



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ
ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

Εκφόρτωση Δεδομένων HTTP Μέσω Διακομιστή Μεσολάβησης TCP Πολλαπλών Διαδρομών

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Χρήστος Σ. Πολλάλης

Επιβλέπων : Ευστάθιος Συκάς
Καθηγητής Ε.Μ.Π.

Αθήνα, Σεπτέμβριος 2015



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ
ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

Εκφόρτωση Δεδομένων HTTP Μέσω Διακομιστή Μεσολάβησης TCP Πολλαπλών Διαδρομών

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Χρήστος Σ. Πολλάλης

Επιβλέπων : Ευστάθιος Συκάς
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 28^η Σεπτεμβρίου 2015.

.....
Ευστάθιος Συκάς
Καθηγητής Ε.Μ.Π.

.....
Μιχαήλ Θεολόγου
Καθηγητής Ε.Μ.Π.

.....
Γεώργιος Στασινόπουλος
Καθηγητής Ε.Μ.Π.

Αθήνα, Σεπτέμβριος 2015

.....
Χρήστος Σ. Πολλάλης

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Copyright © Χρήστος Πολλάλης, 2015.

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

Περίληψη

Εξαιτίας της σημερινής πληθώρας έξυπνων συσκευών και της διάχυτης χρήσης σχεδόν κάθε είδους τεχνολογίας μεταφοράς δεδομένων, τα δίκτυα υπολογιστών έρχονται αντιμέτωπα με δυσκολίες όσο αφορά την ανταπόκρισή τους στις απαιτήσεις των τελικών χρηστών σχετικά με την διαθεσιμότητα εύρους ζώνης και την ποιότητα εμπειρίας (Quality of Experience – QoE). Το Multipath TCP (MPTCP), ή διαφορετικά TCP Πολλαπλών Διαδρομών, ως μία επέκταση του παραδοσιακού TCP, επιτρέπει, διάφανα ως προς το Στρώμα Εφαρμογής, την ταυτόχρονη αξιοποίηση των πολλαπλών διεπαφών μίας έξυπνης συσκευής, η οποία υποστηρίζει παραπάνω από μία τεχνολογία πρόσβασης στο Διαδίκτυο.

Ως αποτέλεσμα, το MPTCP δίνει την δυνατότητα συσσωμάτωσης εύρους ζώνης από περισσότερες από μια διεπαφές του αντίστοιχου έξυπνου τερματικού, ενώ επίσης, συνεισφέρει στην επίτευξη αδιάκοπης κάθετης διαπομπής μεταξύ των διαθέσιμων τεχνολογιών μεταφοράς δεδομένων. Επιπρόσθετα, ο τροποποιημένος αλγόριθμος ελέγχου συμφόρησης επιτρέπει την δυναμική εκφόρτωση όγκου δεδομένων βασιζόμενος στην ύπαρξη εναλλακτικών διαδρομών δρομολόγησης.

Παρ' όλα αυτά, οι σημερινές πλατφόρμες παροχής υπηρεσιών υστερούν όσο αφορά την ενσωμάτωση του MPTCP. Σκοπός της παρούσας διπλωματικής εργασίας είναι η αξιολόγηση ενός μηχανισμού υπό την μορφή ενός HTTP Διακομιστή Μεσολάβησης (HTTP Proxy Server), ο οποίος θα λειτουργεί ως μετατροπέας πρωτοκόλλου. Με αυτόν τον τρόπο, έξυπνα τερματικά με την δυνατότητα αξιοποίησης πολλαπλών διαδρομών θα μπορούν να επωφεληθούν από το Multipath TCP ακόμα και σε περίπτωση επικοινωνίας με έναν διακομιστή που υποστηρίζει την παραδοσιακή έκδοση του πρωτοκόλλου.

Λέξεις Κλειδιά: Multipath TCP, Έλεγχος Συμφόρησης, Κάθετη Διαπομπή, Πολιτική Δρομολόγησης, Εκφόρτωση Δεδομένων, Ασύρματο Δίκτυο

Abstract

Due to today's plethora of smart devices and the pervasive usage of almost every kind of data communication technology, data networks start to have a hard time meeting end users' demands regarding available bandwidth and desired quality of experience (QoE). Multipath TCP, as an extension to Legacy TCP, permits transparently to the application layer the concurrent use of the multiple available interfaces on a multihomed device.

Consequently, MPTCP allows for bandwidth aggregation, as well as seamless vertical handover due to the presence of more than one data communication technology per smart terminal and the simultaneous utilization of the multiple available interfaces. Additionally, MPTCP's modified congestion control algorithm allows for dynamic data offloading as a consequence of the existence of alternative routing paths.

Nevertheless, MPTCP implementation is still lagging behind, as service delivery platforms opt not to adapt the protocol in their native infrastructure. The purpose of this thesis is the evaluation of a mechanism in the form of an HTTP Proxy that will serve as a protocol converter. As a result, MPTCP-capable hosts will be able to still benefit from Multipath TCP even when communicating with a legacy server.

Keywords: Multipath TCP; Congestion Control; Vertical Handover; Policy Routing; Offloading; Wireless

Περιεχόμενα

Περίληψη	5
Abstract	7
Περιεχόμενα	9
Πίνακας Σχημάτων	11
Πίνακας Πινάκων	13
1. Εισαγωγή.....	15
1.1. Multipath TCP.....	15
1.2. Δομή Κειμένου	16
2. Θεωρητικό Υπόβαθρο	19
2.1. Επίπεδο Ελέγχου	19
2.1.1. Τριμερής Χειραψία MPTCP	19
2.1.2. Διαχείριση Ροών TCP	20
2.1.3. Τερματισμός Σύνδεσης MPTCP	21
2.1.4. Συμβατότητα με το Σημερινό Internet	22
2.2. Επίπεδο Δεδομένων	22
2.2.1. Αντιστοίχιση Πακέτων Μεταξύ Επιπέδων TCP/MPTCP	22
2.2.2. Επιβεβαίωση Λήψης Δεδομένων (ACKs)	24
2.2.3. Απαίτηση Χωρητικότητας TCP Buffers	24
2.2.4. Ενοποιημένος Αλγόριθμος Ελέγχου Συμφόρησης	26
2.3. Λειτουργίες του MPTCP.....	27
2.4. Διακομιστής Μεσολάβησης MPTCP (Proxy)	29
2.4.1. Οφέλη του MPTCP Proxy.....	30
2.4.2. Είδη MPTCP Proxy	30
3. Έλεγχος Διαθέσιμου Εύρους Ζώνης σε Linux Kernel.....	33
3.1. Διαχείριση Ουρών Εξερχόμενης Διαδικτυακής Κίνησης.....	33
3.1.1. Ο Αλγόριθμος Hierarchical Token Bucket.....	35
3.2. Netfilter Framework	36
3.2.1. IPtables	37
4. Μηχανισμός Επιβολής Bandwidth Throttling	39
4.1. Scapy (Packet) Sniffer	39
4.2. JSON-RPC	41
4.3. Bandwidth Throttling Daemon	43
4.4. Αρχείο Παραμετροποίησης Μηχανισμού	45
5. Αξιολόγηση Πρωτοκόλλου	47
5.1. Legacy TCP.....	48
5.2. Σύγκριση Αλγορίθμων Ελέγχου Συμφόρησης	49
5.3. Δυναμική Εκφόρτωση Δεδομένων	52
5.3.1. TCP Reno.....	52

5.3.2. Linked Increase Algorithm	53
5.3.3. Opportunistic Linked Increase Algorithm	54
5.4. Επιπτώσεις Χωρητικότητας TCP Buffers	55
5.5. Επιπτώσεις Μεγέθους MTU	60
5.6. Κάθετη Διαπομπή	62
5.7. Bandwidth Throttling	64
6. Επίλογος	73
6.1. Σχετική Έρευνα	73
6.2. Σύνοψη	74
Βιβλιογραφία	77
Παράρτημα	81

Πίνακας Σχημάτων

1	Διαστρωμάτωση MPTCP	19
2	Τριμερής χειραψία MPTCP και συγχρονισμός ροών	20
3	Διαφήμιση διαθέσιμων διευθύνσεων IP	21
4	Αφαίρεση διευθύνσεων IP από την τρέχουσα σύνδεση MPTCP.....	21
5	Επιλογή σήματος ακολουθίας δεδομένων (Subtype = 0x2)	23
6	Διάταξη διακομιστή μεσολάβησης TCP πολλαπλών διαδρομών	29
7	Παρεμβολή διαφανή MPTCP Proxy.....	30
8	Παρεμβολή MPTCP Proxy εκτός προκαθορισμένης διαδρομής	31
9	Προσομοίωση πολλαπλών ουρών εξερχόμενης διαδικτυακής κίνησης ..	34
10	Σχηματική αναπαράσταση αλγορίθμου HTB	36
11	Διάσχιση σταδίων του Netfilter Framework	37
12	Πλήρης απεικόνιση Netfilter Framework και IPtables	38
13	HTB qdisc	45
14	Πειραματική διάταξη	47
15	Επίδοση παραδοσιακού TCP σε Mb/s	48
16	Επίδοση MPTCP σε Mb/s με χρήση Reno και LIA.....	49
17	Επίδοση MPTCP σε Mb/s με χρήση OLIA	51
18	Επίδοση MPTCP σε Mb/s υπό συμφόρηση με χρήση Reno	52
19	Επίδοση MPTCP σε Mb/s υπό συμφόρηση με χρήση LIA.....	54
20	Επίδοση MPTCP σε Mb/s υπό συμφόρηση με χρήση OLIA	55
21	Επίπτωση χωρητικότητας TCP recv buffer.....	57
22	Επίδοση MPTCP σε Mb/s συναρτήσσει TCP recv buffer size (1).....	58
23	Επίδοση MPTCP σε Mb/s συναρτήσσει TCP recv buffer size (2).....	59
24	Επίδοση MPTCP σε Mb/s συναρτήσσει TCP recv buffer size (3).....	59
25	Επίδοση MPTCP σε Mb/s για MTU 1200	61
26	Επίδοση MPTCP σε Mb/s για MTU 1000	62
27	Κάθετη διαπομπή	63
28	Κάθετη διαπομπή εν απουσία κατάλληλης σηματοδοσίας.....	63
29	Bandwidth throttling στο 1 Mb/s για αρχείο 50 KB	65
30	Bandwidth throttling στο 1 Mb/s για αρχείο 1 MB	66
31	Bandwidth throttling στο 1 Mb/s για αρχεία 2, 3, 4 MB.....	66
32	Bandwidth throttling στο 1 Mb/s για αρχείο 5 MB	67
33	Bandwidth throttling στο 1 Mb/s για αρχεία 10, 20 MB	68
34	Bandwidth throttling στο 1 Mb/s για αρχείο 50 MB	68
35	Bandwidth throttling στο 1 Mb/s για αρχείο 100 MB.....	69
36	Bandwidth throttling στο 1 Mb/s για αρχείο 200 MB.....	70
37	Bandwidth throttling στα 5 Mb/s για αρχείο 100 MB.....	71
38	Bandwidth throttling στα 8 Mb/s για αρχείο 100 MB.....	71

Πίνακας Πινάκων

1	Επιλογές MPTCP	41
2	Αριθμητικές μετρήσεις σχήματος 16	50
3	Αριθμητικές μετρήσεις σχήματος 17	51
4	Αριθμητικές μετρήσεις σχήματος 18	53
5	Αριθμητικές μετρήσεις σχήματος 19	53
6	Αριθμητικές μετρήσεις σχήματος 20	55
7	Αριθμητικές μετρήσεις σχήματος 21	57
8	Αριθμητικές μετρήσεις σχήματος 22	59
9	Αριθμητικές μετρήσεις σχήματος 23	60
10	Αριθμητικές μετρήσεις σχήματος 24	60
11	Αριθμητικές μετρήσεις σχήματος 25	61
12	Αριθμητικές μετρήσεις σχήματος 26	61

1. Εισαγωγή

Ο κόσμος του Internet εξελίσσεται με πολύ ταχείς ρυθμούς. Ο αριθμός των συνδεδεμένων στο Διαδίκτυο συσκευών αυξάνεται συνεχώς και εκτιμάται πως το 2020 θα υπάρχουν περίπου 25 δισεκατομμύρια συσκευές που θα ζητούν πρόσβαση στον παγκόσμιο ιστό [1]. Ακόμα και σήμερα, τα δίκτυα μεταφοράς δεδομένων, ειδικά όταν αναφερόμαστε σε κινητές τηλεπικοινωνίες, αντιμετωπίζουν δυσκολίες αναφορικά με την υποστήριξη της συνεχώς αυξανόμενης κατανάλωσης εύρους ζώνης, ενώ την ίδια στιγμή οι πάροχοι υπηρεσιών υποχρησιμοποιούν τις πολλαπλές διαθέσιμες ραδιοηλεκτρικές διεπαφές ανά τερματικό.

1.1. Multipath TCP

Ο συνεχόμενα αυξανόμενος αριθμός έξυπνων συσκευών, η ύπαρξη διακομιστών (servers) που υποστηρίζουν υποδίκτυα διαφορετικών τεχνολογιών, καθώς και η παροχή πλεονάζουσων διαδρομών από τα κέντρα δεδομένων, τείνουν να απαρχαιώσουν την διαστρωμάτωση του TCP/IP, η οποία είχε αρχικά σχεδιαστεί σε μία εποχή που τα τερματικά αποτελούνταν αυστηρά από μία δικτυακή διεπαφή. Ενώ ήδη διάφορες τεχνικές έχουν προταθεί για την επίτευξη εκφόρτωσης όγκου δεδομένων μεταξύ διαφορετικών ενσύρματων και ασύρματων τεχνολογιών πρόσβασης, το Transmission Control Protocol (TCP) υστερεί της δυνατότητας περί αποδοτικής χρησιμοποίησης όλων των διαθέσιμων διεπαφών ενός έξυπνου τερματικού, το οποίο υποστηρίζει παραπάνω από μία τεχνολογία μεταφοράς δεδομένων.

Το Multipath TCP (MPTCP) [2] επεκτείνει το παραδοσιακό TCP επιτρέποντάς του την διαπραγμάτευση και εγκατάσταση επιπρόσθετων υπο-ροών (subflows) μεταξύ των τελικών χρηστών ως μέρος της ίδιας σύνδεσης TCP, την οποία παρουσιάζει στο Στρώμα Εφαρμογής. Έτσι, η εκάστοτε σύνδεση TCP έχει την δυνατότητα να χρησιμοποιήσει ταυτόχρονα πολλαπλές διαδρομές δρομολόγησης προσφέροντας βελτιωμένη απόδοση (throughput), αποδοτικότερη χρησιμοποίηση των δικτυακών πόρων, ανθεκτικότητα σε τυχόν δυσλειτουργία του εκάστοτε υποδικτύου, καθώς και δυναμική εκφόρτωση δεδομένων αναφορικά με τα χαρακτηριστικά των αντίστοιχων ζεύξεων. Παράλληλα, η μόχλευση του MPTCP με (κυψελωτά) δίκτυα κινητών τηλεπικοινωνιών μπορεί να ενισχύσει την ποιότητα εμπειρίας των χρηστών προσφέροντας ευρωστία ως προς την χρονικά μεταβαλλόμενη διαθεσιμότητα των τεχνολογιών μεταφοράς δεδομένων και την υποβάθμιση της ποιότητας υπηρεσιών κατά την διάρκεια (οριζόντιων ή κάθετων) διαπομπών.

Ο έλεγχος συμφόρησης εκτελείται ξεχωριστά για κάθε υπο-ροή κάτω από την επίβλεψη ενός ενοποιημένου μηχανισμού ελέγχου συμφόρησης [3]. Αυτή η ιδιαιτερότητα επιτρέπει στο στρώμα του MPTCP να παρακολουθεί την δυναμική συμπεριφορά της χρονικά διαφοροποιημένης κατάστασης

συμφόρησης του εκάστοτε υποδικτύου με σκοπό την απομάκρυνση της διαδικτυακής κίνησης από διαδρομές υποβαθμισμένης αποδοτικότητας. Ως αποτέλεσμα, έξυπνα τερματικά και διακομιστές με πρόσβαση σε υποδίκτυα διαφορετικών τεχνολογιών μέσω πολλαπλών διαδρομών έχουν την δυνατότητα να εκφορτώσουν όγκο δεδομένων μεταξύ των ταυτόχρονα χρησιμοποιούμενων διαδρομών δρομολόγησης. Παράλληλα, το MPTCP μπορεί να συνδράμει στην αποδοτικότερη διαχείριση υποδομών κινητών τηλεπικοινωνιών εκφορτώνοντας δεδομένα από το κυψελωτό ασύρματο δίκτυο κινητής επικοινωνίας στο δίκτυο Wi-Fi, όποτε αυτό είναι διαθέσιμο, με σκοπό την ελαχιστοποίηση της συμφόρησης του αντίστοιχου δικτύου. Ο συγκεκριμένος συνδυασμός της ευρείας κάλυψης που προσφέρει το δίκτυο 4G/3G, το οποίο όμως, μπορεί να υποφέρει από σημαντικό βαθμό συμφόρησης, με τις υψηλές ταχύτητες μεταφοράς δεδομένων που παρέχουν τα σημεία πρόσβασης Wi-Fi, μπορεί να οδηγήσει σε μία συνολικότερη ενίσχυση της ποιότητας υπηρεσιών και της γενικότερης απόδοσης.

Επιπροσθέτως, το MPTCP μπορεί να χρησιμοποιηθεί από μία έξυπνη (κινητή) συσκευή για την επίτευξη αδιάκοπης κάθετης διαπομπής. Σήμερα, οι έξυπνες συσκευές, κινητά τηλέφωνα και tablets, συνδέονται στο Internet μέσω αποκλειστικά μίας δικτυακής διεπαφής, κάτι που καθιστά πολλές φορές την μετάβαση μεταξύ διαφορετικών ασύρματων τεχνολογιών πρόσβασης μη-ομαλή με αποτέλεσμα την αποδιοργάνωση ή/και τερματισμό ήδη εγκατεστημένων συνδέσεων. Από την άλλη μεριά, το TCP Πολλαπλών Διαδρομών εφαρμόζει έναν *make-before-break* μηχανισμό [4] κατά τον οποίο στοχεύει στην εγκατάσταση μίας νέας υπο-ροής πριν την απόσπαση της αρχικής σύνδεσης. Με αυτόν τον τρόπο, το MPTCP έχει την δυνατότητα να υποστηρίξει σενάρια κινητικότητας χωρίς την πρόκληση δυσλειτουργίας στις υπάρχουσες συνδέσεις.

Παρ' όλα αυτά, το Multipath TCP δεν έχει υιοθετηθεί ακόμα από τις σημερινές πλατφόρμες παροχής υπηρεσιών. Η παρεμβολή ενός μηχανισμού μετατροπής πρωτοκόλλου υπό την μορφή ενός HTTP Διακομιστή Μεσολάβησης (Proxy) θα μπορούσε να λειτουργήσει ευεργετικά κατά την σταδιακή εγκατάσταση του πρωτοκόλλου στις υποδομές των παρόχων υπηρεσιών διαδικτύου, επιτρέποντας σε χρήστες με την δυνατότητα αξιοποίησης πολλαπλών διαδρομών δρομολόγησης να συνεχίσουν να επωφελούνται από τα πλεονεκτήματα που εισάγει το επεκταμένο πρωτόκολλο ακόμα και σε περίπτωση επικοινωνίας με συστήματα που επιμένουν να χρησιμοποιούν την παραδοσιακή έκδοση του TCP.

1.2. Δομή Κειμένου

Το υπόλοιπο του κειμένου είναι δομημένο ως ακολούθως:

Κεφάλαιο 2: Σε αυτό το κεφάλαιο, αρχικά, παρέχεται το θεωρητικό υπόβαθρο με σκοπό την εξοικείωση του αναγνώστη με τα τεχνικά χαρακτηριστικά του Multipath TCP τόσο σε επίπεδο ελέγχου, όσο και σε επίπεδο δεδομένων. Επίσης, γίνεται ανάλυση της ανάγκης για αυξημένη

χωρητικότητα των buffers του εκάστοτε συστήματος, ώστε να καθίσταται δυνατή η επεξεργασία του αυξημένου αριθμού πακέτων που στέλνονται/λαμβάνονται ανά χρονική στιγμή. Στην συνέχεια, ακολουθεί αναφορά στις λειτουργίες που προσφέρει το MPTCP ως προς την αξιοποίηση των διαθέσιμων διεπαφών του αντίστοιχου έξυπνου τερματικού. Τέλος, επικεντρωνόμαστε στην λειτουργία του διακομιστή μεσολάβησης, ο οποίος μπορεί να συμβάλει στην βελτιστοποίηση της απόδοσης μεταφοράς δεδομένων, ενώ, παράλληλα, δίνει την δυνατότητα για εφαρμογή πολιτικών δρομολόγησης με σκοπό την καλύτερη διαχείριση των διαθέσιμων πόρων.

Κεφάλαιο 3: Στην συνέχεια, περιγράφεται μία τεχνική ελέγχου του διαθέσιμου εύρους ζώνης ανά διεπαφή στον πυρήνα του λειτουργικού συστήματος Linux. Συγκεκριμένα, εξετάζουμε την διαχείριση των ουρών της εξερχόμενης κίνησης δεδομένων σκοπεύοντας στην χειραγώγηση της επιλογής διεπαφής δρομολόγησης.

Κεφάλαιο 4: Σε αυτό το κεφάλαιο, εισαγάγουμε έναν μηχανισμό για τον έλεγχο του προσφερόμενου εύρους ζώνης ανά διεπαφή σε ένα έξυπνο τερματικό με σκοπό την δυναμική χειραγώγηση της εξερχόμενης διαδικτυακής κίνησης δεδομένων.

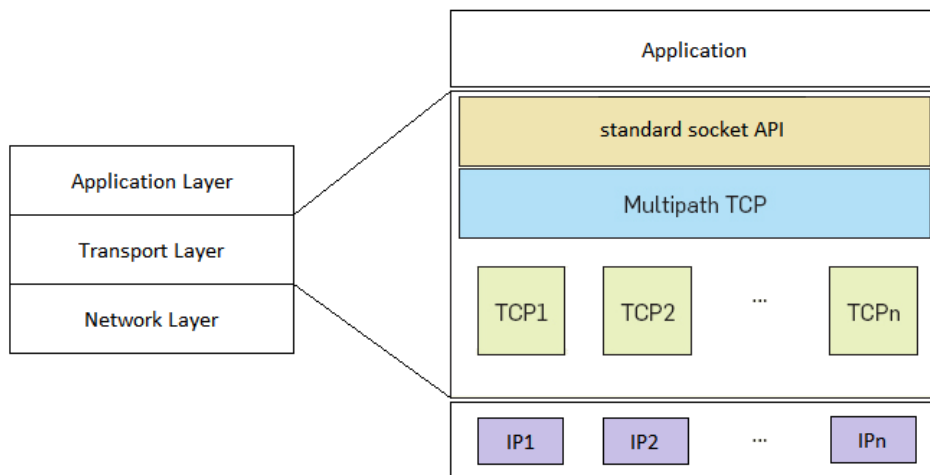
Κεφάλαιο 5: Σκοπός του συγκεκριμένου κεφαλαίου είναι η αξιολόγηση του πρωτοκόλλου, καθώς και η επίδειξη της λειτουργίας του μηχανισμού διαχείρισης εύρους ζώνης με σκοπό την επιβολή πολιτικών δρομολόγησης, κάνοντας αναφορά σε περισσότερο ρεαλιστικά σενάρια.

Κεφάλαιο 6: Εν κατακλείδι, συνοψίζουμε και σχολιάζουμε επί των αποτελεσμάτων της αξιολόγησης, ενώ, επίσης, κάνουμε αναφορές σε σχετικές έρευνες και στην τρέχουσα βιβλιογραφία.

2. Θεωρητικό Υπόβαθρο

2.1. Επίπεδο Ελέγχου

Το Multipath TCP χρησιμοποιεί ως μηχανισμό μεταφοράς δεδομένων σε επίπεδο υπο-ροής το παραδοσιακό TCP, όπως φαίνεται και στο *Σχήμα 1*. Αυτό συνεπάγεται πως μία νέα σύνδεση με την δυνατότητα αξιοποίησης ταυτόχρονα πολλαπλών διαδρομών ξεκινά ως μία απλή συνεδρία TCP. Το MPTCP χρειάζεται να σηματοδοτήσει τις επιπρόσθετες πληροφορίες ελέγχου με σκοπό να αξιοποιηθούν οι επιπλέον λειτουργίες του. Για τον σκοπό αυτό, χρησιμοποιείται το πεδίο επιλογών (options) του TCP, καθώς θεωρείται ο πλέον παραδοσιακός τρόπος υπόδειξης επιπρόσθετης λειτουργικότητας και παράλληλα, ο πιο συμβατός με το θεμελιώδες πρωτόκολλο [5].



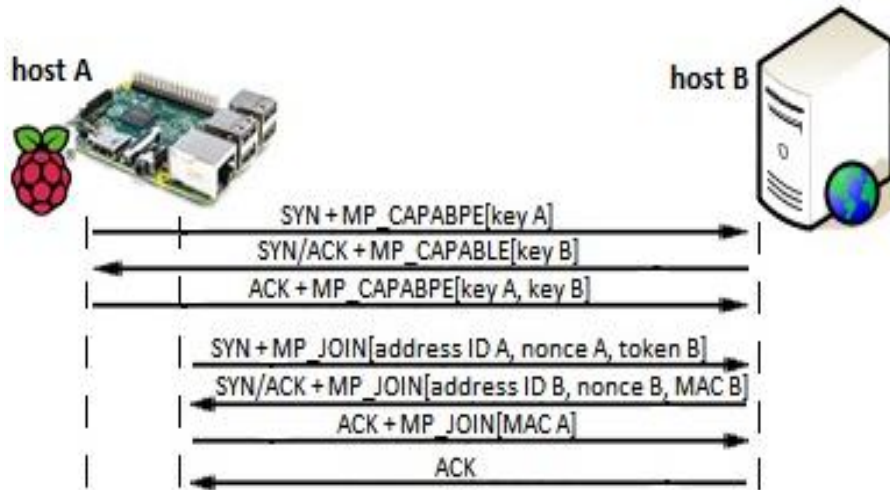
Σχήμα 1: Διαστρωμάτωση MPTCP

2.1.1. Τριμερής Χειραψία MPTCP

Όπως σημειώθηκε παραπάνω, η τριμερής χειραψία (3-way-handshake) του TCP χρησιμοποιείται ως μέσο τόσο για την εγκατάσταση νέων συνδέσεων από άκρο σε άκρο, όσο και για την διαπραγμάτευση επιπρόσθετης λειτουργικότητας ενσωματωμένη στο πεδίο επιλογών της επικεφαλίδας του. Έτσι, ένα τερματικό με ενεργοποιημένο το MPTCP χρειάζεται επίσης να χρησιμοποιήσει τον παραδοσιακό τρόπο συγχρονισμού, ώστε να ανακοινώσει την δυνατότητά αξιοποίησης πολλαπλών διαδρομών δρομολόγησης κατά την εγκατάσταση της πρωταρχικής ροής (primary flow) TCP. Τα πακέτα SYN, SYN/ACK και ACK της τριμερής χειραψίας συγχρονισμού πρέπει να είναι εφοδιασμένα με την επιλογή MP_CAPABLE, καθώς και επιπρόσθετες πληροφορίες πιστοποίησης/αυθεντικοποίησης με σκοπό την επιτυχή εγκατάσταση επιπλέον υπο-ροών.

Το *Σχήμα 2* απεικονίζει με σαφή τρόπο τον πρωταρχικό συγχρονισμό των εμπλεκόμενων άκρων σε συνδυασμό με την ανταλλαγή των πληροφοριών

ασφαλείας που απαιτούνται για την διασφάλιση της φερεγγυότητας της συνεδρίας, καθώς και την εγκατάσταση επιπρόσθετων υπο-ροών, η οποία ξεκινάει με την λήψη ενός πακέτου συγχρονισμού εφοδιασμένο με την επιλογή MP_JOIN.



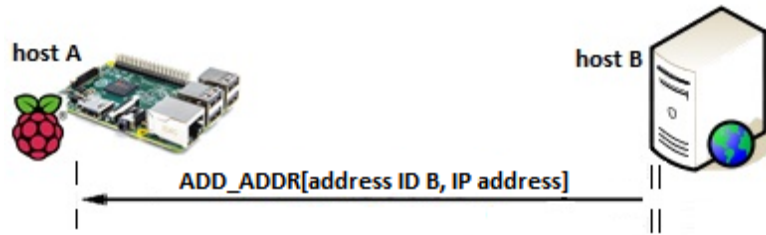
Σχήμα 2: Τριμερής χειραψία MPTCP και συγχρονισμός ροών

2.1.2. Διαχείριση Ροών TCP

Σε αυτό το σημείο, ο Διαχειριστής Διαδρομών (Path Manager) καθίσταται αρμόδιος για την προσθήκη και αφαίρεση δευτερευόντων ροών (secondary flows), ενώ ο Προγραμματιστής Πακέτων (Packet Scheduler) είναι υπεύθυνος για την χρησιμοποίησή τους [5]. Η λήψη ενός μηνύματος MP_JOIN σηματοδοτεί την παραγωγή ενός αναγνωριστικού για την σύνδεση (token) από το κλειδί του άκρου B, το οποίο αποστέλλεται από το άκρο A με σκοπό τον εντοπισμό-δήλωση της σύνδεσης MPTCP στην οποία επιθυμεί να ενταχθεί η υπο-ροή TCP. Η αποπολύπλεξη των μηνυμάτων συγχρονισμού γίνεται μέσω του token σε αντίθεση με το παραδοσιακό TCP, όπου χρησιμοποιείται η πόρτα προορισμού (destination port). Έπειτα, η 5-άδα: IP διεύθυνση πηγής, IP διεύθυνση προορισμού, πόρτα πηγής, πόρτα προορισμού, πρωτόκολλο, αντιστοιχίζεται στο κατάλληλο token και η αποπολύπλεξη των πακέτων πληροφορίας συνεχίζεται παραδοσιακά. Επίσης, ανταλλάσσονται κάποιες Ταυτότητες (Address IDs) μεταξύ των εμπλεκόμενων άκρων, οι οποίες αναφέρονται στην διεύθυνση πηγής των μηνυμάτων, ώστε η εγκατάσταση νέων συνδέσεων να μην επηρεάζεται από την παρουσία κόμβων μετάφρασης διευθύνσεων (NATs – Network Address Translation). Τέλος, δημιουργούνται οι Κωδικοί Αυθεντικοποίησης (Hash-based Message Authentication Code – HMAC) από τα αντίστοιχα κλειδιά σε συνδυασμό με τυχαίους αριθμούς (nonce) ως μέσο ασφάλισης των MP_JOIN μηνυμάτων και αποτροπής παράνομης εξωτερικής παρεμβολής σε εγκατεστημένες συνδέσεις [6].

Όμως, η ευρεία χρησιμοποίηση κόμβων μετάφρασης διευθύνσεων δημιουργούν δυσκολίες στην λήψη των μηνυμάτων συγχρονισμού σε

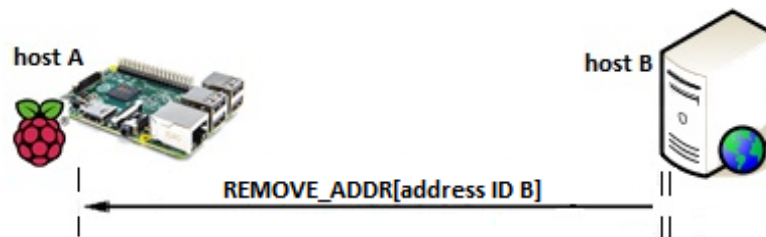
περιπτώσεις χρηστών εφοδιασμένων με μία δικτυακή επαφή, οι οποίοι βρίσκονται πίσω από NAT. Στο συγκεκριμένο σενάριο, απαιτείται η υπόδειξη των επιπρόσθετων διαθέσιμων διεπαφών από το δημόσιο προς το ιδιωτικό τερματικό. Η διαδικασία αυτή καθίσταται δυνατή μέσω της επιλογής ADD_ADDR, όπως απεικονίζεται και στο Σχήμα 3, η οποία επιτρέπει την διαφήμιση επιπλέον διευθύνσεων IP μέσω ήδη εγκατεστημένων υπο-ροών. Με αυτόν τον τρόπο, η συσκευή του ιδιωτικού δικτύου ενημερώνεται για τις πλεονάζουσες απομακρυσμένες διεπαφές και επιδιώκει η ίδια την εγκατάσταση νέων συνδέσεων TCP [7].



