



ΔΗΜΟΚΡΙΤΕΙΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΡΑΚΗΣ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ
ΜΗΧΑΝΙΚΩΝ & ΜΗΧΑΝΙΚΩΝ
ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΛΟΓΙΣΜΙΚΟΥ ΚΑΙ
ΑΝΑΠΤΥΞΗΣ ΕΦΑΡΜΟΓΩΝ

Υβριδικά πληροφοριοκεντρικά δίκτυα

Διπλωματική Εργασία
του **Καραγιάννη Χρήστου**

Υπό την επίβλεψη του καθηγητή
κ. **Τσαουσίδη Βασιλείου**

Ξάνθη, 2019

Ευχαριστίες

Ολοκληρώνοντας την προπτυχιακή μου πορεία θα ήθελα να ευχαριστήσω πρωτίστως τον κ. Τσαουσίδα Βασίλειο που με εμπιστεύθηκε και μου ανέθεσε να φέρω εις πέρας το συγκεκριμένο θέμα της διπλωματικής μου εργασίας.

Επιπλέον τις θερμές μου ευχαριστίες οφείλω στον κ. Σάρρο Χρήστο-Αλέξανδρο που από την θέση του ως διδακτορικός φοιτητής μου επέδειξε πλήρη στήριξη, καθοδήγηση και με τις υποδείξεις και τις ιδέες του με βοήθησε ουσιαστικά καθ' όλη τη διάρκεια εκπόνησης αυτής της εργασίας.

Τέλος, ένα τεράστιο ευχαριστώ στην οικογένεια μου που με στήριξε με κάθε τρόπο κατά την διάρκεια των σπουδών μου και σε όλους τους φίλους μου που είναι δίπλα μου καθημερινά και με στηρίζουν σε κάθε μου προσπάθεια.

Η εργασία αυτή είναι εις μνήμην της αγαπημένης μου γιαγιάς Μαρίας που έφυγε από τη ζωή πρόσφατα.

Καραγιάννης Χρήστος

Πίνακας Περιεχομένων

Κεφάλαιο 1: Εισαγωγή.....	1
1.1 Το διαδίκτυο	1
1.2 Περιγραφή του προβλήματος	2
1.3 Θέμα	3
1.4 Στόχοι εργασίας	5
1.5 Διάρθρωση.....	6
Κεφάλαιο 2: Σχετικές δουλειές	7
2.1 Εισαγωγή.....	7
2.2 ICN (Information-Centric Networking)	7
2.2.1 Ονοματοδοσία πληροφοριών	8
2.2.2 Παράδοση πληροφοριών	9
2.2.3 Ευκινησία.....	10
2.2.4 Ασφάλεια	12
2.3 CCN/NDN (Content-Centric Networking / Named Data Networking)	13
2.3.1 Η αρχιτεκτονική.....	14
2.3.2 Διαδικασία προώθησης.....	15
2.4 hICN (hybrid Information-Centric Networking)	17
2.4.1 Αρχιτεκτονική.....	17
2.4.2 Τελικοί κόμβοι	21
2.4.3 Ονοματοδοσία	22
2.4.3.1 Πρόθεμα ονόματος	22
2.4.3.2 Επίθεμα ονόματος	23
2.4.4 Μορφή πακέτων.....	23
2.4.4.1 Πακέτο ενδιαφέροντος.....	23
2.4.4.2 Πακέτο δεδομένων	25
2.4.5 Κρυφή μνήμη	27
2.4.6 Προώθηση.....	27
2.4.6.1 Διαδρομή προώθησης πακέτων τύπου «Interest».....	28
2.4.6.2 Διαδρομή προώθησης πακέτων τύπου «Data»	30
2.4.7 Ασφάλεια	31

2.4.7.1	Επικύρωση κεφαλίδων.....	32
2.4.7.2	Μανιφέστο μεταφοράς.....	34
Κεφάλαιο 3: Εργαλεία		35
3.1	Εισαγωγή.....	35
3.2	νICN (virtualized Information-Centric Networking)	35
3.2.1	Εισαγωγή	35
3.2.2	Περιγραφή ανάπτυξης	37
3.2.3	Δομή της νICN	38
3.2.3.1	Πρακτική αρχιτεκτονική	39
3.2.3.2	Μοντέλο πόρων	40
3.2.3.3	Επεξεργαστής πόρων	43
3.2.3.4	Ενορχηστρωτής και χρονοπρογραμματιστής	43
3.2.4	Εφαρμογή της νICN	44
3.2.4.1	Κώδικας.....	44
3.2.4.2	Διαμοιρασμός	45
3.2.4.3	IP και ICN τοπολογίες.....	46
3.2.4.4	Εξομοίωση συνδέσεων	47
3.2.4.5	Παρακολούθηση (Monitoring)	48
3.3	NS-3 (Network Simulator-3)	48
3.3.1	Μοντέλα προσομοίωσης	49
3.3.2	Δομές WiFi και LTE	50
Κεφάλαιο 4: Τοπολογίες, σενάρια και παράμετροι.....		51
4.1	Εισαγωγή.....	51
4.2	Τοπολογίες	51
4.2.1	Dumbbell	51
4.2.2	Dumbbell πολλαπλών διαδρομών	53
4.2.3	Tree 2-επιπέδων	55
4.2.4	Δημιουργία τοπολογιών	56
4.2.5	Παρακολούθηση τοπολογίας	61
4.3	Σενάρια	63
4.4	Παράμετροι	65
4.4.1	Πολιτική αντικατάστασης κρυφής μνήμης	65

4.4.2 Έλεγχος συμφόρησης.....	65
Κεφάλαιο 5: Αποτελέσματα.....	67
5.1 Αποτελέσματα Σεναρίου 1 ^{ου}	67
5.1.1 Ενσύρματη σύνδεση	67
5.1.2 Ασύρματη σύνδεση μέσω WiFi.....	71
5.1.3 Ασύρματη σύνδεση μέσω LTE	74
5.2 Αποτελέσματα Σεναρίου 2 ^{ου}	78
5.2.1 Ενσύρματη σύνδεση	78
5.2.2 Ασύρματη σύνδεση μέσω WiFi.....	80
5.2.3 Ασύρματη σύνδεση μέσω LTE	83
5.3 Αποτελέσματα Σεναρίου 3 ^{ου}	85
5.3.1 Ενσύρματη σύνδεση	85
5.3.2 Ασύρματη σύνδεση μέσω WiFi.....	88
5.3.3 Ασύρματη σύνδεση μέσω LTE	91
5.4 Σύγκριση συνδέσεων σεναρίων.....	94
5.4.1 Σύγκριση συνδέσεων Σεναρίου 1 ^{ου}	94
5.4.2 Σύγκριση συνδέσεων Σεναρίου 3 ^{ου}	97
Κεφάλαιο 6: Συμπεράσματα	100
Κεφάλαιο 7: Βιβλιογραφία	102

Κεφάλαιο 1: Εισαγωγή

1.1 Το διαδίκτυο

Το διαδίκτυο σήμερα είναι ένα τεράστιο σύστημα διασυνδεδεμένων δικτύων υπολογιστών που χρησιμοποιεί δεκάδες πρωτόκολλα ώστε να συνδέει συσκευές σε παγκόσμια κλίμακα. Από την ίδρυση του κίβλας, στις αρχές της δεκαετίας του 1970, η αρχιτεκτονική που επιβλήθηκε βασίζεται σε ένα μοντέλο επικοινωνίας σημείου-σημείου μεταξύ δύο τερματικών κόμβων. Τον καιρό εκείνο το πρωτόκολλο που κλήθηκε να λύσει το πρόβλημα της επικοινωνίας ήταν το γνωστό μέχρι και σήμερα πρωτόκολλο TCP/IP, το οποίο έχει αποδειχθεί άξιο των προσδοκιών αφού μέχρι και σήμερα εξυπηρετεί σπουδαία τις συνεχώς αυξανόμενες ανάγκες του διαδικτύου.

Η εξέλιξη της χρήσης του διαδικτύου είναι ραγδαία, αφού σε αντίθεση με τις πρώιμες μέρες του που το διαδίκτυο χρησιμοποιούνταν κυρίως ως ηλεκτρονικό ταχυδρομείο αλλά και για την μεταφορά αρχείων, σήμερα έχει γίνει απαραίτητο κομμάτι της καθημερινότητας του ανθρώπινου είδους. Είναι αδύνατον να αναφέρουμε κάθε δυνατότητα που προσφέρει το διαδίκτυο σήμερα αλλά μερικές από αυτές είναι το διαδικτυακό εμπόριο, τα μέσα κοινωνικής δικτύωσης, οι ηλεκτρονικοί τρόποι πληρωμής, το ηλεκτρονικό ταχυδρομείο, η ηλεκτρονική ενημέρωση, η επικοινωνία μέσω διαδικτύου σε πραγματικό χρόνο (είτε γραπτά, είτε μέσω φωνής/βίντεο), η διαδικτυακή τηλεόραση και μουσική και αμέτρητα ακόμα είδη υπηρεσιών. Αξίζει να σημειωθεί ότι οι ειδικοί ονομάζουν ήδη τον αιώνα που διανύουμε εποχή της πληροφορίας.

Παράλληλα με την εξέλιξη του διαδικτύου υπήρξε και μια άλλη σπουδαία εξέλιξη, αυτή των έξυπνων συσκευών, με την αρχή να γίνεται από τα κινητά τηλέφωνα που απέκτησαν την δυνατότητα να συνδέονται στο διαδίκτυο και να εκτελούν σχεδόν κάθε δραστηριότητα που εκτελεί και ένας ηλεκτρονικός υπολογιστής. Ακολουθώντας, την δεκαετία που διανύουμε ακόμα περισσότερες μηχανές και οικιακές συσκευές, όπως tablets, αυτοκίνητα, ψυγεία, ρολόγια, θερμοστάτες κ.α μπορούν να συνδεθούν σε δίκτυα και να επικοινωνήσουν μεταξύ τους.

1.2 Περιγραφή του προβλήματος

Η συνεχής εξέλιξη τόσο της τεχνολογίας των υπολογιστών και του διαδικτύου όσο και των εφαρμογών που τρέχουν στα ανώτερα επίπεδα του μοντέλου αναφοράς OSI, αλλά και η ταχεία αύξηση της χρήση αυτών έχει επιφέρει την ανάγκη για αλλαγές στον τρόπο επικοινωνίας των τερματικών κόμβων. [1]

Κάποιες νέες απαιτήσεις από την αρχιτεκτονική που πρόσφατα έχουν αναδυθεί είναι η ανάγκη για επεκτασιμότητα, ευκινησία και ασφάλεια του δικτύου κ.α. Με διάφορα πρωτόκολλα και λογισμικά ενημέρωσης, όπως το Mobile IP, επιχειρήθηκε ανά καιρούς να ικανοποιηθούν αυτές οι λειτουργικές απαιτήσεις ωστόσο τα περισσότερα από αυτά κατάφεραν με προσωρινές λύσεις να κάνουν την τρέχουσα αρχιτεκτονική του διαδικτύου ακόμη πιο πολυσύνθετη.

Όπως είναι κατανοητό η πρακτική των συνεχών επιδιορθώσεων (patches) αν και χρησιμοποιείται ευρέως δεν είναι η πλέον αποδοτική πρακτική, ιδίως όταν μιλάμε για την αρχιτεκτονική ενός αντικειμένου που παρουσιάζει τεράστια διέγερση και συνεχή εξέλιξη. Έτσι νέες ερευνητικές κοινότητες συστάθηκαν με στόχο να αναγνωρίσουν τους περιορισμούς που θέτει το τρέχον μοντέλο, να υποδείξουν τα σημεία κλειδιά στα οποία θα στηρίζεται το διαδίκτυο στο μέλλον και τελικώς να προτείνουν νέες αρχιτεκτονικές με παραδείγματα και εφαρμογές.

1.3 Θέμα

Η παρούσα διπλωματική εργασία εστιάζει σε μια νέα και πολλά υποσχόμενη αρχιτεκτονική η οποία θεωρείται υποψήφια διάδοχος της υπάρχουσας για το διαδίκτυο του μέλλοντος, την Information-Centric Networking (ICN) [1]. Σήμερα η είσοδος των ηλεκτρονικών συσκευών όπως smartphones και tablets στην αγορά έχει γεννήσει την ανάγκη για φορητότητα των τερματικών κόμβων ενώ παράλληλα το εύρος του τύπου των δεδομένων που παράγεται και διακινείται έχει μεγαλώσει με αμέτρητα βίντεο, φωτογραφίες, βιντεοκλήσεις, ηχητικά και γραπτά μηνύματα να αποτελούν τον κύριο όγκο του περιεχομένου.

Μπροστά στις αλλαγές αυτές το IP πρωτόκολλο χρειάστηκε να προσαρμοστεί (π.χ IPv6) έτσι ώστε να μπορεί να εξυπηρετήσει τον συνεχώς αυξανόμενο αριθμό κινητών συσκευών αλλά σε τελική ανάλυση ο σκοπός της θεμελίωσης του ήταν η επικοινωνία δύο τερματικών μεταξύ τους. Επιπλέον, κατά τη κατασκευή του IP πρωτόκολλου δεν υπήρχε τόσο έντονη η ανάγκη για ασφάλεια καθώς η ίδια η πρόσβαση ήταν περιορισμένη. Ωστόσο με την τροπή που έχει πάρει το διαδίκτυο και με εκατομμύρια χρήστες να αναπτύσσουν οικονομική δραστηριότητα ψηφιακά, η είσοδος κακοηθών χρηστών και λογισμικών στον κλάδο δεν είναι κάτι αναπάντεχο. Οι παραβιάσεις ασφαλείας τόσο χρηστών όσο και επιχειρήσεων πλέον είναι καθημερινό φαινόμενο και η ιδιωτικότητα δεν θεωρείται καθόλου δεδομένη.

Καθώς λοιπόν η πληροφορία στο διαδίκτυο χρήζει μαζικής διάδοσης σε αντίθεση με το παρελθόν όπου είχαμε επικοινωνία τερματικού-τερματικού και αφού πρέπει να δοθεί ιδιαίτερη προσοχή στο κομμάτι της προστασίας των δεδομένων, η αρχιτεκτονική ICN φαίνεται να προσεγγίζει ορθότερα τα ζητήματα που προκύπτουν.

Η θεμελιώδης αρχή της ICN [2] αρχιτεκτονικής για την εξέλιξη της υποδομής του διαδικτύου είναι η μοναδική ονοματοδοσία του περιεχομένου που δίνουν στην πληροφορία τον κεντρικό ρόλο στην επικοινωνία και όχι σε κάποια IP διεύθυνση, καθιστώντας έτσι την επικοινωνία ανεξάρτητη της τοποθεσίας. Η πρόσβαση στα δεδομένα ανεξαρτητοποιείται της τοποθεσίας, της εφαρμογής και της αποθήκευσης ενώ επιτρέπει την αποθήκευση δεδομένων εντός του

δικτύου (in-network caching), την εφαρμογή μηχανισμών πολυεκπομπής (multicast mechanisms) και την φορητότητα χωρίς άγκυρα (anchor-less mobility). Τα άμεσα οφέλη για το σύστημα είναι η βελτίωση της αποδοτικότητας με μικρότερους χρόνους παράδοσης των δεδομένων στους χρήστες, η δυνατότητα επεκτασιμότητας, η βελτιωμένη διαχείριση της φορητότητας και η επιβολή αρχών ασφαλείας.

Την τελευταία δεκαετία αρκετές υλοποιήσεις της ICN αρχιτεκτονικής έχουν σχεδιαστεί με σκοπό να αντικαταστήσουν το υπάρχον μοντέλο που βασίζεται στην επικοινωνία με έναν τερματικό κόμβο. Παρόλο που ακόμη βρίσκονται σε στάδιο ανάπτυξης, οι υλοποιήσεις αυτές στοχεύουν να διευθετήσουν ακριβώς τα ίδια κύρια σημεία με διαφορετικές βέβαια προσεγγίσεις. Αν και εκτενέστερη αναφορά θα γίνει στα επόμενα κεφάλαια, επιγραμματικά τα σημεία στα οποία εστιάζουν είναι αυτά της ονοματοδοσίας, της ανάλυσης ονόματος και δρομολόγησης δεδομένων, της αποθήκευσης δεδομένων για μελλοντική δρομολόγηση, της φορητότητας των χρηστών και της ασφάλειας.

Οι σημαντικότερες από τις υλοποιήσεις της ICN αρχιτεκτονικής αναφέρονται συνοπτικά παρακάτω:

- DONA από το Berkeley
- PSIRP και στη συνέχεια PURSUIT με Ευρωπαϊκή χρηματοδότηση
- CCN και στη συνέχεια NDN με Αμερικανική χρηματοδότηση
- ANR Connect (βασίζεται στην NDN) με Γαλλική χρηματοδότηση
- 4WARD και στη συνέχεια SAIL
- COMET
- CONVERGENCE
- MobilityFirst

1.4 Στόχοι εργασίας

Μετά τη σύντομη εισαγωγή στη φύση του θέματος είναι σκόπιμο να τεθεί ο στόχος της εργασίας καθώς και τα μέσα που θα επιστρατευθούν για την περάτωσή της. Ο στόχος λοιπόν που τέθηκε μετά τη θεωρητική μελέτη της αρχιτεκτονικής ICN είναι η κατασκευή διάφορων σεναρίων και τοπολογιών δικτύου σε κάποιο περιβάλλον προσομοίωσης, έτσι ώστε να εξεταστούν οι λύσεις που παρέχει η συγκεκριμένη αρχιτεκτονική και οι επιδόσεις αυτών υπό την μεταβολή διάφορων παραμέτρων.

Το βασικό εργαλείο στο οποίο βασίστηκε η προσομοίωση των τοπολογιών είναι ένα σύστημα διαχείρισης δεδομένων οδηγούμενο από το μοντέλο που εισάγουμε και το οποίο μας παρέχει υπηρεσίες ενορχήστρωσης του συστήματος για την κατασκευή μεγάλης κλίμακας υλοποιήσεων της αρχιτεκτονικής ICN. Το εργαλείο αυτό ονομάζεται Virtualized Information-Centric Networking (vICN) [3].

Το vICN είναι γραμμένο στη γλώσσα προγραμματισμού Python και προσομοιώνει δίκτυα μεγάλης κλίμακας χρησιμοποιώντας της τυπικές μονάδες του λογισμικού Linux, τα Linux containers. Για την προσομοίωση της ασύρματης επικοινωνίας τόσο μέσω WiFi όσο και μέσω LTE η υλοποίηση χρησιμοποιεί και κάποιες εφαρμογές που προσφέρονται από τον προσομοιωτή δικτύων ns-3. Περαιτέρω αναφορά στα εργαλεία αυτά και τις δυνατότητες που προσφέρουν θα γίνει στα επόμενα κεφάλαια.

1.5 Διάρθρωση

Μετά το πέρας του πρώτου εισαγωγικού κεφαλαίου όπου γίνεται συνοπτικά η περιγραφή του προβλήματος, γίνεται μια πρώτη επαφή με το θέμα που ασχολείται η διπλωματική εργασία και παρατίθενται οι στόχοι της, έπειτα ακολουθούν ακόμη τέσσερα κεφάλαια.

Στο δεύτερο κεφάλαιο παρουσιάζονται οι σχετικές δουλειές που έχουν γίνει στο παρελθόν και η σύνδεση τους με την παρούσα εργασία. Πιο συγκεκριμένα στην πρώτη ενότητα αναλύεται διεξοδικά η αρχιτεκτονική ICN, στη δεύτερη ενότητα γίνεται αναφορά σε δύο πολύ σημαντικές υλοποιήσεις της ICN αρχιτεκτονικής και συγκεκριμένα στη CCN και στη NDN, ενώ στη τρίτη ενότητα εξετάζεται η Hybrid ICN (hICN) αρχιτεκτονική που αποτελεί μια υλοποίηση της λογικής ICN προσανατολισμένη στη σταδιακή αντικατάσταση του σημερινού διαδικτύου που βασίζεται στο πρωτόκολλο IP.

Στο τρίτο κεφάλαιο περιγράφονται τα εργαλεία που χρησιμοποιήθηκαν για την κατασκευή και την μελέτη των διάφορων τοπολογιών και τελικώς για την περάτωση της εργασίας. Πιο συγκεκριμένα στην πρώτη ενότητα παρουσιάζεται εις βάθος το εργαλείο nICN που προαναφέρθηκε στο κεφάλαιο αυτό και στη δεύτερη ενότητα παρουσιάζεται ο προσομοιωτής ns-3 που χρησιμοποιείται από το nICN για να πετύχει την προσομοίωση της ασύρματης επικοινωνίας μέσω WiFi και LTE.

Στο τέταρτο κεφάλαιο παρατίθενται οι διάφορες τοπολογίες δικτύου που δημιουργήθηκαν στα πλαίσια της μελέτης, τα σενάρια που εξετάστηκαν καθώς και οι παράμετροι που τέθηκαν για τις μετρήσεις.

Στο πέμπτο κεφάλαιο παρουσιάζονται τα αποτελέσματα των μετρήσεων που έγιναν σε κάθε σενάριο και τοπολογία και γίνεται προσπάθεια για την ερμηνεία τους. Επιπλέον, παρατίθενται αναλυτικά τα συμπεράσματα που προκύπτουν από τη διπλωματική εργασία και γίνονται προτάσεις για πιθανές μελλοντικές μελέτες επί του θέματος.

Τέλος, ακολουθεί αναφορά σε κάθε πηγή που αξιοποιήθηκε για την σύνθεση της εργασίας.

Κεφάλαιο 2: Σχετικές δουλειές

2.1 Εισαγωγή

Σε αυτό το κεφάλαιο γίνεται αναφορά στο υπόβαθρο που πρέπει να έχει ο αναγνώστης της διπλωματικής εργασίας, ώστε να μπορέσει να την κατανοήσει. Η βαθύτερη κατανόηση των τεχνολογιών που παρουσιάζονται στις ενότητες αυτού του κεφαλαίου αποτέλεσαν και κρίσιμο σημείο για την τελική υλοποίηση της μελέτης. Αρχικά θα εστιάσουμε στις βασικές αρχές μιας αρχιτεκτονικής ICN που αποτελεί και την βάση της εργασίας και έπειτα θα αναλύσουμε τις υλοποιήσεις CCN και NDN αλλά και την αρχιτεκτονική hICN που αποτελεί μια νέα προέκταση της ICN.

2.2 ICN (Information-Centric Networking)

Σε αυτή την ενότητα θα εστιάσουμε στις βασικές αρχές της αρχιτεκτονικής ICN και πως αυτές σκοπεύουν να λύσουν κάποια από τα προβλήματα του σημερινού διαδικτύου. Πρωτού περάσουμε όμως σε αυτά, ας δούμε κάποιες πληροφορίες σχετικά με την πορεία που έχει πάρει το διαδίκτυο τα τελευταία χρόνια και το πώς θα μοιάζει στο κοντινό μέλλον.

Το Διαδίκτυο, όπως είναι γνωστό έχει μετατραπεί σε μια παγκόσμια δομή μαζικής διάδοσης πληροφοριών. Ενδεικτικά, κάποια προγνωστικά μιλούν για μεταφορά 4.8 zettabytes δεδομένων παγκοσμίως κάθε χρόνο από το έτος 2022, όταν το 2017 η τιμή αυτή ήταν 1.5 zettabytes. Ο αριθμός των δικτυωμένων συσκευών αναμένεται να φτάσει τα 28.5 δισεκατομμύρια μέχρι το 2022, όταν το 2017 ήταν 18 δισεκατομμύρια. Ακόμα, το 2022 προβλέπεται ότι μόνο το 29% της IP κίνησης παγκοσμίως θα είναι από ενσύρματες συσκευές, όπως υπολογιστές, ενώ το 71% της κίνησης θα είναι από ασύρματες και φορητές συσκευές. Το 2017 οι αντίστοιχες τιμές ήταν 48% για ενσύρματες και 52% για ασύρματες. Τέλος, όσον αφορά τον τύπο δεδομένων με τα μεγαλύτερα ποσοστά αυτός είναι η μετάδοση βίντεο. Το 2017 το 75% όλης της IP κίνησης ήταν μετάδοση βίντεο ενώ η τιμή αυτή υπολογίζεται ότι θα φτάσει το 82% όλης της IP κίνησης μέχρι το 2022.

2.2.1 Ονοματοδοσία πληροφοριών

Όπως είναι κατανοητό οι χρήστες του διαδικτύου ενδιαφέρονται όλο και περισσότερο στο να έχουν πρόσβαση σε πληροφορίες, δεδομένα και περιεχόμενα που επιθυμούν όπου και αν βρίσκονται αυτά, χωρίς να τους ενδιαφέρει ιδιαίτερα η επικοινωνία με κάποιον διακομιστή. Ωστόσο το μοντέλο επικοινωνίας που επικρατεί μέχρι και σήμερα απαιτεί από τον χρήστη να προσδιορίζει εκτός από τις πληροφορίες που επιθυμεί και τον συγκεκριμένο διακομιστή από τον οποίο θα τις πάρει. Αυτό συμβαίνει γιατί οι μηχανισμοί του επιπέδου δικτύου του σημερινού πρωτοκόλλου δεν μπορούν να εντοπίσουν και να φέρουν τις αιτούμενες πληροφορίες από την βέλτιστη τοποθεσία όπου φιλοξενούνται σε σχέση με τον χρήστη, εκτός αν ο ίδιος ο χρήστης γνωρίζει την τοποθεσία και αλλάξει τον διακομιστή με τον οποίο θέλει να επικοινωνήσει, πράγμα σπάνιο.

Βασικό εγχείρημα της ICN αρχιτεκτονικής είναι να διαχωρίσει την πληροφορία από την τοποθεσία που βρίσκεται, έτσι πλέον δίνεται στην πληροφορία ένα όνομα, ανεξάρτητα από την τοποθεσία της, δηλαδή μπορεί να βρίσκεται οπουδήποτε μέσα στο δίκτυο. Στην αρχιτεκτονική ICN, κατά την επικοινωνία, αντί να ορίζεται ένα ζευγάρι διευθύνσεων πηγής-προορισμού, κάθε κομμάτι της πληροφορίας έχει το δικό του όνομα και είναι προσπελάσιμο βάσει αυτού.

Το πρώτο βασικό συμπέρασμα που μπορούμε να βγάλουμε εδώ είναι ότι σε αντίθεση με την τρέχουσα αρχιτεκτονική όπου ο αποστολέας έχει τον πλήρη έλεγχο των δεδομένων που θα αποστείλει, στην ICN ο διακομιστής δεν μπορεί να στείλει δεδομένα εάν αυτά δεν έχουν πρώτα ζητηθεί ρητά από τον πελάτη. Το συγκεκριμένο μοντέλο μπορεί να χαρακτηριστεί και ως επικοινωνία τύπου “pull”, σε αντίθεση με την επικοινωνία τύπου “push” που πρακτικά συμβαίνει με τη σημερινή στοίβα πρωτοκόλλων. Με τη μεταβολή από ένα μοντέλο ονοματοδοσίας του εξυπηρετητή σε ένα μοντέλο ονοματοδοσίας της πληροφορίας η παραλαβή της πληροφορίας καθίσταται οδηγούμενη από τον παραλήπτη. Έτσι, μόνο αφού ο πελάτης στείλει ρητό αίτημα, το δίκτυο είναι υπεύθυνο για την εύρεση της βέλτιστης διαδρομής για τα επιθυμητά δεδομένα.

Είναι πολύ σημαντικό να αναφερθεί σε αυτό το σημείο ότι η δρομολόγηση των δεδομένων γίνεται αφού βρεθεί η καλύτερη ως προς τον πελάτη πηγή της πληροφορίας, με βάση το όνομα της πληροφορίας και ανεξάρτητα της τοποθεσίας της.

2.2.2 Παράδοση πληροφοριών

Στο διαδίκτυο του μέλλοντος, όπως προβλέπεται υπάρχει η απαίτηση για αποδοτική παράδοση μεγάλου όγκου πληροφοριών και χειρισμό των απότομων αυξήσεων στην κίνηση, γνωστών με τον αγγλικό όρο *flash crowds*. Η εμπειρία μας από το σημερινό διαδίκτυο μας έχει δείξει ότι οι απότομες αυξήσεις του πλήθους των κινήσεων προκαλούν προβλήματα συμφόρησης στο δίκτυο και καθιστούν την παράδοση των πληροφοριών μη αποτελεσματική.

Κατά την μετακίνηση πληροφοριών στο διαδίκτυο, τα δεδομένα μεταφέρονται από μια συγκεκριμένη πηγή (*source*) στον τελικό προορισμό (*destination*) και μεταχειρίζονται από το δίκτυο σαν μια σειρά από *bytes*. Έτσι τα διάφορα μέρη του δικτύου δεν γνωρίζουν τίποτα για τα δεδομένα που μεταφέρουν και αδυνατούν να πραγματοποιήσουν κάποιες κρίσιμες βελτιστοποιήσεις όπως για παράδειγμα αποθήκευση των δεδομένων μέσα στο δίκτυο για μελλοντική χρήση (*in-network caching*), αναπαραγωγή της πληροφορίας εάν αυτό είναι επιθυμητό (*information replication*) και κυκλοφορία με αντίληψη του περιεχομένου της πληροφορίας (*information-aware traffic*). Προκειμένου να εκμεταλλευτούν οι υπάρχουσες δυνατότητες αποθήκευσης μέσα στο δίκτυο, το διαδίκτυο θα πρέπει να επεκταθεί με κάποιους μηχανισμούς ενημέρωσης της πληροφορίας εντός του δικτύου, ώστε να γίνεται αναγνώριση της πληροφορίας και μετάδοση της από την βέλτιστη τοποθεσία.

Ένα παρόμοιο σύστημα με αυτό που περιγράφηκε παραπάνω είναι τα ήδη υπάρχοντα Content Delivery Networks (CDNs) [4] τα οποία βέβαια εφαρμόζουν αυτές τις τεχνικές σε επίπεδο εφαρμογής (OSI layer-7). Πρόκειται για δίκτυα διανομής του ζητούμενου περιεχομένου από αναπληρωματικούς διακομιστές αποθήκευσης (*surrogate servers*), οι οποίοι βρίσκονται πιο κοντά στον χρήστη

απ'ότι οι αρχικοί διακομιστές, με μοναδικό στόχο την γρηγορότερη και πιο αξιόπιστη εξυπηρέτηση των χρηστών. Ωστόσο σε περιπτώσεις flash crowds, το μέγεθος, η τοποθεσία και ο προορισμός της κίνησης δεν είναι δυνατόν να προβλεφθούν πάντοτε και πέραν αυτού η επένδυση για ένα CDN δεν είναι η πλέον βιώσιμη, ειδικά με την συνεχή αύξηση του όγκου των πληροφοριών που δημιουργούνται από τους χρήστες. Επιπλέον, διάφορες αναλύσεις που έχουν γίνει σε Open CDNs δείχνουν ότι μερικές φορές το πλήθος των κινήσεων αυξάνει τόσο γρήγορα που ο καταμερισμός των πόρων σε επίπεδο εφαρμογής αδυνατεί να ακολουθήσει αυτή την αύξηση.

Στην αρχιτεκτονική ICN το δίκτυο μπορεί να ικανοποιήσει ένα αίτημα για πληροφορία όχι μόνο από την εύρεσή της στην αρχική πηγή της πληροφορίας, αλλά και ανατρέχοντας στις κρυφές μνήμες που υπάρχουν μέσα στο δίκτυο και που κρατούν αντίγραφα της επιθυμητής πληροφορίας ή κομμάτια αυτής. Αυτό συμβαίνει γιατί η αρχιτεκτονική σε επίπεδο δικτύου (OSI layer-3) λειτουργεί βασισμένη σε ονομασμένες πληροφορίες. Μάλιστα προς επίτευξη ενός επιπέδου διαφάνειας, τα ονόματα που δίνονται στα πακέτα πληροφοριών βασίζονται στην πληροφορία που περιέχουν.

Με την τεχνική της ονοματοδοσίας της πληροφορίας υπάρχει επιπλέον το πλεονέκτημα της συσσώρευσης αιτημάτων για μια συγκεκριμένη πληροφορία που διευκολύνει και κάνει αποδοτικότερη την παράδοση αφού τα αιτήματα μπορούν να εξυπηρετηθούν ταυτόχρονα (multicast forwarding). Ακόμα, οι έλεγχοι πρόσβασης, δηλαδή ποιος έχει πρόσβαση σε τι, μπορούν να εφαρμοστούν σε επίπεδο δικτύου αφού τα μέρη του δικτύου γνωρίζουν τις πληροφορίες περιέχεται σε κάθε πακέτο.

2.2.3 Ευκινησία

Πριν από μερικές δεκαετίες, όταν το διαδίκτυο ήταν ακόμη υπό μελέτη και κατασκευή, οι σχεδιαστές του δημιούργησαν το μοντέλο διευθυνσιοδότησης έχοντας υπόψιν τους ότι η τοποθεσία των χρηστών παραμένει σταθερή, το ίδιο και οι IP διευθύνσεις τους. Φυσικά κάτι τέτοιο δεν ισχύει σήμερα αφού όπως προείπαμε οι ασύρματες συνδέσεις έχουν ξεπεράσει κατά πολύ τις ενσύρματες.

Οι ασύρματες φορητές συσκευές ωστόσο αλλάζουν συχνά δίκτυα και IP διευθύνσεις κι έτσι χρειάζονται νέους τρόπους επικοινωνίας που να επιτυγχάνουν την συνεχή σύνδεση τους στο διαδίκτυο όταν μετακινούνται χωρίς να προκύπτουν προβλήματα.

Το πρωτόκολλο Mobile IP, το οποίο δημιουργήθηκε ακριβώς για να λύσει το πρόβλημα της ευκινησίας στο τρέχον διαδίκτυο χρησιμοποιεί την τεχνική της τριγωνικής δρομολόγησης (Triangle Routing) [5] και αυτό είναι ένα από τα βασικά της προβλήματα. Εν συντομία, η τεχνική λειτουργεί ως εξής: όταν ένα πακέτο πρέπει να δρομολογηθεί για το μετακινούμενο χρήστη, πρώτα πάει σε έναν “πράκτορα” που βρίσκεται στο δίκτυο του χρήστη (Home agent), από εκεί μεταφέρονται μέσω τούνελ (tunneling) σε έναν ξένο “πράκτορα” (Foreign Agent) που βρίσκεται στο δίκτυο που είναι συνδεδεμένος εκείνη την στιγμή ο χρήστης και έπειτα στο μετακινούμενο χρήστη. Όπως είναι κατανοητό η τεχνική αυτή είναι μη αποδοτική καθώς η διαδρομή που ακολουθείται για την παράδοση της πληροφορίας είναι πολύ μεγαλύτερη της βέλτιστης και το πρόβλημα γίνεται ακόμη πιο έντονο όσο οι αποστάσεις μεγαλώνουν.

Ένα δεύτερο πρόβλημα στο Mobile IP πρωτόκολλο εντοπίζεται στο γεγονός ότι τείνει να παραβιάζει την “valley free” [6] πολιτική δρομολόγησης του Border Gateway Protocol (BGP), αφού τα πακέτα δρομολογούνται πρώτα στο home agent του μετακινούμενου χρήστη και έπειτα επαναδρομολογούνται στο δίκτυο που βρίσκεται την δεδομένη χρονική στιγμή.

Στην αρχιτεκτονική ICN το θέμα της ευκινησίας επιλύεται με την εφαρμογή ενός μοντέλου επικοινωνίας δημοσίευσης/συνδρομής (publish/subscribe model). Γενικότερα, το μοντέλο δημοσίευσης/συνδρομής βασίζεται στους χρήστες που ενδιαφέρονται για μια πληροφορία και ενημερώνουν το δίκτυο για το ενδιαφέρον τους και στους χρήστες που προσφέρουν πληροφορίες για κατανάλωση. Εντός του δικτύου, υπάρχουν και οι διαμεσολαβητές που είναι υπεύθυνοι για την αντιστοίχιση της κάθε συνδρομής στην κατάλληλη δημοσίευση. Πιο συγκεκριμένα στην ICN το μοντέλο λειτουργεί ως εξής: η δημοσίευση πραγματοποιείται με την μορφή ανακοίνωσης των διαθέσιμων πληροφοριών στο δίκτυο ενώ η συνδρομή αναφέρεται σε ήδη δημοσιευμένες πληροφορίες.

Το μεγάλο πλεονέκτημα που προσφέρει το μοντέλο επικοινωνίας δημοσίευσης/συνδρομής έγκειται στο γεγονός ότι οι ενέργειες της δημοσίευσης και της συνδρομής διαχωρίζονται χρονικά, δηλαδή οι δύο ενέργειες δεν χρειάζεται να είναι συγχρονισμένες χρονικά. Αυτό σημαίνει ότι ο συνδρομητής μπορεί να ζητήσει κάποια πληροφορία πολύ μετά από την ανακοίνωση της δημοσίευσης της. Επιπλέον, συνήθως όσοι δημοσιεύουν δεν κρατούν αναφορές από τους συνδρομητές τους ούτε και γνωρίζουν πόσοι συνδρομητές λαμβάνουν την δημοσίευσή τους. Το ίδιο συμβαίνει και από την πλευρά των συνδρομητών που δεν γνωρίζουν πόσοι δημοσιεύουν την πληροφορία που αναζητούν. Αυτή η συνθήκη διευκολύνει την αποδοτική ευκινησία καθώς οι κινητοί κόμβοι μπορούν πολύ εύκολα να επανεκδώσουν «συνδρομές» μετά από κάποια μετακίνησή τους και το δίκτυο μπορεί να εξυπηρετήσει αυτές τις συνδρομές από κοντινότερες κρυφές μνήμες αντί για τον αρχικό κόμβο δημοσίευσης.

2.2.4 Ασφάλεια

Ενώ αρχικά το διαδίκτυο σχεδιάστηκε ώστε να λειτουργεί σαν ένα αξιόπιστο περιβάλλον, σήμερα η εικόνα έχει αλλάξει. Τότε ήταν ανοιχτό και εύκαμπτο στην συμμετοχή περισσότερων χρηστών, ενώ σήμερα η πιστοποίηση χρηστών και δεδομένων και η ιδιωτικότητα των χρηστών είναι βασικές απαιτήσεις. Ανέκαθεν ωστόσο το διαδίκτυο προωθούσε οποιαδήποτε κυκλοφορία στο δίκτυο και αυτό αποτελούσε ευκαιρία για κακόβουλους χρήστες ώστε να επιτίθενται (π.χ Denial of Service ή DoS) σε άλλους χρήστες και υπηρεσίες, ενώ κάλυπταν εύκολα τα ίχνη τους. Έτσι προς ενδυνάμωση της ασφάλειας αναπτύχθηκαν πολλοί μηχανισμοί όπως τείχη προστασίας (firewalls) και φίλτρα εμπιστοσύνης (trust filters), αλλά και πρωτόκολλα ασφαλείας όπως το IPSec και το DNSSec. Βέβαια αυτοί οι μηχανισμοί αποτελούν μόνο πρόσθετες διορθώσεις του λογισμικού ασφαλείας και δεν λύνουν το πρόβλημα καθώς δεν καλύπτουν τα κατώτερα επίπεδα του δικτύου και έτσι τα κακόβουλα δεδομένα εξακολουθούν να προωθούνται και κωλύουν το σύστημα ή ακόμα και να ξεγελούν τους μηχανισμούς φιλτραρίσματος.

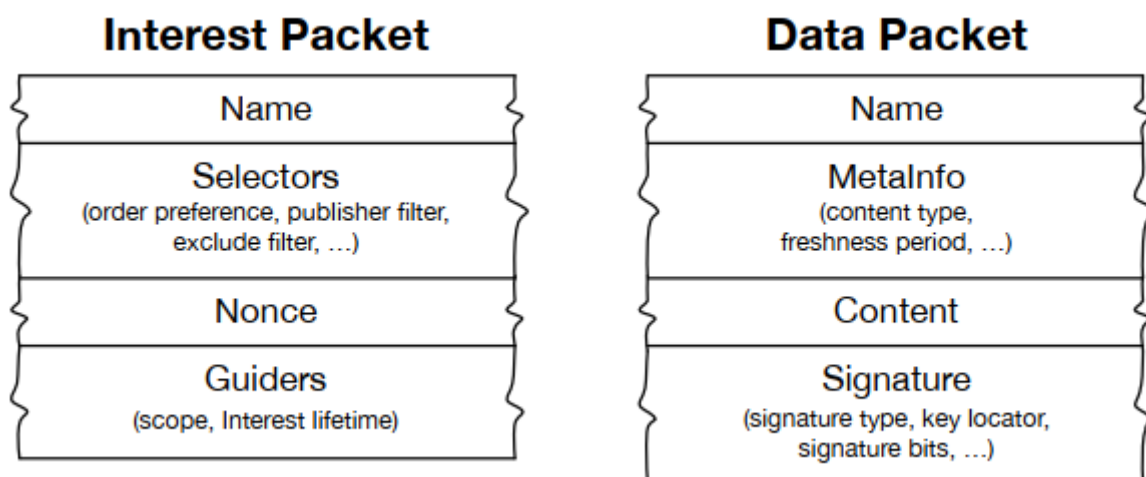
Στην αρχιτεκτονική ICN το μοντέλο επικοινωνίας είναι οδηγούμενο από το ενδιαφέρον για πληροφορία, δηλαδή δεν υπάρχει ροή δεδομένων εκτός και αν ο χρήστης ζητήσει κάποια πληροφορία. Το γεγονός αυτό προφανώς μειώνει σημαντικά τον όγκο των ανεπιθύμητων μεταφορών δεδομένων και επιπλέον διευκολύνει την εφαρμογή μηχανισμών στο δίκτυο οι οποίοι χειρίζονται τα σήματα ενδιαφέροντος και διαθεσιμότητας. Επιπρόσθετα, σε ICN υλοποιήσεις με αυτοπιστοποιημένα ονόματα για τις πληροφορίες το φιλτράρισμα των κακόβουλων δεδομένων μπορεί να γίνεται και από μηχανισμούς εντός του δικτύου. Ακόμα, οι περισσότερες υλοποιήσεις της ICN περιλαμβάνουν έναν ενδιάμεσο ανάμεσα στους χρήστες που ζητούν μια πληροφορία και στους χρήστες που την κατέχουν, διαχωρίζοντας έτσι την επικοινωνία των δύο χρηστών. Αυτός ο διαχωρισμός αυτόματα καταπολέμα τις επιθέσεις Denial of Service, αφού τα αιτήματα για την πληροφορία μπορούν να αξιολογηθούν στον ενδιάμεσο κόμβο πρώτου φτάσουν στον τελικό προορισμό τους. Τέλος, η χρήση ενδιάμεσου κόμβου ενισχύει και την ιδιωτικότητα του χρήστη καθώς ο τελικός εκδότης της πληροφορίας δεν χρειάζεται να γνωρίζει ποιός είναι ο αιτών την πληροφορία.

