nodes	Exh	Heur1	Heur2	Διαφορά heur1 (ms)	Διαφορά heur2 (ms)
1	1,57	1,57	1,57	0,00	0,00
2	1,16	1,33	1,39	0,17	0,24
3	0,97	1,11	1,41	0,14	0,45
4	0,80	0,92	1,26	0,11	0,46
5	0,69	0,79	1,05	0,10	0,36
6	0,59	0,66	0,94	0,07	0,36
7	0,50	0,56	0,76	0,06	0,26
8	0,43	0,46	0,75	0,03	0,31
9	0,38	0,40	0,62	0,02	0,24
10	0,34	0,36	0,53	0,01	0,19
11	0,30	0,32	0,55	0,01	0,24
12	0,27	0,28	0,55	0,01	0,28
13	0,24	0,24	0,55	0,01	0,32
14	0,21	0,21	0,47	0,01	0,27
15	0,18	0,18	0,43	0,00	0,25
16	0,15	0,16	0,39	0,00	0,23
17	0,13	0,13	0,34	0,00	0,21
18	0,10	0,11	0,28	0,00	0,18
19	0,08	0,08	0,25	0,00	0,17
20	0,07	0,07	0,22	0,00	0,15
21	0,05	0,05	0,18	0,00	0,12
22	0,04	0,04	0,15	0,00	0,11
23	0,03	0,03	0,09	0,00	0,06
24	0,01	0,01	0,06	0,00	0,05

Πίνακας 20. Αποτελέσματα διακύμανσης σε τοπολογία lattice, central distribution

# Nodes	Exh	Heur1	Heur2	Διαφορά heur1	Διαφορά heur2
2	0,60	5,70	0,50	5,10	-0,10
3	4,83	14,83	0,33	10,00	-4,50
4	0,98	11,65	0,25	10,67	-0,73
5	1,94	4,15	0,00	2,21	-1,94
6	1,89	2,61	0,17	0,72	-1,72
7	1,19	2,02	0,29	0,83	-0,90
8	0,92	1,50	0,12	0,57	-0,80
9	0,62	0,92	0,19	0,30	-0,43
10	0,50	0,88	0,28	0,38	-0,22
11	0,44	0,70	0,22	0,26	-0,22
12	0,43	0,65	0,08	0,22	-0,35
13	0,44	0,56	0,08	0,12	-0,37
14	0,43	0,49	0,18	0,06	-0,25
15	0,37	0,45	0,24	0,09	-0,13
16	0,34	0,45	0,26	0,11	-0,08
17	0,34	0,41	0,26	0,07	-0,08
18	0,32	0,32	0,25	0,00	-0,07
19	0,25	0,28	0,23	0,03	-0,02
20	0,21	0,23	0,20	0,02	-0,01
21	0,16	0,18	0,16	0,02	0,00
22	0,12	0,13	0,12	0,01	0,00
23	0,08	0,08	0,08	0,00	0,00
24	0,04	0,04	0,04	0,00	0,00

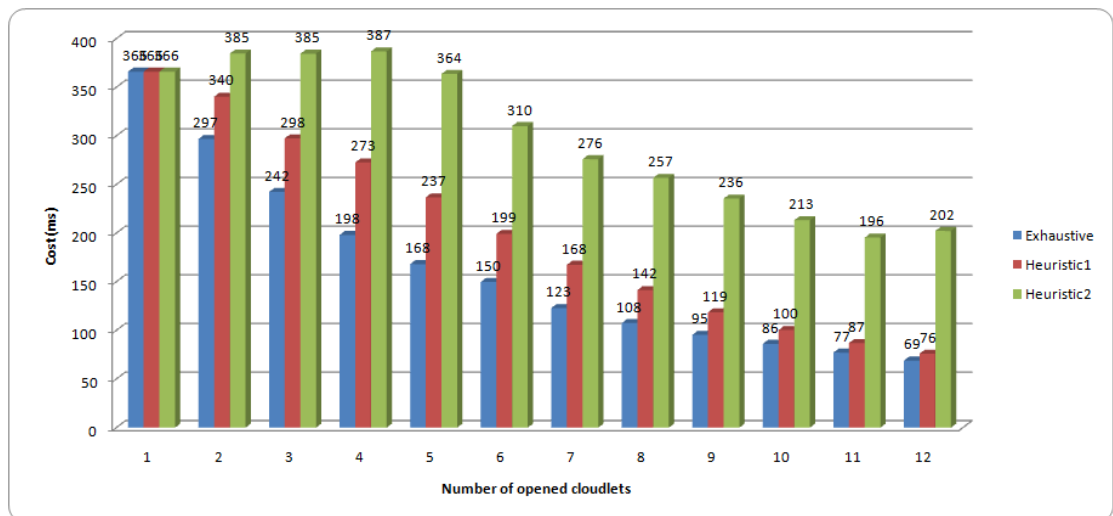
4.3 Σενάριο 3^ο Κατανομή στους περιφερειακούς κόμβους

4.3.1 Τοπολογία scale free

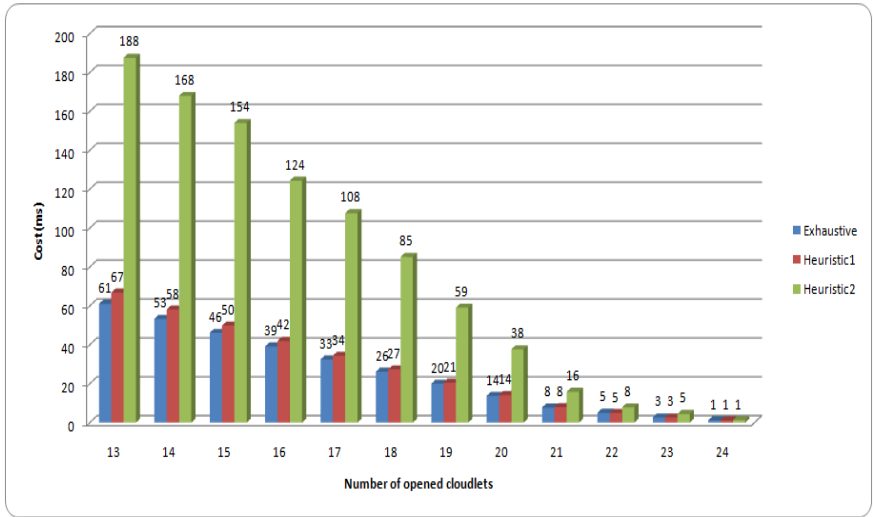
Το συγκεκριμένο σενάριο κατανομής βαρών, παρατηρήθηκε αύξηση του κόστους καθυστέρησης κατά 19,46% συνολικά. Η μεγαλύτερη διαφορά παρουσιάζεται στη περίπτωση των πέντε service nodes, όπου το αποτέλεσμα του ευριστικού επιφέρει αύξηση της τάξης του 41,17%, 1,44ms ανά χρήστη έναντι 1,02ms ανά χρήστη του εξαντλητικού. Από τους οκτώ service nodes και μετά η απόδοση του αλγορίθμου βελτιώνεται αισθητά. Ομοίως και η διακύμανση, μειώνεται αισθητά η διαφορά της έπειτα από τους οκτώ service nodes. Συνολικά παρατηρείται αύξηση κατά 74,32%, 9,64 μονάδες έναντι 5,53 μονάδων του εξαντλητικού.

Το κόστος ανά χρήστη βάσει των επιλογών του δεύτερου αλγορίθμου υπολογίζεται σε 1,14ms, παρατηρείται αύξηση κατά 0,57ms ανά χρήστη (97,85%). Η διακύμανση παρουσιάζει μείωση κατά 5,53 μονάδες (96,8%). Συγκριτικά με τον πρώτο ευριστικό παρουσιάζει μείωση 98,04%, με αύξηση του κόστους καθυστέρησης κατά 65,63%.