Σχήμα 3: Διαφήμιση διαθέσιμων διευθύνσεων IP

2.1.3. Τερματισμός Σύνδεσης MPTCP

Από την άλλη μεριά, η επιλογή REMOVE_ADDR επιτρέπει την αφαίρεση διευθύνσεων IP από την εκάστοτε σύνδεση MPTCP και οδηγεί κατ' επέκταση στον τερματισμό των αντίστοιχων υπο-ροών. Κατά τον ίδιο τρόπο, όπως φαίνεται και στο Σχήμα 4 που ακολουθεί, η συγκεκριμένη επιλογή δηλώνει την προς αφαίρεση διεύθυνση IP μέσω του address ID, με σκοπό την αποφυγή αλλοίωσης της διαδικασίας από κόμβους μετάφρασης διευθύνσεων.



Σχήμα 4: Αφαίρεση διευθύνσεων IP από την τρέχουσα σύνδεση MPTCP

Όπως και στην παραδοσιακή έκδοση του TCP, έτσι και στο Multipath TCP τα πακέτα FIN που σηματοδοτούν τον τερματισμό κάποιας TCP ροής αποστέλλονται ανεξάρτητα σε κάθε υπο-ροή. Με αυτόν τον τρόπο προσδίδεται μεγαλύτερη ευελιξία όσο αφορά την διαχείριση των πολλαπλών συνδέσεων από άκρο σε άκρο. Όταν, όμως, ένας κόμβος επιδιώκει τον τερματισμό της γενικότερης συνεδρίας MPTCP, τότε απαιτείται ένας ισοδύναμος μηχανισμός υψηλότερου επιπέδου. Το DATA_FIN [6], σε αυτήν την περίπτωση, υποδεικνύει την μη-παρουσία περαιτέρω όγκου δεδομένων προς αποστολή. Το DATA_FIN σηματοδοτείται ως επιλογή του Σήματος Ακολουθίας Δεδομένων (Data Sequence Signal), το οποίο θα αναλύσουμε στο υποκεφάλαιο 2.2. Ουσιαστικά, το DATA_FIN έχει την ίδια συμπεριφορά

με το αντίστοιχο TCP FIN, αλλά σε επίπεδο MPTCP αντί σε επίπεδο υπο-ροής TCP. Μόλις το DATA_FIN έχει αναγνωριστεί από ένα DATA_ACK στο στρώμα σύνδεσης MPTCP, τότε όλες οι ενεργές υπο-ροές πρέπει να τερματιστούν μέσω πρότυπων σημάτων FIN.

2.1.4. Συμβατότητα με το Σημερινό Internet

Ταυτόχρονα, το MPTCP παραμένει διαφανές στο Στρώμα Εφαρμογής, διατηρεί το πρότυπο TCP API (Application Programming Interface) και κατ' επέκταση την συμβατότητα με το παραδοσιακό TCP τόσο σε επίπεδο δικτύου όσο και σε επίπεδο εφαρμογής [5]. Η ανάγκη για προς-τα-πίσω συμβατότητα λογίζεται ως ζωτικής σημασίας δεδομένου ότι η επέκταση του πρωτοκόλλου βρίσκεται σε πρώιμο στάδιο και διάφοροι κόμβοι του διαδικτύου μπορεί να επαναγράψουν ή ολοκληρωτικά διαγράψουν τις άγνωστες επιλογές του MPTCP εντός των πεδίων της επικεφαλίδας του TCP. Σε αυτήν την περίπτωση, επιτακτική είναι η παρουσία μίας εφεδρικής σύνδεσης TCP ως μηχανισμό διασφάλισης της συνέχισης μεταφοράς δεδομένων, καθώς δυσλειτουργία μπορεί να παρουσιαστεί οποιαδήποτε χρονική στιγμή κατά την διάρκεια σύνδεσης.

Εναλλακτικοί οδοί έχουν ήδη εξεταστεί για την προστασία των σημάτων ελέγχου από διάφορους ενδιάμεσους διαδικτυακούς κόμβους. Μία εξ αυτών είναι η κωδικοποίηση των σημάτων αυτών και η ενθυλάκωσή τους εντός του ωφέλιμου φορτίου των πακέτων TCP υπό την μορφή TLV [8] (Type-Length-Value) [9]. Παρ' όλα αυτά, η σηματοδότηση μέσω επιλογών στην επικεφαλίδα του TCP έχει επικρατήσει για λόγους ομαλότητας και συμβατότητας με τις υπάρχουσες πρακτικές αποστολής σημάτων ελέγχου.

2.2. Επίπεδο Δεδομένων

Το Multipath TCP διαιρεί την ροή δεδομένων, όπως αυτή προέρχεται από το Στρώμα Εφαρμογής, και την δρομολογεί μέσω πολλαπλών υπο-ροών. Συνεπώς, απαιτούνται ιδιαίτεροι μηχανισμοί αντιστοίχισης των πακέτων πληροφορίας μεταξύ των επιπέδων MPTCP και TCP, καθώς και επανασυναρμολόγησης της πληροφορίας, η οποία προέρχεται από διαφορετικά μονοπάτια δρομολόγησης, πριν αυτή προωθηθεί ξανά στο Στρώμα Εφαρμογής. Ακόμα, η υλοποίηση ενός τροποποιημένου αλγόριθμου ελέγχου συμφόρησης είναι άκρως σημαντική για την αποδοτική αξιοποίηση του πολλαπλάσιου διαθέσιμου εύρους ζώνης.

2.2.1. Αντιστοίχιση Πακέτων Μεταξύ Επιπέδων TCP/MPTCP

Η ροή δεδομένων μπορεί να ελέγχεται και να επανασυναρμολογείται μέσω της Αντιστοίχισης Ακολουθίας Δεδομένων (Data Sequence Mapping) που

παρέχει το Σήμα Ακολουθίας Δεδομένων (Data Sequence Signal – DSS) [6], όπως παρουσιάζεται στο Σχήμα 5. Όπως κάθε άλλο σήμα ελέγχου, έτσι και το συγκεκριμένο βρίσκεται εντός του πεδίου επιλογών της επικεφαλίδας του TCP. Η Αντιστοίχιση Ακολουθίας Δεδομένων αποτελείται από τον Αριθμό Ακολουθίας Δεδομένων (Data Sequence Number - DSN), τον Αριθμό Ακολουθίας Υπο-ροής (Subflow Sequence Number - SSN) και το Μήκος (Data Length) για το οποίο η εκάστοτε αντιστοίχιση δεδομένων μεταξύ των επιπέδων TCP και MPTCP καθίσταται έγκυρη. Επίσης, υπάρχει ένα πεδίο Checksum, το οποίο χρησιμοποιείται για τον έλεγχο του ανάλογου ωφέλιμου φορτίου προς αντιστοίχιση. Ακόμα, το πεδίο Kind κατέχει σταθερά την τιμή '30', η οποία υποδηλώνει την υποστήριξη επεκταμένης λειτουργικότητας του πρωτοκόλλου TCP.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Kind				Length				Subtype				Reserved				F	m	M	a	A											
Data Acknowledgment (4 or 8 bytes, depending on flag <i>a</i>)																															
Data Sequence (4 or 8 bytes, depending on flag <i>m</i>)																															
Relative Subflow Sequence Number																															
Data-Length																Checksum															

Σχήμα 5: Επιλογή σήματος ακολουθίας δεδομένων (Subtype = 0x2)

Η παρουσία των αντίστοιχων σημαιών (flags) καθορίζουν κάθε φορά την σηματοδότηση πληροφοριών χαρτογράφησης ή/και αθροιστικής επιβεβαίωσης (acknowledgement) λήψης δεδομένων, την οποία, όμως, θα αναλύσουμε περαιτέρω στο επόμενο υποκεφάλαιο. Συγκεκριμένα:

- A = DATA_ACK
- a = DATA_ACK καταλαμβάνει 8 οκτάδες, διαφορετικά 4
- M = DSN, SSN, Data-Level Length, Checksum
- m = DSN καταλαμβάνει 8 οκτάδες, αλλιώς 4
- F = DATA_FIN

Όπως έχουμε ήδη αναφέρει, η Αντιστοίχιση Ακολουθίας Δεδομένων σκοπεύει στην επιτυχή επανασυναρμολόγηση της πληροφορίας προερχόμενης από πολλαπλές υπο-ροές δρομολόγησης, ώστε αυτή να παραδοθεί επιτυχώς στο Στρώμα Εφαρμογής. Αναλυτικότερα, το DSN αριθμεί μοναδικά όλα τα δεδομένα που στέλνονται ως μέρος της γενικότερης σύνδεσης MPTCP, ενώ το SSN λαμβάνει υπόψη μόνο τα πακέτα πληροφορίας της αντίστοιχης υπο-ροής TCP, έχοντας έτσι αποκλειστικά τοπική σημασία και επιτρέποντας την επαναχρησιμοποίησή του. Έτσι, με την λήψη ενός πακέτου από οποιαδήποτε υπο-ροή, αναζητείται η αντιστοίχιση του αύξοντος αριθμού του από το επίπεδο TCP στο στρώμα MPTCP. Σε αυτό το σημείο, αξίζει να σημειωθεί πως δεν καθίσταται απαραίτητη η αποστολή πληροφοριών αντιστοίχισης μαζί με κάθε πακέτο TCP, αφού οι συγκεκριμένες πληροφορίες αποστέλλονται κάθε φορά για ένα δεδομένο εύρος όγκου δεδομένων. Το γεγονός αυτό μπορεί να βοηθήσει στην αύξηση του ποσοστού του ωφέλιμου προς το

συνολικό φορτίο σημαντικού αριθμού πακέτων, ενώ παράλληλα, να αντισταθμίσει την ανάγκη τεμαχισμού και αποστολής ενός DSS σε πολλαπλά τμήματα εξαιτίας της περιορισμένης χωρητικότητας του πεδίου επιλογών του TCP.

2.2.2. Επιβεβαίωση Λήψης Δεδομένων (ACKs)

Όπως αναφέρθηκε προηγουμένως, το MPTCP έχει την δυνατότητα, μέσω του Σήματος Ακολουθίας Δεδομένων, επιβεβαίωσης λήψης δεδομένων (ACKs) τόσο σε επίπεδο υπο-ροής TCP, όσο και σε επίπεδο σύνδεση MPTCP. Η συγκεκριμένη δυνατότητα προσδίδει επιπρόσθετη ανθεκτικότητα σε περίπτωση δυσλειτουργίας σε οποιοδήποτε σημείο του δικτύου διατηρώντας την βάση του παραδοσιακού μηχανισμού επαλήθευσης μεταφοράς δεδομένων.

Η εισαγωγή δύο επιπέδων επαλήθευσης λήψης πακέτων πληροφορίας παρέχει ανεξαρτησία στην διαχείριση της εισερχόμενης διαδικτυακής κίνησης μεταξύ των πολλαπλών υπο-ροών, καθώς και συμβατότητα με το παραδοσιακό TCP. Συγκεκριμένα, η αποστολή πακέτων επιβεβαίωσης (ACK) στο επίπεδο TCP έχει ως ρόλο μία πρόιμη επαλήθευση λήψης των κομματιών πληροφορίας με απώτερο σκοπό την αναμετάδοση χαμένων τμημάτων. Από την άλλη μεριά, η αθροιστική επιβεβαίωση λήψης δεδομένων (DATA_ACK), η οποία λαμβάνει χώρα στο επίπεδο MPTCP, αντικαθιστά το αντίστοιχο, ισοδύναμο ACK της παραδοσιακής έκδοσης του πρωτοκόλλου. Το DATA_ACK υποδεικνύει τον επόμενο αύξων αριθμό πακέτου που αναμένεται, δηλαδή υποδηλώνει τον συνολικό όγκο δεδομένων που έχει ληφθεί επιτυχώς από το αντίστοιχο τερματικό [6].

Η διπλή επαλήθευση λήψης των πακέτων πληροφορίας σε συνδυασμό με τα δύο επίπεδα αυξόντων αριθμών δίνει στο MPTCP τη δυνατότητα για αναμετάδοση χαμένων τμημάτων μέσω διαφορετικών υπο-ροών από την αρχική. Με αυτόν τον τρόπο, το πρωτόκολλο επιτυγχάνει έναν από τους σχεδιαστικούς του στόχους: την μετακίνηση της ροής πληροφορίας μακριά από μονοπάτια που επιφέρουν απώλειες πακέτων σε περισσότερο αξιόπιστες διαδρομές δρομολόγησης. Γι' αυτό το λόγο, το εκάστοτε σύστημα αποστολής δεδομένων απαιτείται να αποθηκεύει προσωρινά τα πακέτα πληροφορίας έως ότου λάβει επιβεβαίωση λήψης τόσο στο επίπεδο TCP, όσο και στο επίπεδο σύνδεσης MPTCP [6]. Με αυτόν τον τρόπο, το πρωτόκολλο αποκτά καλύτερη γνώση και έλεγχο των διαφόρων συνδέσεων, επιτρέποντάς του κατ' επέκταση την κατάλληλη προσαρμογή του στην δυναμικά μεταβαλλόμενη κατάσταση των εμπλεκόμενων ζεύξεων.

2.2.3. Απαίτηση Χωρητικότητας TCP Buffers

Το διαφημιζόμενο παράθυρο λήψης δεδομένων (receive window) χρησιμοποιείται από το TCP ως ένας δευτερεύον μηχανισμός ελέγχου της ροής πληροφορίας πέρα της υλοποίησης κάποιου αλγόριθμου ελέγχου

συμφόρησης από την πλευρά του αποστολέα. Όπως το παραδοσιακό TCP, έτσι και το MPTCP εφαρμόζει ένα ενιαίο παράθυρο λήψης δεδομένων στο επίπεδο σύνδεσης, το οποίο επιτρέπει την συνέχιση της ροής δεδομένων βασιζόμενο στα DATA_ACKs [6]. Η συγκεκριμένη επιλογή βασίζεται στο γεγονός χρησιμοποίησης πολλαπλών διαδρομών δρομολόγησης των πακέτων πληροφορίας. Η υλοποίηση ξεχωριστών παραθύρων λήψης δεδομένων ανά υπο-ροή TCP θα μπορούσε να οδηγήσει σε περαιτέρω καθυστέρηση προώθησης της πληροφορίας στο Στρώμα Εφαρμογής εξαιτίας της πιο αργής εξέλιξης μεμονωμένων παραθύρων λήψης ως αποτέλεσμα του χαμηλότερου ρυθμού μετάδοσης.

Μόλις κάποιο ληφθέν πακέτο κριθεί πως βρίσκεται εντός του επιτρεπτού παραθύρου λήψης, και σε περίπτωση που ο αύξων αριθμός του δεν συμβαδίζει με τον αναμενόμενο, τότε χρειάζεται να αποθηκευτεί σε έναν προσωρινό χώρο αναμονής (buffer) με σκοπό την μελλοντική χωρική αναδιάταξη των πακέτων και την προώθησή τους στο Στρώμα Εφαρμογής. Επομένως, το μέγεθος του buffer είναι ένα άκρως σημαντικό ζητούμενο, καθώς θα πρέπει να κατέχει την δυνατότητα αποθήκευσης πιθανόν μεγάλου αριθμού πακέτων με στόχο την αποφυγή απόρριψής τους από το εκάστοτε σύστημα, κάτι που θα σηματοδοτούσε συνεχείς αναμεταδόσεις και σπατάλη εύρους ζώνης.

Στην περίπτωση του απλού TCP το συνιστούμενο μέγεθος buffer ισούται με το διπλάσιο του γινομένου Εύρους Ζώνης-Καθυστέρησης (Bandwidth-Delay Product – BDP) [5]. Ένα BDP προορίζεται για την δυνατότητα αναδιάταξης των εκτός αναμενόμενης σειράς ληφθέντων πακέτων, ενώ το 2^ο BDP επιτρέπει την συνέχιση ανταλλαγής πληροφοριών κατά το στάδιο Γρήγορης Αναμετάδοσης (Fast Retransmit) [10] του αλγορίθμου ελέγχου συμφόρησης του TCP. Από την άλλη μεριά, η σχεδίαση του MPTCP απαιτεί την κατανομή περισσότερου αποθηκευτικού χώρου για την εξυπηρέτηση των αναγκών των πολλαπλών ροών TCP. Η απουσία επαρκή χώρου αναμονής πακέτων σε συνδυασμό με την απώλεια και αναμετάδοση τμημάτων πληροφορίας σε ορισμένες υπο-ροές θα μπορούσε να επηρεάσει αρνητικά την απόδοση των υγιών ροών TCP. Έτσι, υπό συνθήκες χειρότερου σεναρίου ο ελάχιστος χώρος προσωρινής αποθήκευσης δεδομένων κρίνεται πως χρειάζεται να ισούται με: $\sum(BW_i) * RTO_{max}$, όπου BW_i το διαθέσιμο εύρος ζώνης της i -οστής ροής TCP και RTO_{max} το μέγιστο Χρονικό Όριο Αναμετάδοσης (Retransmission Time-Out) μεταξύ όλων των υπο-ροών. Οι συγκεκριμένες προδιαγραφές επιτρέπουν στον παραλήπτη την προσωρινή αποθήκευση δεδομένων σε περίπτωση που πυροδοτηθεί RTO στην ζεύξη μικρότερης απόδοσης. Παρ' όλα αυτά, η παραπάνω υλοποίηση θεωρείται αρκετά ακριβή τόσο σε οικονομικούς όσο και σε πρακτικούς όρους. Έτσι, δεδομένης της αποφυγής time-outs το προτεινόμενο μέγεθος των αποθηκευτικών χώρων λήψης ανέρχεται σε $2 * \sum(BW_i) * RTT_{max}$ [5], το οποίο είναι το αντίστοιχο ισοδύναμο του παραδοσιακού TCP σε περιόδους γρήγορης αναμετάδοσης πακέτων πληροφορίας.

Αντίστοιχες προδιαγραφές, φυσικά, χρειάζεται να ισχύουν και για τον αποστολέα [5]. Όπως τονίσαμε στην αμέσως προηγούμενη ενότητα, η πηγή των πακέτων πληροφορίας οφείλει να αποθηκεύει προσωρινά τα

απεσταλμένα δεδομένα μέχρι να ληφθούν οι ανάλογες επιβεβαιώσεις τόσο σε επίπεδο TCP, όσο και σε επίπεδο MPTCP. Επομένως, και σε αυτήν την περίπτωση απαιτείται ένας buffer μεγέθους $2 * \sum(BW_i) * RTT_{max}$ για την εξυπηρέτηση του αυξημένου αριθμού εξερχόμενων πακέτων.

2.2.4. Ενοποιημένος Αλγόριθμος Ελέγχου Συμφόρησης

Ο κλασικός αλγόριθμος ελέγχου συμφόρησης που εφαρμόζεται από το TCP αποτελείται από μία εκθετικά αυξανόμενη φάση Αργής Εκκίνησης (Slow Start), μία γραμμικά μεταβαλλόμενη φάση Αποφυγής Συμφόρησης (Congestion Avoidance) και πολλαπλασιαστική μείωση του παραθύρου συμφόρησης σε περίπτωση γεγονότος απώλειας πακέτων [10]. Επιτρέποντας στις υπο-ροές μίας σύνδεσης πολλαπλών διαδρομών την υλοποίηση ανεξάρτητων ελέγχων συμφόρησης βασισμένων αποκλειστικά στον μηχανισμό του παραδοσιακού TCP, θα προσέδιδε σε μία σύνδεση MPTCP μεγαλύτερο ποσοστό του διαθέσιμου εύρους ζώνης σε περίπτωση ανταγωνισμού με μία απλή σύνδεση TCP. Το γεγονός αυτό θα καταπάταγε έναν από τους σχεδιαστικούς στόχους του επεκταμένου πρωτοκόλλου, ο οποίος αποσκοπεί στον δίκαιο διαμοιρασμό των διαθέσιμων πόρων μεταξύ των ροών μίας σύνδεσης MPTCP και μίας κλασικής συνεδρίας TCP. Από την άλλη μεριά, η εφαρμογή ενός μοναδικού παραθύρου συμφόρησης στο επίπεδο MPTCP θα απέτρεπε μία σύνδεση πολλαπλών ροών να αξιοποιήσει επιτυχώς όλο το διαθέσιμο εύρος ζώνης και να επιτύχει αποτελεσματική εκφόρτωση δεδομένων μεταξύ των εμπλεκόμενων διαδρομών.

Προκειμένου το Multipath TCP να επιτύχει τους σχεδιαστικούς του στόχους, καθώς και αποτελεσματική συγκέντρωση πόρων (“resource pooling”) [11], κάθε υπο-ροή χρειάζεται να διατηρεί την δικιά της κατάσταση συμφόρησης κάτω από την ομπρέλα ενός ενοποιημένου μηχανισμού ελέγχου συμφόρησης [3]. Ο Αλγόριθμος Ενοποιημένης Αύξησης (Linked Increase Algorithm – LIA) τροποποιεί μόνο το στάδιο Αποφυγής Συμφόρησης υπαγορεύοντας την επιτρεπτή αύξηση του παραθύρου συμφόρησης (congestion window – cwnd) κατά την λήψη ενός πακέτου ACK. Ως αποτέλεσμα, ο τροποποιημένος αλγόριθμος, όπως παρουσιάζεται παρακάτω, επιτρέπει στο Multipath TCP να επιτύχει αποτελεσματική κατανομή πόρων, ενώ, παράλληλα, τείνει να εξανεμίσει τον ανταγωνισμό μεταξύ συνδέσεων MPTCP και TCP διαμοιράζοντας δίκαια το διαθέσιμο εύρος ζώνης.

$$cwnd_i = cwnd_i + \left[\min \left(\frac{\alpha}{cwnd_{total}}, \frac{1}{cwnd_i} \right) \right]$$

$$\alpha = \frac{\max(\frac{cwnd_i}{RTT_i^2})}{\left[\sum \left(\frac{cwnd_i}{RTT_i} \right) \right]^2} * cwnd_{total}$$

Βάσει των παραπάνω, ο αλγόριθμος τείνει να αυξήσει το παράθυρο συμφόρησης της εκάστοτε υπο-ροής σύμφωνα με την στιγμιαία τιμή του

cwnd. Ως αποτέλεσμα, ροές TCP με αυξημένο cwnd θα δεχθούν μεγαλύτερη διεύρυνση του παραθύρου τους, αφού ένα ήδη ευρύ παράθυρο συμφόρησης συνεπάγεται αποδοτικότερη μεταφορά δεδομένων μέσω της αντίστοιχης διαδρομής. Με αυτόν τον τρόπο, το MPTCP δύναται να μεταφέρει την δικτυακή κίνηση μακριά από μη-αποδοτικά ή/και αναξιόπιστα μονοπάτια ευνοώντας διαδρομές με μικρότερο RTT (Round-Trip Time). Παρ' όλα αυτά, ένα μικρό μέρος της κίνησης είναι απαραίτητο να δρομολογείται μέσω διαδρομών χαμηλότερης απόδοσης με απώτερο σκοπό την συνεχή ανίχνευση της κατάστασης συμφόρησης των ζεύξεων και την κατ' επέκταση άμεση ενημέρωση της πηγής πληροφορίας σε περίπτωση ανάγκης εκ νέου αλλαγής της διαδρομής δρομολόγησης [12].

Όπως έχει ήδη αναφερθεί, ένας από τους σχεδιαστικούς στόχους του Multipath TCP αποτελεί η δίκαιη κατανομή του διαθέσιμου εύρους ζώνης σε περίπτωση ανταγωνισμού μεταξύ μίας σύνδεσης MPTCP και μίας κλασσικής σύνδεσης TCP. Πιο συγκεκριμένα, μία σύνδεση TCP πολλαπλών διαδρομών δεν πρέπει να λαμβάνει μεγαλύτερο ποσοστό των διαθέσιμων δικτυακών πόρων συγκριτικά με μία απλή συνεδρία TCP, όταν υπάρχει ταυτόχρονος ανταγωνισμός. Η παράμετρος α περιγράφει την «επιθετικότητα» των πολλαπλών διαδρομών και έχει ως σκοπό την ελαχιστοποίησή της όταν τίθεται υπό κίνδυνο ο προαναφερθής σχεδιαστικός στόχος του πρωτοκόλλου. Γι' αυτό το λόγο, η αύξηση του παραθύρου συμφόρησης της εκάστοτε υποροής είναι το πολύ ίση με αυτή του παραδοσιακού TCP: $(1/cwnd)$.

Επιπρόσθετα, όπως είναι λογικό, κάτω από συνθήκες χειρότερου σεναρίου το Multipath TCP οφείλει να είναι τουλάχιστον τόσο αποδοτικό όσο θα ήταν μία κλασσική σύνδεση TCP κατά μήκος της καλύτερης εκ των διαθέσιμων διαδρομών. Αντίθετα, εν απουσία οποιασδήποτε ανταγωνιστικής διαδικτυακής κίνησης η απόδοση του πρωτοκόλλου πρέπει να ισούται με το αθροιστικό εύρος ζώνης των εμπλεκόμενων ζεύξεων.

Ως ενισχυμένη, εναλλακτική εκδοχή του LIA παρουσιάζεται ο Αλγόριθμος Ευκαιριακής Ενοποιημένης Αύξησης (Opportunistic Linked Increase Algorithm – OLIA) [13]. Η βασική τροποποίηση που καθιστά τον OLIA επικρατέστερο του προκατόχου του είναι η καλύτερη προσαρμογή και βέλτιστη σύγκλιση του στις απαιτήσεις που προκύπτουν λόγω της δυναμικά μεταβαλλόμενης κατάστασης συμφόρησης του δικτύου. Ως αποτέλεσμα, ο σχεδιασμός του OLIA ενισχύει την ανταπόκριση του εκάστοτε έξυπνου τερματικού επιτρέποντας του την ακόμη αποτελεσματικότερη συγκέντρωση πόρων και εκφόρτωση δεδομένων μεταξύ των διαθέσιμων διαδρομών.

2.3. Λειτουργίες του MPTCP

Το Multipath TCP κατέχει την ικανότητα διαίρεσης της ροής πληροφορίας και της ταυτόχρονης δρομολόγησής της μέσω πολλαπλών διαδρομών. Επίσης, όπως αναλύσαμε στην προηγούμενη ενότητα, ο ενοποιημένος αλγόριθμος συμφόρησης δίνει την δυνατότητα στην εκάστοτε έξυπνη συσκευή να επιτύχει αποτελεσματική εκφόρτωση δεδομένων μεταξύ των

διαθέσιμων μονοπατιών δρομολόγησης ανακουφίζοντας με αυτόν τον τρόπο το εκάστοτε υποδίκτυο και προσφέροντας ενισχυμένη ποιότητα υπηρεσιών (Quality of Service – QoS). Ακόμα, οι χρήστες είναι σε θέση να επωφεληθούν από αυξημένους ρυθμούς μετάδοσης δεδομένων λόγω της συσσωμάτωσης εύρους ζώνης από τις πολλαπλές διαθέσιμες διεπαφές.

Το MPTCP επιτυγχάνει τα παραπάνω δημιουργώντας ένα πλήρες πλέγμα υπο-ροών [14] μεταξύ όλων των διαθέσιμων διεπαφών. Με αυτόν τον τρόπο, επιτυγχάνεται μέγιστη συσσωμάτωση εύρους ζώνης και ενισχυμένη ανθεκτικότητα σε περίπτωση δυσλειτουργίας οποιασδήποτε συναφούς υπο-ροής. Σε ενδεχόμενο διαταραχής της λειτουργίας κάποιας ροής TCP ή γενικότερα βλάβης κάποιας εκ των ενεργών διεπαφών, τότε ακολουθεί ο τερματισμός των αντίστοιχων υπο-ροών και παύει η μετάδοση δεδομένων. Όλα τα χαμένα τμήματα πληροφορίας προγραμματίζονται προς αναμετάδοση μέσω των υγείων διαδρομών αποτρέποντας έτσι την πρόκληση αισθητών επιπτώσεων στην ποιότητα εμπειρίας του χρήστη.

Επιπρόσθετα, το MPTCP είναι εφοδιασμένο με την επιλογή MP_PRIO [6] μέσω της οποίας καθίσταται δυνατή η διάκριση των διαφόρων υπο-ροών ως ενεργών ή εφεδρικών. Αναλυτικότερα, η σηματοδότηση της επιλογής MP_PRIO μέσω κάποιας ροής TCP ενημερώνει το απομακρυσμένο τερματικό περί την μη χρησιμοποίηση της συγκεκριμένης ζεύξης για μεταφορά δεδομένων με το πέρας της εγκατάστασης της σύνδεσης, αλλά για την χρήση της ως εφεδρική ζεύξη [14] σε περίπτωση μη εναλλακτικής διαδρομής δρομολόγησης.

Σε περιβάλλοντα κινητών τηλεπικοινωνιών, το παραδοσιακό TCP στερείται της δυνατότητας επίτευξης ομαλής μετάβασης μεταξύ των διαφόρων διαθέσιμων ασύρματων τεχνολογιών και κυψελών κινητής τηλεφωνίας. Συνήθως, απαιτείται επανασυγχρονισμός των εμπλεκόμενων άκρων, το οποίο συνεπάγεται νέα τριμερή χειραψία για τον εκ νέου συγχρονισμό της εκάστοτε σύνδεσης. Η επιπλέον σηματοδότηση αναβάλλει την μεταφορά δεδομένων καθιστώντας την παροχή υπηρεσιών προβληματική.

Εναλλακτικές προσεγγίσεις για την αντιμετώπιση των διαταραχών της σύνδεσης κατά την διάρκεια διαπομπών έχουν ήδη εξεταστεί. Το πρωτόκολλο Mobile IP [15] αποτελεί μία εξ αυτών. Υποστηρίζει σενάρια κινητικότητας επιτρέποντας την διαφανή εναλλαγή διευθύνσεων IP από ένα κινητικό τερματικό με την βοήθεια δύο ενδιάμεσων κόμβων (home/foreign agent) προσωρινής αποθήκευσης δεδομένων. Παρ' όλα αυτά, η χρήση του στις υλοποιήσεις λογισμικού έξυπνων κινητών τηλεφώνων είναι σπάνια. Αντίθετα, το Multipath TCP, ως επέκταση του κλασσικού TCP, μπορεί να συμβάλλει στην επίτευξη τόσο οριζόντιας όσο και κάθετης αδιάκοπης διαπομπής χωρίς την ανάγκη παρεμβολής βοηθητικών κόμβων για την προσωρινή αποθήκευση δεδομένων ή προώθηση σημάτων ελέγχου. Το επεκταμένο πρωτόκολλο, επιτρέποντας την ταυτόχρονη αξιοποίηση πολλαπλών διαδρομών, κατέχει την δυνατότητα εγκατάστασης μίας νέας υπο-ροής πριν την απόσπαση της πρωταρχικής σύνδεσης [4]. Έτσι, έξυπνα κινητά τηλέφωνα μπορούν να υποστηρίξουν επιτυχώς σενάρια κινητικότητας αποφεύγοντας την υποβάθμιση υπηρεσιών εξαιτίας απρόσμενων

τερματισμών συνεδριών εν απουσία εναλλακτικών οδών δρομολόγησης (break-before-make event) [4].

Σε ένα πιο ρεαλιστικό σενάριο, η δυνατότητα εγκατάστασης εφεδρικών υποροών θα μπορούσε να συμβάλει στην επίτευξη οριζόντιων και κάθετων διαπομπών φιλικά ως προς τα προγράμματα παροχής δεδομένων που προσφέρονται από τους παρόχους υπηρεσιών Internet (Internet Service Providers – ISPs). Πιο συγκεκριμένα, η χρήση εφεδρικών συνδέσεων μέσω των δικτύων 4G/3G σε περίπτωση πλήρους εξάλειψης του σήματος Wi-Fi θα μπορούσε να αποτελέσει μία άκρως αποτελεσματική λύση για την επίτευξη αδιάκοπης κάθετης διαπομπής, η οποία συγκλίνει στην σημερινή στατική πολιτική των έξυπνων τηλεφώνων: «Χρησιμοποίησε το δίκτυο 4G/3G, εκτός αν το Wi-Fi είναι διαθέσιμο».