2.3 CCN/NDN (Content-Centric Networking / Named Data Networking)

Αυτά τα δύο μοντέλα έχουν τη βάση τους στην προαναφερθείσα αρχιτεκτονική ICN και ουσιαστικά αποτελούν υλοποιήσεις αυτής. Λόγω των πολλών κοινών χαρακτηριστικών μεταξύ τους, αφού η NDN είναι η συνέχεια της CCN, συνηθίζεται να αντιμετωπίζονται σαν μια. Έτσι οι αρχές λειτουργίας που θα αναφέρουμε παρακάτω διέπουν και τις δύο υλοποιήσεις, παρόλο που εμείς θα αναφερθούμε στην υλοποίηση NDN.

2.3.1 Η αρχιτεκτονική

Στην NDN [7] υλοποίηση υπάρχουν δύο ειδών τερματικοί κόμβοι: οι παραγωγοί (producers) και οι καταναλωτές ή παραλήπτες (consumers). Η επικοινωνία είναι οδηγούμενη από τους καταναλωτές μέσω δύο τύπων πακέτων, τα πακέτα ενδιαφέροντος (Interest packets) και τα πακέτα δεδομένων (Data packets). Οι δύο τύποι πακέτων φέρουν ένα όνομα που προσδιορίζει ένα κομμάτι δεδομένων που μπορεί να μεταδοθεί σε ένα πακέτο δεδομένων. Σε μια τυπική επικοινωνία, ένας καταναλωτής δηλώνει το όνομα της πληροφορίας που επιθυμεί σε ένα πακέτο ενδιαφέροντος και το αποστέλλει στο δίκτυο. Έπειτα οι δρομολογητές (routers) προωθούν το πακέτο αυτό με βάση το όνομα του στον κατάλληλο παραγωγό ή παραγωγούς. Όταν το πακέτο ενδιαφέροντος φτάσει στον κατάλληλο κόμβο παραγωγό, αυτός θα επιστρέψει ένα πακέτο δεδομένων που θα περιέχει τόσο το όνομα όσο και το περιεχόμενο, μαζί με μια υπογραφή με το κλειδί του παραγωγού που συνδέει αυτά τα δύο. Το πακέτο δεδομένων θα ακολουθήσει ακριβώς την αντίστροφη διαδρομή ώστε να επιστρέψει στον καταναλωτή που το ζήτησε.



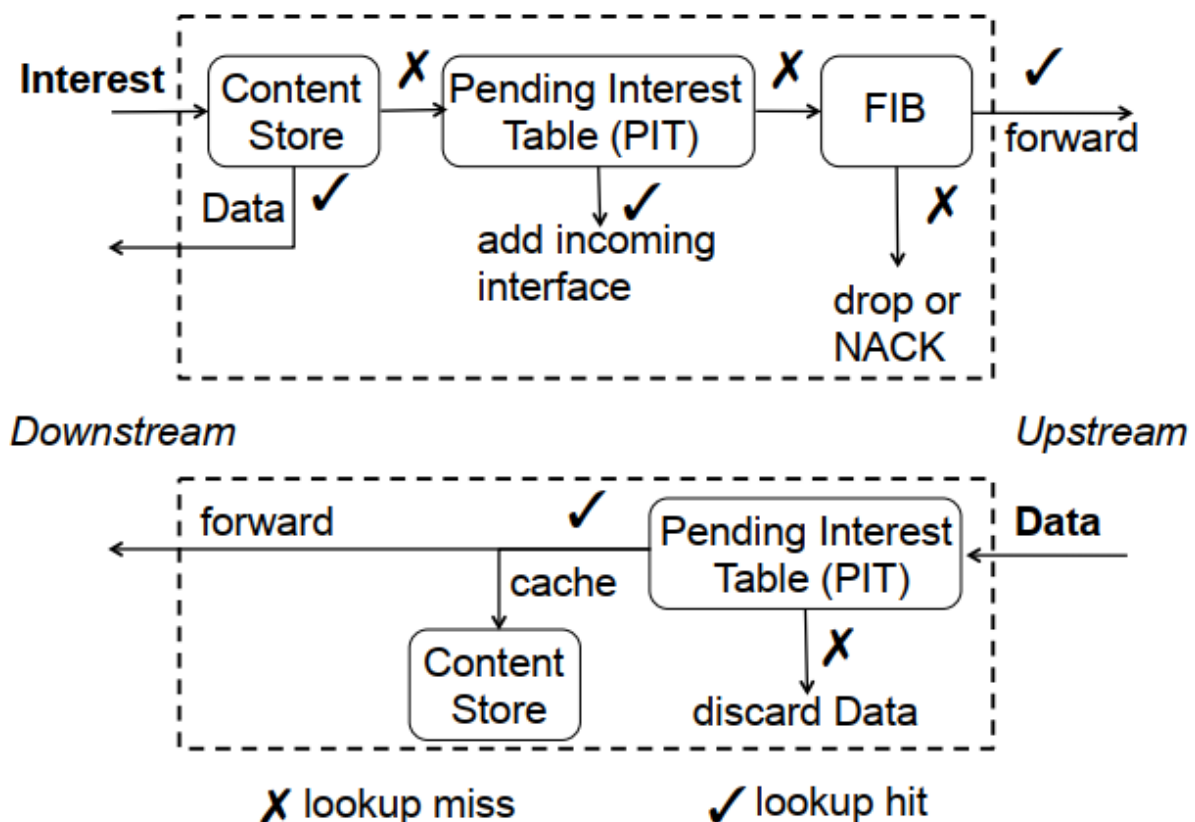
Εικόνα 1: Πακέτα NDN αρχιτεκτονικής

Η πολύ καλώς σχεδιασμένη αρχιτεκτονική της NDN επιτρέπει τη χρήση της και ως βάση ανάπτυξης νέων αρχιτεκτονικών, όπως η *UMOBILE* [8]. Η *UMOBILE* προσαρμόζει το πληροφοριοκεντρικό μοντέλο επικοινωνίας ώστε να ανταποκριθεί στις απαιτήσεις των ευκαιριακών επικοινωνιών. Ένα από τα

αξιοσημείωτα οράματα της *UMOBILE* είναι η επίτευξη επικοινωνίας σε μέρη που υπάρχουν τεχνικές δυσκολίες ασύρματης σύνδεσης, όπως σε δάση και βουνά, ή ακόμα και μέρη στην καρδιά των πόλεων όπως το μετρό. Επιπλέον, με βάση τη *UMOBILE* αρχιτεκτονική έχουν αναπτυχθεί υπηρεσίες άκρων [9] με ανεκτικότητα στην καθυστέρηση (Delay Tolerant) που εξυπηρετούν σε περιπτώσεις φυσικών καταστροφών, όπου το δίκτυο ενδέχεται να αντιμετωπίζει υπερβολική συμφόρηση, καθιστώντας τη ραχοκοκαλιά μη λειτουργική.

2.3.2 Διαδικασία προώθησης

Για την διαδικασία προώθησης των πακέτων ενδιαφέροντος και δεδομένων, κάθε NDN δρομολογητής διατηρεί μια στρατηγική προώθησης (Forwarding Strategy) και τρεις δομές δεδομένων οι οποίες φαίνονται στην επόμενη εικόνα.



Εικόνα 2: Διαδικασία προώθησης σε NDN κόμβο

Πρόκειται για τις δομές *Pending Interest Table (PIT)*, *Forwarding Information Base (FIT)* και *Content Store (CS)*. Στη PIT αποθηκεύονται όλα τα ενδιαφέροντα που έχει προωθήσει ένας δρομολογητής αλλά δεν έχουν ικανοποιηθεί ακόμα. Κάθε καταχώρηση στη PIT καταγράφει το όνομα πακέτου που μεταφέρεται στο δίκτυο, μαζί με την εισερχόμενη και εξερχόμενη διεπαφή. Κατά την άφιξη ενός πακέτου ενδιαφέροντος, ο NDN δρομολογητής αρχικά ελέγχει τη CS για αντιστοιχία δεδομένων. Αν εκεί υπάρχει ταιριαστό πακέτο δεδομένων ο δρομολογητής το επιστρέφει μέσω της διεπαφής στην οποία έλαβε το πακέτο ενδιαφέροντος, διαφορετικά αναζητά το όνομα στη PIT του και αν υπάρχει αντίστοιχη καταχώρηση απλώς καταγράφει την εισερχόμενη διεπαφή του παρόντος ενδιαφέροντος στη PIT. Σε περίπτωση που δεν υπάρχει αντίστοιχη καταχώρηση στη PIT, ο δρομολογητής προωθεί το ενδιαφέρον προς τους παραγωγούς δεδομένων βάσει των πληροφοριών της FIB καθώς και της στρατηγικής προώθησης του δρομολογητή. Έτσι όταν ένας δρομολογητής λαμβάνει πολλά ενδιαφέροντα για το ίδιο όνομα από διαφορετικούς κόμβους, χρειάζεται να προωθήσει μόνο το πρώτο προς τους παραγωγούς δεδομένων. Η δομή FIB συμπληρώνεται από ένα πρωτόκολλο δρομολόγησης που βασίζεται στο πρόθεμα ονόματος και μπορεί να έχει πολλαπλές διεπαφές εξόδου για το κάθε πρόθεμα.

Η στρατηγική προώθησης (*Forwarding Strategy*) μπορεί να αποφασίσει να απορρίψει ένα ενδιαφέρον σε ορισμένες καταστάσεις, για παράδειγμα εάν οι συνδέσεις είναι συμφορημένες ή το ενδιαφέρον θεωρηθεί μέρος μιας DoS επίθεσης. Για κάθε ενδιαφέρον, η στατηγική προώθησης επιλέγει την ταιριστή καταχώρηση με το μεγαλύτερο πρόθεμα ονόματος και αποφασίζει πότε και πού να προωθήσει το ενδιαφέρον. Η δομή CS είναι μια προσωρινή κρυφή μνήμη των πακέτων δεδομένων που έχει λάβει ο δρομολογητής. Επειδή ακριβώς ένα πακέτο δεδομένων στη NDN υλοποίηση έχει νόημα ανεξάρτητα από πού έρχεται ή από πού προωθείται, αυτό αποθηκεύεται προσωρινά ώστε να ικανοποιηθεί και πιθανά μελλοντικά ενδιαφέροντα.

Όταν καταφθάνει ένα πακέτο δεδομένων, ο NDN δρομολογητής εντοπίζει την αντίστοιχη καταχώρηση στη PIT και προωθεί τα δεδομένα σε όλες τις διεπαφές που εξέφρασαν ενδιαφέρον για τη συγκεκριμένη πληροφορία. Έπειτα

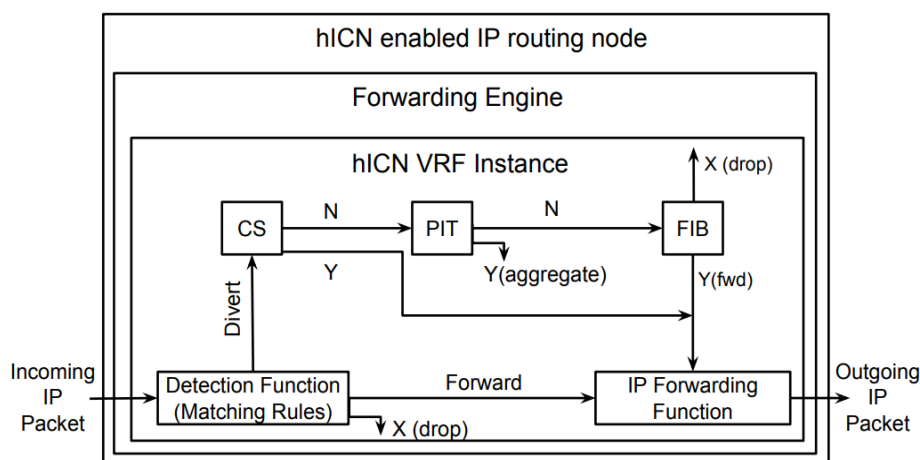
αφαιρεί αυτή την καταχώρηση από τη PIT και αποθηκεύει το πακέτο δεδομένων στη CS. Τα πακέτα δεδομένων ακολουθούν πάντα την αντίθετη πορεία από αυτή των πακέτων ενδιαφέροντος και στη βέλτιστη περίπτωση που δεν υπάρχει απώλεια πακέτων, ένα πακέτο ενδιαφέροντος οδηγεί σε ένα πακέτο δεδομένων παρέχοντας έτσι ισορροπία στη ροή πακέτων. Για την ανάκτηση μεγάλου όγκου περιεχομένου που περιλαμβάνει πολλά πακέτα, τα πακέτα ενδιαφέροντος λαμβάνουν στη ροή της κυκλοφορίας ένα ρόλο παρόμοιο με των TCP ACKs στο σημερινό διαδίκτυο. Τα πακέτα ενδιαφέροντος δεν φέρουν διευθύνσεις αλλά δρομολογούνται από το δρομολογητή στους παραγωγούς με βάση τα ονόματά τους, ενώ τα πακέτα δεδομένων δρομολογούνται πίσω στους καταναλωτές με βάση τις καταχωρήσεις στη PIT που δημιουργήθηκαν προηγουμένως από τα αντίστοιχα πακέτα ενδιαφέροντος. Αυτή η συμμετρία ανταλλαγής πακέτων δημιουργεί έναν βήμα-προς-βήμα βρόγχο ελέγχου και εξαλείφει την ανάγκη για κόμβους πηγής-προορισμού σε αντίθεση με το σημερινό διαδίκτυο.

2.4 hICN (hybrid Information-Centric Networking)

Σε αυτή την ενότητα θα αναλύσουμε πώς η αρχιτεκτονική hICN εφαρμόζει διάφορες λειτουργίες της ICN στο πρωτόκολλο IP και έτσι εκμεταλλεύεται όλη την υποδομή του IP πρωτοκόλλου. Θα εστιάσουμε στην ονοματοδοσία, στην μορφή των πακέτων, στην αποθήκευση των πακέτων στην κρυφή μνήμη, στην προώθηση των πακέτων, στους τελικούς προορισμούς και στην ασφάλεια.

2.4.1 Αρχιτεκτονική

Βασικές προϋποθέσεις για τον συνδυασμό της ICN αρχιτεκτονικής με το IP πρωτόκολλο είναι η ύπαρξη των γνωστών IP δρομολογητών, η ύπαρξη hICN δρομολογητών εφοδιασμένων με την εφαρμογή του παραγωγού (producer) ή του καταναλωτή (consumer) αλλά και η τροποποίηση δρομολογητών ώστε να μπορούν να επεξεργάζονται και να προωθούν τόσο κανονικά IP πακέτα όσο και IP πακέτα ενισχυμένα με στοιχεία της ICN. Στους τροποποιημένους δρομολογητές όταν εισέρχεται ένα πακέτο αναλύεται η IP επικεφαλίδα και αναγνωρίζεται πώς θα μεταχειριστεί το πακέτο, όπως φαίνεται και στην παρακάτω εικόνα.



Εικόνα 3: Εσωτερική αρχιτεκτονική hICN κόμβου

Για την λειτουργία της hICN αρχιτεκτονικής [10] υπάρχουν οι παρακάτω απαιτήσεις :

- **Προώθηση** - οι τροποποιημένοι με hICN δρομολογητές όπως και οι IP δρομολογητές χρησιμοποιούν τα ίδια τμήματα προώθησης.
- **Αποθήκευση** - η κρυφή μνήμη είναι διαθέσιμη στους τροποποιημένους με hICN δρομολογητές και τα αιτήματα μπορούν να εξυπηρετηθούν αν το περιεχόμενο που ζητείται είναι διαθέσιμο εκεί.
- **Διαχείριση** - για χρήση της αρχιτεκτονικής σε μεγάλης κλίμακας συστήματα ο καταναλωτής και ο παραγωγός πρέπει να έχουν μια συσχέτιση των ονομάτων των περιεχομένων με τις αντίστοιχες IP διευθύνσεις, ώστε τα ICN πακέτα να μπορούν να επεξεργαστούν και από μη τροποποιημένους δρομολογητές.
- **Ασφάλεια** - η αρχιτεκτονική παρέχει τις ίδιες δυνατότητες ασφαλείας με την ICN αρχιτεκτονική. Ωστόσο, οι IP δρομολογητές δεν μπορούν να ελέγξουν τα πακέτα δεδομένων ως προς την ακεραιότητα και την αυθεντικότητα τους. Για τον λόγο αυτό τουλάχιστον ένας τροποποιημένος με hICN δρομολογητής θα πρέπει να υπάρχει στη διαδρομή μεταξύ καταναλωτή και παραγωγού.

Όσον αφορά την αρχιτεκτονική, στο επίπεδο δικτύου (network layer) συμπεριλαμβάνεται η επιφάνεια προώθησης (forwarding plane) αλλά όχι η επιφάνεια δρομολόγησης (routing plane). Το επίπεδο δικτύου (network layer) στην hICN χρησιμοποιεί έναν IPv6 πίνακα πληροφοριών προώθησης ή όπως είναι ευρύτερα γνωστό IPv6 FIB για να καθορίσει τον επόμενο δρομολογητή στον οποίο θα προωθήσει το αίτημα ή χρησιμοποιεί την τοπική κρυφή μνήμη για να διαπιστώσει αν ένα αίτημα μπορεί να εξυπηρετηθεί τοπικά. Η κρυφή μνήμη είναι ένα απαραίτητο στοιχείο που προστίθεται στο υπάρχον IPv6 μέσο διάδοσης (packet pipeline) για επεξεργασία πακέτων. Το μέσο διάδοσης των πακέτων σε έναν κόμβο της hICN πάντοτε συμπεριλαμβάνει μια αναζήτηση στην κρυφή μνήμη τόσο για αιτήματα όσο και για την απαιτούμενη απάντηση σε αιτήματα.

Η αρχιτεκτονική hICN μπορεί να τρέξει σε κάθε επίπεδο ζεύξης δεδομένων (OSI layer-2) που υποστηρίζει το πρωτόκολλο IPv6. Τα ονόματα των δεδομένων είναι εφικτό να δρομολογηθούν παγκόσμια και είναι ορατά στο επίπεδο εφαρμογής των τελικών προορισμών. Αντιθέτως, το επίπεδο εφαρμογής δε γνωρίζει τις διευθύνσεις των διεπαφών δικτύου (network interfaces). Το επίπεδο δικτύου προωθεί δύο ειδών μονάδες δεδομένων, τα πακέτα ενδιαφέροντος (αίτημα για πληροφορία) και τα πακέτα δεδομένων (απάντηση σε αίτημα).

Ένας τελικός κόμβος είναι πάντοτε συνδεδεμένος σε ένα μονής κατεύθυνσης κανάλι που χρησιμοποιείται είτε για να στείλει είτε για να λάβει δεδομένα. Αν στέλνει δεδομένα ο τελικός κόμβος ονομάζεται παραγωγός (producer) ενώ αν λαμβάνει δεδομένα ονομάζεται καταναλωτής (consumer). Ένας κόμβος παραγωγός, παράγει δεδομένα με ένα ανεξάρτητο της τοποθεσίας όνομα το οποίο είναι ένα IPv6 πρόθεμα (IPv6 prefix). Ο καταναλωτής απορροφά τα δεδομένα χρησιμοποιώντας το ακριβές όνομα που δόθηκε από τον παραγωγό. Οι υπηρεσίες της παραγωγής και της κατανάλωσης ξεκινούν στην εφαρμογή ανοίγοντας υποδοχές επικοινωνίας (communication sockets) με ένα API για την εκτέλεση των βασικών λειτουργιών μεταφοράς δεδομένων: κατανομής, αρχικοποίησης, διαμόρφωσης, μετάδοσης και λήψης.

Οι υποδοχές επικοινωνίας του παραγωγού και του καταναλωτή μπορούν να υλοποιήσουν διάφορετικούς τύπους υπηρεσιών όπως stream ή datagram. Η διαφορά των stream sockets από τα datagram sockets είναι ότι τα stream παρέχουν αφοσιωμένο και από άκρο σε άκρο κανάλι μεταξύ εξυπηρετητή και πελάτη, χρησιμοποιούν το TCP πρωτόκολλο για την μετάδοση των πληροφοριών, είναι αξιόπιστα και χωρίς απώλειες, τα δεδομένα φτάνουν στον προορισμό με την ίδια σειρά που στάλθηκαν ωστόσο υπάρχει καθυστέρηση σε περίπτωση που υπάρχουν χαμένα/λανθασμένα δεδομένα. Τα datagram sockets αντίθετα δεν παρέχουν αφοσιωμένο και από άκρο σε άκρο κανάλι μεταξύ εξυπηρετητή και πελάτη, χρησιμοποιούν το UDP πρωτόκολλο για την μετάδοση των πληροφοριών, δεν είναι απόλυτα αξιόπιστα και μπορεί να υπάρχουν απώλειες, τα δεδομένα δεν φτάνουν στον προορισμό με την ίδια σειρά που στάλθηκαν ωστόσο δεν υπάρχει καθυστέρηση ή πολλές φορές και ενδιαφέρον σε περίπτωση που υπάρχουν χαμένα/λανθασμένα δεδομένα. Βασικό κριτήριο για την επιλογή χρήσης κάθε υπηρεσίας είναι η φύση της εφαρμογής, για παράδειγμα θα επιλέγαμε stream sockets για την μεταφορά αρχείων (File transfer) ενώ θα επιλέγαμε datagram sockets για τηλεφωνία μέσω διαδικτύου (VoIP).

Σε κάθε περίπτωση οι υπηρεσίες μεταφοράς είναι άνευ σύνδεσης (connection-less), δηλαδή η υπηρεσία του παραγωγού παράγει ονομασμένα δεδομένα σε μια υποδοχή μνήμης (socket memory) η οποία είναι προσβάσιμη από κάθε έγκυρο αίτημα από έναν ή και παραπάνω καταναλωτές. Ο καταναλωτής με τη σειρά του ανακτά ονομασμένα δεδομένα χρησιμοποιώντας ανεξάρτητα της τοποθεσίας ονόματα τα οποία δεν συνδέονται με κάποιο αναγνωριστικό διεπαφής (locator). Αυτό το μοντέλο επιτρέπει την αξιόπιστη φορητότητα του καταναλωτή χωρίς κάποιο πρωτόκολλο διαχείρισης φορητότητας. Η hICN επιπλέον επιτρέπει την επικοινωνία χρηστών που είναι συνδεδεμένη σε περισσότερα από ένα δίκτυα χωρίς κάποια ιδιαίτερη μεταχείριση σε επίπεδο μεταφοράς (OSI layer-4) ενώ σε επίπεδο δικτύου (OSI layer-3) μπορεί να εφαρμοστεί προώθηση πολλαπλών διαδρομών (multi-path forwarding) σε IPv6 δίκτυα καθώς οι ισορροπημένες ροές αιτημάτων/απαντήσεων αυτοσταθεροποιούν την συμμόρφωση του δικτύου.

2.4.2 Τελικοί κόμβοι

Όπως έχει προαναφερθεί, στην αρχιτεκτονική έχουμε δύο ειδών τελικούς κόμβους, τους παραγωγούς (producers) και τους καταναλωτές (consumers) [11]. Έτσι διακρίνουμε δύο τύπους υποδοχέων επικοινωνίας (communication sockets) ο καθένας με ένα συγκεκριμένο API: υποδοχέας παραγωγού και υποδοχέας καταναλωτή. Αυτοί οι υποδοχείς είναι σχεδιασμένοι ώστε να ανταλλάσουν δεδομένα από πολλαπλά σημεία σε πολλαπλά σημεία. Στην hICN υπάρχει η γενική ιδέα της ύπαρξης μηνυμάτων κατανεμημένων σε όλη την διαδρομή της επικοινωνίας. Η πρώτη μνήμη της διαδρομής είναι αυτή του παραγωγού, στην οποία δημιουργούνται πακέτα δεδομένων και φιλοξενούνται κάτω από έναν συγκεκριμένο χώρο ονομάτων. Ο καταναλωτής μπορεί να ανακτήσει αυτά τα δεδομένα από την μνήμη μέσω του επιπέδου δικτύου της hICN.

Ο τρόπος με το οποίο παραγωγοί και καταναλωτές συγχρονίζονται εξαρτάται από τις απαιτήσεις της εφαρμογής και το επίπεδο μεταφοράς (OSI layer-4) διαθέτει ποικιλία υπηρεσιών για παράδειγμα stream/datagram, αξιόπιστα/αναξιόπιστα, με ή χωρίς χειρισμό καθυστέρησης. Ανεξάρτητα των απαιτήσεων της εφαρμογής, ο υποδοχέας του παραγωγού πάντοτε εκτελεί κατάτμηση (segmentation) των δεδομένων από το ανώτερο επίπεδο και υπολογίζει ψηφιακές υπογραφές στον φάκελο ασφαλείας των πακέτων. Αυτός ο φάκελος μπορεί να υπολογιστεί και για μια ομάδα πακέτων συμπεριλαμβάνοντας το κρυπτογραφικό hash κάθε πακέτου στο μανιφέστο μεταφοράς και τελικώς υπογράφοντας μόνο το μανιφέστο.

Ο υποδοχέας του καταναλωτή πάντοτε επανασυρμολογεί τα πακέτα δεδομένων, εκτελεί έλεγχο ακεραιότητας του hash και επαλήθευση της υπογραφής. Συνήθως, ο υποδοχέας του καταναλωτή χρησιμοποιεί ταυτότητα που δύναται να επαληθευτεί όταν έχει πρόσβαση σε χώρους ονομάτων που έχουν εκχωρηθεί. Ο υποδοχέας του παραγωγού πρέπει να είναι σε θέση να διαχειριστεί την χαρτογράφηση της ταυτότητας του και του διαθέσιμου χώρου ονομάτων, εκδίδοντας ψηφιακά πιστοποιητικά για την χαρτογράφηση. Από την άλλη πλευρά, ο υποδοχέας του καταναλωτή πρέπει να ανακτήσει τα σχετικά πιστοποιητικά για να εκτελέσει βασικές λειτουργίες.

2.4.3 Ονοματοδοσία

Η ονοματοδοσία στην hICN [11] περιλαμβάνει δύο βασικά κομμάτια: το πρόθεμα του ονόματος (name prefix) και το επίθεμα του ονόματος (name suffix). Το πρόθεμα χρησιμεύει στην αναγνώριση ενός αντικείμενου εφαρμογής, μιας υπηρεσίας ή γενικά μιας πηγής δεδομένων σε επίπεδο εφαρμογής μέσα στο δίκτυο και ενσωματώνεται στο σύστημα μέσω ενός υποδοχέα που δεσμεύεται στο πρόθεμα του ονόματος. Το επίθεμα χρησιμοποιείται για να υποδείξει τα επιμέρους κατακερματισμένα δεδομένα στο πεδίο του προθέματος που χρησιμοποιεί η εφαρμογή. Για παράδειγμα, ένας HTTP εξυπηρετητής υπακούει σε ένα πρόθεμα ονόματος για να εξυπηρετήσει HTTP αιτήματα. Μια απάντηση HTTP μεγάλη σε όγκο δεδομένων θα χρειαστεί να κατακερματιστεί από το επίπεδο μεταφοράς σε μονάδες δεδομένων σύμφωνα με κάποιο MTU, δηλαδή με κάποιο μέγιστο όριο μεγέθους μονάδας. Τα επιθέματα θα κατατάξουν κάθε κομμάτι στον υποδοχέα για μετάδοση.

2.4.3.1 Πρόθεμα ονόματος

Η μορφή του προθέματος ονόματος στην hICN φαίνεται παρακάτω:

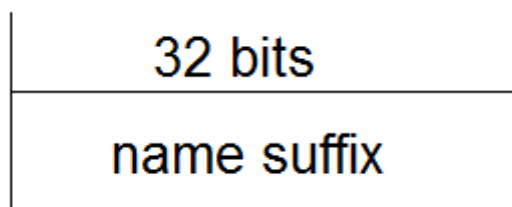
64 bits	64 bits
routable prefix	data identifier

Εικόνα 4: hICN πρόθεμα ονόματος

Αποτελείται από ένα δρομολογούμενο πρόθεμα IPv6 το οποίο πρέπει να είναι παγκόσμια δρομολογούμενο. Επιπλέον περιέχει ένα αναγνωριστικό των δεδομένων κωδικοποιημένο σε 64 bits. Κάθε εφαρμογή μπορεί να χρησιμοποιήσει διάφορα αναγνωριστικά αν αυτό χρειάζεται. Έτσι προκύπτει ένα πρόθεμα ονόματος, ανεξάρτητο της τοποθεσίας, κωδικοποιημένο σε μια IPv6 διεύθυνση.

2.4.3.2 Επίθεμα ονόματος

Η μορφή του επιθέματος ονόματος στην hICN φαίνεται παρακάτω:



Εικόνα 5: hICN επίθεμα ονόματος

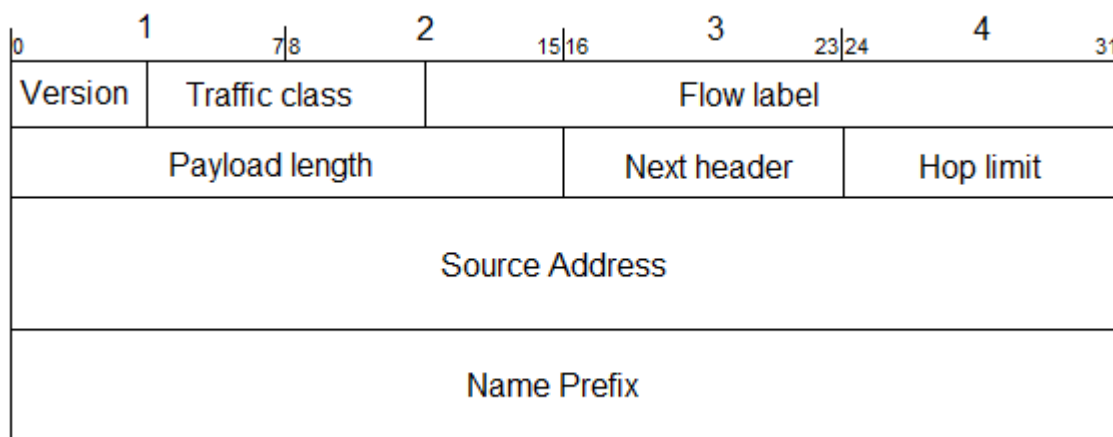
Το επίθεμα ονόματος χρησιμοποιείται από το επίπεδο μεταφοράς ώστε να κατατάξει σε μορφή πίνακα τα κομμάτια δεδομένων. Κάθε κομμάτι πρέπει να καταταχθεί στους τελικούς κόμβους και στο δίκτυο με το ίδιο επίθεμα. Αυτό συνεπάγεται ότι υπάρχει ένα κομμάτι προς μεταφορά σε κάθε IP πακέτο και ότι απαγορεύεται ο κατακερματισμός των IP πακέτων. Επίσης ο παραγωγός είναι αυτός που καθορίζει πως θα εκτελεστεί ο κατακερματισμός ανάλογα με την περίπτωση.

2.4.4 Μορφή πακέτων

Υπάρχουν δύο μορφές μονάδων δεδομένων: τα πακέτα ενδιαφέροντος (interest) και τα πακέτα δεδομένων (data) [11]. Αποτελούνται από μια κεφαλίδα μεταφοράς η οποία είναι ίδια και στα δύο είδη και από μια κεφαλίδα μεταφοράς η οποία διαφέρει ελαφρώς.

2.4.4.1 Πακέτο ενδιαφέροντος

Το πακέτο ενδιαφέροντος σε επίπεδο δικτύου στην hICN μοιάζει με ένα IPv6 πακέτο μόνο που το πεδίο της διεύθυνσης προορισμού αντικαθιστάται από το πρόθεμα ονόματος. Παρακάτω φαίνεται μια σχετική εικόνα μιας κεφαλίδας.



Εικόνα 6: IPv6 κεφαλίδα πακέτου ενδιαφέροντος (L3)

Είναι συνολικά μεγέθους 40 byte και αποτελείται από τα εξής πεδία:

Version: αριθμός πρωτοκόλλου δηλαδή ο αριθμός 6.

Traffic class: υποδεικνύει την κλάση ή την προτεραιότητα του πακέτου.

Flow label: υποδεικνύει στους ενδιαμέσους δρομολογητές ότι το πακέτο ανήκει σε μια συγκεκριμένη ακολουθία πακέτων μεταξύ πηγής-προορισμού.

Payload length: δηλώνει το μέγεθος του φορτίου.

Next header: δηλώνει είτε την πρώτη επέκταση κεφαλίδας (αν υπάρχει) είτε το πρωτόκολλο στο ανώτερο επίπεδο (π.χ TCP,UDP).

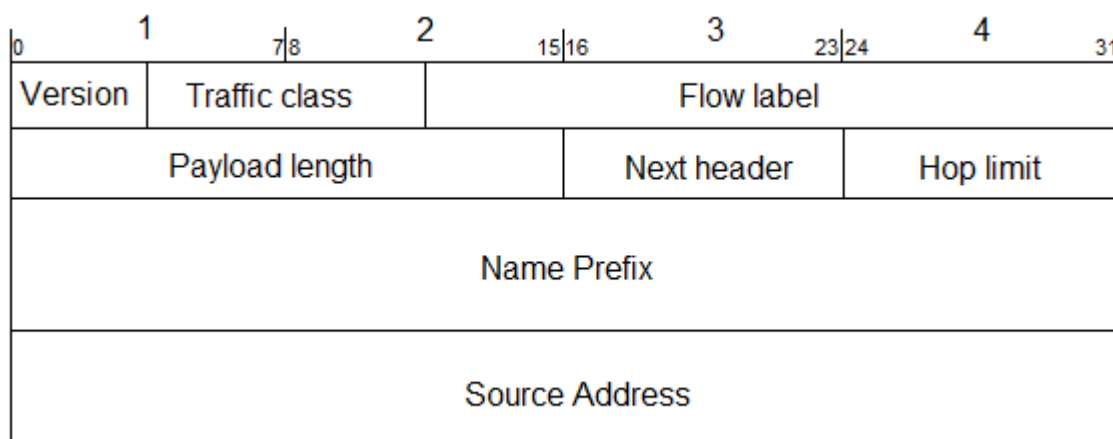
Hop limit: δηλώνει το μέγιστο αριθμό κόμβων στους οποίους μπορεί να ταξιδέψει το πακέτο έως ότου να απορριφθεί.

Source Address: είναι η μεγέθους 128-bit διεύθυνση της πηγής του πακέτου, μπορεί να είναι ανήκει σε κάποιο παραγωγό ή κάποιο hICN κόμβο.

Name Prefix: είναι το μεγέθους 128-bit πρόθεμα του ονόματος της υπηρεσίας που επιθυμούμε.

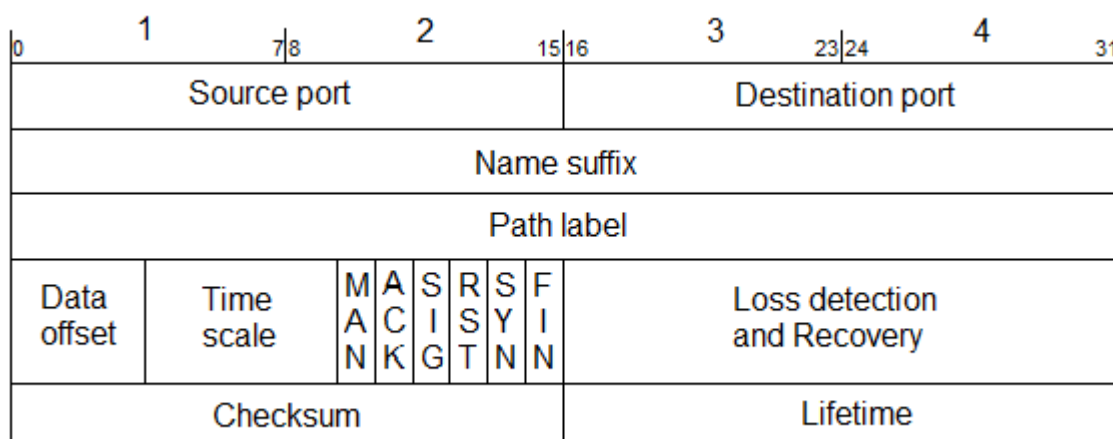
2.4.4.2 Πακέτο δεδομένων

Το πακέτο δεδομένων σε επίπεδο δικτύου στην IHCN έχει την ίδια ακριβώς μορφή με το πακέτο ενδιαφέροντος με την διαφορά ότι αντιστρέφονται τα πεδία Source Address και Name prefix όπως φαίνεται στην παρακάτω εικόνα.



Εικόνα 7: IPv6 κεφαλίδα πακέτου δεδομένων (L3)

Σε επίπεδο μεταφοράς η κεφαλίδα δεν διαφέρει στα δύο είδη μονάδων δεδομένων και έχει την μορφή που φαίνεται στην παρακάτω εικόνα.



Εικόνα 8: Κεφαλίδα πακέτων ενδιαφέροντος και δεδομένων (L4)

Source port: η θύρα που χρησιμοποιείται από την πηγή για την επικοινωνία.

Destination port: η θύρα που χρησιμοποιείται από τον προορισμό για την επικοινωνία.

Name Suffix: το μεγέθους 32-bit επίθεμα του ονόματος της υπηρεσίας που επιθυμούμε.

Path label: πρόκειται για μια ετικέτα των 32-bit που μεταφέρει μια κωδικοποίηση της διαδρομής μεταξύ του καταναλωτή και του αποστολέα δεδομένων (ο αποστολέας μπορεί να είναι ένας ενδιάμεσος κόμβος ή ο ίδιος ο παραγωγός).

Time scale: ένας φυσικός αριθμός μεταξύ 1-64 που αναπαριστάται με 6-bit και χρησιμοποιείται ως συντελεστής κλιμάκωσης για τον υπολογισμό της διάρκειας ζωής του πακέτου.

Loss detection and Recovery: ένας φυσικός αριθμός που αναπαριστάται με 16-bit και ελέγχει την αλληλουχία δεδομένων ανά περιοχή ώστε να εντοπίσει ζημίες κατά την ανάκτηση.

Lifetime: δηλώνει την διάρκεια ζωής του πακέτου σε χιλιοστά του δευτερολέπτου.

Checksum: χρησιμοποιείται για να ανιχνεύσει τυχόν αλλοιώσεις στις κεφαλίδες των πακέτων.

Επιπλέον περιέχονται και έξι δυαδικές σημαίες (flags) :

MAN: δηλώνει αν το πακέτο μεταφέρει κάποιο μανιφέστο μεταφοράς στο φορτίο του.

ACK: χρησιμοποιείται για να δηλώσει την επιτυχή παραλαβή ενός πακέτου.

SIG: δηλώνει αν το πακέτο φέρει κεφαλίδα ελέγχου ταυτότητας με υπογραφή. Το πακέτο ενδιαφέροντος δεν φέρει υπογραφές.

RST: δηλώνει στον αποστολέα ότι ένα πακέτο που ελήφθει είναι ανεπιθύμητο.

SYN: είναι σημαία συγχρονισμού και χρησιμοποιείται σαν πρώτο βήμα εγκαθίδρυσης επικοινωνίας μεταξύ δύο χρηστών.

FIN: είναι σημαία τερματισμού και χρησιμοποιείται όταν ο αποστολέας δεν έχει άλλα δεδομένα προς αποστολή, δηλαδή στο τελευταίο πακέτο της επικοινωνίας.

2.4.5 Κρυφή μνήμη

Η κρυφή μνήμη πακέτων [11] είναι μία τοπική μνήμη στους δρομολογητές όπου αποθηκεύονται προσωρινά αιτήματα και απαντήσεις. Η απλούστερη εκδοχή της κρυφής μνήμης πρέπει να δημιουργεί έναν κατάλογο των πακέτων με βάση το όνομά τους, για παράδειγμα μια σύνδεση του προθέματος με το επίθεμα τους. Η διαδικασία εισαγωγής και διαγραφής των πακέτων στην κρυφή μνήμη περιγράφεται σε επόμενη ενότητα.

2.4.6 Προώθηση

Στην αρχιτεκτονική hICN έχουμε δύο είδη διαδρομών προώθησης, την διαδρομή των πακέτων τύπου Interest («αιτήματα/ενδιαφέροντα») και την διαδρομή των πακέτων τύπου Data («δεδομένα»). Τα αιτήματα και οι απαντήσεις μεταχειρίζονται από τους hICN κόμβους με διαφορετικό τρόπο. Αμφότερες οι διαδρομές προώθησης απαιτούν την ύπαρξη κρυφής μνήμης ενσωματωμένης στον δρομολογητή, ώστε να αποθηκεύονται προσωρινά τα διάφορα αιτήματα και απαντήσεις. Όταν ένα αίτημα αποθηκεύεται στην κρυφή μνήμη ενός hICN κόμβου, η απάντηση σε αυτό μπορεί να μεταδοθεί στους σωστούς κόμβους αφού στο πεδίο διεύθυνσης πηγής του πακέτου ενδιαφέροντος περιέχονται οι κατάλληλες πληροφορίες για τον κόμβο που μετέδωσε το αίτημα. Αυτό συνεπάγεται ότι η διαδρομή προώθησης ενδιαφέροντος βασίζεται σε αναζητήσεις σε έναν IP πίνακα πληροφοριών προώθησης (IP FIB), παρόμοια με τα IP πακέτα, με μια επιπρόσθετη αναζήτηση για να ελεγχθεί εάν η αναμενόμενη απάντηση υπάρχει ήδη στην τοπική κρυφή μνήμη για ταχύτερη απάντηση.