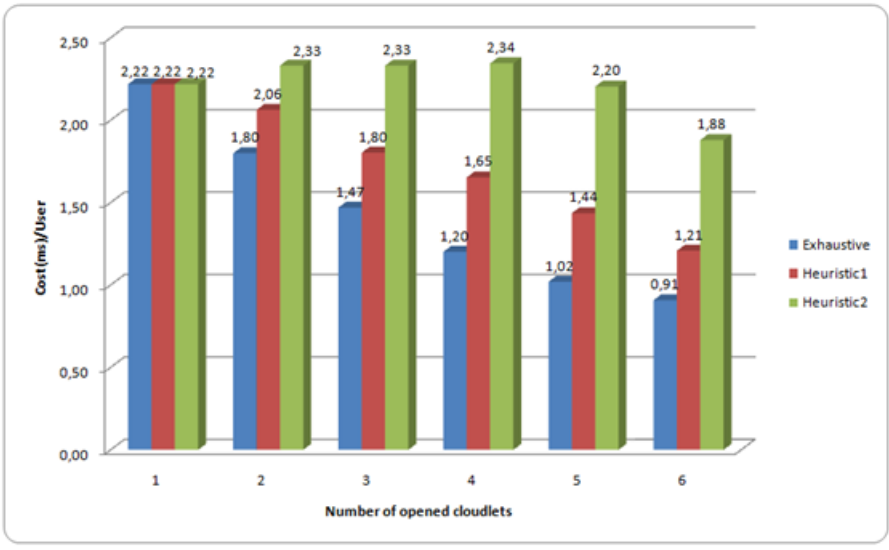
4.3.1.1 Γραφήματα



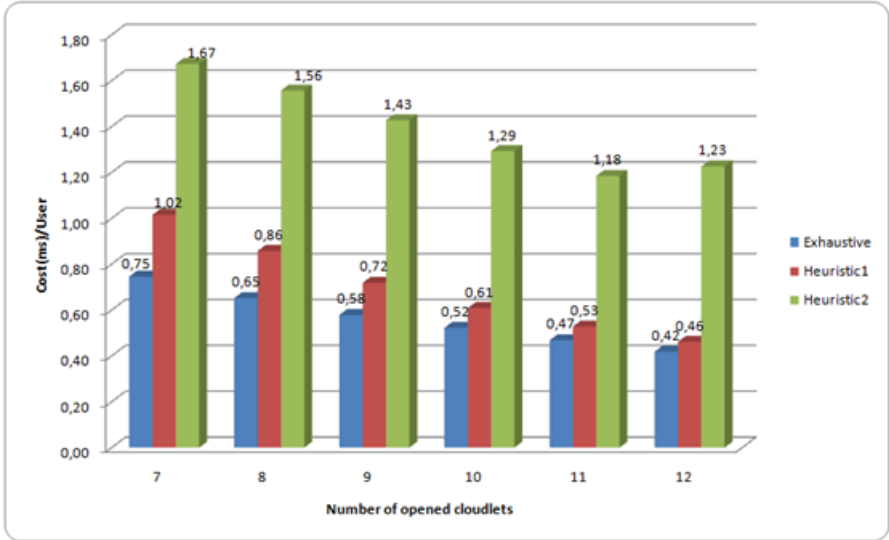
Εικόνα 55. Γράφημα κόστους σε τοπολογία scale free, edge distribution



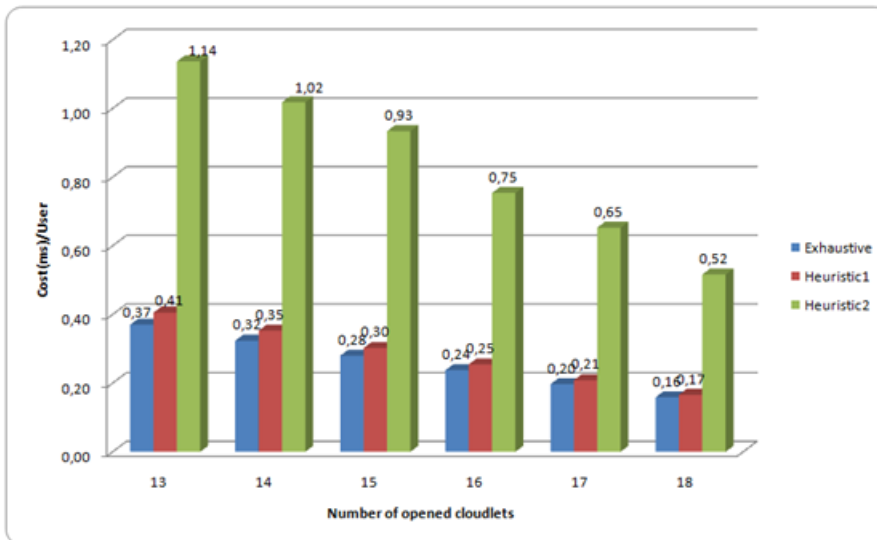
Εικόνα 56. Γράφημα κόστους σε τοπολογία scale free, edge distribution



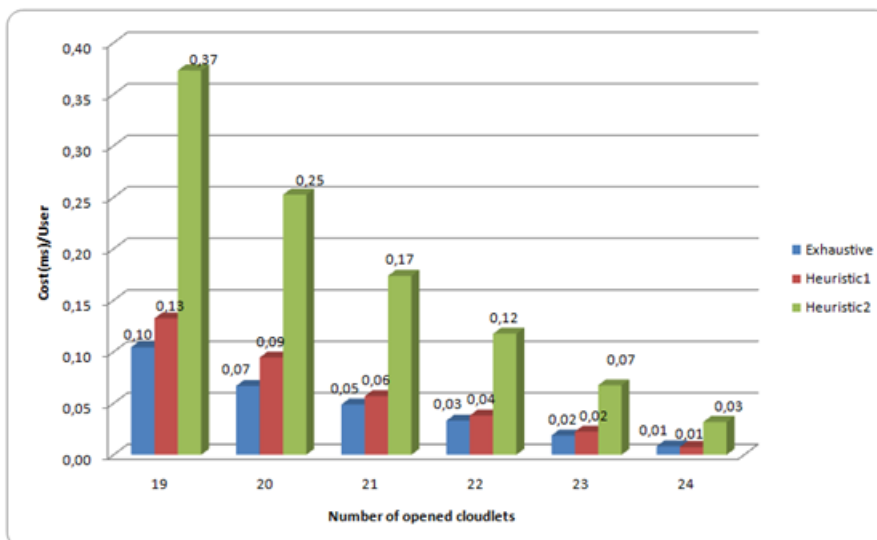
Εικόνα 57. Γράφημα κόστους ανά χρήστη σε τοπολογία scale free, edge distribution



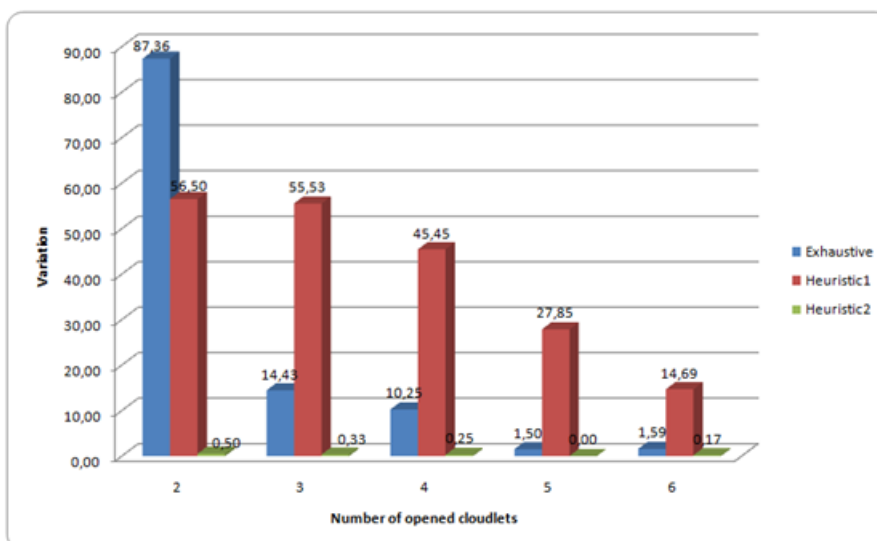
Εικόνα 58. Γράφημα κόστους ανά χρήστη σε τοπολογία scale free, edge distribution



