

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΡΗΤΗΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΕΠΙΣΤΗΜΗΣ ΥΠΟΛΟΓΙΣΤΩΝ

**Παροχή διαφοροποιημένων υπηρεσιών με μηχανισμούς ελέγχου
συμφόρησης από άκρη – σε – άκρη**

Γεώργιος Ι. Μαργέτης

Μεταπτυχιακή Εργασία

Ηράκλειο, Νοέμβριος 2001

Περίληψη

Η αρχιτεκτονική διαφοροποιημένων υπηρεσιών (DiffServ) προσφέρει μία λύση για παροχή διαφοροποιημένων υπηρεσιών στο Internet. Σύμφωνα με αυτήν, η υλοποίηση ενός μικρού αριθμού από συμπεριφορές προώθησης πακέτων στους δρομολογητές του δικτύου είναι αρκετή για την παροχή διαφοροποιημένης υπηρεσίας. Τα μειονεκτήματα αυτής της προσέγγισης είναι η αυξημένη πολυπλοκότητα που παρουσιάζει, σε σύγκριση με τους μηχανισμούς προώθησης πακέτων που χρησιμοποιούνται σήμερα στο Internet, και η ανάγκη για συνεργασία μεταξύ των δρομολογητών, ώστε να προσφέρονται υπηρεσίες από άκρο – σε – άκρο.

Μια διαφορετική προσέγγιση υποθέτει ένα απλό δίκτυο, που υποστηρίζει μηχανισμό σημάτων ανάδρασης, ο οποίος ονομάζεται Explicit Congestion Notification (ECN), και βασίζεται στο μαρκάρισμα πακέτων σε καταστάσεις συμφόρησης. Ο μηχανισμός αυτός προειδοποιεί τα “έξυπνα” συστήματα στα άκρα του δικτύου για καταστάσεις συμφόρησης που παρουσιάζονται, ώστε να ρυθμίσουν το ρυθμό αποστολής τους ανάλογα με τις ανάγκες τους και τα σήματα ανάδρασης που λαμβάνουν.

Στα πλαίσια της παρούσας εργασίας προτείνουμε και μελετήσαμε αλγόριθμους, τόσο για τη φάση αποφυγής της συμφόρησης (congestion avoidance), όσο και για τη φάση γρήγορης σύγκλισης (fast convergence), που προσφέρουν διαφοροποίηση υπηρεσιών ακολουθώντας την παραπάνω προσέγγιση. Παράλληλα, διερευνήσαμε την αλληλεπίδρασή των μηχανισμών αυτών με αλγόριθμους μαρκαρίσματος που χρησιμοποιούνται στο δίκτυο. Η υλοποίηση των αλγορίθμων και η εκτέλεση των πειραμάτων έγινε τόσο σε περιβάλλον προσομοίωσης, με τη χρήση του προσομοιωτή δικτύου Network Simulator version 2 (NS 2), όσο και σε ένα πραγματικό σύστημα με σταθμούς εργασίας που τρέχουν την FreeBSD έκδοση του λειτουργικού συστήματος Unix.

Abstract

The Differentiated Services (DiffServ) architecture provides one solution for service differentiation in the Internet. It follows the logic that, a small number of forwarding behaviors implemented at the routers are sufficient for providing service differentiation. The drawbacks of the approach are the increased complexity, compared to the Internet today and the need for co-operation among the routers in order to provide end – to – end services.

A different approach considers a network that provides a simple feedback mechanism, namely Explicit Congestion Notification (ECN), which is based on packet marking every time there is congestion. This mechanism sends congestion information to intelligent end-systems, which adjust their transmission rate in response to feedback.

The bulk of the research in this Master's thesis deals with the study of algorithms, both for the congestion avoidance and fast convergence phases, which provide service differentiation according to the above approach. In parallel, we studied, how different packet marking algorithms, affected to the algorithms we studied. The implementation of the algorithms and the execution of our experiments was done, both in simulation, using Network Simulator 2 (NS2) and in a real test-bed, which is comprised of three workstations running the Unix based operation system, FreeBSD.

Ευχαριστίες

Στο σημείο αυτό, θα ήθελα να ευχαριστήσω θερμά τον Βασίλη Σύρη, για την πολύτιμη βοήθεια που μου παρείχε καθ' όλη τη διάρκεια της παρούσας εργασίας, αλλά και για την μεγάλη εμπειρία που απέκτησα στην περιοχή της έρευνας από τη συνεργασία μας. Επίσης θα ήθελα να εκφράσω τις ευχαριστίες μου στον επόπτη μου κ. Απόστολο Τραγανίτη, καθώς και στα υπόλοιπα μέλη της τριμελούς επιτροπής κ. κ. Δημήτρη Πλεξουσάκη και Ευάγγελο Μαρκάτο, για τις πολύτιμες διορθώσεις που έκαναν στην παρούσα εργασία.

Παράλληλα θέλω να εκφράσω ένα μεγάλο ευχαριστώ στους γονείς μου Γιάννη και Φωτεινή καθώς και στην αδερφή μου Έφη για την αμέριστη συμπαράσταση και βοήθεια που μου προσέφεραν όλο το διάστημα, των ακαδημαϊκών μου σπουδών. Τέλος, θα ήθελα να ευχαριστήσω ιδιαίτερος την Σταυρούλα, που χωρίς την φροντίδα και την αφοσίωσή της δε θα έφτανα σήμερα να γράφω αυτές τις γραμμές.

Στους γονείς μου Γιάννη και Φωτεινή.

Περιεχόμενα

ΠΕΡΙΛΗΨΗ.....	I
ABSTRACT	II
1. ΕΙΣΑΓΩΓΗ	1
1.1. DIFFSERV	2
1.2. ΕΝΑΛΛΑΚΤΙΚΗ ΠΡΟΣΕΓΓΙΣΗ.....	4
1.2.1. <i>Explicit Congestion Notification (ECN)</i>	5
1.2.2. <i>Χρέωση Συμφόρησης και ωφελιμότητα χρηστών</i>	6
1.3. ΣΚΟΠΟΣ ΤΗΣ ΕΡΓΑΣΙΑΣ ΚΑΙ ΟΡΓΑΝΩΣΗ ΚΕΙΜΕΝΟΥ	9
2. ΜΗΧΑΝΙΣΜΟΙ ΔΙΑΦΟΡΟΠΟΙΗΣΗΣ ΥΠΗΡΕΣΙΩΝ ΣΤΟ INTERNET	12
2.1. TCP CONGESTION AVOIDANCE.....	13
2.2. MULTCP.....	15
2.2.1. <i>MulTCP Congestion Avoidance</i>	15
2.3. WILLINGNESS TO PAY CONGESTION AVOIDANCE (WTP)	16
2.4. ΤΟ ΠΡΟΒΛΗΜΑ ΤΗΣ ΣΥΓΚΛΙΣΗΣ.....	18
2.4.1. <i>MulTCP Slow start</i>	20
2.4.2. <i>Weighted Slow start</i>	21
2.4.3. <i>Large K</i>	21
2.4.4. <i>TCP Slow start</i>	22
2.5. ΜΗΧΑΝΙΣΜΟΙ ΜΑΡΚΑΡΙΣΜΑΤΟΣ.....	23
2.5.1. <i>Virtual Queue</i>	23
2.5.2. <i>Random Early Detect</i>	24
2.5.3. <i>Load Based Marking</i>	24
3. ΥΛΟΠΟΙΗΣΗ.....	26
3.1. ΥΛΟΠΟΙΗΣΗ ΣΕ ΠΡΟΣΟΜΟΙΩΤΗ	26
3.1.1. <i>Ο προσομοιωτής δικτύου “Network Simulator 2”</i>	26
3.1.2. <i>Υλοποίηση του WTP και αλγορίθμων για γρήγορη σύγκλιση</i>	30
3.1.3. <i>Υλοποίηση των αλγορίθμων μαρκαρίσματος</i>	32
3.2. ΠΡΑΓΜΑΤΙΚΗ ΥΛΟΠΟΙΗΣΗ.....	34
3.2.1. <i>Περιγραφή συστήματος</i>	34
3.2.2. <i>Υλοποίηση WTP στον πυρήνα του λειτουργικού</i>	35
3.2.3. <i>Θέματα υλοποίησης του WTP σε πραγματικό περιβάλλον</i>	37
3.2.4. <i>Υλοποίηση αλγορίθμων μαρκαρίσματος</i>	38
3.2.5. <i>Θέματα υλοποίησης της πολιτικής εξυπηρέτησης ουρών VQ σε πραγματικό περιβάλλον</i>	40
4. ΠΕΙΡΑΜΑΤΙΚΑ ΑΠΟΤΕΛΕΣΜΑΤΑ	42
4.1. ΜΕΛΕΤΗ ΑΛΓΟΡΙΘΜΩΝ ΓΙΑ ΤΗ ΦΑΣΗ ΑΠΟΦΥΓΗΣ ΤΗΣ ΣΥΜΦΟΡΗΣΗΣ	43

4.1.1.	Διαφοροποίηση ως προς την επιθυμία χρέωσης.....	43
4.1.2.	Επίδραση του παράγοντα k στην ταχύτητα σύγκλισης	46
4.1.3.	Σύγκριση WTP και MultTCP	47
4.2.	ΜΕΛΕΤΗ ΑΛΓΟΡΙΘΜΩΝ ΓΙΑ ΤΗ ΦΑΣΗ ΓΡΗΓΟΡΗΣ ΣΥΓΚΛΙΣΗΣ	49
4.2.1.	Ταχύτητα σύγκλισης και διαφοροποίηση.....	49
4.2.2.	Διαφοροποίηση ως προς την επιθυμία χρέωσης.....	52
4.2.3.	Χρόνοι σύγκλισης.....	54
5.	ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΗ ΕΡΓΑΣΙΑ.....	56
6.	ΑΝΑΦΟΡΕΣ	61

Σχήματα

Σχήμα 1. Φάσεις γρήγορης σύγκλισης και αποφυγής της συμφόρησης για συνδέσεις στο Internet.....	12
Σχήμα 2 Απλοποιημένη άποψη του NS2.....	27
Σχήμα 3 Αρχιτεκτονική του NS2.....	29
Σχήμα 4 Ιεραρχία αντικειμένων C++ και OTcl.....	30
Σχήμα 5 Διάγραμμα ροής της ουράς GKQ.....	31
Σχήμα 6 Σύστημα πραγματικής υλοποίησης.....	34
Σχήμα 7 Σχέση των συναρτήσεων του TCP με τις υπόλοιπες του πυρήνα.....	35
Σχήμα 8 Ψευδοκώδικας για τον αλγόριθμο WTP.....	37
Σχήμα 9 Αρχιτεκτονική της πλατφόρμας ALTQ.....	39
Σχήμα 10 Ψευδοκώδικας για τον αλγόριθμο VQ.....	41
Σχήμα 11 Βασική τοπολογία δικτύου.....	42
Σχήμα 12 Λόγος της διαπερατότητας (προσομοίωση): $k=0.5$, $C=10$ Mbps, $RTT=200$ msec, $N=10$. RED: $\min_{thr}=5$, $\max_{thr}=15$, $\max_p=0.1$, $w_q=0.002$. VQ: $vb=0.95\cdot30$, $vc=0.95\cdot10$ Mbps. LBM: $t_{avg}=0.5$, $p_0=0.6$, $\alpha=0.71$	43
Σχήμα 13 Λόγος διαπερατότητας (πραγματική υλοποίηση): $k=0.5$, $C=10$ Mbps, $RTT=200$ msec, $N=10$. RED: $\min_{thr}=5$, $\max_{thr}=15$, $\max_p=0.1$, $w_q=0.002$. VQ: $vb=0.7\cdot30$, $vc=0.7\cdot10$ Mbps.	44
Σχήμα 14 Λόγος μαρκαρισμένων πακέτων (προσομοίωση): $k=0.5$, $C=10$ Mbps, $RTT=200$ msec, $N=10$. RED: $\min_{thr}=5$, $\max_{thr}=15$, $\max_p=0.1$, $w_q=0.002$. VQ: $vb=0.95\cdot30$, $vc=0.95\cdot10$ Mbps. LBM: $t_{avg}=0.5$, $p_0=0.6$, $\alpha=0.71$	45
Σχήμα 15 Επίδραση του παράγοντα \bar{k} (αποτελέσματα προσομοίωσης): $C=10$ Mbps, $RTT=200$ msec $w=1$. VQ: $vb=0.9\cdot30$, $vc=0.9\cdot10$ Mbps.....	46
Σχήμα 16 Επίδραση του παράγοντα \bar{k} (αποτελέσματα πραγματικής υλοποίησης): $C=10$ Mbps, $RTT=200$ msec $w=1$. VQ: $vb=0.9\cdot30$, $vc=0.9\cdot10$ Mbps.	47
Σχήμα 17 Διαφοροποίηση υπηρεσίας για τους μηχανισμούς MulTCP και WTP: $C=10$ Mbps, $RTT=200$ ms, $k=0.5$, $N=10$, $Nl=5$ ($wl=1$).....	47
Σχήμα 18 Μέση χρησιμοποίηση δικτύου για τους μηχανισμούς MulTCP και WTP: $C=10$ Mbps, $RTT=200$ ms, $k=0.5$, $N=10$, $Nl=5$ ($wl=1$).....	48
Σχήμα 19 Σύγκλιση συνδέσεων WTP με τη χρήση του μηχανισμού Slow start.....	49
Σχήμα 20 Σύγκλιση συνδέσεων WTP με τη χρήση του μηχανισμού Large K.....	50
Σχήμα 21 Σύγκλιση συνδέσεων WTP για τους μηχανισμούς Slow start και Large K.....	50
Σχήμα 22 Λόγος της διαπερατότητας συνδέσεων WTP με χρήση μηχανισμού γρήγορης σύγκλισης Slow start.....	52
Σχήμα 23 Λόγος της διαπερατότητας συνδέσεων WTP με χρήση μηχανισμού γρήγορης σύγκλισης weighted Slow start.....	52
Σχήμα 24 Λόγος της διαπερατότητας συνδέσεων WTP με χρήση μηχανισμού γρήγορης σύγκλισης MulTCP Slow start.....	53
Σχήμα 25 Λόγος της διαπερατότητας συνδέσεων WTP με χρήση μηχανισμού γρήγορης σύγκλισης Large K.....	54

Σχήμα 26 Χρόνοι σύγκλισης	55
Σχήμα 27 Αναπαράσταση σύγκλισης για τους αλγόριθμους <i>Slow start</i> και <i>Large K</i>	57
Σχήμα 28 Παράδειγμα αργής σύγκλισης στο σημείο ισορροπίας κατά τη μετάβαση της σύνδεσης από τη φάση γρήγορης σύγκλισης στη φάση αποφυγής της συμφόρησης για το μηχανισμό <i>Large K</i>	58
Σχήμα 29 Παράδειγμα αξιοποίησης αλγόριθμων διαφοροποίησης υπηρεσιών στο <i>Internet</i>	59

1. Εισαγωγή

Το Internet σήμερα στερείται από μηχανισμούς οι οποίοι επιτρέπουν την αποδοτική και σταθερή λειτουργία του δικτύου αλλά και την “υγιή” ανάπτυξη αυτού, ώστε να καθίσταται δυνατή η αντιμετώπιση του φαινομένου της αυξανόμενης ζήτησης που παρουσιάζεται στις μέρες μας. Πιο συγκεκριμένα, η ραγδαία αύξηση των απαιτήσεων των χρηστών στο Internet, οδηγεί στην ανάγκη παροχής διαφοροποιημένων υπηρεσιών, ώστε να απολαμβάνουν καλύτερη απόδοση όσοι τη χρειάζονται και είναι διατεθειμένοι να πληρώσουν γι’ αυτή. Παράλληλα η ανάγκη διαφοροποίησης υπηρεσιών στο Internet, γίνεται επιτακτική και από την πλευρά των παροχών δικτύου, αφού επιτυγχάνεται με τον τρόπο αυτό αποδοτικότερη διαχείριση των πόρων ενός δικτύου αλλά αυξάνεται και η ανταγωνιστικότητά τους.

Η αρχιτεκτονική των διαφοροποιημένων υπηρεσιών (Differentiated Services – DiffServ) αποτελεί μια προσέγγιση για την αντιμετώπιση των παραπάνω θεμάτων, προσφέροντας διαφοροποίηση υπηρεσιών στο Internet. Στην αρχιτεκτονική αυτή, ένας μικρός αριθμός από συμπεριφορές προώθησης (forwarding behaviors), υλοποιούνται στους δρομολογητές και προσφέρουν διαφοροποίηση στην ποιότητα υπηρεσίας που παρέχεται από το δίκτυο. Τα μειονεκτήματα αυτής της προσέγγισης είναι η αυξημένη πολυπλοκότητα, σε σχέση με τους μηχανισμούς προώθησης πακέτων που υπάρχουν στο Internet σήμερα, καθώς και η ανάγκη συνεργασίας μεταξύ των δρομολογητών του δικτύου ώστε να υπάρξει δυνατότητα παροχής υπηρεσιών από άκρο σε άκρο (end-to-end services).