2.4. Διακομιστής Μεσολάβησης MPTCP (Proxy)

Επί του παρόντος, ένα έξυπνο τερματικό μπορεί να επωφεληθεί από τα πλεονεκτήματα που προσφέρει το Multipath TCP μόνο αν στο απομακρυσμένο άκρο ανταλλαγής πληροφοριών είναι επίσης εγκατεστημένο το επεκταμένο πρωτόκολλο. Προφανώς, αυτή η απαίτηση δημιουργεί φραγμούς στην υιοθέτηση του πρωτοκόλλου από πλατφόρμες παροχής υπηρεσιών, αφού τέτοιου είδους αναβαθμίσεις είναι συχνά δύσκολο να διεξαχθούν σε διακομιστές ή/και τερματικά εξυπηρέτησης μεγάλου ποσοστού διαδικτυακής κίνησης. Αντίθετα, έξυπνες συσκευές, tablets και κινητά τηλέφωνα, όπως το iPhone της Apple [16], είναι πιο εύκολο να ενσωματώσουν την επέκταση του TCP λόγω της λήψης συχνών ενημερώσεων λογισμικού. Η εισαγωγή ενός μετατροπέα πρωτοκόλλου υπό την μορφή ενός διακομιστή μεσολάβησης μεταξύ των εμπλεκόμενων άκρων θα βοηθούσε στην καταπολέμηση του πρωταρχικού αυτού προβλήματος.

Ο Διακομιστής Μεσολάβησης TCP Πολλαπλών Διαδρομών (MPTCP Proxy) έχει την δυνατότητα παροχής τεχνικής υποστήριξης εκ μέρους της πλευράς της σύνδεσης με εγκατεστημένη την παραδοσιακή έκδοση του TCP μέσω μετατροπής πρωτοκόλλου. Ο Proxy είναι αρμόδιος για την ανίχνευση και την διαπραγμάτευση της δυνατότητας χρησιμοποίησης του MPTCP και την τελική διάσπαση της σύνδεσης MPTCP/TCP, όπως φαίνεται στο Σχήμα 6.



Σχήμα 6: Διάταξη διακομιστή μεσολάβησης TCP πολλαπλών διαδρομών

Επίσης, ο MPTCP Proxy είναι υπεύθυνος για την τροποποίηση και αντιστοίχιση των πακέτων πληροφορίας μεταξύ των συνεδριών MPTCP και TCP, ενώ, όπως είναι λογικό, οφείλει να διατηρεί ταυτοχρόνως δύο

ξεχωριστούς μηχανισμούς ελέγχου συμφόρησης για κάθε μία εκ των συνδέσεων.

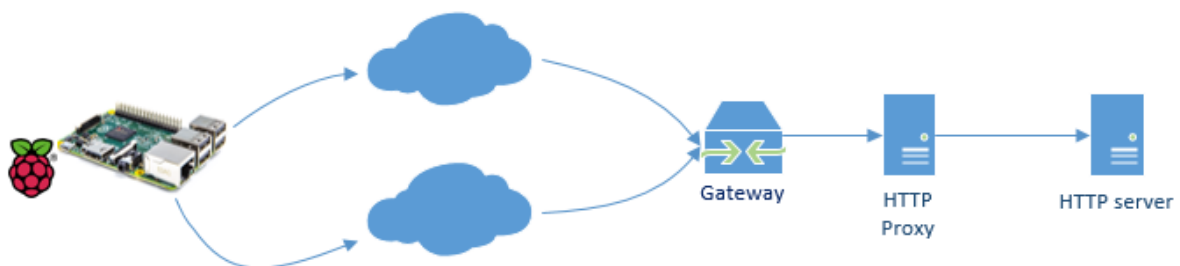
2.4.1. Οφέλη του MPTCP Proxy

Η υλοποίηση ενός Διακομιστή Μεσολάβησης TCP Πολλαπλών Διαδρομών θα μπορούσε να συνδράμει στην προσπάθεια των παρόχων υπηρεσιών Internet για αποτελεσματικότερη διαχείριση και χρησιμοποίηση των διαθέσιμων διαδικτυακών πόρων. Πιο συγκεκριμένα, η εφαρμογή μίας τέτοιας διάταξης θα ήταν εξαιρετικά χρήσιμη σε περιβάλλοντα ασύρματης πρόσβασης ως μέσο αποδοτικότερης διαχείρισης των απαιτήσεων του συνεχώς αυξανόμενου αριθμού έξυπνων συσκευών. Τερματικά εφοδιασμένα με πολλαπλές ραδιοηλεκτρικές διεπαφές θα έχουν την δυνατότητα ταυτόχρονης αξιοποίησής τους, κάτι που θα ενίσχυε την μέχρι σήμερα πολλές φορές ανεπαρκή αντοχή τους σε σενάρια κινητικότητας λόγω της απουσίας εναλλακτικών μονοπατιών δρομολόγησης δεδομένων. Έτσι, οι τελικοί χρήστες θα είναι σε θέση να βιώσουν ενισχυμένη ποιότητα εμπειρίας και να επιτύχουν αποδοτικότερη αξιοποίηση του διαθέσιμου εύρους ζώνης.

Επιπρόσθετα, προσφέρεται στους διαχειριστές δικτύων η δυνατότητα να επιτύχουν συγκέντρωση πόρων από διαφορετικές ασύρματες τεχνολογίες πρόσβασης. Με αυτόν τον τρόπο, καθίσταται δυνατή η δυναμική εκφόρτωση δεδομένων από αδειοδοτημένα σε μη-αδειοδοτημένα κομμάτια του φάσματος ραδιοσυχνοτήτων με απώτερο σκοπό την ανακούφιση του δικτύου του παρόχου υπηρεσιών από τυχόν καταστάσεις συμφόρησης. Ακόμα, ο MPTCP Proxy, ως ενδιάμεσος κόμβος, δίνει την ευκαιρία για εφαρμογή περαιτέρω πολιτικών ελέγχου της κίνησης δεδομένων με στόχο την μείωση των δαπανών χρήσης τόσο για το ίδιο το δίκτυο όσο και για τον τελικό χρήστη [17].

2.4.2. Είδη MPTCP Proxy

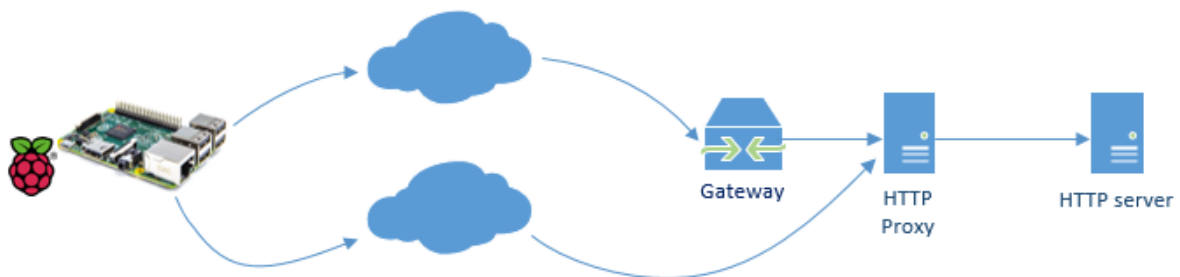
Η επιτυχία της διαδικασίας μετατροπής πρωτοκόλλου που εισάγει ο MPTCP Proxy εξαρτάται σε μεγάλο βαθμό από το σημείο εγκατάστασης του μηχανισμού. Υπάρχουν δύο βασικά μοντέλα: διαφανής παρεμβολή και παρεμβολή εκτός της προκαθορισμένης διαδρομής δρομολόγησης [18].



Σχήμα 7: Παρεμβολή διαφανή MPTCP Proxy

Το Σχήμα 7 της προηγούμενης σελίδας απεικονίζει με απλοϊκό τρόπο την διάταξη ενός διαφανή MPTCP Proxy, ο οποίος παρεμβάλλεται κατά μήκος της προεπιλεγμένης διαδρομής δρομολόγησης. Σε αυτό το σενάριο, το άκρο που επιδιώκει την έναρξη μίας συνεδρίας TCP με την υποστήριξη πολλαπλών διαδρομών δεν απαιτείται να είναι ενήμερο για την παρουσία του μετατροπέα πρωτοκόλλου. Το εκάστοτε τερματικό ξεκινάει την διαδικασία εγκατάστασης σύνδεσης μέσω της τριμερούς χειραψίας με την απομακρυσμένη πλατφόρμα παροχής υπηρεσιών. Διάφανα ως προς και τα δύο εμπλεκόμενα άκρα, ο διακομιστής μεσολάβησης αναλαμβάνει χρέη μετατροπέα πρωτοκόλλου. Έτσι, το έξυπνο τερματικό απολαμβάνει τα πλεονεκτήματα του επεκταμένου πρωτοκόλλου με τις υπο-ροές TCP να τερματίζονται στον διακομιστή μεσολάβησης, απ' όπου ξεκινά η παραδοσιακή σύνδεση TCP με προορισμό τον απομακρυσμένο server εξυπηρέτησης. Ουσιαστικά, τα δύο τερματικά άκρα δρομολογούν τα πακέτα πληροφορίας με βάση την ψευδαίσθηση της άμεσης επικοινωνίας, ενώ στην πραγματικότητα ο MPTCP Proxy επιθεωρεί, τροποποιεί και επαναδρομολογεί τα πακέτα ανάλογα με την δυναμικότητα του αντίστοιχου προορισμού.

Από την άλλη μεριά, όπως απεικονίζεται στο Σχήμα 8, όταν ο διακομιστής μεσολάβησης παρεμβάλλεται εκτός της προκαθορισμένης διαδρομής δρομολόγησης των πακέτων πληροφορίας, τότε απαιτείται ρητή σηματοδότηση για την εισαγωγή του MPTCP Proxy στην προς εγκατάσταση συνεδρία. Η ανακατεύθυνση των υπο-ροών TCP ενημερώνει ταυτόχρονα το άκρο αφετηρίας της σύνδεσης για την παρουσία του μηχανισμού. Σε αυτήν την περίπτωση, η συσκευή εγκαθιστά άμεση επικοινωνία με τον μετατροπέα πρωτοκόλλου, ο οποίος καθίσταται υπεύθυνος για την εγκατάσταση της κλασικής σύνδεσης TCP με το απομακρυσμένο τερματικό και την αναμετάδοση των πακέτων.



Σχήμα 8: Παρεμβολή MPTCP Proxy εκτός προκαθορισμένης διαδρομής

Σε κάθε σενάριο, αν και τα δύο τερματικά άκρα έχουν την δυνατότητα εγκατάστασης και αξιοποίησης πολλαπλών διαδρομών δρομολόγησης, τότε ο διακομιστής μεσολάβησης παύει να κατέχει τον ρόλο μετατροπέα πρωτοκόλλου. Αντίθετα, λειτουργεί ως απλός δικτυακός κόμβος αναμετάδοσης-προώθησης πακέτων, χωρίς την ευθύνη εκπόνησης περαιτέρω λειτουργιών σε υψηλότερο επίπεδο του TCP/IP stack.

3. Έλεγχος Διαθέσιμου Εύρους Ζώνης σε Linux Kernel

Το Internet, όπως το βιώνουμε σήμερα, στοχεύει στην παροχή υπηρεσιών υπό συνθήκες βέλτιστης προσπάθειας. Η διαδικτυακή κίνηση επεξεργάζεται το ταχύτερο δυνατόν και προωθείται από τους ενδιάμεσους κόμβους δίχως την διασφάλιση βέλτιστης μετάδοσης ανεξαρτήτως της κατάστασης του εκάστοτε διασχιζόμενου δικτύου. Ενώ τα δίκτυα υπολογιστών επεκτείνονται συνεχώς, η ανάγκη για την αποδοτικότερη αξιοποίηση του διαθέσιμου εύρους ζώνης παύει να αποτελεί το αποκλειστικό ζήτημα προς επίλυση από τους ISPs και τους διαχειριστές δικτύων. Αντιθέτως, καθίσταται επιτακτική η ανάγκη για περαιτέρω μελέτη σεναρίων καθυστέρησης και αξιοπιστίας των παρεχόμενων υπηρεσιών.

Το TCP/IP stack με την παρούσα μορφή του αποδεικνύεται πως δεν κατέχει την δυνατότητα αποδοτικής αξιοποίησης του διαθέσιμου εύρους ζώνης και διασφάλισης της ποιότητας υπηρεσιών. Από την άλλη μεριά, το Multipath TCP είναι σε θέση να προσφέρει ενισχυμένη ανθεκτικότητα σε τυχόν δυσλειτουργίες κατά μήκος των διαδρομών δρομολόγησης μέσω της παράλληλης χρησιμοποίησης πολλαπλών ζεύξεων και της δυναμικής εκφόρτωσης δεδομένων. Παρ' όλα αυτά, η πλήρης αξιοποίηση της δυναμικότητας του επεκταμένου πρωτοκόλλου αποκλίνει από τα υπάρχοντα σχήματα πολιτικών δρομολόγησης. Η ταυτόχρονη, συνεχόμενη, πλήρης αξιοποίηση των διαθέσιμων διαδικτυακών διεπαφών ενός τερματικού απαιτεί την αντίστοιχη αναθεώρηση των σημερινών πολιτικών δρομολόγησης σε περισσότερο δυναμικές τεχνικές ελέγχου της διαδικτυακής κίνησης.

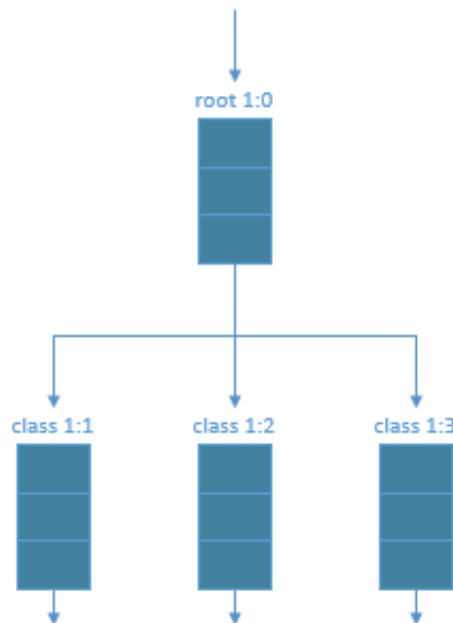
Σε αυτό το κεφάλαιο, θα θέσουμε τις βάσεις για την εφαρμογή μίας τεχνικής «στραγγαλισμού» του διαθέσιμου εύρους ζώνης (bandwidth throttling) [19], η οποία στοχεύει στην εκ προθέσεως μείωση της παρεχόμενης χωρητικότητας επιλεγμένων ροών. Σε οποιοδήποτε ρεαλιστικό σενάριο παραδοσιακού TCP, το bandwidth throttling θα είχε αρνητικό αντίκτυπο στην συνολική ποιότητα εμπειρίας του εκάστοτε χρήστη, καθώς θα ελάττωνε το ρυθμό μετάδοσης δεδομένων. Αντιθέτως, στην περίπτωση του MPTCP, ο «στραγγαλισμός» μίας ζεύξης μπορεί να χρησιμοποιηθεί ως μέσο εξαναγκασμού με απώτερο σκοπό την χειροκίνητη εκφόρτωση δεδομένων σε επιθυμητά μονοπάτια δρομολόγησης.

3.1. Διαχείριση Ουρών Εξερχόμενης Διαδικτυακής Κίνησης

Η επιτυχής εφαρμογή του bandwidth throttling βασίζεται σε μεγάλο ποσοστό στην σωστή διαχείριση των ουρών της εξερχόμενης διαδικτυακής κίνησης στον πυρήνα του λειτουργικού συστήματος Linux. Συγκεκριμένα, ο πυρήνας Linux παρέχει την δυνατότητα προσομοίωσης πολλαπλών ουρών ανά διεπαφή, η κάθε μία εκ των οποίων μπορεί να παραμετροποιηθεί ξεχωριστά με σκοπό την εξυπηρέτηση διαφορετικών ειδών εξερχόμενων πακέτων πληροφορίας ανάλογα με τα αντίστοιχα κριτήρια, τα οποία θα εξεταστούν σε επόμενο υποκεφάλαιο. Με αυτόν τον τρόπο, καθίσταται δυνατή η ιεράρχηση

των διαθέσιμων ζεύξεων όπως τις αντιλαμβάνεται το Multipath TCP, αφού ελέγχοντας το διαθέσιμο εύρος ζώνης ανά ζεύξη επηρεάζεται κατ' επέκταση το RTT και το latency των αντίστοιχων πακέτων. Έτσι, ζεύξεις με κατώτερη θεωρητικά απόδοση μπορούν να καθιστούν προτιμότερες για την μετάδοση δεδομένων σύμφωνα με τους αλγορίθμους ελέγχου συμφόρησης.

Ουσιαστικά, η συγκεκριμένη διαδικασία αποσκοπεί στην εσωτερική εικονική διαίρεση, όπως απεικονίζεται στο Σχήμα 9, και προηγμένη διαχείριση των αποθηκευτικών χώρων των προς-αποστολή-πακέτων (send buffers). Ακόμα, η παρουσία διάφορων αλγορίθμων προγραμματισμού προσδίδει ποικιλία ως προς την επιλογή πολιτικών προγραμματισμού αποστολής πακέτων (queuing disciplines) [20]. Ως αποτέλεσμα, η ποικίλη διαχείριση των buffers δίνει την δυνατότητα για χειραγώγηση της εξερχόμενης κίνησης τόσο σε σενάρια επιβολής πολιτικών σχετικών με QoS, όσο και ως μέσο χειροκίνητης κατανομής των διαθέσιμων διαδικτυακών πόρων.



Σχήμα 9: Προσομοίωση πολλαπλών ουρών εξερχόμενης διαδικτυακής κίνησης

Η περισσότερο διαδεδομένη πολιτική αποστολής πακέτων είναι η FIFO (First-In-First-Out), η οποία ανήκει στην κατηγορία των αταξικών προγραμματισμών (classless queuing disciplines). Σε αυτήν την περίπτωση, τα πακέτα εξυπηρετούνται με την σειρά άφιξής τους στον buffer, ο οποίος αποτελείται από αποκλειστικά μία ουρά εξυπηρέτησης κίνησης. Συνεπώς, η πολιτική FIFO δεν είναι εφοδιασμένη με την δυνατότητα λήψης περίπλοκων αποφάσεων σχετικά με την αποστολή πακέτων πληροφορίας ή την εισαγωγή στοχευόμενων φίλτρων με σκοπό την υπαγόρευση συγκεκριμένων πολιτικών δρομολόγησης. Από την άλλη μεριά, ο συνδυασμός πολλαπλών εικονικών ουρών οδηγεί στην υλοποίηση ταξικών πολιτικών προγραμματισμού αποστολής πακέτων (classful queuing disciplines), οι οποίες μπορούν να χρησιμοποιηθούν για την εξυπηρέτηση διαφορετικών ειδών διαδικτυακής κίνησης. Ουσιαστικά, ο συνδυασμός των κατάλληλων queuing disciplines με την επιβολή φίλτρων, τα οποία θα αναλύσουμε στο υποκεφάλαιο 3.2.1,

μπορεί να οδηγήσει στον επιτυχή διαχωρισμό του συνόλου των πακέτων δεδομένων και την κατ' επέκταση απορρόφηση τους από διαφορετικές κλάσεις, όπως απεικονίζονται στο Σχήμα 9. Ως αποτέλεσμα, καθίσταται δυνατός ο έλεγχος ροής της πληροφορίας μέσω της ανεξάρτητης παραμετροποίησης πολλαπλών ουρών εξυπηρέτησης της εξερχόμενης διαδικτυακής κίνησης.

3.1.1. Ο Αλγόριθμος Hierarchical Token Bucket

Η δυνατότητα προσομοίωσης και αντιστοίχισης πολλαπλών ουρών σε μία φυσική διεπαφή έχει ως απώτερο σκοπό τον διαχωρισμό και την εξυπηρέτηση ανεξάρτητων ροών πληροφορίας. Συγκεκριμένα, οι διάφοροι αλγόριθμοι προγραμματισμού (queuing disciplines) χρησιμοποιούνται για την εικονική διαίρεση των buffers και την κατ' επέκταση επιλογή των κατάλληλων ουρών για την μετάδοση των εκάστοτε πακέτων. Εν προκειμένω, θα εξεταστεί ο αλγόριθμος Hierarchical Token Bucket (HTB) [21], ο οποίος προσδίδει επιπρόσθετη ευελιξία στην διαχείριση του διαθέσιμου εύρους ζώνης της εξερχόμενης διαδικτυακής κίνησης.

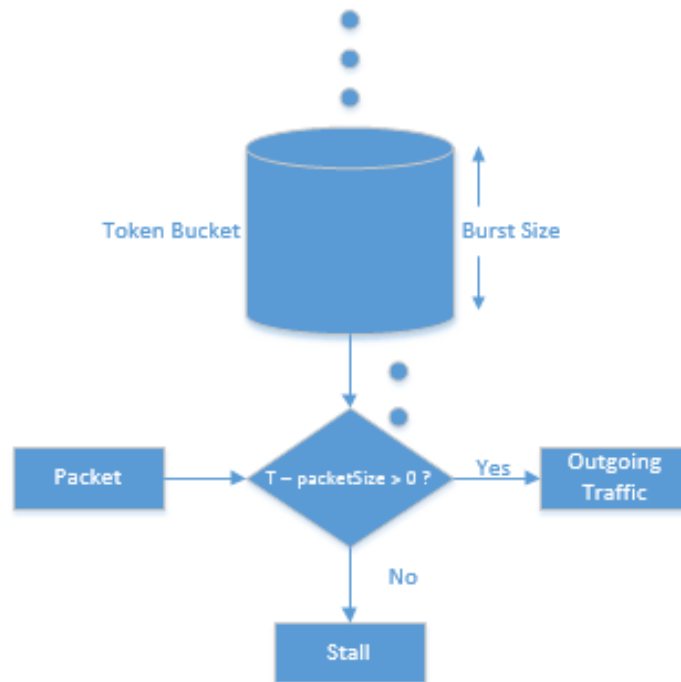
Ο HTB, ως ένας ταξικός αλγόριθμος προγραμματισμού, τείνει να διαφοροποιήσει την ουρά αναμονής του πυρήνα του λειτουργικού συστήματος στα πρότυπα του σχήματος της προηγούμενης υποενότητας. Πιο συγκεκριμένα, η επιθυμητή διεπαφή εφοδιάζεται με την θεμελιώδη πολιτική προγραμματισμού αποστολής πακέτων (root queuing discipline), μέσω της οποίας καθίσταται δυνατή η προσκόλληση διαφορετικών κλάσεων και η παραμετροποίησή τους. Σε κάθε qdisc (queuing discipline) εκχωρείται ένα "handle" για λόγους αναφοράς, το οποίο αποτελείται από δύο μέρη: <major number>:<minor number>. Τυπικά, στο root qdisc εκχωρείται η τιμή "1:0". Οι εμφωλευμένες κλάσεις εφοδιάζονται με ένα "classid" και μία υπόδειξη του γονέα, ώστε να καθίσταται δυνατή η επικοινωνία μεταξύ τους, καθώς δεν λαμβάνει χώρα άμεση επικοινωνία με το ανώτερο επίπεδο της αντίστοιχης διεπαφής.

Ο αλγόριθμος HTB παρέχει την διαβεβαίωση παροχής εύρους ζώνης ίσο με τουλάχιστον το ελάχιστο μεταξύ του απαιτούμενου και του ήδη προεπιλεγμένου εκχωρημένου. Ακόμα, όταν μία κλάση απαιτεί μειωμένη χωρητικότητα συγκριτικά με την δυναμικότητά της, το πλεονάζον εύρος ζώνης μπορεί να διαμοιραστεί στις υπόλοιπες κλάσεις ανάλογα με τις υπάρχουσες ρυθμίσεις τους. Στην συνέχεια, παρουσιάζονται οι τρεις βασικοί παράμετροι μέσω των οποίων προσδιορίζονται τα χαρακτηριστικά των ουρών HTB:

- *Rate*: υποδεικνύει το εύρος ζώνης που αντιστοιχίζεται στην εκάστοτε ουρά εξερχόμενης διαδικτυακής κίνησης, το οποίο καθίσταται πάντα διαθέσιμο από τον kernel.
- *Rate Ceiling*: το μέγιστο εύρος ζώνης που δύναται να κατανοηθεί σε μία κλάση. Ως προεπιλογή λαμβάνεται το rate. Ουσιαστικά, η

συγκεκριμένη παράμετρος τείνει να οριοθετήσει το εύρος ζώνης που μπορεί να δανειστεί από μία υποχρησιμοποιούμενη ουρά.

- *Burst Rate*: υποδεικνύει τον όγκο δεδομένων που μπορεί να αποσταλεί υπό μέγιστο ρυθμό μετάδοσης εν απουσία καθυστέρησης λόγω περιορισμού χωρητικότητας.



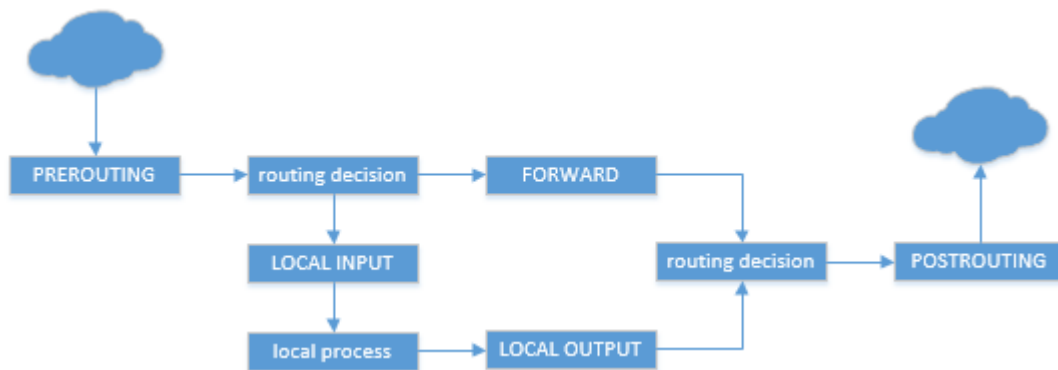
Σχήμα 10: Σχηματική αναπαράσταση αλγορίθμου HTB

Το Σχήμα 10 απεικονίζει την εκτέλεση του αλγορίθμου. Σε κάθε ουρά HTB αντιστοιχίζεται ένας κουβάς με *tokens*, τα οποία εισέρχονται σε αυτόν με σταθερό ρυθμό ίσο με *rate*. Μόλις ένα πακέτο πληροφορίας μεγέθους *packetSize* είναι έτοιμο προς αποστολή, τότε πραγματοποιείται ο έλεγχος εντός του ρόμβου, όπου T ο αριθμός των διαθέσιμων *tokens* εντός του *token bucket*. Σε περίπτωση επιτυχίας του ελέγχου, δηλαδή $T > packetSize$, αφαιρείται από τον κουβά *packetSize* αριθμός *tokens*, ενώ, παράλληλα, το πακέτο απελευθερώνεται και μεταδίδεται. Διαφορετικά, η ουρά εισέρχεται σε μία κατάσταση αναμονής παύοντας την αποστολή δεδομένων μέχρι ο κουβάς να περιλαμβάνει ικανοποιητικό αριθμό *tokens*. Στο ενδιάμεσο, προτεραιότητα λαμβάνουν οι υπόλοιπες ουρές της ιεραρχίας. Όπως φαίνεται και από το παραπάνω σχήμα, το βάθος του *token bucket* αντιστοιχεί στο επιλεγμένο *burst size*, δηλαδή στον όγκο δεδομένων που δύναται να αποσταλεί στον μέγιστο δυνατό ρυθμό μετάδοσης χωρίς να προκληθεί *stall*.

3.2. Netfilter Framework

Το Netfilter Framework [22] αποτελείται από μία σειρά σταδίων, τα οποία επιτρέπουν την περαιτέρω διαχείριση πακέτων πάνω από την στοίβα πρωτοκόλλων στο λειτουργικό σύστημα Linux. Στην περίπτωση του IPv4,

υλοποιούνται πέντε (5) στάδια, όπως φαίνεται στο Σχήμα 11, με σκοπό την δυνατότητα διαχείρισης πακέτων σε διαφορετικά σημεία κατά την διάσχιση της εκάστοτε στοίβας πρωτοκόλλων δικτύου.



Σχήμα 11: Διάσχιση σταδίων του Netfilter Framework

- *PREROUTING*: στο αρχικό αυτό στάδιο, τα πακέτα λαμβάνονται από τον εκάστοτε διαδικτυακό κόμβο και λαμβάνουν χώρα έλεγχοι ασφαλείας και ασφαλείας.
- *LOCAL INPUT*: η πρώτη απόφαση δρομολόγησης υποδεικνύει εάν το ληφθέν πακέτο προορίζεται για κάποια τοπική διαδικασία ή άλλη διεπαφή. Στην πρώτη περίπτωση, διανέμεται το παρόν στάδιο πριν την παράδοση του πακέτου δεδομένων στην αντίστοιχη εφαρμογή.
- *FORWARD*: πακέτα που δεν προορίζονται για το τοπικό μηχάνημα διασχίζουν το συγκεκριμένο στάδιο με σκοπό την προώθησή τους μέσω της κατάλληλης διεπαφής.
- *LOCAL OUTPUT*: το παρόν στάδιο διασχίζεται από τοπικά παραχθέντα πακέτα και αποτελεί το πρωταρχικό σημείο χειραγώγησης τους.
- *POSTROUTING*: το συγκεκριμένο στάδιο διασχίζεται από όλα τα εξερχόμενα πακέτα μετά την λήψη της απόφασης δρομολόγησης, πριν σταλούν στο φυσικό μέσο.

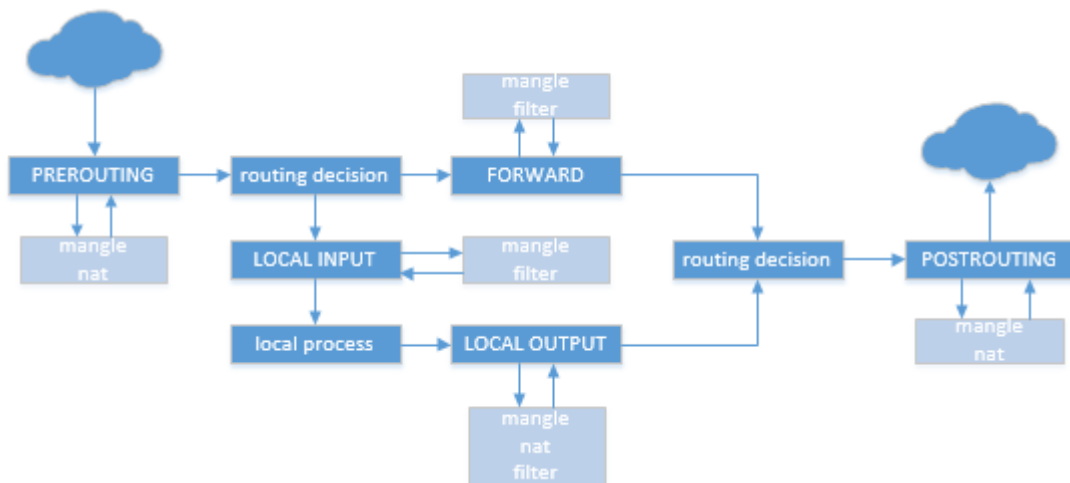
Σε κάθε στάδιο δύναται να καταχωρηθεί μία αλυσίδα κανόνων με σκοπό το επιτυχές πατρωνάρισμα των πακέτων πληροφορίας. Η κλήση του εκάστοτε σταδίου κατά την διάσχισή του από ένα πακέτο ενεργοποιεί την αντίστοιχη αλυσίδα κανόνων. Οι καταχωρημένες λειτουργίες επεξεργάζονται με σειρά προτεραιότητας επιτρέποντας την διαφανή χειραγώγηση των πακέτων βάσει συγκεκριμένων επιλεγμένων κριτηρίων.