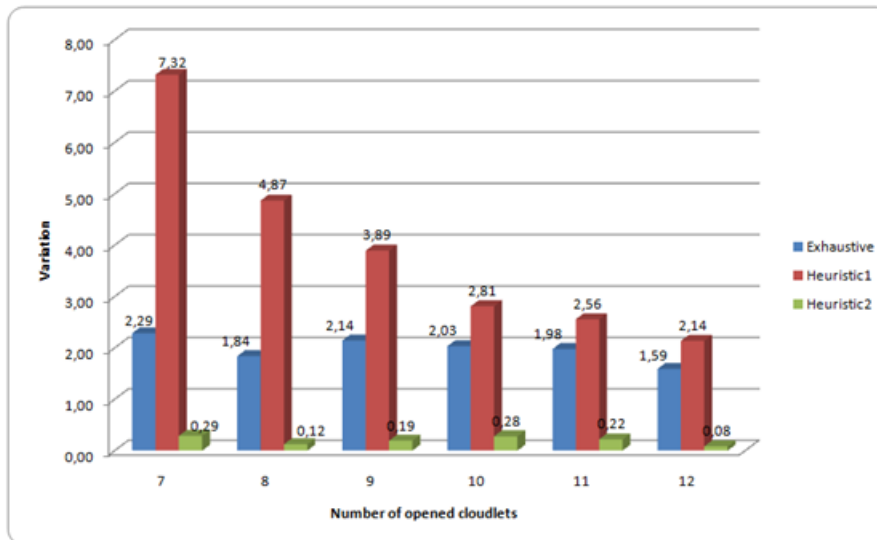
Εικόνα 59. Γράφημα κόστους ανά χρήστη σε τοπολογία scale free, edge distribution



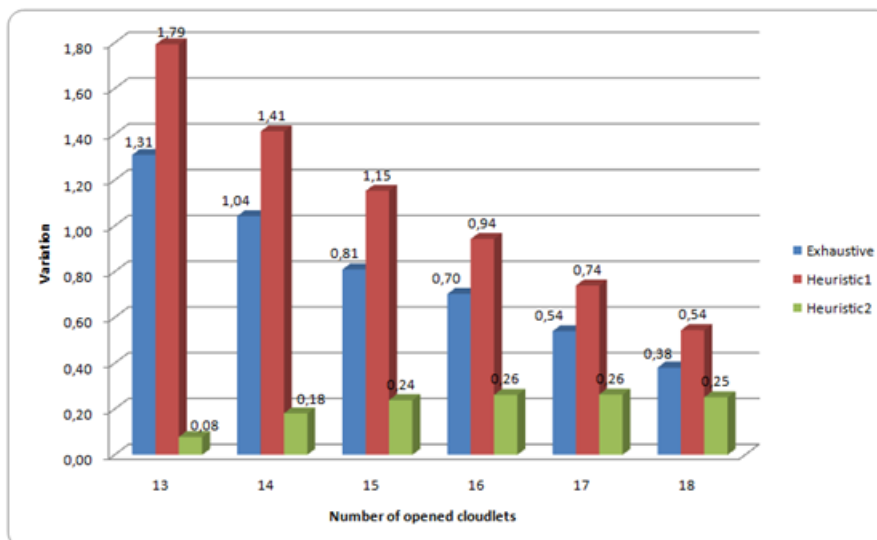
Εικόνα 60. Γράφημα κόστους ανά χρήστη σε τοπολογία scale free, edge distribution



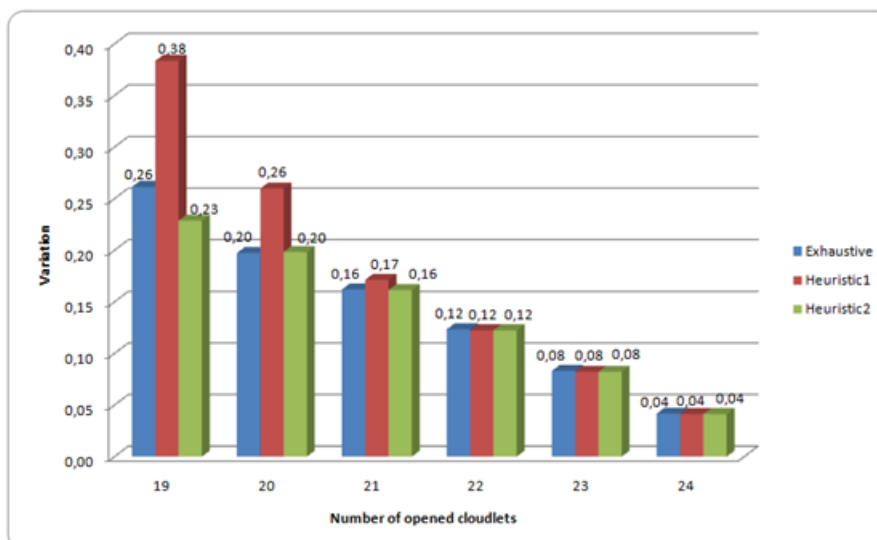
Εικόνα 61. Γράφημα διακύμανσης access node σε τοπολογία scale free, edge distribution



Εικόνα 62. Γράφημα διακύμανσης access node σε τοπολογία scale free, edge distribution



Εικόνα 63. Γράφημα διακύμανσης access node σε τοπολογία scale free, edge distribution



Εικόνα 64. Γράφημα διακύμανσης access node σε τοπολογία scale free, edge distribution

4.3.1.2 Συγκεντρωτικά αποτελέσματα

Πίνακας 21. Αποτελέσματα κόστους σε τοπολογία scale free, edge distribution

# Nodes	Exh	Heur1	Heur2	Διαφορά heur1 (ms)	Διαφορά heur2 (ms)
1	366	366	366	0	0
2	297	340	385	44	88
3	242	298	385	55	142
4	198	273	387	75	189
5	168	237	364	69	196
6	150	199	310	50	160
7	123	168	276	45	153
8	108	142	257	34	149
9	95	119	236	23	140
10	86	100	213	14	127
11	77	87	196	10	118
12	69	76	202	7	133
13	61	67	188	6	127
14	53	58	168	5	115
15	46	50	154	4	108
16	39	42	124	3	85
17	33	34	108	2	75
18	26	27	85	1	59
19	20	21	59	1	39
20	14	14	38	0	24
21	8	8	16	0	8
22	5	5	8	0	3
23	3	3	5	0	2
24	1	1	1	0	0

Πίνακας 22. Αποτελέσματα κόστους ανά χρήστη σε τοπολογία scale free, edge distribution

# Nodes	Exh	Heur1	Heur2	Διαφορά heur1 (ms)	Διαφορά heur2 (ms)
1	2,22	2,22	2,22	0,00	0,00
2	1,80	2,06	2,33	0,26	0,53
3	1,47	1,80	2,33	0,33	0,86
4	1,20	1,65	2,34	0,45	1,14
5	1,02	1,44	2,20	0,42	1,18
6	0,91	1,21	1,88	0,30	0,97
7	0,75	1,02	1,67	0,27	0,93
8	0,65	0,86	1,56	0,21	0,90
9	0,58	0,72	1,43	0,14	0,85
10	0,52	0,61	1,29	0,09	0,77
11	0,47	0,53	1,18	0,06	0,72
12	0,42	0,46	1,23	0,04	0,81
13	0,37	0,41	1,14	0,04	0,77
14	0,32	0,35	1,02	0,03	0,69
15	0,28	0,30	0,93	0,02	0,65
16	0,24	0,25	0,75	0,02	0,52
17	0,20	0,21	0,65	0,01	0,46
18	0,16	0,17	0,52	0,01	0,36
19	0,12	0,12	0,36	0,00	0,24
20	0,08	0,09	0,23	0,00	0,14
21	0,05	0,05	0,10	0,00	0,05
22	0,03	0,03	0,05	0,00	0,02
23	0,02	0,02	0,03	0,00	0,01
24	0,01	0,01	0,01	0,00	0,00