Από την άλλη μεριά ένα νέο πεδίο έρευνας έχει σήμερα αναδειχθεί [1, 2, 3], στα πλαίσια του οποίου ερευνάται πώς τα ζητήματα που αναφέραμε στην αρχή μπορούν να αντιμετωπιστούν, διαθέτοντας ένα “απλό” δίκτυο το οποίο υποστηρίζει σήματα ανάδρασης (feedback) μέσω των οποίων ενημερώνει τους χρήστες για το επίπεδο της συμφόρησης που προκαλούν σε αυτό. Τέτοιου είδους σήματα ανάδρασης μπορούν να παραχθούν χρησιμοποιώντας το μηχανισμό Explicit Congestion Notification (ECN), ο οποίος αποτελεί νέο πρότυπο του Internet [4]. Συνδυάζοντας τον παραπάνω μηχανισμό με ένα μικρό χρηματικό ποσό, για παράδειγμα χρεώνοντας κάθε μαρκαρισμένο πακέτο που στέλνει το δίκτυο στους χρήστες, οι χρήστες θα έχουν την απαραίτητη πληροφορία αλλά και το κατάλληλο κίνητρο να αντιδρούν στα σήματα συμφόρησης με ένα τρόπο ο οποίος οδηγεί το σύστημα (δίκτυο και χρήστες) σε μία αποδοτική και σταθερή λειτουργία. Έτσι οι χρήστες είναι ελεύθεροι να αντιδρούν στα σήματα συμφόρησης όπως εκείνοι επιθυμούν, ανάλογα με τις ανάγκες και τις απαιτήσεις τους. Η παραπάνω προσέγγιση είναι διαφορετική από τον τρόπο της λειτουργίας του Internet όπως αυτή υφίσταται σήμερα, αφού η σταθερότητα του Internet βασίζεται σε μεγάλο βαθμό, από το γεγονός ότι όλα τα τελικά-συστήματα (end-systems) χρησιμοποιούν τον ίδιο τρόπο προσαρμογής σε καταστάσεις

συμφόρησης, όπως αυτός υλοποιείται από το πρωτόκολλο μεταφοράς Transmission Control Protocol (TCP).

Στα πλαίσια της παρούσας μεταπτυχιακής εργασίας, μελετήσαμε μία παραλλαγή του μηχανισμού αποφυγής της συμφόρησης (Congestion Avoidance) του TCP, που υποστηρίζει διαφοροποίηση υπηρεσιών και στηρίζεται στην παραπάνω προσέγγιση. Ερευνήσαμε τη συμπεριφορά του, όσον αφορά την ικανότητα για παροχή διαφοροποίησης υπηρεσιών και μελετήσαμε την αλληλεπίδρασή του με αλγόριθμους μαρκαρίσματος, όπως για παράδειγμα ο Random Early Detect (RED) [6]. Ταυτόχρονα προτείναμε και μελετήσαμε αλγόριθμους για τη γρήγορη σύγκλιση των συνδέσεων, οι οποίοι συνεργάζονται με τον παραπάνω αλγόριθμο αποφυγής της συμφόρησης, ώστε να παρέχεται διαφοροποίηση υπηρεσίας στο Internet.

1.1. DiffServ

Η αρχιτεκτονική Differentiated Services (DiffServ) προσφέρει ένα περιβάλλον στο οποίο παρέχεται προς τους πελάτες ένας αριθμός από διαφοροποιημένες υπηρεσίες τόσο σε ποιότητα όσο και σε χρέωση.

Όταν μία ροή κίνησης ενός χρήστη αιτείται παροχή ποιότητας υπηρεσίας από το δίκτυο, τα πακέτα της σημαδεύονται ανάλογα, στην είσοδο του δικτύου. Συγκεκριμένα ανατίθεται στο Type Of Service (TOS) byte της επικεφαλίδας του πρωτοκόλλου IPv4 μια συγκεκριμένη κωδική τιμή (Differentiated Services Code Point - DSCP). Σύμφωνα με την τιμή του DSCP τα πακέτα έχουν διαφορετική αντιμετώπιση στους δρομολογητές του δικτύου. Η αντιμετώπιση αυτή ονομάζεται Per Hop Behavior (PHB) και αφορά στην προτεραιότητα της προώθησης των πακέτων, στις πιθανές απώλειες πακέτων και στις εγγυήσεις σε εύρος ζώνης. Ο τρόπος χρήσης του TOS byte ορίζεται από την αρχιτεκτονική DiffServ και σε αυτή αναφέρεται ως πεδίο DS. Σε κάθε τιμή DSCP αντιστοιχεί μια συγκεκριμένη PHB. Η αντιστοιχία μεταξύ των DSCP και των PHB αφορά στο διαχειριστή του δικτύου. Με τον τρόπο αυτό μπορούν να υλοποιηθούν διάφορες κλάσεις ποιότητας υπηρεσίας, στις οποίες κατηγοριοποιείται ένα πακέτο βάσει της τιμής του πεδίου DS της επικεφαλίδας του. Στην πραγματικότητα, οι διάφορες PHB είναι μόνο συστατικά για το χτίσιμο ολοκληρωμένων κλάσεων ποιότητας υπηρεσίας. Οι παροχείς υπηρεσιών δικτύου (ISP) συνδυάζοντας τις υλοποιήσεις των PHB με ελεγκτές κυκλοφορίας (που μορφοποιούν, σημαδεύουν ή απορρίπτουν κίνηση ανάλογα), με στρατηγικές παροχής υπηρεσιών και με μοντέλα χρέωσης, μπορούν να δώσουν ολοκληρωμένες κλάσεις ποιότητας υπηρεσίας στους πελάτες τους.

Για να παρέχονται σε ένα πελάτη διαφοροποιημένες υπηρεσίες είναι αναγκαίο να υπάρχει μεταξύ του πελάτη και του ISP μια συμφωνία για το επίπεδο υπηρεσίας (Service Level Agreement – SLA). Το SLA αποτελεί συμφωνία σχετικά με το ποιες διαφοροποιημένες υπηρεσίες παρέχονται σε έναν συγκεκριμένο πελάτη, τα κριτήρια

ελέγχου (προσδιορισμός ροής, προγραμματισμός παροχής, το κόστος και άλλα) της κίνησης που εισάγεται στο δίκτυο και τις παραμέτρους της κίνησης. Ένα υποσύνολο του SLA ορίζει τα τεχνικά χαρακτηριστικά (π.χ. τις παραμέτρους ελέγχου κίνησης του μοντέλου Leaky Bucket, τις εγγυήσεις σε εύρος ζώνης και σε καθυστέρηση) των υπηρεσιών που παρέχονται και ονομάζεται Service Level Specification (SLS). Οι υπηρεσίες DiffServ παρέχονται μόνο για κίνηση μιας κατεύθυνσης. Για το λόγο αυτό η συμπεριφορά του δικτύου για κάθε κατεύθυνση ορίζεται από διαφορετικά SLAs. Ένα SLA μπορεί να είναι στατικό ή δυναμικό. Για τα στατικά SLAs η διαπραγμάτευση μεταξύ του πελάτη και του ISP γίνεται ανά μεγάλα χρονικά διαστήματα, για παράδειγμα μηνιαίως ή ετησίως. Οι πελάτες με δυναμικά SLAs πρέπει να χρησιμοποιούν κάποιο πρωτόκολλο σηματοδότησης (π.χ. RSVP), προκειμένου να ζητούν άλλες υπηρεσίες κατά βούληση.

Οι απαραίτητες λειτουργίες για την εφαρμογή των SLAs στα πακέτα των διαφόρων ροών κίνησης, υλοποιούνται στους δρομολογητές του δικτύου DiffServ από τις εξής οντότητες: την Packet Classifier, την Meter, την Marker και την traffic Conditioner.

- Η οντότητα Packet Classifier, είναι υπεύθυνη για την επεξεργασία ενός ή περισσότερων πεδίων του πακέτου προκειμένου να το μεταχειριστεί σύμφωνα με το SLA της ροής κίνησης στην οποία ανήκει. Υπάρχουν δυο τύποι ταξινομών (classifiers): ο Bandwidth Aggregate (BA) Classifier, ο οποίος διαχωρίζει την κίνηση στο επίπεδο ομαδοποιημένων ροών κίνησης με βάση το πεδίο DS των πακέτων της κυκλοφορίας, και ο Multi-Field (MF) Classifier, ο οποίος διαχωρίζει την κίνηση, με βάση την τιμή ενός συνδυασμού πεδίων της επικεφαλίδας του πακέτου (π.χ. διεύθυνση πηγής και προορισμού, πεδίο DS, διεπαφή εισόδου, κ.α.) προκειμένου να βρει τη ροή στην οποία ανήκει κάθε πακέτο.
- Η οντότητα Meter συλλέγει στοιχεία για την κυκλοφορία σε κάθε PHB και άλλα στατιστικά στοιχεία για την κίνηση.
- Η οντότητα Marker σημαδεύει τα ασημάδευτα πακέτα (το πεδίο DS) ανάλογα με τη ροή κίνησης στην οποία ανήκουν ή τα σημαδεμένα πακέτα σύμφωνα με την τιμή του DSCP που έχουν ήδη (σε περίπτωση αντιστοίχισης μεταξύ DSCP από τον διαχειριστή του δικτύου ή όπως ορίζει η πολιτική που ακολουθείται τοπικά).
- Η οντότητα Traffic Conditioner εφαρμόζει σε κάθε πακέτο μιας ροής κίνησης τη PHB που του αντιστοιχεί. Συχνά περιλαμβάνει τις λειτουργίες της Meter και της Marker, αλλά οι κυρίως λειτουργίες που επιτελεί είναι: η επιλογή της ουράς εξυπηρέτησης για τα εισερχόμενα πακέτα, καθώς και λειτουργίες μορφοποίησης και απόρριψης πακέτων ανά ροή κίνησης ή σε

επίπεδο ομαδοποιημένων ροών, προκειμένου να συμμορφώνεται η κίνηση με τους όρους του SLA που έχει επιλεγθεί και με τη συνολική χωρητικότητα ανά PHB.

Η λειτουργικότητα των παραπάνω στοιχείων, πραγματοποιείται σε επίπεδο ροών ή ομαδοποιημένων ροών σε κάθε δρομολογητή ενός δικτύου DiffServ. Ο διαχωρισμός κίνησης τύπου MF είναι εξαιρετικά επιβαρυντικός σε υπολογιστικό φόρτο για ένα δρομολογητή. Επίσης, το μαρκάρισμα της κίνησης, όπως έχει ήδη αναφερθεί, πρέπει να γίνεται στην είσοδο του δικτύου DiffServ. Η συμμόρφωση της κίνησης στους όρους των συμβολαίων που τη διέπουν, ανά ροή, είναι και αυτή εξαιρετικά επιβαρυντική σε υπολογιστικό φόρτο για τους δρομολογητές. Για τους λόγους αυτούς οι παραπάνω λειτουργίες πρέπει να λαμβάνουν χώρα στους δρομολογητές εισόδου του δικτύου ή ακόμα και στους υπολογιστές των πελατών που αποστέλλουν την κίνηση, αν είναι δυνατό και μπορεί το δίκτυο να τους εμπιστευθεί. Στο εσωτερικό του δικτύου πρέπει να λαμβάνουν χώρα μόνο διαχωρισμός κίνησης τύπου BA και συμμόρφωση της κίνησης στο επίπεδο ομαδοποιημένων ροών. Με τον τρόπο αυτό ο υπολογιστικός φόρτος δεν επιβαρύνει το εσωτερικό ενός δικτύου DiffServ. Τα παραπάνω πρέπει να γίνουν και για έναν επιπλέον λόγο: τα δίκτυα των ISP αποτελούνται από δρομολογητές πρόσβασης και από δρομολογητές κορμού. Οι δρομολογητές κορμού πρέπει να προωθούν τα πακέτα πολύ γρήγορα και γι' αυτό οι λειτουργίες που κάνουν πρέπει να είναι απλές, ενώ οι δρομολογητές πρόσβασης δεν είναι ανάγκη να προωθούν τα πακέτα τόσο γρήγορα λόγω του ότι οι γραμμές πρόσβασης των πελατών είναι σχετικά αργές. Οπότε οι δρομολογητές πρόσβασης είναι σε θέση να ξοδέψουν υπολογιστικό χρόνο για να κάνουν επεξεργασία (μαρκάρισμα, μορφοποίηση, απόρριψη) της κίνησης ανά ροή. Όταν ένα πακέτο εξέρχεται από ένα διαχειριζόμενο περιβάλλον δικτύου και εισέρχεται σε ένα άλλο το πεδίο DS της επικεφαλίδας του μπορεί να μαρκαριστεί ξανά και σύμφωνα με το SLA μεταξύ των διαχειριζόμενων δικτύων.

Το κυριότερο πλεονέκτημα της αρχιτεκτονικής DiffServ είναι η δυνατότητα μεγάλης επεκτασιμότητας που προσφέρει. Παρόλα αυτά, οι απαιτήσεις υλοποίησης των μηχανισμών που αποτελούν την αρχιτεκτονική αυτή ξεπερνούν κατά πολύ σε πολυπλοκότητα τους ήδη υπάρχοντες μηχανισμούς προώθησης πακέτων που χρησιμοποιούνται σήμερα στο Internet. Ταυτόχρονα, αξίζει να σημειωθεί σε αυτό το σημείο, ότι για να παρέχεται ποιότητα υπηρεσιών από άκρο σε άκρο με τη χρήση της αρχιτεκτονικής DiffServ απαιτείται η συνεργασία των δρομολογητών που συμμετέχουν. Στην επόμενη ενότητα θα δούμε μία διαφορετική προσέγγιση η οποία δεν συναντάει τα προβλήματα αυτά.

1.2.Εναλλακτική προσέγγιση

Σήμερα έχει εμφανιστεί η ανάγκη για μία εναλλακτική προσέγγιση παροχής διαφοροποιημένων υπηρεσιών στο Internet [1, 2, 3], η οποία προτείνει ότι η

διαφοροποίηση στις υπηρεσίες, καθώς και η αποδοτική και σταθερή λειτουργία του δικτύου αλλά και η ανάπτυξή του ώστε να καλύπτει τις αυξανόμενες ανάγκες, μπορεί να επιτευχθεί από ένα δίκτυο το οποίο παρέχει έναν απλό μηχανισμό ανάδρασης προς τους χρήστες. Ένα παράδειγμα τέτοιου μηχανισμού είναι ο “Explicit Congestion Notification”, ο οποίος πληροφορεί τους χρήστες για το κόστος συμφόρησης που προκαλεί η κίνησή τους. Έτσι ο χρήστης θα μπορεί να αντιδρά σε αυτά τα σήματα συμφόρησης χρησιμοποιώντας κάποιον αλγόριθμο ελέγχου του ρυθμού αποστολής.

1.2.1. Explicit Congestion Notification (ECN)

Ο μηχανισμός Explicit Congestion Notification (ECN) αποστέλλει ένα είδος σήματος ειδοποίησης, το οποίο δηλώνει την ύπαρξη συμφόρησης στο δίκτυο, προς ένα τελικό χρήστη ή προς μία εφαρμογή. Με την πληροφορία αυτή, οι χρήστες μπορεί να ρυθμίζουν το ρυθμό αποστολής δεδομένων επιτυγχάνοντας την αποδοτική και χωρίς συμφόρηση λειτουργία του δικτύου. Η βασική ιδέα του μηχανισμού αυτού αποτέλεσε τον κεντρικό πυρήνα παλαιότερων προτάσεων, όπως για παράδειγμα ο μηχανισμός “DECbit” [4] ο οποίος αποτελεί μία από τις πρώτες προσεγγίσεις για τέτοιου είδους μηχανισμούς ανάδρασης. Ένα άλλο παράδειγμα είναι η αρχιτεκτονική δικτύων Frame Relay η οποία χρησιμοποιεί τον μηχανισμό FECN, καθώς επίσης και το σχήμα δυαδικής ανάδρασης του ABR που χρησιμοποιεί το μηχανισμό Explicit Forward Congestion Indication.