3.2.1. IPtables

Το IPtables [23] αποτελεί μία εφαρμογή χτισμένη πάνω στο Netfilter Framework. Ουσιαστικά, πρόκειται για ένα σύστημα επιλογής και

χειραγώγησης πακέτων μέσω του οποίου καθίσταται δυνατό το φιλτράρισμα και η διαφανής τροποποίηση της διαδικτυακής κίνησης. Το Σχήμα 12 απεικονίζει με ακρίβεια τον συνδυασμό Netfilter Framework – IPtables. Σε κάθε στάδιο, modules του πυρήνα του λειτουργικού συστήματος Linux επιτρέπουν την δημιουργία καταχωρήσεων περιλαμβανομένων αλυσίδων κανόνων, οι οποίες αντικατοπτρίζουν τα χαρακτηριστικά των υπό εξέταση πακέτων και των επιθυμητών περαιτέρω ενεργειών.

Το IPtables επικαλείται κατά την προσπέλαση του εκάστοτε σταδίου του Netfilter Framework, αλλά με διαφοροποιημένες δυνατότητες κάθε φορά. Επίσης, οι συνθήκες αντιστοίχισης των πακέτων είναι δυνατό να επεκταθούν σε παραπάνω από ένα επίπεδο του μοντέλου του OSI. Με αυτόν τον τρόπο, καθίστανται δυνατές ενέργειες, όπως η επίτευξη NAT (Network Address Translation), το φιλτράρισμα πακέτων, καθώς και η τροποποίησή τους. Στο προκείμενο, θα επικεντρωθούμε στο *packet mangling* ως μέσο μερικής, προσωρινής τροποποίησης των πακέτων.



Σχήμα 12: Πλήρης απεικόνιση Netfilter Framework και IPtables

Ο συγκεκριμένος πίνακας (mangle table) επιτρέπει την εισαγωγή κανόνων με στόχο το σημάδεμα (mark) πακέτων. Ως αποτέλεσμα, καθίσταται δυνατή η χειραγώγηση πακέτων σχετικά με την επιλογή διαδρομών δρομολόγησης, καθώς και κατάλληλων qdiscs ανάλογα με το εκάστοτε *mark value*. Σε αυτό το σημείο, αξίζει να σημειωθεί πως οι τιμές που αντιστοιχίζονται στα πακέτα δεν τυπώνονται στο εσωτερικό τους οδηγώντας στην τροποποίησή τους, αλλά ορίζονται ως μέσο αναφοράς στον πυρήνα του λειτουργικού συστήματος. Έτσι, οι δοθείσες τιμές δεν διατηρούνται μετά την απομάκρυνση των αντίστοιχων πακέτων από τον kernel. Επίσης, να τονισθεί πως οι κανόνες εντός του πίνακα mangle επεξεργάζονται με σειρά προτεραιότητας. Η διάσχιση του πίνακα σταματάει μόλις λάβει χώρα η ικανοποίηση ενός εκ των κανόνων ή διασχιστεί ολόκληρος ο πίνακας δίχως την επιτυχή αντιστοίχιση κάποιου πακέτου πληροφορίας σε κάποιον κανόνα της αλυσίδας.

4. Μηχανισμός Επιβολής Bandwidth Throttling

Οι απαιτούμενες βάσεις επιβολής *bandwidth throttling*, όπως αναλύθηκαν στο προηγούμενο κεφάλαιο, αποτελούνται από την δυνατότητα εικονικής διαίρεσης και κατ' επέκταση διαχείρισης των *send buffers* μέσω κατάλληλων *queuing disciplines* σε συνδυασμό με την εισαγωγή των απαραίτητων κανόνων μέσω *IPtables* για τον επιτυχή διαχωρισμό του συνολικού όγκου της εξερχόμενης διαδικτυακής κίνησης. Ως αποτέλεσμα, τα πακέτα πληροφορίας απορροφούνται από τις αντίστοιχες ουρές αναμονής πακέτων ανάλογα με τα υπάρχοντα *mark values* και αποστέλλονται βάσει της εκάστοτε προσφερθείσας χωρητικότητας. Με αυτόν τον τρόπο, καθίσταται δυνατή η παρεμβολή στην διαδικασία επιλογής μονοπατιών δρομολόγησης σε μία σύνδεση *MPTCP*, όταν αξιοποιούνται πλήρως ταυτόχρονα περισσότερες από μία υπο-ροές *TCP*.

Για τον σκοπό της παρούσας διπλωματικής εργασίας, δημιουργήσαμε έναν μηχανισμό για την δυναμική επιβολή *bandwidth throttling* ανάλογα με τα χαρακτηριστικά των εισερχόμενων συνεδριών *MPTCP*. Συγκεκριμένα, ο μηχανισμός αποτελείται από έναν *packet sniffer*, ο οποίος ελέγχει το πεδίο επιλογών των εισερχόμενων πακέτων *SYN*. Σε περίπτωση αντιστοίχισης κάποιου πακέτου συγχρονισμού σε μία δευτερεύον υπο-ροή μίας ήδη εγκατεστημένης σύνδεσης *MPTCP*, παράμετροι της υπο-ροής *TCP* αποστέλλονται σε ένα ταυτόχρονα εκτελέσιμο πρόγραμμα, το οποίο καθίσταται υπεύθυνο για την επιτυχή επιβολή *bandwidth throttling* μέσω της κατάλληλης αναπροσαρμογής των *queuing disciplines* στον πυρήνα του λειτουργικού συστήματος *Linux*. Παράλληλα, διατηρούμε ένα αρχείο παραμετροποίησης του μηχανισμού μέσω του οποίου έχουμε την δυνατότητα αναθεώρησης της διαθέσιμης-προσφερόμενης χωρητικότητας ανά διεπαφή.

4.1. Scapy (Packet) Sniffer

Ο μηχανισμός δυναμικής επιβολής *bandwidth throttling* αποτελείται ουσιαστικά από δύο κομμάτια πηγαίου κώδικα. Το πρώτο εξ αυτών αντιστοιχεί σε έναν *Scapy* [24] *sniffer*, ο οποίος ως πρωταρχικό στόχο έχει την επιθεώρηση των εισερχόμενων πακέτων με απώτερο σκοπό την ερμηνεία της δυνατότητας αξιοποίησης πολλαπλών διαδρομών δρομολόγησης από το εκάστοτε απομακρυσμένο τερματικό.

Το εργαλείο *Scapy* αποτελεί ένα διαδραστικό πρόγραμμα χειραγώγησης πακέτων πληροφορίας, το οποίο υλοποιείται εγγενώς σε *Linux*, ενώ επίσης, είναι δυνατό να χρησιμοποιηθεί ως βιβλιοθήκη στην γλώσσα προγραμματισμού *Python*. Το εκτεταμένο εύρος λειτουργιών του επιτρέπει την αξιοποίηση του εργαλείου πέρα των περιορισμών των συνήθων προγραμμάτων επιθεώρησης πακέτων δεδομένων ή γενικότερης σάρωσης του δικτύου, όπως παραδείγματος χάριν το *Wireshark* [25]. Πιο συγκεκριμένα, καθίσταται δυνατή η ολική δημιουργία και αποκωδικοποίηση πακέτων μίας ευρείας γκάμας πρωτοκόλλων, η αποστολή τους στο φυσικό μέσο και η

σύλληψή τους από τον εκάστοτε διαδικτυακό κόμβο, καθώς και η ταυτοποίηση αιτήσεων-απαντήσεων μεταξύ τερματικών, κλπ. Επιπρόσθετα, το εργαλείο Scapy παρέχει την δυνατότητα παρεμβολής στην αυτοματοποιημένη διαδικασία δημιουργίας πακέτων πληροφορίας με σκοπό τον σχηματισμό ιδιαίτερα διαμορφωμένων πακέτων, είτε έγκυρων είτε άκυρων, στοχεύοντας στον έλεγχο της ορθής και φερέγγυας λειτουργίας του εκάστοτε υποδικτύου.

Εν προκειμένω, το πρόγραμμα Scapy, όπως έχει ήδη αναφερθεί, θα χρησιμοποιηθεί ως μέσο *packet sniffing*. Σε αντίθεση με άλλα προγράμματα παρόμοιων δυνατοτήτων, ο Scapy Sniffer δεν τυπώνει τις ζητούμενες πληροφορίες εκ των προτέρων ερμηνευμένες και σε φιλική προς τον χρήστη μορφή. Τα δεδομένα εξόδου παρέχονται σε ακατέργαστη μορφή επιτρέποντας στον χρήστη την εξ ολοκλήρου επεξεργασία τους και τον πλήρη έλεγχο των παρεχόμενων πληροφοριών. Η συγκεκριμένη ιδιαιτερότητα καθίσταται σημαντική για την επιτυχή ερμηνεία των σημάτων ελέγχου του MPTCP εντός του πεδίου επιλογών της επικεφαλίδας TCP, τα οποία μπορεί να αποδειχθούν δυσνόητα για κάποιο εναλλακτικό πρόγραμμα επιθεώρησης πακέτων, αφού, όπως έχουμε ήδη τονίσει, η γενικότερη υιοθέτηση του επεκταμένου πρωτοκόλλου από τους διάφορους διαδικτυακούς κόμβους ή/και πλατφόρμες παροχής υπηρεσιών βρίσκεται ακόμα σε πρώιμο στάδιο.

Σε αυτό το σημείο, για εύληπτους λόγους θα αναφερθούμε συνοπτικά στην τοπολογία διεξαγωγής των πειραμάτων αξιολόγησης του πρωτοκόλλου, η οποία αποτελείται από ένα έξυπνο τερματικό με την δυνατότητα αξιοποίησης πολλαπλών διεπαφών, έναν απομακρυσμένο HTTP Server εφοδιασμένο με την παραδοσιακή έκδοση του TCP και έναν μετατροπέα πρωτοκόλλου υπό την μορφή ενός HTTP Proxy. Το σύνολο του μηχανισμού επιβολής *bandwidth throttling* υλοποιείται εντός του MPTCP Proxy, ο οποίος κατέχει μία μοναδική δικτυακή διεπαφή και τείνει να περιορίσει το διαθέσιμο εύρος ζώνης που παρέχεται ανά διεπαφή του έξυπνου τερματικού.

Τα αρχεία πηγαίου κώδικα ολόκληρου του μηχανισμού μπορούν να βρεθούν στο Παράρτημα στο τέλος του κειμένου. Για ευνόητους λόγους, παρακάτω παρουσιάζεται ο αλγόριθμος του Scapy Sniffer υπό την μορφή ψευδοκώδικα:

```
while True sniff packets
```

```
if (IP header) is present then get (source IP address)
```

```
if (TCP header) is present then get (TCP flags) and (TCP options)
```

```
if (TCP SYN flag) and (MPTCP option) is present then
```

```
if (packet belongs to primary subflow) then
```

```
register (primary IP address) to JSON-RPC server
```

```
if (packet belongs to secondary subflow) then
```

```
register (secondary IP address) to JSON-RPC server
```

```
register (MPTCP back-up flag) to JSON-RPC server
```

```
register (JSON-RPC kill function) to JSON-RPC server
```

```
activate JSON-RPC server
```

Ο sniffer επιθεωρεί τα εισερχόμενα πακέτα και αναλύει τις επικεφαλίδες των πακέτων TCP/IP. Μόλις ληφθεί ένα πακέτο SYN, το οποίο είναι υπεύθυνο για την έναρξη της συνεδρίας MPTCP υποδεικνύοντας την δυνατότητα αξιοποίησης πολλαπλών διαδρομών από το εκάστοτε τερματικό, τότε αποθηκεύεται η αντίστοιχη IP διεύθυνση πηγής. Με την λήψη περαιτέρω πακέτων συγχρονισμού, τα οποία υποδεικνύουν δευτερεύοντες υπο-ροές TCP μέσω των αντίστοιχων MPTCP flags [6], καταχωρούνται οι αντίστοιχες IP διευθύνσεις πηγής παράλληλα με την επιλογή του MPTCP που επισημαίνει την λειτουργία, ενεργή ή εφεδρική, υπό την οποία εισάγεται η νέα υπο-ροή TCP.

Τιμή	Περιγραφή
x00	Η πρωταρχική υπο-ροή εγκαθιστά μία σύνδεση TCP
x10	Η δευτερεύον υπο-ροή προστίθεται στην συνεδρία MPTCP
x11	Η δευτερεύον υπο-ροή προστίθεται ως εφεδρική ζεύξη

Πίνακας 1: Επιλογές MPTCP

Σε αυτό το σημείο, απαιτείται μία εσωτερική διαδικασία αποστολής των παραπάνω παραμέτρων, διευθύνσεις IP και επιλογές MPTCP, στο δεύτερο κομμάτι πηγαίου κώδικα που εκτελείται ταυτοχρόνως εντός του MPTCP Proxy και καθίσταται υπεύθυνο για την τελική εφαρμογή του *bandwidth throttling* μέσω της κατάλληλης εναλλαγής των *queuing disciplines*.

4.2. JSON-RPC

Η επικοινωνία μεταξύ των δύο παράλληλα εκτελέσιμων προγραμμάτων επιτυγχάνεται μέσω ενός RPC (Remote Procedure Call), δηλαδή μέσω της δυνατότητας κλήσης απομακρυσμένων διαδικασιών. Το πρωτόκολλο υπακούει στο κλασικό σχήμα client-server, όπου ο client έχει την δυνατότητα να επικαλεστεί διαδικασίες-συναρτήσεις εγγεγραμμένες στον απομακρυσμένο server. Εν προκειμένω, όπως περιγράφεται και από τον ψευδοκώδικα της προηγούμενης ενότητας, ο Scapy Sniffer κατέχει τον ρόλο του server, ενώ το πρόγραμμα δυναμικής τροποποίησης των *queuing disciplines* λειτουργεί ως ο αντίστοιχος client.

Στην παρούσα περίπτωση, οι απομακρυσμένες μέθοδοι επικαλούνται μέσω αιτήσεων κωδικοποιημένων κατά JSON (JavaScript Object Notation) [26]. Το συγκεκριμένο σχήμα σειριοποίησης δεδομένων αποτελεί μία ευανάγνωστη μορφή ανταλλαγής δεδομένων, εναλλακτική της XML [27], η οποία παράλληλα, είναι ελαφριά για τα υπολογιστικά συστήματα ανάλυσης και παραγωγής της. Η δομή των παραγόμενων αντικειμένων (objects) είναι της μορφής name/value.

Στην συνέχεια, παρουσιάζονται τα είδη μηνυμάτων ενός JSON-RPC [28]:

- *Request*: το αίτημα κλήση μίας διαδικασίας από τον client, η οποία υλοποιείται σε κάποιο απομακρυσμένο μηχανήμα ή παράλληλα

εκτελέσιμο τοπικό πρόγραμμα. Οι τρεις (3) ιδιότητες του συγκεκριμένου είδους μηνυμάτων είναι:

- *method* – μία συμβολοσειρά που εμπεριέχει το όνομα της προς κλήση μεθόδου
 - *params* – το σύνολο των ζητούμενων παραμέτρων που θα περαστούν στην υπό κλήση μέθοδο
 - *id* – το αναγνωριστικό του σταλθέντος αιτήματος, το οποίο χρησιμοποιείται για την αντιστοίχιση του μηνύματος απόκρισης
- *Response*: η απόκριση στο αντίστοιχο αίτημα επίκλησης μίας απομακρυσμένης διαδικασίας με ιδιότητες:
 - *result* – το αντικείμενο που επιστρέφεται από την κλήση της αντίστοιχης μεθόδου. Το συγκεκριμένο πεδίο πρέπει να εμπεριέχει την τιμή null σε περίπτωση σφάλματος
 - *error* – αντικείμενο υπόδειξης σφάλματος κατά την διαδικασία επίκλησης κάποιας απομακρυσμένης μεθόδου. Το παρόν αντικείμενο οφείλει να είναι null εν απουσία σφάλματος
 - *id* – το αναγνωριστικό του μηνύματος απόκρισης, το οποίο πρέπει να έχει την ίδια τιμή με το αντίστοιχο request id
 - *Notification*: μία ειδική μορφή αιτήματος, η οποία δεν απαιτεί την αποστολή μηνύματος απόκρισης. Το παρόν είδος μηνυμάτων κατέχει τις ίδιες ιδιότητες με το απλό request με την ιδιαιτερότητα ότι το στοιχείο id είναι null

Οι προδιαγραφές του πρωτοκόλλου δεν απαιτούν αυστηρά την παρουσία συγκεκριμένου μέσου για την μετάδοση των δεδομένων. Από την μία πλευρά, δύνανται να χρησιμοποιηθούν *sockets* για την αποστολή των αιτήσεων/αποκρίσεων μέσω της ροής δεδομένων, ενώ από την άλλη μεριά, η επικοινωνία είναι δυνατό να πραγματοποιηθεί μέσω του πρωτοκόλλου HTTP. Στην τελευταία περίπτωση, η οποία αντικατοπτρίζει την τρέχουσα υλοποίηση, η πλευρά του client αιτείται τις επιθυμητές απομακρυσμένες διαδικασίες μέσω της μεθόδου HTTP POST, η οποία περιλαμβάνει τα σειριοποιημένα αντικείμενα.

Εν προκειμένω, ο JSON-RPC server, ο οποίος υλοποιείται εντός του Scapy Sniffer, τροφοδοτείται με τις διευθύνσεις IP και τις επιλογές του MPTCP που υποδηλώνουν την λειτουργία της εκάστοτε υπο-ροής TCP. Η πλευρά του client, όπως θα δούμε αναλυτικά στο επόμενο υποκεφάλαιο, απόκτα πρόσβαση στις εγγεγραμμένες συναρτήσεις μόλις ανακαλυφθεί η παρουσία κάποιας δευτερεύουσας υπο-ροής TCP. Σε αντίθετη περίπτωση, ο server παραμένει απενεργοποιημένος, αφού δεν υπάρχουν καταχωρημένες επαρκείς πληροφορίες για την εφαρμογή bandwidth throttling εν απουσία περισσότερων από μίας ροών TCP.

Μία ιδιαιτερότητα του RPC server κατά την εκκίνησή του αποτελεί η επ' αόριστον εξυπηρέτηση αιτημάτων. Η ενεργοποίηση του server πυροδοτείται με την εκτέλεση της εντολής *serve_forever()*, η οποία δημιουργεί έναν ατέρμονο βρόχο που τείνει να εγκλωβίσει το σύνολο του μηχανισμού. Ουσιαστικά, η παραπάνω εντολή εμποδίζει την εκτέλεση του επακόλουθου

πηγαίου κώδικα αποτρέποντας με αυτόν τον τρόπο τον sniffer να συνεχίσει την επιθεώρηση της διαδικτυακής κίνησης. Γι' αυτόν τον λόγο, όπως περιγράφεται και από τον ψευδοκώδικα του προηγούμενου υποκεφαλαίου, χρειάστηκε να υλοποιήσουμε ένα *kill signal* με σκοπό τον απομακρυσμένο τερματισμό του server από τον client. Το συγκεκριμένο σήμα προστίθεται στην βάση του RPC server και αποστέλλεται ως συνάρτηση στον client μαζί με τις αιτούμενες παραμέτρους των υπο-ροών TCP. Μόλις ολοκληρωθεί η μεταφορά των απαιτούμενων πληροφοριών ο client εκτελεί την απαραίτητη συνάρτηση τερματισμού με στόχο την απομακρυσμένη απενεργοποίηση του server και την συνέχιση της εκτέλεσης των αντίστοιχων προγραμμάτων.

4.3. Bandwidth Throttling Daemon

Η λήψη των ζητούμενων παραμέτρων από τον JSON-RPC client, ο οποίος υλοποιείται ως μέρος του προγράμματος δυναμικής εναλλαγής των *queuing disciplines*, πυροδοτεί την εκτέλεση των απαραίτητων διεργασιών για την τελική επιβολή του *bandwidth throttling*. Αρχικά, όπως περιγράφεται και από τον παρακάτω ψευδοκώδικα, διεξάγεται έλεγχος του πίνακα mangle του σταδίου OUTPUT της εφαρμογής IPtables ως προς την ύπαρξη των αναγκαίων κανόνων με στόχο το σημάδεμα πακέτων.

```
while True
    try connecting to JSON-RPC server

        get (primary IP address)
        get (secondary IP address)
        get (MPTCP back-up flag)

        run JSON-RPC kill function

        check IPtables mangle table for IP addresses mark values
        if entries exist then
            pass
        else
            insert rules

        if secondary subflow joins as active link then
            if qdisc = FIFO then
                qdisc = HTB
            else if qdisc = HTB then
                pass
        else if secondary subflow joins as back-up link then
            if qdisc = HTB then
                qdisc = FIFO
            else if qdisc = FIFO then
                pass

    except if JSON-RPC server is down
```

Αν οι αντίστοιχες καταχωρήσεις δεν συμπεριλαμβάνονται, τότε απαιτείται η εισαγωγή τους συναρτήσει των ληφθέντων διευθύνσεων IP. Για λόγους κατανόησης, παρουσιάζεται ένα παράδειγμα της χρησιμοποιούμενης εντολής:

```
> iptables -t mangle -A OUTPUT -p tcp -s 'MPTCP Proxy IP address' --
source-port 8080 -d 'IP address corresponding to the to-be-throttled link' -j
MARK --set-mark 'x'
```

```
> iptables -t mangle -A OUTPUT -p tcp -s 'MPTCP Proxy IP address' --
source-port 8080 -d 'IP address corresponding to the immutable link' -j
MARK --set-mark 'y'
```

-s: η IP διεύθυνση πηγής, η οποία αντιστοιχεί στην διεπαφή του μετατροπέα πρωτοκόλλου

--source-port 8080: η πόρτα μέσω της οποίας ο διακομιστής μεσολάβησης λαμβάνει και προωθεί πακέτα

-d: η IP διεύθυνση προορισμού, η οποία κάθε φορά αντικατοπτρίζει μία εκ των πολλαπλών διεπαφών του έξυπνου τερματικού

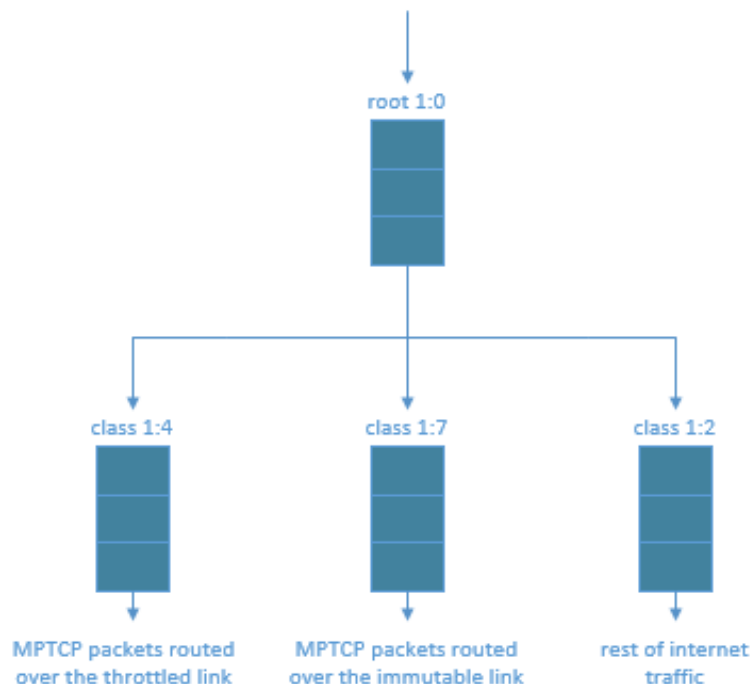
-j MARK --set-mark: η τελική ενέργεια που διεξάγει ο κανόνας, δηλαδή το σημάδεμα των σύμφωνων πακέτων με το αντίστοιχο value

Η επιτυχής ολοκλήρωση της παραπάνω διαδικασίας, η οποία λαμβάνει χώρα μετά την μετατροπή πρωτοκόλλου, δρομολογεί την αναδιαμόρφωση των ουρών της εξερχόμενης διαδικτυακής κίνησης με στόχο τον «στραγγαλισμό» του διαθέσιμου εύρους ζώνης των επιθυμητών ζεύξεων με απώτερο σκοπό την χειροκίνητη εκφόρτωση δεδομένων. Μετά την επιτυχημένη εγκατάσταση της πρωταρχικής σύνδεσης TCP, η κατάλληλη προσαρμογή των *queuing disciplines* εξαρτάται από την επιλογή λειτουργίας των δευτερευόντων υποροών. Σε περίπτωση εισαγωγής μίας επιπρόσθετης, ενεργής υπο-ροής TCP, τότε λαμβάνει χώρα η αναδιαμόρφωση της προεπιλεγμένης ουράς FIFO σε HTB qdisc σύμφωνα με το Σχήμα 13.

Συγκεκριμένα, λόγω της έξυπνης συσκευής που χρησιμοποιούμε κατά την διεξαγωγή των πειραμάτων αξιολόγησης του Multipath TCP, η οποία είναι εφοδιασμένη με δύο (2) δικτυακές διεπαφές, επιλέγουμε να διαιρέσουμε εικονικά την ουρά διαχείρισης της διαδικτυακής κίνησης του kernel σε τρεις (3) ανεξάρτητες. Η μία εξ αυτών (class 1:2) διατηρεί ένα εύλογο διαθέσιμο εύρος ζώνης για την εξυπηρέτηση κλασσικών πακέτων δεδομένων ασυσχέτιστων με το επεκταμένο πρωτόκολλο. Με αυτόν τον τρόπο, παράλληλα εκτελέσιμες εφαρμογές ή/και υπηρεσίες παραμένουν τυπικά ανεπηρέαστες από την μείωση του προσφερόμενου εύρους ζώνης. Οι εναπομείναντες δύο κλάσεις καθίστανται υπεύθυνες για την απορρόφηση των πακέτων MPTCP σύμφωνα με τα αντίστοιχα *mark values* και την προώθησή τους μέσω των διαθέσιμων διαδρομών σύμφωνα με το παρεχόμενο εύρος ζώνης.

Από την άλλη μεριά, η εισαγωγή μίας νέας υπο-ροής TCP ως εφεδρική στην συνεδρία MPTCP οδηγεί στην διατήρηση της κλασσικής ουράς FIFO, αφού πρακτικά δεν πραγματοποιείται άμεση αξιοποίηση όλων των διαθέσιμων

συνδέσεων μέσω της συσσώρευσης εύρους ζώνης. Ως εκ τούτου, η επιβολή *bandwidth throttling* θα οδηγούσε σε αρνητικές συνέπειες ως προς την ποιότητα εμπειρίας ανάλογες του περιορισμού εύρους ζώνης σε μία παραδοσιακή σύνδεση TCP.



Σχήμα 13: HTB qdisc

Τέλος, να υπογραμμίσουμε πως η παρούσα υλοποίηση των *queuing disciplines* αποτρέπει τον διαμοιρασμό του πλεονάζοντος εύρους ζώνης μεταξύ των διαθέσιμων ουρών HTB. Με αυτόν τον τρόπο, καθίσταται δυνατός ο πλήρης έλεγχος της παρεχόμενης χωρητικότητας ανά διαδρομή δρομολόγησης δίχως τυχόν απρόσμενες αυξομειώσεις της απόδοσης των εμπλεκόμενων υπο-ροών οφειλόμενες σε πιθανές μεταβολές της προσφερόμενης χωρητικότητας.

4.4. Αρχείο Παραμετροποίησης Μηχανισμού

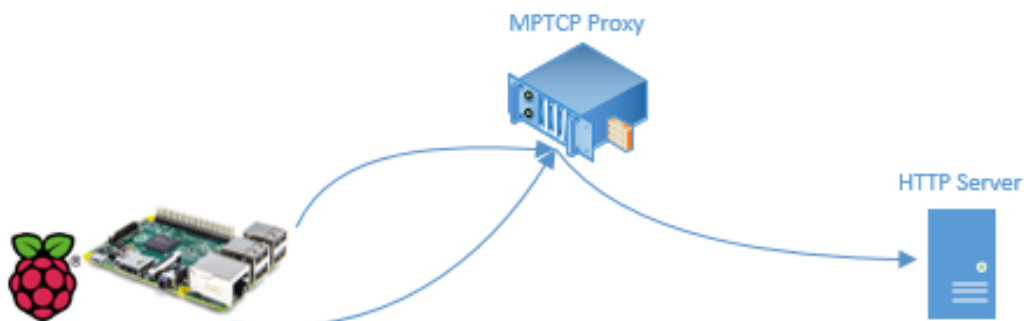
Το αρχείο παραμετροποίησης παρέχει την δυνατότητα ρύθμισης του μηχανισμού επιβολής *bandwidth throttling* στα επιθυμητά επίπεδα παροχής εύρους ζώνης ανά ζεύξη. Συγκεκριμένα, δίνεται η ευκαιρία διαχωρισμού των εμπλεκόμενων υποδικτύων με σκοπό την κατηγοριοποίηση των διαθέσιμων διεπαφών βάσει των ανατεθειμένων διευθύνσεων IP. Στο προκείμενο, το έξυπνο τερματικό που διαθέτουμε για την αξιολόγηση του Multipath TCP χρησιμοποιεί δύο διαδικτυακές διεπαφές, μία ενσύρματη και μία ασύρματη, οι οποίες συνδέονται σε διαφορετικά υποδίκτυα. Το αρχείο ρυθμίσεων παρέχει στον μηχανισμό *bandwidth throttling* τις διευθύνσεις IP των αντίστοιχων υποδικτύων, ώστε να καθίσταται δυνατή η ανακάλυψη των εμπλεκόμενων υπο-ροών. Ακόμα, το αρχείο παραμετροποίησης καταδεικνύει το παρεχόμενο *rate* και *rate ceiling* των ουρών HTB, επιτρέποντας έτσι την

αναπροσαρμογή των πολιτικών δρομολόγησης σύμφωνα με τα υπάρχοντα κριτήρια χρησιμοποίησης των διαθέσιμων τεχνολογιών πρόσβασης στο διαδίκτυο.

Σε ένα πιο ρεαλιστικό σενάριο τηλεπικοινωνιών, ο παραπάνω διαχωρισμός των δύο διαθέσιμων τεχνολογιών πρόσβασης βάσει των διευθύνσεων IP θα μπορούσε να αντικατοπτρίζει τα δίκτυα 4G/3G και Wi-Fi. Σε αυτήν την περίπτωση, η τεχνική *bandwidth throttling* θα ήταν δυνατό να χρησιμοποιηθεί ως μέσο ανακούφισης των δικτύων 4^{ης}/3^{ης} γενιάς από την παρουσία αυξημένης συμφόρησης με την ταυτόχρονη αξιοποίηση μη-αδειοδοτούμενου φάσματος. Παράλληλα, καθίσταται δυνατή η εν μέρει παροχή διασφάλισης στους τελικούς χρήστες ενάντια στην υπερχρέωση των συνδρομών προγραμμάτων δεδομένων κατά την συνεχή αξιοποίηση όλων των διαθέσιμων ραδιοηλεκτρικών διεπαφών μίας έξυπνης συσκευής ή κινητού τηλεφώνου.

5. Αξιολόγηση Πρωτοκόλλου

Η πλατφόρμα δοκιμών για την αξιολόγηση του πρωτοκόλλου, η οποία απεικονίζεται στο Σχήμα 14, αποτελείται από μία έξυπνη συσκευή, έναν απομακρυσμένο HTTP server και έναν MPTCP Proxy. Το έξυπνο τερματικό είναι ένα Raspberry Pi (Model B) με εγκατεστημένο το λειτουργικό σύστημα Raspbian (Debian Wheezy) στην κορυφή ενός ειδικά διαμορφωμένου Linux kernel 3.12.35+ εφοδιασμένο με την έκδοση 0.88.13 του Multipath TCP [2]. Ο MPTCP Proxy, ο οποίος στεγάζεται εντός ενός ESXi hypervisor [29], έχει εγκατεστημένο τον Squid Proxy [30], ο οποίος δραστηριοποιείται στην πόρτα 8080. Ο μετατροπέας πρωτοκόλλου χρησιμοποιεί το λειτουργικό σύστημα Ubuntu 14.04.1 με πυρήνα 3.14.0, στον οποίο έχει χορηγηθεί η σταθερή έκδοση 0.89.2 του επεκταμένου πρωτοκόλλου. Τέλος, ο παραδοσιακός HTTP server, ο οποίος έχει εγκατεστημένο τον Apache2 [31] στην πόρτα 80, βρίσκεται σε απομονωμένο περιβάλλον με σκοπό την αποκλειστική εξυπηρέτηση των πειραματικών αιτημάτων.