Πίνακας 23. Αποτελέσματα διακύμανσης σε τοπολογία scale free, edge distribution

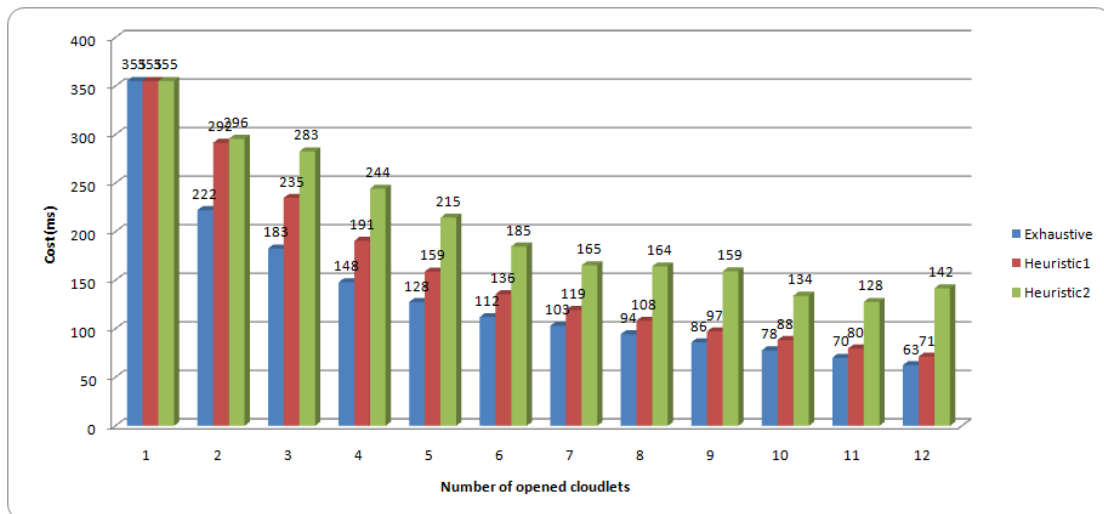
# Nodes	Exh	Heur1	Heur2	Διαφορά heur1	Διαφορά heur2
2	87,36	56,50	0,50	-30,86	-86,86
3	14,43	55,53	0,33	41,10	-14,10
4	10,25	45,45	0,25	35,20	-10,00
5	1,50	27,85	0,00	26,35	-1,50
6	1,59	14,69	0,17	13,10	-1,42
7	2,29	7,32	0,29	5,03	-2,00
8	1,84	4,87	0,12	3,03	-1,72
9	2,14	3,89	0,19	1,75	-1,95
10	2,03	2,81	0,28	0,78	-1,75
11	1,98	2,56	0,22	0,58	-1,76
12	1,59	2,14	0,08	0,55	-1,51
13	1,31	1,79	0,08	0,48	-1,23
14	1,04	1,41	0,18	0,37	-0,86
15	0,81	1,15	0,24	0,34	-0,57
16	0,70	0,94	0,26	0,24	-0,44
17	0,54	0,74	0,26	0,20	-0,28
18	0,38	0,54	0,25	0,16	-0,13
19	0,26	0,38	0,23	0,12	-0,03
20	0,20	0,26	0,20	0,06	0,00
21	0,16	0,17	0,16	0,01	0,00
22	0,12	0,12	0,12	0,00	0,00
23	0,08	0,08	0,08	0,00	0,00
24	0,04	0,04	0,04	0,00	0,00

4.3.2 Τοπολογία lattice

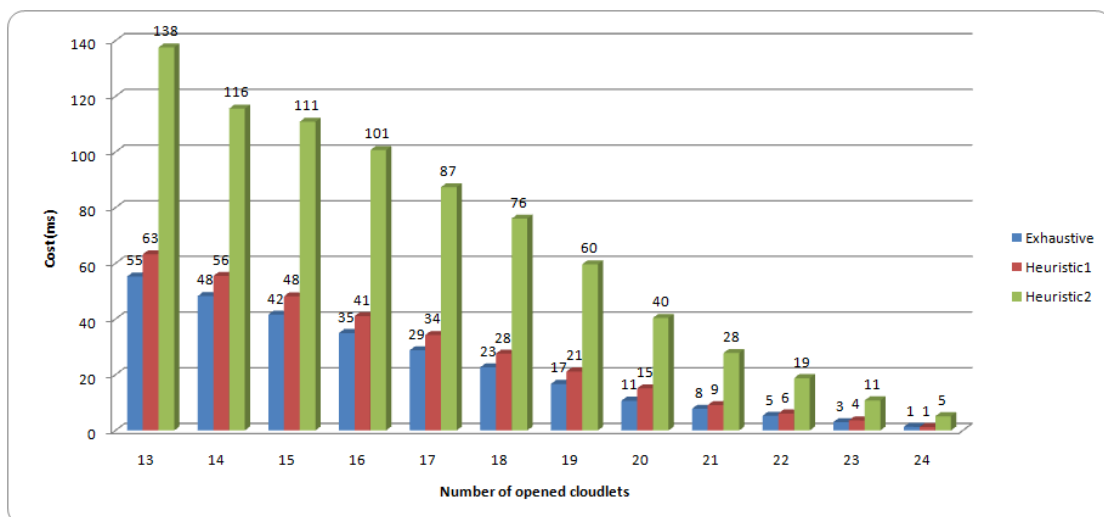
Στην συγκεκριμένη τοπολογία παρατηρείται μια διαφορά της τάξης του 17,79%, όσον αφορά την καθυστέρηση του δικτύου, στα αποτελέσματα του πρώτου ευριστικού συγκριτικά με τα αντίστοιχα αποτελέσματα του εξαντλητικού αλγορίθμου. Οι μεγαλύτερες διαφορές παρουσιάζονται στις περιπτώσεις που έχουμε δυο, τρεις και τέσσερις services nodes, από εκείνο το σημείο και μετά παρουσιάζεται μείωση των διαφορών. Όσον αφορά την διακύμανση παρατηρούνται μεγάλες διαφορές, κυρίες για τις περιπτώσεις που απαιτούνται τρεις, τέσσερις και πέντε service nodes, όπου υπάρχει μια μέση διαφορά της τάξης των 6,5 μονάδων. Συγκεκριμένα έχουμε αύξηση κατά 98,09%, σε 2,05 μονάδες, ενώ η διακύμανση του εξαντλητικού ανέρχεται σε 1,01 μονάδα.

Για τον δεύτερο ευριστικό, παρατηρείται αύξηση του κόστους καθυστέρησης κατά 0,4ms ανά χρήστη (60,5%), σε σχέση με τον εξαντλητικό. Παράλληλα μειώνεται η διακύμανση κατά 77,16%. Σε σχέση με τον πρώτο, ο δεύτερος ευριστικός επιτυγχάνει βελτίωση της διακύμανσης κατά 86,61%, και επιφέρει αύξηση στο κόστος καθυστέρησης ανά χρήστη, ίση με (45,65%).