Η Floyd [5] πρότεινε τη χρήση του μηχανισμού ECN σε συνδυασμό με το TCP. Πιο συγκεκριμένα σε ένα κείμενο “αίτησης για σχόλια” (Request For Comments – RFC) του οργανισμού IETF [4] περιγράφει με λεπτομέρειες μία υλοποίηση, που αποτελεί την επέκταση του ήδη υπάρχοντος μηχανισμού για έλεγχο συμφόρησης του ευρέως διαδεδομένου πρωτοκόλλου μεταφοράς, TCP. Αναλυτικότερα, η πρόταση αυτή θεωρεί ότι ο μηχανισμός ECN συνυπάρχει με το TCP έτσι ώστε να επιτυγχάνεται “ισότιμη δικαιοσύνη” θεωρώντας ότι όλες οι ροές που υποστηρίζουν το μηχανισμό ECN, θα αποκτούσαν σε μεγάλες χρονικές κλίμακες την ίδια διαπερατότητα με αυτή που αποκτούν οι συνδέσεις TCP που δεν υποστηρίζουν το ECN (TCP friendliness), αλλά χρησιμοποιούν την απώλεια πακέτων ως ένδειξη συμφόρησης.

Το παραπάνω κείμενο προτείνει τη χρήση ενός bit (το CE της επικεφαλίδας IP) για να δηλώνει την ύπαρξη συμφόρησης (δηλαδή ένα απλό σήμα ανάδρασης), ένα bit (ECT) για να δηλώνεται η υποστήριξη ή όχι του μηχανισμού αυτού από μία πηγή και ένα bit για τη δήλωση της μείωσης του παραθύρου συμφόρησης (Congestion Window Reduced – CWR). Η βασική προϋπόθεση είναι ότι υπάρχει ένας μηχανισμός ελέγχου ουράς (queue management control) ο οποίος ενεργοποιεί τα σήματα ECN. Για ένα τέτοιο σκοπό θα μπορούσαμε, για παράδειγμα, να χρησιμοποιήσουμε το μηχανισμό Random Early Detection (RED) [6], με τη διαφορά ότι οι δρομολογητές θα μαρκάρουν τα πακέτα που ανήκουν σε ροές που υποστηρίζουν το μηχανισμό ECN,

αντί να τα πετούν. Παρακάτω θα παρουσιάσουμε μια επέκταση της παραπάνω πρότασης, αναλύοντας πώς μια διαφορετική αντίληψη της δικαιοσύνης μεταξύ των ροών μπορεί να επιτευχθεί αφήνοντας τους τελικούς χρήστες να αντιδρούν λίγο διαφορετικά από ότι στο TCP.

Μοντέλα τα οποία αναπτύχθηκαν ύστερα από διάφορες μελέτες [23, 24, 25] έδειξαν ότι αν τα πακέτα χάνονται ανεξάρτητα το ένα με το άλλο, ας πούμε με πιθανότητα p , τότε η διαπερατότητα (throughput) στη σταθερή κατάσταση (steady state) μίας σύνδεσης TCP, είναι ανάλογη του $\frac{MSS}{RTT\sqrt{p}}$ ¹, όταν αυτή βρίσκεται σε

περίοδο αποφυγής της συμφόρησης (congestion avoidance – CA). Ο παράγοντας της τετραγωνικής ρίζας εμφανίζεται επειδή το TCP στις περιπτώσεις απώλειας πακέτου αντιδρά μειώνοντας στο μισό το παράθυρο συμφόρησης, σε συνδυασμό με το γεγονός ότι ο αριθμός των απολεσθέντων πακέτων είναι ανάλογος με το παράθυρο συμφόρησης. Ένας τρόπος να επιτευχθεί διαφοροποίηση στην ποιότητα υπηρεσίας είναι να αφήσουμε μία σύνδεση να συμπεριφερθεί σαν να ήταν N ταυτόχρονες συνδέσεις, με N φορές τη διαπερατότητα μίας σύνδεσης (βλ. Κεφάλαιο 2, MulTCP) [7]. Και σε αυτή τη περίπτωση, παρόλα αυτά, ο ρυθμός αποστολής συνεχίζει να είναι αντιστρόφως ανάλογος της τετραγωνικής ρίζας της πιθανότητας απώλειας πακέτου. Παρόλα αυτά, η εξάρτηση αυτή δεν είναι εύκολο να αντιμετωπιστεί ακόμα και σε αυτή την περίπτωση, αφού αποτελεί αποτέλεσμα του γεγονότος ότι οι συνδέσεις, που χρησιμοποιούν τέτοιου είδους μηχανισμούς, παρουσιάζουν απότομες μεταβολές του ρυθμού τους. Μία πιο “φυσική” εξάρτηση για τη διαπερατότητα των συνδέσεων θα ήταν αν η αναλογία ήταν $1/p$. Παρακάτω θα παρουσιάσουμε έναν αλγόριθμο που πετυχαίνει μία τέτοια αναλογία. Ο αλγόριθμος περιλαμβάνει σταθερή ελάττωση του ρυθμού σαν αντίδραση σε ένα σήμα συμφόρησης, παρά να μειώνεται το παράθυρο συμφόρησης στο μισό.

1.2.2. Χρέωση Συμφόρησης και ωφελιμότητα χρηστών

Μία προσέγγιση για παροχή διαφοροποιημένης ποιότητας υπηρεσίας στους χρήστες είναι η αναγωγή του προβλήματος σε μία μορφή διαφορετικής χρέωσης. Από τη στιγμή που υιοθετείται μία τέτοια προσέγγιση, είναι φυσικό να χρησιμοποιούμε σήματα συμφόρησης τα οποία αντανακλούν κατά κάποιον τρόπο το κόστος συμφόρησης [17]. Με τον τρόπο αυτό μπορούμε να δώσουμε στους χρήστες ένα βαθμό ελευθερίας, δίνοντας τον έλεγχο στα συστήματα-άκρων (end-systems) και επιτρέποντας τους να καθορίσουν μόνα τους την ποιότητα υπηρεσίας που επιθυμούν

¹ Ως MSS συμβολίζεται το μέγιστο μέγεθος πακέτου που μπορεί να στείλει μία σύνδεση TCP. Το RTT συμβολίζει τον χρόνο αποστολής – άφιξης που υπολογίζεται από το TCP, ενώ συμβολίζουμε με p την πιθανότητα απώλειας πακέτου.

να απολαμβάνουν, γνωρίζοντας ότι θα χρεωθούν αντίστοιχα. Χρησιμοποιώντας “οικονομικούς” όρους, η συμφόρηση είναι μία “συγκυρία”, και υποχρεώνοντας τους χρήστες να χρεώνονται για την επίδρασή τους που έχουν προς τους άλλους που συμμετέχουν σε αυτή τη συγκυρία, προωθούμε την συνεργασία μεταξύ των χρηστών και του δικτύου.

Στηριζόμενοι στα παραπάνω μπορούμε να ορίσουμε ένα βασικό πλαίσιο που χαρακτηρίζεται από δύο θέματα κλειδιά:

α) το δίκτυο παρέχει σήματα ανάδρασης στους χρήστες, τα οποία αντικατοπτρίζουν το κόστος συμφόρησης (πιο συγκεκριμένα τη λογιστική τιμή της συμφόρησης – shadow price) και β) οι χρήστες επιβαρύνονται με κάποιο κόστος το οποίο συνδέεται με τις πράξεις τους, αλλά από την άλλη μεριά έχουν τη δυνατότητα να αντιδρούν όπως αυτοί επιθυμούν.

Θεωρούμε ότι το κόστος της συμφόρησης είναι ο αριθμός των απολεσθέντων πακέτων. Τότε, όλα τα πακέτα που συνεισφέρουν σε κάποια απώλεια πρέπει να “μαρκαριστούν” (χρησιμοποιώντας μαρκαρίσματα ECN) έτσι ώστε να αντανakλούν στη λογιστική τιμή της συμφόρησης. Για παράδειγμα, σε μία ουρά ενός δρομολογητή, τα πακέτα που θα μαρκαριστούν είναι εκείνα που θα κατέφθαναν μεταξύ της έναρξης και λήξης της κατάστασης έντονης κίνησης (busy period), που προηγείται της απώλειας πακέτων.

Σε ένα πιο αφηρημένο πλαίσιο, αν μία πηγή έχει φόρτο y , το οποίο υφίσταται κόστος $C(y)$, τότε η λογιστική τιμή (shadow price) της συμφόρησης είναι η παράγωγος:

$$p(y) = \frac{d}{dy} C(y) \quad (1)$$

Αν ο φόρτος y είναι μία τυχαία μεταβλητή και το κόστος θεωρείται ως ο αναμενόμενος ρυθμός απώλειας, τότε ο αλγόριθμος μαρκαρίσματος σε μία ουρά, όπως περιγράφηκε παραπάνω, επιστρέφει την κατάλληλη πληροφορία της παραγωγού, ως συνάρτηση του φόρτου [1, 8].

Υποθέτουμε ότι R είναι ένας πεπερασμένος αριθμός χρηστών που χρησιμοποιεί ένα πεπερασμένο σύνολο από πόρους J , όπου ο πίνακας A_{jr} , που περιέχει τιμές 0 ή 1, δηλώνει κατά πόσο ο χρήστης r χρησιμοποιεί ή όχι τον πόρο j . Σύμφωνα με τον Shenker [16], χαρακτηρίζουμε την προτίμηση ενός ελαστικού χρήστη για κάποιο ποσό εύρους ζώνης x_r , με μία κοίλη συνάρτηση ωφελιμότητας $U_r(x)$. Η μεγιστοποίηση της κοινωνικής ευημερίας (social welfare) περιλαμβάνει την μεγιστοποίηση της ποσότητας:

$$\text{Μεγιστοποίηση} \quad \sum_r U_r(x_r) - \sum_j C_j \left(\sum_r A_{jr} x_r \right) \quad (2)$$

Η λύση αυτού του προβλήματος βελτιστοποίησης (για μη μηδενικά x_r) είναι:

$$U'_r(x_r) = \sum_{y \in r} p_j(y_j) \quad (3)$$

με τον αντίστοιχο φόρτο σε ένα πόρο j να είναι:

$$y_j = \sum_r A_{jr} x_r \quad (4)$$

Έτσι στο σημείο που φτάνουμε σε κοινωνικό βέλτιστο, η παράγωγος ωφελιμότητας ενός χρήστη ταιριάζει απόλυτα με το άθροισμα των λογιστικών τιμών όλων των πόρων από τους οποίους περνάει η κίνηση του χρήστη.

Συνήθως το δίκτυο δεν γνωρίζει τις συναρτήσεις ωφελιμότητας των χρηστών. Παρόλα αυτά, αν κάποιος χρήστης χρεώνεται με ρυθμό ανάλογο του ποσού εύρους ζώνης x_r που χρησιμοποιεί, ας πούμε με ρυθμό $t_r x_r$, τότε οι χρήστες θα προσπαθούν να μεγιστοποιήσουν το κέρδος δικτύου, δηλαδή θα προσπαθούν να λύσουν το πρόβλημα:

$$\text{Μεγιστοποίηση} \quad U_r - t_r x_r \quad \text{όπου } x_r \geq 0 \quad (5)$$

Είναι προφανές ότι αν οι τιμές είναι σωστές, δηλαδή αν

$$t_r = \sum_{j \in r} p_j(y_j) \quad (6)$$

και αν κάθε χρήστης προσπαθεί να μεγιστοποιήσει το δικό του κέρδος, τότε το σύστημα θα οδηγηθεί στο κοινωνικό βέλτιστο. Η ιδιότητα της ταύτισης των κινήτρων των μεμονωμένων χρηστών με το στόχο της μεγιστοποίησης της κοινωνικής ευημερίας ονομάζεται συμβατότητα κινήτρων (incentive compatibility). Καθώς ο φόρτος του δικτύου αλλάζει, το δίκτυο θα χρειαστεί να ανανεώσει τις τιμές του.

Εν κατακλείδι, χρησιμοποιούμε ένα σήμα ανάδρασης $x_r \sum_{j \in r} p_j(y_j)$, όπου η ανάδραση για κάποια ροή πρέπει να είναι ανάλογη της ροής και επίσης ανάλογη των λογιστικών τιμών (congestion costs) κατά τη διαδρομή από όπου περνά η ροή. Ένα τέτοιο σήμα ανάδρασης σε ένα δίκτυο ECN, μπορεί να αποτελέσουν τα μαρκαρισμένα πακέτα που φτάνουν στους χρήστες, αν σε κάθε δρομολογητή η πιθανότητα μαρκαρίσματος είναι ίση με $p(y)$ της σχέσης 6. Για να ισχύει αυτό, η

πιθανότητα μαρκαρίσματος σε κάθε δρομολογητή θα πρέπει να είναι μικρή, ώστε να μην είναι πιθανό το ίδιο πακέτο να μαρκαριστεί σε περισσότερους από έναν δρομολογητές. Σε αντίθετη περίπτωση επειδή χρησιμοποιείται ένα bit για το μαρκάρισμα των πακέτων δε θα ισχύει το άθροισμα της σχέσης 6.

Είναι ενδιαφέρον να αναφέρουμε στο σημείο αυτό ότι αν χρησιμοποιήσουμε μια συνάρτηση ωφελιμότητας της μορφής:

$$U_r(x_r) = w_r \log x_r \quad (7)$$

για κάποια σταθερά “επιθυμίας χρέωσης χρήστη” (willingness to pay) w_r , τότε όταν το σύστημα φτάσει σε κοινωνικό βέλτιστο θα ισχύει:

$$w_r = x_r \sum_{j \in r} p_j(y_j) \quad (8)$$

Το παραπάνω αποτέλεσμα δηλώνει ότι η χρησιμοποίηση της παραπάνω συνάρτησης ωφελιμότητας προσφέρει “σύμμετρη δικαιοσύνη” (proportional fairness) στους χρήστες.

1.3. Σκοπός της εργασίας και οργάνωση κειμένου

Στα πλαίσια της μεταπτυχιακής αυτής εργασίας, ασχοληθήκαμε με την προσέγγιση που περιγράψαμε στην ενότητα 1.2. Πιο συγκεκριμένα για ένα δίκτυο όπου τα σήματα συμφόρησης έχουν την μορφή μαρκαρίσματος ECN και δεν εμφανίζονται απώλειες πακέτων, μελετήσαμε και συγκρίναμε αλγόριθμους, διερευνώντας παράλληλα την αλληλεπίδρασή τους με αλγόριθμους μαρκαρίσματος, τόσο για τη φάση αποφυγής της συμφόρησης (congestion avoidance), όσο και για τη φάση γρήγορης σύγκλισης (οι οποίοι υποστηρίζουν διαφοροποίηση υπηρεσιών).

Στο πρώτο μέρος της μεταπτυχιακής μας εργασίας στοχεύσαμε στη μελέτη της διαφοροποίησης που επιτυγχάνεται από τους μηχανισμούς ελέγχου συμφόρησης, ενός αλγόριθμου ο οποίος βασίζεται στην τεχνική του παραθύρου (window – based), όπως και το TCP και ο οποίος παρέχει διαφοροποίηση στο ρυθμό αποστολής των χρηστών με τη χρήση βαρών. Παράλληλα, ερευνήσαμε πώς η απόδοση του αλγορίθμου αυτού (στον οποίο θα αναφερόμαστε στη συνέχεια ως Willingness To Pay - WTP) εξαρτάται από τους αλγόριθμους μαρκαρίσματος που χρησιμοποιούνται στα ενεργά στοιχεία του δικτύου. Η μελέτη αφορούσε συνδέσεις μεγάλης χρονικής διάρκειας (για παράδειγμα συνδέσεις FTP για μεταφορά μεγάλων αρχείων). Αναφερόμενοι στη λέξη διαφοροποίηση εννοούμε την ικανότητα των ακραίων συστημάτων (end-systems), συνεργαζόμενα με τους αλγόριθμους μαρκαρίσματος στους δρομολογητές, να προσφέρουν διαφορετική διαπερατότητα στις συνδέσεις με διαφορετικά βάρη ή τιμές επιθυμίας χρέωσης (willingness to pay).

Οι αλγόριθμοι μαρκαρίσματος που χρησιμοποιήσαμε στα πλαίσια της μελέτης μας είναι:

- ο αλγόριθμος Random Early Detect (RED), ο οποίος τροποποιήθηκε κατάλληλα ώστε τα πακέτα να μαρκάρονται μόνο και να μην απορρίπτονται
- ο αλγόριθμος Virtual Queue (VQ), κατά τον οποίο το μαρκάρισμα των πακέτων εξαρτάται από το αν έχει υπερχειλίσει κάποια ιδεατή ουρά (virtual queue) της οποίας η χωρητικότητα και το μέγεθος είναι ένα ποσοστό της πραγματικής χωρητικότητας και μεγέθους της πραγματικής ουράς του συνδέσμου αντίστοιχα
- ο αλγόριθμος Load Based, κατά τον οποίο η πιθανότητα μαρκαρίσματος αποτελεί μία γραμμική συνάρτηση της χρησιμοποίησης (utilization) του συνδέσμου, η οποία υπολογίζεται σε χρονικά διαστήματα ίσης διάρκειας.