Σχήμα 14: Πειραματική διάταξη

Καθ' όλη την διάρκεια των πειραμάτων, έχει απενεργοποιηθεί ο μηχανισμός προσωρινής αποθήκευσης δεδομένων του Squid με στόχο την ακριβή καταγραφή της καθαρής απόδοσης της ικανότητας προώθησης πακέτων. Ακόμα, έχει τεθεί εκτός λειτουργίας η ικανότητα αποθήκευσης του *slow start threshold* του μηχανισμού ελέγχου συμφόρησης για κάθε IP διεύθυνση προορισμού. Η συγκεκριμένη επιλογή TCP αποθηκεύει την παραπάνω τιμή σκοπεύοντας στην μελλοντική της χρησιμοποίηση από νέες ροές TCP. Στην περίπτωση του επεκταμένου πρωτοκόλλου, η ενεργοποίηση της παραπάνω επιλογής θα οδηγούσε τυχόν νέες υπο-ροές TCP να εισέλθουν πρώιμα στο στάδιο Αποφυγής Συμφόρησης του αλγορίθμου ελέγχου συμφόρησης, μία κατάσταση που θα επέφερε υποβάθμιση της γενικότερης επίδοσης της συνεδρίας MPTCP.

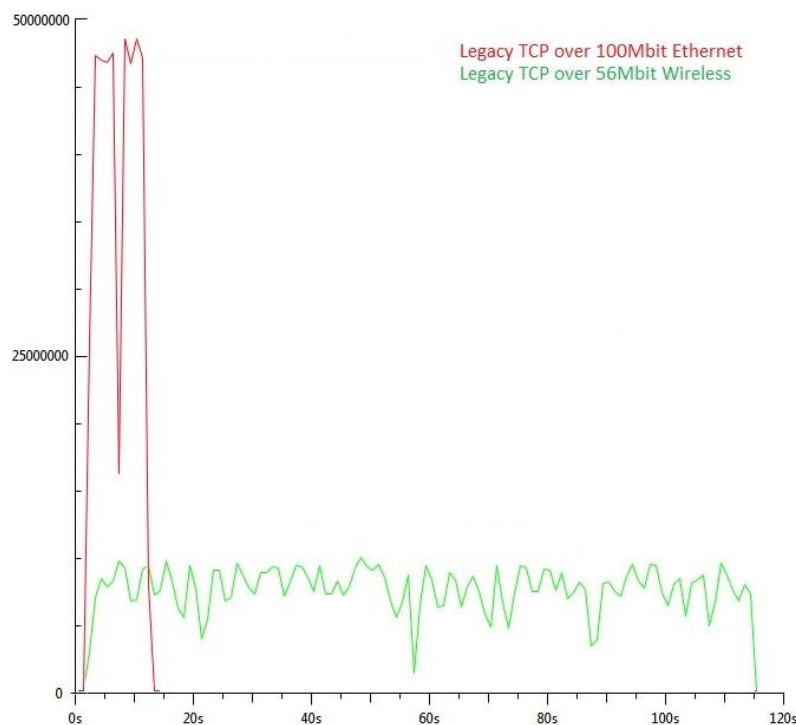
Η επιλογή ενός HTTP διακομιστή μεσολάβησης ως τη βάση της υλοποίησης του MPTCP Proxy βασίζεται στην ευρεία χρησιμοποίηση του πρωτοκόλλου HTTP και στην θεμελιώδη αξία του για τον Παγκόσμιο Ιστό [32]. Σήμερα, το πρωτόκολλο HTTP μεταφέρει την συντριπτική πλειοψηφία της IP κίνησης μεταξύ των δικτύων υπολογιστών. Η πλοήγηση του διαδικτύου, η μεταφορά αρχείων, η επικοινωνία μέσω e-mail, καθώς και online υπηρεσίες παροχής υπηρεσιών video, όλα καθίστανται εφικτά μέσω του HTTP, ενώ τα διάχυτα

χρησιμοποιούμενα Web APIs απαιτούν την πρόσβασή τους μέσω τυπικών μεθόδων HTTP.

Η σύνδεση ενός χρήστη σε κάποια απομακρυσμένη πλατφόρμα παροχής υπηρεσιών μέσω του MPTCP Proxy απαιτεί την κατάλληλη υπόδειξη της τελικής IP διεύθυνσης προορισμού, ώστε να εγκατασταθεί επιτυχώς η σύνδεση από άκρο σε άκρο. Στην περίπτωση ενός HTTP Proxy, η διεύθυνση του απομακρυσμένου server εμπεριέχεται στην επικεφαλίδα του HTTP. Μόλις ο διακομιστής μεσολάβησης εισαχθεί ρητά στην σύνδεση, οι πολλαπλές υπο-ροές TCP τερματίζουν στην πλευρά του client και η νέα συνεδρία TCP εγκαθίσταται μεταξύ του MPTCP Proxy και του επιθυμητού server παροχής υπηρεσιών.

5.1. Legacy TCP

Στο πρωταρχικό πειραματικό σενάριο, καταδεικνύουμε τα επίπεδα επίδοσης που επιτυγχάνει το παραδοσιακό TCP κατά την λήψη ενός αρχείου 100 MB, όταν χρησιμοποιείται μία διεπαφή Ethernet της τάξης των 100 Mbps και μία ασύρματη ζεύξη δυναμικότητας 56 Mbps, αντίστοιχα. Καθ' όλη την διάρκεια της πειραματικής αξιολόγησης, αξιοποιούμε τις δύο προαναφερθείσες διεπαφές του Raspberry Pi. Παρ' όλα αυτά, οι ίδιες αρχές ισχύουν για έξυπνα τερματικά με δυνατότητα ταυτόχρονης υποστήριξης διαφορετικών ειδών υποδικτύων και κινητούς κόμβους εξοπλισμένους με πολλαπλές ραδιοηλεκτρικές διεπαφές.



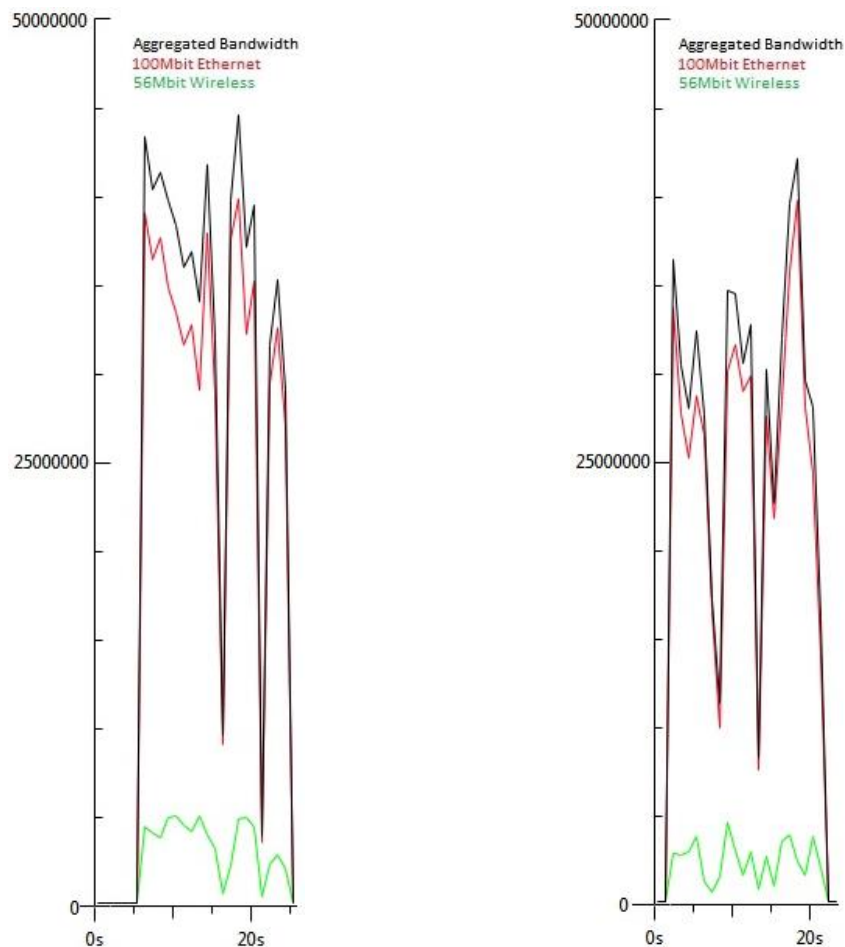
Σχήμα 15: Επίδοση παραδοσιακού TCP σε Mb/s

Το Σχήμα 15 παρουσιάζει την απόδοση του TCP κατά την άμεση επικοινωνία με τον απομακρυσμένο HTTP server εν απουσία του MPTCP Proxy. Η

πλευρά του πελάτη επιδιώκει την διαπραγμάτευση υποστήριξης πολλαπλών διαδρομών μέσω της αρχικής τριμερής χειραψίας. Όμως, το μήνυμα απάντησης από τον server δεν υποδεικνύει την δυνατότητα αξιοποίησης του MPTCP. Ως εκ τούτου, το Raspberry Pi εγκαταλείπει την προσπάθεια αξιοποίησης πολλαπλών διαδρομών δρομολόγησης και χρησιμοποιεί αποκλειστικά μία διεπαφή.

5.2. Σύγκριση Αλγορίθμων Ελέγχου Συμφόρησης

Δεδομένης της αδυναμίας του TCP να αξιοποιήσει πλήρως όλες τις διαθέσιμες διεπαφές που κατέχει το Raspberry Pi, παρεμβάλλουμε τον MPTCP Proxy στην διαδρομή ανάμεσα στα δύο άκρα επικοινωνίας. Κατά την διάρκεια διεξαγωγής των πειραμάτων αξιολόγησης, ο Squid Proxy δεν τίθεται σε διαφανή λειτουργία. Επομένως, απαιτείται ρητή σηματοδότηση για την επιτυχή ανακάλυψή του από την συσκευή εκκίνησης της σύνδεσης MPTCP. Από την άλλη μεριά, σε περισσότερο ρεαλιστικά σενάρια ανάπτυξης πανομοιότυπων εφαρμογών συνιστάται η ρύθμιση διαφανής λειτουργίας, ώστε να αποφευχθεί η ανάγκη ειδικής παραμετροποίησης των συσκευών στην πλευρά του χρήστη, ιδιαίτερα σε περιπτώσεις κινητών κόμβων.



Σχήμα 16: Επίδοση MPTCP σε Mb/s με χρήση Reno και LIA

Σε αυτήν την ενότητα, παρουσιάζονται οι αποδόσεις, χαρακτηριστικά και ιδιαιτερότητες, τριών αλγορίθμων ελέγχου συμφόρησης. Το *Σχήμα 16* απεικονίζει την επίδοση του MPTCP κατά την λήψη του ίδιου αρχείου μεγέθους 100 MB για δύο διαφορετικές υλοποιήσεις του μηχανισμού. Αρχικά, το αριστερό γράφημα παρουσιάζει την επίδοση του Multipath TCP όταν χρησιμοποιείται ο κλασικός αλγόριθμος Reno του παραδοσιακού πρωτοκόλλου, ο οποίος δεν ενοποιείται στο επίπεδο MPTCP. Σε αυτήν την περίπτωση, κάθε υπο-ροή TCP πραγματοποιεί ξεχωριστό έλεγχο της διαδικτυακής κίνησης δίχως την δυνατότητα δυναμικής εκφόρτωσης δεδομένων ανάλογα με την κατάσταση συμφόρησης της εκάστοτε ζεύξης. Παρ' όλα αυτά, επιτυγχάνεται ο σχεδιαστικός στόχος του πρωτοκόλλου ως προς την προσφορά ενισχυμένης ανθεκτικότητας σε περίπτωση αστοχίας μεταφοράς των δεδομένων, αφού καθίσταται δυνατός ο επαναπρογραμματισμός μετάδοσης της διαδικτυακής κίνησης μέσω οποιαδήποτε διαθέσιμη διαδρομή δρομολόγησης. Όμως, η υλοποίηση μη-ενοποιημένων παραθύρων συμφόρησης τείνει να αυξήσει την «επιθετικότητα» του συνόλου των υπο-ροών μίας σύνδεσης MPTCP ως προς μία παραδοσιακή συνεδρία TCP, οδηγώντας με αυτόν τον τρόπο σε άδικο διαμοιρασμό της διαθέσιμης χωρητικότητας. Οι πολλαπλές υπο-ροές TCP επιδιώκουν την συσσώρευση μεγαλύτερου ποσοστού του διαθέσιμου εύρους ζώνης, μία κατάσταση που τείνει να παραβιάσει μία εκ των σχεδιαστικών αρχών του επεκταμένου πρωτοκόλλου, η οποία επιδιώκει την δίκαιη, ισόρροπη, κατανομή των δικτυακών πόρων ανάμεσα σε μία σύνδεση MPTCP και μία συνεδρία TCP.

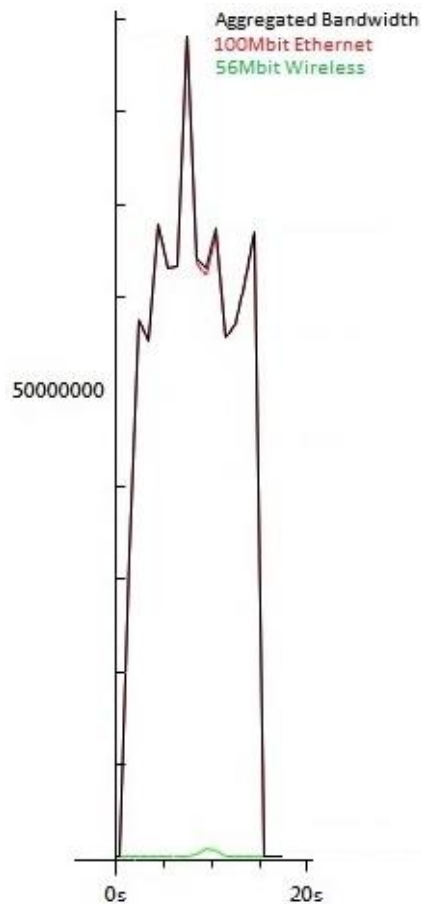
Μέτρηση	Τιμή			
	TCP Reno		LIA	
	Ethernet	Wireless	Ethernet	Wireless
Throughput	31.298 Mbps	3.555 Mbps	26.397 Mbps	2.467 Mbps
Bytes/Link	73192473	8310512	64254573	6004376
Packets/Link	81028	9339	72408	7009

Πίνακας 2: Αριθμητικές μετρήσεις σχήματος 16

Από την άλλη μεριά, η υλοποίηση ενός ενοποιημένου μηχανισμού ελέγχου συμφόρησης, όπως ο LIA (δεξί γράφημα του σχήματος), επιδιώκει την συσσώρευση δίκαιου ποσοστού των διαθέσιμων διαδικτυακών πόρων, ενώ, ταυτόχρονα, επιτρέπει την δυναμική εκφόρτωση δεδομένων μεταξύ των διαθέσιμων διαδρομών δρομολόγησης. Τα παραπάνω αποδεικνύονται τόσο από την μειωμένη απόδοση της εκάστοτε ζεύξης, η οποία οδηγεί σε αντίστοιχη μείωση της γενικότερης επίδοσης της σύνδεσης MPTCP, όσο και από τις αυξομειώσεις της ταχύτητας μεταφοράς δεδομένων στο ασύρματο περιβάλλον μετάδοσης, οι οποίες είναι κατά κύριο λόγο αντιστρόφως ανάλογες των αντίστοιχων της διεπαφής Ethernet. Κατ' αυτόν τον τρόπο, η συμπεριφορά του MPTCP ανταποκρίνεται πλήρως στο θεωρητικό υπόβαθρο που έχει τεθεί αναφορικά με τις σχεδιαστικές αρχές του πρωτοκόλλου και τον μηχανισμό ελέγχου συμφόρησης.

Το *Σχήμα 17*, το οποίο ακολουθεί απεικονίζει την επίδοση του MPTCP, όταν υλοποιείται ο μηχανισμός OLIA για τον έλεγχο της δικτυακής συμφόρησης. Κατά την διάρκεια λήψης του ίδιου αρχείου μεγέθους 100 MB, ο MPTCP

Proxy επιλέγει να χρησιμοποιήσει την διεπαφή Ethernet του Raspberry Pi για την μετάδοση των πακέτων πληροφορίας, ενώ η ασύρματη ζεύξη υποχρησιμοποιείται πλήρως. Λόγω της ταχύτερης και αποδοτικότερης σύγκλισης της βελτιωμένης έκδοσης του αλγορίθμου LIA στις εκάστοτε συνθήκες δρομολόγησης, η αυξημένη χωρητικότητα της ζεύξης Ethernet σε συνδυασμό με την αισθητά μικρότερη τιμή RTT συγκριτικά με το ασύρματο περιβάλλον μετάδοσης οδηγούν στην ταχύτερη αύξηση του παραθύρου συμφόρησης της αντίστοιχης υπο-ροής. Έτσι, η ασύρματη ζεύξη δεν προάγεται ως κατάλληλη, εναλλακτική διαδρομή δρομολόγησης για την εξισορρόπηση του συνολικού όγκου του φορτίου πληροφορίας ή/και την εκφόρτωση δεδομένων μεταξύ των διαθέσιμων μονοπατιών εν απουσία συμφόρησης στην ενσύρματη ζεύξη.



Σχήμα 17: Επίδοση MPTCP σε Mb/s με χρήση OLIA

Μέτρηση	Τιμή	
	Ethernet	Wireless
Throughput	63.019 Mbps	0.163 Mbps
Bytes/Link	107283619	277600
Packets/Link	89775	380

Πίνακας 3: Αριθμητικές μετρήσεις σχήματος 17

Παρ' όλα αυτά, κατά περίπου το 10^ο δευτερόλεπτο της μεταφοράς του αρχείου παρατηρούμε πως η ασύρματη ζεύξη τείνει να αυξήσει ελαφρώς το τοπικό παράθυρο συμφόρησης αντιδρώντας στην μείωση της επίδοσης της πρωταρχικής διαδρομής δρομολόγησης. Ακόμα, διαπιστώνεται πως η

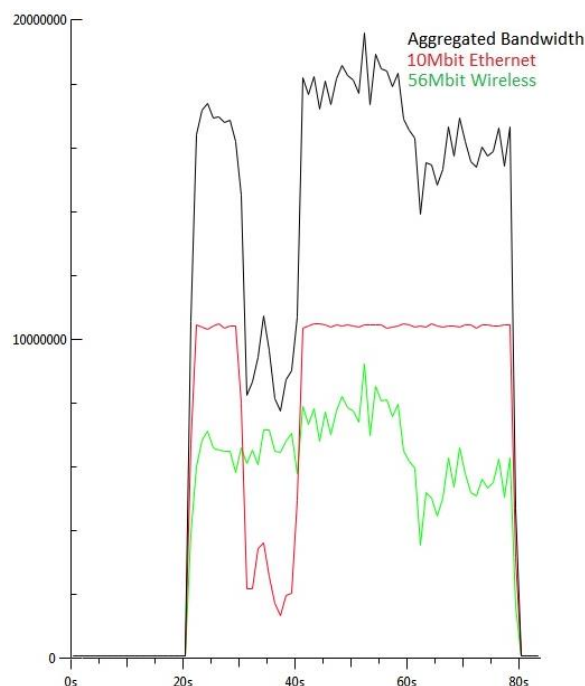
σύνδεση MPTCP λαμβάνει χωρητικότητα ίση με τουλάχιστον το διαθέσιμο εύρος ζώνης που θα κατανεμόταν στην αντίστοιχη παραδοσιακή συνεδρία TCP κατά την αξιοποίησης του αποδοτικότερου εκ των διαθέσιμων μονοπατιών δρομολόγησης.

5.3. Δυναμική Εκφόρτωση Δεδομένων

Αφού υποβαθμίσουμε την διεπαφή Ethernet του Raspberry Pi στα 10 Mb/s, επανεκτιμούμε την επίδοση του Multipath TCP. Σε αυτό το πειραματικό σενάριο, επίσης, επάγουμε τεχνητή συμφόρηση υπό την μορφή δεδομενογραμμμάτων UDP σε ρυθμό της τάξης των 9 Mbps με σκοπό την επαλήθευση και παρακολούθηση του δυναμικού επαναπρογραμματισμού μετάδοσης της διαδικτυακής κίνησης μέσω περισσότερο αξιόπιστων μονοπατιών δρομολόγησης.

5.3.1. TCP Reno

Πρώτα απ' όλα, παρουσιάζεται η επίδοση και η συμπεριφορά των υπο-ροών TCP, όταν ο έλεγχος συμφόρησης πραγματοποιείται υλοποιώντας τον κλασσικό αλγόριθμο Reno. Λόγω της μη-ενοποίησης του μηχανισμού στο επίπεδο MPTCP, όπως έχει ήδη τονισθεί, κάθε υπο-ροή διεξάγει ξεχωριστό έλεγχο της διαδικτυακής κίνησης δίχως την δυνατότητα δυναμικής εκφόρτωσης δεδομένων από συμφορημένα μονοπάτια σε περισσότερο αξιόπιστες διαδρομές δρομολόγησης. Συνεπώς, τυχόν αυξομειώσεις στην επίδοση της εκάστοτε υπο-ροής TCP εξαρτώνται αποκλειστικά από την κατάσταση συμφόρησης της τοπικής ζεύξης.



Σχήμα 18: Επίδοση MPTCP σε Mb/s υπό συμφόρηση με χρήση Reno

Η υποβάθμιση της ενσύρματης ζεύξης παροτρύνει τον MPTCP Proxy να δρομολογήσει πακέτα ταυτόχρονα μέσω του ασύρματου μονοπατιού μεταφοράς δεδομένων επιδιώκοντας την συσσώρευση πόρων και την επίτευξη υψηλότερων ρυθμών μετάδοσης. Παρ' όλα αυτά, κατά την εκκίνηση της συνεδρίας MPTCP η ενσύρματη ζεύξη δυναμικότητας 10 Mbps υπερτερεί της ασύρματης διαδρομής των 56 Mbps. Η συγκεκριμένη επίδοση είναι δυνατό να οφείλεται στην τυχαιότητα του ασύρματου περιβάλλοντος μετάδοσης, η οποία μπορεί να οδηγήσει σε απρόσμενες απώλειες πακέτων, σε συνδυασμό με την σαφώς μικρότερη τιμή RTT της ζεύξης Ethernet.

Εν προκειμένω, τα δεδομενογράμματα UDP επάγονται στην ζεύξη με κατεύθυνση την διεπαφή Ethernet του Raspberry Pi. Όπως απεικονίζεται στο Σχήμα 18 και περιγράφεται αναλυτικά στον Πίνακα 4, η συμφόρηση που λαμβάνει χώρα στην προαναφερθείσα διαδρομή τείνει να μειώσει κατακόρυφα την επίδοση της αντίστοιχης υπο-ροής. Όμως, ο παραπάνω υποβιβασμός δεν οδηγεί σε εκφόρτωση δεδομένων στην ασύρματη ζεύξη και σε κατ' επέκταση αύξηση της επίδοσής της. Αντίθετα, οι συμπεριφορές των αντίστοιχων διαδρομών στα δύο περιβάλλοντα διάδοσης παραμένουν πλήρως ασυσχέτιστες. Το παραπάνω αποδεικνύεται και από τις τυχαίες, ραγδαίες εναλλαγές της ταχύτητας μετάδοσης που επιτυγχάνεται κατά μήκος της ασύρματης διαδρομής μεταφοράς δεδομένων μετά το πέρας της επαγόμενης συμφόρησης.

Μέτρηση	Τιμή			
	Κανονική Περίοδος		Περίοδος Συμφόρησης	
	Ethernet	Wireless	Ethernet	Wireless
Throughput	10.34 Mbps	6.5 Mbps	3.330 Mbps	6.460 Mbps
Bytes/Link	60194925	37853028	4736002	9168576
Packets/Link	72482	45172	5677	11240

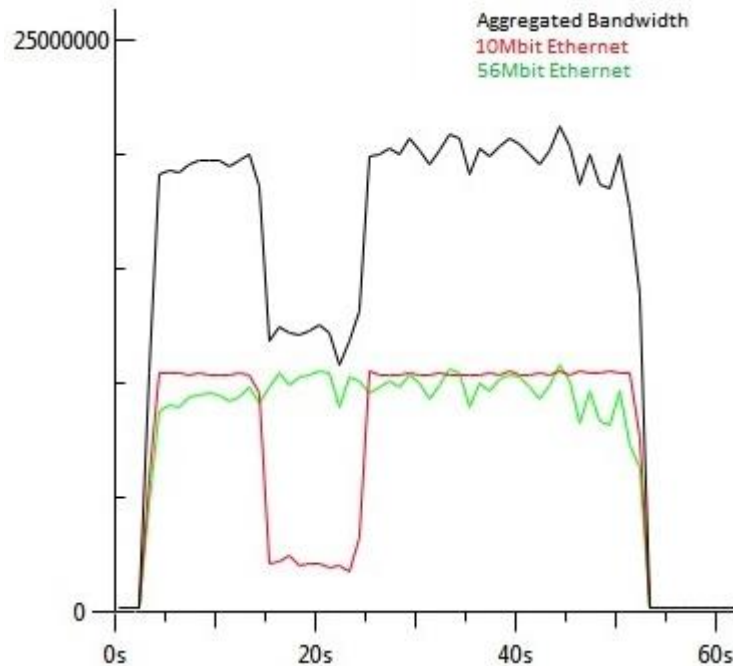
Πίνακας 4: Αριθμητικές μετρήσεις σχήματος 18

5.3.2. Linked Increase Algorithm

Η χρησιμοποίηση αλγορίθμων ελέγχου συμφόρησης, οι οποίοι ενοποιούνται στο στρώμα MPTCP, τείνει να προσφέρει βέλτιστη απόδοση σε συνδυασμό με την δυνατότητα δυναμικής εκφόρτωσης δεδομένων μέσω των διαθέσιμων ενσύρματων ή/και ραδιοηλεκτρικών διεπαφών. Το Σχήμα 19 επιβεβαιώνει τα παραπάνω, ενώ παράλληλα, απεικονίζει την αποτελεσματική συγκέντρωση πόρων που δύναται να επιτύχει το επεκταμένο πρωτόκολλο.

Μέτρηση	Τιμή			
	Κανονική Περίοδος		Περίοδος Συμφόρησης	
	Ethernet	Wireless	Ethernet	Wireless
Throughput	10.106 Mbps	9.292 Mbps	2.585 Mbps	9.973 Mbps
Bytes/Link	49410687	45549904	3347240	12913610
Packets/Link	56293	55600	3958	16003

Πίνακας 5: Αριθμητικές μετρήσεις σχήματος 19



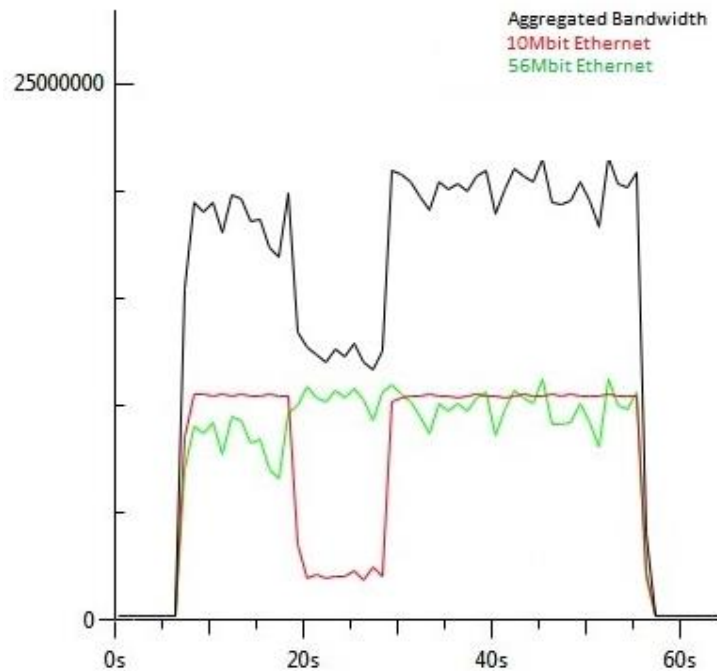
Σχήμα 19: Επίδοση MPTCP σε Mb/s υπό συμφόρηση με χρήση LIA

Πιο συγκεκριμένα, κατά την περίοδο συμφόρησης, παρά την γενικότερη υποβάθμιση της επίδοσης της σύνδεσης MPTCP, η οποία αποδεικνύεται αναπόφευκτη, παρατηρείται η επιτυχής ανακατανομή των διαθέσιμων πόρων. Το MPTCP τείνει να δρομολογήσει ελαφρώς περισσότερο όγκο δεδομένων μέσω της ασύρματης ζεύξης επιδιώκοντας να ανακουφίσει έτσι το συμφορημένο μονοπάτι. Ως αποτέλεσμα, παρέχεται ενισχυμένη ανθεκτικότητα και ποιότητα εμπειρίας ενάντια σε τυχόν δυσλειτουργίες των εμπλεκόμενων ζεύξεων, οι οποίες υπό διαφορετικές συνθήκες θα ήταν δυνατό να οδηγήσουν σε αισθητή υποβάθμιση των παρεχόμενων υπηρεσιών.

5.3.3. Opportunistic Linked Increase Algorithm

Η χρήση της βελτιωμένης έκδοσης του ειδικά σχεδιασμένου μηχανισμού ελέγχου συμφόρησης του Multipath TCP συνεπάγεται ακόμα αποτελεσματικότερη κατανομή των διαθέσιμων δικτυακών πόρων, ενώ ταυτόχρονα, τείνει να άρει την πιθανότητα άδικου διαμοιρασμού της προσφερόμενης χωρητικότητας σε περίπτωση διάσχισης της αντίστοιχης ζεύξης από πολλαπλές υπο-ροές MPTCP και συνδέσεις παραδοσιακού TCP. Το τελευταίο αποδεικνύεται γραφικά από το Σχήμα 20, όπου απεικονίζεται η εκ των προτέρων μειωμένη επίδοση του ασύρματου μονοπατιού μετάδοσης συγκριτικά με το πείραμα της προηγούμενης ενότητας. Αναλυτικότερα, ο OLIA, αναφορικά με τον προκάτοχό του (LIA), τείνει να δρομολογήσει ακόμα μικρότερο ποσοστό πακέτων μέσω διαδρομών περιορισμένης απόδοσης, περιορίζοντας κατ' επέκταση την μεταφορά της διαδικτυακής κίνησης κατά μήκος λιγότερο αξιόπιστων ζεύξεων. Ως αποτέλεσμα, ελαττώνεται η «επιθετικότητα» του συνόλου των υπο-ροών TCP, ενώ παράλληλα, τείνουν να αξιοποιηθούν οι αποδοτικότερες και περισσότερες

αξιόπιστες διαδρομές έχοντας ως συνεπακόλουθο την βελτιωμένη επίδοση του επεκταμένου πρωτοκόλλου.



Σχήμα 20: Επίδοση MPTCP σε Mb/s υπό συμφόρηση με χρήση OLIA

Μέτρηση	Τιμή			
	Κανονική Περίοδος		Περίοδος Συμφόρησης	
	Ethernet	Wireless	Ethernet	Wireless
Throughput	10.21 Mbps	9.027 Mbps	2.439 Mbps	10.165 Mbps
Bytes/Link	49890753	45871942	2968488	12369856
Packets/Link	60030	53361	3490	15160

Πίνακας 6: Αριθμητικές μετρήσεις σχήματος 20

Επίσης, το παραπάνω γράφημα αποδεικνύει την αποτελεσματικότερη προσαρμογή του αλγορίθμου στην μεταβαλλόμενη κατάσταση συμφόρησης των εμπλεκόμενων ζεύξεων. Όπως απεικονίζεται, η επίδοση της ασύρματης ζεύξης αυξάνεται κατακόρυφα μέχρι να λάβει την μέγιστη τιμή της αντιδρώντας στην επαγόμενη συμφόρηση στο ενσύρματο μονοπάτι μετάδοσης. Η βέλτιστη σύγκλιση του μηχανισμού οφείλεται στην κλασική δυνατότητα επιθεώρησης των τιμών RTT, όσο και στην αδιάκοπη παρακολούθηση του πλήθους των συνεχόμενων μεταδιδόμενων bits μετά την παρατήρηση συμφόρησης στην αντίστοιχη ζεύξη.