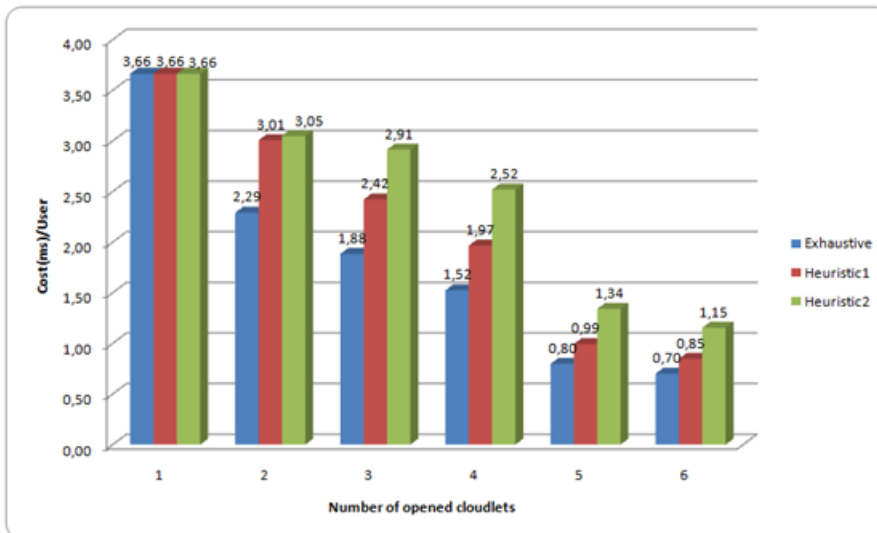
4.3.2.1 Γραφήματα



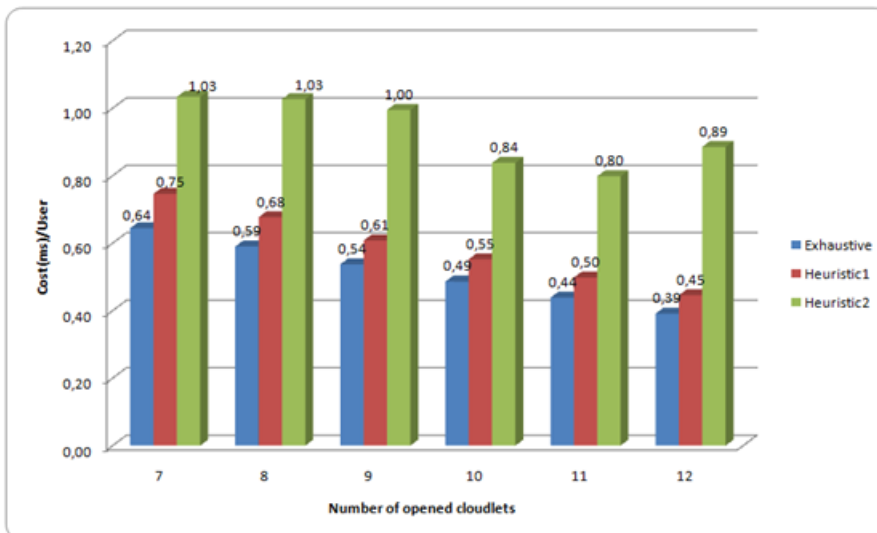
Εικόνα 65. Γράφημα κόστους σε τοπολογία lattice, edge distribution



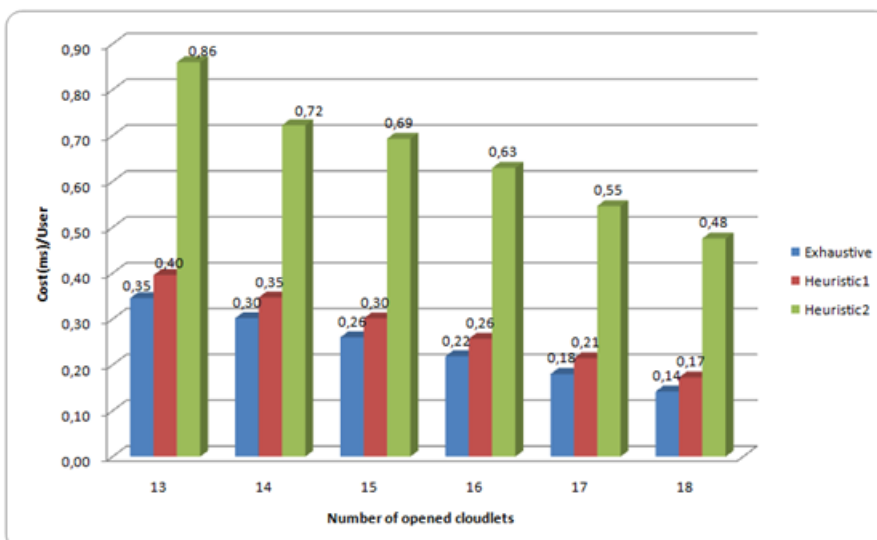
Εικόνα 66. Γράφημα κόστους σε τοπολογία lattice, edge distribution



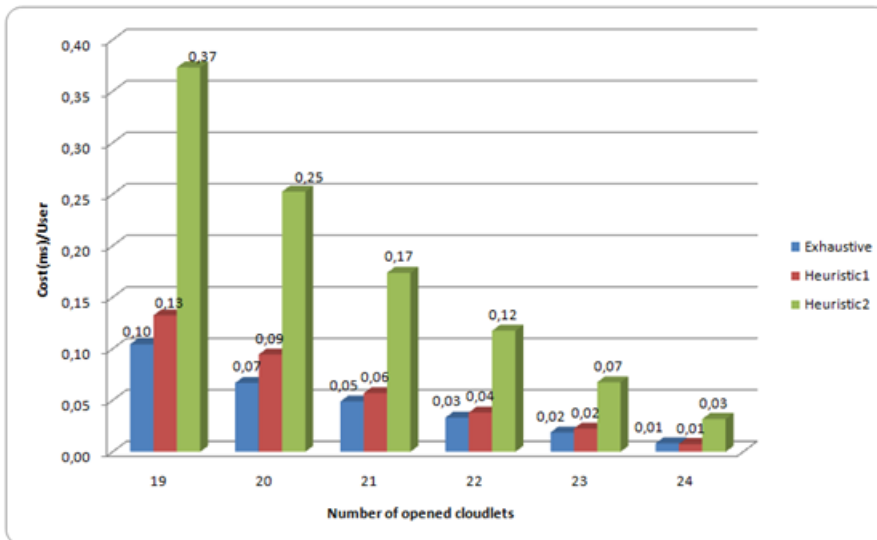
Εικόνα 67. Γράφημα κόστους ανά χρήστη σε τοπολογία lattice, edge distribution



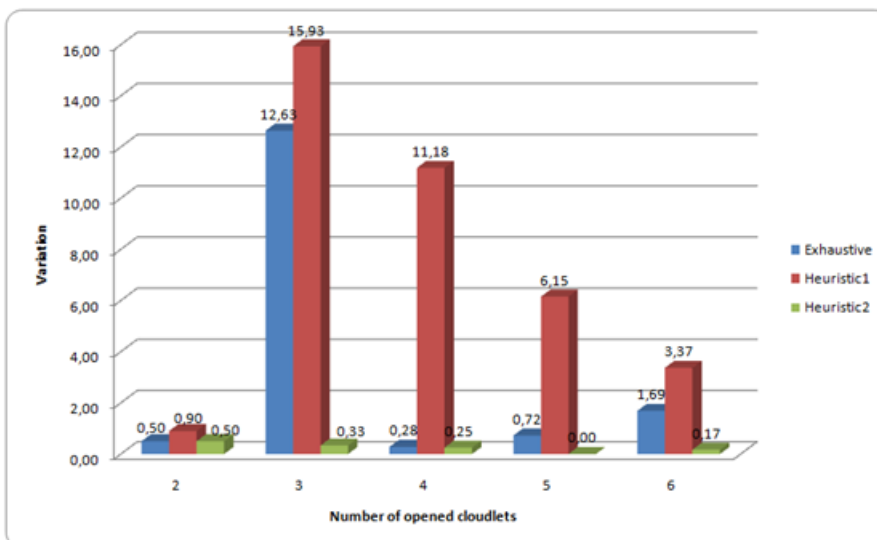
Εικόνα 68. Γράφημα κόστους ανά χρήστη σε τοπολογία lattice, edge distribution