Στο δεύτερο μέρος της μεταπτυχιακής μας εργασίας μελετήσαμε τη διαφοροποίηση που προσφέρεται από το WTP για συνδέσεις μικρής χρονικής διάρκειας (για παράδειγμα συνδέσεις HTTP), διαπιστώνοντας ότι τέτοιου είδους συνδέσεις μπορεί να προλάβουν να τερματίσουν χωρίς ο αλγόριθμος WTP να έχει προλάβει να συγκλίνει στο σημείο ισορροπίας. Για το λόγο αυτό προτείναμε και μελετήσαμε εναλλακτικούς αλγόριθμους, οι οποίοι οδηγούν στην γρήγορη σύγκλιση των συνδέσεων, έτσι ώστε στη συνέχεια να εφαρμόζονται οι μηχανισμοί αποφυγής της συμφόρησης του WTP στο σημείο ισορροπίας. Πιο συγκεκριμένα οι αλγόριθμοι που μελετήσαμε είναι:

- Slow start, ο οποίος αποτελεί το μηχανισμό για γρήγορη σύγκλιση στο TCP.
- MulTCP Slow start, που αποτελεί το μηχανισμό για γρήγορη σύγκλιση μίας επέκτασης του TCP (MulTCP), η οποία υποστηρίζει διαφοροποίηση υπηρεσίας.
- Weighted Slow start, ο οποίος αποτελεί μία τροποποίηση του αλγόριθμου Slow start του TCP, σε μία απόπειρα για διαφοροποίηση υπηρεσιών, με τη χρήση βαρών.
- Large K, ο οποίος χρησιμοποιεί τον αλγόριθμο WTP για μεγάλες τιμές του παράγοντα σύγκλισης του αλγόριθμου αυτού.

Στο σημείο αυτό θα πρέπει να τονίσουμε ότι για την εξαγωγή των πειραματικών μας αποτελεσμάτων, υλοποιήσαμε όλους τους αλγόριθμους που περιγράψαμε στον προσομοιωτή δικτύου Network Simulator version 2 (NS2). Παράλληλα υλοποιήσαμε

τον αλγόριθμο WTP καθώς και τον αλγόριθμο μαρκαρίσματος Virtual Queue σε ένα πραγματικό σύστημα ώστε να μπορέσουμε να επαληθεύσουμε τα πειραματικά αποτελέσματα που πήραμε από την προσομοίωση και σε ένα πραγματικό περιβάλλον αλλά και να μελετήσουμε την επιτευξιμότητα των αλγορίθμων αυτών σε πραγματικές συνθήκες.

Στη συνέχεια η παρούσα εργασία οργανώνεται ως εξής:

Στο Κεφάλαιο 2, περιγράφουμε τους αλγόριθμους παροχής διαφοροποιημένων υπηρεσιών στα άκρα του δικτύου που ερευνήσαμε, τόσο για τη φάση της γρήγορης σύγκλισης των συνδέσεων όσο και για τη φάση αποφυγής της συμφόρησης. Παράλληλα περιγράφουμε τους μηχανισμούς μαρκαρίσματος, που μελετήσαμε στα πειράματά μας και οι οποίοι υλοποιούνται στους δρομολογητές δικτύου.

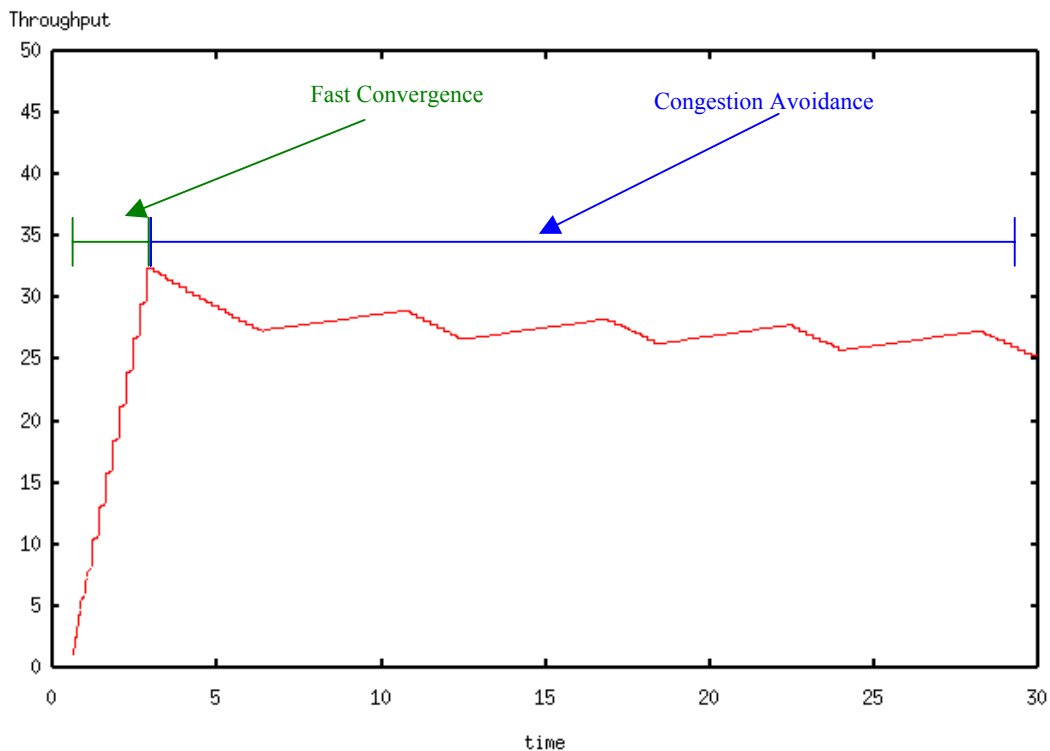
Στο Κεφάλαιο 3, περιγράφουμε την υλοποίηση των παραπάνω αλγορίθμων, τόσο σε προσομοίωση όσο και σε πραγματικό περιβάλλον. Παράλληλα συζητάμε τα διάφορα θέματα που προέκυψαν κατά την υλοποίηση αυτή.

Στο Κεφάλαιο 4, αναφέρουμε τα πειραματικά μας αποτελέσματα, που αφορούν την έρευνά μας γύρω από τους αλγόριθμους που περιγράψαμε στο Κεφάλαιο 2. Για τη σύγκριση αλγορίθμων για τη φάση γρήγορης σύγκλισης χρησιμοποιήσαμε συνδέσεις μικρής διάρκειας, ενώ για τη σύγκριση αλγορίθμων για τη φάση αποφυγής της συμφόρησης χρησιμοποιήσαμε συνδέσεις μεγάλης διάρκειας.

Τέλος στο Κεφάλαιο 5 συνοψίζουμε τα συμπεράσματα της μεταπτυχιακής μας εργασίας και αναφέρουμε επεκτάσεις της εργασίας και σχετικά θέματα για μελλοντική διερεύνηση.

2. Μηχανισμοί διαφοροποίησης υπηρεσιών στο Internet

Στο Σχήμα 1, βλέπουμε τις φάσεις στις οποίες μπαίνουν οι διάφορες συνδέσεις στο Internet, ανάλογα με το αν υφίσταται συμφόρηση ή όχι σε κάποιο σύνδεσμο του δικτύου. Αναλυτικότερα, μία σύνδεση ξεκινώντας την αποστολή των δεδομένων, αυξάνει σχεδόν εκθετικά, τον ρυθμό της προσπαθώντας με αυτόν τον τρόπο, να αποκτήσει όσο το δυνατόν γρηγορότερα το εύρος ζώνης που μπορεί να πάρει από το δίκτυο. Αυτή τη φάση θα την ονομάζουμε φάση γρήγορης σύγκλισης (fast convergence). Ο αυξανόμενος αυτός ρυθμός αποστολής των συνδέσεων, όμως, οδηγεί το δίκτυο σε καταστάσεις συμφόρησης. Έτσι για να αποφευχθεί μια πιθανή κατάρρευση του δικτύου λόγω συμφόρησης, οι συνδέσεις μπαίνουν σε μία δεύτερη φάση που προσπαθεί να αντιμετωπίσει τις καταστάσεις συμφόρησης που εμφανίστηκαν και την οποία ονομάζουμε φάση αποφυγής της συμφόρησης (congestion avoidance phase).



Σχήμα 1. Φάσεις γρήγορης σύγκλισης και αποφυγής της συμφόρησης για συνδέσεις στο Internet.

Μία κατάσταση συμφόρησης, μπορεί να γίνει αντιληπτή από μία σύνδεση όταν παρουσιαστεί απώλεια κάποιου πακέτου. Στην περίπτωση όμως, που είναι επιθυμητό η συμφόρηση να προβλεφθεί, αποφεύγοντας με αυτόν τον τρόπο οποιαδήποτε απώλεια πακέτου, χρησιμοποιούνται μηχανισμοί ανάδρασης από το δίκτυο, π.χ. ECN, ειδοποιώντας έτσι, τα άκρα να ελαττώσουν τους ρυθμούς αποστολής τους.

Στη συνέχεια του κεφαλαίου αυτού περιγράφουμε τους διάφορους μηχανισμούς στα άκρα του δικτύου, οι οποίοι υποστηρίζουν διαφοροποίηση υπηρεσίας (εκτός του TCP) και δρουν ακολουθώντας τις φάσεις που περιγράψαμε παραπάνω. Ταυτόχρονα, περιγράφουμε και τους μηχανισμούς μαρκαρίσματος που χρησιμοποιούνται από το δίκτυο, το οποίο υποστηρίζει το μηχανισμό ECN, ώστε να προειδοποιεί τα άκρα για καταστάσεις συμφόρησης.

2.1. TCP Congestion Avoidance

Στη συνέχεια παρουσιάζουμε και αναλύουμε τον αλγόριθμο “Αποφυγής της Συμφόρησης” (Congestion Avoidance) όπως αυτός προτάθηκε από τον Jacobson [9] και υλοποιείται σήμερα στις περισσότερες εκδόσεις του πρωτοκόλλου TCP.

Εν γένει, τα πακέτα σε ένα δίκτυο χάνονται για δύο λόγους: α) καταστράφηκαν κατά τη μετάδοσή τους ή β) το δίκτυο βρίσκεται σε κατάσταση συμφόρησης και κάπου στο μονοπάτι δεν υπήρχε αρκετός αποθηκευτικός χώρος (buffer capacity). Στα περισσότερα δίκτυα, η απώλεια λόγω καταστροφής πακέτων είναι αρκετά σπάνια (<< 1%). Έτσι είναι πολύ πιθανό ότι κάποια απώλεια πακέτου οφείλεται σε συμφόρηση δικτύου.

Μία στρατηγική αποφυγής καταστάσεων συμφόρησης, όπως προτάθηκε από τους Ramakrishnan και Chiu [10], θα πρέπει να αποτελείται από δύο τμήματα: α) το δίκτυο στέλνει σήματα στα άκρα ότι βρίσκεται σε κατάσταση συμφόρησης και β) τα άκρα έχουν μία πολιτική βάσει της οποίας ελαττώνουν τη χρησιμοποίηση του δικτύου στην περίπτωση που έλαβαν σήμα συμφόρησης, ενώ αυξάνουν τη χρησιμοποίηση του δικτύου σε αντίθετη περίπτωση. Ακολουθώντας την προσέγγιση του Jacobson, κάθε απώλεια πακέτου μπορεί να θεωρηθεί ως σήμα συμφόρησης, αφού στις περισσότερες περιπτώσεις οι απώλειες πακέτων είναι αποτέλεσμα συμφόρησης. Το σημαντικό πλεονέκτημα της προσέγγισης αυτής είναι το γεγονός ότι σε όλα τα υπάρχοντα δίκτυα, τέτοιου είδους σήματα μεταδίδονται αυτόματα χωρίς να χρειάζεται καμία μετατροπή (αντίθετα στο [10] απαιτεί ένα νέο bit στις επικεφαλίδες των πακέτων και μετατροπές σε όλες τις θύρες δικτύου (gateways) ώστε να θέτουν αυτό το bit).

Το άλλο τμήμα μίας στρατηγικής για αποφυγή συμφόρησης, είναι οι ενέργειες που θα πρέπει να κάνουν τα άκρα. Στη περίπτωση του TCP υιοθετείται το σχήμα DEC/ISO που ακολουθεί το μοντέλο χρονοσειρών πρώτης τάξης (first-order time-series) για δίκτυα. Αναλυτικότερα ας θεωρήσουμε ότι ο φόρτος ενός δικτύου υπολογίζεται ως η μέση τιμή του μεγέθους της ουράς η οποία μετράται σε συγκεκριμένες χρονικές στιγμές (κοντά στο RTT). Αν L_i είναι ο φόρτος τη χρονική στιγμή i , τότε ένα μη συμφορημένο δίκτυο μπορεί να μοντελοποιηθεί λέγοντας ότι το L_i αλλάζει αργά σε σχέση με το χρόνο δειγματοληψίας: $L_i = N$ (όπου N σταθερό). Αν το δίκτυο τείνει προς συμφόρηση, τότε αυτό το μοντέλο μηδενικής τάξης δεν ισχύει πια. Η μέση τιμή του μεγέθους της ουράς σπάει σε ένα άθροισμα δύο όρων: ο όρος N

όπως παραπάνω, που μετράει τη μέση τιμή του ρυθμού άφιξης των νέων πακέτων και την εσωτερική καθυστέρηση και ένας νέος όρος, που μετράει το ποσοστό της κίνησης που παρέμεινε από την τελευταία χρονική περίοδο ανανέωσης και την επίδραση αυτής της κίνησης (π.χ. τις προκύπτουσες αναμεταδόσεις). Τότε έχουμε: $L_i = N + \gamma L_{i-1}$.

Όταν το δίκτυο βρίσκεται σε κατάσταση συμφόρησης, ο όρος γ στην παραπάνω σχέση πρέπει να έχει μεγάλες τιμές και τα μεγέθη των ουρών θα αυξάνουν εκθετικά (το σύστημα συμπεριφέρεται ως: $L_i \approx \gamma L_{i-1}$, που αποτελεί μία διαφορική ισότητα με λύση $L_n = \gamma^n L_0$ η οποία τείνει εκθετικά στο άπειρο για $\gamma > 1$). Το σύστημα, σε μία τέτοια περίπτωση, θα σταθεροποιηθεί μόνο αν οι πηγές κίνησης μειώσουν το ρυθμό αποστολής, τουλάχιστο όσο γρήγορα μεγαλώνουν και οι ουρές. Στη περίπτωση του TCP ο φόρτος που προσφέρεται στο δίκτυο ελέγχεται με το παράθυρο συμφόρησης W , οπότε σύμφωνα με το Jacobson θα πρέπει να ισχύει:

$$W_i = dW_{i-1} \quad (d < 1) \quad (9)$$

Δηλαδή θα πρέπει να παρουσιάζεται πολλαπλασιαστική μείωση (multiplicative decrease) του παραθύρου (η οποία γίνεται εκθετική μείωση στη ροή του χρόνου αν η κατάσταση συμφόρησης διαρκεί).

Αν δεν υπάρχει συμφόρηση τότε ο παράγοντας γ πρέπει να παίρνει τιμές κοντά στο μηδέν και ο φόρτος να είναι σχεδόν σταθερός. Το δίκτυο ειδοποιεί τα άκρα, μέσω της απώλειας πακέτου, όταν η ζήτηση είναι υπερβολική, αλλά δεν ανακοινώνει τίποτα στην περίπτωση που η σύνδεση χρησιμοποιεί λιγότερο εύρος ζώνης από ότι δικαιούται. Έτσι μία σύνδεση πρέπει να αυξάνει τη χρησιμοποίηση εύρους ζώνης ώστε να ανακαλύψει το τρέχον όριο. Για παράδειγμα, έστω ότι κάποιος χρήστης μοιράζεται το μονοπάτι δικτύου με κάποιον άλλο και συγκλίνει σε ένα παράθυρο το οποίο του προσφέρει το μισό εύρος ζώνης. Σε περίπτωση που ο άλλος χρήστης τερματίσει τη σύνδεση, το 50% του συνδέσμου θα μείνει αχρησιμοποίητο εκτός και αν το παράθυρο του ενεργού χρήστη αυξηθεί. Οπότε τίθεται το ζήτημα ποια θα είναι η πολιτική αυτής της αύξησης.

Μια προσέγγιση θα ήταν η χρησιμοποίηση πολλαπλασιαστικής αύξησης (σαν σύμμετρη αντιμετώπιση της πολλαπλασιαστικής μείωσης), πιθανότατα με μεγαλύτερη χρονική διάρκεια μετρήσεων, δηλαδή $W_i = bW_{i-1}$, $1 < b \leq 1/d$. Το αποτέλεσμα όμως μίας τέτοιας προσέγγισης θα ήταν η εμφάνιση μεγάλων ταλαντώσεων του ρυθμού αποστολής γύρω από το σημείο ισορροπίας, το οποίο θα οδηγούσε σε μικρό εύρος ζώνης. Η ύπαρξη του φαινομένου αυτού έχει να κάνει με το γεγονός ότι είναι εύκολο να οδηγήσεις το δίκτυο σε κορεσμό αλλά είναι δύσκολη η επιστροφή του σε σταθερή κατάσταση [11].