Η διαδρομή προώθησης δεδομένων μοιάζει με τον μηχανισμό εναλλαγής ετικετών (Label swapping) [12]. Εν συντομία, ο τρόπος με τον οποίο λειτουργεί η εναλλαγή ετικετών για την διαδικασία της προώθησης είναι ότι ο κόμβος εξετάζει την ετικέτα στο πάνω μέρος της στοίβας ετικετών. Έπειτα χρησιμοποιεί έναν χάρτη ετικετών ώστε να εκχωρήσει την ετικέτα σε έναν πίνακα με τους επόμενους κόμβους μετάβασης των ετικετών (next hop). Τέλος, χρησιμοποιώντας τον πίνακα αυτόν προωθεί τα πακέτα και εκτελεί μια ενέργεια στην στοίβα ετικετών (π.χ αλλαγή ετικέτας κορυφής, αφαίρεση ετικέτας κορυφής).

Στην αρχιτεκτονική hICN το ρόλο της ετικέτας παίζει το αναγνωριστικό του ονόματος του πακέτου, δηλαδή το πρόθεμα και επίθεμα του ονόματος. Ο επόμενος κόμβος μεταβίβασης της απάντησης πράγματι επιλέγεται χρησιμοποιώντας πληροφορίες από ένα αίτημα που ταιριάζει και βρίσκεται αποθηκευμένο στην κρυφή μνήμη.

Το πρόθεμα του ονόματος στην κεφαλίδα του πακέτου, τόσο στα αιτήματα όσο και στις απαντήσεις, δεν τροποποιείται ποτέ κατά μήκος της διαδρομής. Αντίθετα, το αναγνωριστικό διεπαφής (locator) που αναγράφεται στο πεδίο της διεύθυνσης πηγής ή προορισμού, είτε πρόκειται για πακέτα ενδιαφέροντος είτε για πακέτα δεδομένων, τροποποιείται στην έξοδο του δρομολογητή.

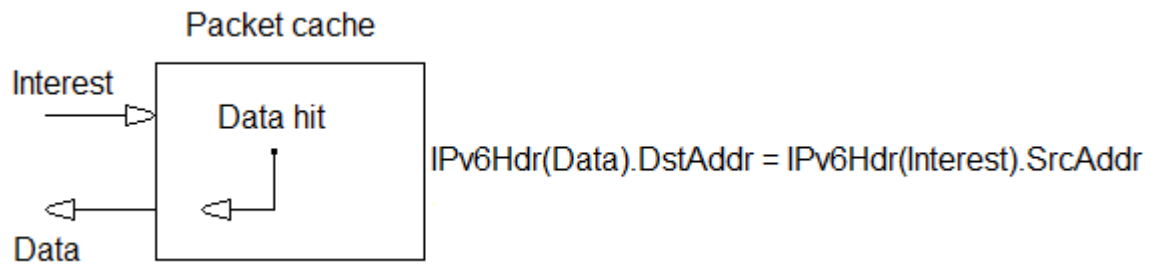
2.4.6.1 Διαδρομή προώθησης πακέτων τύπου «Interest»

Στην είσοδο του δρομολογητή αναλύεται το εισερχόμενο πακέτο ενδιαφέροντος (Interest packet) ώστε να ληφθούν το πρόθεμα και το επίθεμα του ονόματος. Έπειτα, πραγματοποιείται μια ακριβής αναζήτηση στην κρυφή μνήμη πακέτων χρησιμοποιώντας το πλήρες όνομα του πακέτου σαν κλειδί αναζήτησης.

Ας υποθέσουμε ότι έχουμε το παραπάνω σενάριο και το πακέτο ονομάζεται 'Interest'. Ανάλογα με τα αποτελέσματα της αναζήτησης υπάρχουν οι ακόλουθες επιλογές:

i. Υπάρχει τουλάχιστον ένα ταίριασμα.

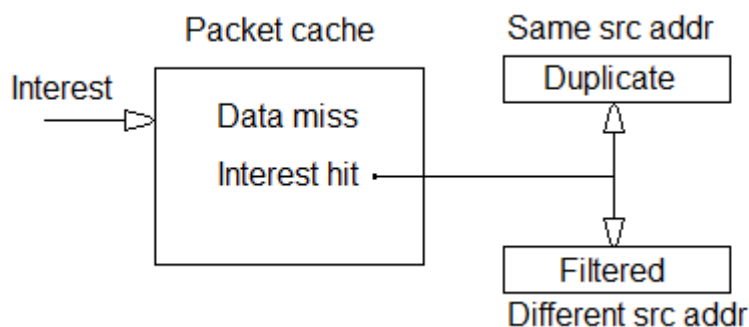
➤ Εάν το ταίριασμα είναι με το πακέτο δεδομένων 'Data', τα υπόλοιπα ταιριάσματα αγνοούνται και το 'Data' ετοιμάζεται για μετάδοση τοποθετώντας στην διεύθυνση προορισμού του 'Data' την διεύθυνση πηγής του 'Interest'. Έπειτα το πακέτο 'Data' μεταφέρεται στην έξοδο του κόμβου για περαιτέρω επεξεργασία πριν την μετάδοση. Για παράδειγμα, ο επόμενος κόμβος μετάβασης μπορεί να επιλεγεί με βάση τον IPv6 πίνακα πληροφοριών προώθησης του δρομολογητή (μεγαλύτερη αντιστοιχία προθέματος). Η αναζήτηση πιθανόν να αποφευχθεί σε περίπτωση που ο επόμενος κόμβος μετάβασης μπορεί να ληφθεί, με επεξεργασία του 'Interest', από πληροφορίες που αντλήθηκαν προηγουμένως. Τέλος, το πακέτο 'Interest' απορρίπτεται.



Εικόνα 9: Πακέτο ενδιαφέροντος βρίσκει ταιριαστό πακέτο δεδομένων στην κρυφή μνήμη

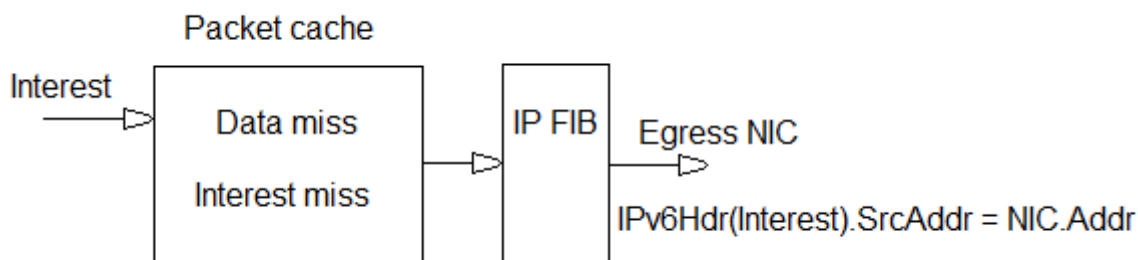
➤ Εάν υπάρχουν ένα ή περισσότερα ταιριάσματα και όλα είναι πακέτα ενδιαφέροντος.

- εάν ένα πακέτο ενδιαφέροντος ταιριάζει και έχει την ίδια διεύθυνση πηγής με το πακέτο 'Interest', τότε το 'Interest' κατατάσσεται ως διπλότυπο αίτημα και επεξεργάζεται περαιτέρω ως διπλότυπο.
- εάν ένα ή πολλά πακέτα ενδιαφέροντος ταιριάζουν αλλά έχουν διαφορετικές διευθύνσεις πηγής, τότε το πακέτο 'Interest' ταξινομείται ως φιλτραρισμένο και αποθηκεύεται στην κρυφή μνήμη.



Εικόνα 10: Πακέτο ενδιαφέροντος βρίσκει ταιριαστό πακέτο ενδιαφέροντος στην κρυφή μνήμη

ii. Δεν υπάρχει κάποιο ταίριασμα και το πακέτο 'Interest' περνά στην έξοδο του δρομολογητή για περαιτέρω ανάλυση ώστε να βρεθεί ο επόμενος κόμβος μετάβασης με βάση τον IP πίνακα πληροφοριών προώθησης του δρομολογητή.



Εικόνα 11: Πακέτο ενδιαφέροντος δεν βρίσκει ταιριαστό πακέτο στην κρυφή μνήμη και προωθείται

Το πεδίο διεύθυνσης προορισμού σε ένα πακέτο ενδιαφέροντος είναι πολυμορφικό αφού έχει δύο τύπους ανάλογα με την αναζήτηση που εκτελείται. Έχει τον τύπο ενός ονόματος ανεξάρτητου της τοποθεσίας όταν χρησιμοποιείται για να βρεθεί ένα ταίριασμα στην κρυφή μνήμη πακέτων, ενώ έχει τον τύπο ενός προθέματος διεύθυνσης όταν χρησιμοποιείται για να βρεθεί ο επόμενος κόμβος μετάβασης στον IP πίνακα πληροφοριών προώθησης. Ο πολυμορφισμός είναι φάνερος στην επιφάνεια προώθησης αλλά έχει επιπτώσεις στην επιφάνεια ελέγχου.

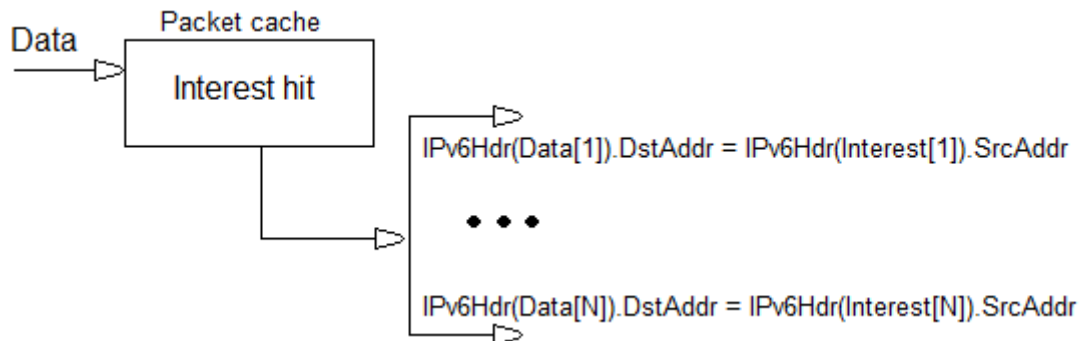
2.4.6.2 Διαδρομή προώθησης πακέτων τύπου «Data»

Στην είσοδο του δρομολογητή αναλύεται το εισερχόμενο πακέτο δεδομένων ώστε να ληφθούν το πρόθεμα και το επίθεμα του ονόματος. Έπειτα, πραγματοποιείται μια ακριβής αναζήτηση στην κρυφή μνήμη πακέτων χρησιμοποιώντας το πλήρες όνομα του πακέτου σαν κλειδί αναζήτησης.

Ας υποθέσουμε ότι έχουμε το παραπάνω σενάριο και το πακέτο ονομάζεται 'Data'. Ανάλογα με τα αποτελέσματα της αναζήτησης υπάρχουν οι ακόλουθες επιλογές:

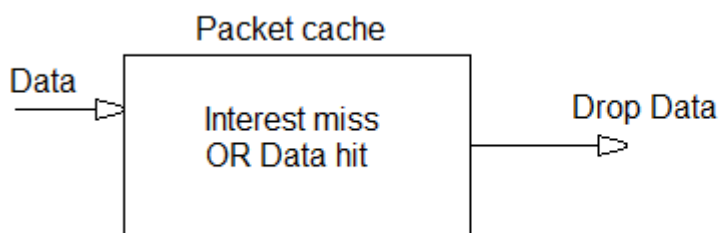
ι. Υπάρχουν ένα ή περισσότερα ταιριάσματα πακέτων ενδιαφέροντος. Τότε το πακέτο δεδομένων 'Data' κλωνοποιείται ώστε να έχει τόσα αντίγραφα όσα και ο αριθμός των πακέτων ενδιαφέροντος που ταιριάζουν. Το πεδίο διεύθυνσης προορισμού κάθε αντίγραφου του πακέτου 'Data' συμπληρώνεται με το πεδίο διεύθυνσης πηγής από κάθε πακέτο ενδιαφέροντος. Έπειτα, όλα τα αντίγραφα

μεταφέρονται στην έξοδο του κόμβου για περαιτέρω επεξεργασία πριν την μετάδοση ώστε να βρεθεί ο επόμενος κόμβος μετάβασης.



Εικόνα 12: Πακέτο δεδομένων βρίσκει πακέτο ενδιαφέροντος στην κρυφή μνήμη

ii. Δεν υπάρχουν πακέτα ενδιαφέροντος που να ταιριάζουν και το πακέτο 'Data' απορρίπτεται.



Εικόνα 13 : Πακέτο δεδομένων δεν βρίσκει πακέτο ενδιαφέροντος ή βρίσκει ταιριαστό πακέτο δεδομένων και απορρίπτεται

2.4.7 Ασφάλεια

Η αρχιτεκτονική hICN φυσικά χρησιμοποιεί το μοντέλο ασφάλειας της ICN. Αυτό συνεπάγεται ότι η ακεραιότητα, η αυθεντικότητα της προέλευσης των δεδομένων και η εμπιστευτικότητα συνδέονται με το περιεχόμενο και όχι με το κανάλι επικοινωνίας.

Η ακεραιότητα και η αυθεντικότητα της προέλευσης των δεδομένων παρέχονται με την χρήση ψηφιακής υπογραφής η οποία υπολογίζεται από τον

παραγωγό των δεδομένων και συμπεριλαμβάνεται σε κάθε πακέτο δεδομένων. Υπάρχουν δύο προσεγγίσεις που εξασφαλίζουν αυτές τις δύο βασικές αρχές ασφαλείας. Η πρώτη βασίζεται στην επικύρωση των κεφαλίδων IP (Authenticated Header), ενός μηχανισμού γνωστού στο IPSec πρωτόκολλο, και η δεύτερη στα μανιφέστα μεταφοράς (Transport manifests).

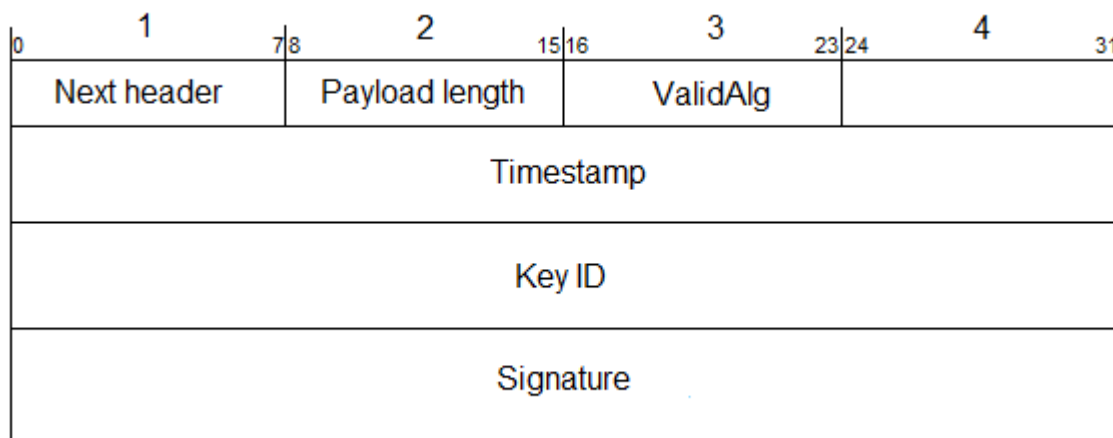
2.4.7.1 Επικύρωση κεφαλίδων

Ο μηχανισμός της επικύρωσης κεφαλίδων IP [11] είναι ένα μέρος του πρωτοκόλλου IPSec, ο οποίος έχει σαν στόχο να πιστοποιεί την προέλευση των IP πακέτων και να εγγυάται την ακεραιότητα των δεδομένων, διασφαλίζοντας ότι το περιεχόμενό τους (κεφαλίδα και πληροφορία) δεν έχουν αλλάξει κατά την μετάδοση.

Η επικυρωμένη κεφαλίδα IP δεν χρησιμοποιείται ως κεφαλίδα επέκτασης του IPv6 αλλά προσαρτάται μετά την κεφαλίδα μεταφοράς. Ωστόσο, στην περίπτωση της hICN αρχιτεκτονικής η επιλογή αυτής της προσέγγισης έχει γίνει ώστε να αξιοποιηθούν ήδη υπάρχουσες υλοποιήσεις πρωτοκόλλων στα τελικά σημεία.

Όταν χρησιμοποιείται η προσέγγιση της επικυρωμένης κεφαλίδας, η ψηφιακή υπογραφή υπολογίζεται πάνω:

- σε πεδία IP ή πεδία επέκτασης κεφαλίδας είτε αυτά είναι αμετάβλητα κατά την μεταφορά είτε προβλέψιμα σε αξία κατά την άφιξη στον καταναλωτή.
- στην επικυρωμένη κεφαλίδα με το πεδίο της υπογραφής να έχει την τιμή μηδέν. Υπενθυμίζεται ότι, στην hICN το πεδίο προορισμού στην κεφαλίδα δεν είναι αμετάβλητο ούτε και προβλέψιμο και πρέπει να του τεθεί η τιμή μηδέν για τον υπολογισμό της υπογραφής. Επιπλέον, η επικυρωμένη κεφαλίδα τοποθετείται μετά την TCP κεφαλίδα προκειμένου να αποφευχθεί κάθε είδος φιλτραρίσματος από συσκευές δικτύου.



Εικόνα 14: Επικεφαλίδα ελέγχου ταυτότητας IP για τη μεταφορά των υπογραφών των πακέτων

Next Header: προσδιορίζει τον τύπο του περιεχομένου μετά την επικυρωμένη επικεφαλίδα. Η τιμή του πεδίου επιλέγεται μεταξύ συγκριμένων αριθμών καθορισμένων από το IANA. Παραδείγματος χάριν, η τιμή 4 δηλώνει IPv4 τύπο, η τιμή 41 δηλώνει IPv6 ενώ η τιμή 6 δηλώνει TCP τύπο.

Payload Length: δηλώνει το μήκος της επικυρωμένης κεφαλίδας σε λέξεις των 32-bit.

ValidAlg: πρόκειται για έναν 8-bit δείκτη που δηλώνει ποιός αλγόριθμος επαλήθευσης πρέπει να χρησιμοποιηθεί για επιβεβαίωση της υπογραφής.

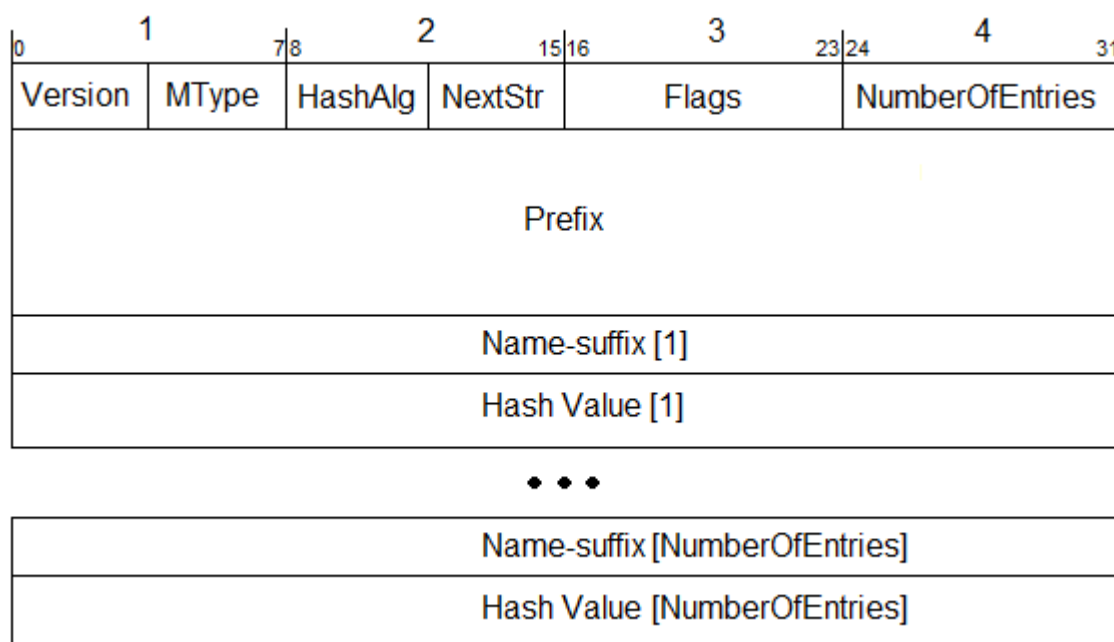
Timestamp: πρόκειται για χρονοσήμανση των 64-bit που υποδεικνύει την εγκυρότητα της υπογραφής.

Key ID: είναι το πεδίο με το αναγνωριστικό κλειδιού σε 256-bit.

Signature: πρόκειται για ένα πεδίο μεταβλητού μήκους που φέρει την κρυπτογραφική υπογραφή του φακέλου ασφαλείας. Ενδεικτικά, για το κρυπτοσύστημα RSA-1024 είναι 128 bytes, για το RSA-2048 είναι 256 bytes, για το EDCSA 192 είναι 56 bytes και για το EDCSA 256 είναι 72 bytes.

2.4.7.2 Μανιφέστο μεταφοράς

Το μανιφέστο μεταφοράς είναι μια οντότητα του επιπέδου μεταφοράς (OSI layer-4) που υπολογίζεται στον παραγωγό, όπου υπάρχει η λίστα των ονομάτων μιας ομάδας πακέτων προς μεταφορά στον καταναλωτή. Αντί για υπογραφές υπολογίζονται hICN κρυπτογραφικά hashes για τα πακέτα δεδομένων πάνω σε αμετάβλητα πεδία, όπως περιγράφηκε και στην προηγούμενη ενότητα. Ακόμα, το μανιφέστο πρέπει να υπογραφεί ώστε να εγγυάται ένα επίπεδο ασφάλειας παρόμοιο με αυτό των υπογραφών πακέτων.



Εικόνα 15: Το μανιφέστο μεταφοράς που δημιουργείται από τον παραγωγό για τον καταναλωτή

Κεφάλαιο 3: Εργαλεία

3.1 Εισαγωγή

Στο κεφάλαιο που ακολουθεί θα παρουσιαστούν τα εργαλεία που επιστρατεύτηκαν για την περάτωση αυτής της διπλωματικής εργασίας. Αρχικά, θα εστιάσουμε στην πλατφόρμα διαχείρισης vICN (virtualized ICN) που αποτέλεσε και την βάση για την μελέτη και τα πειράματα που διεξάχθηκαν. Έπειτα θα περιγράψουμε τον προσομοιωτή ns-3 και ειδικότερα κάποια κομμάτια αυτού που χρησιμοποιούνται από το εργαλείο vICN προς επίτευξη της προσομοίωσης συγκεκριμένων λειτουργιών.

3.2 vICN (virtualized Information-Centric Networking)

Σε αυτή την ενότητα θα αναλύσουμε το εργαλείο vICN ξεκινώντας από την παρουσίασή του και συνεχίζοντας με την αρχιτεκτονική του, την υλοποίηση των διαφόρων στοιχείων του και τις δυνατότητες που προσφέρει στον χρήστη του.

3.2.1 Εισαγωγή

Καθώς τα αποτελέσματα από έρευνες που έχουν γίνει σχετικά με το αν η αρχιτεκτονική ICN μπορεί να πρωτοστατήσει στο διαδίκτυο του μέλλοντος είναι θετικές, η ICN κοινότητα αναγνώρισε την σημασία της περαιτέρω ρεαλιστικής πειραματικής αλλά και εφαρμοσμένης έρευνας. Έκτοτε έχουν δημιουργηθεί πολλά εργαλεία αλλά και πλατφόρμες για να βοηθήσουν στην έρευνα μέσω προσομοιώσεων και εξομοιώσεων των διαφόρων πειραμάτων, τα κυριότερα των οποίων είναι τα CCNx, NDN, CCNlite, MiniCCNx, MiniNDN. Τα εργαλεία αυτά λειτουργούν για περιβάλλοντα αποκλειστικά ICN δικτύων, αντισταθμίζοντας τα γενικά χαρακτηριστικά του δικτύου για εναλλαγή κλίμακας και προσφέροντας περιορισμένη ευελιξία στην τροποποίηση βασικών λειτουργιών της ICN, τοπολογιών, ρυθμίσεων ή API εφαρμογών.

Στην παρούσα διπλωματική χρησιμοποιήθηκε ένα διαφορετικό εργαλείο, το vICN (virtualized ICN) [13]. Πρόκειται για μια ευέλικτη ενοποιημένη πλατφόρμα ανοιχτού κώδικα που βοηθά στην ανάπτυξη, την ενορχήστρωση, τη διαχείριση ICN δικτύων και επιτρέπει την διεξαγωγή δοκιμών σε περιπτώσεις χρήσης όπως, πειράματα μεγάλης κλίμακας και λεπτομερούς ελέγχου σε γενικά περιβάλλον δοκιμών (testbed) και δημιουργία αξιόπιστου ICN δικτύου με πραγματικές εφαρμογές και απόδειξη της πρακτικής δυναμικής.

Οι λειτουργίες που απαιτούνται για την ανάπτυξη ενός δικτύου ICN περιλαμβάνουν την εγκατάσταση, την παραμετροποίηση και την παρακολούθηση μιας νέας συλλογής πρωτοκόλλων δικτύου, συχνά αποκαλούμενης στοίβα δικτύου (network stack), στους κόμβους προώθησης ή στους υποδοχείς API που χρησιμοποιούνται απο εφαρμογές στους τελικούς κόμβους. Εφόσον η φόρτωση μιας στοίβας δικτύου σε υλικό γενικού σκοπού είναι εύκολα πραγματοποιήσιμη, τότε οι απαιτήσεις που προκύπτουν είναι αυτές της εξαιρετικής αξιοπιστίας, της υψηλής ταχύτητας και της προβλεψιμότητας και άλλες. Η vICN έχει τους ίδιους υψηλού επιπέδου στόχους με τις αρχιτεκτονικές της Δικτύωσης Καθορισμένης από Λογισμικό (SDN) και της Εικονοποίησης Δικτυακών Λειτουργιών (NFV), αλλά με επιπλέον δυνατότητες ICN που συνήθως δεν απαιτούνται από IP υπηρεσίες.

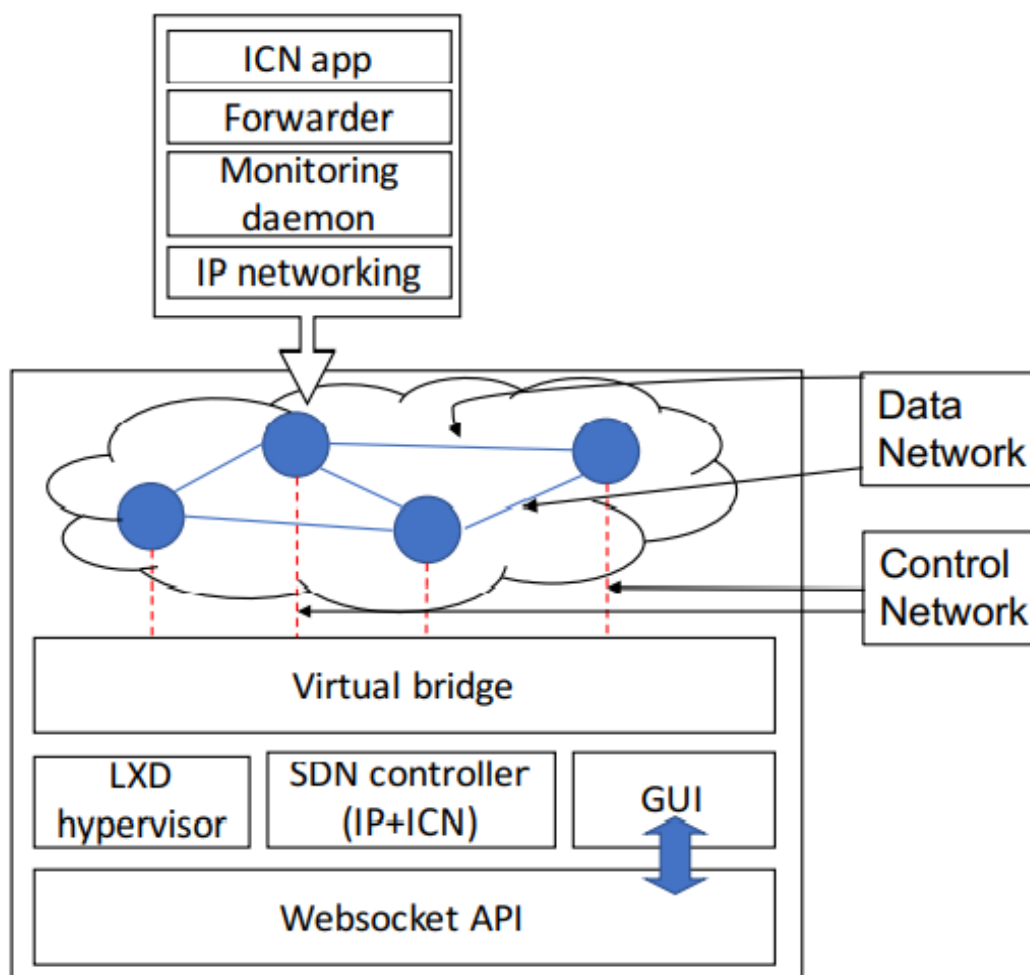
Συνολικά, εντοπίζουμε τρεις κύριες προκλήσεις που αντιμετωπίζει το vICN και το διαφοροποιούν σε σχέση με την τελευταία λέξη της τεχνολογίας:

- Προγραμματισιμότητα: η ανάγκη της παραγωγής ενός απλού και ενοποιημένου API, αρκετά εύληπτου ώστε να διευκολύνει την εκκίνηση, αρκετά εκφραστικού ώστε να ικανοποιεί τόσο τη διαμόρφωση των πόρων όσο και την παρακολούθηση και αρκετά ευέλικτο ώστε να επιτρέπει στον χρήστη να αποφασίσει σχετικά με το βαθμό ελέγχου που εγκρίνεται.
- Επεκτασιμότητα: η vICN στοχεύει να συνδυάσει την γρήγορη επεξεργασία πακέτων, τον τεμαχισμό και την εικονοποίηση του δικτύου και τον παράλληλο και ελάχιστης καθυστέρησης προγραμματισμό των εργασιών.

- Αξιοπιστία: μια βασική ιδιότητα της vICN είναι η ικανότητα της να διατηρεί την κατάσταση της εγκατάστασης, να αναρρώνει μετά από αποτυχίες και να αντιμετωπίζει αυτόματα διάφορα προβλήματα. Αυτά απαιτούν συνολικά το λογισμικό να είναι σε θέση να προσαρμόζει την προγραμματιζόμενη παρακολούθηση των λειτουργιών και τον εντοπισμό σφαλμάτων.

3.2.2 Περιγραφή ανάπτυξης

Μια τυπική vICN ανάπτυξη [13] μοιάζει με την εικόνα παρακάτω:



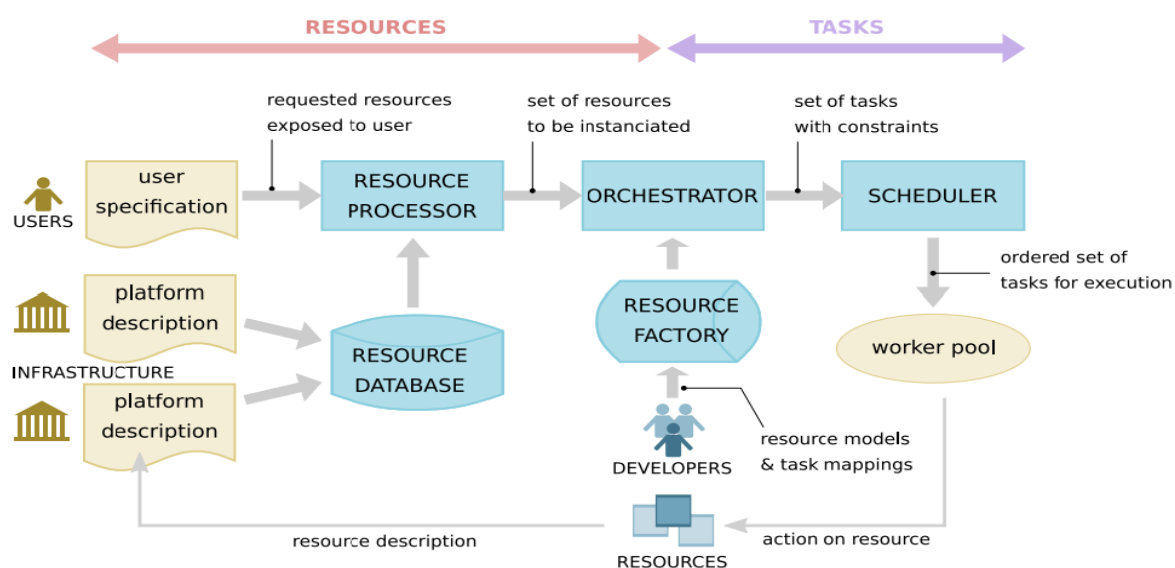
Εικόνα 16: Μονάδες που αναπτύσσονται από τη vICN

Αποτελείται από μονάδες του λογισμικού Linux, τα Linux Containers, συνδεδεμένες μεταξύ τους μέσω εικονικών διεπαφών Ethernet (veth). Πάνω από την τοπολογία, δηλαδή εικονικούς κόμβους και συνδέσεις, η vICN εφαρμόζει το δικό της ελεγκτή Δικτύωσης Καθορισμένης από Λογισμικό (SDN controller), προσαρμοσμένο για την δημιουργία ICN και IP διευθυνσιοδότησης και δρομολόγησης. Επιπλέον δημιουργεί ένα δίκτυο ελέγχου βασισμένο στο IP με την χρήση μίας εικονικής γέφυρας (virtual bridge), παρέχοντας έτσι στα containers δυνατότητα σύνδεσης στο διαδίκτυο ώστε να μπορούν να εγκαθιστούν τυχόν εφαρμογές που λείπουν και να δίνουν πληροφορίες από την παρακολούθηση του δικτύου. Στα containers οι σχετιζόμενες ICN εφαρμογές ξεκινούν και συνδέονται με τον ICN μεταφορέα (ICN forwarder).

Η vICN επιτρέπει στον χρήστη να καθορίσει την τοπολογία δηλώνοντας τα επιμέρους στοιχεία της. Ακόμα, χρησιμοποιώντας αντιστοίχιση σχέσεων των αντικειμένων μεταξύ των διαφόρων πόρων δικτύων και μηχανές πεπερασμένων καταστάσεων για την παρακολούθηση αυτών, είναι σε θέση να αναπτύξει ένα πλήρες ICN δίκτυο. Η ανάπτυξη της επιθυμητής τοπολογίας δηλώνεται μέσω ενός JSON αρχείου, όπως θα δούμε και σε παράδειγμα στο επόμενο κεφάλαιο, όπου ο χρήστης μπορεί να περιγράψει το δίκτυο που επιθυμεί όπως και τις θέσεις των παραγωγών και καταναλωτών.

3.2.3 Δομή της vICN

Η παρακάτω εικόνα δείχνει την αρχιτεκτονική σε υψηλό επίπεδο της vICN.



Εικόνα 17: Αρχιτεκτονική σε υψηλό επίπεδο της vICN

3.2.3.1 Πρακτική αρχιτεκτονική

Ο πόρος (Resource) είναι η βασική μονάδα πληροφορίας στην vICN [14]. Αυτό αποτελείται από ένα αφηρημένο μοντέλο το οποίο χρησιμοποιείται από ένα σύνολο χαρτογράφων για να μεταφραστούν ενέργειες στον πόρο σε μια σειρά εργασιών που πρέπει να εκτελεστούν. Οι πόροι αποθηκεύονται στο εργοστάσιο πόρων (Resource factory) και ποικίλλουν (μια εικονική μηχανή, μια εφαρμογή, μια IP διαδρομή). Επιπλέον μπορούν να συνδυαστούν ή να επεκταθούν ώστε να σχηματίσουν νέους πόρους. Η αρχιτεκτονική vICN διαφοροποιεί το ρόλο των χρηστών, των προγραμματιστών και των παρόχων υποδομής. Οι προγραμματιστές ενσωματώνουν εργαλεία δημιουργώντας νέους πόρους ή επεκτείνοντας παλαιότερους, ενώ τόσο οι χρήστες όσο και οι πάροχοι υποδομής χρησιμοποιούν το σύνολο των πόρων για να περιγράψουν αντίστοιχα τι ζητούν (χρήστες) ή τι μπορούν να διαθέσουν (παροχείς υποδομής).

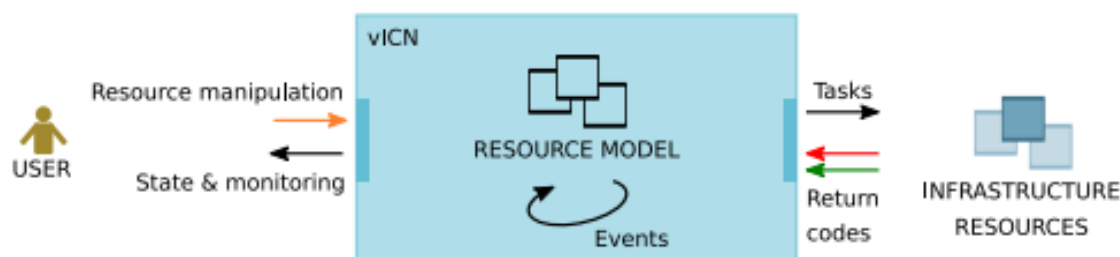
Οι πόροι προσδιορίζονται με διαφορετικούς βαθμούς λεπτομέρειας, για παράδειγμα η υποδομή περιγράφεται λεπτομερώς ώστε να σχηματιστεί μια βάση δεδομένων πόρων (Resource database) που χρησιμεύει ως βάση για την ανάπτυξη, ενώ οι προδιαγραφές του χρήστη μπορεί να είναι γενικές ή αφηρημένες και να αναφέρουν μόνο τους πόρους που ενδιαφέρουν τον χρήστη. Ο ρόλος του επεξεργαστή πόρων (Resource processor) είναι να μετατρέπει μια αφηρημένη ή ανεκπλήρωτη περιγραφή σε ένα σύνολο πόρων που αντιστοιχίζονται στην υποδομή και οδηγούν σε συνεπή ανάπτυξη.

Από τη στιγμή που επιλεγθούν οι κατάλληλοι πόροι, ο ενορχηστρωτής (orchestrator) τους μεταφράζει σε ένα σύνολο ενεργειών που πρέπει να εκτελεστούν, με βάση την τρέχουσα κατάσταση της ανάπτυξης και τους περιορισμούς λόγω συγχρονισμού ή διαδοχής των εργασιών. Οι ενέργειες που προκύπτουν, επεξεργάζονται από τον χρονοπρογραμματιστή (scheduler), ο οποίος εξάγει ένα σχέδιο εκτέλεσης και αποστέλλει παράλληλες εργασίες σε μια ομάδα εργατών (worker pool) με στόχο την ελαχιστοποίηση τους χρόνου ανάπτυξης.

Έτσι προκύπτει το γενικό συμπέρασμα ότι η vICN αποτελείται από δύο έννοιες, τους πόρους (Resources) που αποτελούν εξωτερικές μονάδες πληροφοριών και που εκτίθενται στους χρήστες, τους προγραμματιστές και τους παροχείς υποδομής και τις εργασίες (Tasks) που αποτελούν εσωτερικές μονάδες πληροφοριών, ορισμένες από τους προγραμματιστές, που μεταφράζουν τα αιτήματα πόρων σε αλλαγές της κατάστασης ανάπτυξης του δικτύου.

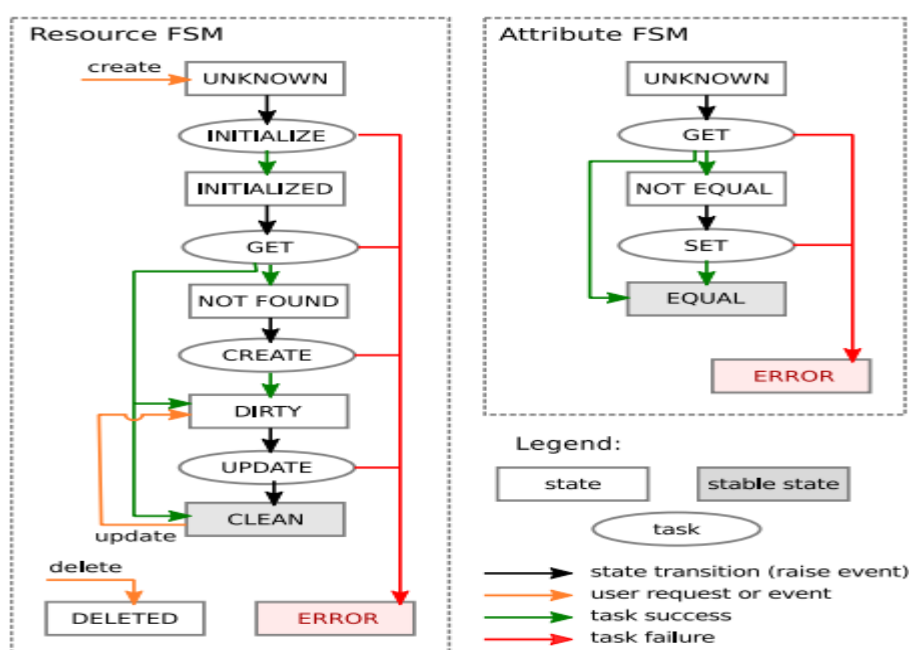
3.2.3.2 Μοντέλο πόρων

Στην πλατφόρμα vICN το εσωτερικό μοντέλο πόρων στο οποίο προαναφερθήκαμε εκτίθεται στους χρήστες και στους προγραμματιστές μέσω μιας γλώσσας ερωτήσεων (query language), στο πνεύμα της SQL γλώσσας, οι οποίες βασίζονται και επεκτείνουν ένα σχεσιακό μοντέλο αντικειμένων. Το μοντέλο [14] ορίζει ένα βασικό αντικείμενο ως ένα σύνολο τυποποιημένων χαρακτηριστικών και μεθόδων, όπου οι τύποι αναφέρονται σε ακεραίους, αλφαριθμητικά κ.α ή σε νέα καθορισμένα αντικείμενα. Η γλώσσα ερωτήσεων που έχει κτιστεί πάνω στο μοντέλο χρησιμοποιείται για τη δημιουργία, την καταστροφή και το χειρισμό αυτών των αντικειμένων, είτε για τη ρύθμιση των πόρων είτε για την ανάκτηση των πληροφοριών παρακολούθησης. Αυτό το μοντέλο επωφελείται από τη δύναμη και την εκφραστικότητα του σχεσιακού μοντέλου και μερικές βασικές έννοιες του αντικειμενοστραφούς προγραμματισμού, συγκεκριμένα της σύνθεσης (composition) και της κληρονομικότητας (inheritance). Επιπρόσθετα, χρησιμεύει σαν μια ενσωματωμένη διεπαφή βασισμένη σε έννοιες αναγνώσιμες τόσο από τον άνθρωπο όσο και από τη μηχανή.