5.4. Επιπτώσεις Χωρητικότητας TCP Buffers

Όπως έχει ήδη τονισθεί στην ενότητα 2.2.3, ένα από τα κεντρικά ζητήματα επίδοσης του Multipath TCP συγκριτικά με το παραδοσιακό πρωτόκολλο, αποτελεί η πιθανή εισαγωγή επιπλέον καθυστέρησης στον δέκτη λόγω της ανάγκης αναδιάταξης των ληφθέντων πακέτων, τα οποία είναι πιθανό να προέρχονται από μονοπάτια ποικίλων καθυστερήσεων και διαφορετικών

επιδόσεων. Ως εκ τούτου, καθίσταται απαραίτητη η προσωρινή αποθήκευση ενός σεβαστού αριθμού πακέτων με σκοπό την αναδιοργάνωσή τους και την εν σειρά τελική παράδοσή τους στο Στρώμα Εφαρμογής. Η κατανομή μεγαλύτερου ποσοστού μνήμης κρίνεται αναγκαία για την αποφυγή υποβάθμισης της επίδοσης μίας σύνδεσης MPTCP, η οποία είναι δυνατό να υποφέρει από απόρριψη των ληφθέντων πακέτων στον παραλήπτη ως αποτέλεσμα της περιορισμένης χωρητικότητας των χώρων προσωρινής αποθήκευσης δεδομένων.

Στο παρόν πειραματικό σενάριο, διατηρούμε την χωρητικότητα του send buffer στον MPTCP Proxy αμετάβλητη, ενώ μεταβάλλουμε το μέγεθος του receive buffer στο Raspberry Pi, στο οποίο, ως ο τελικός προορισμός των δεδομένων, λαμβάνει χώρα η απαραίτητη αναδιάταξη των πακέτων πληροφορίας. Η τροποποίηση του αποθηκευτικού χώρου λήψης πραγματοποιείται με την παρακάτω εντολή, η οποία, στην συγκεκριμένη περίπτωση, καταδεικνύει τις προεπιλεγμένες τιμές παραμετροποίησης του RPi:

```
> sysctl -w net.ipv4.tcp_rmem 4096 87380 1499264
```

Η πρώτη τιμή ορίζει το ελάχιστο μέγεθος buffer (σε bytes), το οποίο ο kernel υποχρεούται να καταναίμει στην εκάστοτε σύνδεση TCP ακόμα και υπό συνθήκες υψηλής κατανάλωσης μνήμης. Η δεύτερη τιμή υποδεικνύει την προεπιλεγμένη χωρητικότητα buffer που δύναται να κατανεμηθεί σε κάθε TCP socket, ενώ η τελευταία τιμή καθορίζει το μέγιστο μέγεθος χώρου προσωρινής αποθήκευσης που είναι δυνατό να οριστεί.

Στο προκείμενο, η αξιολόγηση της επίδοσης του MPTCP πραγματοποιείται για ταχύτητα της διεπαφής Ethernet του Raspberry Pi της τάξης των 100 Mbps και τρεις (3) διαφορετικές παραμετροποιήσεις του αποθηκευτικού χώρου λήψης, μία εκ των οποίων είναι η παραπάνω προεπιλογή. Στην συνέχεια, παρουσιάζονται οι δύο εναπομείναντες ρυθμίσεις:

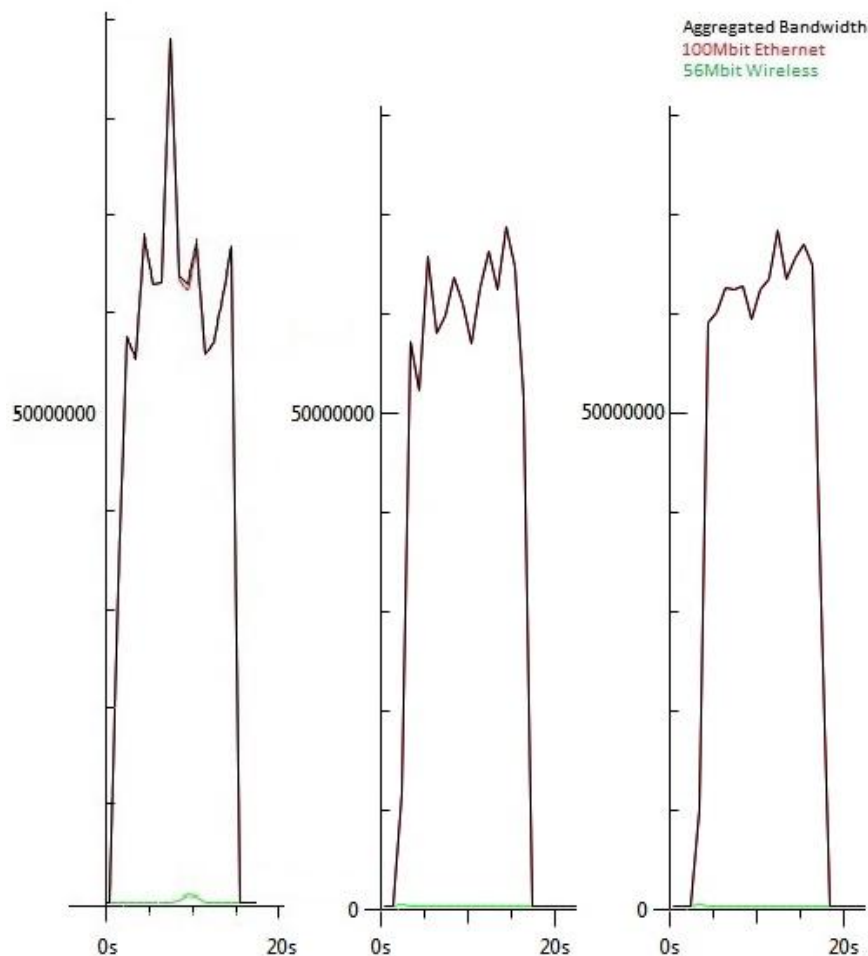
```
> sysctl -w net.ipv4.tcp_rmem 2048 43690 749632  
> sysctl -w net.ipv4.tcp_rmem 1024 21845 374816
```

Σε αυτό το σημείο, αξίζει να σημειωθεί πως η θεωρητικά συνιστώμενη χωρητικότητα του TCP buffer λήψης υπό τις παρούσες συνθήκες μετάδοσης είναι:

$$2 * \left(\frac{100 * 10^6}{8} + \frac{56 * 10^6}{8} \right) * 100 * 10^{-3} = 3,7 \text{ MB}$$

Συγκρίνοντας την παραπάνω προκύπτουσα τιμή της απαιτούμενης χωρητικότητας του αποθηκευτικού χώρου λήψης του Raspberry Pi με την προεπιλεγμένη παραμετροποίηση της πλατφόρμας, διαπιστώνουμε πως η υπάρχουσα ρύθμιση υστερεί της θεωρητικά αναμενόμενης τιμής. Συνεπώς, το τερματικό δεν κατέχει την απαραίτητη χωρητικότητα για την προσωρινή αποθήκευση πακέτων με σκοπό την αναδιάταξή τους και την επιτυχή ανταπόκριση της συσκευής στην εισαγόμενη καθυστέρηση.

Το Σχήμα 21 απεικονίζει την επίδοση του Multipath TCP για τις τρεις (3) παραπάνω παραμετροποιήσεις του buffer λήψης, αντίστοιχα. Όπως φαίνεται από το πρώτο γράφημα, ακόμα και η προεπιλεγμένη κατανομή μνήμης για την παρούσα σύνδεση δεν είναι αρκετή οδηγώντας σε αυξομειώσεις της επίδοσης ως αποτέλεσμα της αναπόφευκτης απόρριψης των ληφθέντων πακέτων, τα οποία δεν καθίσταται δυνατό να αποθηκευτούν προσωρινά μέχρι την λήψη των επόμενων αναμενόμενων τμημάτων βάσει των αριθμών ακολουθίας της σύνδεσης MPTCP. Βέβαια, στο παρόν σενάριο, η ασύρματη διαδρομή δρομολόγησης υποχρησιμοποιείται πλήρως. Επομένως, η αποσταθεροποίηση της συνεδρίας MPTCP είναι δυνατό να οφείλεται απλούστατα στην περιορισμένη δυναμικότητα του Raspberry Pi, το οποίο δυσκολεύεται να ανταπεξέλθει στην ροή της πληροφορίας λόγω της αυξημένης χωρητικότητας της ενσύρματης ζεύξης.



Σχήμα 21: Επίπτωση χωρητικότητας TCP recv buffer

Μέτρηση	Τιμή					
	Config. #1		Config. #2		Config. #3	
	<u>Eth.</u>	<u>Wir.</u>	<u>Eth.</u>	<u>Wir.</u>	<u>Eth.</u>	<u>Wir.</u>
Throughput	63.019	0.163	61.072	0.020	62.911	0.021
Bytes/Link	107283619	277600	107193631	34616	107193201	34992
Packets/Link	89775	380	85683	56	85502	56

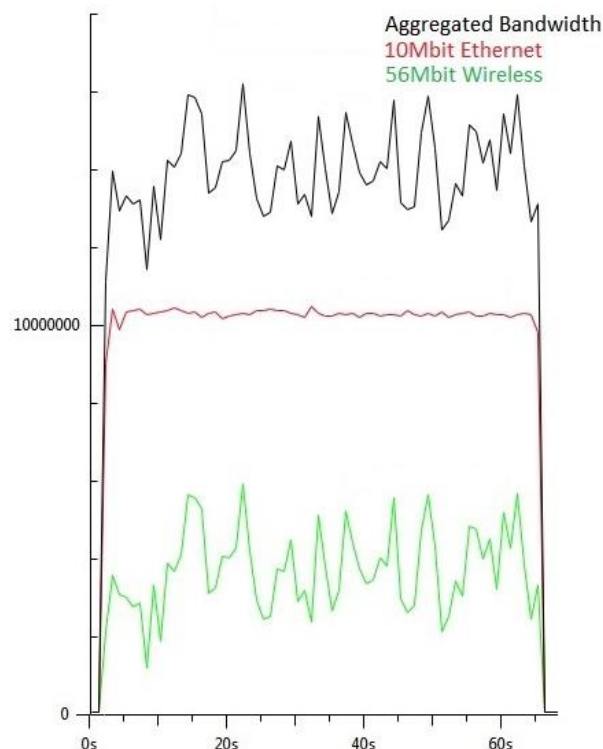
Πίνακας 7: Αριθμητικές μετρήσεις σχήματος 21

Εξετάζοντας τα επόμενα δύο γραφήματα, διαπιστώνεται η μείωση της μέγιστης επίδοσης της σύνδεσης MPTCP για ελάττωση κατά το ήμισυ της χωρητικότητας του buffer λήψης. Στην πρώτη εκ των δύο απεικονίσεων, οι αυξομειώσεις της επίδοσης τείνουν να επιτύχουν ελαφρώς υψηλότερη ταχύτητα συγκριτικά με την αντίστοιχη απόδοση του τελευταίου γραφήματος. Παρ' όλα αυτά, αναμενόταν περαιτέρω μείωση της επίδοσης του τρίτου γραφήματος για περαιτέρω μείωση της χωρητικότητας του αποθηκευτικού χώρου λήψης εξαιτίας της προσδόκιμης απόρριψης μεγάλου αριθμού ληφθέντων πακέτων.

Σε αυτό το σημείο, επανεξετάζουμε την επίδοση του MPTCP υποβιβάζοντας την ταχύτητα της διεπαφής Ethernet του Raspberry Pi στα 10 Mbps. Ως εκ τούτου, η θεωρητικά αναμενόμενη χωρητικότητα του buffer λήψης ανέρχεται σε:

$$2 * \left(\frac{10 * 10^6}{8} + \frac{56 * 10^6}{8} \right) * 100 * 10^{-3} = 1,57 \text{ MB}$$

Η συγκεκριμένη τιμή δεν αποκλίνει σε μεγάλο βαθμό από την μέγιστη προεπιλεγμένη τιμή παραμετροποίησης του έξυπνου τερματικού, η οποία ισούται με 1,43 MB. Όπως απεικονίζεται και στο Σχήμα 22, η επίδοση της ζεύξης Ethernet παραμένει σταθερή στην μέγιστη τιμή της, ενώ η δεύτερη υπο-ροή υποφέρει από συνεχόμενες, ραγδαίες αυξομειώσεις της προσφερόμενης ταχύτητάς της, οι οποίες είναι δυνατό να οφείλονται κατά κύριο λόγο σε τυχόν παρεμβολές στο ασύρματο περιβάλλον μετάδοσης και σε μικρότερο ποσοστό στην πιθανή ανεπάρκεια διαθέσιμου χώρου αποθήκευσης πακέτων.

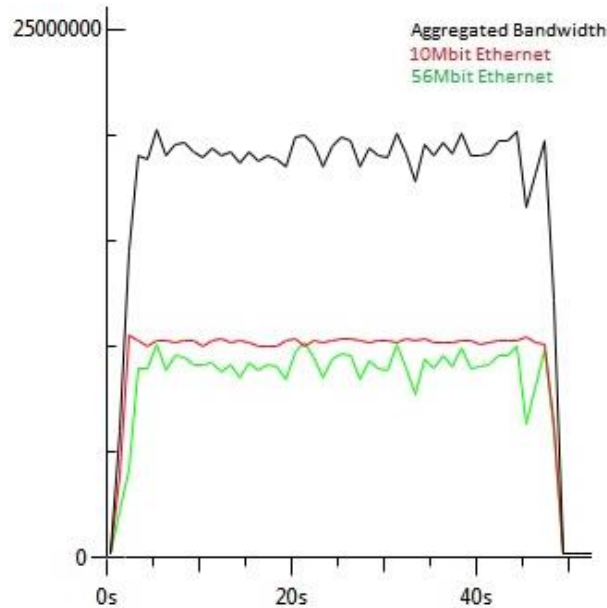


Σχήμα 22: Επίδοση MPTCP σε Mb/s συναρτήσει TCP recv buffer size (1)

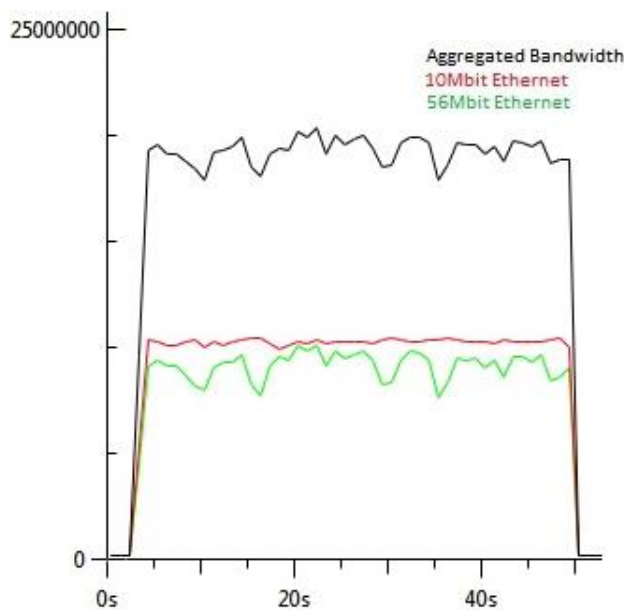
Μέτρηση	Τιμή	
	Ethernet	Wireless
Throughput	10.233 Mbps	3.667 Mbps
Bytes/Link	81642919	29329434
Packets/Link	90747	36233

Πίνακας 8: Αριθμητικές μετρήσεις σχήματος 22

Στην συνέχεια, παρουσιάζεται η επίδοση που επιτυγχάνει το επεκταμένο πρωτόκολλο για τις δύο εναπομείναντες, προαναφερθείσες κατανομές διαθέσιμης μνήμης:



Σχήμα 23: Επίδοση MPTCP σε Mb/s συναρτήσει TCP recv buffer size (2)



Σχήμα 24: Επίδοση MPTCP σε Mb/s συναρτήσει TCP recv buffer size (3)

Σε γενικές γραμμές, παρατηρείται περισσότερο σταθερή επίδοση της ασύρματης υπο-ροής TCP συγκριτικά με το Σχήμα 22, τα αίτια της οποίας

καθίστανται απροσδιόριστα εξαιτίας της τυχαιότητας του αντίστοιχου περιβάλλοντος μετάδοσης. Παρ' όλα αυτά, το Σχήμα 24 απεικονίζει πιο απότομες εναλλαγές στην επίδοση κατά μήκος του ασύρματου μονοπατιού μετάδοσης συγκριτικά με το Σχήμα 23, οι οποίες μπορεί να οφείλονται σε συνδυασμό απωλειών πακέτων και κατ' επέκταση αναμεταδόσεων ως συνέπεια παρεμβολών στο μέσο μετάδοσης, καθώς και της αδυναμίας προσωρινής αποθήκευσης περαιτέρω όγκου δεδομένων.

Μέτρηση	Τιμή	
	<u>Ethernet</u>	<u>Wireless</u>
Throughput	10.067 Mbps	8.851
Bytes/Link	59121153	51968738
Packets/Link	62428	65175

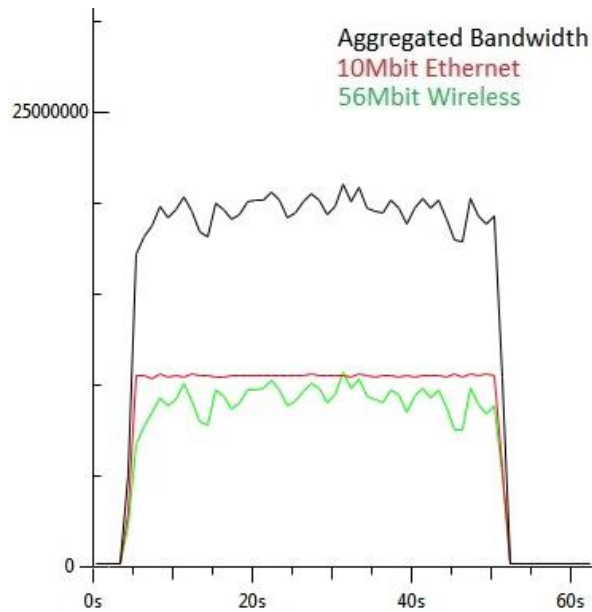
Πίνακας 9: Αριθμητικές μετρήσεις σχήματος 23

Μέτρηση	Τιμή	
	<u>Ethernet</u>	<u>Wireless</u>
Throughput	10.115 Mbps	8.995 Mbps
Bytes/Link	58783717	52266634
Packets/Link	63000	65465

Πίνακας 10: Αριθμητικές μετρήσεις σχήματος 24

5.5. Επιπτώσεις Μεγέθους MTU

Επεκτείνοντας τον συλλογισμό της προηγούμενης ενότητας, το παρόν πείραμα αξιολόγησης επικεντρώνεται στις επιπτώσεις του μεγέθους της μέγιστης μεταδιδόμενης μονάδας (Maximum Transmitted Unit - MTU) στην επίδοση του MPTCP. Ο συνδυασμός της συγκεκριμένης παραμέτρου με την διαθέσιμη χωρητικότητα του buffer λήψης αποτελεί έναν εκ των καταλυτικότερων παραμέτρων επηρεασμού της συμπεριφοράς μίας σύνδεσης με την δυνατότητα αξιοποίησης πολλαπλών διαδρομών. Αναλυτικά, όπως παρουσιάστηκε λεπτομερώς στην προηγούμενη υπο-ενότητα, η χωρητικότητα του αποθηκευτικού χώρου λήψης δεδομένων είναι άρρηκτα συνδεδεμένη με την δυνατότητα επιτυχής αναδιάταξης των ληφθέντων πακέτων, τα οποία προέρχονται από διαδρομές ποικίλων καθυστερήσεων και κατ' επέκταση λαμβάνονται εκτός της αναμενόμενης ακολουθίας, αποτρέποντας με αυτόν τον τρόπο την άμεση παράδοσή τους στο Στρώμα Εφαρμογής. Επομένως, όπως είναι λογικό, η μείωση του μεγέθους της μέγιστης μεταδιδόμενης μονάδας για σταθερή χωρητικότητα του *receive buffer* τείνει να εισάγει περαιτέρω καθυστέρηση στον δέκτη, καθώς απαιτείται ο έλεγχος και η αναδιάταξη ακόμα μεγαλύτερου αριθμού πακέτων. Από την άλλη μεριά, μεγαλύτερα μεγέθη MTU οδηγούν στην επίτευξη υψηλότερων επιδόσεων λόγω της ανάγκης διαχείρισης και αναδιοργάνωσης μειωμένου όγκου ληφθέντων πακέτων. Σε αυτό το σημείο, αξίζει να σημειωθεί πως η αξιολόγηση της επίδοσης του επεκταμένου πρωτοκόλλου συναρτήσκει της χωρητικότητας του αποθηκευτικού χώρου λήψης πραγματοποιήθηκε για σταθερό μέγεθος MTU των 1500 bytes.



Σχήμα 25: Επίδοση MPTCP σε Mb/s για MTU 1200

Στο παρόν πειραματικό σενάριο, εξετάζεται η επίδοση του MPTCP για δύο (2) διαφορετικές τιμές της μέγιστης μεταδιδόμενης μονάδας, 1200 και 1000 bytes, αντίστοιχα, και την προεπιλεγμένη παραμετροποίηση του buffer λήψης. Το Σχήμα 25 απεικονίζει την επιτευχθείσα επίδοση του πρωτοκόλλου για το πρώτο εκ των προαναφερθέντων μεγεθών MTU. Εν γένει, η συμπεριφορά τόσο της ενσύρματης, όσο και της ασύρματης υπο-ροής δεν αποκλίνει από την αναμενόμενη. Η ταχύτητα της ζεύξης Ethernet διατηρείται σταθερή κατά την διάρκεια μεταφοράς του ζητούμενου αρχείου, ενώ η επίδοση του ασύρματου μονοπατιού μετάδοσης υποφέρει από μικρές διαλείψεις, οι οποίες πιθανολογικά προέρχονται από τυχόν παρεμβολές κατά μήκος της διαδρομής δρομολόγησης.

Μέτρηση	Τιμή	
	<u>Ethernet</u>	<u>Wireless</u>
Throughput	10.364 Mbps	9.026 Mbps
Bytes/Link	60596961	52769088
Packets/Link	90186	73148

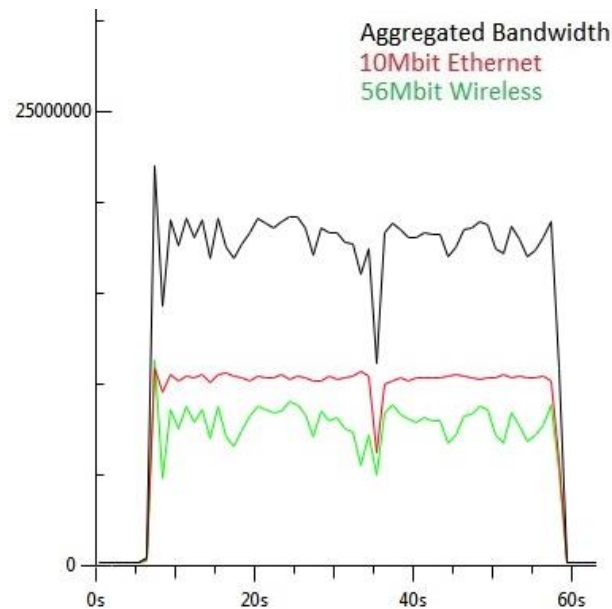
Πίνακας 11: Αριθμητικές μετρήσεις σχήματος 25

Μέτρηση	Τιμή	
	<u>Ethernet</u>	<u>Wireless</u>
Throughput	10.110 Mbps	7.745 Mbps
Bytes/Link	65280791	49993390
Packets/Link	105933	88425

Πίνακας 12: Αριθμητικές μετρήσεις σχήματος 26

Αντιθέτως, η περαιτέρω ελάττωση του μεγέθους της μέγιστης μεταδιδόμενης μονάδας στα 1000 bytes έχει ως συνεπακόλουθο την μείωση της επίδοσης της σύνδεσης MPTCP, όπως παρουσιάζεται στο Σχήμα 26. Σε αυτήν την περίπτωση, παρατηρείται επιπλέον αποσταθεροποίηση της επίδοσης της ασύρματης ζεύξης, ενώ και η ενσύρματη σύνδεση με την σειρά της εμφανίζει ισχνές διακυμάνσεις της προσφερόμενης ταχύτητας. Η παραπάνω υποβαθμισμένη συμπεριφορά μπορεί επίσης να αποδειχθεί με απλούστερο τρόπο εξετάζοντας τον απαιτούμενο χρόνο μεταφοράς του ζητούμενου αρχείου. Τέλος, άξια παρατήρησης αποτελεί η απότομη μείωση της επίδοσης της ζεύξης Ethernet μετά το 35^ο

δευτερόλεπτο της μετάδοσης, η οποία πιθανότατα είναι συνέπεια του αυξημένου αριθμού μεταδιδόμενων πακέτων σε συνδυασμό με την ανεπάρκεια χώρου προσωρινής αποθήκευσης δεδομένων. Η πλήρης χρησιμοποίηση του buffer λήψης ως συνεπακόλουθο του *head-of-line blocking* έχει ως αντίκτυπο την εξ αναγκασμού απόρριψη πακέτων στον παραλήπτη, η οποία οδηγεί κατ' επέκταση σε αναπόφευκτη μείωση της επίδοσης της σύνδεσης MPTCP.



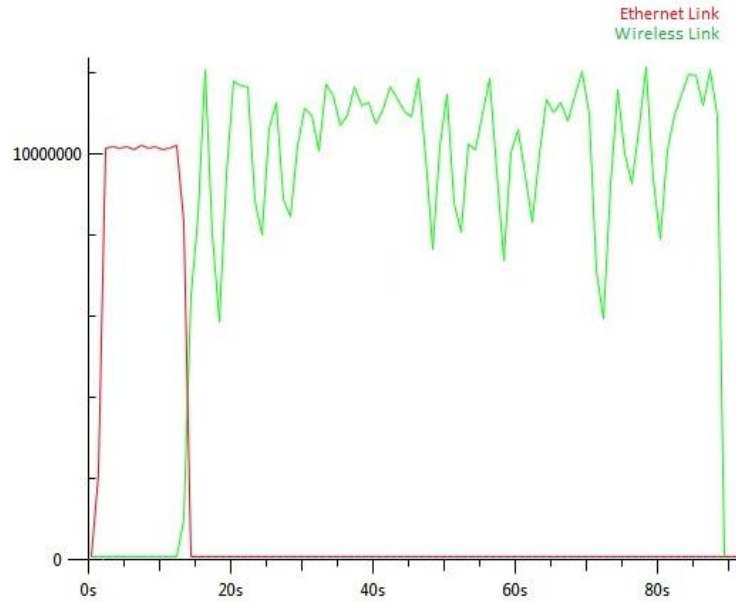
Σχήμα 26: Επίδοση MPTCP σε Mb/s για MTU 1000

5.6. Κάθετη Διαπομπή

Η παρούσα ενότητα αποσκοπεί στην αξιολόγηση της ανταπόκρισης του MPTCP σε περίπτωση συμβάντος διαπομπής μεταξύ των διαθέσιμων τεχνολογιών πρόσβασης στο Διαδίκτυο. Η διεξαγωγή του πειράματος βασίζεται στην ιεράρχηση των συνδέσεων TCP μέσω διαχωρισμού των εμπλεκόμενων ζεύξεων ως ενεργών ή εφεδρικών. Εν προκειμένω, η πρωταρχική υπο-ροή TCP εγκαθίσταται χρησιμοποιώντας την ενσύρματη διεπαφή του Raspberry Pi ταχύτητας 10 Mbps, ενώ η ασύρματη ζεύξη ρυθμίζεται ως εφεδρική.

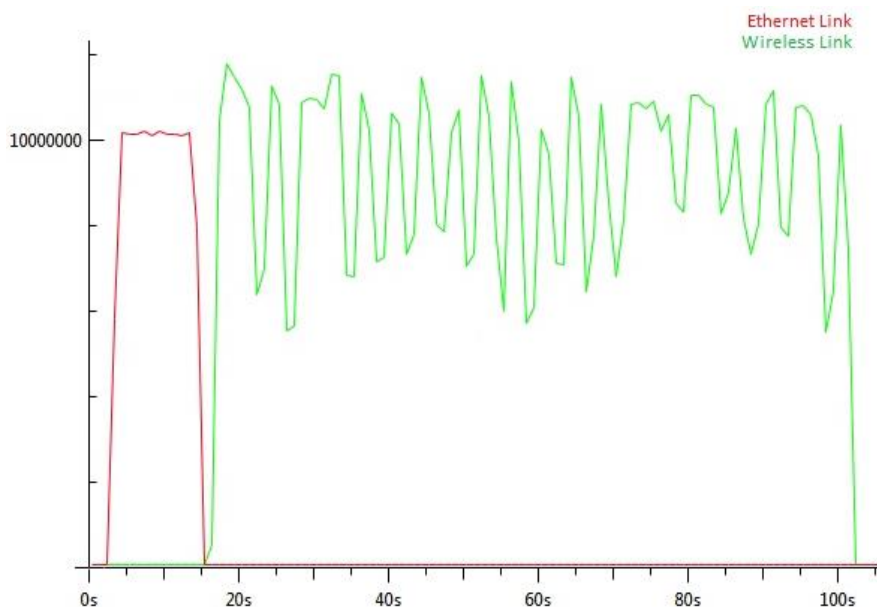
Κατά την διάρκεια μεταφοράς ενός αρχείου μεγέθους 100 MB απενεργοποιούμε την ζεύξη που αντιστοιχίζεται στην ενσύρματη διεπαφή του Raspberry Pi μέσω κατάλληλων σημάτων ελέγχου του MPTCP και παρατηρούμε την συμπεριφορά του επεκταμένου πρωτοκόλλου ως προς τον επαναπρογραμματισμό μετάδοσης των εναπομεινάντων πακέτων μέσω της ασύρματης διαδρομής δρομολόγησης. Εξετάζοντας το Σχήμα 27, διαπιστώνεται η ενεργοποίηση της ασύρματης ζεύξης, η οποία λαμβάνει χώρα πριν την εξ ολοκλήρου παύση μετάδοσης δεδομένων μέσω του ενσύρματου μονοπατιού. Η εναλλαγή της διαδρομής δρομολόγησης συνεπάγεται την επιβεβαίωση λήψης (ACK) των ήδη σταλμένων δεδομένων από το στρώμα MPTCP πριν την αποδέσμευση της αντίστοιχης ζεύξης. Ως εκ τούτου, το MPTCP επιτυγχάνει την μετατόπιση της διαδικτυακής κίνησης

και την εν τέλει δρομολόγησή της με προορισμό την ασύρματης διεπαφή του έξυπνου τερματικού αποφεύγοντας τόσο τυχόν απώλειες πακέτων, όσο και την εισαγωγή επιπρόσθετου φορτίου σηματοδοσίας ως απόρροια ανάγκης συγχρονισμού νέων συνδέσεων.



Σχήμα 27: Κάθετη διακοπή

Στην συνέχεια, επανεξετάζουμε την συμπεριφορά του Multipath TCP υπό ένα σενάριο χειρότερης περίπτωσης (worst-case scenario). Κατά την μετάδοση του ίδιου αρχείου μεγέθους 100 MB αποσυνδέουμε το καλώδιο Ethernet από το Raspberry Pi προκειμένου να προσομοιώσουμε απότομη απώλεια της σύνδεσης και να αξιολογήσουμε εκ νέου την ανταπόκριση του πρωτοκόλλου. Όπως απεικονίζεται στο Σχήμα 28, ακόμη και εν απουσία κατάλληλης σηματοδοσίας για την ομαλή επίτευξη διακοπής το MPTCP είναι σε θέση να ανακάμψει ταχέως και να ενεργοποιήσει την εφεδρική ζεύξη με στόχο την συνέχιση της μεταφοράς δεδομένων.