Μία πιο σωστή πολιτική αύξησης του παραθύρου είναι αυτή που υιοθετείται από το TCP. Πιο συγκεκριμένα αυτό που συμβαίνει είναι να επιτελούνται μικρές και

σταθερές αλλαγές στο μέγεθος του παραθύρου. Έτσι σε καταστάσεις μη συμφόρησης: $W_i = W_{i-1} + u$ με $u \ll W_{max}$, όπου W_{max} είναι το γινόμενο καθυστέρησης – εύρους ζώνης (delay – bandwidth product) μείον τις επιβαρύνσεις (overhead) του πρωτοκόλλου. Αυτή η μέθοδος αποτελεί την “προσθετική αύξηση” (additive increase) που σε συνδυασμό με την “πολλαπλασιαστική μείωση” που αναφέραμε παραπάνω υιοθετούνται από το TCP.

Οι ενέργειες που χρειάζεται να υλοποιήσει το TCP για την εφαρμογή του ελέγχου συμφόρησης είναι:

- Σε κάθε απώλεια πακέτου, το παράθυρο συμφόρησης παίρνει τη μισή τιμή (πολλαπλασιαστική μείωση).
- Για κάθε επιβεβαίωση λήψης καινούριων δεδομένων, το παράθυρο συμφόρησης αυξάνεται κατά $1/cwnd$ όπου $cwnd$ είναι η τρέχουσα τιμή του παραθύρου (προσθετική αύξηση).
- Ο αριθμός των πακέτων που πρόκειται να αποσταλούν κάθε φορά, είναι το ελάχιστο του διαφημιζόμενου παραθύρου (advertised window) του παραλήπτη και του παραθύρου συμφόρησης.

2.2. *MulTCP*

Ο μηχανισμός *MulTCP* αποτελεί μία επέκταση του πρωτοκόλλου TCP στο οποίο μία ροή που χρησιμοποιεί τον αλγόριθμο αυτό συμπεριφέρεται σαν να ήταν μία συλλογή από πολλαπλές ιδεατές συνδέσεις TCP. Σκοπός του αλγορίθμου *MulTCP* [7] είναι να υπάρξει ένας μηχανισμός ελέγχου παρόμοιος με αυτόν του TCP, ο οποίος θα δέχεται ως παράμετρο ένα παράγοντα N και θα έχει σαν αποτέλεσμα να παίρνει σε ένα συμφορημένο σύνδεσμο το ίδιο ποσό εύρους ζώνης με αυτό που θα λάμβαναν N συνδέσεις TCP.

2.2.1. *MulTCP Congestion Avoidance*

Όταν μία σύνδεση TCP βρίσκεται σε φάση αποφυγής της συμφόρησης (Congestion Avoidance) το παράθυρο συμφόρησης αυξάνεται κατά ένα πακέτο ανά RTT, ή κατά $1/cwnd$ για κάθε επιβεβαίωση λήψης. Οι N συνδέσεις TCP αυξάνουν το παράθυρο συμφόρησης κατά N πακέτα ανά RTT ή κατά $N/cwnd$ ανά επιβεβαίωση λήψης. Επομένως με τον ίδιο τρόπο δρα και ο αλγόριθμος *MulTCP*, όταν πρέπει να αυξήσει το παράθυρο συμφόρησης σε κατάσταση αποφυγής της συμφόρησης.

Όταν μία σύνδεση TCP ανακαλύπτει συμφόρηση παρατηρώντας απώλεια πακέτου, μειώνει το παράθυρο συμφόρησης στο μισό, θέτει το κατώφλι Slow start (slow start threshold – ssthresh) σε τιμή ίση με αυτή του παραθύρου συμφόρησης και επιστρέφει στη φάση γραμμικής αύξησης. Όταν N συνδέσεις TCP στέλνουν δεδομένα

και ένα πακέτο χαθεί, μόνο μία από αυτές μειώνει το παράθυρο συμφόρησης στο μισό. Έτσι και ο αλγόριθμος MulTCP, όταν ανακαλύπτει απώλεια πακέτου, μειώνει μόνο το $1/N$ από το παράθυρο συμφόρησης θέτοντάς του τιμή ίση με $(N-0.5)/N$ του τρέχοντος παραθύρου συμφόρησης. Με το τρόπο αυτό θεωρείται ότι τη χρονική στιγμή της απώλειας πακέτου όλες οι N ιδεατές συνδέσεις TCP έχουν την ίδια τιμή για το παράθυρο συμφόρησης. Αυτό, μακροσκοπικά, είναι αληθές από τη στιγμή που υπάρχει ίσος διαμοιρασμός των πόρων μεταξύ των ιδεατών συνδέσεων. Επιπλέον, αναλύοντας το γεγονός αυτό σε μεγαλύτερο βάθος, μπορούμε εύκολα να παρατηρήσουμε ότι N συνδέσεις TCP έχουν στο σύνολο k απώλειες πακέτων τυχαία μοιρασμένες μεταξύ τους και ένα σύνολο από παράθυρα συμφόρησης τα οποία παρουσιάζουν μέση τιμή ίση με $((N-0.5)/N)^k$. Αυτό είναι ίσο με το παράθυρο συμφόρησης μίας σύνδεσης TCP η οποία μειώνει το παράθυρό της κατά $(N-0.5)/N$ για κάθε απώλεια πακέτου.

2.3. Willingness To Pay congestion avoidance (WTP)

Στην ενότητα αυτή, περιγράφουμε το μηχανισμό WTP, που αναφέραμε στο Κεφάλαιο 1 και ο οποίος αποτέλεσε βασικό αντικείμενο της μεταπτυχιακής μας έρευνας. Ο αλγόριθμος αυτός, όπως θα αναλύσουμε στη συνέχεια είναι μία επέκταση του μηχανισμού αποφυγής της συμφόρησης του TCP, σε δίκτυα που υποστηρίζουν το μηχανισμό ECN.

Στη συνέχεια αναπτύσσουμε τη θεωρητική προσέγγιση του αλγορίθμου και καταλήγουμε πώς αυτός μπορεί να εφαρμοστεί στο TCP με πολύ μικρές τροποποιήσεις στους μηχανισμούς αποφυγής της συμφόρησης.

Θεωρούμε ένα δίκτυο από J πόρους [12, 13]. Κάθε πόρος (σύνδεσμος) j μαρκάρει τα πακέτα με πιθανότητα $p_j(y_j)$, όπου y_j είναι ο συνολικός ρυθμός άφιξης στον πόρο j .

Έστω R είναι το σύνολο των διαδρομών ή των συνδέσεων, οι οποίες είναι ενεργές στο δίκτυο. Κάθε σύνδεση r ανανεώνει τον ρυθμό της x_r σύμφωνα με την ισότητα:

$$\frac{dx_r(t)}{dt} = k_r \left(w_r - x_r(t) \sum_{j \in r} \mu_j(t) \right) \quad (10)$$

όπου w_r είναι ένα βάρος, k_r είναι μία σταθερά που ελέγχει το ρυθμό σύγκλισης και

$$\mu_j(t) = p_j \left(\sum_{s: j \in s} x_s(t) \right) \quad (11)$$

είναι η πιθανότητα μαρκαρίσματος στον πόρο j . Με τη σχέση (10), μία σύνδεση r ρυθμίζει το ρυθμό αποστολής της x_r , ώστε ο ρυθμός των μαρκαρισμένων πακέτων που δέχεται $x_r(t) \sum_{j \in R} \mu_j(t)$ να γίνει ίσος με το βάρος w_r . Έτσι, αν θεωρήσουμε ότι το δίκτυο χρεώνει ένα μικρό ποσό για κάθε μαρκαρισμένο πακέτο που επιστρέφει στα άκρα, τότε το βάρος w_r αναπαριστά την επιθυμία χρέωσης για τη σύνδεση r .

Έστω ότι η πιθανότητα μαρκαρίσματος $p(y_j(t))$ εξαρτάται από το κόστος $C_j(y_j)$ το οποίο υφίσταται στον πόρο j ως εξής:

$$p_j(y_j) = \frac{dC_j(y_j)}{dt}. \quad (12)$$

Αποδεικνύεται ότι αν $C_j(y_j(t))$ παραγοντοποιείται και η ανάδραση (μαρκάρισμα πακέτων) είναι στιγμιαία, τότε το παραπάνω σύστημα συγκλίνει στο σημείο που μεγιστοποιεί τη σχέση:

$$\sum_{j \in R} w_r \log x_r - \sum_{j \in J} C_j \left(\sum_{s: j \in s} x_s(t) \right). \quad (13)$$

Αν $U_r(x_r) = w_r \log x_r$ είναι η συνάρτηση ωφελιμότητας για την σύνδεση r , τότε η εξίσωση (13) αναπαριστά την κοινωνική ευημερία του συστήματος.

Τα παραπάνω αποτελέσματα για την κοινωνική ευημερία του συστήματος μπορούν να γενικευτούν και στην περίπτωση που η σύνδεση r έχει συνάρτηση χρησιμότητας με γενική μορφή $U_r(x_r)$, αν η επιθυμία χρέωσης μεταβάλλεται ομαλά σύμφωνα με την εξίσωση:

$$w_r(t) = x_r(t) U'_r(x_r(t)). \quad (14)$$

Οι σχέσεις (10) και (11) μπορούν να προκύψουν, με φυσικό τρόπο σε επίπεδο πρωτοκόλλων μεταφοράς και πιο συγκεκριμένα να υλοποιηθούν από το πρωτόκολλο TCP. Ξεκινώντας την ανάλυση για το πώς αυτό μπορεί να γίνει εφικτό, πρέπει να σημειώσουμε το πολύ σημαντικό χαρακτηριστικό γνώρισμα του TCP, το ότι είναι δηλαδή ένας αυτο-συγχρονιζόμενος αλγόριθμος (ο αποστολέας χρησιμοποιεί μία επιβεβαίωση λήψης που του στέλνει ο παραλήπτης για να προχωρήσει ένα βήμα παραπέρα, οπότε αυτό δημιουργεί μία εξάρτηση “κλειδί” προς το χρόνο αποστολής-άφιξης T της σύνδεσης). Αναλυτικότερα, το TCP διαχειρίζεται, όπως είδαμε παραπάνω, ένα παράθυρο των μεταδομένων αλλά όχι επιβεβαιωμένων πακέτων το οποίο ονομάζεται παράθυρο συμφόρησης. Ο ρυθμός αποστολής x και το μέγεθος του παραθύρου $cwnd$ ικανοποιούν, κατά προσέγγιση, τη σχέση: $cwnd = xT$. Κάθε θετική

επιβεβαίωση λήψης αυξάνει το παράθυρο (όταν το TCP βρίσκεται σε φάση αποφυγής της συμφόρησης) κατά $1/cwnd$. Αντίθετα κάθε ειδοποίηση για συμφόρηση μειώνει το παράθυρο στο μισό.

Ας θεωρήσουμε, τώρα, μία παραλλαγή του μηχανισμού αυτού ώστε το παράθυρο να αυξάνεται κατά:

$$\bar{k} \left(\frac{\bar{w}}{cwnd} - f \right) \quad (15)$$

για κάθε επιβεβαίωση λήψης, όπου f είναι 1 ή 0 ανάλογα αν το πακέτο που επιβεβαιώθηκε μαρκαρίστηκε ή όχι αντίστοιχα. Από τη στιγμή που ο χρόνος μεταξύ των βημάτων ανανέωσης είναι περίπου $T/cwnd$, η αναμενόμενη αλλαγή στο ρυθμό αποστολής x ανά μονάδα χρόνου είναι κατά προσέγγιση:

$$\frac{\bar{k} \left(\frac{\bar{w}}{cwnd} - p \right)}{\frac{T}{cwnd}} = k(w - xp). \quad (16)$$

όπου $k = \bar{k}/T$, $w = \bar{w}/T$ και p είναι η πιθανότητα μαρκαρίσματος. Η έκφραση αυτή ανταποκρίνεται στη μορφή της γραμμικής αύξησης και πολλαπλασιαστικής μείωσης που περιγράφηκε από την εξίσωση (10), όπου η πιθανότητα να μαρκαριστεί ένα πακέτο κατά στην διαδρομή του προσεγγίζεται με το άθροισμα των πιθανοτήτων μαρκαρίσματος κάθε πόρου ξεχωριστά που ανήκει στη διαδρομή.

Ο παραπάνω αλγόριθμος τον οποίο θα ονομάζουμε WTP, αποτελεί μία μικρή παραλλαγή του μηχανισμού αποφυγής συμφόρησης του TCP, ο οποίος μπορεί να προσφέρει διαφοροποίηση υπηρεσιών, καθώς και αποδοτική και σταθερή λειτουργία του δικτύου αλλά και αντιμετώπιση της ραγδαίας ανάπτυξης του ώστε να καλύπτει τις αυξανόμενες ανάγκες. Βέβαια ο μηχανισμός αυτός προϋποθέτει ένα δίκτυο το οποίο παρέχει έναν απλό μηχανισμό ανάδρασης προς τους χρήστες. Ένας τέτοιος μηχανισμός είναι ο ECN ο οποίος σήμερα έχει ψηφιστεί από τον οργανισμό IETF ως πρότυπο.

2.4. Το πρόβλημα της σύγκλισης

Στη συνέχεια θα αναφερθούμε σε μία θεωρητική ανάλυση του αλγορίθμου WTP και θα μελετήσουμε το ρόλο που διαδραματίζει η παράμετρος \bar{k} όσον αφορά τη σύγκλιση και ισορροπία του WTP.

Ας υποθέσουμε ότι ο φόρτος που φτάνει σε έναν πόρο σε κάποια χρονική περίοδο τ είναι κατανομημένος βάσει μίας Γκαουσιανής κατανομής, με μέση τιμή $\chi\tau$ και διασπορά $\chi\sigma^2$ και ότι ένα πακέτο μαρκάρεται αν τη στιγμή που καταφτάνει ο φόρτος στον πόρο είναι μεγαλύτερος από μία μέγιστη ανεκτή τιμή B . Τότε βασιζόμενοι στην σταθερή κατανομή για μία ανακλαστική κίνηση Brown [14], θα ισχύει:

$$p(x) = \exp\left\{\frac{-2B(C-x)}{x\sigma^2}\right\} \quad (17)$$

Η συνθήκη που πρέπει να ισχύει για μη ταλαντευόμενη, αλλά σταθερή ισορροπία [2] είναι:

$$kT\left(1 + \frac{2BC}{x\sigma^2}\right)p(x) < \frac{1}{e} \quad (18)$$

Η αριστερή μεριά της συνθήκης (18) αυξάνει στο w (το οποίο στην κατάσταση ισορροπίας είναι ίσο με $\chi p(x)$) και έτσι η συνθήκη ικανοποιείται για κάθε $w < C$ αν

$$kT\left(1 + \frac{2B}{\sigma^2}\right) < \frac{1}{e} \quad (19)$$

Θα πρέπει να σημειώσουμε εδώ ότι καθώς η μέγιστη ανεκτή τιμή B αυξάνει, ή καθώς η διακύμανση της κίνησης σε επίπεδο πακέτων σ^2 μειώνεται, μεγαλώνουν οι πιθανότητες για μία συμπεριφορά με φαινόμενα καθυστέρησης και ταλαντώσεων [4].

Παράλληλα το μέγεθος του k , επιδρά και στη διακύμανση γύρω από το σημείο ισορροπίας, επηρεάζοντας το μέγεθος των ταλαντώσεων και το χρόνο σύγκλισης στο σημείο ισορροπίας [13]. Αναλυτικότερα, μικρές τιμές του k μειώνουν τις τυχαίες διακυμάνσεις των ρυθμών κίνησης στο σημείο ισορροπίας. Ενδεικτικά αναφέρουμε ότι για $C/\sigma^2=10$ η συνθήκη (19) ικανοποιείται για $k \approx 0.0018$. Από την άλλη μεριά, πρέπει να τονίσουμε ότι μεγαλύτερες τιμές του k οδηγούν γρηγορότερα τις συνδέσεις σε ισορροπία.

Στηριζόμενοι λοιπόν στην παραπάνω ανάλυση διαπιστώνουμε ότι τον WTP θα πρέπει να τον συνοδεύει ένας μηχανισμός που οδηγεί στη γρήγορη σύγκλιση των συνδέσεων (όπως στο TCP χρησιμοποιείται ο αλγόριθμος Slow start). Βέβαια θα πρέπει να τονίσουμε στο σημείο αυτό, ότι σε συνδέσεις μεγάλης χρονικής διάρκειας η χρήση μόνο του WTP μπορεί να μην είναι απαγορευτική για τη σταθερότητα του συστήματος, στο βαθμό που ο μηχανισμός αυτός κατάφερνε να συγκλίνει στο σημείο ισορροπίας. Σε περιπτώσεις όμως σύντομων συνδέσεων (π.χ. συνδέσεις τύπου HTTP) η ύπαρξη ενός μηχανισμού όπως το Slow start είναι απαραίτητη.