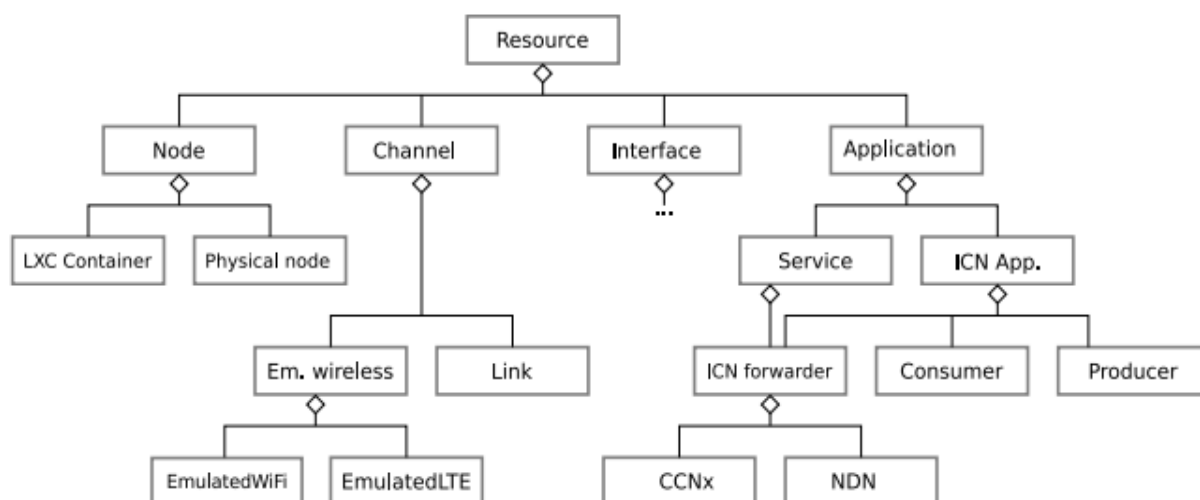
Εικόνα 18: Ροή πληροφοριών στη vICN

- **Πόροι:** Οι πόροι στην vICN αποτελούν λογικές αναπαραστάσεις φυσικών ή και απομακρυσμένων στοιχείων, η κατάσταση των οποίων πρέπει να διατηρείται συγχρονισμένη. Η κατάσταση ενός πόρου μπορεί να επηρεαστεί από ερωτήματα χρηστών (queries), από γεγονότα που περιλαμβάνουν τον πόρο ή από την παρακολούθηση ερωτημάτων που εκδίδονται στον πόρο. Για παράδειγμα, ένα στοιχείο δρομολόγησης μπορεί να αναπροσαρμόσει τις διαδρομές όταν ειδοποιηθεί για μια αλλαγή στο σύνολο των κόμβων, των διεπαφών ή των συνδέσεων. Η ροή της πληροφορίας φαίνεται στην παραπάνω εικόνα.
- **Κατάσταση πόρων:** Η κατάσταση ενός πόρου παρακολουθείται, για αξιοπιστία και σταθερότητα, από μια μηχανή πεπερασμένων καταστάσεων (FSM) η οποία μοντελοποιεί τις πιθανές καταστάσεις (τετράγωνα) του και τις εργασίες ή διαδικασίες που εκκρεμούν (κύκλοι). Οι μεταβάσεις υπαγορεύονται από ενέργειες του χρήστη ή εσωτερικά συμβάντα (βελάκια). Το INITIALIZE καλείται όταν ο πόρος και το αντικείμενο του δημιουργούνται για εσωτερική ρύθμιση. Το CREATE και DELETE μπορούν να δημιουργήσουν ή να καταστρέψουν αντίστοιχα τον απομακρυσμένο πόρο και να ορίσουν κάποια χαρακτηριστικά. Το GET ανακτά την τρέχουσα κατάσταση ενός πόρου, καθώς και την κατάσταση ορισμένων χαρακτηριστικών του. Η UPDATE ανανεώνει τα χαρακτηριστικά και στην πραγματικότητα εκτελεί παράλληλες κλήσεις της attribute-FSM που φαίνεται δεξιά στην παρακάτω εικόνα.



Εικόνα 19: Μηχανή πεπερασμένων καταστάσεων της vICN

➤ Χαρτογράφος πόρων: Για κάθε μετάβαση μεταξύ καταστάσεων, ο προγραμματιστής μπορεί να συσχετίσει τις εντολές που πρέπει να εκτελεστούν χάρη στους χαρτογράφους πόρων. Αυτές οι εντολές διαχειρίζονται από το νICN μέσω των αφηρημένων εργασιών. Είναι εξειδικευμένοι ώστε να διαχειρίζονται πολλαπλές διεπαφές στο στρώμα χαμηλότερου επιπέδου, όπως για παράδειγμα NETCONF / YANG, SSH / Bash ή LXD REST κλήσεις. Επιπλέον, νέες εργασίες μπορούν να δημιουργηθούν με χρήση σύνθεσης (composition) ή κληρονομικότητας (inheritance), χρησιμοποιώντας αλγεβρικούς τελεστές για να ενημερώνουν για την παράλληλη ή γραμμική εκτέλεση τους. Τα αντικείμενα πόρων είναι εφοδιασμένα με παρόμοιους τελεστές, ώστε η κληρονομικότητα και η σύνθεση να παράγουν μια παρόμοια σύνθεση εργασιών. Τόσο οι πόροι όσο και οι εργασίες ορίζουν μια άλγεβρα, έτσι ο χρονοπρογραμματιστής (scheduler) θα μπορεί να την χρησιμοποιήσει για να πραγματοποιήσει υπολογισμούς και να βελτιστοποιήσει το σχέδιο εκτέλεσης. Ένα υποσύνολο των πόρων φαίνεται στην παρακάτω εικόνα, που δείχνει τις τέσσερις βασικές έννοιες του κόμβου (Node), της διεπαφής (Interface), του καναλιού (Channel) και της εφαρμογής (Application) από όπου και κληρονομούν οι περισσότεροι πόροι.



Εικόνα 20: Ιεραρχία πόρων στη νICN

3.2.3.3 Επεξεργαστής πόρων

Ο επεξεργαστής πόρων [14] αναλαμβάνει το βασικό ρόλο της προσαρμογής των αιτημάτων του χρήστη στις πολιτικές της πλατφόρμας και στους διαθέσιμους πόρους. Για παράδειγμα, ένας κόμβος μπορεί να εφαρμοστεί είτε ως ένα container είτε ως μια εικονική μηχανή (VM). Αυτή η επιλογή μπορεί είτε να καθοριστεί ρητά από τις προτιμήσεις του χρήστη είτε να συμπεραίνεται, από το ίδιο το εργαλείο, από τα συμφραζόμενα.

Ο επεξεργαστής πόρων είναι επίσης υπεύθυνος για τη χαρτογράφηση των πόρων που θα αναπτυχθούν στους διαθέσιμους φυσικούς εξυπηρετητές, ελέγχοντας όλους τους περιορισμούς και τις πολιτικές που απαιτούνται από τον χρήστη, τον προγραμματιστή ή τον πάροχο υποδομής. Μια τέτοια αρμοδιότητα μπορεί να εξομοιωθεί σε ένα πρόβλημα Περιορισμού-Ικανοποίησης (Constraint-Satisfaction Problem), καθώς το σύστημα πρέπει να φιλοξενήσει διάφορους πόρους σε ένα σύστημα πεπερασμένης χωρητικότητας όσον αφορά τη δικτύωση, τους υπολογισμούς και τη μνήμη. Τόσο τα χαρακτηριστικά που καθορίζονται από το χρήστη όσο και οι πολιτικές που καθορίζονται από τον πάροχο υποδομής, λαμβάνονται υπόψη στο πρόβλημα Περιορισμού-Ικανοποίησης ως πρόσθετοι περιορισμοί. Η έξοδος είναι μια χαρτογράφηση από την προδιαγραφή στην υλοποίηση, η οποία χρησιμοποιείται επίσης για να εκθέτει την παρακολούθηση του συστήματος πίσω στον χρήστη με τρόπο συνεπή.

3.2.3.4 Ενορχηστρωτής και χρονοπρογραμματιστής

Ο ρόλος του ενορχηστρωτή (orchestrator) [14] είναι ουσιαστικά να διατηρεί μια μηχανή πεπερασμένων καταστάσεων για κάθε πόρο και να εξασφαλίζει ότι αυτές θα φτάσουν στην κατάσταση που ζητήθηκε από τον χρήστη. Το αποτέλεσμα του είναι ένα γράφημα εξάρτησης εργασίας, το οποίο μοιράζεται με τον χρονοπρογραμματιστή (scheduler). Οι εξαρτήσεις εργασίας προέρχονται από τις εξαρτήσεις των πόρων, την δομή της μηχανής πεπερασμένων καταστάσεων καθώς και από τους περιορισμούς που θέτουν η κληρονομικότητα και η σύνθεση και που σχετίζονται με τους πόρους και τους χαρτογράφους.

Ο χρονοπρογραμματιστής εξασφαλίζει την επεκτασιμότητα της ανάπτυξης, προγραμματίζοντας την παράλληλη εκτέλεση των εργασιών από μια δεξαμενή νημάτων εργασίας (threads). Η προσέγγιση αυτή μοιάζει με ένα κλασσικό πρόβλημα προγραμματισμού κατευθυνόμενου μη-κυκλικού γραφήματος (DAG problem).

Λόγω της αρχιτεκτονικής της, η απόδοση της vICN επηρεάζεται επιπλέον και από τον χρόνο μετάδοσης του δικτύου. Προς ελάφρυνση αυτού του ζητήματος, γίνεται διαθέσιμη η ομαδοποίηση και κατηγοριοποίηση εργασιών όταν δύο συνεχόμενες εργασίες στοχεύουν στην ίδια διεπαφή ενός κόμβου. Η αλγεβρική δομή των εργασιών καθιστά επίσης δυνατή την αναδιοργάνωση της δομής των γραφημάτων για την βελτιστοποίηση της εκτέλεσης ή για την αυξημένη δυνατότητα ομαδοποίησης και κατηγοριοποίησης των εργασιών.

3.2.4 Εφαρμογή της vICN

3.2.4.1 Κώδικας

Η πρώτη έκδοση της vICN ήταν ανοιχτού κώδικα στα πλαίσια της ερευνητικής μελέτης Community ICN (CICN). Ο κώδικας είναι γραμμένος στην έκδοση 3 της γλώσσας προγραμματισμού Python και διατίθεται υπό την άδεια Apache v2.0. Αυτή η κυκλοφορία υλοποιεί όλα τα δομικά στοιχεία που φαίνονται στην εικόνα {17} και δύναται να κατασκευάσει αρκετά πολύπλοκες υλοποιήσεις ICN. Παράλληλα, διατίθεται μια προπαρασκευασμένη εικόνα LXD που περιέχει ολόκληρη την σουίτα CICN (συμπεριλαμβάνει μεταφορείς, στοίβα ICN, χρήσιμες εφαρμογές), έτσι ώστε ένα πλήρες ICN δίκτυο να μπορεί να ξεκινήσει σε ελάχιστο χρόνο. Επιπλέον, η σουίτα αυτή περιλαμβάνει έναν μεταφορέα υψηλής ταχύτητας που βασίζεται στο πλαίσιο VPP και ο οποίος μπορεί να χειρίζεται σχεδόν 1 εκατομμύριο πακέτα ανά δευτερόλεπτο (1Mrps) στην πρώτη του έκδοση.

3.2.4.2 Διαμοιρασμός

Η vICN είναι σε θέση να διαμοιράζει κόμβους και συνδέσεις που προσφέρονται από την υποδομή μέσω ενός συνόλου τεχνολογιών. Αυτό είναι ζωτικής σημασίας για την σωστή απομόνωση του πειράματος και για την υλοποίηση ξεχωριστών πλάνων ελέγχου και διαχείρισης.

Οι εικονικοί κόμβοι μπορούν να υλοποιηθούν είτε ως containers είτε ως εικονικές μηχανές. Τα containers (μέσω της χρήσης του LXD) μοιάζουν να είναι καλύτερη επιλογή για χρήση ως βασική τεχνολογία καθώς είναι πιο ελαφριά και αποδοτικά, χάρη στους μηχανισμούς zero-copy, το σύστημα αρχείων ZFS και την απλοποιημένη πρόσβαση σε φυσικούς πόρους του συστήματος. Τα θέματα ασφαλείας και οι περιορισμοί που μπορεί να προκύπτουν, όπως για παράδειγμα η κοινή χρήση του ίδιου πυρήνα (kernel), δεν είναι περιοριστικοί, αφού οι περισσότερες λειτουργίες του ICN υλοποιούνται με δικαιώματα χρήστη στο λογισμικό.

Το δίκτυο είναι διαμοιρασμένο στο επίπεδο 2 μέσω του λογισμικού Open vSwitch (OVS), που παρέχει προηγμένες λειτουργίες, όπως VLAN και κανόνες OpenFlow, απαραίτητες για τους ασύρματους εξομοιωτές και για την γεφύρωση εξωτερικών πραγματικών συσκευών με το εικονικό περιβάλλον. Η vICN απομονώνει πλήρως το δίκτυο που αναπτύσσει από τον έξω κόσμο δημιουργώντας μια μοναδική και απομονωμένη γέφυρα (bridge) ανα ανάπτυξη, χρησιμοποιώντας πίνακες IP (iptables) ως μέθοδο NAT για την παροχή εξωτερικής σύνδεσης. Επιπλέον, μειώνεται το φορτίο στην γέφυρα και απομονώνεται ο έλεγχος της κυκλοφορίας από το επίπεδο δεδομένων. Έτσι, τα συνδεδεμένα containers συνδέονται απευθείας μέσω ζευγών διεπαφών Virtual Ethernet (veth), παρακάμπτοντας έτσι τη γέφυρα. Εάν τα συνδεδεμένα containers δημιουργούνται σε διαφορετικούς εξυπηρετητές σε κάποιο σύμπλεγμα (cluster) τότε αυτά συνδέονται μέσω ενός GRE τούνελ.

Τελικώς, η vICN πρέπει να μεσολαβεί για τους κοινόχρηστους πόρους στον φυσικό κεντρικό υπολογιστή που εκτελείται, είτε πρόκειται για ονόματα container ή διεπαφών, αναγνωριστικά των VLAN, ακόμα και MAC ή IP διευθύνσεις ανάλογα με το επιθυμητό επίπεδο απομόνωσης του δικτύου. Είναι σημαντικό η ονοματοδοσία να γίνει σωστά όχι μόνο για λόγους ορθότητας, αλλά και για τον εντοπισμό και την αντιμετώπιση τυχόν σφαλμάτων. Ακόμα, η vICN επιβάλλει συνέπεια στα ονόματα που χαρακτηρίζουν μοναδικά έναν πόρο, ώστε να διευκολύνεται η ταχύτερη ανίχνευση και ανάκτηση όταν το εργαλείο επανεκκινείται ή πρέπει να ανακατασκευάσει το ίδιο πείραμα.

3.2.4.3 IP και ICN τοπολογίες

Χρησιμοποιώντας όλους τους παραπάνω μηχανισμούς, καθίσταται δυνατή η κατασκευή ενός διαγράμματος στο επίπεδο 2 του μοντέλου OSI όπου το vICN μπορεί να ρυθμίσει την σύνδεση IP και ICN. Για την IP δικτύωση, ένας κεντρικός IP πόρος καταμερισμού (Allocation Resource) αναλαμβάνει την κατανομή IP προθεμάτων και διευθύνσεων στα διάφορα τμήματα δικτύου του διαγράμματος. Έπειτα διασφαλίζεται η ευρύτερη IP συνδεσιμότητα μέσω του υπολογισμού των διαδρομών που θα εγκατασταθούν στους κόμβους. Το vICN παρέχει μια μονάδα δρομολόγησης που εφαρμόζει διάφορους αλγόριθμους (Dijkstra, Maximum-Flow) λαμβάνοντας ως είσοδο ένα διάγραμμα επιπέδου 2 και ένα σύνολο προθεμάτων (διευθύνσεις IP που έχουν κατανεμηθεί) και εξάγει ένα σύνολο διαδρομών που κωδικοποιούνται σαν πόροι του vICN. Η ρύθμιση των διαδρομών οδηγείται στη συνέχεια από έναν πίνακα δρομολόγησης (Routing Table) με ιδιότητες πόρου, από τον οποίο λαμβάνουν τους πίνακες δρομολόγησης τα Linux και το VPP.

Η διαδικασία είναι παρόμοια και για το ICN, εκτός από το γεγονός ότι κατασκευάζουμε αρχικά απεικονίσεις (IP ή Ethernet) βασισμένες σε διαμορφώσιμες προσεγγίσεις (π.χ L2 adjacency). Έπειτα μπορούμε να επαναχρησιμοποιήσουμε την ίδια μονάδα δρομολόγησης τροφοδοτώντας της με το διάγραμμα απεικονίσεων και το σύνολο προθεμάτων που βρίσκονται στους πόρους του παραγωγού περιεχομένου.

Ο αντίστοιχος πίνακας δρομολόγησης στην περίπτωση αυτή εφαρμόζεται από τον ICN forwarder. Σε αυτή τη περίπτωση, η χρήση συστημάτων δρομολόγησης πολλαπλών διαδρομών (multipath routing) έχει περισσότερο νόημα. Αυτή η διαδικασία έχει ως αποτέλεσμα μια ανάπτυξη που διευκολύνει τη συνύπαρξη IP και ICN και ως εκ τούτου επιτρέπει την σύγκριση των επιδόσεων των δύο αρχιτεκτονικών.

3.2.4.4 Εξομοίωση συνδέσεων

Μια λειτουργία που λείπει από τα περισσότερα παρόμοια εργαλεία είναι η δυνατότητα μέτρησης της απόδοσης των εφαρμογών που εκτελούνται πάνω σε εικονικά δίκτυα με συγκεκριμένο εύρος ζώνης ή καθυστερήσεις διάδοσης. Το vICN προσφέρει χαρακτηριστικά των πόρων στο επίπεδο ελέγχου κυκλοφορίας των Linux (Traffic Control Layer) ώστε να διαμορφώσει το εύρος ζώνης της σύνδεσης και να εξομοιώσει περιορισμένα δίκτυα.

Μια επιπλέον αξιοσημείωτη λειτουργία που προσφέρεται είναι η δυνατότητα χρήσης προσομοιωμένων ασύρματων πόρων ως εναλλακτική λύση, όπως συμβαίνει και στην πραγματικότητα. Οι δύο τύποι ασύρματων καναλιών που υποστηρίζονται προς το παρόν είναι το WiFi και το LTE και βασίζονται αμφότεροι σε λειτουργίες προσομοίωσης σε πραγματικό χρόνο του προσομοιωτή ns-3. Ο πόρος ασύρματου καναλιού του vICN συνδέει σταθμούς και σημείο πρόσβασης (ή όπως αναφέρονται UserEquipment και BaseStation) μέσω ενός διαμορφώσιμου ασύρματου καναλιού και αποκρύπτει την εσωτερική καλωδίωση από το χρήστη. Έπειτα η εξομοίωση φροντίζει για όλα τα χαρακτηριστικά μίας ασύρματης σύνδεσης όπως πλαίσιο συγχρονισμού, παρεμβολές ραδιοσυχνοτήτων, διαμάχες καναλιών, προσαρμογή ρυθμού και φορητότητα. Η προσομοίωση σε πραγματικό χρόνο επεκτείνεται με την χρήση πολλαπλών γεγονότων (instances) που ενορχηστρώνονται από έναν κύριο πόρο διαχείρισης της φορητότητας στο vICN, μέσω επικοινωνίας σε πραγματικό χρόνο με τους διάφορους εξομοιωτές. Διαμέσου αυτής της διαδικασίας συλλέγονται και εκτίθενται στο εσωτερικό μοντέλο πληροφορίες από την προσομοίωση, επιτυγχάνοντας έτσι την παρακολούθηση του συστήματος.

3.2.4.5 Παρακολούθηση (Monitoring)

Η παρακολούθηση [14] αποτελεί μέρος του vICN που εφαρμόζεται τοπικά ως εγκάρσια λειτουργία. Η γλώσσα ερωτήσεων που προσφέρεται από το vICN επιτρέπει την αναζήτηση των χαρακτηριστικών κάθε αντικείμενου, συμπεριλαμβανομένων και σχολιασμών που γίνονται από τον επεξεργαστή πόρων και τον ενορχηστρωτή σχετικά με την κατάσταση ανάπτυξης των πόρων. Η ίδια διεπαφή χρησιμοποιείται από τον ενορχηστρωτή και για την αναζήτηση της τρέχουσας κατάστασης ενός απομακρυσμένου πόρου και για την επικοινωνία με τους εξομοιωτές, ενώ από το χρήστη ώστε να αλληλεπιδρά με την πλατφόρμα για να αλλάξει ένα χαρακτηριστικό ή να δημιουργήσει ένα νέο πόρο κατά τη διάρκεια εκτέλεσης.

Όπως έχει προαναφερθεί, το συντακτικό της γλώσσας ερωτήσεων είναι όμοιο με αυτό της γλώσσας SQL. Πιο συγκεκριμένα, το αντικείμενο μίας ερώτησης περιέχει τα ακόλουθα στοιχεία: το όνομα αντικείμενου, έναν τύπο ερωτήματος (όπως CREATE, GET κ.α), ένα σύνολο φίλτρων και χαρακτηριστικών και τελικώς τις τιμές των χαρακτηριστικών που ορίζονται.

Για περιοδικές μετρήσεις, όπως η αξιοποίηση της σύνδεσης, παρέχεται επιπλέον ένα πρόγραμμα που εκτελείται στο υπόβαθρο (daemon) και μπορεί να εγκατασταθεί στους κόμβους ώστε να εκθέτει πληροφορίες μέσω μιας παρόμοιας διεπαφής. Η επικοινωνία μεταξύ των επιμέρους στοιχείων διασφαλίζεται με την χρήση από την vICN των ρυθμίσεων υποστρώματος του IP.

3.3 NS-3 (Network Simulator-3)

Ο ns-3 [15] είναι ένας προσομοιωτής δικτύων που χρησιμοποιείται κυρίως για ερευνητικούς και εκπαιδευτικούς λόγους. Πρόκειται για ένα ελεύθερο λογισμικό, υπό την άδεια GNU GPLv2 και διατίθεται δημοσίως για έρευνα, εκπαίδευση και χρήση. Ο προσομοιωτής διατηρείται από μια παγκόσμια ομάδα εθελοντών συντήρησης.

Ο στόχος του ns-3 είναι να διατηρεί ένα ανοιχτού κώδικα περιβάλλον προσομοίωσης κατάλληλο για έρευνα στον τομέα των δικτύων. Έτσι θα πρέπει να συμβαδίζει με τις ανάγκες προσομοίωσης της σύγχρονης έρευνας δικτύων και θα πρέπει να ενθαρρύνει την κοινοτική συμβολή, την αξιολόγηση από ομότιμα μέλη και την επικύρωση του λογισμικού.

3.3.1 Μοντέλα προσομοίωσης

Το πρόγραμμα του ns-3 δεσμεύεται για τη δημιουργία ενός πυρήνα προσομοίωσης που να είναι καλά τεκμηριωμένος, εύκολος στη χρήση και στον εντοπισμό σφαλμάτων και να εξυπηρετεί τις ανάγκες ολόκληρης της ροής των εργασιών προσομοίωσης, από τη ρύθμιση των παραμέτρων της προσομοίωσης έως τη συλλογή και ανάλυση των δεδομένων. Επιπλέον, η υποδομή του λογισμικού ενθαρρύνει την ανάπτυξη μοντέλων προσομοίωσης που είναι αρκετά ρεαλιστικά ώστε ο προσομοιωτής να χρησιμοποιείται και σαν εξομοιωτής πραγματικού χρόνου, διασυνδεδεμένος με τον πραγματικό κόσμο, επιτρέποντας την χρήση πολλών υλοποιήσεων πραγματικών πρωτοκόλλων εντός του ns-3.

Στον ns-3, ο πυρήνας της προσομοίωσης υποστηρίζει την έρευνα τόσο σε δίκτυα IP όσο και σε δίκτυα που δεν βασίζονται σε IP. Ωστόσο, η μεγάλη πλειοψηφία των χρηστών του επικεντρώνεται σε προσομοιώσεις ασύρματων και IP δικτύων που περιλαμβάνουν τα μοντέλα Wi-Fi, LTE ή άλλα ασύρματα συστήματα για τα επίπεδα 1 και 2 (OSI layers 1,2). Άλλα δημοφιλή ερευνητικά θέματα αφορούν την απόδοση του TCP και την απόδοση του πρωτοκόλλου δρομολόγησης ad hoc. Επίσης, υποστηρίζεται ένας χρονοπρογραμματιστής πραγματικού χρόνου για αλληλεπίδραση με πραγματικά συστήματα. Έτσι, οι χρήστες μπορούν να εκπέμπουν και να λαμβάνουν πακέτα που παράγονται από τον ns-3 σε πραγματικές συσκευές δικτύου και ο ns-3 μπορεί να χρησιμοποιηθεί ως πλατφόρμα διασύνδεσης μεταξύ εικονικών μηχανών (VMs).

3.3.2 Δομές WiFi και LTE

Οι κόμβοι μιας προσομοίωσης του ns-3 μπορούν να περιέχουν μια συλλογή αντικειμένων δικτύου (NetDevice), σαν ένας πραγματικός υπολογιστής να διαθέτει ξεχωριστές κάρτες διασύνδεσης για Ethernet, WiFi, Bluetooth. Με την προσθήκη αντικειμένων WifiNetDevice στους κόμβους του ns-3, μπορούμε να δημιουργήσουμε μοντέλα υποδομών βασισμένων στο IEEE 802.11.

Στα πλαίσια της δικής μας μελέτης, η δομή WiFi [16] του προσομοιωτή ns-3 ενσωματώνεται από το εργαλείο nICN για την εξομοίωση ασύρματων συνδέσεων μέσω WiFi. Συγκεκριμένα, τα πρότυπα που διατίθενται είναι τα 802.11a/b/g/n/ac καθώς και κάποια επιπλέον πρότυπα όπως 802.11_5MHZ, 802.11_10MHZ, 802.11_5GHZ και 802.11n_2_4GHZ. Αντίστοιχα, υπάρχει και η δομή LTE του προσομοιωτή ns-3 μέσω της οποίας μπορούμε να προσομοιώσουμε ασύρματη επικοινωνία.

Κεφάλαιο 4: Τοπολογίες, σενάρια και παράμετροι

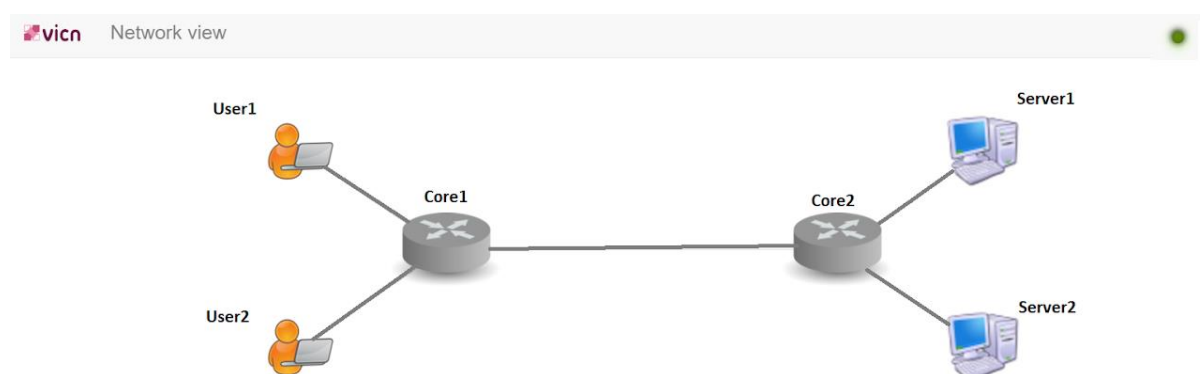
4.1 Εισαγωγή

Στο κεφάλαιο αυτό θα παρουσιάσουμε τις τοπολογίες που δημιουργήσαμε μαζί με τον κώδικα JSON που χρειάζεται για την ανάπτυξη της κάθε μίας από αυτές. Λόγω της ευανάγνωστης από τους ανθρώπους μορφής των JSON θα αναλύσουμε κάθε βασικό χαρακτηριστικό που εισάγουμε στην τοπολογία. Στο δεύτερο μέρος του κεφαλαίου θα αναφερθούμε στα σενάρια που ορίσαμε για κάθε τοπολογία και έπειτα σε κάποιες παραμέτρους που θέσαμε στο σύστημα.

4.2 Τοπολογίες

Στα πλαίσια της συγκεκριμένης μελέτης δημιουργήθηκαν τρεις κοινές τοπολογίες, μια απλή dumbbell, μια dumbbell πολλαπλών διαδρομών και μια tree. Στα πειράματα που διεξάχθηκαν, για κάθε μια από τις παραπάνω τοπολογίες εισάγαγαμε τριών ειδών επικοινωνίες, ενσύρματη, ασύρματη μέσω Wi-Fi και ασύρματη μέσω LTE. Αρχικά όμως για να κατανοήσουμε τις τοπολογίες μια προς μια, θα πρέπει να τις οπτικοποιήσουμε και να παραθέσουμε τα βασικά συστατικά στοιχεία που τις συνθέτουν.

4.2.1 Dumbbell

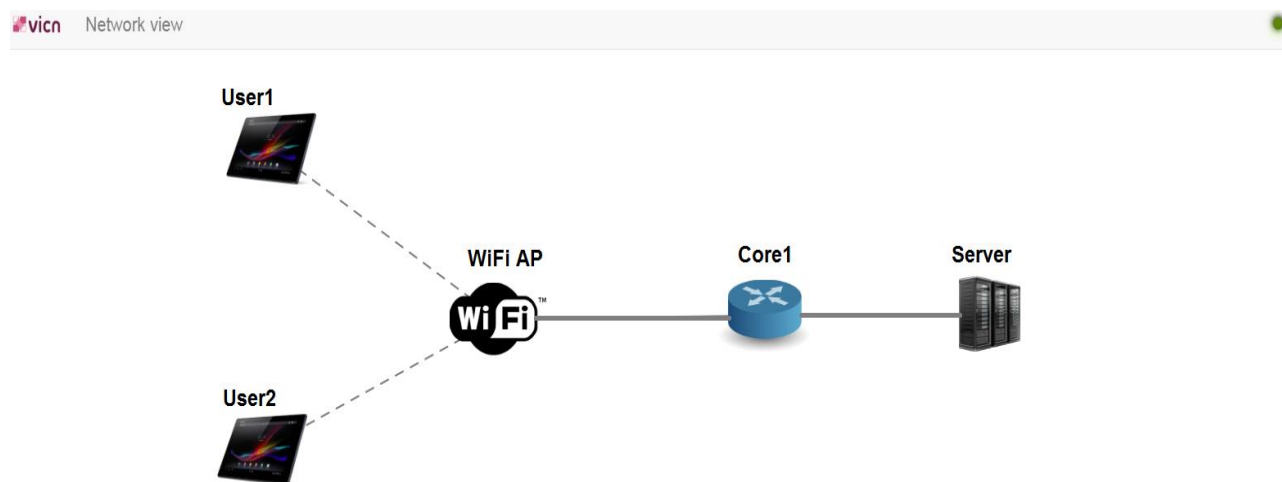


Εικόνα 21: Ενσύρματη Dumbbell τοπολογία

Σε αυτή την κοινή δικτυακή τοπολογία dumbbell περιλαμβάνονται δύο παραγωγοί περιεχομένου (Server1, Server2) και δύο καταναλωτές (User1, User2). Έτσι, στο συγκεκριμένο σύστημα και με την βοήθεια των εφαρμογών producer-test και iget, οι κόμβοι παραγωγοί παράγουν και εκθέτουν περιεχόμενο στο δίκτυο και οι κόμβοι καταναλωτές ζητούν και παραλαμβάνουν το περιεχόμενο που επιθυμούν, εφόσον είναι διαθέσιμο.

Επιπλέον, υπάρχει η σύνδεση των παραπάνω κόμβων με δύο δρομολογητές (Core1, Core2) οι οποίοι προωθούν την κυκλοφορία των δεδομένων και επιπλέον διαθέτουν κρυφή μνήμη για μελλοντική εξυπηρέτηση του τοπικά αποθηκευμένου περιεχομένου. Θα πρέπει να αναφερθεί ότι η σύνδεση μεταξύ Core1-Core2 αποτελεί και το bottleneck του δικτύου.

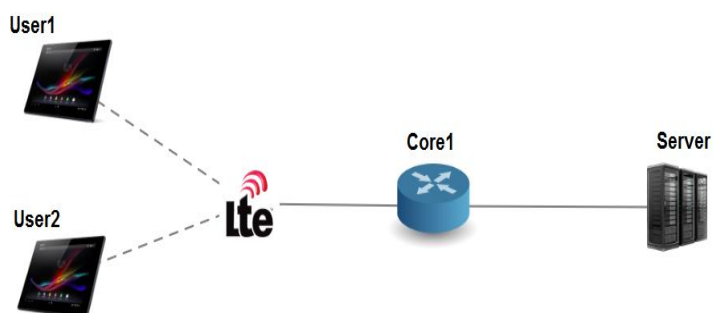
Παρόμοια είναι η τοπολογία και με την χρήση ασύρματης επικοινωνίας μέσω WiFi. Εδώ οι δύο καταναλωτές είναι ασύρματα συνδεδεμένοι σε ένα WiFi Access Point, το οποίο με την σειρά του είναι συνδεδεμένο με τον δρομολογητή (Core1). Στην άλλη πλευρά της τοπολογίας υπάρχει ένας παραγωγός (Server), όπου και είναι αρχικά διαθέσιμο το περιεχόμενο.



Εικόνα 22: Ασύρματη WiFi Dumbbell τοπολογία

Ακριβώς στην ίδια λογική είναι και η ασύρματη επικοινωνία μέσω LTE με την μόνη διαφορά ότι έχουμε ένα ασύρματο Access Point για LTE συνδέσεις.

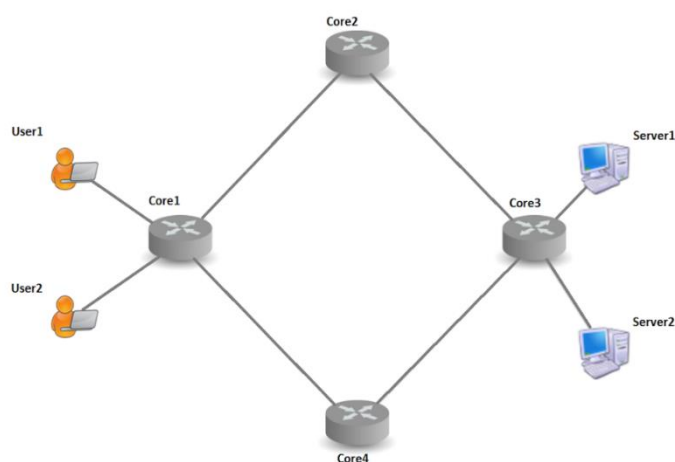
vicn Network view



Εικόνα 23: Ασύρματη LTE Dumbbell τοπολογία

4.2.2 Dumbbell πολλαπλών διαδρομών

vicn Network view

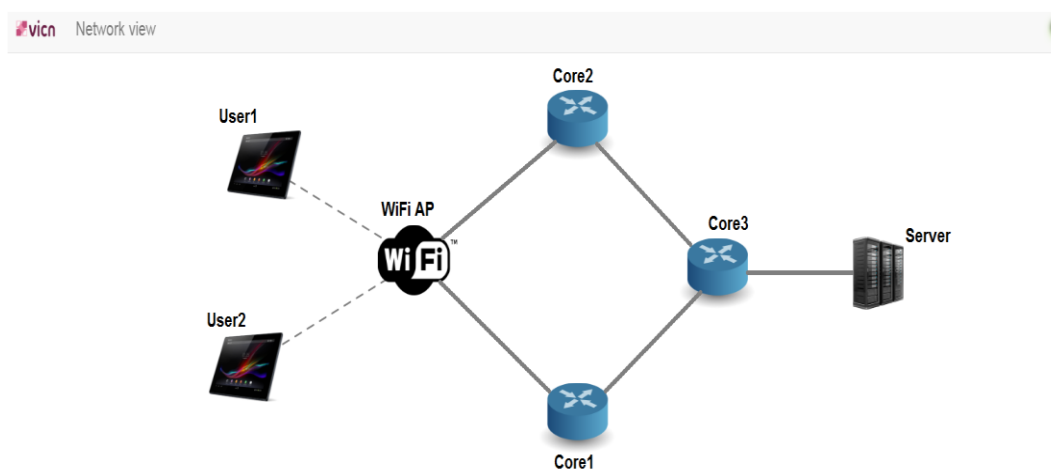


Εικόνα 24: Ενσύρματη Dumbbell πολλαπλών διαδρομών τοπολογία

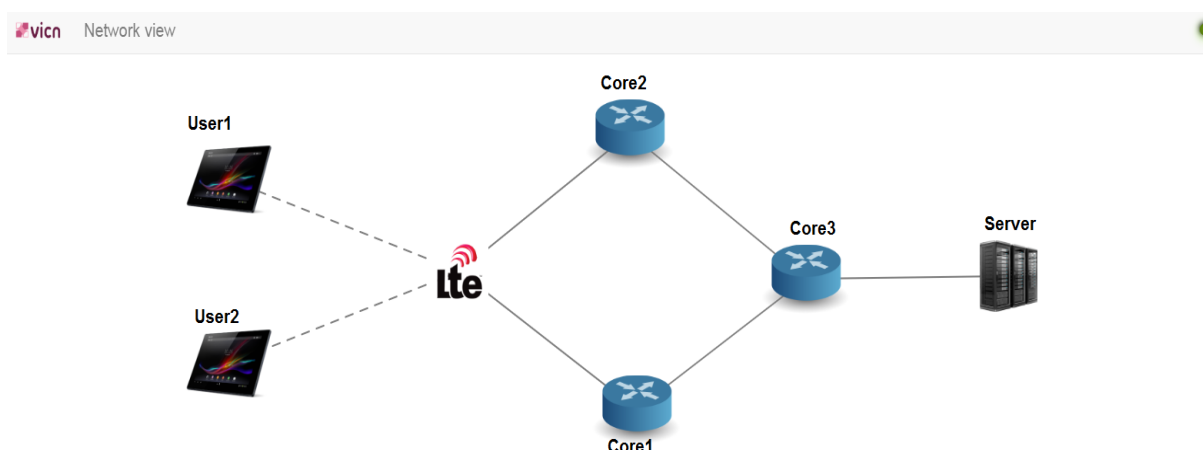
Στην κοινή αυτή δικτυακή τοπολογία dumbbell με δύο διαδρομές (multipath) περιλαμβάνονται δύο παραγωγοί περιεχομένου (Server1, Server2) και δύο καταναλωτές (User1, User2). Η διαφορά με την τοπολογία dumbbell που παρουσιάστηκε παραπάνω είναι ότι εδώ υπάρχουν τέσσερις δρομολογητές (Core1, Core2, Core3, Core4) που με τον τρόπο που έχουν τοποθετηθεί, δημιουργούν δύο διαδρομές για την εξυπηρέτηση των καταναλωτών.

Ένα βασικό πλεονέκτημα που προκύπτει από την χρήση της συγκεκριμένης τοπολογίας είναι ότι δύναται να εξαλειφθεί το bottleneck από το δίκτυο. Επιπλέον, γίνεται εφικτή η χρήση της στρατηγικής load balancing που μπορεί να αποφέρει σημαντικές βελτιώσεις στην απόδοση του συστήματος.

Ακριβώς το ίδιο συμβαίνει και κατά την ασύρματη σύνδεση των καταναλωτών στο Access Point μέσω WiFi αλλά και μέσω LTE, όπως φαίνεται στις δύο παρακάτω τοπολογίες.