Σχήμα 28: Κάθετης διακοπή εν απουσία κατάλληλης σηματοδοσίας

Ο χρόνος απόκρισης του MPTCP Proxy ανέρχεται σε σχεδόν μισό δευτερόλεπτο. Η διαπίστωση δυσλειτουργίας σε οποιοδήποτε μέρος της σύνδεσης έχει ως συνεπακόλουθο την αποστολή ενός πακέτου RST από τον MPTCP Proxy ως σήμα επικείμενου τερματισμού της αντίστοιχης υπο-ροής TCP και την αναμονή επιβεβαίωσης από το απομακρυσμένο άκρο. Ύστερα από περίπου 1,7 δευτερόλεπτα αδράνειας, η κατάσταση της ζεύξης ερμηνεύεται ως αποτυχία σύνδεσης. Ως αποτέλεσμα, η ροή δεδομένων τείνει να δρομολογηθεί μέσω του ασύρματου περιβάλλοντος μετάδοσης, ενώ ταυτόχρονα, πραγματοποιείται η αναμετάδοση των ανεπιβεβαίωτων πακέτων ως απόρροια της απώλειάς τους εξαιτίας της διαταραγμένης αποκοπής του ενσύρματου μονοπατιού δρομολόγησης.

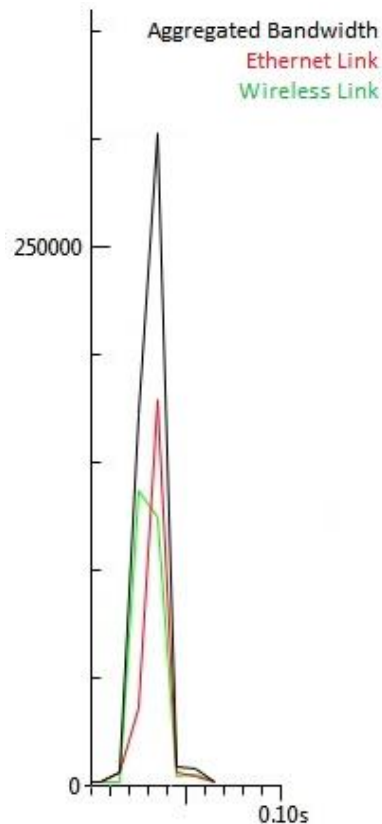
5.7. Bandwidth Throttling

Η τεχνική *bandwidth throttling*, η οποία έχει υλοποιηθεί θεωρητικά και περιγραφεί αναλυτικά στα κεφάλαια 3 και 4 του κειμένου, έχει ως στόχο τον περιορισμό του όγκου δεδομένων που λαμβάνονται/αποστέλλονται από κάποιον διαδικτυακό κόμβο. Ως αποτέλεσμα, η συγκεκριμένη τεχνική είναι δυνατό να χρησιμοποιηθεί ως μέτρο άμυνας με απώτερο σκοπό την ελαχιστοποίηση της δικτυακής συμφόρησης, καθώς και ως μέσο ελέγχου και περαιτέρω ρύθμισης της δρομολογούμενης ροής δεδομένων. Παραδείγματος χάριν, το *bandwidth throttling* έχει ήδη αποτελέσει κοινή πρακτική των παρόχων υπηρεσιών διαδικτύου κυρίως σε συστήματα τιμολόγησης κινητών τηλεπικοινωνιών, ενώ ταυτόχρονα, μπορεί να χρησιμοποιηθεί και ως μέσο επιβολής της προβλεπόμενης ή επιθυμητής ποιότητας υπηρεσιών (QoS – Quality of Service).

Στην περίπτωση του Multipath TCP, η παρούσα τεχνική είναι δυνατό να χρησιμοποιηθεί με στόχο την χειραγώγηση των εξερχόμενων πακέτων πληροφορίας με απώτερο σκοπό την χειροκίνητη εκφόρτωσή τους μεταξύ των διαθέσιμων υπο-ροών TCP. Σε περισσότερο πρακτικά σενάρια, η κατάλληλη επιβολή *bandwidth throttling* θα μπορούσε να ενισχύσει την προσπάθεια αποφόρτισης των δικτύων κινητών επικοινωνιών 4^{ης} και 3^{ης} γενιάς με παράλληλη χρήση μη-εξουσιοδοτημένου κομματιού του διαθέσιμου φάσματος ραδιοσυχνότητας (Wi-Fi). Ταυτοχρόνως, καθίσταται δυνατή η εν μέρει θωράκιση των τελικών χρηστών ενάντια στην κατάχρηση του εγγεγραμμένου προγράμματος δεδομένων, ενώ οι ίδιοι είναι σε θέση να επωφελούνται από τα πλεονεκτήματα του επεκταμένου πρωτοκόλλου.

Για τους σκοπούς διεξαγωγής της σειράς πειραμάτων μας, αποφασίσαμε να «στραγγάλισουμε» την ζεύξη Ethernet με προορισμό την αντίστοιχη διεπαφή του Raspberry Pi, ενώ η κατάσταση του ασύρματου μονοπατιού διάδοσης παραμένει αμετάβλητη. Το Σχήμα 29 παρουσιάζει την προσπάθεια επιβολής *bandwidth throttling* από τον MPTCP Proxy κατά την λήψη ενός αρχείου μεγέθους μόλις 50 KB από το RPi. Παρατηρώντας προσεκτικά το γράφημα, διαπιστώνεται η αδυναμία έγκαιρης επιβολής της επιλεγμένης πολιτικής. Η λήψη του αρχείου ολοκληρώνεται ταχέως σε μόλις περίπου 0.05 δευτερόλεπτα, χρόνος που χαρακτηρίζεται ανεπαρκής για την καίρια απόκριση του μηχανισμού επιβολής των επιλεγμένων πολιτικών δρομολόγησης. Η εναλλαγή των ουρών της εξερχόμενης διαδικτυακής κίνησης με χρήση του αλγορίθμου HTB καθυστερεί από τον πυρήνα του λειτουργικού συστήματος Linux συγκριτικά με τον απαιτούμενο χρόνο μεταφοράς του αντίστοιχου αρχείου. Επιπρόσθετα, όπως φαίνεται και στην παρακάτω απεικόνιση, οι εμπλεκόμενες ζεύξεις χρησιμοποιούν εξαιρετικά μικρό ποσοστό της υπάρχουσας χωρητικότητάς τους με αποτέλεσμα η μεταφορά αρχείων αντίστοιχου μεγέθους

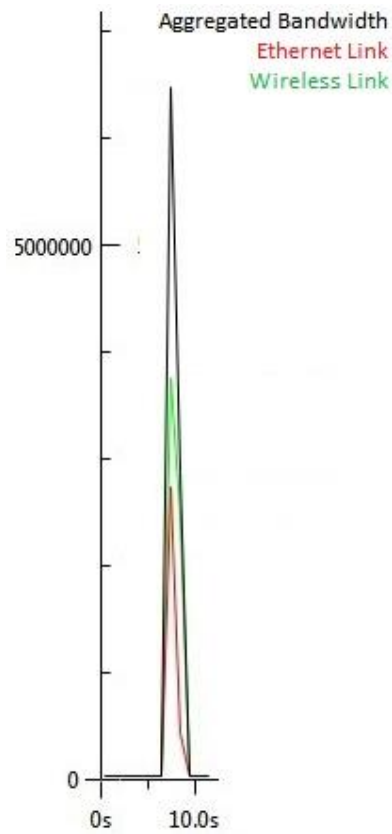
να παραμένει ανεπηρέαστη από τυχόν επιβολή *bandwidth throttling* του παρόντος επιπέδου (1 Mbps). Βέβαια, η μεταφορά αρχείων περιορισμένου μεγέθους δεν τείνει να προκαλέσει ραγδαίες, αξιοσημείωτες μεταβολές στην κατάσταση συμφόρησης του εκάστοτε δικτύου έχοντας ως άμεσο συνεπακόλουθο την υποβάθμιση της ποιότητας υπηρεσιών.



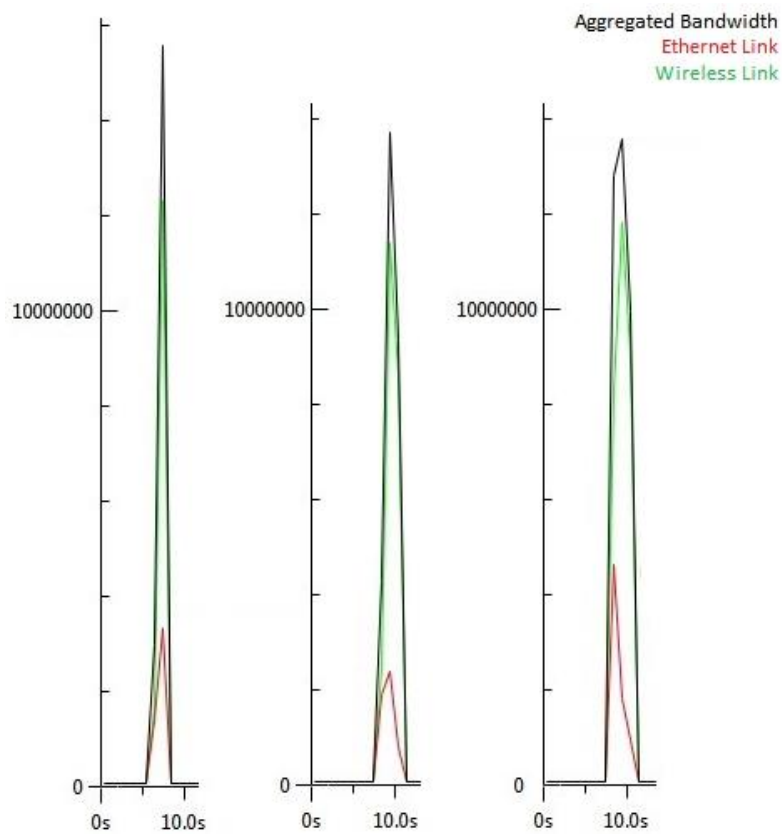
Σχήμα 29: Bandwidth throttling στο 1 Mb/s για αρχείο 50 KB

Παρ' όλα αυτά, η καθυστερημένη εναλλαγή των *queuing disciplines*, η οποία εν τέλει ολοκληρώνεται επιτυχώς, δεν αναιρείται. Οι νέες ουρές διαχείρισης της εξερχόμενης διαδικτυακής κίνησης παραμένουν εφαρμοσμένες και χρησιμοποιούνται για την δρομολόγηση των πακέτων πληροφορίας μελλοντικών συνδέσεων με την δυνατότητα υποστήριξης πολλαπλών διαδρομών δρομολόγησης. Έτσι, ο MPTCP Proxy δεν χρειάζεται να επαναλαμβάνει την ίδια διαδικασία οδηγούμενος σε σπατάλη διαθέσιμης μνήμης και επεξεργαστικής ισχύς για την ίδια συνεδρία MPTCP. Από την άλλη μεριά, εάν το εκάστοτε τερματικό επιχειρήσει τον επανασυγχρονισμό της σύνδεσης με ενεργοποιημένη την επιλογή εφεδρικών συνδέσεων, τότε ο μηχανισμός επιβολής *bandwidth throttling* θα προγραμματίσει τα qdiscs σύμφωνα με τον κλασσικό, προεπιλεγμένο αλγόριθμο FIFO.

Στην συνέχεια, επαναφέροντας τον αλγόριθμο προγραμματισμού αποστολής των πακέτων πληροφορίας σε FIFO, επανεξετάζουμε την απόκριση του μηχανισμού επιβολής *bandwidth throttling* σε συνδυασμό με την επίδοση του Multipath TCP για αρχεία τεσσάρων (4) διαφορετικών μεγεθών. Πρώτα απ' όλα, το Σχήμα 30 απεικονίζει την επίδοση των υποροών TCP για την μεταφορά ενός αρχείου 1 MB. Παρατηρώντας το γράφημα, διαπιστώνουμε πως η συμπεριφορά της σύνδεσης MPTCP δεν αποκλίνει ιδιαίτερα από την παραπάνω περίπτωση. Όμως, ενώ η επίδοση των εμπλεκόμενων ζεύξεων παραμένει συγκρίσιμη, η ασύρματη ζεύξη υπερτερεί πλέον σε προσφερόμενη χωρητικότητα της διεπαφής Ethernet. Η συγκεκριμένη συμπεριφορά οφείλεται στον έγκαιρα ολοκληρωμένο επαναπρογραμματισμό των ουρών διαχείρισης των εξερχόμενων πακέτων.

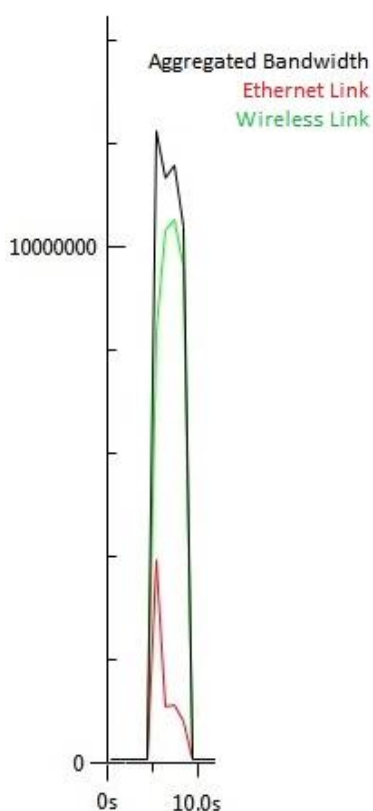


Σχήμα 30: Bandwidth throttling στο 1 Mb/s για αρχείο 1 MB



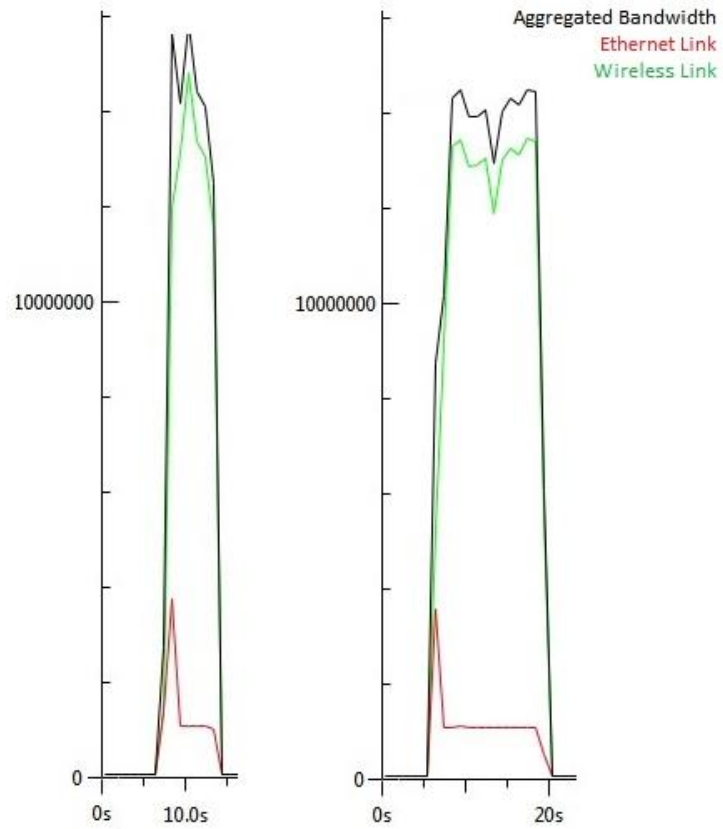
Σχήμα 31: Bandwidth throttling στο 1 Mb/s για αρχεία 2, 3, 4 MB

Η μεταφορά των τριών (3) εναπομεινάντων αρχείων παρουσιάζεται στο *Σχήμα 31*. Η συμπεριφορά της εκάστοτε σύνδεσης MPTCP παραμένει στο ίδιο μοτίβο αναφορικά με το *Σχήμα 30*. Σε όλες τις περιπτώσεις, λαμβάνει χώρα η επιτυχής επιβολή των επιθυμητών πολιτικών δρομολόγησης χωρίς, όμως, να γίνονται αντιληπτές οι επακόλουθες συνέπειες. Το περιορισμένο μέγεθος όλων των προαναφερθέντων αρχείων επιτρέπει την ολοκλήρωση της μεταφοράς τους με μοναδικό αντίκτυπο την μερικώς περιορισμένη προσφερόμενη ταχύτητα της ενσύρματης διαδρομής δρομολόγησης. Σε αυτό το σημείο, αξίζει να υπογραμμίσουμε πως σε περίπτωση εκ νέου αξιολόγησης της επίδοσης των υπο-ροών TCP χρησιμοποιώντας τον υπάρχοντα αλγόριθμο προγραμματισμού αποστολής πακέτων (HTB), θα γινόταν αντιληπτός ο περιορισμός του διαθέσιμου εύρους ζώνης της διεπαφής Ethernet. Παρ' όλα αυτά, η απόδοση της συνεδρίας MPTCP θα βίωνε μόνο ελαφριά υποβάθμιση λόγω του περιορισμένου όγκου των μεταφερόμενων αρχείων.

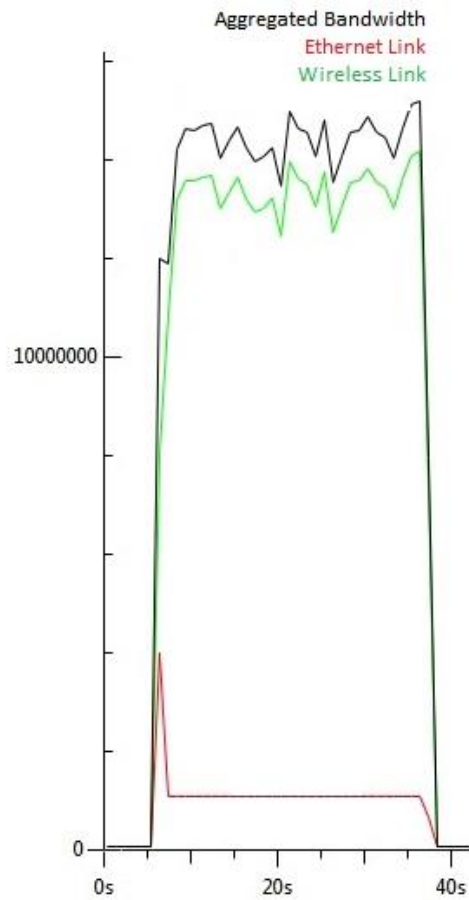


Σχήμα 32: Bandwidth throttling στο 1 Mb/s για αρχείο 5 MB

Το *Σχήμα 32* παρουσιάζει ελαφρώς για πρώτη φορά τον τελικό περιορισμό του εύρους ζώνης που επιτυγχάνεται ως απόρροια της δυναμικής εναλλαγής των *qdiscs* στον πυρήνα του λειτουργικού συστήματος Linux. Εξετάζοντας προσεκτικά το γράφημα, παρατηρούμε πως ο μηχανισμός επιβολής *bandwidth throttling* απαιτεί περίπου 3 δευτερόλεπτα για την επιτυχή εκτέλεσή του μόνο, όμως, κατά την προσπάθεια εγκατάστασης της πρώτης σύνδεσης MPTCP από την πλευρά του χρήστη, όπως έχει προαναφερθεί. Το *Σχήμα 33*, στη συνέχεια, αναπαριστά ακόμα καλύτερα την συμπεριφορά των υπο-ροών TCP κατά την επιβολή του ορίου παροχής εύρους ζώνης για μεγαλύτερα μεγέθη αρχείων, 10 και 20 MB, αντίστοιχα. Τέλος, το *Σχήμα 34* καθιστά ευκολότερα κατανοητό τον περιορισμό της διαθέσιμης χωρητικότητας, ενώ παράλληλα, απεικονίζει αναλυτικότερα την ανταπόκριση και συμπεριφορά τόσο της ενσύρματης υπο-ροής, όσο και της ασύρματης ζεύξης, η οποία τείνει να μεταδώσει το μεγαλύτερο ποσοστό του μεταφερόμενου φορτίου.

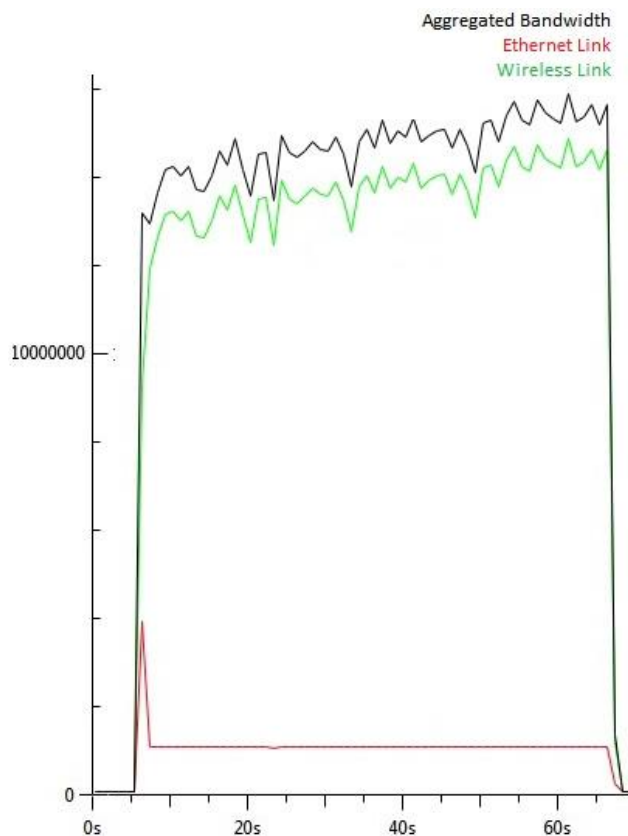


Σχήμα 33: Bandwidth throttling στο 1 Mb/s για αρχεία 10, 20 MB



Σχήμα 34: Bandwidth throttling στο 1 Mb/s για αρχείο 50 MB

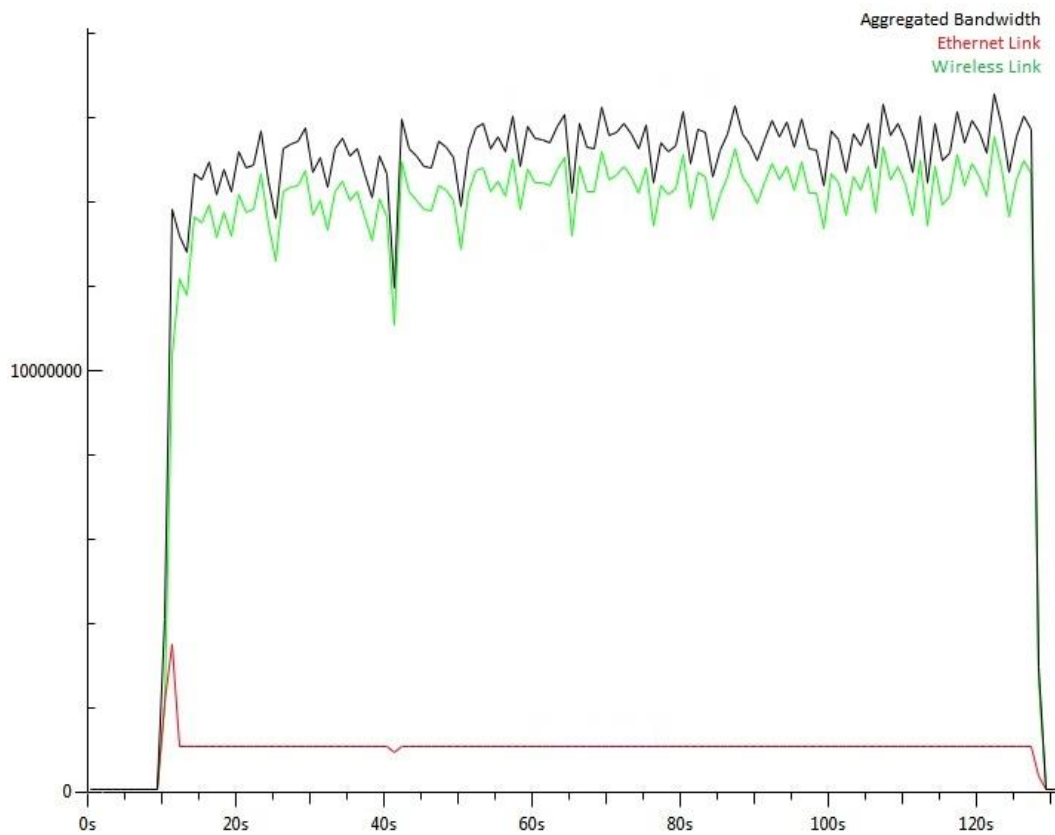
Σε αυτό το σημείο, παρουσιάζεται η επίδοση του Multipath TCP σε συνδυασμό με την επιβολή *bandwidth throttling* για αρχεία μεγέθους 100 και 200 MB, αντίστοιχα. Το Σχήμα 35 απεικονίζει την επίδοση που επιτυγχάνεται κατά την μεταφορά του πρώτου εκ των δύο αρχείων. Ενώ χρησιμοποιούνται αποδοτικά και οι δύο εμπλεκόμενες υπο-ροές TCP, η διαθέσιμη χωρητικότητα της ενσύρματης ζεύξης ανέρχεται σε μόλις 1 Mbps. Ως αποτέλεσμα, το βασικότερο ποσοστό των μεταδιδόμενων δεδομένων τείνει να μεταφερθεί μέσω του ασύρματου περιβάλλοντος διάδοσης. Η συγκεκριμένη επιλογή δρομολόγησης επηρεάζει, φυσικά, και την λειτουργία του μηχανισμού ελέγχου συμφόρησης, ο οποίος τείνει να αυξήσει σταδιακά το παράθυρο συμφόρησης της ασύρματης ζεύξης ανταποκρινόμενος στην μείωση της επίδοσης της ζεύξης Ethernet, μία κατάσταση που συμβαδίζει πλήρως με το θεωρητικό υπόβαθρο που έχει τεθεί. Παρόμοια συμπεριφορά παρατηρείται και στο Σχήμα 36. Σε αυτήν την περίπτωση, όμως, η ασύρματη ζεύξη πάσχει από εντονότερες αυξομειώσεις της επίδοσής της αποτρέποντας κατ' επέκταση την συνεχόμενη αύξηση του παραθύρου συμφόρησης της αντίστοιχης υπο-ροής. Αντιθέτως, η απόδοση του ασύρματου μονοπατιού διάδοσης παραμένει ουσιαστικά στα ίδια επίπεδα μετά το 40^ο δευτερόλεπτο.



Σχήμα 35: Bandwidth throttling στο 1 Mb/s για αρχείο 100 MB

Σύμφωνα με όλα τα προαναφερθέντα και την γενικότερη ιδέα που εξυπηρετεί η επιβολή του *bandwidth throttling* σε συνδυασμό με το επεκταμένο πρωτόκολλο, τίθεται πάντα ένα δίλημμα (trade-off) μεταξύ των διαθέσιμων επιλογών παραμετροποίησης και των επακόλουθων συνεπειών, θετικών ή αρνητικών. Στην παρούσα περίπτωση, όπως έχουμε ήδη υπογραμμίσει, ο περιορισμός του προσφερόμενου εύρους ζώνης ανά υπο-ροή TCP είναι δυνατό να χρησιμοποιηθεί τόσο για την ανακούφιση συμφορημένων δικτύων, όσο και την προστασία των τελικών χρηστών σε περιβάλλον κινητών τηλεπικοινωνιών. Από την άλλη μεριά, όμως, τα παραπάνω έχουν ως συνεπακόλουθο την αναπόφευκτη θυσία, διαθέσιμων υπό άλλες συνθήκες, δικτυακών πόρων και την κατ' επέκταση καθυστερημένη

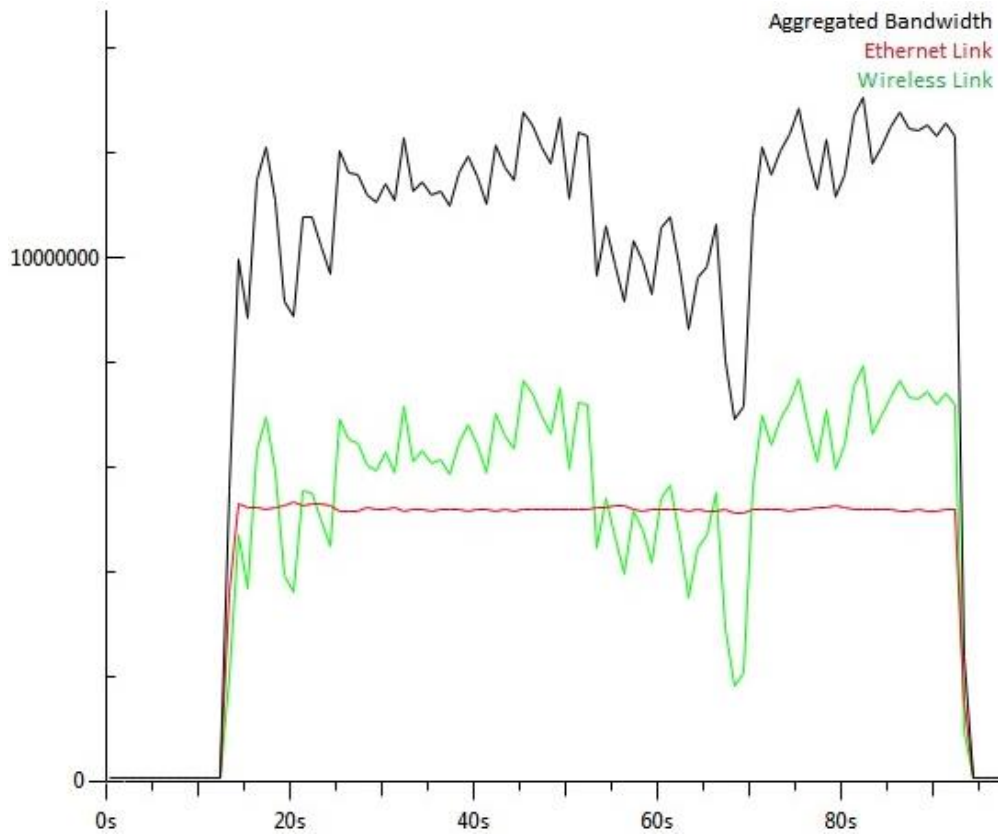
μετάδοση των επιλεγμένων δεδομένων. Παρ' όλα αυτά, η τελική ανακατανομή του εν τέλει διαθέσιμου εύρους ζώνης τείνει να υπερéχει συγκριτικά με την περίπτωση του κλασσικού πρωτοκόλλου, επαληθεύοντας με αυτόν τον τρόπο παραπάνω από έναν εκ των σχεδιαστικών στόχων του MPTCP.



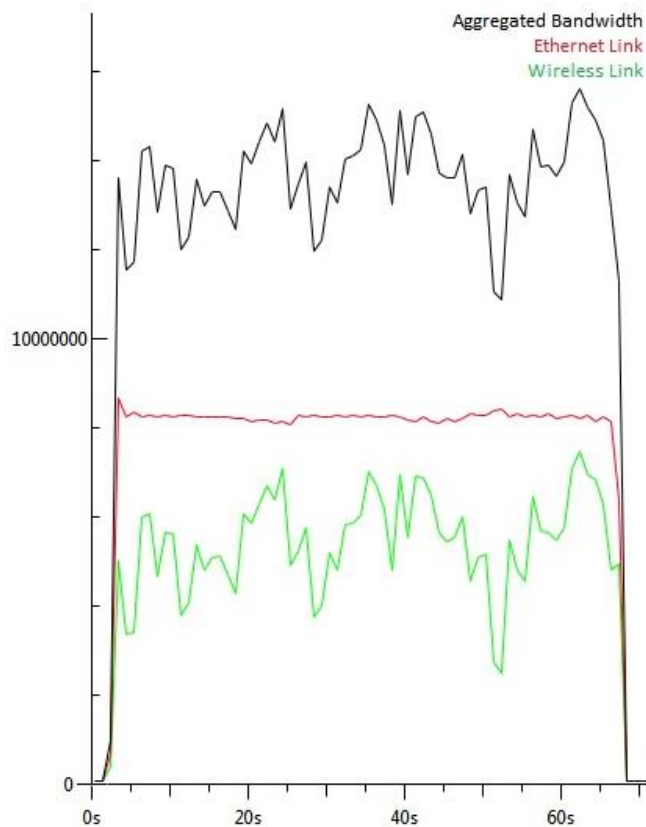
Σχήμα 36: Bandwidth throttling στο 1 Mb/s για αρχείο 200 MB

Τέλος, ολοκληρώνοντας τα πειραματικά σενάρια αξιολόγησης της επίδοσης του MPTCP σε συνδυασμό με το μηχανισμό διαχείρισης του διαθέσιμου εύρους ζώνης, επανεξετάζουμε την μεταφορά του αρχείου μεγέθους 100 MB για διαφοροποιημένα επίπεδα περιορισμού της διαθέσιμης χωρητικότητας και συγκρίνουμε την συμπεριφορά των υπο-ροών TCP με το Σχήμα 35. Όπως ήταν αναμενόμενο, η αύξηση της επίδοσης της ασύρματης ζεύξης έχει ως άμεση συνέπεια την μείωση του μεταφερόμενου όγκου δεδομένων κατά μήκος της ασύρματης διαδρομής δρομολόγησης. Το Σχήμα 37, καθώς και το Σχήμα 38, παρουσιάζουν γραφικά την παραπάνω συμπεριφορά για επιβολή *bandwidth throttling* στα 5 και 8 Mbps, αντίστοιχα. Παράλληλα, παρατηρείται αποσταθεροποίηση της επίδοσης της ασύρματης ζεύξης, η οποία είναι δυνατό να οφείλεται τόσο σε τυχόν παρεμβολές στο περιβάλλον μετάδοσης, όσο και στην προσπάθεια του MPTCP να δρομολογήσει μεγαλύτερο ποσοστό εξερχόμενων πακέτων μέσω της ζεύξης Ethernet.