Από την άλλη μεριά, όπως είδαμε παραπάνω, οι συνδέσεις που χρησιμοποιούν τον αλγόριθμο Slow start στο TCP, πετυχαίνουν γρήγορη σύγκλιση αλλά δεν

τυγχάνουν καμίας διαφοροποίησης, αφού ο μηχανισμός αυτός “κατασκευάστηκε” βάσει της λογικής ότι όλες οι συνδέσεις θα πρέπει να απολαμβάνουν ίση μεταχείριση από το δίκτυο. Στην περίπτωση όμως που επιθυμούμε διαφοροποίηση υπηρεσιών στο Internet, με τη χρήση αλγορίθμων όπως ο WTP, θα πρέπει να γενικεύσουμε τη διαφοροποίηση αυτή και στους μηχανισμούς γρήγορης σύγκλισης των συνδέσεων. Αυτό είναι αναγκαίο περισσότερο στις σύντομες συνδέσεις όπου η επίδραση μηχανισμών γρήγορης σύγκλισης διαρκεί σε μεγαλύτερο ποσοστό του συνολικού χρόνου μετάδοσης.

Ωθούμενοι από την παραπάνω ανάγκη προτείναμε δύο μηχανισμούς για γρήγορη σύγκλιση, τους οποίους ονομάσαμε weighted Slow start και Large K και τους οποίους περιγράφουμε στη συνέχεια. Παράλληλα αναφέρουμε και τους ήδη υπάρχοντες μηχανισμούς για γρήγορη σύγκλιση, Slow start του TCP και Slow start του MulTCP, τους οποίους χρησιμοποιήσαμε στα πειράματά μας για να τους συγκρίνουμε με τους δύο παραπάνω αλγόριθμους.

2.4.1. MulTCP Slow start

Στη περίπτωση του μηχανισμού Slow start μία σύνδεση TCP ανοίγει το παράθυρο συμφόρησης εκθετικά, στέλνοντας δύο πακέτα για κάθε επιβεβαίωση λήψης που λαμβάνει. Είναι ενδιαφέρον το γεγονός, ότι N συνδέσεις TCP οι οποίες βρίσκονται σε φάση Slow start αποστέλλουν μόνο δύο πακέτα για κάθε επιβεβαίωση λήψης που λαμβάνουν. Παρόλα αυτά, N συνδέσεις TCP θα ξεκινούσαν στέλνοντας N πακέτα, με αποτέλεσμα να ληφθούν N επιβεβαιώσεις λήψης και $2N$ πακέτα να σταλούν ύστερα από ένα RTT. Την ίδια συμπεριφορά θα πρέπει να έχει και ο MulTCP, δηλαδή να στείλει N πακέτα στην αρχή και στη συνέχεια δύο πακέτα για κάθε επιβεβαίωση λήψης που φτάνει από το δίκτυο. Κάτι τέτοιο όμως, οδηγεί σε πολύ εκρηκτικές καταστάσεις αν το N είναι μεγάλο. Η εκρηκτικότητα αυτή μπορεί να οδηγήσει σε εκρηκτικούς ρυθμούς απωλειών πακέτων εμποδίζοντας με αυτό το τρόπο την σύνδεση να φτάσει στη κατάσταση ισορροπίας γρήγορα. Για το λόγο αυτό ο MulTCP χρησιμοποιεί έναν ομαλότερο μηχανισμό. Ξεκινά όπως ένα κανονικό TCP, στέλνοντας ένα πακέτο. Μετά από αυτό, στέλνει τρία πακέτα για κάθε επιβεβαίωση λήψης που λαμβάνεται, μέχρι να έχει ανοίξει το παράθυρο συμφόρησης όσο θα το είχαν ανοίξει N συνδέσεις TCP.

Μετά από k RTT, N συνδέσεις TCP θα έχουν παράθυρο συμφόρησης ίσο με $N2^k$. Ένα MulTCP στέλνοντας τρία πακέτα για κάθε επιβεβαίωση λήψης θα είχε παράθυρο συμφόρησης ίσο με 3^k . Έτσι θα είχαν την ίδια τιμή παραθύρου συμφόρησης έπειτα από k_N RTT όπου:

$$k_N = \frac{\log N}{\log 3 - \log 2} \quad (20)$$

το οποίο συμβαίνει όταν το παράθυρο συμφόρησης είναι:

$$w_N = 3^{k_N} \quad (21)$$

2.4.2. Weighted Slow start

Η ύπαρξη του μηχανισμού αυτού είναι αποτέλεσμα της προσπάθειάς μας να μπορέσουμε να υποστηρίξουμε διαφοροποίηση χρηστών στο μηχανισμό Slow start του TCP. Ο μηχανισμός αυτός λειτουργεί ως εξής:

- Για κάθε σύνδεση που ξεκινά, το παράθυρο συμφόρησης αυξάνει κατά ένα για κάθε θετική επιβεβαίωση λήψης, όπως και στο Slow start.
- Όταν ληφθεί μαρκαρισμένο πακέτο, έστω ότι το παράθυρο συμφόρησης έχει τιμή $cwnd_{ECN}$. Τότε η σύνδεση συνεχίζει να αυξάνει το παράθυρο συμφόρησης κατά ένα για κάθε επιβεβαίωση λήψης που λαμβάνει (μαρκαρισμένη ή μη), μέχρις ότου το παράθυρο συμφόρησης πάρει τιμή ίση με $\bar{w} * cwnd_{ECN}$.
- Στην περίπτωση αυτή παύει η λειτουργία του Weighted Slow start και ξεκινάει ο μηχανισμός WTP.

Η λειτουργία αυτού του μηχανισμού στηρίζεται στη λογική ότι οι χρήστες με μεγαλύτερες επιθυμίες χρέωσης, θα πρέπει να πάρουν κατά μέσο όρο μεγαλύτερο εύρος ζώνης. Δεδομένου ότι μέχρι τη στιγμή της συμφόρησης θα πρέπει να έχουν όλες οι συνδέσεις περίπου το ίδιο εύρος ζώνης, οι συνδέσεις με μεγαλύτερη επιθυμία χρέωσης δε θα πρέπει να σταματήσουν τη γρήγορη αύξηση του ρυθμού τους παρά μόνο όταν θα έχουν φτάσει κοντά στο σημείο σύγκλισης.

2.4.3. Large K

Στην ενότητα 2.5, αναλύσαμε ότι οι τιμές του \bar{k} θα πρέπει να είναι μικρές ώστε να εξασφαλίζεται η σταθερή λειτουργία των συνδέσεων στο σημείο ισορροπίας. Παρόλα αυτά, μικρές τιμές του \bar{k} οδηγούν στην αργή σύγκλιση των συνδέσεων, γεγονός που οδήγησε και στη δημιουργία του προβλήματος που συζητάμε. Από την άλλη μεριά, παρατηρήσαμε στα πειράματα που εκτελέσαμε στη διάρκεια της έρευνας του πρώτου μέρους της μεταπτυχιακής μας εργασίας, ότι μεγάλες τιμές του \bar{k} οδηγούν σε χρόνους σύγκλισης παρόμοιους με αυτούς του Slow start. Έτσι καταλήξαμε στο συμπέρασμα ότι αν το \bar{k} παίρνει μεγάλες τιμές κατά τη φάση γρήγορης σύγκλισης και μικρές τιμές κατά τη φάση αποφυγής της συμφόρησης μπορούμε να επιτύχουμε τόσο γρήγορη σύγκλιση όσο και ευσταθή λειτουργία. Πιο συγκεκριμένα τα βήματα του μηχανισμού είναι τα εξής:

- Κατά την έναρξη των συνδέσεων το παράθυρο συμφόρησης αυξάνεται κατά $k_s \frac{\bar{w}}{cwnd}$ για κάθε θετική επιβεβαίωση λήψης, όπου $cwnd$ η τρέχουσα τιμή του παραθύρου και k_s μία τιμή του \bar{k} που οδηγεί σε γρήγορη σύγκλιση του WTP.
- Όταν ληφθεί μαρκαρισμένο πακέτο, τότε ο αλγόριθμος WTP λειτουργεί όπως τον περιγράψαμε στην ενότητα 2.3, αλλά με τιμές του \bar{k} για τις οποίες ισχύει η σχέση (19) ώστε να υπάρξει εγγύηση σταθερής κατάστασης στο σημείο ισορροπίας.

2.4.4. TCP Slow start

Ο μηχανισμός αυτός χρησιμοποιείται από το TCP για τη γρήγορη σύγκλιση των συνδέσεων στο σημείο ισορροπίας.

Μπορούμε να πούμε σε ένα γενικό βαθμό ότι ο αποστολέας σε μία σύνδεση TCP, χρησιμοποιεί τις επιβεβαιώσεις λήψης (acknowledgements), σαν ένα “ρολόι” για την αποστολή των πακέτων στο δίκτυο. Στο βαθμό που ο παραλήπτης μπορεί να δημιουργεί επιβεβαιώσεις λήψης όχι γρηγορότερα από τη στιγμή που παραλαμβάνει πακέτα από το δίκτυο, διαπιστώνουμε ότι το πρωτόκολλο TCP είναι αυτο-συγχρονιζόμενο (self-clocking). Τα αυτο-συγχρονιζόμενα συστήματα προσαρμόζονται αυτόματα στις διακυμάνσεις εύρους ζώνης που παρουσιάζονται στο δίκτυο. Μάλιστα η προσαρμογή αυτή λειτουργεί για ένα μεγάλο εύρος καθυστερήσεων και για μεγάλο σύνολο εύρων ζώνης. Για παράδειγμα το TCP λειτουργεί αποδοτικά σε δίκτυα καναλιών Cray (Cray channels) των 800 Mbps μέχρι και σε ράδιο-συνδέσμους πακέτων (radio packet links) των 1200 bps. Από την άλλη μεριά, ο ίδιος λόγος που κάνει έναν αυτό-συγχρονιζόμενο μηχανισμό να είναι σταθερός, τον δυσκολεύει να ξεκινήσει (για να έχουμε ροή δεδομένων θα πρέπει να υπάρχουν επιβεβαιώσεις λήψης για να συγχρονίζονται τα πακέτα αλλά για να υπάρξουν επιβεβαιώσεις λήψης θα πρέπει να υπάρχει ροή δεδομένων).

Για να ξεκινήσει λοιπόν το “ρολόι”, ο Jacobson [9], πρότεινε ένα μηχανισμό που ονομάζεται Slow start, ο οποίος αυξάνει σταδιακά τα δεδομένα που πρόκειται να μεταδοθούν. Η εφαρμογή του μηχανισμού αυτού στο πρωτόκολλο TCP, απαιτεί τις εξής τρεις ενέργειες:

- Για κάθε σύνδεση, διατηρείται μία μεταβλητή που ονομάζεται παράθυρο συμφόρησης (congestion window) και δηλώνει πόσα πακέτα μπορούν να σταλούν στο δίκτυο.

- Όταν ξεκινά η αποστολή ή υπάρξει αναμετάδοση των δεδομένων ύστερα από μία απώλεια πακέτου, το παράθυρο συμφόρησης παίρνει τιμή ίση με ένα πακέτο.
- Για κάθε επιβεβαίωση λήψης νέων δεδομένων, η τιμή του παραθύρου συμφόρησης αυξάνεται κατά ένα πακέτο.
- Ο αριθμός των πακέτων που πρόκειται να αποσταλούν κάθε φορά, είναι το ελάχιστο του διαφημιζόμενου παραθύρου (advertised window) του παραλήπτη και του παραθύρου συμφόρησης.

Στην πραγματικότητα, στο μηχανισμό Slow start η αύξηση του παραθύρου συμφόρησης δεν είναι και τόσο αργή: παίρνει χρόνο ίσο με $R \log_2 W$, όπου R είναι ο χρόνος αποστολής-άφιξης (round trip time – RTT) και W το μέγεθος του παραθύρου συμφόρησης μετρούμενο σε πακέτα. Αυτό σημαίνει ότι το παράθυρο συμφόρησης ανοίγει αρκετά γρήγορα ώστε να παρουσιάζεται μία αμελητέα επίδραση στην απόδοση, ακόμα και για συνδέσεις με μεγάλο γινόμενο εύρους ζώνης και καθυστέρησης (bandwidth-delay product). Παράλληλα ο μηχανισμός αυτός εγγυάται ότι η σύνδεση θα παράγει δεδομένα με ρυθμό το πολύ διπλάσιο από ότι επιτρέπει το μονοπάτι της διαδρομής των πακέτων.

2.5. Μηχανισμοί μαρκαρίσματος

Σε αυτή την ενότητα θα περιγράψουμε τους μηχανισμούς μαρκαρίσματος τους οποίους χρησιμοποιήσαμε για τη διεξαγωγή των πειραμάτων μας.

2.5.1. Virtual Queue

Ο μηχανισμός Virtual Queue (VQ) διαχειρίζεται ένα ιδεατό ενταμιευτή (virtual buffer - VB) ο οποίος έχει μέγεθος θB και εξυπηρετείται με ρυθμό θC , όπου B, C είναι ο πραγματικός ενταμιευτής και η πραγματική χωρητικότητα του συνδέσμου εξόδου, αντίστοιχα. Ας σημειώσουμε εδώ, ότι το B δεν είναι απαραίτητα το συνολικό μέγεθος του συνδέσμου εξόδου, αλλά μπορεί να είναι μία τιμή η οποία ανταποκρίνεται στη μέγιστη ανεκτή καθυστέρηση. Ο αλγόριθμος μαρκάρει όλα τα πακέτα που φτάνουν στο σύνδεσμο από τη στιγμή που συμβαίνει μία απώλεια πακέτου στον ιδεατό ενταμιευτή μέχρι τη στιγμή που ο ιδεατός ενταμιευτής αδειάσει. Η περίοδος αυτή ονομάζεται περίοδος αιχμής (busy period) του ιδεατού ενταμιευτή.

Όπως θα δούμε στην ενότητα των πειραματικών αποτελεσμάτων ένα χαρακτηριστικό του αλγορίθμου αυτού είναι το γεγονός ότι διαφοροποιεί τις ροές βάσει της εκρηκτικότητάς τους. Ένα άλλο χαρακτηριστικό είναι, ότι δεν αποφεύγει περιπτώσεις συγχρονισμού φάσης σε αλγόριθμους αποφυγής της συμφόρησης κλειστού βρόγχου, όπως παρόμοια έχει παρατηρηθεί και στους δρομολογητές απόρριψης – ουράς (drop tail routers). Τέτοιες καταστάσεις οδηγούν ορισμένες

συνδέσεις να αποκτούν μεγάλη διαπερατότητα και άλλες να περιορίζονται σε πολύ μικρές. Πραγματικά, ένα κίνητρο εμφάνισης δρομολογητών τύπου RED, ήταν τέτοιες καταστάσεις.

2.5.2. Random Early Detect

Στο μηχανισμό Random Early Detect (RED) [6], η πιθανότητα μαρκαρίσματος είναι μία τμηματική γραμμική συνάρτηση του μέσου μεγέθους της ουράς \bar{q} , το οποίο υπολογίζεται χρησιμοποιώντας εκθετικό υπολογισμό μέσου όρου (exponential averaging) με παράγοντα βάρους w_q . Πιο συγκεκριμένα, αν το \bar{q} είναι μικρότερο από κάποιο ελάχιστο μέγεθος της ουράς \min_{thr} τότε η πιθανότητα μαρκαρίσματος είναι μηδέν. Αν το \bar{q} βρίσκεται μεταξύ του \min_{thr} και κάποιου μέγιστου μεγέθους της ουράς \max_{thr} τότε η πιθανότητα μαρκαρίσματος κυμαίνεται γραμμικά από μηδέν μέχρι κάποια μέγιστη τιμή \max_p . Τέλος, αν το \bar{q} πάρει τιμές μεγαλύτερες του \max_{thr} τότε τα πακέτα απορρίπτονται. Η έκδοση “gentle_” του αλγόριθμου RED προτείνει η πιθανότητα μαρκαρίσματος να κυμαίνεται μεταξύ \max_p και μονάδας όταν το \bar{q} παίρνει τιμές μεταξύ \max_{thr} και $2\max_{thr}$.

Για τη διεξαγωγή των πειραμάτων μας χρησιμοποιήσαμε την παραπάνω έκδοση του αλγόριθμου RED την οποία τροποποιήσαμε ώστε τα πακέτα να μην πετάγονται ποτέ αλλά μόνο να μαρκάρονται.