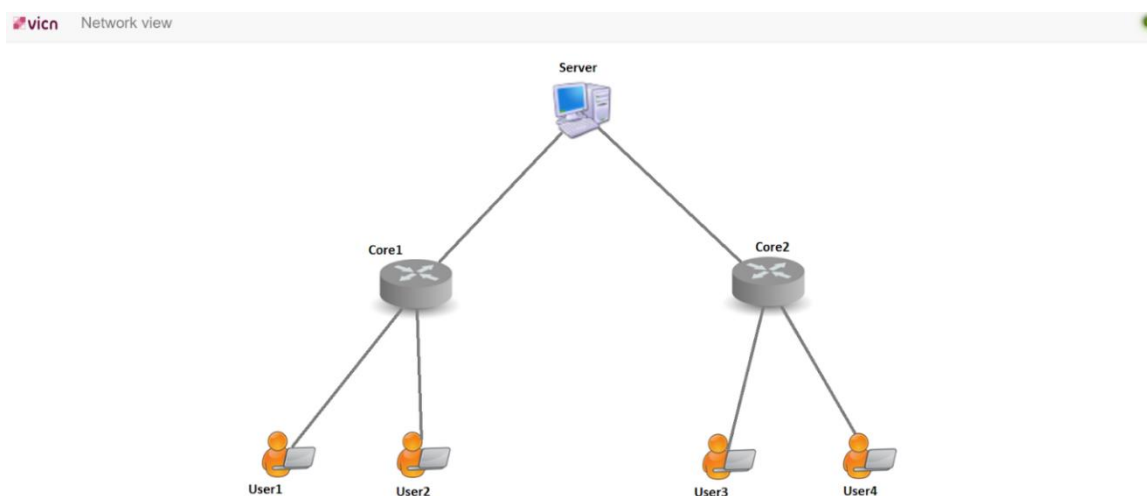


Εικόνα 25: Ασύρματη WiFi Dumbbell πολλαπλών διαδρομών τοπολογία



Εικόνα 26: Ασύρματη LTE Dumbbell πολλαπλών διαδρομών τοπολογία

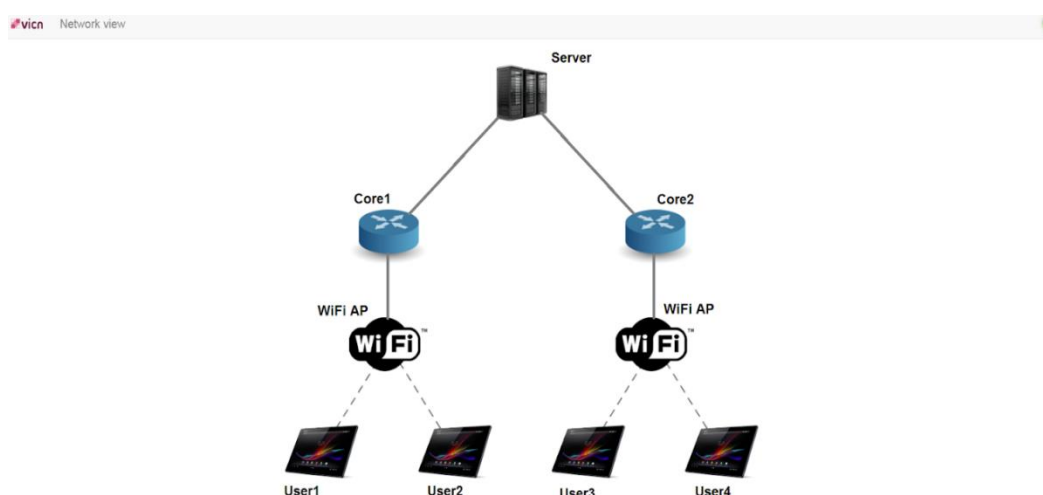
4.2.3 Tree 2-επιπέδων



Εικόνα 27: Ενσύρματη Tree τοπολογία

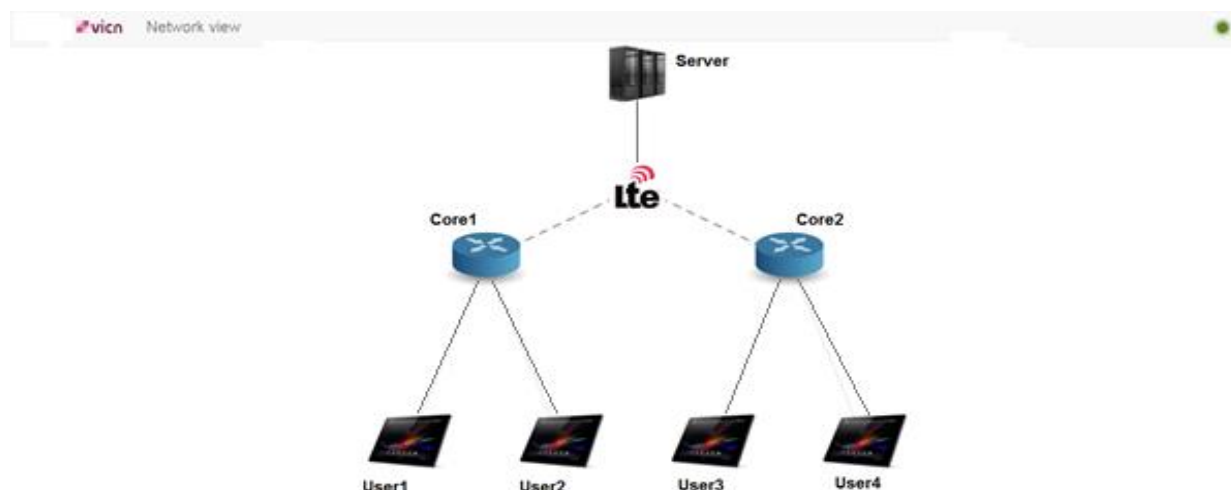
Τις τοπολογίες που δημιουργήθηκαν στα πλαίσια της μελέτης συμπληρώνει η τοπολογία tree 2-επιπέδων, όπου υπάρχει ένας παραγωγός (Server1) συνδεδεμένος με δύο δρομολογητές (Core1, Core2). Κάθε ένας από τους δρομολογητές εξυπηρετεί δύο καταναλωτές (User1, User2, User3, User4).

Η ίδια τοπολογία με εφαρμογή ασύρματης σύνδεσης των καταναλωτών στο Access Point μέσω WiFi δεν διαφέρει πολύ από την προηγούμενη και παρουσιάζεται παρακάτω.



Εικόνα 28: Ασύρματη WiFi Tree τοπολογία

Κατά την χρήση ασύρματης σύνδεσης μέσω LTE μεταφέραμε το Access Point πριν από τους δρομολογητές (Core Layer) λόγω αδυναμίας προσομοίωσης δύο LTE Access Points στην ίδια τοπολογία.



Εικόνα 29: Ασύρματη LTE Tree τοπολογία

4.2.4 Δημιουργία τοπολογιών

Προκειμένου να δημιουργήσουμε μια τοπολογία απαιτείται να δηλώσουμε ξεκάθαρα σε ένα αρχείο JSON όλα τα επιμέρους στοιχεία που συντελούν στην συγκεκριμένη τοπολογία. Στην ενότητα αυτή θα παραθέσουμε τον κώδικα που χρησιμοποιήθηκε για την κατασκευή της κοινής τοπολογίας *dumbbell* που προαναφέρθηκε.

- Το πρώτο κομμάτι ονομάζεται *settings* (ρυθμίσεις) και σε αυτό δηλώνουμε το υποδίκτυο που θα δημιουργήθει κατά την ανάπτυξη και μια παράμετρο που χρησιμοποιείται από το λογισμικό για την επεξεργασία των διαδικασιών.

```

"settings": {
    "network": "192.168.128.0/24",
    "ulimit-n": 10000
}

```

- Το δεύτερο κομμάτι ονομάζεται *resources* (πόροι) και είναι αυτό στο οποίο καθορίζουμε όλους τους πόρους που θα αναπτυχθούν, αλλά και τις σχέσεις μεταξύ τους. Είναι μεγαλύτερης κρισιμότητας από το πρώτο καθώς σε αυτό το σημείο ουσιαστικά χτίζουμε όλο το σύστημα και πρέπει να δώσουμε ιδιαίτερη προσοχή ώστε να έχουμε τα αποτελέσματα που επιθυμούμε.

```
"resources": [
  {
    "type": "Physical",
    "name": "server",
    "hostname": "localhost"
  },
  {
    "type": "NetDevice",
    "device_name": "enp16s0f0",
    "managed": false,
    "node": "server"
  }
]
```

Στο παραπάνω κομμάτι κώδικα αρχικά δηλώνουμε το φυσικό εξυπηρετητή με το όνομα *localhost*, στον οποίο η ανάπτυξη λαμβάνει χώρα σε τοπικό περιβάλλον. Έπειτα δηλώνουμε μια νέα δικτυακή διεπαφή για την σύνδεση του κεντρικού εξυπηρετητή στο διαδίκτυο, το όνομα αυτής (*enp16s0f0*), καθώς και τον εξυπηρετητή στον οποίο ανήκει.

Παρακάτω δηλώνουμε μια εικόνα Linux Container (LXC), συμπεριλαμβάνουμε ουσιαστικά στην ανάπτυξή μας μια έκδοση της CICN σουίτας που έχει προαναφερθεί ότι προσφέρεται. Έπειτα δημιουργούμε ένα γκρουπ από πόρους με το όνομα *routed*. Ο επόμενος πόρος που δηλώνουμε είναι αυτός της απεικόνισης του συστήματος σε γραφικό περιβάλλον μέσω μιας συγκεκριμένης θύρας.

Οι επόμενοι πόροι που δεσμεύονται είναι δύο *containers*, δύο κόμβοι δηλαδή τους συστήματος μας. Τα κοινά χαρακτηριστικά τους είναι ότι βρίσκονται στο ίδιο γκρουπ πόρων, ανήκουν στον ίδιο φυσικό εξυπηρετητή και δημιουργούνται με βάση την ίδια LXC εικόνα. Αυτό που τα κάνει να ξεχωρίζουν είναι πρωτίστως το όνομα τους, αλλά και η κατηγορία που ανήκουν με το ένα από αυτά να είναι χρήστης (*user*) ενώ το άλλο δρομολογητής (*ip-router*).

Οι μεταβλητές x και y χρησιμοποιούνται από τον πόρο απεικόνισης για την τοποθέτηση των κόμβων στο γραφικό περιβάλλον.

```
{
  "type": "LxcImage",
  "name": "ubuntu1604-cicnsuite-rc4",
  "image": "ubuntu1604-cicnsuite-rc4",
  "node": "server"
},
{
  "type": "Group",
  "name": "routed"
},
{
  "type": "GUI",
  "groups": ["routed"],
  "port": 8000
},
{
  "type": "LxcContainer",
  "name": "core1",
  "groups": [ "routed" ],
  "image": "ubuntu1604-cicnsuite-rc4",
  "category": "ip-router",
  "y": 10,
  "x": 5,
  "node": "server"
},
{
  "type": "LxcContainer",
  "name": "user1",
  "groups": [ "routed" ],
  "image": "ubuntu1604-cicnsuite-rc4",
  "category": "user",
  "y": 8,
  "x": 2,
  "node": "server"
}
```

Με ίδια ακριβώς λογική δημιουργούνται και οι υπόλοιποι τέσσερις κόμβοι της συγκεκριμένης τοπολογίας.

```

{
  "type": "LxcContainer",
  "name": "user2",
  "groups": [ "routed" ],
  "image": "ubuntu1604-cicnsuite-rc4",
  "category": "user",
  "y": 12,
  "x": 2,
  "node": "server"
},
{
  "type": "LxcContainer",
  "name": "core2",
  "groups": [ "routed" ],
  "image": "ubuntu1604-cicnsuite-rc4",
  "category": "ip-router",
  "y": 10,
  "x": 13,
  "node": "server"
}

```

```

{
  "type": "LxcContainer",
  "name": "server1",
  "groups": [ "routed" ],
  "image": "ubuntu1604-cicnsuite-rc4",
  "category": "server",
  "y": 8,
  "x": 16,
  "node": "server"
},
{
  "type": "LxcContainer",
  "name": "server2",
  "groups": [ "routed" ],
  "image": "ubuntu1604-cicnsuite-rc4",
  "category": "server",
  "y": 12,
  "x": 16,
  "node": "server"
}

```

Εν συνεχεία, για κάθε έναν από τους έξι κόμβους που δηλώσαμε θα πρέπει να προσθέσουμε και έναν πόρο μεταφορέα (*Metis Forwarder*) ο οποίος είναι υπεύθυνος να διαβιβάζει τα πακέτα μεταξύ των κόμβων χρησιμοποιώντας διάφορες ειδικές εφαρμογές, όπως πρωτόκολλα δρομολόγησης, πάνω στον κόμβο. Επιπλέον, κατά τη δήλωση του πόρου μεταφορέα μπορούμε να εισάγουμε στον κόμβο και κρυφή μνήμη (μετρούμενη σε KB). Όπως φαίνεται παρακάτω, στην δική μας τοπολογία οι παραγωγοί και οι καταναλωτές δεν διαθέτουν κρυφή μνήμη, ενώ οι δρομολογητές διαθέτουν κρυφή μνήμη 20MB.

```

{
  "type": "MetisForwarder",
  "cache_size": 0,
  "node": "user1"
},
{
  "type": "MetisForwarder",
  "cache_size": 0,
  "node": "user2"
},
{
  "type": "MetisForwarder",
  "cache_size": 20000,
  "node": "core1"
}

```

```

{
  "type": "MetisForwarder",
  "cache_size": 0,
  "node": "server1"
},
{
  "type": "MetisForwarder",
  "cache_size": 0,
  "node": "server2"
},
{
  "type": "MetisForwarder",
  "cache_size": 20000,
  "node": "core2"
}

```


Επόμενος πόρος που απαιτείται είναι αυτός της εφαρμογής ενός ICN HTTP εξυπηρετητή, ώστε να μπορεί ο παραγωγός να εκθέτει το περιεχόμενο που κατέχει και ο καταναλωτής να το απορροφά. Για το λόγο αυτό, δηλώνουμε και το πρόθεμα το οποίο χρησιμοποιεί ο εξυπηρετητής όταν παράγει δεδομένα.

```
{
  "type": "WebServer",
  "prefixes": [
    "/webserver1"
  ],
  "node": "server1"
}
{
  "type": "WebServer",
  "prefixes": [
    "/webserver2"
  ],
  "node": "server2"
}
```

Αφού έχουμε δηλώσει όλους τους κόμβους που θέλουμε να δημιουργήσουμε, θα πρέπει να δηλώσουμε και τις συνδέσεις μεταξύ τους. Οι συνδέσεις αποτελούν ξεχωριστούς πόρους που δεσμεύονται από το σύστημα και σε αυτούς δηλώνουμε τους δύο κόμβους που ενώνουν (χωρίς να μας ενδιαφέρει ποιός κόμβος δηλώνεται σαν πηγή και ποιός σαν προορισμός), τη μέγιστη δυνατή ταχύτητα μετάδοσης δεδομένων σε εκατομμύρια bits ανά δευτερόλεπτο (Mbps), καθώς και το γκρουπ στο οποίο ανήκει.

```
{
  "type": "Link",
  "groups": [ "routed" ],
  "dst_node": "user1",
  "capacity": 50,
  "src_node": "core1"
},
{
  "type": "Link",
  "groups": [ "routed" ],
  "dst_node": "user2",
  "capacity": 50,
  "src_node": "core1"
},
{
  "type": "Link",
  "groups": [ "routed" ],
  "dst_node": "server1",
  "capacity": 50,
  "src_node": "core2"
}
{
  "type": "Link",
  "groups": [ "routed" ],
  "dst_node": "server2",
  "capacity": 50,
  "src_node": "core2"
},
{
  "type": "Link",
  "groups": [ "routed" ],
  "dst_node": "core1",
  "capacity": 50,
  "src_node": "core2"
}
```

Οι δύο τελευταίοι πόροι που περιέχονται στο αρχείο JSON της dumbbell τοπολογίας ονομάζονται *CentrallP* και *CentrallCN*. Ο πρώτος χρησιμεύει για την ανάθεση IP διευθύνσεων και την εγκαθίδρυση της IP δρομολόγησης στην τοπολογία, ενώ ο δεύτερος για την σύσταση των ICN διαδρομών και διεπαφών.

```
{
  "type": "CentrallP",
  "groups": [ "routed" ],
  "ip_routing_strategy": "spt",
  "ip6_data_prefix": "9002::/16",
  "ip4_data_prefix": "192.168.15.0/24"
}

{
  "type": "CentralCN",
  "groups": [ "routed" ],
  "face_protocol": "udp4"
}
```

4.2.5 Παρακολούθηση τοπολογίας

Αφού αναπτύξουμε την συγκεκριμένη τοπολογία εισάγοντας στο αρχείο JSON τον κώδικα που παραθέσαμε στην προηγούμενη ενότητα, είναι δυνατόν να δημιουργήσουμε ροή δεδομένων και να παρακολουθήσουμε το σύστημα εν δράση. Αρχικά, μπορούμε να δούμε ότι η ανάπτυξη ολοκληρώθηκε και όλοι οι κόμβοι και οι συνδέσεις δημιουργήθηκαν όπως φαίνεται παρακάτω.

```
alimus89@chris-pc:~$ lxc list
```

NAME	STATE	IPV4	IPV6	TYPE	SNAPSHOTS
core1	RUNNING	192.168.15.9 (core2) 192.168.15.3 (user2) 192.168.15.1 (user1) 10.21.241.107 (vici_mgmt)	fd42:a4cb:799d:5243:216:3eff:fe8a:f05f (vici_mgmt) 9002::2 (user1) 9002:0:0:4::2 (core2) 9002:0:0:1::2 (user2)	PERSISTENT	0
core2	RUNNING	192.168.15.8 (core1) 192.168.15.7 (server2) 192.168.15.5 (server1) 10.21.241.91 (vici_mgmt)	fd42:a4cb:799d:5243:216:3eff:feae:b5e5 (vici_mgmt) 9002:0:0:4::1 (core1) 9002:0:0:3::2 (server2) 9002:0:0:2::2 (server1)	PERSISTENT	0
server1	RUNNING	192.168.15.4 (core2) 10.21.241.86 (vici_mgmt)	fd42:a4cb:799d:5243:216:3eff:fe28:c319 (vici_mgmt) 9002:0:0:2::1 (core2)	PERSISTENT	0
server2	RUNNING	192.168.15.6 (core2) 10.21.241.207 (vici_mgmt)	fd42:a4cb:799d:5243:216:3eff:feb7:c3a1 (vici_mgmt) 9002:0:0:3::1 (core2)	PERSISTENT	0
user1	RUNNING	192.168.15.0 (core1) 10.21.241.200 (vici_mgmt)	fd42:a4cb:799d:5243:216:3eff:fe0a:b74d (vici_mgmt) 9002::1 (core1)	PERSISTENT	0
user2	RUNNING	192.168.15.2 (core1) 10.21.241.203 (vici_mgmt)	fd42:a4cb:799d:5243:216:3eff:fe67:27c0 (vici_mgmt) 9002:0:0:1::1 (core1)	PERSISTENT	0

Εικόνα 30: Λίστα containers των κόμβων της ανάπτυξης

Έπειτα, μπορούμε να εισέλθουμε σε κάθε ένα container από τα παραπάνω και να ενεργοποιήσουμε εφαρμογές. Στο συγκεκριμένο σύντομο παράδειγμα που φαίνεται στις επόμενες εικόνες, έχουμε εκκινήσει ένα απλό σενάριο στο οποίο ο παραγωγός *server1* εκθέτει περιεχόμενο μεγέθους 20MB στο δίκτυο ενεργώντας ως *webserver* και ο καταναλωτής *user1* ζητά και παίρνει αυτό το περιεχόμενο. Κατ' αντιστοιχία, το ίδιο κάνουν και ο *server2* με τον *user2*.

```
alimus89@chris-pc:~$ lxc shell server1
mesg: ttyname failed: Success
root@server1:~# producer-test -s 20000000 ccnx:/webserver1/get/content1
Setting name.. ccnx:/webserver1/get/content1
Route set correctly!
```

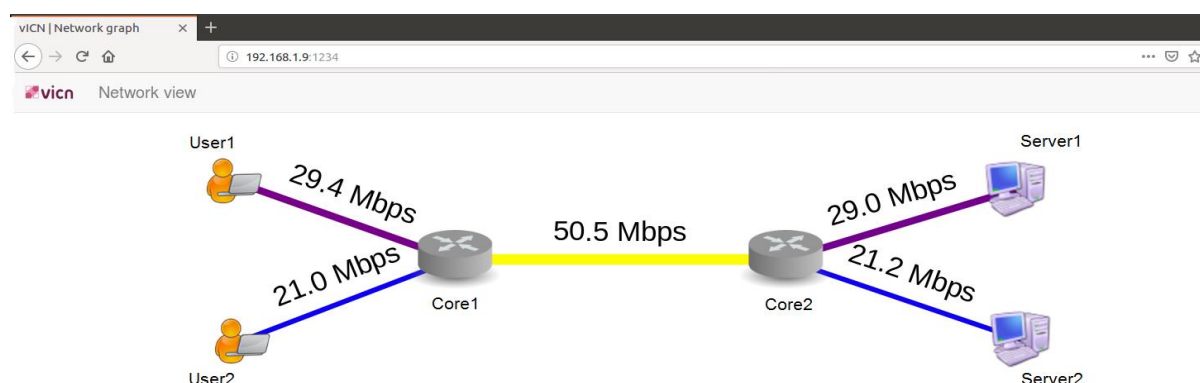
Εικόνα 31: Εντολή εκκίνησης παραγωγής περιεχομένου από τον παραγωγό

```
alimus89@chris-pc:~$ lxc shell user1
mesg: ttyname failed: Success
root@user1:~# iget http://webserver1/content1
```

Εικόνα 32: Εντολή εκκίνησης κατανάλωσης περιεχομένου από τον καταναλωτή

Οι δύο παραπάνω εικόνες απεικονίζουν τη διαδικασία της δημιουργίας και διαφήμισης του περιεχομένου από τον παραγωγό και τη διαδικασία απορρόφησής του από τον καταναλωτή αντίστοιχα.

Τέλος, η παρακάτω εικόνα απεικονίζει το εργαλείο ζωντανής παρακολούθησης του συστήματος. Σε αυτό φαίνονται όλες οι ροές κατά τη διάρκεια μετάδοσης των δεδομένων μαζί με τις ταχύτητες μετάδοσης. Όπως φαίνεται, το εργαλείο παρακολούθησης γίνεται διαθέσιμο τοπικά μέσω ενός φυλλομετρητή σε μια συγκεκριμένο θύρα που ορίζεται από τον χρήστη.



Εικόνα 33: Εργαλείο παρακολούθησης ανάπτυξης

4.3 Σενάρια

Προκειμένου να εξετάσουμε τις τοπολογίες που δημιουργήσαμε και να βγάλουμε κάποια αποτελέσματα όσον αφορά την απόδοση του συστήματος, είναι απαραίτητο να δημιουργήσουμε κάποια σενάρια. Τα σενάρια που εξετάστηκαν είναι κοινά για όλες τις τοπολογίες που έχουν παρουσιαστεί στην προηγούμενη ενότητα. Πρόκειται για τρία βασικά σενάρια με κάποια υποσενάρια και παρουσιάζονται παρακάτω.

Οι δύο μετρήσεις που μας ενδιαφέρουν κατά την διεξαγωγή των πειραμάτων είναι ο πραγματικός όγκος πληροφοριών που μεταφέρεται ανά μονάδα χρόνου (*Throughput*) και ο χρόνος που απαιτείται για την μεταφορά των ζητούμενων πληροφοριών (*Delay*).

1^ο σενάριο: Μεταβολή *cache size*

Κρατάμε σταθερό το μέγιστο όγκο πληροφοριών που δύναται να μεταφερθεί ανά μονάδα χρόνου (*50 Mbps Bandwidth*) και το μέγεθος του περιεχομένου που ζητείται (*10 MB*). Η τιμή που μεταβάλλεται είναι το μέγεθος της κρυφής μνήμης (*cache size*) των δρομολογητών του συστήματος.

Περαιτέρω χωρίζουμε το παρόν σενάριο στα δύο παρακάτω υποσενάρια:

- Οι καταναλωτές ζητούν ταυτόχρονα το ίδιο περιεχόμενο από τον ίδιο παραγωγό.
- Οι καταναλωτές ζητούν ταυτόχρονα διαφορετικό περιεχόμενο από διαφορετικό παραγωγό. Στην περίπτωση της *tree* τοπολογίας ζητούν διαφορετικό περιεχόμενο από τον μοναδικό παραγωγό.

Προκειμένου να πετύχουμε τη χρήση της κρυφής μνήμης, εκτελούμε το κάθε ένα από τα δύο υποσενάρια δύο φορές ώστε στην πρώτη να έχουμε την αποθήκευση του περιεχομένου στην κρυφή μνήμη και στην δεύτερη να έχουμε κατανάλωση του περιεχομένου κατευθείαν από την κρυφή μνήμη. Τελικά, παίρνουμε τον μέσο όρο των μετρήσεων για κάθε υποσενάριο.

2^ο σενάριο: Μεταβολή bandwidth

Κρατάμε σταθερό το μέγεθος της κρυφής μνήμης (20 MB *cache size*) των δρομολογητών του συστήματος και το μέγεθος του περιεχομένου που ζητείται (10 MB). Η τιμή που μεταβάλλεται είναι ο μέγιστος όγκος πληροφοριών που δύναται να μεταφερθεί ανά μονάδα χρόνου (*Bandwidth*).

Περαιτέρω χωρίζουμε το παρόν σενάριο στα δύο παρακάτω υποσενάρια:

- Οι καταναλωτές ζητούν ταυτόχρονα το ίδιο περιεχόμενο από τον ίδιο παραγωγό.
- Οι καταναλωτές ζητούν ταυτόχρονα διαφορετικό περιεχόμενο από διαφορετικό παραγωγό. Στην περίπτωση της *tree* τοπολογίας ζητούν διαφορετικό περιεχόμενο από τον μοναδικό παραγωγό.

Προκειμένου να πετύχουμε τη χρήση της κρυφής μνήμης, εκτελούμε το κάθε ένα από τα δύο υποσενάρια δύο φορές ώστε στην πρώτη να έχουμε την αποθήκευση του περιεχομένου στην κρυφή μνήμη και στην δεύτερη να έχουμε κατανάλωση του περιεχομένου κατευθείαν από την κρυφή μνήμη. Τελικά, παίρνουμε τον μέσο όρο των μετρήσεων για κάθε υποσενάριο.

3^ο σενάριο: Μεταβολή αριθμού περιεχομένων

Κρατάμε σταθερό το μέγεθος της κρυφής μνήμης (20MB *cache size*) των δρομολογητών του συστήματος, το μέγιστο όγκο πληροφοριών που δύναται να μεταφερθεί ανά μονάδα χρόνου (50 Mbps *Bandwidth*) και το μέγεθος του περιεχομένου που ζητείται (10 MB). Εδώ η μεταβλητή είναι ο αριθμός των διαθέσιμων περιεχομένων στον παραγωγό.

Οι καταναλωτές απορροφούν, ανά τακτά χρονικά διαστήματα, τυχαία ένα από τα διαθέσιμα περιεχόμενα του παραγωγού. Μερικά από τα αιτήματα ικανοποιούνται από τις κρυφές μνήμες του δικτύου και άλλα από τον παραγωγό. Τελικώς, υπολογίζουμε τον μέσο όρο των μετρήσεων για κάθε καταναλωτή.

4.4 Παράμετροι

Κατά την διάρκεια της εκτέλεσης των πειραμάτων και τις εξαγωγής των μετρήσεων, επιλέξαμε να θέσουμε σε κάποιες παραμέτρους τους συστήματος σταθερές τιμές έτσι ώστε όλα τα πειράματα να διεξαχθούν κάτω από τις ίδιες συνθήκες.

4.4.1 Πολιτική αντικατάστασης κρυφής μνήμης

Η πρώτη παράμετρος αφορά την πολιτική αντικατάστασης κρυφής μνήμης του συστήματος. Η παράμετρος αυτή υποστηρίζει δύο τιμές, δύο αλγόριθμους δηλαδή για την εφαρμογή της πολιτικής αντικατάστασης.

Ο πρώτος αλγόριθμος είναι ο *Least Recently Used* (LRU) [17], ο οποίος όπως φαίνεται και από το όνομα του απορρίπτει τα λιγότερο πρόσφατα χρησιμοποιημένα αντικείμενα από τη μνήμη. Ο αλγόριθμος αυτός παρακολουθεί τότε χρησιμοποιήθηκε κάθε ένα από τα αντικείμενα της μνήμης και ανάλογα αποφασίζει ποιό θα απορριφθεί. Παρότι είναι υπολογιστικά πιο ακριβός από τον δεύτερο, είναι αυτός που χρησιμοποιήσαμε καθόλη της διάρκεια της μελέτης καθώς παρατηρήσαμε αποδοτικότερη εξυπηρέτηση από τις κρυφές μνήμες.

Ο δεύτερος αλγόριθμος είναι ο *First In First Out* (FIFO) [17], ο οποίος λειτουργεί ακριβώς όπως μια ουρά, όπου το πρώτο και παλαιότερο αντικείμενο που εισήχθει στην κρυφή μνήμη είναι και αυτό που αντικαθιστάται όταν αυτός απαιτείται. Παρόλο που τον δοκιμάσαμε, απορρίψαμε τη χρήση του καθώς η πολιτική LRU είναι πιο αποτελεσματική στη διαχείριση μνήμης τύπου cache.

4.4.2 Έλεγχος συμφόρησης

Η δεύτερη παράμετρος αφορά τον αλγόριθμο που χρησιμοποιείται για τον έλεγχο της συμφόρησης στο σύστημα. Η παράμετρος αυτή μπορεί να λάβει δύο τιμές, δύο αλγόριθμους δηλαδή για τον έλεγχο της συμφόρησης στο σύστημα.

Ο πρώτος αλγόριθμος είναι μια προσέγγιση του αλγόριθμου *TCP Vegas* [18], όπου η συνολική καθυστέρηση υπολογίζεται για κάθε πακέτο που μεταδίδεται. Επιπλέον, ο αλγόριθμος αυτός χρησιμοποιεί προσθετική αύξηση και πολλαπλασιαστική μείωση του παραθύρου συμφόρησης (*AIMD of congestion window*) με την εφαρμογή της τεχνικής *slow start*.

Ο δεύτερος αλγόριθμος, ο οποίος και χρησιμοποιήθηκε στα πλαίσια των πειραμάτων μας, ονομάζεται *Remote Adaptive Active Queue Management* (RAAQM) [19] και εκτελεί από την πλευρά του καταναλωτή έναν έλεγχο ανά διαδρομή των ουρών συμφόρησης κατά μήκος της διαδρομής. Ο αλγόριθμος χρησιμοποιεί επίσης την τεχνική *AIMD* με *slow start* για την αποστολή πακέτων ενδιαφέροντος από τον καταναλωτή στον παραγωγό και υπολογίζει τη συνολική μέγιστη και ελάχιστη καθυστέρηση ανά συγκεκριμένο όγκο πακέτων (στην περίπτωση μας ανά 200 πακέτα).

Παρακάτω παρουσιάζεται μια εικόνα με διάφορες παραμέτρους που απαιτούνται για την εφαρμογή των αλγορίθμων ελέγχου συμφόρησης. Πρόκειται για συντελεστές απόρριψης, το χρόνο ζωής πακέτων, τον αριθμό επαναμεταδόσεων, τις καθυστερήσεις μέσω τόσο για μετάδοση με καλωδίωση όσο και για ασύρματη μετάδοση.

```
; this file contains the parameters for RAAQM
autotune = no
lifetime = 500
retransmissions = 128
beta = 0.99
drop = 0.003
beta_wifi_ = 0.99
drop_wifi_ = 0.6
beta_lte_ = 0.99
drop_lte_ = 0.003
wifi_delay_ = 200
lte_delay_ = 9000

alpha = 0.95
batching_parameter = 200

;Choice of rate estimator:
;0 --> an estimation each $(batching_parameter) packets
;1 --> an estimation "a la TCP", estimation at the end of the download of the segment
rate_estimator = 0
```

Εικόνα 34: Λίστα παραμέτρων αλγορίθμου ελέγχου συμφόρησης

Κεφάλαιο 5: Αποτελέσματα

Στο κεφάλαιο αυτό παρουσιάζονται σε μορφή διαγραμμάτων τα αποτελέσματα που εξήχθησαν από τα πειράματα μας για κάθε ένα από τα προαναφερθέντα σενάρια. Επιπλέον, γίνεται μια σύγκριση των αποτελεσμάτων που επιφέρουν κάθε ένας από τους τρόπους σύνδεσης των κόμβων.

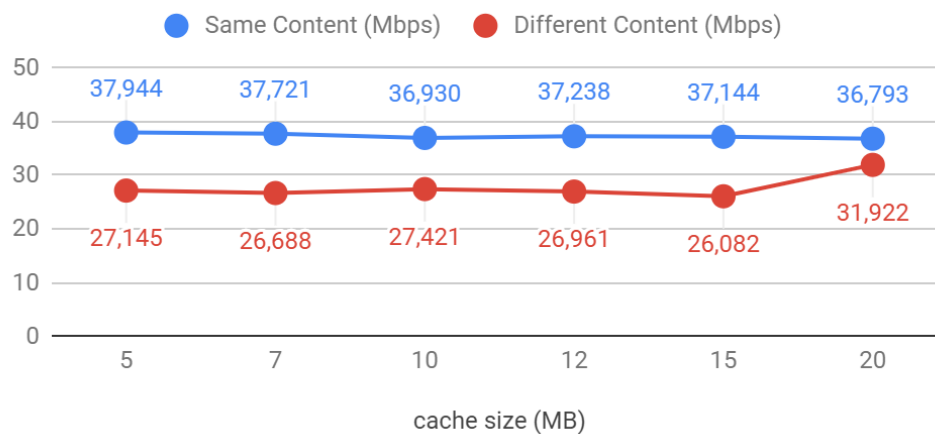
5.1 Αποτελέσματα Σεναρίου 1^{ου}

Για ανάλυση Σεναρίου 1 βλέπε ενότητα 4.3.

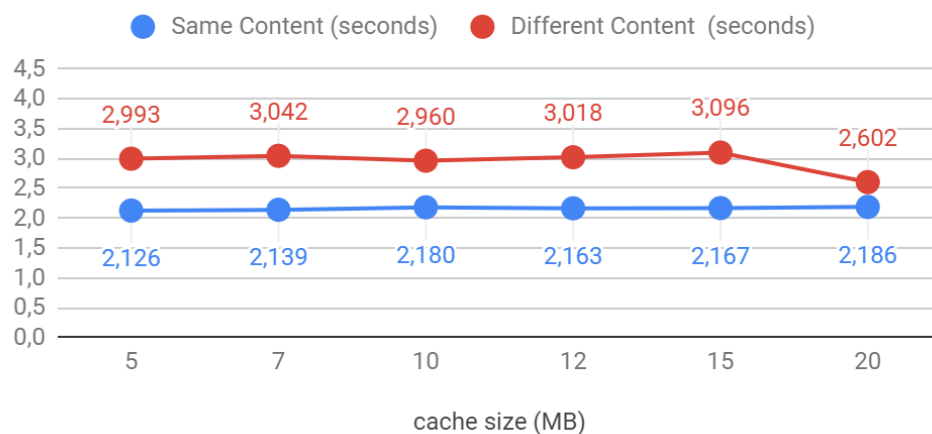
5.1.1 Ενσύρματη σύνδεση

➤ Αίτημα για περιεχόμενο στην τοπολογία Dumbbell.

Throughput (Dumbbell)

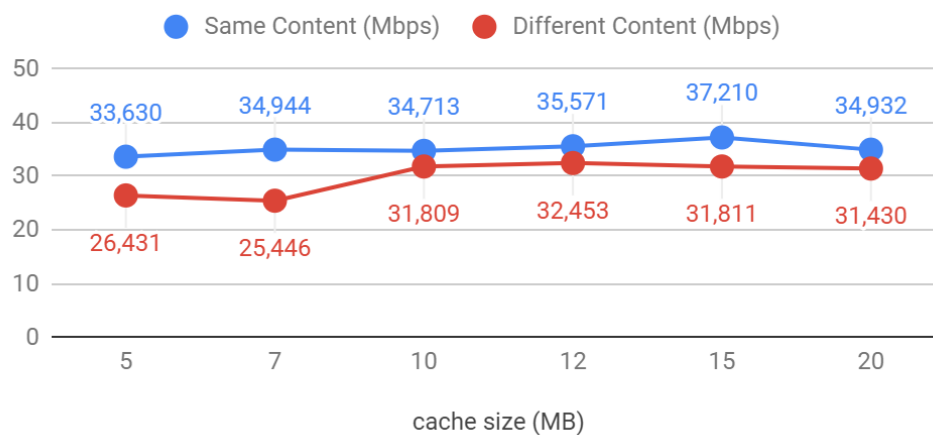


Delay (Dumbbell)

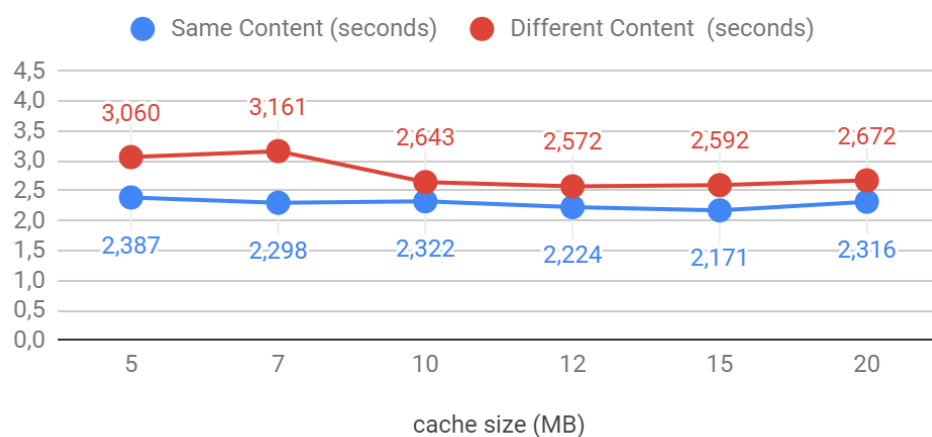


- Αίτημα για περιεχόμενο στην τοπολογία Dumbbell πολλαπλών διαδρομών.

Throughput (Multiple Path Dumbbell)

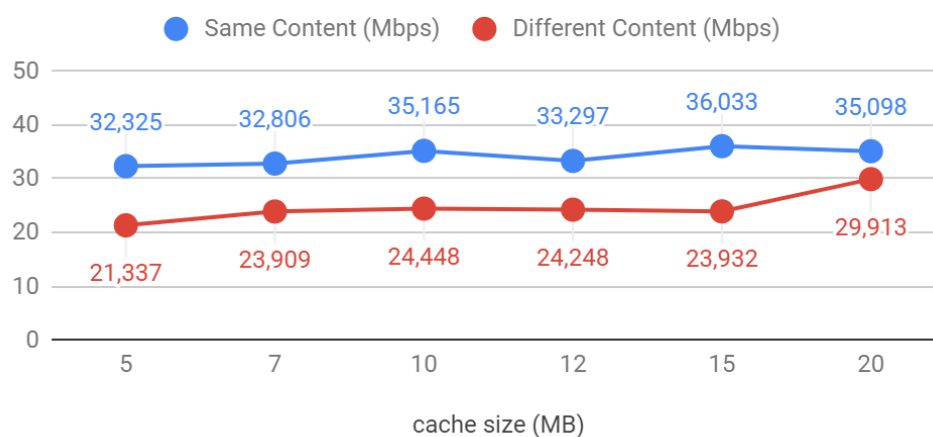


Delay (Multiple Path Dumbbell)

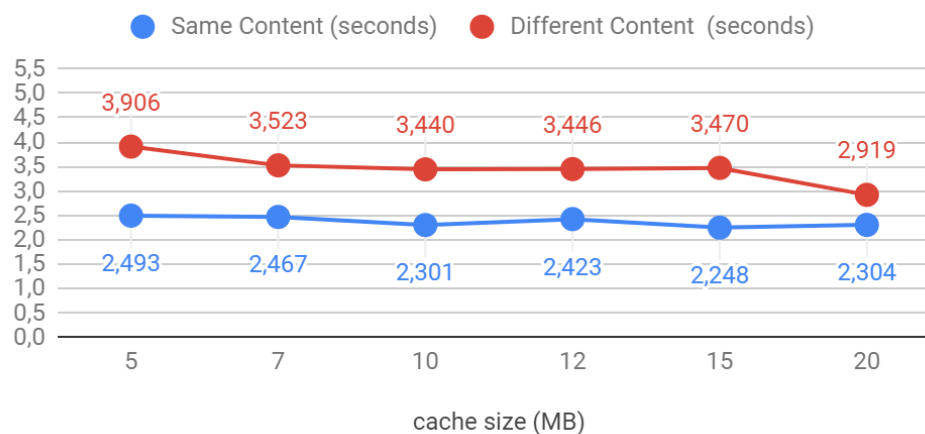


- Αίτημα για περιεχόμενο στην τοπολογία Tree.

Throughput (Tree)

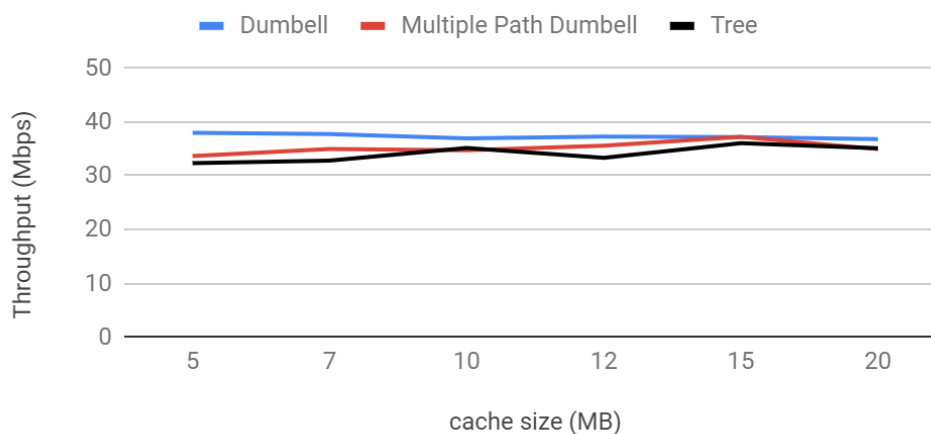


Delay (Tree)

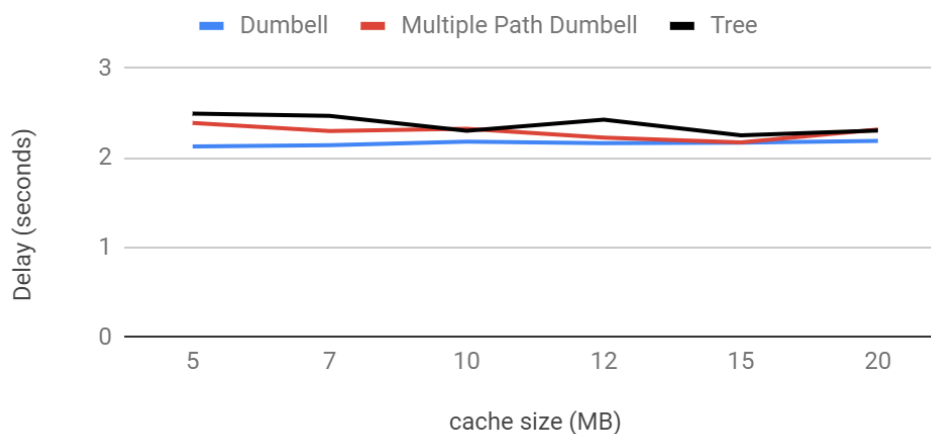


➤ Σύγκριση απόδοσης τοπολογιών όταν ζητείται το ίδιο περιεχόμενο.