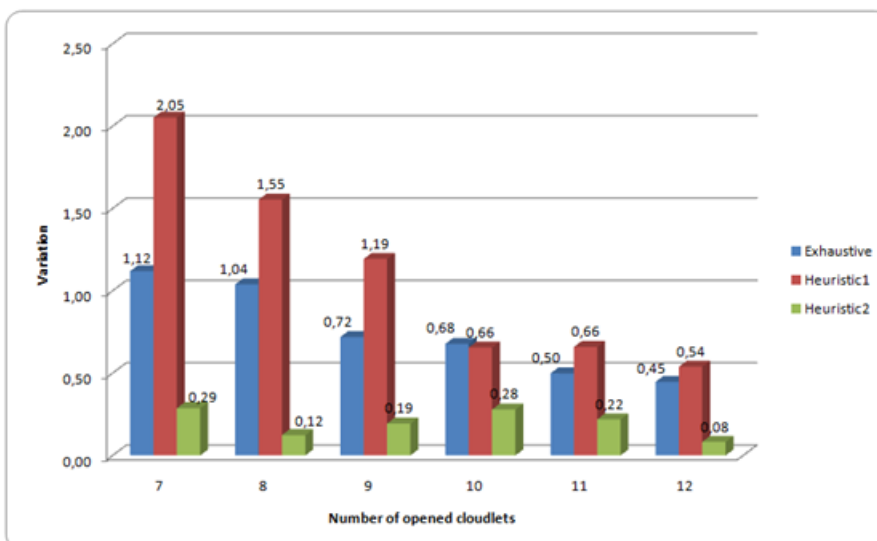
Εικόνα 69. Γράφημα κόστους ανά χρήστη σε τοπολογία lattice, edge distribution



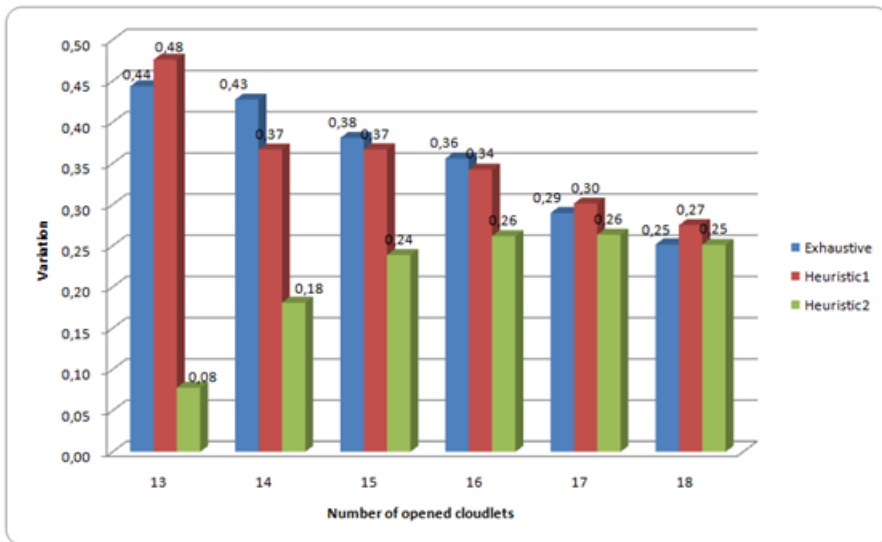
Εικόνα 70. Γράφημα κόστους ανά χρήστη σε τοπολογία lattice, edge distribution



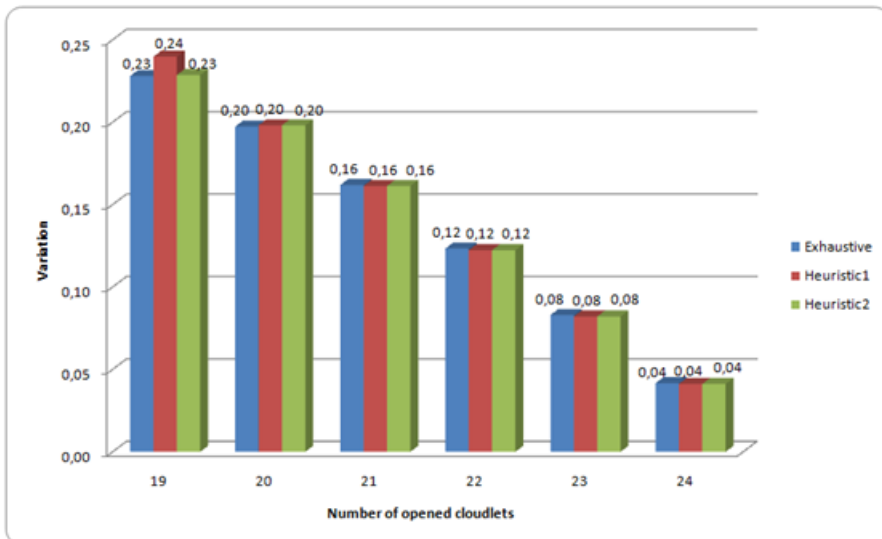
Εικόνα 71. Γράφημα διακύμανσης access node σε τοπολογία lattice, edge distribution



Εικόνα 72. Γράφημα διακύμανσης access node σε τοπολογία lattice, edge distribution



Εικόνα 73. Γράφημα διακύμανσης access node σε τοπολογία lattice, edge distribution



Εικόνα 74. Γράφημα διακύμανσης access node σε τοπολογία lattice, edge distribution

4.3.2.2 Συγκεντρωτικά Αποτελέσματα

Πίνακας 24. Αποτελέσματα κόστους σε τοπολογία lattice, edge distribution

# Nodes	Exh	Heur1	Heur2	Διαφορά heur1 (ms)	Διαφορά heur2 (ms)
1	355	355	355	0	0
2	222	292	296	69	74
3	183	235	283	52	100
4	148	191	244	43	97
5	128	159	215	31	87
6	112	136	185	24	73
7	103	119	165	16	62
8	94	108	164	14	70
9	86	97	159	11	73
10	78	88	134	11	56
11	70	80	128	10	58
12	63	71	142	9	79
13	55	63	138	8	82
14	48	56	116	7	67
15	42	48	111	7	69
16	35	41	101	6	66
17	29	34	87	6	59
18	23	28	76	5	53
19	17	21	60	5	43
20	11	15	40	4	30
21	8	9	28	1	20
22	5	6	19	1	14
23	3	4	11	1	8
24	1	1	5	0	4

Πίνακας 25. Αποτελέσματα κόστους ανά χρήση σε τοπολογία lattice, edge distribution

# Nodes	Exh	Heur1	Heur2	Διαφορά heur1 (ms)	Διαφορά heur2 (ms)
1	3,66	3,66	3,66	0,00	0,00
2	2,29	3,01	3,05	0,72	0,76
3	1,88	2,42	2,91	0,54	1,03
4	1,52	1,97	2,52	0,44	0,99
5	0,80	0,99	1,34	0,20	0,54
6	0,70	0,85	1,15	0,15	0,45
7	0,64	0,75	1,03	0,10	0,39
8	0,59	0,68	1,03	0,09	0,44
9	0,54	0,61	1,00	0,07	0,46
10	0,49	0,55	0,84	0,07	0,35
11	0,44	0,50	0,80	0,06	0,36
12	0,39	0,45	0,89	0,05	0,49
13	0,35	0,40	0,86	0,05	0,51
14	0,30	0,35	0,72	0,05	0,42
15	0,26	0,30	0,69	0,04	0,43
16	0,22	0,26	0,63	0,04	0,41
17	0,18	0,21	0,55	0,03	0,37
18	0,14	0,17	0,48	0,03	0,33
19	0,10	0,13	0,37	0,03	0,27
20	0,07	0,09	0,25	0,03	0,19
21	0,05	0,06	0,17	0,01	0,13
22	0,03	0,04	0,12	0,01	0,08
23	0,02	0,02	0,07	0,00	0,05
24	0,01	0,01	0,03	0,00	0,02

Πίνακας 26. Αποτελέσματα διακύμανσης σε τοπολογία lattice, edge distribution

# Nodes	Exh	Heur1	Heur2	Διαφορά heur1	Διαφορά heur2
2	0,50	0,90	0,50	0,40	0,00
3	12,63	15,93	0,33	3,30	-12,30
4	0,28	11,18	0,25	10,91	-0,03
5	0,72	6,15	0,00	5,43	-0,72
6	1,69	3,37	0,17	1,68	-1,52
7	1,12	2,05	0,29	0,93	-0,83
8	1,04	1,55	0,12	0,51	-0,92
9	0,72	1,19	0,19	0,47	-0,53
10	0,68	0,66	0,28	-0,02	-0,40
11	0,50	0,66	0,22	0,16	-0,28
12	0,45	0,54	0,08	0,09	-0,36
13	0,44	0,48	0,08	0,03	-0,37
14	0,43	0,37	0,18	-0,06	-0,25
15	0,38	0,37	0,24	-0,01	-0,14
16	0,36	0,34	0,26	-0,01	-0,09
17	0,29	0,30	0,26	0,01	-0,03
18	0,25	0,27	0,25	0,02	0,00
19	0,23	0,24	0,23	0,01	0,00
20	0,20	0,20	0,20	0,00	0,00
21	0,16	0,16	0,16	0,00	0,00
22	0,12	0,12	0,12	0,00	0,00
23	0,08	0,08	0,08	0,00	0,00
24	0,04	0,04	0,04	0,00	0,00

4.4 Συμπεράσματα

Τα συμπεράσματα τα οποία μπορούν να εξαχθούν από τις ανωτέρω μετρήσεις είναι ότι ο πρώτος ευριστικός αλγόριθμος παρουσιάζει μια μέση αύξηση του κόστους καθυστέρησης σε σχέση με τον εξαντλητικό, της τάξης του 19,47% και για τα τρία σενάρια στην τοπολογία scale free και 10,84% στην τοπολογία lattice.