2.5.3. Load Based Marking

Στον αλγόριθμο Load Based Marking (LBM), η πιθανότητα μαρκαρίσματος είναι μία τμηματική γραμμική συνάρτηση του μέσου φόρτου (utilization), ο οποίος υπολογίζεται ανά κάποια συγκεκριμένη χρονική περίοδο. Στο σημείο αυτό να υπενθυμίσουμε ότι στην περίπτωση του RED η πιθανότητα μαρκαρίσματος είναι μία γραμμική συνάρτηση του μέσου μεγέθους της ουράς. Στον αλγόριθμο LBM, η πιθανότητα μαρκαρίσματος είναι μηδέν όταν ο μέσος φόρτος είναι μικρότερος από κάποια ελάχιστη τιμή p_0 . Για τιμές του φόρτου p , μεγαλύτερες της p_0 , η πιθανότητα μαρκαρίσματος δίνεται από τη σχέση $\alpha(p-p_0)$. Έτσι, οι παράμετροι του αλγόριθμου είναι τρεις:

- η ελάχιστη τιμή φόρτου p_0
- η παράμετρος α , η οποία προσδιορίζει την κλίση της συνάρτησης για την πιθανότητα μαρκαρίσματος
- η περίοδος υπολογισμού του μέσου φόρτου t_{avg} .

Η περίοδος υπολογισμού του μέσου φόρτου καθορίζει πόσο γρήγορα ο αλγόριθμος προσαρμόζει την πιθανότητα μαρκαρίσματος στις αλλαγές του φόρτου.

Επιπλέον, η περίοδος αυτή καθορίζει τις χρονικές κλίμακες στις οποίες η συμμόρφωση γίνεται αντιληπτή, επομένως καθορίζει τις χρονικές κλίμακες στις οποίες η εκρηκτικότητα της κίνησης επηρεάζει την πιθανότητα μαρκαρίσματος.

3. Υλοποίηση

Όπως αναφέραμε και στο Κεφάλαιο 1 η μελέτη των αλγόριθμων που περιγράψαμε στο Κεφάλαιο 2, έγινε τόσο σε επίπεδο προσομοίωσης όσο και σε επίπεδο πραγματικής υλοποίησης. Στις επόμενες ενότητες περιγράφουμε τις υλοποιήσεις αυτές.

3.1. Υλοποίηση σε προσομοιωτή

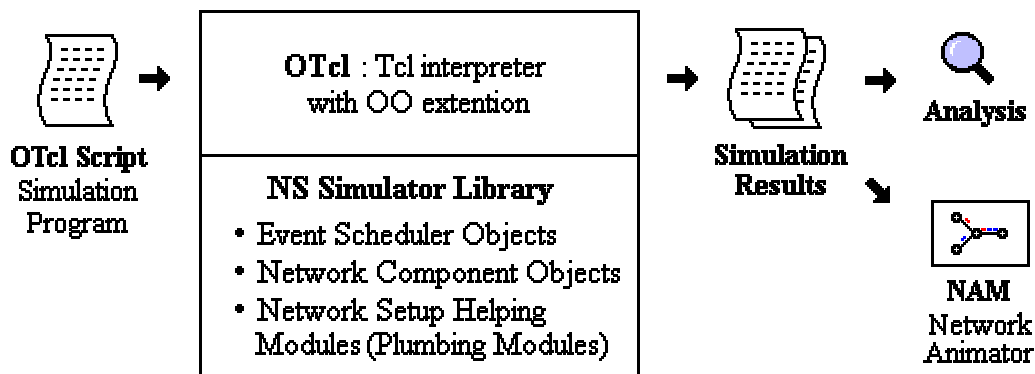
Σε αυτή την ενότητα περιγράφουμε τον προσομοιωτή Network Simulator 2, τον οποίο χρησιμοποιήσαμε για την πραγματοποίηση των πειραμάτων προσομοίωσης που εκτελέσαμε κατά τη διάρκεια της έρευνάς μας. Επίσης, περιγράφουμε, τις επεκτάσεις που επιτελέσαμε στον προσομοιωτή NS2, για την υλοποίηση των αλγόριθμων που περιγράψαμε στο Κεφάλαιο 2.

3.1.1. Ο προσομοιωτής δικτύου “Network Simulator 2”

Ο Network Simulator 2 (NS2), είναι ένας προσομοιωτής, η ροή του οποίου καθοδηγείται από γεγονότα (event driven simulator) και ο οποίος προσομοιώνει μία ποικιλία από δίκτυα IP. Υλοποιεί πρωτόκολλα δικτύων όπως το TCP και το UDP, συμπεριφορές πηγών κίνησης, όπως FTP, Telnet, Web, CBR και VBR, μηχανισμούς διαχείρισης ουρών σε δρομολογητές, όπως Drop Tail, RED και CBQ, αλγόριθμους δρομολόγησης όπως Dijkstra και πολλά άλλα. Ο NS2 υλοποιεί, επίσης, μηχανισμούς multicasting και μερικά πρωτόκολλα επιπέδου MAC για προσομοιώσεις δικτύων LAN. Ο προσομοιωτής αυτός, δημιουργήθηκε στο πανεπιστήμιο του Berkeley και αποτελεί σήμερα μέρος του έργου VINT [18], το οποίο αναπτύσσει εργαλεία για αναπαράσταση και ανάλυση αποτελεσμάτων προσομοίωσης, καθώς και εργαλεία τα οποία μετατρέπουν τοπολογίες δικτύων γνωστών εφαρμογών σε τοπολογίες, οι οποίες είναι αναγνώσιμες από τον NS2.

Πιο συγκεκριμένα, όπως φαίνεται στο Σχήμα 2, ο προσομοιωτής NS2 είναι ένας μεταφραστής σεναρίων, τα οποία είναι γραμμένα στη γλώσσα Object Tcl² (OTcl). Ο προσομοιωτής αυτός περιέχει ένα χρονοπρογραμματιστή γεγονότων προσομοίωσης, αντικειμενοστραφείς βιβλιοθήκες που υλοποιούν στοιχεία ενός δικτύου, καθώς και βιβλιοθήκες μονάδων οργάνωσης δικτύων (network setup modules). Στη πραγματικότητα οι μονάδες αυτές υλοποιούνται ως συναρτήσεις – μέλη (member functions) του βασικού αντικειμένου προσομοίωσης. Με άλλα λόγια για να χρησιμοποιήσει κάποιος τον NS2, θα πρέπει να προγραμματίσει σε γλώσσα OTcl.

² Η γλώσσα προγραμματισμού OTcl, αποτελεί μία αντικειμενοστραφή επέκταση της γνωστής γλώσσας σεναρίων Tcl, η οποία αναπτύχθηκε στο πανεπιστήμιο MIT.



Σχήμα 2 Απλοποιημένη άποψη του NS2

Αναλυτικότερα, για να δημιουργήσει και να τρέξει κάποιος ένα προσομοιωμένο δίκτυο, θα πρέπει να γράψει ένα σενάριο OTcl το οποίο:

- αρχικοποιεί ένα χρονοπρογραμματιστή γεγονότων
- οργανώνει την τοπολογία δικτύου, συνενώνοντας τα αντικείμενα που υλοποιούν τα μέρη του δικτύου (π.χ. δρομολογητές, πηγές κίνησης κτλ.)
- προγραμματίζει τις χρονικές στιγμές έναρξης και τερματισμού της μετάδοσης των πακέτων από τις γεννήτριες κίνησης.

Ο όρος “συνένωση” (plumbing) αντικειμένων χρησιμοποιείται, στο NS2, κατά την οργάνωση ενός δικτύου. Πιο συγκεκριμένα, η οργάνωση ενός δικτύου προϋποθέτει τη συνένωση των πιθανών μονοπατιών δεδομένων με γειτονικά αντικείμενα δικτύου, θέτοντας το “γειτονικό” δείκτη του αντικειμένου που πρέπει να συνενωθεί στη διεύθυνση του κατάλληλου αντικειμένου. Όποτε κάποιος χρήστης επιθυμεί να δημιουργήσει ένα νέο αντικείμενο δικτύου (π.χ. μία επέκταση του πρωτοκόλλου TCP), μπορεί εύκολα να το δημιουργήσει είτε γράφοντας ένα νέο αντικείμενο είτε δημιουργώντας ένα αντικείμενο το οποίο θα είναι συνδυασμός από αντικείμενα της βιβλιοθήκης του προσομοιωτή και να συνενώσει το μονοπάτι δεδομένων μέσω αυτού του αντικειμένου. Αυτό μπορεί να ακούγεται ως μία πολυσύνθετη δουλειά, αλλά το γεγονός ότι χρησιμοποιείται η OTcl απλοποιεί πολύ τα πράγματα. Το μεγάλο εύρος δυνατοτήτων του NS2 οφείλεται σε αυτό το είδος “συνένωσης” που περιγράψαμε.

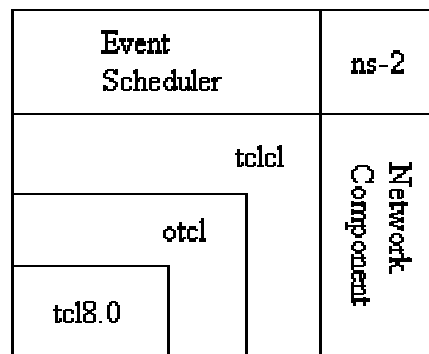
Ένα άλλο κύριο τμήμα του NS2, εκτός των αντικειμένων δικτύου, είναι ο χρονοπρογραμματιστής γεγονότων. Ένα γεγονός στον NS2 είναι ένας αναγνωριστικός αριθμός πακέτου (packet ID), ο οποίος είναι μοναδικός για κάποιο πακέτο με προγραμματισμένο χρόνο και αποτελεί δείκτη σε ένα αντικείμενο το οποίο

διαχειρίζεται το γεγονός αυτό. Στον NS2, ένας χρονοπρογραμματιστής γεγονότων διατηρεί ίχνη του χρόνου προσομοίωσης και πυροδοτεί τα γεγονότα που βρίσκονται στην ουρά γεγονότων και έχουν προγραμματιστεί να εκτελεστούν την τρέχουσα χρονική στιγμή, ενεργοποιώντας τα κατάλληλα τμήματα του δικτύου προσομοίωσης, τα οποία είναι συνήθως εκείνα που δημιούργησαν τα γεγονότα. Στη συνέχεια, τα τμήματα αυτά εκτελούν τις ενέργειες που αφορούν στο πακέτο, το οποίο αντιστοιχεί στο γεγονός που έχει ενεργοποιηθεί. Τα τμήματα του δικτύου επικοινωνούν μεταξύ τους μέσω πακέτων, χωρίς όμως οι ενέργειες αυτές, στην πραγματικότητα, να καταναλώνουν χρόνο προσομοίωσης. Όλα τα τμήματα δικτύου, τα οποία χρειάζονται να ξοδέψουν κάποιο ποσό του προσομοιωμένου χρόνου, για ένα πακέτο (π.χ. καθυστέρηση), χρησιμοποιούν το χρονοπρογραμματιστή δικτύου, θέτοντας ένα γεγονός για το πακέτο και περιμένοντας το γεγονός αυτό να πυροδοτηθεί πριν εκτελέσουν περαιτέρω ενέργειες που αφορούν στο πακέτο. Για παράδειγμα, ένα αντικείμενο μεταγωγέα δικτύου που προσομοιώνει ένα μεταγωγέα με 20 χιλιοστά του δευτερολέπτου καθυστέρηση λόγω θεμάτων μεταγωγής, θέτει ένα γεγονός για κάθε πακέτο που θα δρομολογηθεί στο χρονοπρογραμματιστή, δηλώνοντας ότι αυτό το γεγονός θα πυροδοτηθεί 20 χιλιοστά του δευτερολέπτου αργότερα. Ο χρονοπρογραμματιστής, έπειτα από 20 χιλιοστά του δευτερολέπτου βγάζει από την ουρά γεγονότων το γεγονός και το πυροδοτεί ενεργοποιώντας το αντικείμενο που προσομοιώνει τον μεταγωγέα, το οποίο με τη σειρά του προωθεί το πακέτο στον κατάλληλο σύνδεσμο εξόδου. Μία άλλη χρήση του χρονοπρογραμματιστή γίνεται από τους χρονομέτρους (timers). Για παράδειγμα, το TCP χρειάζεται ένα χρονομέτρη ώστε να διατηρεί ίχνη του χρόνου μετάδοσης ενός πακέτου και να αποφασίζει ποια πακέτα πρέπει να αναμεταδοθούν. Οι χρονομέτρους χρησιμοποιούν τους χρονοπρογραμματιστές γεγονότων με τον ίδιο τρόπο που χρησιμοποιούνται για την υλοποίηση των καθυστερήσεων. Η μόνη διαφορά είναι ότι οι χρονομέτρους μετρούν μία τιμή του χρόνου, η οποία σχετίζεται με ένα πακέτο και επιτελεί μία συγκεκριμένη ενέργεια στο πακέτο, έπειτα από το πέρασμα κάποιου χρονικού διαστήματος και δεν προσομοιώνουν καθυστέρηση.

Ο NS2 δεν είναι γραμμένος μόνο σε OTcl, αλλά επίσης σε C++. Για λόγους απόδοσης, ο NS2 διαχωρίζει την υλοποίηση των μονοπατιών δεδομένων (data paths) από την υλοποίηση των μονοπατιών ελέγχου (control paths). Σε μία προσπάθεια μείωσης του χρόνου επεξεργασίας πακέτων και γεγονότων (όχι του χρόνου προσομοίωσης³), ο χρονοπρογραμματιστής γεγονότων και τα αντικείμενα που προσομοιώνουν τα βασικά τμήματα δικτύου στο μονοπάτι δεδομένων έχουν γραφθεί και μεταγλωττιστεί σε C++. Τα μεταγλωττισμένα αντικείμενα είναι προσβάσιμα στο

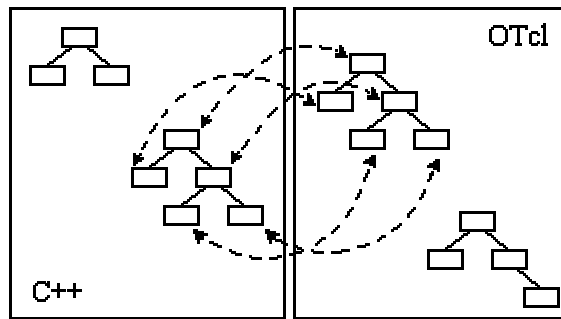
³ Ως χρόνο προσομοίωσης εννοούμε, το χρόνο που συμβαίνει ένα γεγονός κατά τη διάρκεια της προσομοίωσης (π.χ. καθυστέρηση ενός πακέτου), τον οποίο διαχειρίζεται ο χρονοπρογραμματιστής του NS2 και βάσει του οποίου πραγματοποιούνται οι μετρήσεις.

μεταφραστή OTcl μέσω μίας συνδετικής διάταξης που δημιουργεί αντιστοιχία των αντικειμένων OTcl σε κάθε αντικείμενο C++ και μετατρέπει τις συναρτήσεις ελέγχου και τις προσβάσιμες μεταβλητές ενός αντικειμένου C++ σε συναρτήσεις – μέλη (member functions) και μεταβλητές – μέλη (member variables) του αντίστοιχου αντικειμένου OTcl. Με τον τρόπο αυτό ο έλεγχος των αντικειμένων σε C++ δίνεται στα αντικείμενα της OTcl. Καθίσταται, επίσης, εφικτή και η πρόσθεση νέων μελών συναρτήσεων και μεταβλητών σε ένα αντικείμενο C++ το οποίο έχει αντιστοιχηθεί σε ένα αντικείμενο OTcl, με τη χρήση της OTcl. Τα αντικείμενα C++ τα οποία δεν απαιτείται να ελέγχονται στη προσομοίωση ή χρησιμοποιούνται εσωτερικά από κάποιο άλλο αντικείμενο δε χρειάζεται να αντιστοιχηθούν σε αντικείμενα OTcl. Ταυτόχρονα, ένα αντικείμενο (το οποίο δεν ανήκει απαραίτητα στο μονοπάτι δεδομένων) μπορεί να υλοποιηθεί εξολοκλήρου σε OTcl. Το Σχήμα 3, αναπαριστά ένα παράδειγμα ιεραρχίας αντικειμένων γραμμένα σε C++ και OTcl.