Throughput (Wired)

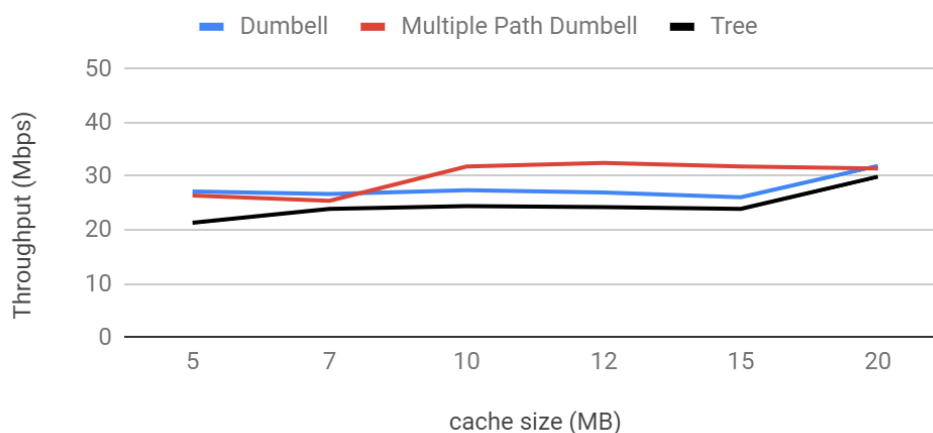


Delay (Wired)

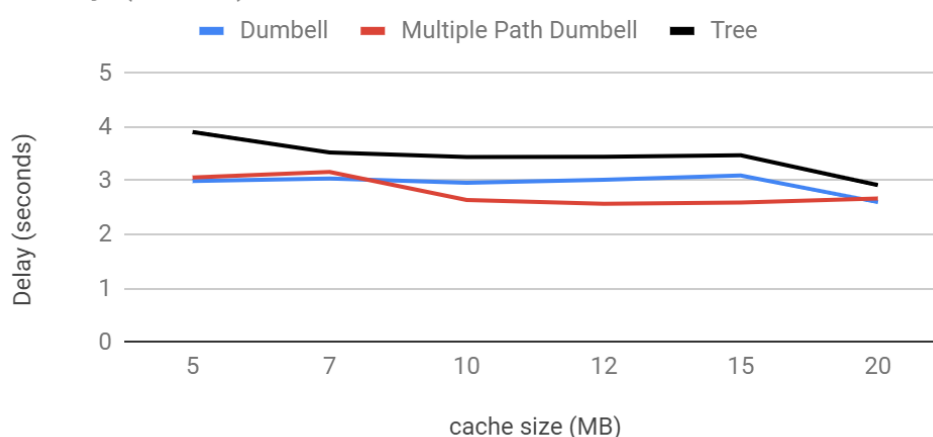


- Σύγκριση απόδοσης τοπολογιών όταν ζητείται διαφορετικό περιεχόμενο.

Throughput (Wired)



Delay (Wired)



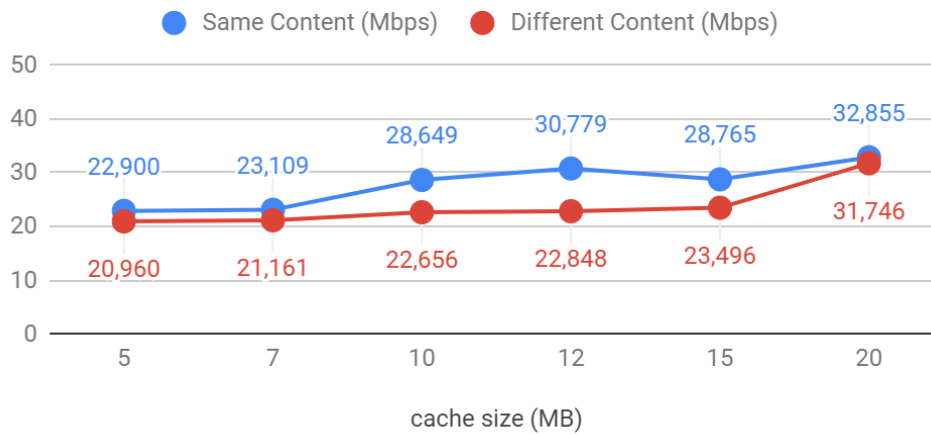
Στο σενάριο 1, όταν η σύνδεση είναι ενσύρματη παρατηρούμε σταθερά καλύτερη απόδοση σε όλες τις τοπολογίες όταν ζητείται το ίδιο περιεχόμενο. Αυτό συμβαίνει γιατί όταν το μέγεθος της κρυφής μνήμης είναι μεγαλύτερο ή ίσο του μεγέθους του περιεχομένου (10MB), τα αιτήματα για ίδιο περιεχόμενο εξυπηρετούνται από την κρυφή μνήμη. Οι τρεις διαφορετικές τοπολογίες πετυχαίνουν αρκετά κοντινές αποδόσεις.

Όταν ζητείται διαφορετικό περιεχόμενο, παρατηρούμε ότι η απόδοση αυξάνεται όταν το μέγεθος της κρυφής μνήμης είναι 20MB και αυτό γιατί μόνο τότε η κρυφή μνήμη μπορεί να αποθηκεύει και να εξυπηρετεί τα δύο διαφορετικά περιεχόμενα των 10MB τοπικά. Σε αυτή την περίπτωση, η τοπολογία Dumbbell πολλαπλών διαδρομών φαίνεται να έχει την καλύτερη απόδοση.

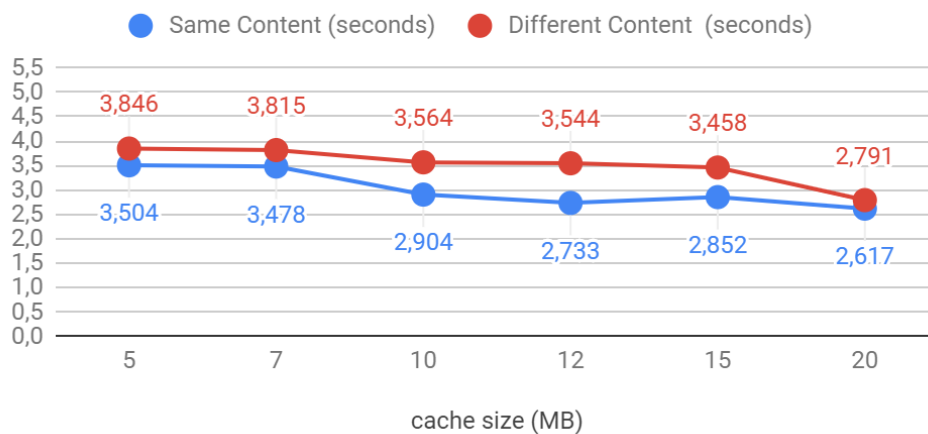
5.1.2 Ασύρματη σύνδεση μέσω WiFi

➤ Αίτημα για περιεχόμενο στην τοπολογία Dumbbell.

Throughput (Dumbbell)

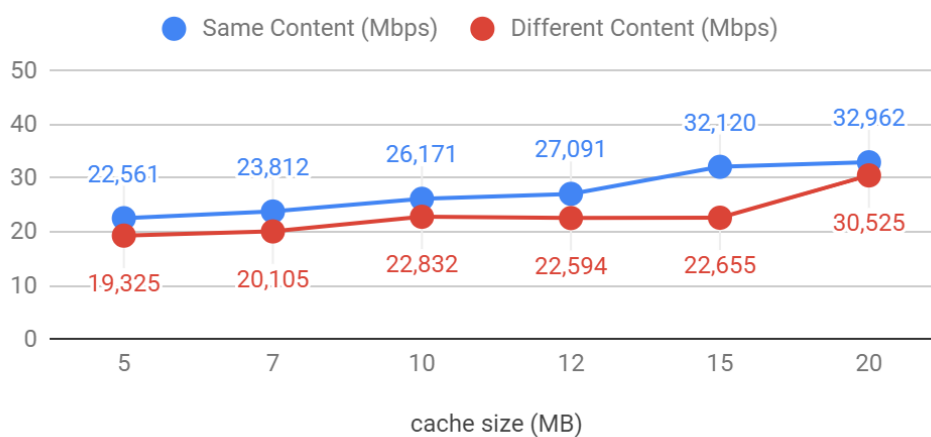


Delay (Dumbbell)

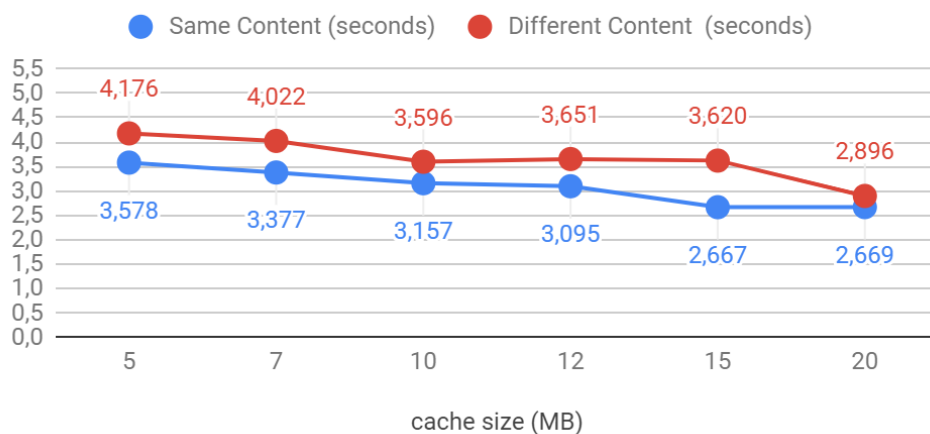


➤ Αίτημα για περιεχόμενο στην τοπολογία Dumbbell πολλαπλών διαδρομών.

Throughput (Multiple Path Dumbbell)

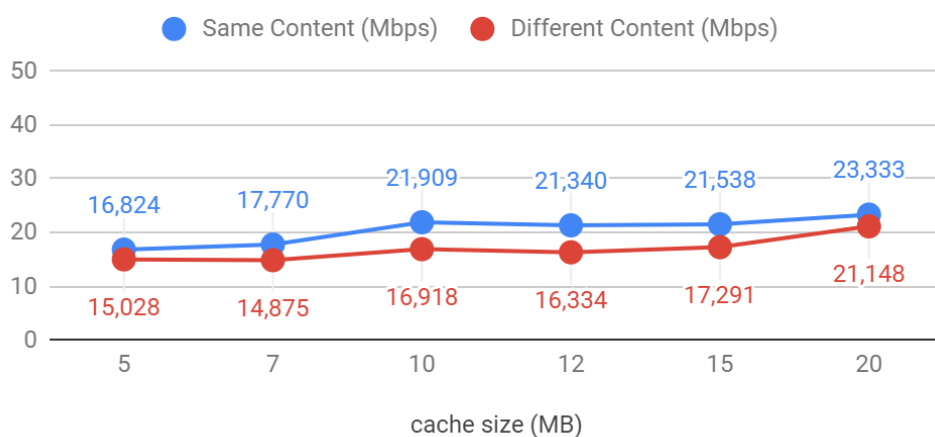


Delay (Multiple Path Dumbbell)

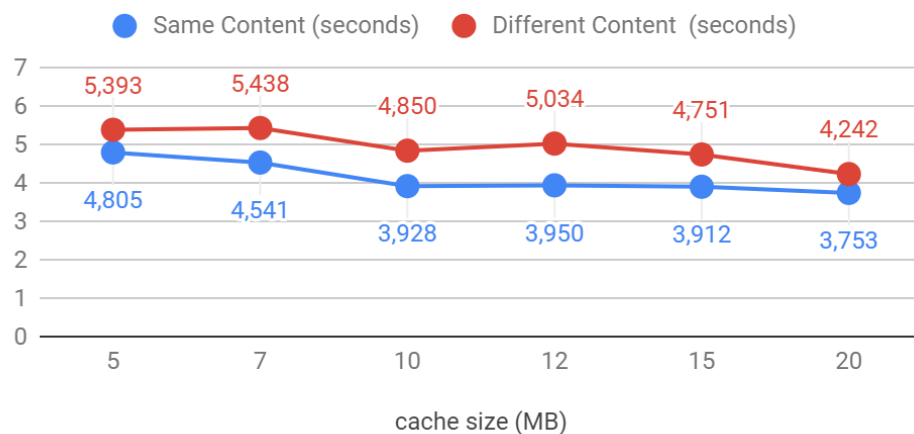


➤ Αίτημα για περιεχόμενο στην τοπολογία Tree.

Throughput (Tree)

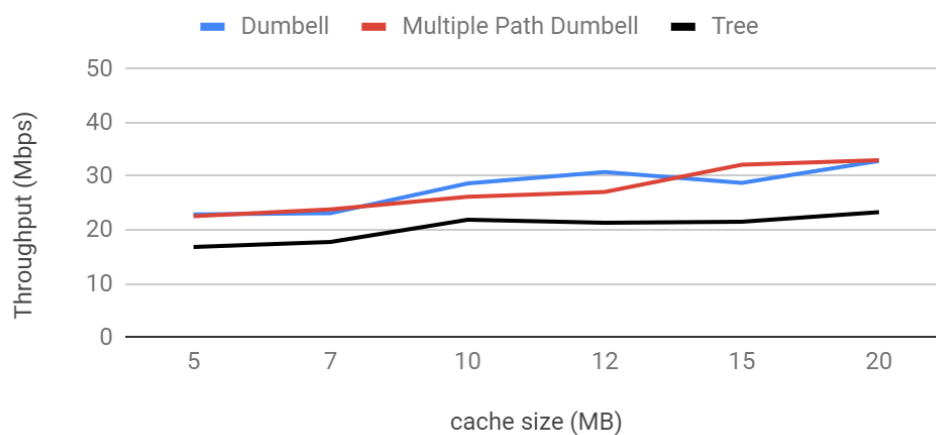


Delay (Tree)

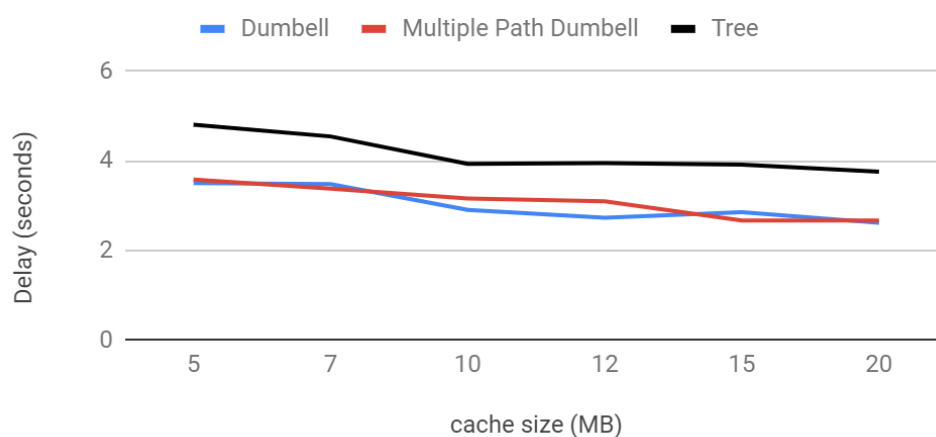


- Σύγκριση απόδοσης τοπολογιών όταν ζητείται το ίδιο περιεχόμενο.

Throughput (WiFi)

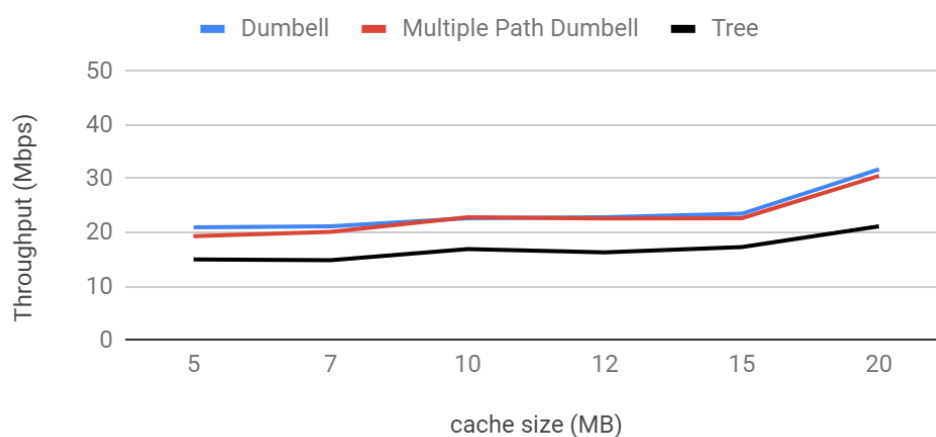


Delay (WiFi)

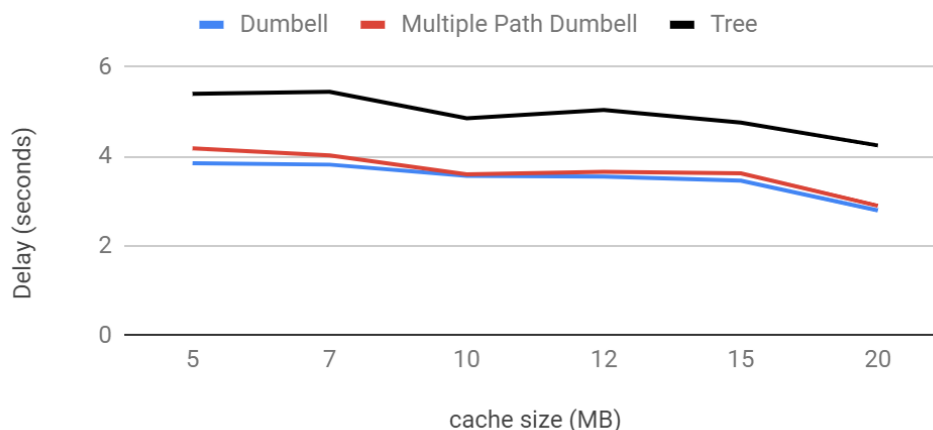


- Σύγκριση απόδοσης τοπολογιών όταν ζητείται διαφορετικό περιεχόμενο.

Throughput (WiFi)



Delay (WiFi)



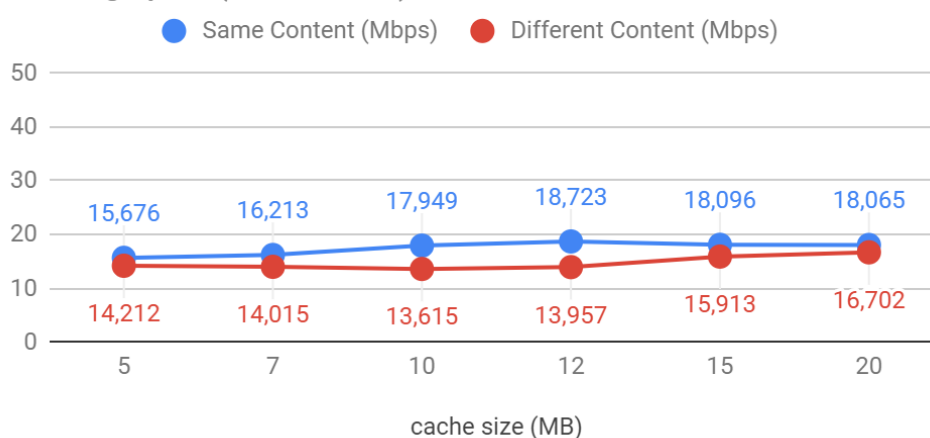
Στην περίπτωση της ασύρματης σύνδεσης μέσω WiFi στο σενάριο 1, η απόδοση του συστήματος όταν ζητείται το ίδιο περιεχόμενο είναι και πάλι καλύτερη σε όλες τις τοπολογίες. Επιπλέον, έχουμε παρόμοια συμπεριφορά με την ενσύρματη σύνδεση λόγω της κρυφής μνήμης.

Όταν ζητείται διαφορετικό περιεχόμενο, βλέπουμε αύξηση της απόδοσης όταν η κρυφή μνήμη γίνεται 20MB ακριβώς όπως και στην ενσύρματη σύνδεση. Οι τοπολογίες Dumbbell και Dumbbell πολλαπλών διαδρομών φαίνεται να έχουν σχεδόν ίδια απόδοση.

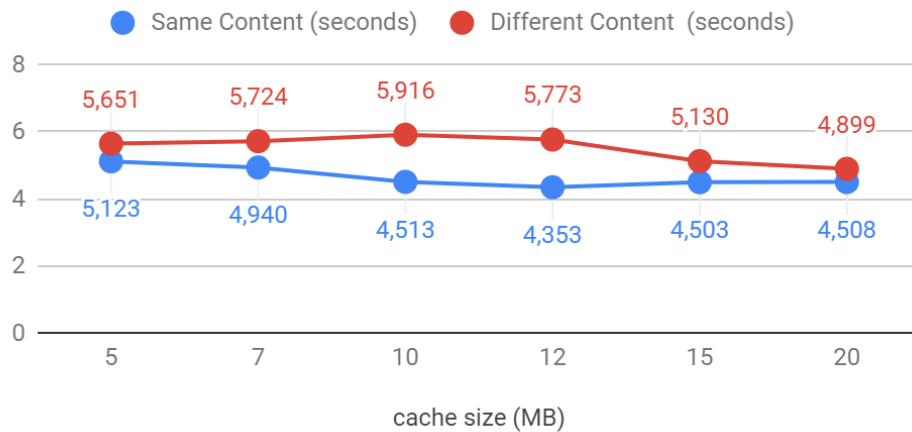
5.1.3 Ασύρματη σύνδεση μέσω LTE

➤ Αίτημα για περιεχόμενο στην τοπολογία Dumbbell.

Throughput (Dumbbell)

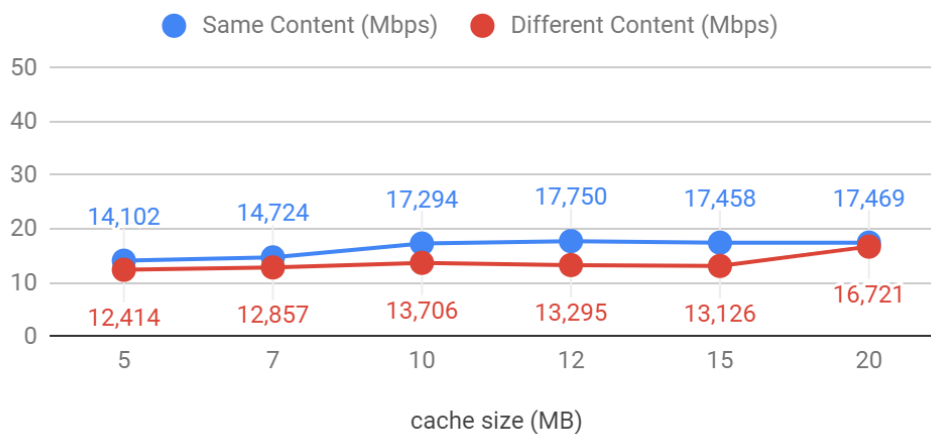


Delay (Dumbbell)

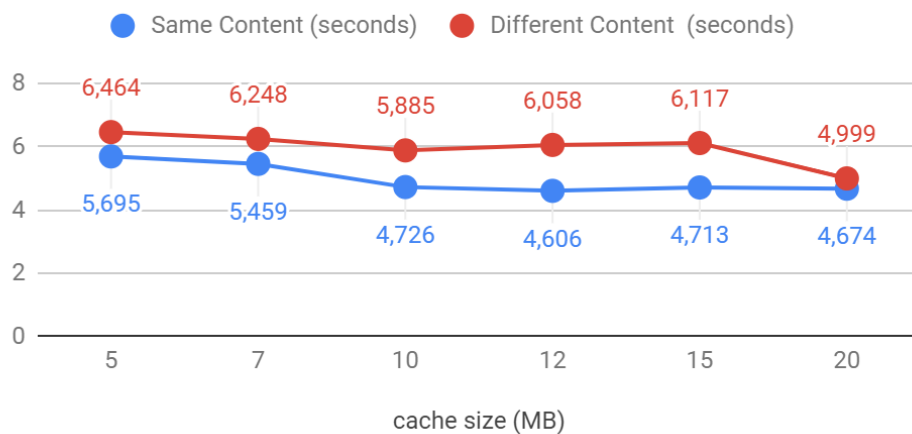


➤ Αίτημα για περιεχόμενο στην τοπολογία Dumbbell πολλαπλών διαδρομών.

Throughput (Multiple Path Dumbbell)

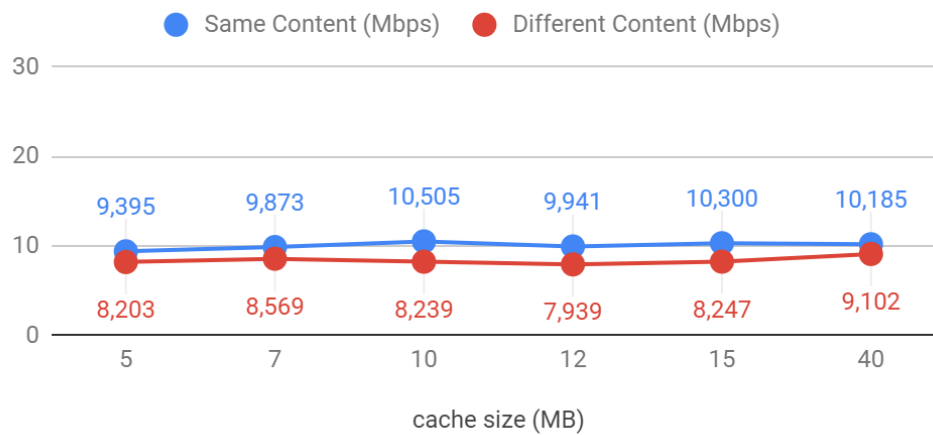


Delay (Multiple Path Dumbbell)

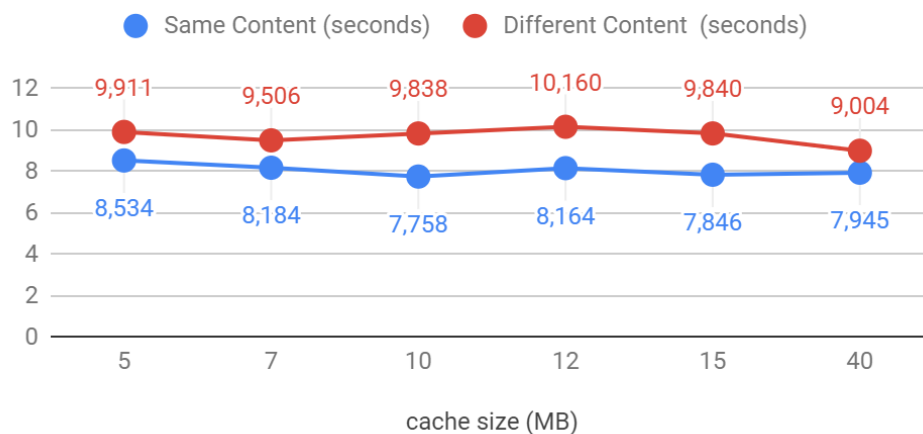


- Αίτημα για περιεχόμενο στην τοπολογία Tree.

Throughput (Tree)

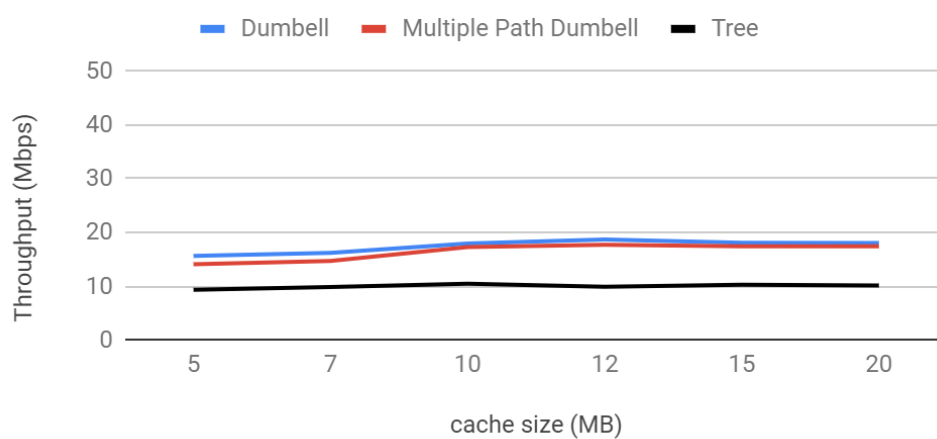


Delay (Tree)

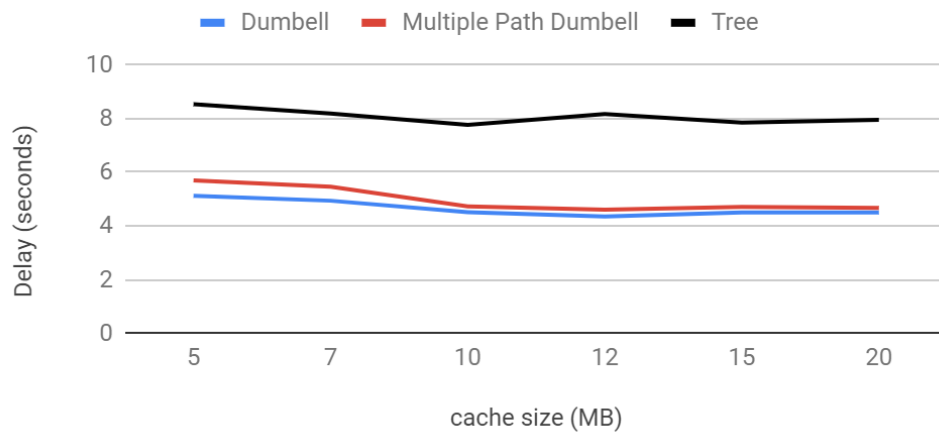


- Σύγκριση απόδοσης τοπολογιών όταν ζητείται το ίδιο περιεχόμενο.

Throughput (LTE)

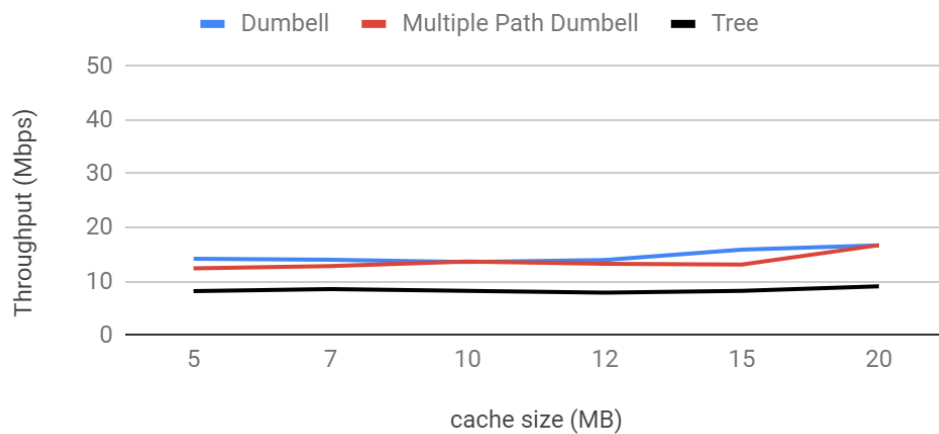


Delay (LTE)

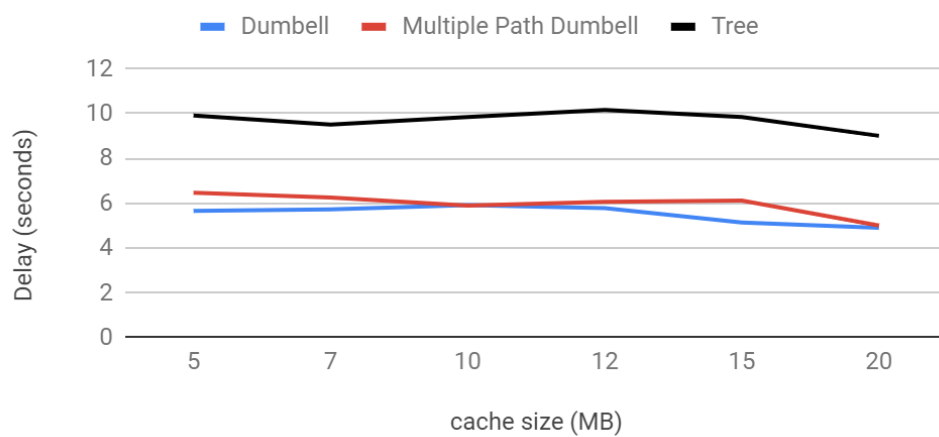


➤ Σύγκριση απόδοσης τοπολογιών όταν ζητείται διαφορετικό περιεχόμενο.

Throughput (LTE)



Delay (LTE)



Κατά την ασύρματη σύνδεση μέσω LTE στο σενάριο 1, παρατηρούμε την ίδια συμπεριφορά με τις υπόλοιπες συνδέσεις του σεναρίου ακριβώς για τους λόγους που εξηγήσαμε παραπάνω. Ωστόσο, μια σημαντική παρατήρηση είναι ότι οι αποδόσεις με την εφαρμογή αυτού του τρόπου σύνδεσης κυμαίνονται συνολικά σε χαμηλότερες τιμές για το Throughput.

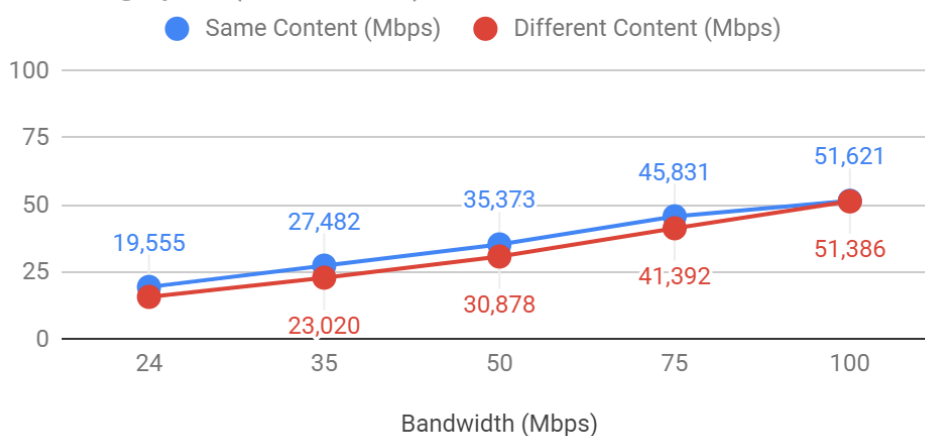
5.2 Αποτελέσματα Σεναρίου 2^{ου}

Για ανάλυση Σεναρίου 2 βλέπε ενότητα 4.3.

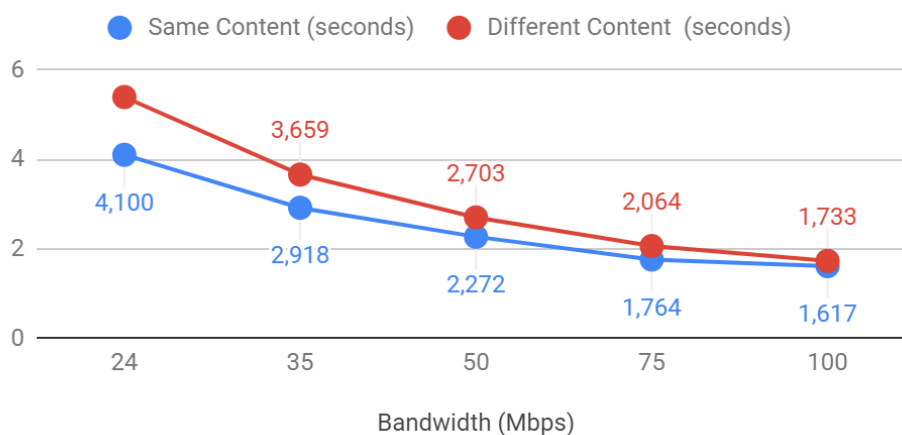
5.2.1 Ενσύρματη σύνδεση

➤ Αίτημα για περιεχόμενο στην τοπολογία Dumbbell.

Throughput (Dumbbell)

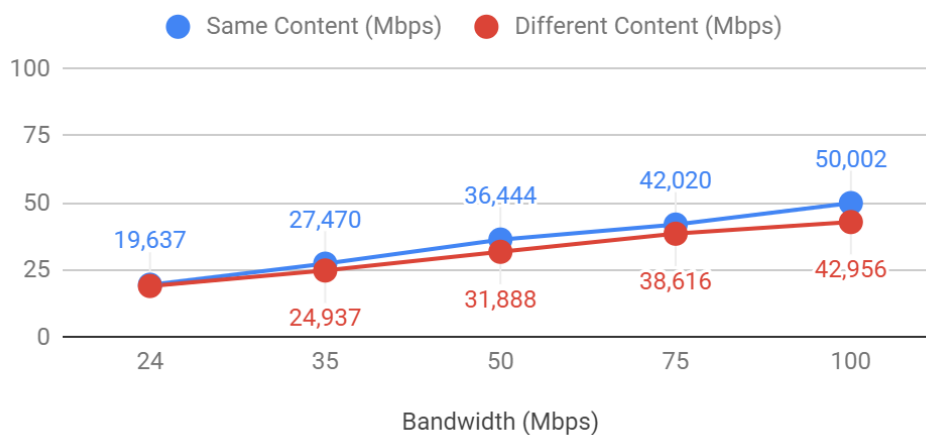


Delay (Dumbbell)

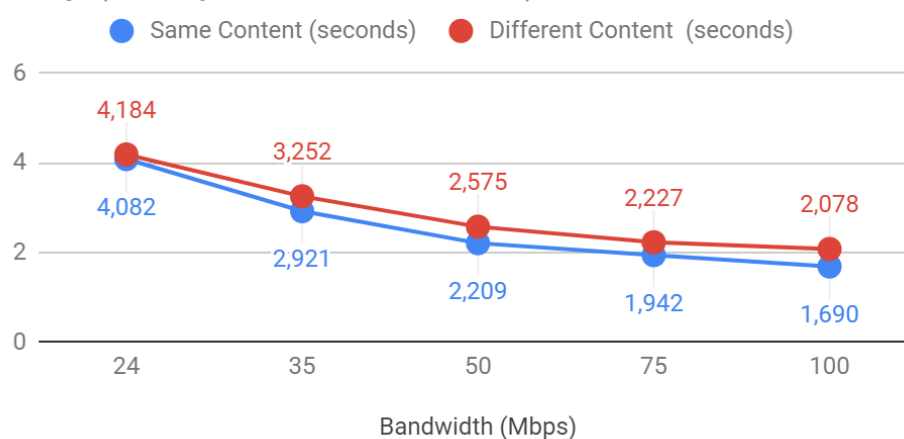


- Αίτημα για περιεχόμενο στην τοπολογία Dumbbell πολλαπλών διαδρομών.

Throughput (Multiple Path Dumbbell)

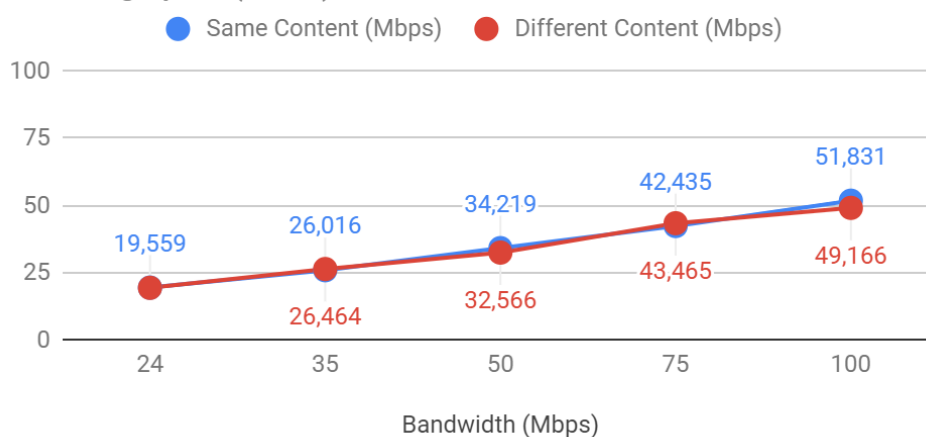


Delay (Multiple Path Dumbbell)

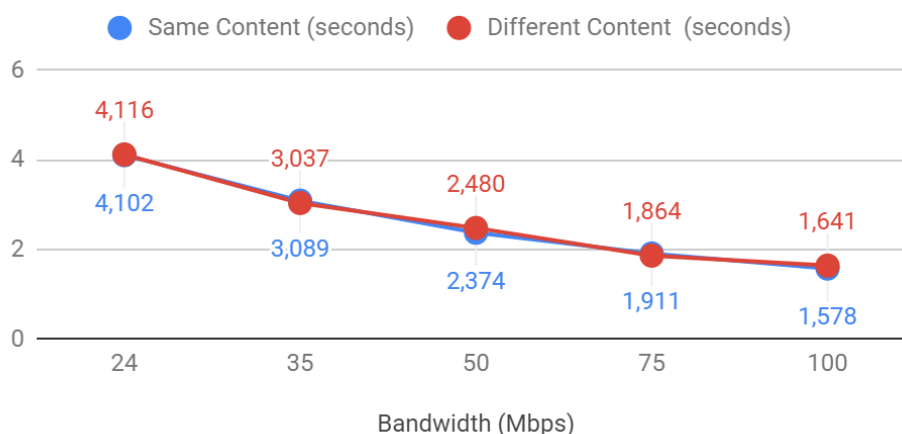


- Αίτημα για περιεχόμενο στην τοπολογία Tree.