Η μικρότερη απόκλιση παρουσιάζεται στην τοπολογία lattice, στο σενάριο όπου υπάρχει ομοιόμορφη κατανομή βαρών. Στο ίδιο σενάριο για τον πρώτο αλγόριθμο υπάρχει και η μεγαλύτερη απόκλιση σχετικά με την κατανομή των χρηστών όπου παρατηρείται αύξηση κατά 2,3 φορές. Όπως φαίνεται και στον παρακάτω συγκεντρωτικό πίνακα, για την τοπολογία lattice όσο αυξάνεται η απόκλιση από τον εξαντλητικό, σε σχέση με το κόστος καθυστέρησης, μειώνεται η απόκλιση σχετικά με την κατανομή των χρηστών.

Ομοίως και στην τοπολογία scale free παρατηρούμε ότι όσο μειώνεται η απόκλιση σχετικά με το κόστος καθυστέρησης, αυξάνεται η απόκλιση του αλγορίθμου σχετικά με την κατανομή των χρηστών, εν συγκρίσει με τον εξαντλητικό αλγόριθμο.

Αναλύοντας τις προαναφερθείσες διαφορές, παρατηρείται ότι στην τοπολογία lattice οι μεγαλύτερες διαφορές εντοπίζονται στις περιπτώσεις από δυο έως τέσσερις service nodes ενώ στην τοπολογία scale free από τέσσερις έως έξι service nodes. Στην κατανομή των χρηστών δεν μπορεί να εξαχθεί κάποιο κοινό συμπέρασμα.

Ο δεύτερος ευριστικός αλγόριθμος οδηγεί σε αύξηση του κόστους καθυστέρησης, σε σχέση με τον εξαντλητικό, ύψους 84,49% στην τοπολογία scale free και 61,08% στην τοπολογία lattice, βελτιώνοντας αντίστοιχα την διακύμανση κατά 93,32% στην τοπολογία scale free και 62,24% στην τοπολογία lattice.

Οι μεγαλύτερες διαφορές στην τοπολογία lattice, παρατηρούνται για δυο έως τέσσερις service nodes, στις περιπτώσεις που υπάρχει μεγαλύτερη κατανομή είτε στις άκρες, είτε στο κέντρο του δικτύου. Στην uniform κατανομή οι μεγαλύτερες αποκλίσεις παρατηρούνται για δεκαέξι έως δεκαοκτώ service nodes.

Στην τοπολογία scale free οι μεγαλύτερες διαφορές και στα τρία σενάρια κατανομής, εντοπίζεται στις περιπτώσεις για τέσσερις έως οκτώ service nodes.

Πίνακας 27. Διαφορές ευριστικών αλγορίθμων σε σχέση με τον εξαντλητικό, ως προς το κόστος καθυστέρησης

Σενάριο κατανομής	Heuristic1		Heuristic2	
	scale free	lattice	scale free	lattice
Uniform κατανομή	24,33%	6,34%	86,63%	63,40%
Κατανομή στους κεντρικούς κόμβους	14,62%	8,39%	68,97%	59,35%
Κατανομή στους περιφερειακούς κόμβους	19,46%	17,79%	97,85%	60,50%
Μέση τιμή	19,47%	10,84%	84,49%	61,08%

Πίνακας 28. Διαφορές ευριστικών αλγορίθμων σε σχέση με τον εξαντλητικό, ως προς την διακύμανση των access nodes στους service nodes

Σενάριο κατανομής	Heuristic1		Heuristic2	
	scale free	lattice	scale free	lattice
Uniform Κατανομή	50,49%	239,78%	-92,48%	-35,58%
Κατανομή στους κεντρικούς κόμβους	124,80%	182,06%	-90,61%	-73,99%
Κατανομή στους περιφερειακούς κόμβους	74,32%	98,09%	-96,58%	-77,16%
Μέση τιμή	83,20%	173,31%	-93,23%	-62,24%

Πίνακας 29. Πίνακας σύγκρισης αλγορίθμων ως προς το κόστος καθυστέρησης ανά χρήστη

Σενάριο κατανομής	SCALE FREE			LATTICE		
	Exhaustive	Heuristic1	Heuristic2	Exhaustive	Heuristic1	Heuristic2
Uniform Κατανομή	0,62	0,78	1,16	0,58	0,62	0,95
Κατανομή στους κεντρικούς κόμβους	0,45	0,52	0,76	0,39	0,42	0,62
Κατανομή στους περιφερειακούς κόμβους	0,58	0,69	1,14	0,65	0,77	1,05

Πίνακας 30. Πίνακας σύγκρισης αλγορίθμων ως προς την διακύμανση

Σενάριο κατανομής	SCALE FREE			LATTICE		
	Exhaustive	Heuristic1	Heuristic2	Exhaustive	Heuristic1	Heuristic2
Uniform Κατανομή	2,52	3,79	0,19	0,29	1,00	0,19
Κατανομή στους κεντρικούς κόμβους	2,02	4,53	0,19	0,73	2,05	0,19
Κατανομή στους περιφερειακούς κόμβους	5,53	9,64	0,19	1,01	2,05	0,19

Κατόπιν σύγκρισης των αποτελεσμάτων των δυο αλγορίθμων μεταξύ τους, προκύπτει πως ο δεύτερος αλγόριθμος για όλα τα σενάρια της τοπολογίας scale free βελτιώνει την διακύμανση κατά 96,29%, αυξάνοντας παράλληλα το κόστος καθυστέρησης κατά 54,39%. Αντίστοιχα στην τοπολογία lattice επιφέρει μείωση της διακύμανσης κατά 87,58%, με αντίστοιχη αύξηση του κόστους καθυστέρησης κατά 45,65%. Συνεπώς σε όλες τις περιπτώσεις η αύξηση του κόστους είναι υποπολλαπλάσια της μείωσης που επέρχεται στην κατανομή των χρηστών.

Πίνακας 31. Διαφορές δεύτερου ευριστικού αλγορίθμου σε σχέση με τον πρώτο, ως προς το κόστος καθυστέρησης

Σενάριο κατανομής	scale free	lattice
Uniform κατανομή	50,11%	53,66%
Κατανομή στους κεντρικούς κόμβους	47,42%	47,02%
Κατανομή στους περιφερειακούς κόμβους	65,63%	36,26%
Μέση τιμή	54,39%	45,65%

Πίνακας 32. Διαφορές δεύτερου ευριστικού αλγορίθμου σε σχέση με τον πρώτο, ως προς την διακύμανση

Σενάριο κατανομής	scale free	lattice
Uniform Κατανομή	-95,01%	-85,34%
Κατανομή στους κεντρικούς κόμβους	-95,82%	-90,78%
Κατανομή στους περιφερειακούς κόμβους	-98,04%	-86,61%
Μέση τιμή	-96,29%	-87,58%

Εν κατακλείδι, συμπεραίνουμε πως βάσει των ανωτέρω διαπιστώσεων οι αλγόριθμοι θα πρέπει να προσαρμοστούν, προκειμένου να παρουσιάζουν μικρότερες αποκλίσεις, εν συγκρίσει με τον εξαντλητικό, τόσο σε επίπεδο καθυστέρησης δικτύου όσο και σε κατανομή των access nodes.