Εν κατακλείδι, καθίσταται κατανοητή η εφαρμογή ενός μηχανισμού περιορισμού του διαθέσιμου εύρους ζώνης ανά διεπαφή και η τελική υλοποίησή του, η οποία όντας συσχετισμένη με το πρωτόκολλο MPTCP, μπορεί να συνδράμει στην περαιτέρω διαχείριση των ταυτόχρονα χρησιμοποιούμενων υπο-ροών. Επιπροσθέτως, η δυνατότητα επιβολής διαφόρων, επιθυμητών επιπέδων χαλιναγωγής της προσφερόμενης χωρητικότητας αντικατοπτρίζει την ανάγκη εφαρμογής δυναμικών και ευέλικτων σχημάτων διαχείρισης της διαδικτυακής κίνησης με απώτερο σκοπό την καθιέρωση πολιτικών δρομολόγησης που αποσκοπούν στην γενικότερη βελτίωση της ποιότητας υπηρεσιών.



Σχήμα 37: Bandwidth throttling στα 5 Mb/s για αρχείο 100 MB



Σχήμα 38: Bandwidth throttling στα 8 Mb/s για αρχείο 100 MB

6. Επίλογος

6.1. Σχετική Έρευνα

Κατά την διάρκεια των τελευταίων ετών, έχουν πραγματοποιηθεί ποικίλες μορφές ερευνών με στόχο την αποτελεσματικότερη κατανομή των δικτυακών πόρων και την αποδοτικότερη χρησιμοποίηση του διαθέσιμου εύρους ζώνης στα διάφορων ειδών δίκτυα υπολογιστών. Η τρέχουσα βιβλιογραφία και έρευνα καταδεικνύουν την ύπαρξη πλεοναζόντων διαδρομών μεταξύ των εκάστοτε ζευγαριών διαδικτυακών κόμβων, οι οποίες προς το παρόν τείνουν να παραμένουν επί το πλείστον αναξιοπίστες. Η υλοποίηση αξιόπιστων λύσεων με στόχο την επιτυχή συσσώρευση και ανακατανομή των διαθέσιμων πόρων μπορεί να συνδράμει στην βελτίωση της απόδοσης των συνδέσεων από άκρο σε άκρο προσφέροντας υψηλότερες επιδόσεις και κατ' επέκταση ενισχυμένη ποιότητα υπηρεσιών.

Μία προγενέστερη προσέγγιση του προβλήματος αποτελεσματικής διαχείρισης των διαθέσιμων διαδικτυακών πόρων αποτέλεσε η ανάπτυξη του πρωτοκόλλου SCTP (Stream Control Transmission Protocol) [33] ως μία προσπάθεια αξιοποίησης πολλαπλών ροών δεδομένων μεταξύ τερματικών εφοδιασμένων με παραπάνω από μία δικτυακές ή ραδιοηλεκτρικές διεπαφές. Παρ' όλα αυτά, η έλλειψη συμβατότητας του πρωτοκόλλου με το σημερινό Internet καθιέρωσε ως σχεδόν απαγορευτική την πλήρη εφαρμογή του. Αντιθέτως, η σχεδίαση του Multipath TCP ως επέκταση του παραδοσιακού πρωτοκόλλου, η διάφανη λειτουργία του, καθώς και η δυνατότητα υποχώρησης σε περίπτωση δυσλειτουργίας και χρήσης του κλασσικού TCP [34], καθιστούν το πρωτόκολλο αξιόπιστο και συμβατό με τα υπάρχοντα συστήματα παροχής υπηρεσιών και ανταλλαγής δεδομένων.

Ποικίλες μελέτες έχουν διεξαχθεί για την ανάπτυξη και σταδιακή βελτίωση του επεκταμένου πρωτοκόλλου [5, 7, 35] οδηγώντας εν τέλει στην βέλτιστη αξιοποίηση των διαθέσιμων δικτυακών πόρων εν παρουσία κατάλληλων μηχανισμών υπεύθυνων για την εκχώρηση πακέτων και την καταστολή της «επιθετικότητας» των συνδέσεων MPTCP [3, 13]. Παράλληλα, έχουν προταθεί παραλλαγές της υπάρχουσας αρχιτεκτονικής, όπως η SCA (Splitter-Combiner Architecture) [36] με σκοπό την μετατροπή πρωτοκόλλου-συνδέσεων TCP/MPTCP και την κατ' επέκταση βελτίωση της επίδοσης ταυτόχρονα χρησιμοποιούμενων διαδρομών δρομολόγησης διαφορετικών καθυστερήσεων. Επιπροσθέτως, έχουν πραγματοποιηθεί περαιτέρω πειράματα αξιολόγησης της απόδοσης του πρωτοκόλλου και επιβεβαίωσης των θεωρητικά αναμενόμενων αποτελεσμάτων υπό ρεαλιστικά σενάρια ασύρματης μετάδοσης [37, 38, 39]. Ακόμα, έχουν καταγραφεί μελέτες αναφορικά με την ανταπόκριση του πρωτοκόλλου σε συμβάντα κινητικότητας και δυνατότητας επίτευξης αδιάκοπης διαπομπής αποφεύγοντας πιθανή υποβάθμιση της ποιότητας υπηρεσιών [4, 14, 39].

Η εφαρμογή του πρωτοκόλλου, όμως, βρίσκεται ακόμα σε πρώιμο στάδιο και η περαιτέρω εξάπλωσή του καθυστερεί λόγω της διστακτικότητας που υπάρχει ως προς τον χρόνο και κόστος αναβάθμισης μεγάλων κέντρων δεδομένων και πλατφόρμων παροχής υπηρεσιών. Ως αποτέλεσμα, έχει προταθεί η ανάπτυξη MPTCP Proxies, οι οποίοι θα λειτουργούν ως μετατροπείς πρωτοκόλλου εκ μέρους των παραδοσιακών TCP συστημάτων και θα επιφέρουν επιπρόσθετες δυνατότητες επιλογών και επεκτάσεις στο MPTCP με σκοπό την υποστήριξη της λειτουργικότητάς τους [40]. Επιπλέον, έχει παρουσιαστεί-αποδειχθεί η λειτουργία ενός απλού μετατροπέα πρωτοκόλλου, ο οποίος συνδυάζει την υποστήριξη

πολλαπλών διαδρομών δρομολόγησης με την κατάλληλη επιβολή δυναμικών σχημάτων ελέγχου της διαδικτυακής κίνησης, συμβατών με το επεκταμένο πρωτόκολλο [41]. Στο ίδιο μήκος κύματος, οι Gregory Detal, Christoph Paasch και Oliver Bonaventure ανέπτυξαν μία εφαρμογή μετατροπής πρωτοκόλλου υπό την ονομασία *MiMBox* [42], την οποία υλοποίησαν ως ένα module υψηλών επιδόσεων στον πυρήνα του λειτουργικού συστήματος Linux. Ο συγκεκριμένος MPTCP Proxy έχει την δυνατότητα να επιτύχει υψηλές ταχύτητες (διάφανης) επεξεργασίας και προώθησης κοντά στις αντίστοιχες ενός απλού IP router χωρίς περιορισμούς στον τύπο πρωτοκόλλου του στρώματος εφαρμογής.

6.2. Σύνοψη

Τα δίκτυα υπολογιστών επεκτείνονται συνεχώς υποστηρίζοντας διακομιστές με την δυνατότητα ταυτόχρονης πρόσβασης σε παραπάνω από ένα υποδίκτυα, κέντρα δεδομένων με παρούσες πλεονάζουσες διαδρομές, καθώς και έξυπνες συσκευές με πολλαπλές ραδιοηλεκτρικές διεπαφές. Η διαρκής ανάπτυξη των τεχνολογιών πρόσβασης στο Διαδίκτυο, καθώς και η αυξανόμενη ζήτηση εύρους ζώνης, έχουν ως συνεπακόλουθο την ανάγκη εξεύρεσης λύσεων για την αποτελεσματικότερη κατανομή των υπαρχόντων δικτυακών πόρων και την αποδοτική αξιοποίηση των πολλαπλών διαθέσιμων διεπαφών ανά τερματικό. Το Multipath TCP, ως επέκταση του κλασσικού πρωτοκόλλου, επιτρέπει στους δικτυακούς κόμβους να αποδεσμευτούν από την σημερινή αρχή του Διαδικτύου, η οποία υπαγορεύει την ύπαρξη συνδέσεων από άκρο σ' άκρο, και να απαγκιστρωθούν από την προσφορά περιορισμένης χωρητικότητας σύμφωνα με την κατάσταση συμφόρησης της παραδοσιακής σύνδεσης TCP.

Καθώς το βασικό μειονέκτημα του Multipath TCP αποτελεί η έλλειψη υποστήριξης από τα υπάρχοντα συστήματα, η παρεμβολή ενός MPTCP Proxy κρίνεται άκρως επωφελής για την βελτιστοποίηση της κίνησης εντός των δικτύων πρόσβασης. Με αυτόν τον τρόπο, έξυπνα τερματικά με την δυνατότητα ταυτόχρονης αξιοποίησης πολλαπλών διαδρομών δρομολόγησης θα έχουν την ευκαιρία συσσώρευσης διαθέσιμου εύρους ζώνης και επιτυχούς ανακατανομής των δικτυακών πόρων κατά την διάρκεια της σταδιακής υιοθέτησης του επεκταμένου πρωτοκόλλου από τις πλατφόρμες παροχής υπηρεσιών και τους κόμβους ανταλλαγής δεδομένων.

Καθ' όλη την διάρκεια των πειραμάτων αξιολόγησης του πρωτοκόλλου, επιβεβαιώθηκε η ικανότητα του MPTCP σχετικά με την επίτευξη συσσωμάτωσης εύρους ζώνης και την αποδοτική εκφόρτωση δεδομένων δια μέσου των διαθέσιμων διεπαφών. Επιπροσθέτως, παρουσιάστηκε η συμπεριφορά και άμεση ανταπόκριση του επεκταμένου πρωτοκόλλου κατά την διάρκεια συμβάντων διαπομπής και η κατ' επέκταση εισαγωγή ενισχυμένης ανθεκτικότητας σε περίπτωση σεναρίων κινητικότητας. Στην συνέχεια, αξιολογήθηκαν οι απαιτήσεις χωρητικότητας των TCP buffers λήψης για την επιτυχή κάλυψη των αναγκών διαχείρισης του αυξημένου πλήθους ληφθέντων πακέτων. Το μέγεθος τόσο των αποθηκευτικών χώρων λήψης, όσο και της μέγιστης μεταδιδόμενης μονάδας σε συνδυασμό με το φαινόμενο *head-of-line blocking* αποτελούν εκ των καταλυτικότερων παραμέτρων επηρεασμού της επίδοσης μίας σύνδεσης MPTCP. Ως εκ τούτου, η εκ νέου αξιολόγηση του κλασσικού αλγορίθμου αναδιάταξης πακέτων και η μελλοντική ανάπτυξη ενός αναθεωρημένου μηχανισμού στα πρότυπα του MPTCP είναι δυνατό να συνδράμει στην γενικότερη βελτίωση της επίδοσης του επεκταμένου πρωτοκόλλου. Ακόμα, η χωρητικότητα των buffers τείνει να επηρεάσει καθοριστικά και το κόστος εγκατάστασης του

πρωτοκόλλου εξαιτίας της επικείμενης, αναγκαίας αντικατάστασης του υπάρχοντος εξοπλισμού, ιδιαίτερα σε μεγάλα κέντρα δεδομένων και ιδιαιτέρως δραστήριους εξυπηρετητές. Τέλος, παρουσιάστηκε μία τεχνική περιορισμού του διαθέσιμου εύρους ζώνης ανά ζεύξη ως μέσο επιβολής πολιτικών δρομολόγησης συμβατών με το επεκταμένο πρωτόκολλο με στόχο τόσο την περαιτέρω χειραγώγηση της διαδικτυακής κίνησης, όσο και την ικανοποίηση οικονομικών κριτηρίων, δίχως την υποβάθμιση της ποιότητας των παρεχόμενων υπηρεσιών.

Επιλογικά, αξίζει να υπογραμμίσουμε την (επικείμενη) ανάγκη υλοποίησης περισσότερο δυναμικών σχημάτων ελέγχου της διαδικτυακής κίνησης συγκριτικά με την σημερινή, στατική πολιτική δρομολόγησης τόσο στον χώρο των σταθερών, όσο και των κινητών τηλεπικοινωνιών. Επιπλέον, το Multipath TCP κατέχει την δυνατότητα να συνδράμει στην διαδικασία μετάβασης στον κλάδο των επικοινωνιών, κατά την οποία τείνουν να αρθούν οριστικά οι διακρίσεις μεταξύ των σταθερών και κινητών δικτύων, παρέχοντας στους τελικούς χρήστες ευελιξία, καθώς και ανώτερη ποιότητα και εμπειρία υπηρεσιών μέσω του συνδυασμού σταθερών ευρυζωνικών και τοπικών ασύρματων τεχνολογιών πρόσβασης (FMC) [43].

Βιβλιογραφία

- [1] Gartner Inc., “Gartner Says 4.9 Billion Connected ‘Things’ Will Be in Use in 2015”, Press Release, 2014, <http://www.gartner.com/newsroom/id/2905717>
- [2] MultiPath TCP - Linux Kernel Implementation, <http://multipath-tcp.org/>
- [3] C. Raiciu, M. Handley, D. Wischik, “Coupled Congestion Control for Multipath Transport Protocols”, RFC 6356, IETF, 2011
- [4] C. Raiciu, D. Niculescu, M. Bagnulo, M. Handley, “Opportunistic Mobility with Multipath TCP”, ACM MobiArch, page 2, 2011
- [5] A. Ford, C. Raiciu, M. Handley, S. Barre, J. Iyengar, “Architectural Guidelines for Multipath TCP Development”, RFC 6182, IETF, 2011
- [6] A. Ford, C. Raiciu, M. Handley, “TCP Extensions for Multipath Operation with Multiple Addresses”, RFC 6824, IETF, 2012
- [7] C. Raiciu, C. Paasch, S. Barre, A. Ford, M. Honda, F. Duchene, O. Bonaventure, M. Handley, “How Hard Can It Be? Designing and Implementing a Deployable Multipath TCP”, Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation, page 3, 2012
- [8] Type-Length-Value, Wikipedia, <https://en.wikipedia.org/wiki/Type-length-value>
- [9] C. Raiciu, “Multipath TCP Signalling”, 2010
- [10] W. Stevens, “TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms”, RFC 2001, Network Working Group, 1997
- [11] D. Wischik, M. Handley, M. B. Braun, “The Resource Pooling Principle”, ACM SIGCOMM CCR, 2008
- [12] D. Wischik, C. Raiciu, A. Greenhalgh, M. Handley, “Design, implementation and evaluation of congestion control for multipath TCP”, NSDI, page 4, 2011
- [13] R. Khalili, N. Gast, M. Popovic, U. Upadhyay, J. Le Boudec, “MPTCP is not Pareto-Optimal: Performance Issues and a Possible Solution”, EPFL, IC-LCA2, 2012
- [14] C. Passch, G. Detal, F. Duchene, C. Raiciu, O. Bonaventure, “Exploring Mobile/WiFi Handover with Multipath TCP”, ACM CellNet, page 2, 2012
- [15] Mobile IP, Wikipedia, https://en.wikipedia.org/wiki/Mobile_IP
- [16] I. Van Beijnum, “Multipath TCP lets Siri seamlessly switch between Wi-Fi and 3G/LTE”, Ars Technica, 2012, <http://arstechnica.com/apple/2013/09/multipath-tcp-lets-siri-seamlessly-switch-between-wi-fi-and-3glte/>

- [17] L. Deng, D. Liu, T. Sun, “MPTCP Proxy for Mobile Networks”, Internet-Draft, 2014
- [18] X. Wei, C. Xiong, “MPTCP proxy mechanisms”, Internet-Draft, pages 4-5, 2014
- [19] Bandwidth Throttling, Wikipedia, https://en.wikipedia.org/wiki/Bandwidth_throttling
- [20] B. Hubert, T. Graf, G. Maxwell, R. van Mook, M. van Oosterhout, P. B. Schroeder, J. Spaans, P. Larroy, “Linux Advanced Routing & Traffic Control HOW TO”, Chapters 9-12, <http://lartc.org/howto/index.html>
- [21] M. Devera, “HTB Linux queuing discipline manual – user guide”, 2002, <http://luxik.cdi.cz/~devik/qos/htb/manual/userg.htm>
- [22] R. Russell, H. Welte, “Linux netfilter Hacking HOW TO”, 2002, <http://www.netfilter.org/documentation/HOWTO/netfilter-hacking-HOWTO.html>
- [23] O. Andreasson, “Iptables Tutorial 1.2.2”, <http://homes.di.unimi.it/sisop/qemu/iptables-tutorial.pdf>
- [24] Scapy, <http://www.secdev.org/projects/scapy/>
- [25] Wireshark, <https://www.wireshark.org/>
- [26] Introducing JSON, <http://json.org/>
- [27] XML, Wikipedia, <https://en.wikipedia.org/wiki/XML>
- [28] JSON-RPC, <http://json-rpc.org/>
- [29] VMware ESX and VMware ESXi, Product Datasheet, <http://www.vmware.com/files/pdf/VMware-ESX-and-VMware-ESXi-DS-EN.pdf>
- [30] Squid Caching Proxy, <http://www.squid-cache.org/>
- [31] Apache, HTTP SERVER PROJECT, <http://httpd.apache.org/>
- [32] World Wide Web, Wikipedia, https://en.wikipedia.org/wiki/World_Wide_Web
- [33] J. R. Iyengar, K. C. Shah, P. D. Amer, “Concurrent Multipath Transfer Using SCTP Multihoming”, IEEE/ACM Transactions on Networking (TON), 2006
- [34] M. Scharf, A. Ford, Alcatel-Lucent Bell Labs, Cisco, “Multipath TCP (MPTCP) Application Interface Considerations”, RFC 6897, IETF, 2013
- [35] M. Zhang, J. Lai, A. Krishnamurthy, L. Peterson, R. Wang, “A Transport Layer Approach for Improving End-to-End Performance and Robustness Using Redundant Paths”, USENIX, 2004
- [36] T. Ayar, L. Budzisz, A. Wolisz, “TCP over Multipath Paths Revisited: Towards Transparent Proxy Solutions”, IEEE ICC, 2012

- [37] Y. Chen, Y. Lim, R. J. Gibbens, E. M. Nahum, R. Khalili, D. Towsley, “A Measurement-based Study of Multipath TCP Performance over Wireless Networks”, ACM IMC, 2013
- [38] S. Deng, R. Netravali, A. Sivaraman, H. Balakrishnan, “WiFi, LTE, or Both? Measuring Multi-Homed Wireless Internet Performance”, ACM IMC, 2014
- [39] Y. Chen, E. M. Nahum, R. J. Gibbens, D. Towsley, Y. Lim, “Characterizing 4G and 3G Networks: Supporting Mobility with Multi-Path TCP”, UMass Amherst Technical Report: UM-CS-2012-022, 2012
- [40] G. Hampel, T. Klein, “MPTCP Proxies and Anchors”, Internet-Draft, 2012
- [41] C. Pollalis, P. Charalampou, E. Sykas, “HTTP Data Offloading Using Multipath TCP Proxy”, Proceedings of the 15th IEEE International Conference on Computer and Information Technology, 2015
- [42] G. Detal, C. Paasch, O. Bonaventure, “Multipath in the Middle(Box)”, ACM HotMiddleBox, 2013
- [43] Fixed-mobile convergence, Wikipedia, https://en.wikipedia.org/wiki/Fixed-mobile_convergence

Παράρτημα

A. Πηγαίος Κώδικας του Scapy (Packet) Sniffer

```
1.  #!/usr/bin/env python
2.
3.  from configparser import ConfigParser
4.  from scapy.all import *
5.  from jsonrpclib.SimpleJSONRPCServer import SimpleJSONRPCServer
6.
7.  # configuration file parser
8.  parser = ConfigParser()
9.  parser.readfp(open('config.txt'))
10.
11. eth = str(parser.get("networks", 'eth_lan')).split('/')[0].strip('0')
12. wlan = str(parser.get("networks", 'wir_lan')).split('/')[0].strip('0')
13.
14. def unpack(pkt):
15.
16.     if scapy.all.IP in pkt:
17.         ip_src = pkt[scapy.all.IP].src
18.         ip_dst = pkt[scapy.all.IP].dst
19.
20.     if scapy.all.TCP in pkt:
21.         sport = pkt[scapy.all.TCP].sport
22.         dport = pkt[scapy.all.TCP].dport
23.         flags = pkt[scapy.all.TCP].flags
24.         options = pkt[scapy.all.TCP].options
25.
26.         # check if host is MPTCP enabled
27.         mptcp_flag = int(str(options[-
28. 1]).strip('(').strip(')').split(',')[0])
29.
30.         # check whether link is set in back-up mode
31.         backup_flag = str(options[-
32. 1]).strip('(').strip(')').split('\\')[1]
33.
34.         '''
35.         tcp flags
36.         U A P R S F
37.         32 16 8 4 2 1
38.         ex. SYN-ACK --> SA --> 2 + 16 = 18
```

```

39.     '''
40.
41.     '''
42.     back-up flags
43.
44.         x00 --
> primary subflow starts tcp connection
45.         x10 --> secondary subflow joins
46.         x11 --> secondary subflow joins as back-
up link
47.     '''
48.
49.     global ip_prim
50.
51.     # get SYN packet
52.     if flags == int(2) and mptcp_flag == int(30):
53.         for i in range(1, 255):
54.             # match packet source ip
55.             if ip_src == (str(eth) + str(i)) or ip_src == (st
r(wlan) + str(i)):
56.                 if backup_flag == 'x00':
57.                     ip_prim = ip_src
58.                     break
59.                 if backup_flag == 'x10' or backup_flag == 'x1
1':
60.                     def sec():
61.                         ip_sec = ip_src
62.                         return ip_sec
63.
64.                     def prim():
65.                         return ip_prim
66.
67.                     def flag():
68.                         return backup_flag
69.
70.                 # shutdown function
71.                 def shutdown():
72.                     server.kill = True
73.                     return 1
74.
75.                 # bind the server to socket
76.                 server = json_rpc_server(('localhost', 80
00))
77.                 # register functions
78.                 server.register_function(prim)
79.                 server.register_function(sec)
80.                 server.register_function(flag)
81.                 server.register_function(shutdown)
82.

```

```

83.             # start the server
84.             server.serve_forever()
85.
86.             break
87. # JSON-RPC Server
88. class json_rpc_server(SimpleJSONRPCServer):
89.
90.     # serve_forever function implementing remote kill signal
91.     kill = False
92.
93.     def serve_forever(self):
94.
95.         while not self.kill:
96.             self.handle_request()
97.
98.     while True:
99.         try:
100.            pkt = sniff(filter="tcp and port 8080", prn=unpack, s
101.            tore=0)
102.        except:
103.            UnboundLocalError, KeyError

```

B. Πηγαίος Κώδικας του Bandwidth Throttling Daemon

```
1.  #!/usr/bin/env python
2.
3.  from configparser import ConfigParser
4.  import subprocess
5.  import jsonrpclib
6.  import socket
7.
8.  # configuration file parser
9.  parser = ConfigParser()
10. parser.readfp(open('config.txt'))
11.
12. su_pass = b'parser.get("superuser password", "su_pass")'
13.
14. rate_4 = str(parser.get("throttled queue", 'eth_rate'))
15. ceil_4 = str(parser.get("throttled queue", 'eth_ceil'))
16.
17. rate_7 = str(parser.get("active queue", 'wir_rate'))
18. ceil_7 = str(parser.get("active queue", 'wir_ceil'))
19.
20. rate = str(parser.get("rest of internet traffic", 'rest_rate'
21. ))
22.
23. ceil = str(parser.get("rest of internet traffic", 'rest_ceil'
24. ))
25.
26. # check existence of iptables rules - add mangle rules
27. def iptables():
28.     iptables_cmd = subprocess.Popen(['sudo', '-
29. S', 'iptables', '-t', 'mangle', '-
30. L'], stdout=subprocess.PIPE, stdin=subprocess.PIPE)
31.     output, err = iptables_cmd.communicate(su_pass + b'\n')
32.     # split the output into lines
33.     iptables_output = str(output).strip('b').strip('').split
34. ('\\n')
35.
36.     # clear empty lines for easier parsing
37.     for line in iptables_output:
38.         if line == '':
39.             # remove empty lines
40.             iptables_output.remove(line)
41.
42.     eth_rule = False
43.     wlan_rule = False
44.
45.     # search for specific output chain rules
46.     for line in iptables_output:
47.         if line == 'MARK          tcp --
48. mptcp.cn.ece.ntua.gr ' + eth + '      tcp spt:http-
49. alt MARK set 0x4':
50.
51.             # rule for pi's ethernet interface is present
```

```

40.         eth_rule = True
41.         if line == 'MARK          tcp  --
mptcp.cn.ece.ntua.gr  ' + wlan + '          tcp spt:http-
alt MARK set 0x7':
42.             # rule for pi's wireless interface is present
43.             wlan_rule = True
44.             if eth_rule is True and wlan_rule is True:
45.                 print('IPTables OK')
46.                 # pass
47.             if eth_rule is False:
48.                 # add rule for pi's ethernet interface if missing
49.                 iptables_rules_add.iptables_rules_eth()
50.                 eth_rule = True
51.             if wlan_rule is False:
52.                 # add rule for pi's wireless interface if missing
53.                 iptables_rules_add.iptables_rules_wlan()
54.                 wlan_rule = True
55.
56. # class for iptables rules input
57. class iptables_rules():
58.
59.     def iptables_rules_eth(self):
60.         eth_rule_add = subprocess.Popen('sudo -S iptables -
t mangle -A OUTPUT -p tcp -s 147.102.40.68 --source-port 8080 -
d ' + eth + ' -j MARK --set-
mark 4', shell=True, stdout=subprocess.PIPE, stdin=subprocess.PIPE
).communicate(su_pass + b'\n')
61.         return
62.
63.     def iptables_rules_wlan(self):
64.         wlan_rule_add = subprocess.Popen('sudo -S iptables -
t mangle -A OUTPUT -p tcp -s 147.102.40.68 --source-port 8080 -
d ' + wlan + ' -j MARK --set-
mark 7', shell=True, stdout=subprocess.PIPE, stdin=subprocess.PIPE
).communicate(su_pass + b'\n')
65.         return
66.
67. iptables_rules_add = iptables_rules()
68.
69. # check current queuing discipline
70. def queue():
71.     queue_cmd = subprocess.Popen(['sudo', '-S', 'tc', '-
s', 'qdisc', 'ls', 'dev', 'eth1'], stdout=subprocess.PIPE, stdin=s
ubprocess.PIPE)
72.     output, err = queue_cmd.communicate(su_pass + b'\n')
73.     # returns the current queuing discipline as a string
74.     queue_output = str(output).strip('b').strip('').split('\
\n')[0].split(':')[0].split(' ')[1]
75.     return queue_output

```

```

76.
77. # server policy based on queuing disciplines
78. class q_disciplines():
79.
80.     def htb(self):
81.         # htb qdisc
82.         root_1 = subprocess.Popen('sudo -
S tc qdisc replace dev eth1 root handle 1: htb default 2', shell=T
rue, stdout=subprocess.PIPE, stdin=subprocess.PIPE).communicate(su
_pass + b'\n')
83.         # class to throttle eth bw
84.         classid_4 = subprocess.Popen('sudo -
S tc class add dev eth1 parent 1: classid 1:4 htb rate ' + rate_4
+ ' ceil ' + ceil_4, shell=True, stdout=subprocess.PIPE, stdin=sub
process.PIPE).communicate(su_pass + b'\n')
85.         # class for wlan bw
86.         classid_7 = subprocess.Popen('sudo -
S tc class add dev eth1 parent 1: classid 1:7 htb rate ' + rate_7
+ ' ceil ' + ceil_7, shell=True, stdout=subprocess.PIPE, stdin=sub
process.PIPE).communicate(su_pass + b'\n')
87.         handle_4 = subprocess.Popen('sudo -
S tc filter add dev eth1 parent 1: handle 4 fw flowid 1:4', shell=
True, stdout=subprocess.PIPE, stdin=subprocess.PIPE).communicate(s
u_pass + b'\n')
88.         handle_7 = subprocess.Popen('sudo -
S tc filter add dev eth1 parent 1: handle 7 fw flowid 1:7', shell=
True, stdout=subprocess.PIPE, stdin=subprocess.PIPE).communicate(s
u_pass + b'\n')
89.         # class for the remaining traffic
90.         classid_2 = subprocess.Popen('sudo -
S tc class add dev eth1 parent 1: classid 1:2 htb rate ' + rate +
' ceil ' + ceil, shell=True, stdout=subprocess.PIPE, stdin=subproc
ess.PIPE).communicate(su_pass + b'\n')
91.
92.     def pfifo_fast(self):
93.         # pfifo_fast qdisc
94.         pfifo = subprocess.Popen('sudo -
S tc qdisc replace dev eth1 root pfifo_fast', shell=True, stdout=s
ubprocess.PIPE, stdin=subprocess.PIPE).communicate(su_pass + b'\n'
)
95.
96. qdisc = q_disciplines()
97.
98. # dynamic configuration -- throttling
99. while True:
100.     try:
101.         # connect to json-rpc server
102.         server = jsonrpclib.Server("http://localhost:8000")
103.         eth = server.prim()

```

```

104.         wlan = server.sec()
105.         backup_flag = server.flag()
106.         # send remote kill signal to the server
107.         server.shutdown()
108.
109.         # check iptables rules for every incoming connection
110.         iptables()
111.
112.         # full mesh transmission
113.         if backup_flag == 'x10':
114.             if queue() == 'pfifo_fast':
115.                 qdisc.htb()
116.             elif queue() == 'htb':
117.                 pass
118.         # secondary link in back-up mode
119.         elif backup_flag == 'x11':
120.             if queue() == 'htb':
121.                 qdisc.pfifo_fast()
122.             elif queue() == 'pfifo_fast':
123.                 pass
124.         else:
125.             Pass
126.
127.         # exception for 'Connection Refused' error when server is
down
128.     except:
129.         socket.error

```

Γ. Configuration File

```
# configuration file for mptcp_daemon
```

```
1. [networks]
2. eth_lan = 147.102.40.0/24
3. wir_lan = 147.102.7.0/24
4.
5. [throttled queue]
6. eth_rate = 1mbit
7. eth_ceil = 1mbit
8.
9. [active queue]
10. wir_rate = 40mbit
11. wir_ceil = 40mbit
12.
13. [rest of internet traffic]
14. rest_rate = 50mbit
15. rest_ceil = 50mbit
16.
17. [superuser password]
18. su_pass = ██████████
```