Throughput (Tree)



Delay (Tree)



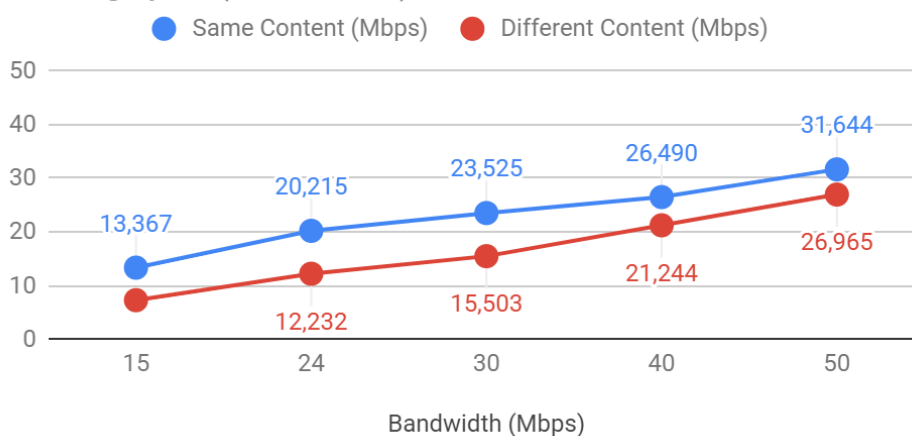
Για το σενάριο 2, όταν ο τρόπος σύνδεσης είναι ενσύρματος παρατηρούμε ότι είτε ζητείται το ίδιο είτε διαφορετικό περιεχόμενο το *Throughput* παραμένει πολύ κοντά και φυσικά αυξάνει όσο αυξάνεται το *Bandwidth*. Το μέγεθος της κρυφής μνήμης είναι 20MB και το μέγεθος του κάθε περιεχομένου είναι 10MB, δηλαδή η κρυφή μνήμη δύναται να αποθηκεύσει και να εξυπηρετεί και τα δύο περιεχόμενα ταυτόχρονα.

Ωστόσο, παρατηρούμε ότι η απόδοση πέφτει όσο αυξάνει το *Bandwidth*. Για παράδειγμα στην τοπολογία Dumbbell όταν το *Bandwidth* είναι 24Mbps, το *Throughput* είναι περίπου 19,5Mbps, δηλαδή η απόδοση είναι 81%. Όταν το *Bandwidth* είναι 100Mbps, το *Throughput* είναι περίπου 51,5Mbps, δηλαδή έχουμε απόδοση 51,5%. Παρόμοια είναι η συμπεριφορά και στις άλλες τοπολογίες.

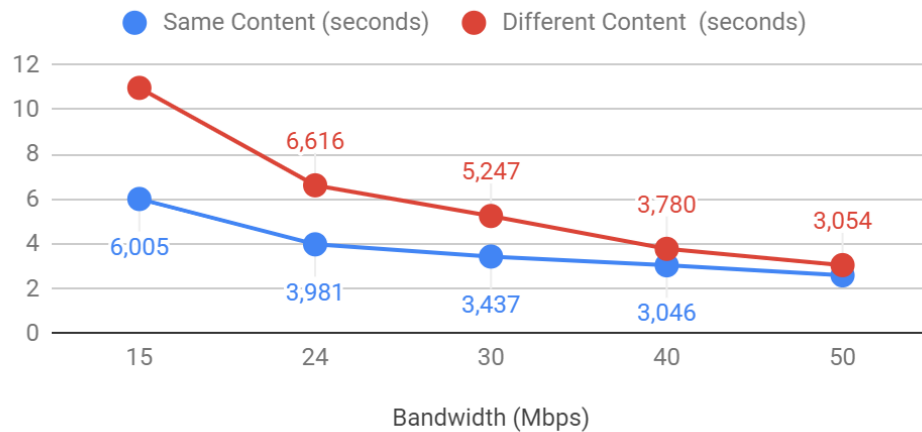
5.2.2 Ασύρματη σύνδεση μέσω WiFi

➤ Αίτημα για περιεχόμενο στην τοπολογία Dumbbell.

Throughput (Dumbbell)

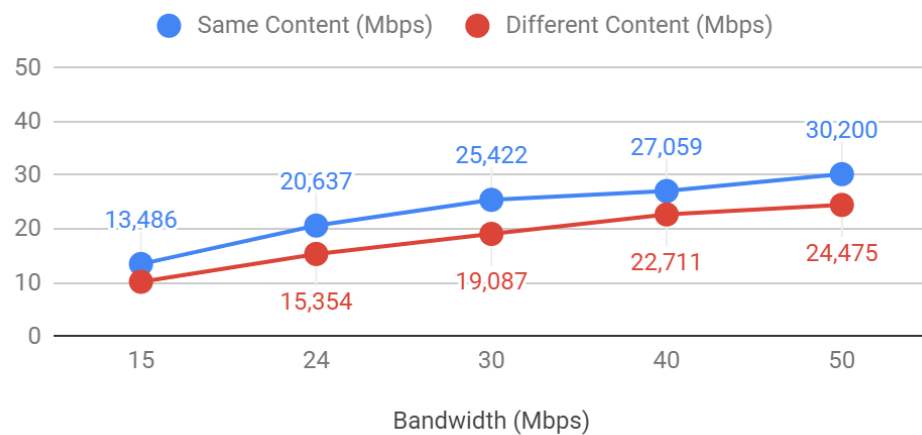


Delay (Dumbbell)

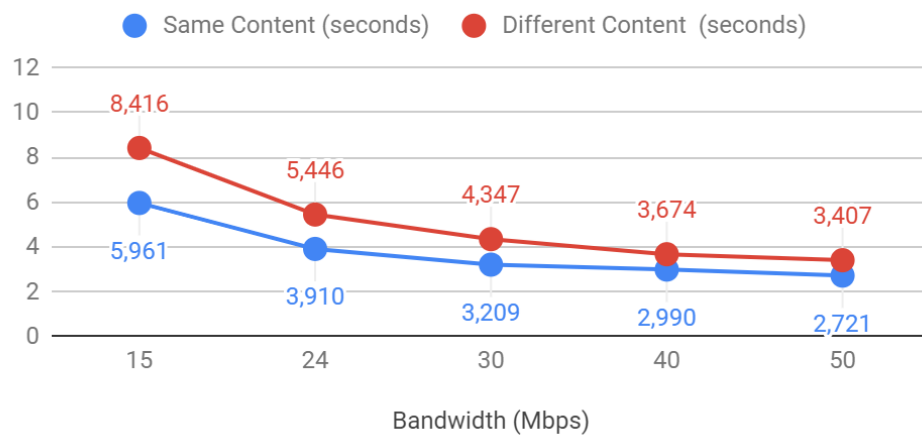


➤ Αίτημα για περιεχόμενο στην τοπολογία Dumbbell πολλαπλών διαδρομών.

Throughput (Multiple Path Dumbbell)

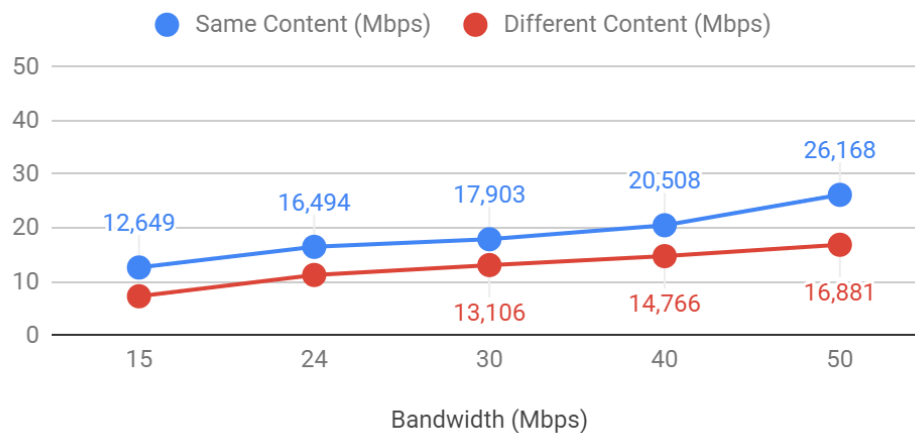


Delay (Multiple Path Dumbbell)

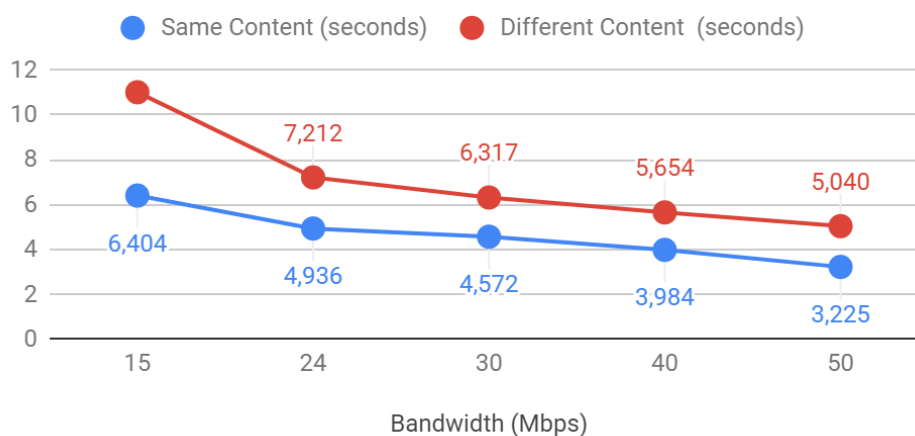


➤ Αίτημα για περιεχόμενο στην τοπολογία Tree.

Throughput (Tree)



Delay (Tree)



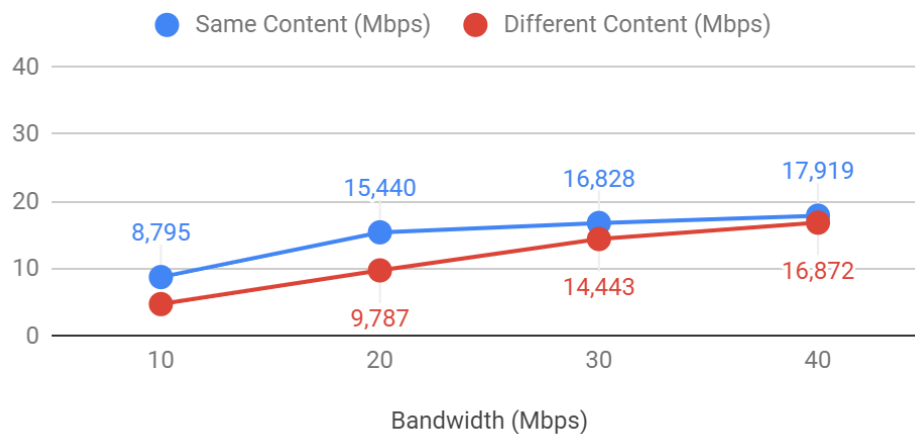
Όταν έχουμε ασύρματη σύνδεση μέσω WiFi, παρατηρούμε ότι το *Throughput* είναι σταθερά καλύτερο όταν ζητείται το ίδιο περιεχόμενο παρόλο που η κρυφή μνήμη μπορεί να αποθηκεύει και τα δύο περιεχόμενα ταυτόχρονα.

Στο θέμα της απόδοσης, παρατηρούμε ότι και πάλι πέφτει όσο αυξάνει το *Bandwidth*. Για παράδειγμα στην τοπολογία Dumbbell όταν το *Bandwidth* είναι 15Mbps, το *Throughput* είναι κατά μέσο όρο 10,5Mbps, δηλαδή η απόδοση είναι 70%. Όταν το *Bandwidth* είναι 50Mbps, το *Throughput* είναι περίπου 29,3Mbps, δηλαδή έχουμε απόδοση 58,6%. Παρόμοια είναι η συμπεριφορά και στις άλλες τοπολογίες.

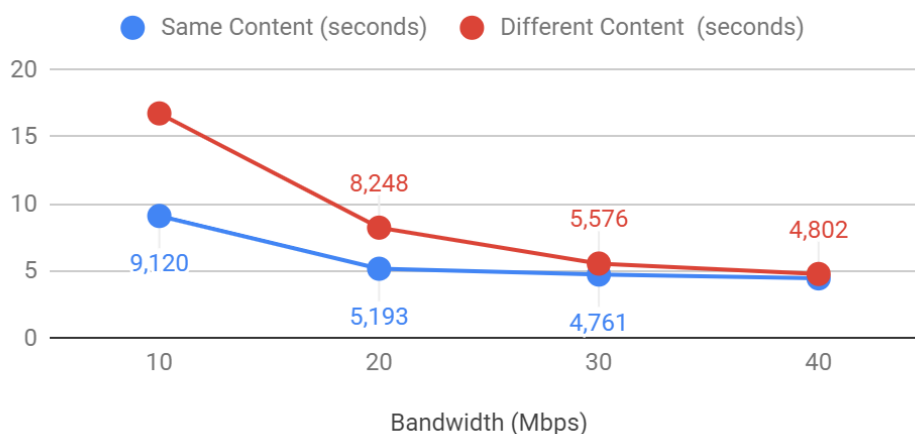
5.2.3 Ασύρματη σύνδεση μέσω LTE

➤ Αίτημα για περιεχόμενο στην τοπολογία Dumbbell.

Throughput (Dumbbell)

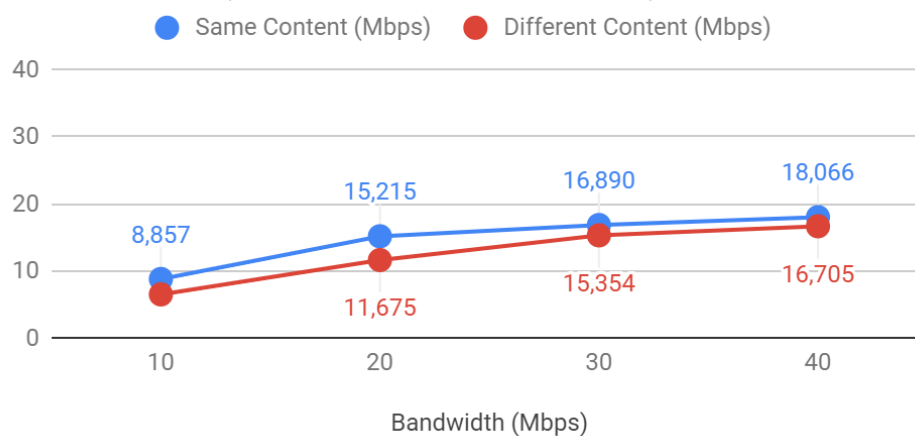


Delay (Dumbbell)

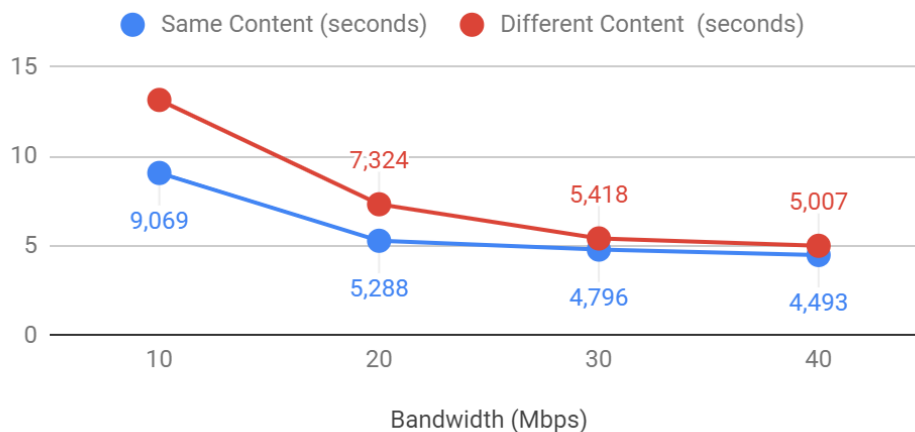


➤ Αίτημα για περιεχόμενο στην τοπολογία Dumbbell πολλαπλών διαδρομών.

Throughput (Multiple Path Dumbbell)

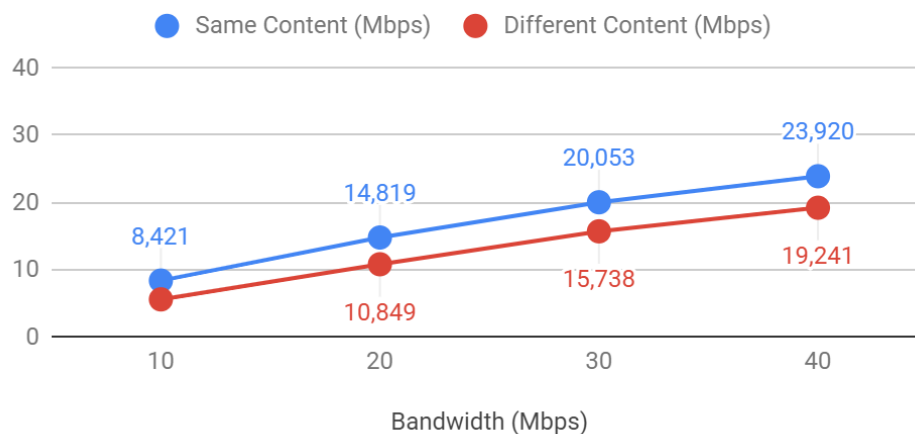


Delay (Multiple Path Dumbbell)

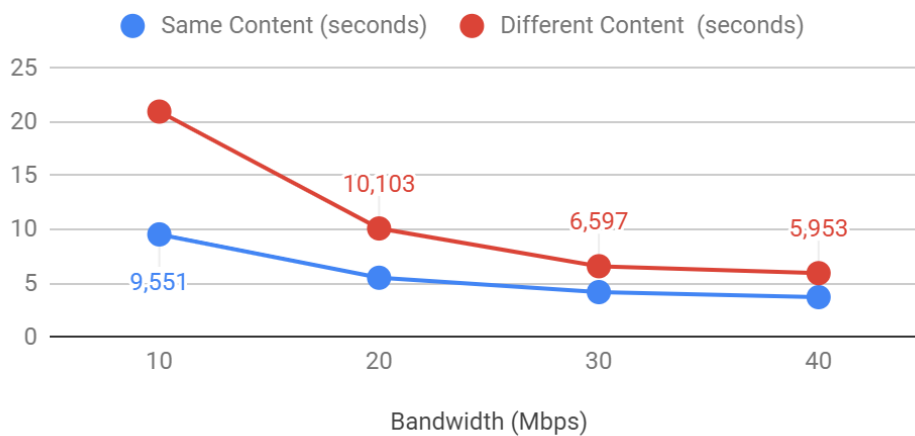


➤ Αίτημα για περιεχόμενο στην τοπολογία Tree.

Throughput (Tree)



Delay (Tree)



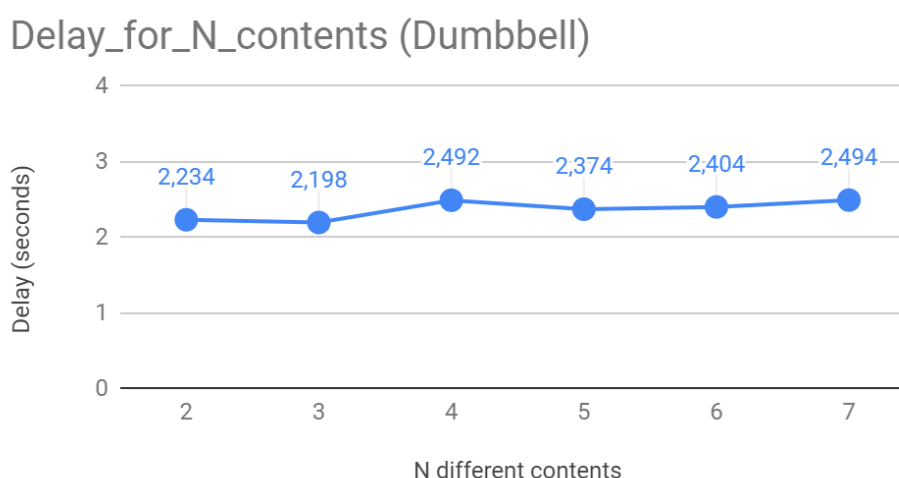
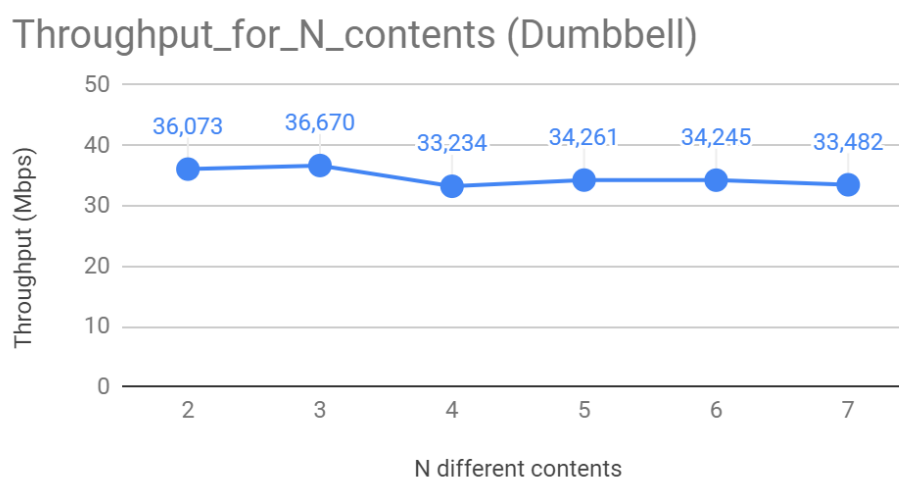
Όταν έχουμε ασύρματη σύνδεση μέσω LTE, παρατηρούμε ότι τόσο η συμπεριφορά του *Throughput* όσο και η συμπεριφορά της απόδοσης μοιάζει σε τεράστιο βαθμό με την περίπτωση που ο τρόπος σύνδεσης είναι ασύρματος μέσω WiFi. Η μόνη διαφορά που πρέπει να αναφερθεί σε αυτό το σημείο είναι ότι το *Bandwidth* και κατ'επέκταση το *Throughput* συνολικά μένει σε χαμηλότερες τιμές.

5.3 Αποτελέσματα Σεναρίου 3^{ου}

Για ανάλυση Σεναρίου 3 βλέπε ενότητα 4.3.

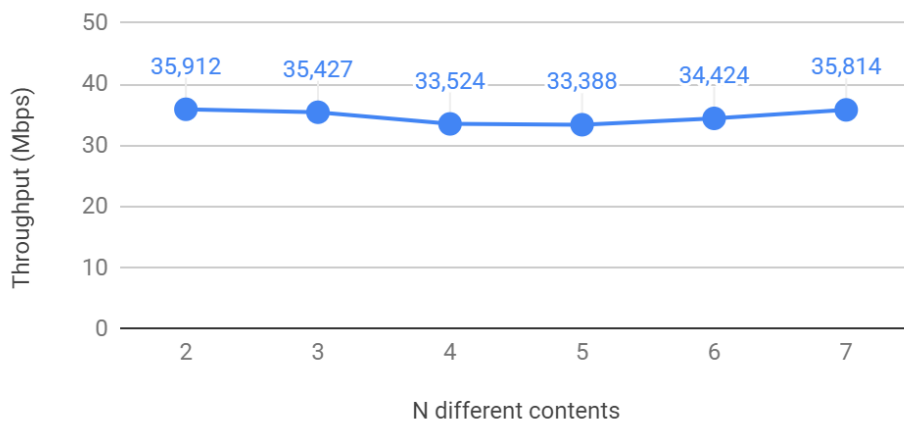
5.3.1 Ενσύρματη σύνδεση

➤ Αιτήματα για περιεχόμενο στην τοπολογία Dumbbell.

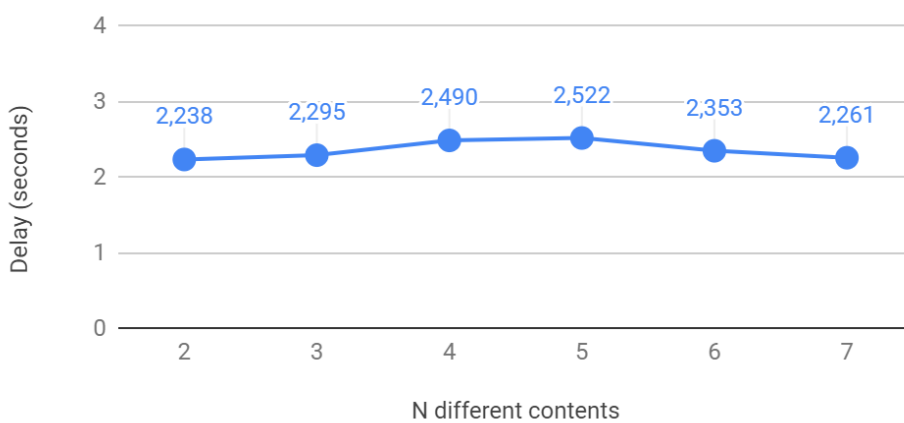


➤ Αιτήματα για περιεχόμενο στην τοπολογία Dumbbell πολλαπλών διαδρομών.

Throughput_for_N_contents (Multiple Path Dumbbell)

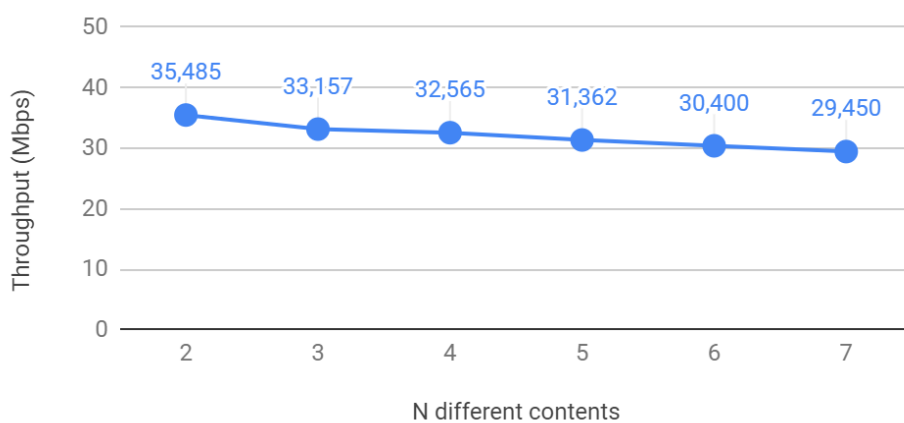


Delay_for_N_contents (Multiple Path Dumbbell)

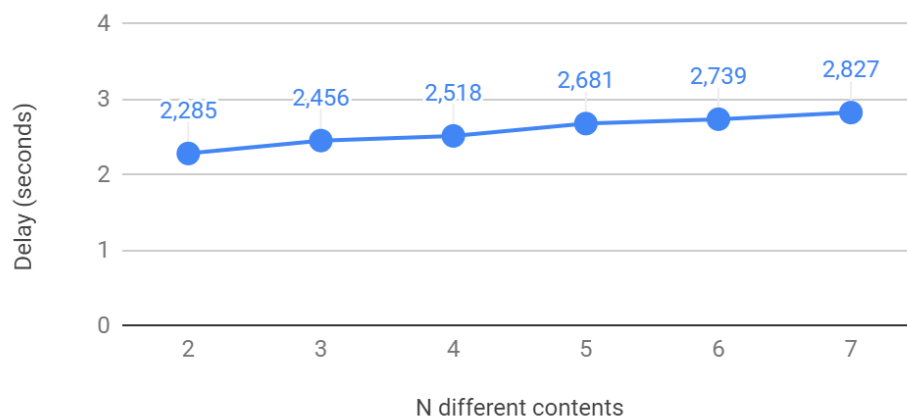


➤ Αιτήματα για περιεχόμενο στην τοπολογία Tree.

Throughput_for_N_contents (Tree)

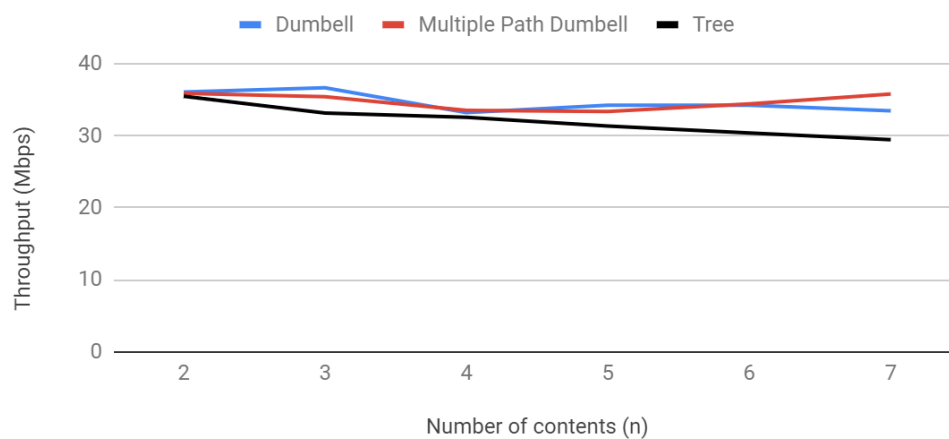


Delay_for_N_contents (Tree)

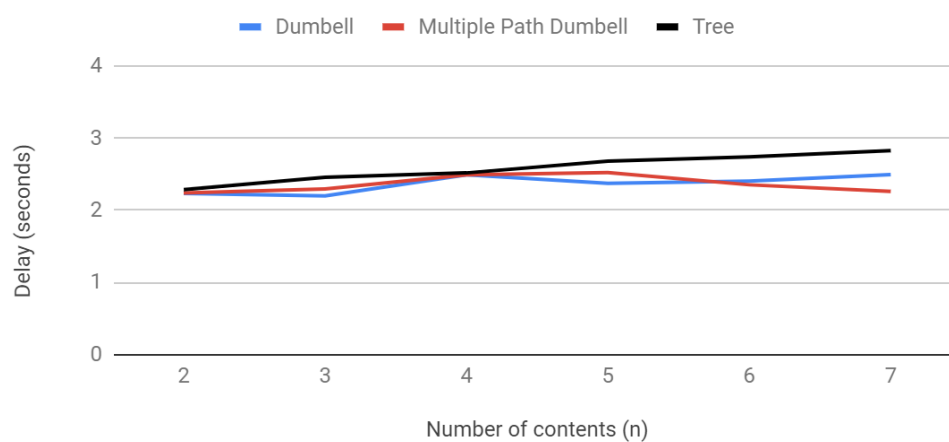


➤ Σύγκριση απόδοσης τοπολογιών.

Throughput (Wired)



Delay (Wired)

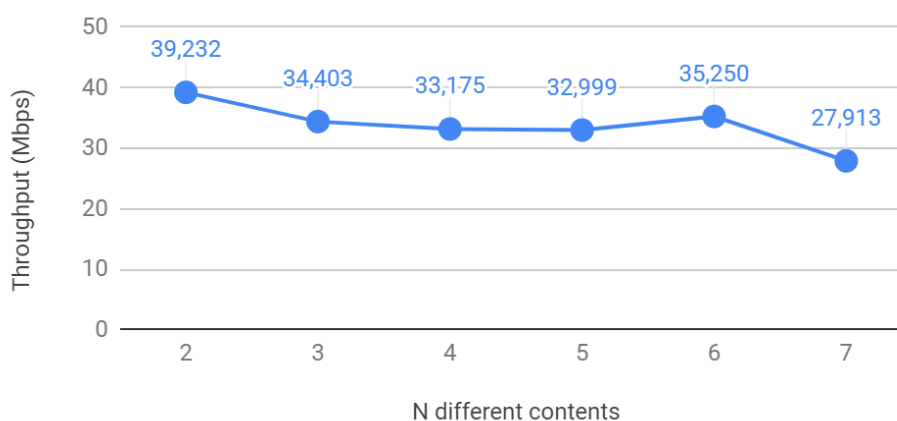


Στο σενάριο 3, όταν έχουμε ενσύρματη σύνδεση παρατηρούμε ότι οι τοπολογίες Dumbbell και Dumbbell πολλαπλών διαδρομών παρουσιάζουν σταθερή συμπεριφορά στο *Throughput* ανεξάρτητα του αριθμού περιεχομένων που υπάρχουν διαθέσιμα στον παραγωγό και ζητούνται από τον καταναλωτή. Στην τοπολογία Tree, βλέπουμε ότι το *Throughput* παρουσιάζει πτωτική τάση όσο αυξάνεται ο αριθμός των διαθέσιμων περιεχομένων.

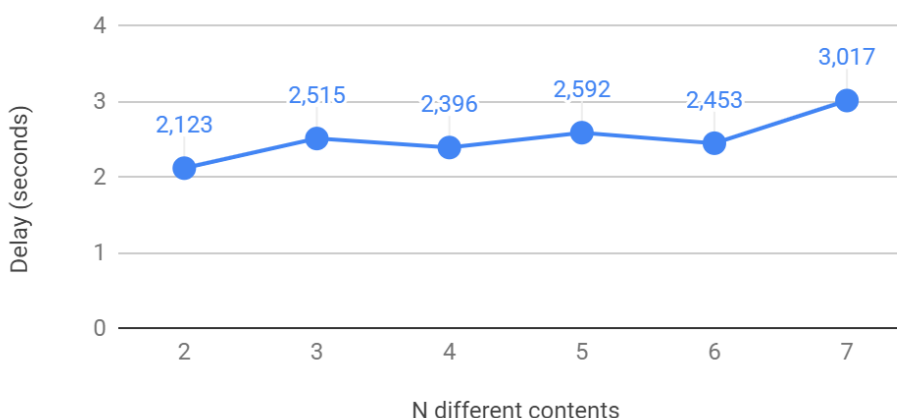
5.3.2 Ασύρματη σύνδεση μέσω WiFi

➤ Αιτήματα για περιεχόμενο στην τοπολογία Dumbbell.

Throughput_for_N_contents (Dumbbell)

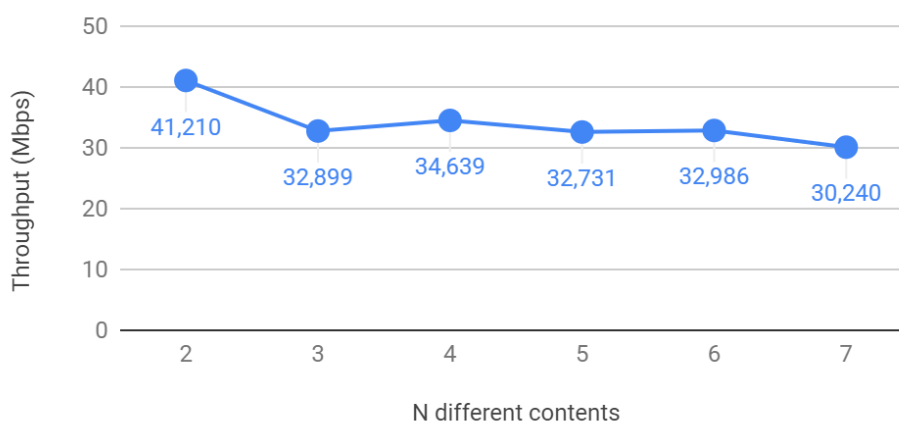


Delay_for_N_contents (Dumbbell)

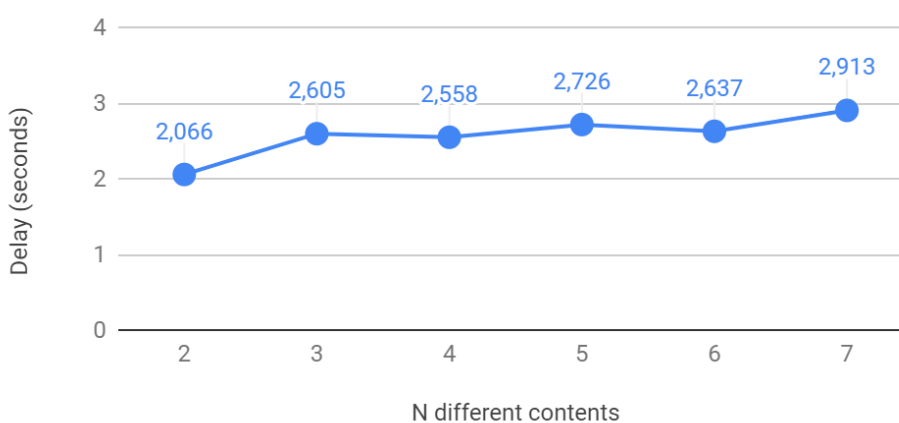


➤ Αιτήματα για περιεχόμενο στην τοπολογία Dumbbell πολλαπλών διαδρομών.

Throughput_for_N_contents (Multiple Path Dumbbell)

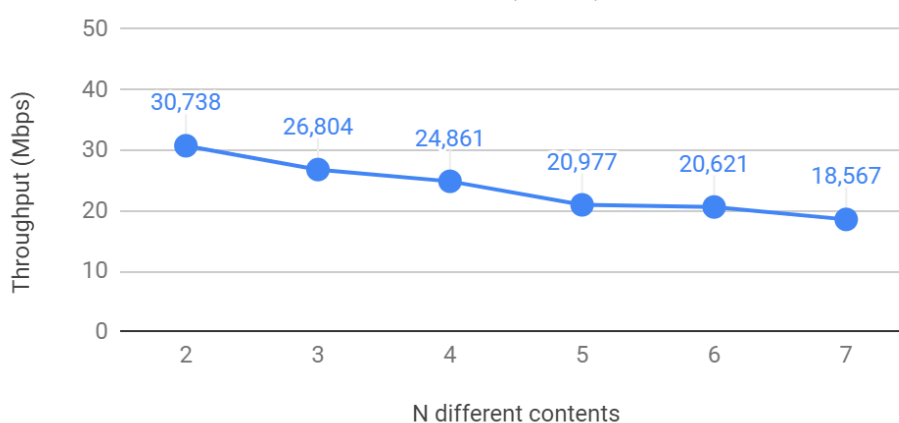


Delay_for_N_contents (Multiple Path Dumbbell)

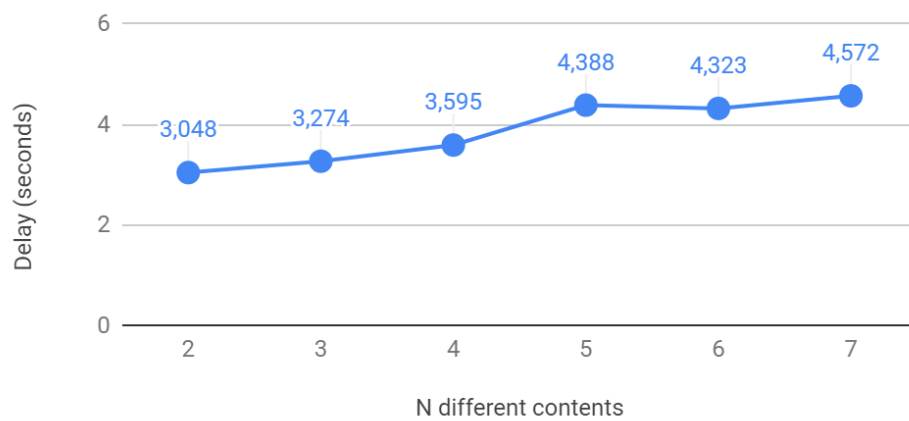


➤ Αιτήματα για περιεχόμενο στην τοπολογία Tree.

Throughput_for_N_contents (Tree)

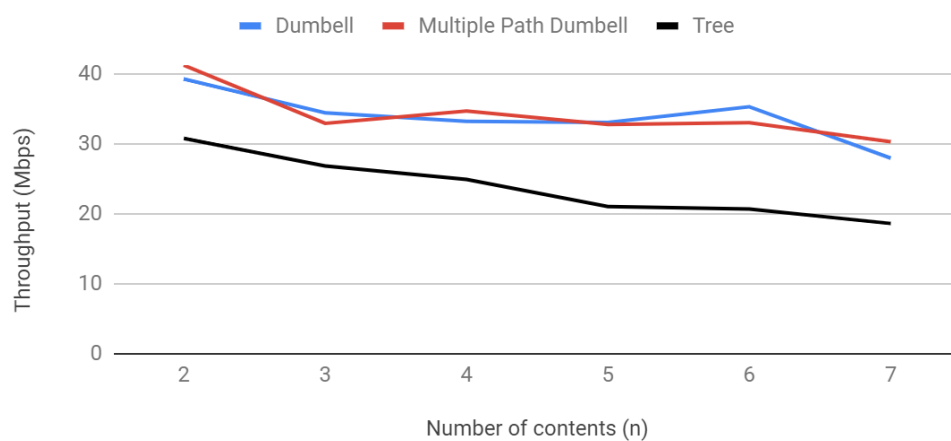


Delay_for_N_contents (Tree)

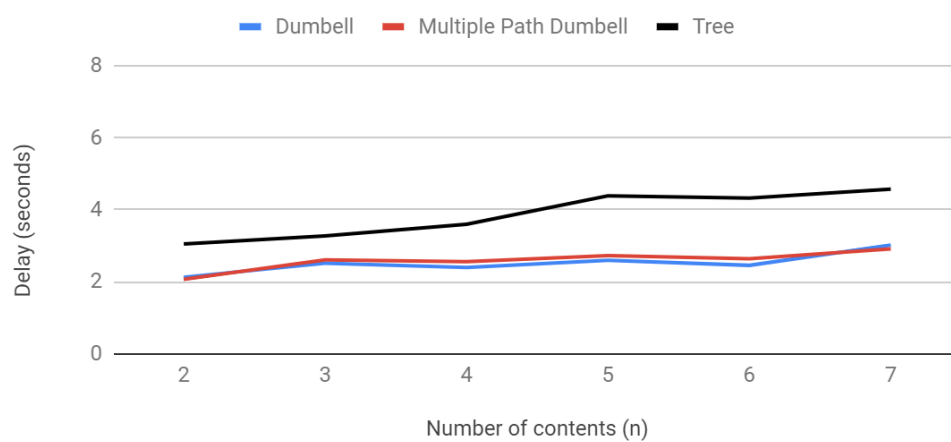


➤ Σύγκριση απόδοσης τοπολογιών.