Βιβλιογραφία

- [1] Mell, P M, and T Grance. "The NIST Definition of Cloud Computing." 2011, doi:10.6028/nist.sp.800-145.
- [2] Buyya, Rajkumar, et al. *Cloud Computing: Principles and Paradigms*. Wiley, 2011.
- [3] "Virtualization in Cloud Computing." *Journal of Information Technology & Software Engineering*, vol. 04, no. 02, 2014, doi:10.4172/2165-7866.1000136.
- [4] D'Angelo, Gabriele, and Moreno Marzolla. "New Trends in Parallel and Distributed Simulation: From Many-Cores to Cloud Computing." *Simulation Modelling Practice and Theory*, vol. 49, 2014, pp. 320–335., doi:10.1016/j.simpat.2014.06.007.
- [5] Gong, Chunye, et al. "The Characteristics of Cloud Computing." 2010 39th International Conference on Parallel Processing Workshops, 2010, doi:10.1109/icppw.2010.45.
- [6] Mahmood, Zaigham. "Cloud Computing: Characteristics and Deployment Approaches." 2011 IEEE 11th International Conference on Computer and Information Technology, 2011, doi:10.1109/cit.2011.75.
- [7] Mietzner, Ralph, et al. "Combining Different Multi-Tenancy Patterns in Service-Oriented Applications." 2009 IEEE International Enterprise Distributed Object Computing Conference, 2009, doi:10.1109/edoc.2009.13.
- [8] Colman-Meixner, C., Develder, C., Tornatore, M., & Mukherjee, B. (2016). A Survey on Resiliency Techniques in Cloud Computing Infrastructures and Applications. *IEEE Communications Surveys & Tutorials*, 18(3), 2244–2281. doi:10.1109/comst.2016.2531104
- [9] Qian, Ling, et al. "Cloud Computing: An Overview." SpringerLink, Springer, Berlin, Heidelberg, 1 Dec. 2009, link.springer.com/chapter/10.1007/978-3-642-10665-1_63.
- [10] Voorsluys, William, et al. "Introduction to Cloud Computing." *Cloud Computing*, 2011, pp. 1–41., doi:10.1002/9780470940105.ch1.
- [11] Coles-Kemp, Lizzie, et al. "Looking at Clouds from Both Sides: The Advantages and Disadvantages of Placing Personal Narratives in the Cloud." *Information Security Technical Report, Elsevier Advanced Technology*, 10 Oct. 2011

- [12] Shshil B., Leena J., Sandeep J.; (2010); "Cloud Computing: A study of infrastructure as a service (IaaS)"; *International Journal of Engineering and Information Technology*;
- [13] Kavis, Michael J. *Architecting the Cloud: Design Decisions for Cloud Computing Service Models*. John Wiley & Sons, 2014.
- [14] Mietzner, Ralph, et al. "Variability Modeling to Support Customization and Deployment of Multi-Tenant-Aware Software as a Service Applications." 2009 ICSE Workshop on Principles of Engineering Service Oriented Systems, 2009, doi:10.1109/pesos.2009.5068815.
- [15] D. Kovachev, Y. Cao, R. Klamma, "Mobile cloud computing: a comparison of application models" *Computing Research Repository*, (2010)
- [16] F. Leymann, "Cloud Computing: The Next Revolution in IT", *Institute of Architecture of Application Systems, University of Stuttgart, Germany, 2009*.
- [17] Fehling, Christoph, et al. "Cloud Computing Fundamentals." *Cloud Computing Patterns*, 2013, pp. 21–78., doi:10.1007/978-3-7091-1568-8_2.
- [18] Sosinsky, Barrie. *Cloud Computing Bible*. Wiley Publ., 2011.
- [19] Dinh, Hoang T., et al. "A Survey of Mobile Cloud Computing: Architecture, Applications, and Approaches." *Wireless Communications and Mobile Computing*, vol. 13, no. 18, 2011, pp. 1587–1611., doi:10.1002/wcm.1203.
- [20] M. Neishaboori, "Implementation and Evaluation of Mobile-Edge Computing Cooperative Caching", *Aalto University School of Science Master's Programme in ICT Innovation, Finland, 2015*.
- [21] Qi, Han, and Abdullah Gani. "Research on Mobile Cloud Computing: Review, Trend and Perspectives." 2012 *Second International Conference on Digital Information and Communication Technology and It's Applications (DICTAP)*, 2012, doi:10.1109/dictap.2012.6215350.
- [22] Alzahrani, Ahmed, et al. "Mobile Cloud Computing." *Proceedings of the 7th Euro American Conference on Telematics and Information Systems - EATIS '14*, 2014, doi:10.1145/2590651.2590670.
- [23] Gupta, Pragya, and Sudha Gupta. "Mobile cloud computing: the future of cloud." *International Journal of Advanced Research in*

Electrical, Electronics and Instrumentation Engineering (2012)

- [24] Verbelen, Tim, et al. "Cloudlets." *Proceedings of the Third ACM Workshop on Mobile Cloud Computing and Services - MCS '12*, 2012, doi:10.1145/2307849.2307858.
- [25] Haoliang Wang. MS Thesis "Accelerating Mobile-Cloud Computing Using A Cloudlet" 2013.
- [26] Jia, Mike, et al. "Cloudlet Load Balancing in Wireless Metropolitan Area Networks." *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, 2016, doi:10.1109/infocom.2016.7524411.
- [27] Abbas, Nasir, et al. "Mobile Edge Computing: A Survey." *IEEE Internet of Things Journal*, vol. 5, no. 1, 2018, pp. 450–465., doi:10.1109/jiot.2017.2750180.
- [28] Satyanarayanan, Mahadev. "The Emergence of Edge Computing." *Computer*, vol. 50, no. 1, 2017, pp. 30–39., doi:10.1109/mc.2017.9.
- [29] Ahmed, Ejaz, and Mubashir Husain Rehmani. "Mobile Edge Computing: Opportunities, Solutions, and Challenges." *Future Generation Computer Systems*, vol. 70, 2017, pp. 59–63., doi:10.1016/j.future.2016.09.015.
- [30] Jararweh, Yaser, et al. "The Future of Mobile Cloud Computing: Integrating Cloudlets and Mobile Edge Computing." *2016 23rd International Conference on Telecommunications (ICT)*, 2016, doi:10.1109/ict.2016.7500486.
- [31] Roman, Rodrigo, et al. "Mobile Edge Computing, Fog Et Al.: A Survey and Analysis of Security Threats and Challenges." *Future Generation Computer Systems*, vol. 78, 2018, pp. 680–698., doi:10.1016/j.future.2016.11.009.
- [32] Oikonomou, Efthymios, and Angelos Rouskas. "Optimized Cloudlet Management in Edge Computing Environment." *2018 IEEE 23rd International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, 2018, doi:10.1109/camad.2018.8514942.
- [33] Zadorozhnyi, V. N., and E. B. Yudin. "Structural Properties of the Scale-Free Barabasi-Albert Graph." *Automation and Remote Control*, vol. 73, no. 4, 2012, pp. 702–716., doi:10.1134/s0005117912040091.
- [34] Barabási, Albert-László, et al. "Scale-Free Characteristics of

Random Networks: the Topology of the World-Wide Web.” *Physica A: Statistical Mechanics and Its Applications*, vol. 281, no. 1-4, 2000, pp. 69–77., doi:10.1016/s0378-4371(00)00018-2.

[35] Liu, J., & Zhang, Q. (2018). *Offloading Schemes in Mobile Edge Computing for Ultra-Reliable Low Latency Communications*. *IEEE Access*, 6, 12825–12837. doi:10.1109/access.2018.2800032

[36] Jia, Mike, et al. “*Cloudlet Load Balancing in Wireless Metropolitan Area Networks*.” *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, 2016, doi:10.1109/infocom.2016.7524411.