Throughput (WiFi)



Delay (WiFi)

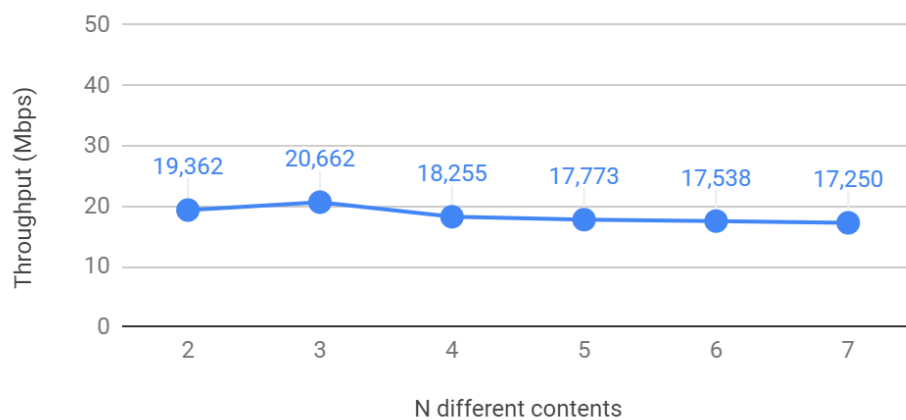


Όταν έχουμε ασύρματη σύνδεση μέσω WiFi, παρατηρούμε ότι το *Throughput* σε όλες τις τοπολογίες παρουσιάζει πτωτική τάση όσο αυξάνεται ο αριθμός των διαθέσιμων περιεχομένων. Τις χαμηλότερες τιμές σε *Throughput* παρατηρούμε στην τοπολογία Tree, ενώ οι υπόλοιπες δύο τοπολογίες παρουσιάζουν παρόμοιες τιμές *Throughput*.

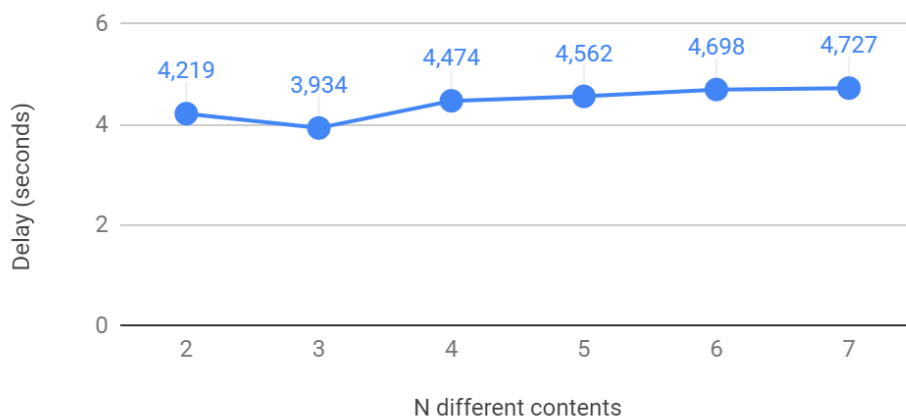
5.3.3 Ασύρματη σύνδεση μέσω LTE

➤ Αιτήματα για περιεχόμενο στην τοπολογία Dumbbell.

Throughput_for_N_contents (Dumbbell)

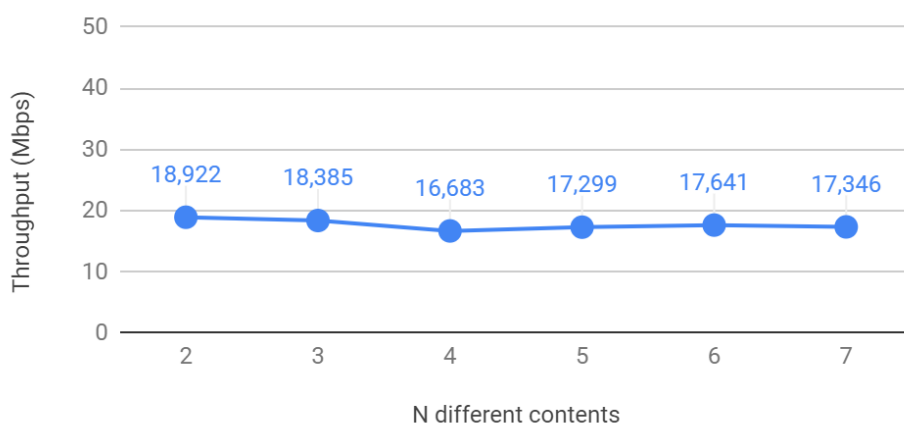


Delay_for_N_contents (Dumbbell)

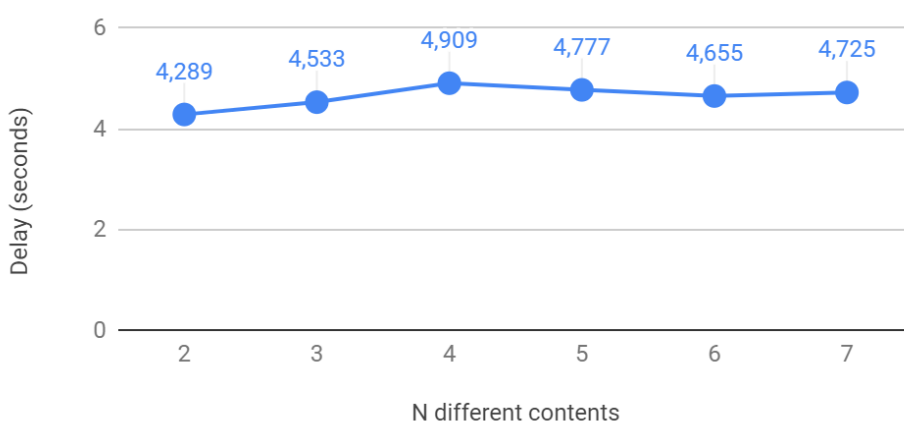


➤ Αιτήματα για περιεχόμενο στην τοπολογία Dumbbell πολλαπλών διαδρομών.

Throughput_for_N_contents (Multiple Path Dumbbell)

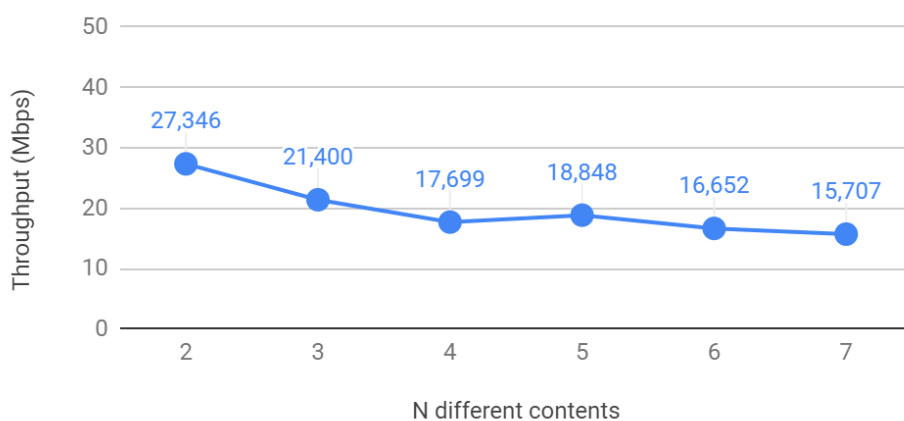


Delay_for_N_contents (Multiple Path Dumbbell)

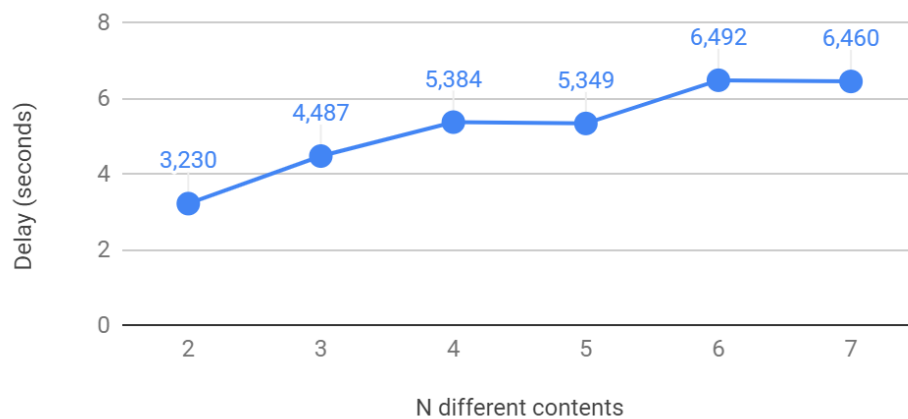


➤ Αιτήματα για περιεχόμενο στην τοπολογία Tree.

Throughput_for_N_contents (Tree)

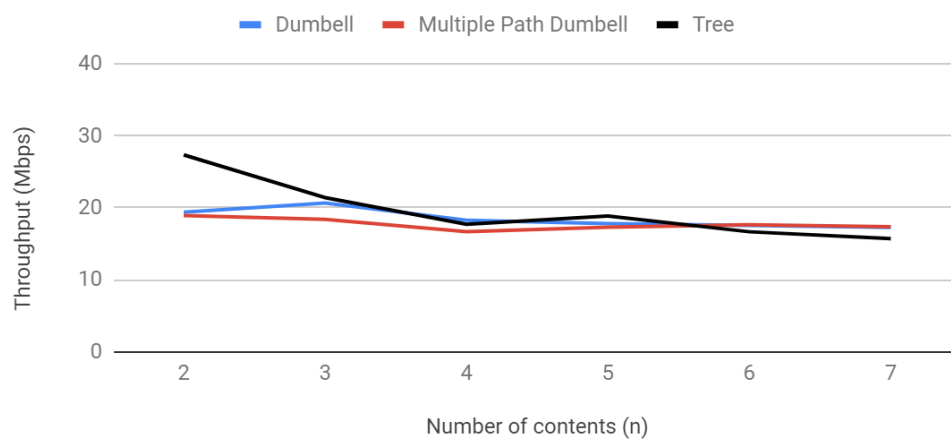


Delay_for_N_contents (Tree)

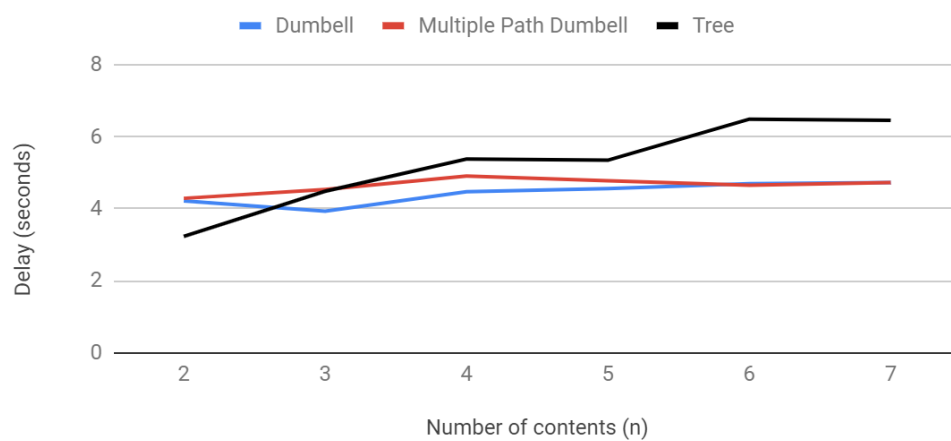


➤ Σύγκριση απόδοσης τοπολογιών.

Throughput (LTE)



Delay (WiFi)



Όταν έχουμε ασύρματη σύνδεση μέσω LTE, παρατηρούμε ότι οι τοπολογίες Dumbbell και Dumbbell πολλαπλών διαδρομών παρουσιάζουν σταθερή συμπεριφορά στο *Throughput* ανεξάρτητα του αριθμού περιεχομένων που υπάρχουν διαθέσιμα στον παραγωγό και ζητούνται από τον καταναλωτή. Στην τοπολογία Tree βλέπουμε ότι το *Throughput* αν και παρουσιάζει την μεγαλύτερη τιμή όταν υπάρχουν λίγα διαθέσιμα περιεχόμενα, ακολουθεί πτωτική πορεία όσο αυξάνεται ο αριθμός των διαθέσιμων περιεχομένων. Σε σύγκριση με τους υπόλοιπους τρόπους σύνδεσης, η ασύρματη σύνδεση μέσω LTE μας δίνει τις χαμηλότερες τιμές *Throughput*.

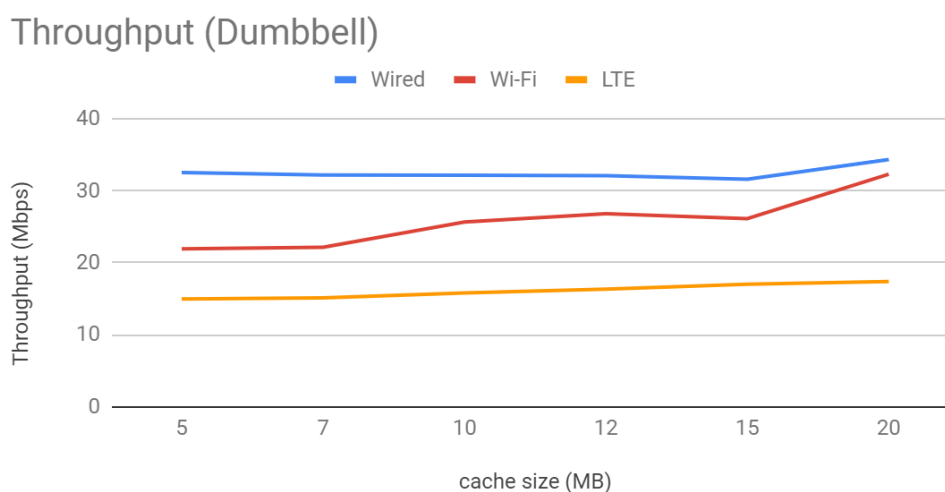
5.4 Σύγκριση συνδέσεων σεναρίων

Σε αυτή την ενότητα συγκρίνουμε τα αποτελέσματα σε *Throughput* και *Delay* που πήραμε για κάθε διαφορετικό τρόπο σύνδεσης σε όλες τις τοπολογίες που εξετάσαμε. Τα παρακάτω διαγράμματα χωρίζονται ανάλογα με το σενάριο από το οποίο προέκυψαν.

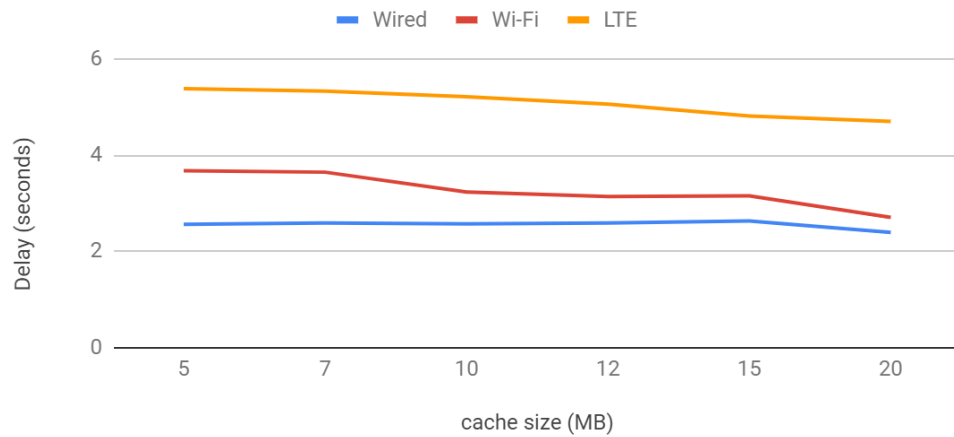
Ωστόσο, η σύγκριση αυτή δεν έχει νόημα για το σενάριο 2 καθώς, όπως φαίνεται και στα διαγράμματα της ενότητας 5.2, ο μέγιστος όγκος πληροφοριών που δύναται να μεταφερθεί ανά μονάδα χρόνου (*Bandwidth*) σε κάθε τρόπο σύνδεσης του σεναρίου 2 διαφέρει σημαντικά.

5.4.1 Σύγκριση συνδέσεων Σεναρίου 1^{ου}

➤ Σύγκριση συνδέσεων στην τοπολογία Dumbbell.

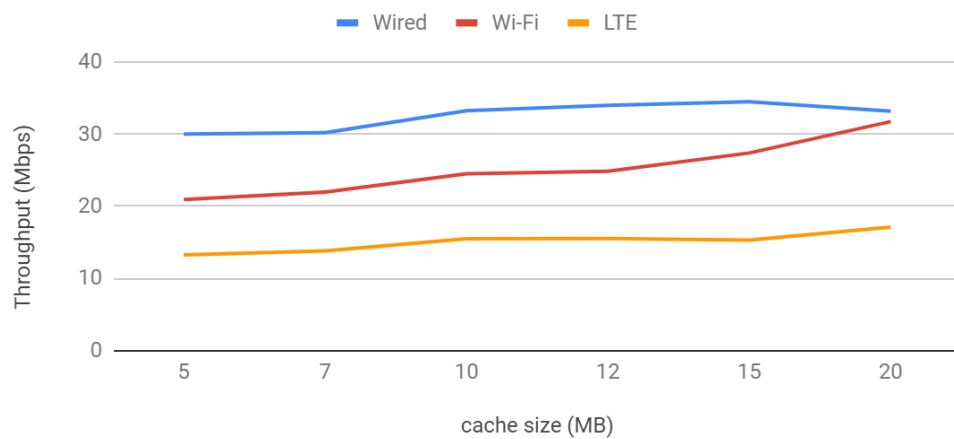


Delay (Dumbbell)

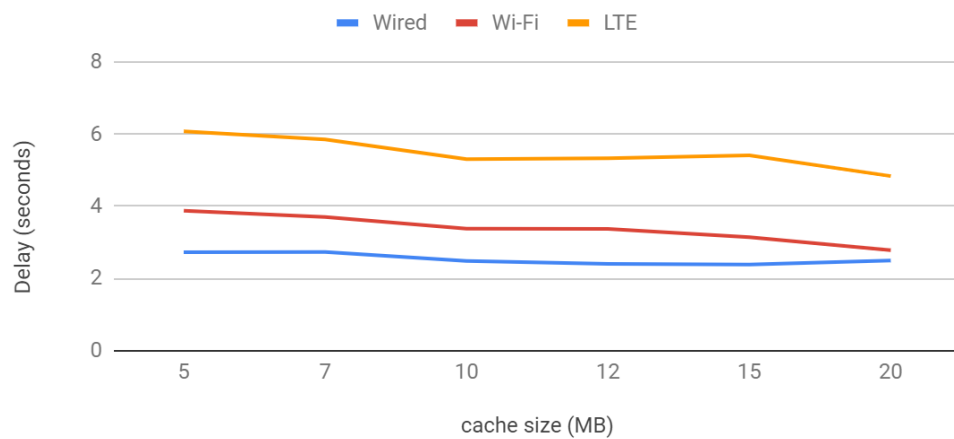


➤ Σύγκριση συνδέσεων στην τοπολογία Dumbbell πολλαπλών διαδρομών.

Throughput (Multiple Path Dumbbell)

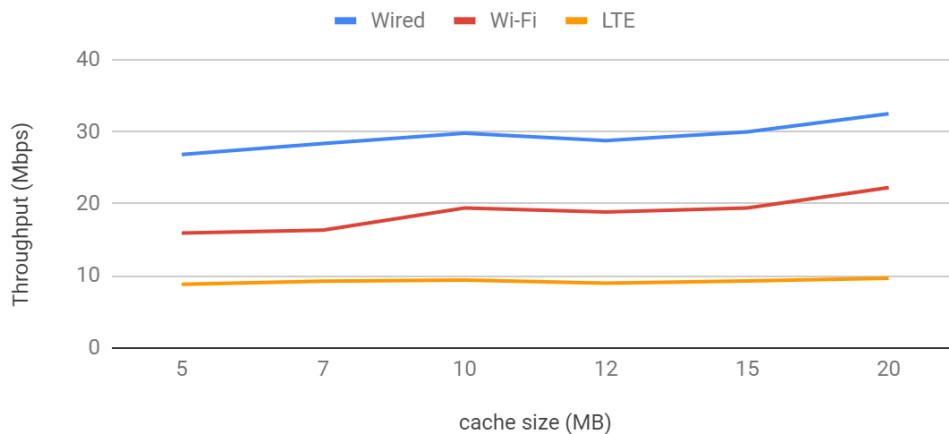


Delay (Multiple Path Dumbbell)

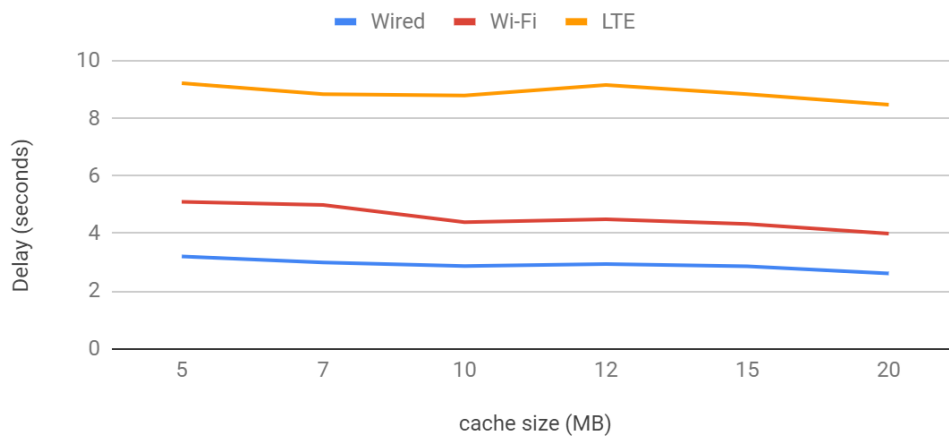


➤ Σύγκριση συνδέσεων στην τοπολογία Tree.

Throughput (Tree)



Delay (Tree)

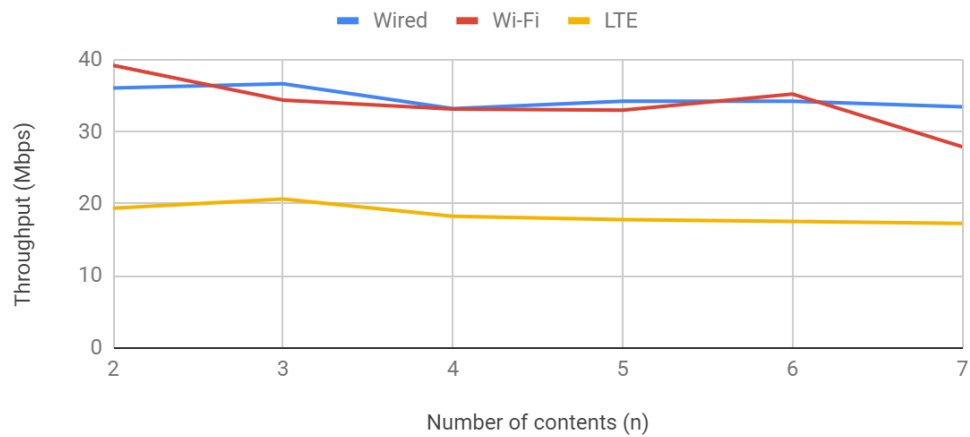


Στη σύγκριση των συνδέσεων του σεναρίου 1, παρατηρούμε συνολικά ότι η ενσύρματη σύνδεση μας δίνει τις καλύτερες τιμές *Throughput* και *Delay* σε όλες τις τοπολογίες ενώ στις τοπολογίες Dumbbell και Dumbbell πολλαπλών διαδρομών όταν το μέγεθος της κρυφής μνήμης είναι 20MB οι τιμές αυτές τείνουν να συγκλίνουν. Η ασύρματη σύνδεση μέσω LTE μας δίνει σε κάθε τοπολογία τις λιγότερο καλές τιμές.

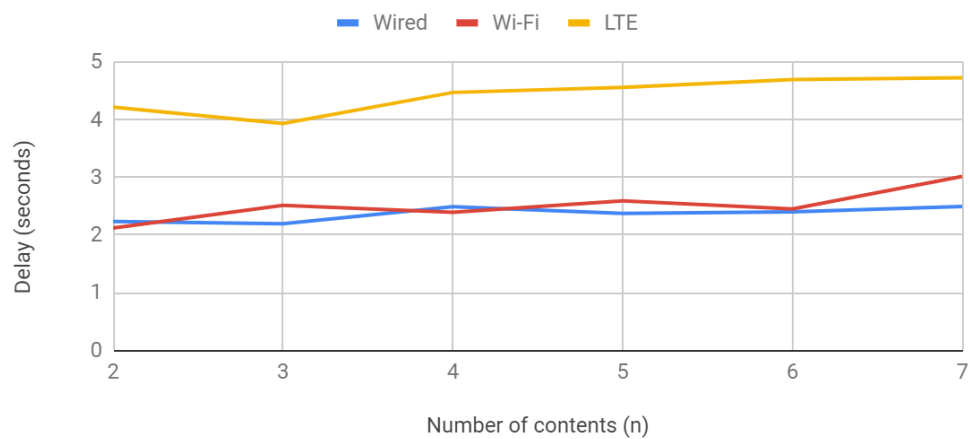
5.4.2 Σύγκριση συνδέσεων Σεναρίου 3^{ου}

➤ Σύγκριση συνδέσεων στην τοπολογία Dumbbell.

Throughput (Dumbbell)

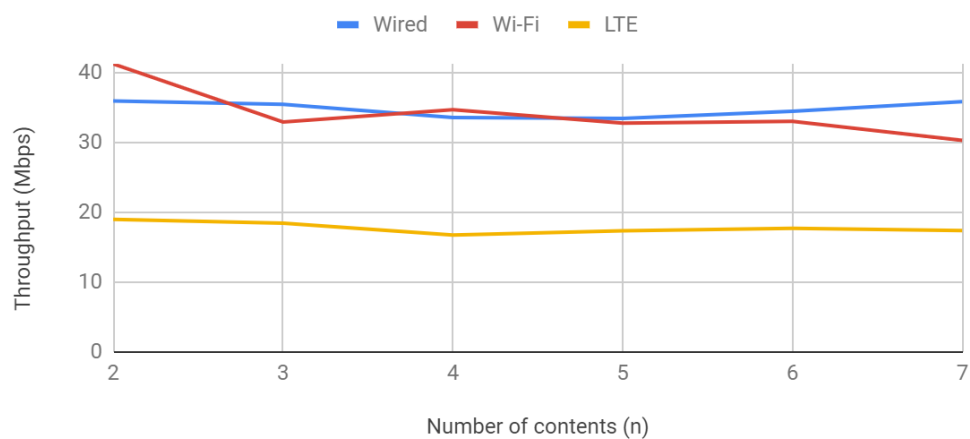


Delay (Dumbbell)

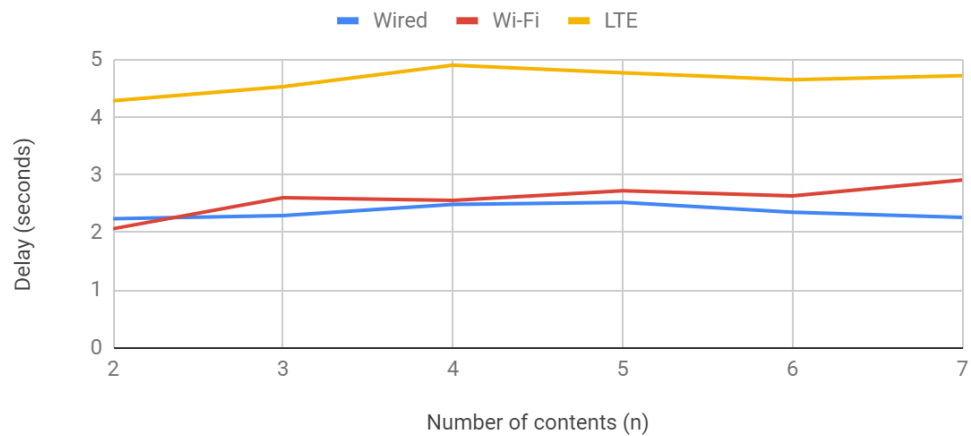


➤ Σύγκριση συνδέσεων στην τοπολογία Dumbbell πολλαπλών διαδρομών.

Throughput (Multiple Path Dumbbell)

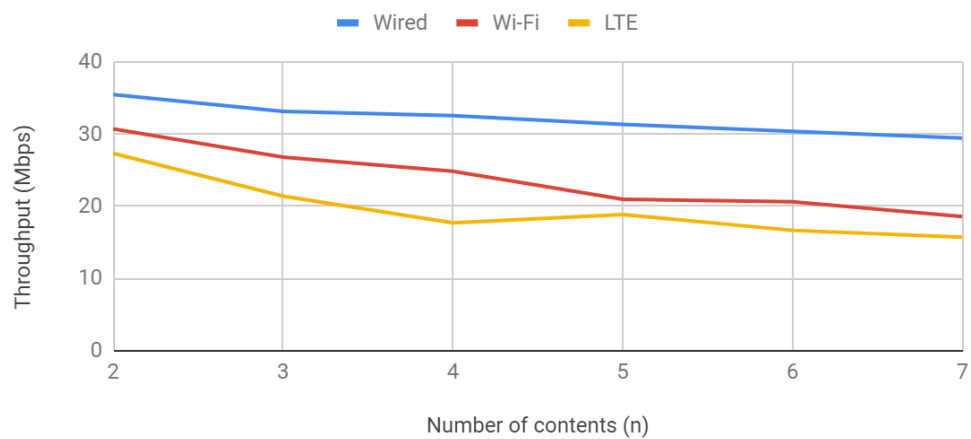


Delay (Multiple Path Dumbbell)

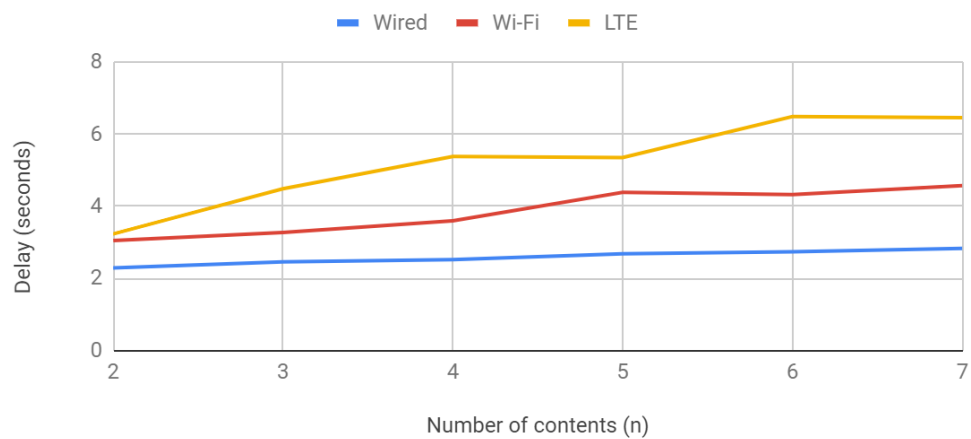


➤ Σύγκριση συνδέσεων στην τοπολογία Tree.

Throughput (Tree)



Delay (Tree)



Στη σύγκριση των συνδέσεων του σεναρίου 3, παρατηρούμε ότι η ενσύρματη σύνδεση και η ασύρματη σύνδεση μέσω WiFi μας δίνουν σχετικά σταθερές και παρόμοιες τιμές *Throughput* στις τοπολογίες Dumbbell και Dumbbell πολλαπλών διαδρομών ενώ η ασύρματη σύνδεση μέσω LTE μας δίνει αρκετά χαμηλότερες τιμές.

Στην τοπολογία Tree, παρατηρούμε ότι το *Throughput* πέφτει όσο αυξάνονται ο αριθμός των διαθέσιμων περιεχομένων στον παραγωγό ανεξάρτητα της τοπολογίας. Ωστόσο, με ενσύρματη σύνδεση παρατηρούμε τις μεγαλύτερες τιμές *Throughput* ενώ με ασύρματη σύνδεση μέσω LTE τις χαμηλότερες.

Κεφάλαιο 6: Συμπεράσματα

Ο σκοπός αυτής της εργασίας ήταν η παρουσίαση μιας νέας αρχιτεκτονικής δικτύων, της hICN, η οποία βασιζόμενη σε ήδη υπάρχουσες σχετικές μελέτες και αρχιτεκτονικές εισάγει στον τεράστιο κλάδο των δικτύων υπολογιστών μια νέα οπτική για τον τρόπο λειτουργίας του διαδικτύου.

Πιο συγκεκριμένα, προσομοιώσαμε την λειτουργία της συγκεκριμένης αρχιτεκτονικής με το εργαλείο nICN και δημιουργήσαμε τρεις πιθανές τοπολογίες δικτύου. Για κάθε μια από τις τρεις τοπολογίες δημιουργήσαμε τρία σενάρια (βλέπε ενότητα 4.3), μεταβλητές των οποίων ήταν ο μέγιστος όγκος πληροφοριών που δύναται να μεταφερθεί ανά μονάδα χρόνου (*Bandwidth*), το μέγεθος της κρυφής μνήμης (*cache size*) των δρομολογητών του συστήματος και ο αριθμός των διαθέσιμων περιεχομένων στον παραγωγό αντίστοιχα. Επιπλέον, εκτελέσαμε τα πειράματα για συνδέσεις μέσω ενσύρματης καλωδίωσης, ασύρματης επικοινωνίας μέσω WiFi και ασύρματης επικοινωνίας μέσω LTE.

Οι δύο βασικές μετρήσεις που εξάγαμε από τα πειράματα μας είναι ο πραγματικός όγκος πληροφοριών που μεταφέρεται ανά μονάδα χρόνου (*Throughput*) και ο χρόνος που απαιτείται για την μεταφορά των ζητούμενων πληροφοριών (*Delay*) και με βάση αυτές είμαστε σε θέση να αξιολογήσουμε την συμπεριφορά και την απόδοση των τοπολογιών που δημιουργήσαμε.

Αρχικά, αναλύοντας τα αποτελέσματα του πρώτου σεναρίου παρατηρούμε ότι το καλύτερο *Throughput* και κατά συνέπεια και *Delay* σε όλες τις τοπολογίες έχει η ενσύρματη σύνδεση τόσο όταν οι χρήστες ζητούν το ίδιο περιεχόμενο όσο και όταν οι χρήστες ζητούν διαφορετικό περιεχόμενο. Τις χαμηλότερες τιμές *Throughput* παρατηρούμε στην ασύρματη σύνδεση μέσω LTE. Γενικότερα, οι ταχύτερες είναι μεγαλύτερες όταν οι χρήστες ζητούν το ίδιο περιεχόμενο και μικρότερες όταν ζητούν διαφορετικό. Δεδομένου ότι το μέγεθος του αρχείου που ζητάμε είναι σταθερά 10MB, όσο μεγαλώνει το μέγεθος της κρυφής μνήμης έχουμε και καλύτερα αποτελέσματα.

Στο δεύτερο σενάριο το μέγεθος της κρυφής μνήμης είναι σταθερό 20MB ενώ το μέγεθος του περιεχομένου είναι σταθερό 10MB. Επομένως, η κρυφή μνήμη χρησιμοποιείται σε κάθε αίτημα των χρηστών εκτός του πρώτου.

Παρατηρούμε ότι και πάλι η ενσύρματη σύνδεση μας δίνει τις μεγαλύτερες τιμές *Throughput* και τις μικρότερες καθυστερήσεις, ενώ η ασύρματη σύνδεση μέσω LTE το αντίθετο. Όπως είναι λογικό όσο αυξάνουμε το *Bandwidth* (έως και 100 Mbps) οι ταχύτητες βελτιώνονται αισθητά, με μέγιστη τιμή τα 58,763 Mbps που σημειώθηκαν στην τοπολογία *Dumbbell*.

Στο τρίτο σενάριο εισαγάγαμε το στοιχείο της τυχαιότητας ώστε να προσομοιώσουμε όσο το δυνατόν καλύτερα τις πραγματικές συνθήκες λειτουργίας ενός δικτύου. Έτσι, οι χρήστες διαλέγουν να καταναλώσουν τυχαία κάποιο από τα διαθέσιμα περιεχόμενα των παραγωγών. Δεδομένου ότι το μέγεθος της κρυφής μνήμης είναι σταθερό 20MB και το μέγεθος του περιεχομένου είναι σταθερό 10MB, η κρυφή μνήμη δύναται να αποθηκεύει προσωρινά μόνο δύο διαφορετικά περιεχόμενα. Ανάλογα με το τυχαίο περιεχόμενο που ζητείται, οι χρήστες μπορεί να εξυπηρετηθούν από την κρυφή μνήμη ενδιάμεσων δρομολογητών ή από τους παραγωγούς. Αυτό που παρατηρήσαμε στο σενάριο αυτό είναι ότι η ενσύρματη σύνδεση και η ασύρματη σύνδεση μέσω WiFi έχουν παρόμοια απόδοση στις τοπολογίες *Dumbbell* και *Dumbbell* πολλαπλών διαδρομών ανεξάρτητα του αριθμού των διαθέσιμων περιεχομένων, ενώ η ασύρματη σύνδεση μέσω LTE μένει πιο χαμηλά. Στην τοπολογία *Tree* παρατηρήσαμε ότι όσο αυξάνεται ο αριθμός των διαθέσιμων περιεχομένων στον παραγωγό οι τιμές του *Throughput* μειώνονται ανεξάρτητα του τρόπου σύνδεσης. Το καλύτερο *Throughput* στην τοπολογία *Tree* επιτυγχάνεται στην τοπολογία *Dumbbell*.

Συνοψίζοντας, κατά τη διάρκεια της διεξαγωγής των πειραμάτων αναγνωρίσαμε την ανάγκη για εισαγωγή ακόμα περισσότερων παραγωγών, καταναλωτών και δρομολογητών ώστε να προσομοιωθεί καλύτερα η λειτουργία του δικτύου. Επιπλέον, υπάρχουν κι άλλες τοπολογίες που χρησιμοποιούνται καθημερινά και αξίζει να μελετηθούν. Τέλος, ένα κρίσιμο σημείο είναι η εισαγωγή του στοιχείου της τυχαιότητας σε κάθε τοπολογία ώστε να φέρουμε το πείραμα όσο το δυνατόν πιο κοντά σε πραγματικές συνθήκες λειτουργίας. Αυτά τα σημεία αποτελούν τις προτάσεις μας για μελλοντική μελέτη.

Κεφάλαιο 7: Βιβλιογραφία

- [1] George Xylomenos, Christopher N. Ververidis, Vasilios A. Siris, Nikos Fotiou, Christos Tsilopoulos, Xenofon Vasilakos, Konstantinos V. Katsaros, and George C. Polyzos. **A Survey of Information-Centric Networking Research** on IEEE Communications Surveys & Tutorials Volume: 16 , Issue: 2 , Second Quarter 2014.
- [2] Ciczn <https://wiki.fd.io/view/Ciczn>
- [3] Vicn <https://wiki.fd.io/view/Vicn>
- [4] George Pallis, Athena Vakali. **Insight and Perspectives for Content Delivery Networks** on Communications Of The ACM Vol. 49, No. 1, January 2006.
- [5] Sherif Kamel Hussein, Prof. Imane Aly Saroit Ismail, Prof. S. H. Ahmed. **Triangle Routing Problem in Mobile IP**.
https://www.academia.edu/5684554/Triangle_Routing_Problem_in_Mobile_IP
- [6] Vasileios Giotsas, Shi Zhou. **Valley-free violation in Internet routing - Analysis based on BGP Community data** on IEEE ICC 2012 - Communication QoS, Reliability and Modeling Symposium.
- [7] Lixia Zhang, Alexander Afanasyev, Jeffrey Burke, Van Jacobson, kc claffy, Patrick Crowley, Christos Papadopoulos, Lan Wang, Beichuan Zhang. **Named Data Networking** on ACM SIGCOMM Computer Communication Review Volume 44, Number 3, July 2014.
- [8] Christos-Alexandros Sarros, Sotiris Diamantopoulos, Sergi Rene, Ioannis Psaras, Adisorn Lertsinsruttavee, Carlos Molina-Jimenez, Paulo Mendes, Rute Sofia, Arjuna Sathiaselalan, George Pavlou, Jon Crowcroft, and Vassilis Tsaoussidis. **Connecting the Edges: A Universal, Mobile-Centric, and Opportunistic Communications Architecture** on IEEE Communications Magazine, February 2018.
- [9] Christos-Alexandros Sarros, Adisorn Lertsinsruttavee , Carlos Molina-Jimenez , Konstantinos Prasopoulos , Sotiris Diamantopoulos , Dimitris Vardalis , Arjuna Sathiaselalan. **ICN-based Edge Service Deployment in Challenged Networks**.
- [10] Mauro Conti, Ankit Gangwal, Muhammad Hassan, Chhagan Lal, Eleonora Losiouk. **The Road Ahead for Networking: A Survey on ICN-IP Coexistence Solutions**.
- [11] L. Muscariello, G. Carofiglio, J. Auge, M. Papalini. **Hybrid Information-Centric Networking** on Internet Area WG, December 2018.
- [12] E. Rosen, A. Viswanathan, R. Callon, M. Papalini. **Multiprotocol Label Switching Architecture** on Network WG, January 2001.

- [13] Jordan Augé, Giovanna Carofiglio, Marcel Enguehard, Luca Muscariello, Mauro Sardara. **Simple and efficient ICN network virtualization with vICN** on ICN '17, September 26–28, 2017, Berlin, Germany.
- [14] Mauro Sardara, Luca Muscariello, Jordan Augé, Marcel Enguehard, Alberto Compagno, Giovanna Carofiglio. **Virtualized ICN (vICN): Towards a Unified Network Virtualization Framework for ICN Experimentation** on ICN '17, September 26–28, 2017, Berlin, Germany.
- [15] NS-3 <https://www.nsnam.org/about/>
- [16] NS-3 Wi-Fi Module <https://www.nsnam.org/docs/models/html/wifi.html>
- [17] Cache Replacement Policies
https://en.wikipedia.org/wiki/Cache_replacement_policies
- [18] TCP Vegas https://en.wikipedia.org/wiki/TCP_Vegas
- [19] Giovanna Carofiglio, Massimo Gallo, Luca Muscariello, Michele Papalini.
Multipath Congestion Control in Content-Centric Networks.