Σχήμα 3 Αρχιτεκτονική του NS2

Το Σχήμα 4 αναπαριστά τη γενική αρχιτεκτονική του NS2. Στο σχήμα αυτό ένας γενικός χρήστης (όχι ένας προγραμματιστής του NS2) μπορεί να αρκестεί μόνο στην αριστερή πάνω γωνία, σχεδιάζοντας και τρέχοντας προσομοιώσεις σε Tcl, χρησιμοποιώντας τα αντικείμενα προσομοίωσης που περιέχονται στη βιβλιοθήκη OTcl. Αντίθετα ένας προγραμματιστής του NS2, θα πρέπει να γνωρίζει το μεγαλύτερο μέρος της ιεραρχίας του Σχήματος 4, ώστε να μπορεί να προσθέτει τα νέα αντικείμενα δικτύου που επιθυμεί, στο κατάλληλο σημείο τις ιεραρχίας αντικειμένων του NS2. Ο χρονοπρογραμματιστής και τα περισσότερα τμήματα δικτύου έχουν υλοποιηθεί σε C++ και είναι διαθέσιμα στην OTcl μέσω της συνδετικής διάταξης που υλοποιείται χρησιμοποιώντας την tclcl. Όλα τα παραπάνω μαζί αποτελούν τον προσομοιωτή NS2, ο οποίος είναι ένας αντικειμενοστραφής μεταφραστής Tcl με βιβλιοθήκες προσομοίωσης δικτύου.



Σχήμα 4 Ιεραρχία αντικειμένων C++ και OTel

3.1.2. Υλοποίηση του WTP και αλγορίθμων για γρήγορη σύγκλιση

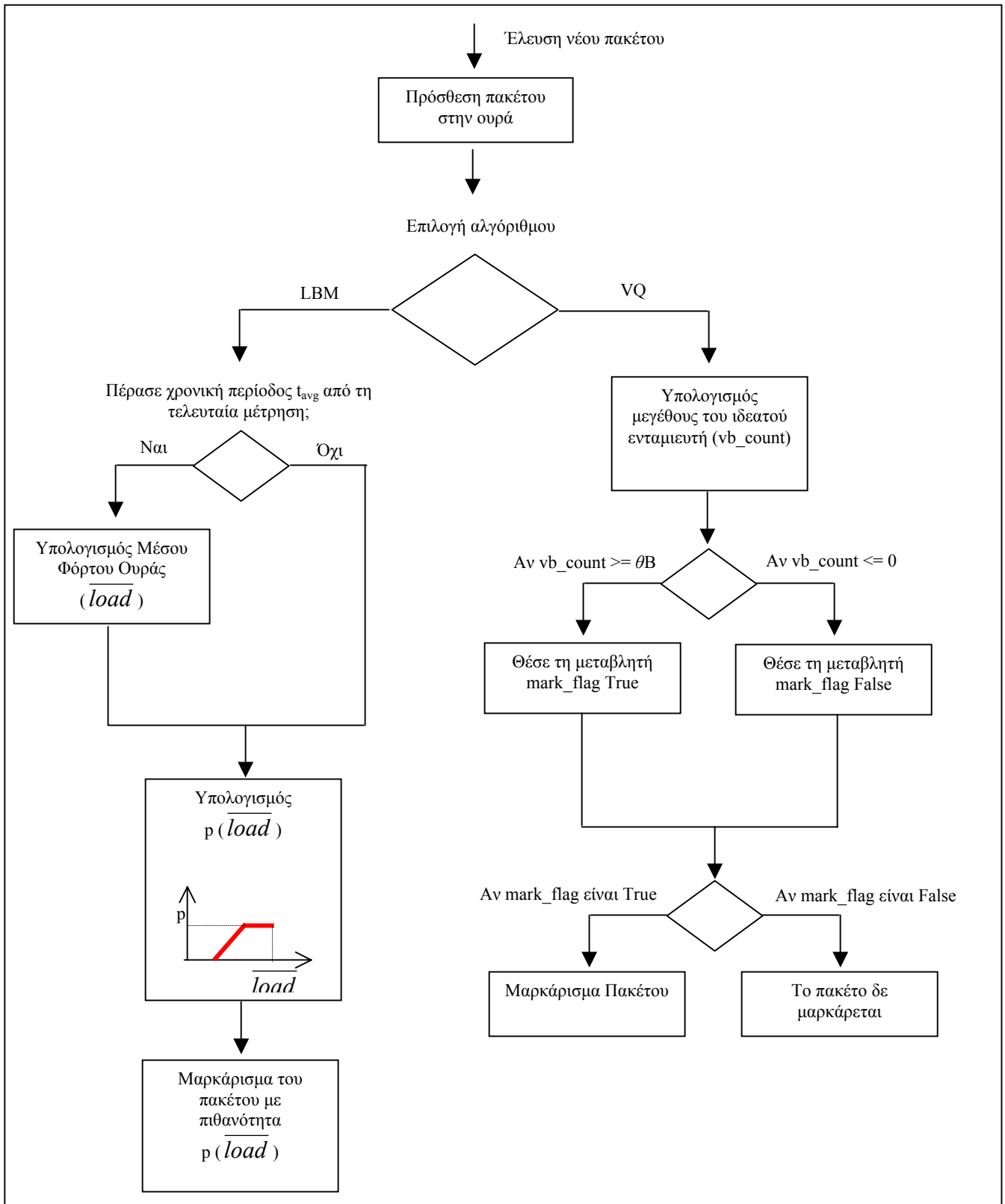
Όλα τα αντικείμενα που υλοποιούν τις διάφορες εκδόσεις του πρωτοκόλλου TCP (π.χ. Reno, Sack κτλ.) στο NS2, αποτελούν υποκλάσεις του αντικειμένου που υλοποιεί την Tahoe έκδοση όπως αυτή περιγράφεται στο [27] και το οποίο θα ονομάζουμε TCP/Tahoe. Ακολουθώντας λοιπόν και εμείς αυτή τη λογική, δημιουργήσαμε ένα αντικείμενο το οποίο υλοποιεί και διαχειρίζεται τον αλγόριθμο αποφυγής της συμφόρησης WTP και όλους τους αλγόριθμους για γρήγορη σύγκλιση που περιγράψαμε στο Κεφάλαιο 2 και προέρχεται από το αντικείμενο TCP/Tahoe.

Πιο συγκεκριμένα, το αντικείμενο αυτό κληρονομεί τις συναρτήσεις που υλοποιούνται στο αντικείμενο TCP/Tahoe τροποποιώντας εκείνες που είναι υπεύθυνες για τις αλλαγές του παραθύρου συμφόρησης ($cwnd$), έτσι ώστε να υλοποιούν τον αλγόριθμο WTP αλλά και τους αλγόριθμους που προτείναμε για γρήγορη σύγκλιση των συνδέσεων. Οι συναρτήσεις αυτές είναι:

- Η *opencwnd()* η οποία καλείται κάθε φορά που λαμβάνεται μία επιβεβαίωση λήψης νέων δεδομένων και φροντίζει την αύξηση του $cwnd$.
- Η *slowdown()* η οποία καλείται όταν έχει ληφθεί μαρκαρισμένη επιβεβαίωση λήψης, όπως αναφέρουμε παρακάτω, ή έχουν ληφθεί τρεις αντίγραφες επιβεβαιώσεις λήψης (duplicate acknowledgments) ή έχει λήξει ο χρονομέτρης αναμετάδοσης πακέτου (retransmit timer) και φροντίζει για τη μείωση του $cwnd$.
- Η *ecn()* η οποία καλείται κάθε φορά που λαμβάνεται μία μαρκαρισμένη λήψη και φροντίζει ώστε η μείωση του παραθύρου να γίνεται μία φορά ανά χρόνο αποστολής – άφιξης (RTT) όπως καθορίζεται στο [4].

Η λειτουργία του WTP καθορίζεται από τις παρακάτω μεταβλητές:

- $w_$: είναι μία μεταβλητή κινητής υποδιαστολής η οποία καθορίζει την τιμή της επιθυμίας χρέωσης ανά χρόνο αποστολής-άφιξης (RTT), \bar{w} .



Σχήμα 5 Διάγραμμα ροής της ουράς GKQ

- k_- : είναι μία μεταβλητή κινητής υποδιαστολής η οποία καθορίζει την τιμή του παράγοντα σύγκλισης ανά χρόνο αποστολής-άφιξης, \bar{k} .

- **slowstart_on_** : αν η μεταβλητή πάρει τιμή αληθή (true) τότε σε κάθε έναρξη μίας σύνδεσης χρησιμοποιείται ένας από τους αλγόριθμους που περιγράψαμε στην ενότητα 2.5, συμπεριλαμβανομένου επίσης και του Slow start του TCP όπως αυτός υλοποιείται στο TCP/Tahoe. Για την επιλογή ενός από τους αλγόριθμους αυτούς υπάρχουν οι μεταβλητές:
 - **slowstart_wtp_** : όταν η μεταβλητή αυτή είναι αληθής τότε χρησιμοποιούμε τον αλγόριθμο *Large K*, θέτοντας στη μεταβλητή **kss_** την τιμή που επιθυμούμε να χρησιμοποιήσουμε για να έχουμε γρήγορη σύγκλιση του WTP (υπενθυμίζουμε ότι κατά τον αλγόριθμο αυτό χρησιμοποιείται αποκλειστικά ο WTP, αλλά με διαφορετικές τιμές του \bar{k} πριν και μετά τη συμφόρηση).
 - **slowstart_multcp_** : όταν η μεταβλητή αυτή είναι αληθής τότε αρχικά χρησιμοποιείται ο μηχανισμός *MulTCP Slow Start* μέχρι τη στιγμή που αναγνωριστεί συμφόρηση (αντίγραφες επιβεβαιώσεις λήψης, λήξη χρονομέτρη αναμετάδοσης ή μαρκαρισμένη επιβεβαίωση λήψης) οπότε και χρησιμοποιείται ο μηχανισμός WTP.
 - **slowstart_w_** : όταν η μεταβλητή αυτή είναι αληθής, τότε αρχικά χρησιμοποιείται ο μηχανισμός *Weighted Slow start* μέχρι τη στιγμή αναγνώρισης συμφόρησης στο δίκτυο, οπότε και χρησιμοποιείται ο μηχανισμός WTP.
 - Τέλος αν καμία από τις παραπάνω μεταβλητές δεν είναι αληθής τότε χρησιμοποιείται ο μηχανισμός Slow start του TCP.

3.1.3. Υλοποίηση των αλγορίθμων μαρκαρίσματος

Για την υλοποίηση των αλγορίθμων μαρκαρίσματος Load Based Marking (LBM) και Virtual Queue (VQ), που περιγράψαμε στο Κεφάλαιο 2⁴, δημιουργήσαμε ένα αντικείμενο, το οποίο αποτελεί υποκλάση του αντικειμένου που υλοποιεί τη λειτουργικότητα μίας ουράς ενός συνδέσμου στο NS2. Στο διάγραμμα ροής του Σχήματος 5 φαίνονται οι ενέργειες που επιτελεί η ουρά κάθε φορά που φτάνει ένα πακέτο.

⁴ Για τον αλγόριθμο Random Early Detect (RED), υπάρχει ήδη υλοποίηση στο NS2. Θα πρέπει, όμως να αναφέρουμε ότι και στην περίπτωση του RED τροποποιήσαμε την λειτουργία του έτσι ώστε τα πακέτα να μαρκάρονται αντί να απορρίπτονται από την ουρά.

Αναλυτικότερα, κάθε φορά που ένα πακέτο φτάνει στην ουρά GKQ, αποθηκεύεται στον ενταμιευτή. Στη συνέχεια ανάλογα με τη τιμή μίας μεταβλητής **mark_method_**, επιλέγουμε τον αλγόριθμο που επιθυμούμε να χρησιμοποιήσουμε ως εξής:

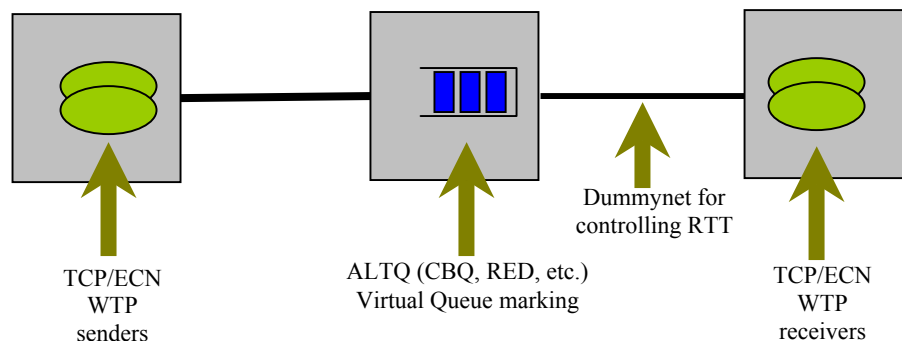
- Αν η τιμή της *mark_method_* είναι 0, ο αλγόριθμος μαρκαρίσματος που έχει επιλεγθεί είναι ο LBM. Στην περίπτωση αυτή:
 - Αν έχει περάσει χρονική διάρκεια ίση με t_{avg} (βλ. Ενότητα 2.4.3) από τον τελευταίο υπολογισμό του μέσου φόρτου της ουράς (η μεταβλητή \overline{load} στο Σχήμα 5), τότε υπολογίζεται η νέα τιμή αυτού.
 - Στη συνέχεια υπολογίζεται η πιθανότητα μαρκαρίσματος ως συνάρτηση του μέσου φόρτου της ουράς που έχει υπολογιστεί.
 - Για να αποφασιστεί αν βάσει της πιθανότητας $p(\overline{load})$ (βλ. Σχήμα 5) που υπολογίσαμε πρέπει το πακέτο να μαρκαριστεί ή όχι, επιλέγουμε μία τυχαία μεταβλητή χ κανονικής κατανομής και ελέγχουμε αν $\chi < p$. Αν ισχύει η ανισότητα τότε το πακέτο μαρκάρεται, αλλιώς το πακέτο δε μαρκάρεται.
- Αν η τιμή της μεταβλητής *mark_method_* είναι 1, ο αλγόριθμος μαρκαρίσματος που έχει επιλεγθεί είναι ο VQ. Στην περίπτωση αυτή:
 - Υπολογίζουμε το μέγεθος του ιδεατού ενταμιευτή (*vb_count* στο Σχήμα 5), αφαιρώντας τα πακέτα που θα είχαν φύγει από την ουρά, τη χρονική διάρκεια που πέρασε από την προηγούμενη έλευση πακέτου, αν η ουρά εξυπηρετείτο με ρυθμό θC .
 - Έπειτα ελέγχεται αν το μέγεθος του ιδεατού ενταμιευτή γίνει μικρότερο ή ίσο με μηδέν (αν έχει αδειάσει), οπότε μία μεταβλητή (*mark_flag* στο Σχήμα 5), η οποία δηλώνει αν ένα πακέτο πρέπει να μαρκαριστεί ή όχι, παίρνει τιμή ψευδή (*false*). Στη συνέχεια ελέγχεται αν το μέγεθος του ιδεατού ενταμιευτή που υπολογίσαμε ξεπερνάει το όριο θB οπότε η μεταβλητή *mark_flag* παίρνει τιμή αληθή (*true*).
 - Ύστερα προσθέτουμε στον ιδεατό ενταμιευτή το μέγεθος του πακέτου που έφτασε στην ουρά.
 - Τέλος, μαρκάρεται ή όχι το πακέτο ανάλογα με την τιμή της μεταβλητής *mark_flag*.

Στο σημείο αυτό θα πρέπει να σημειώσουμε, ότι επειδή η μελέτη των αλγορίθμων που μελετήσαμε, πραγματοποιήθηκε για μία περιοχή που δεν επιθυμούσαμε απώλειες πακέτων, φροντίσαμε ώστε οι μηχανισμοί μαρκαρίσματος που χρησιμοποιήσαμε στα πειράματά μας να μην απορρίπτουν ποτέ πακέτα. Ακολουθήσαμε αυτή την πρακτική για να είμαστε σε θέση να αποκρυσταλλώσουμε μία ξεκάθαρη γνώμη, για τη συμπεριφορά των αλγορίθμων που μελετήσαμε, χωρίς αυτή να επηρεάζεται από μηχανισμούς αναμετάδοσης πακέτων που ενεργοποιούνται σε μία πιθανή απώλεια πακέτου (π.χ. ο μηχανισμός fast-retransmit του TCP). Βέβαια, υποθέτουμε ότι, με μεγάλη πιθανότητα, οι αλγόριθμοι αυτοί θα παρουσιάζουν παρόμοια συμπεριφορά και σε περιπτώσεις που χάνονται πακέτα.

3.2. Πραγματική υλοποίηση

3.2.1. Περιγραφή συστήματος

Το σύστημα στο οποίο εκτελέσαμε τα πειράματα που αφορούσαν στην πραγματική υλοποίηση των αλγορίθμων που περιγράψαμε στο Κεφάλαιο 2, αποτελείται από τρεις σταθμούς εργασίας πλατφόρμας FreeBSD [15] (Σχήμα 6).



Σχήμα 6 Σύστημα πραγματικής υλοποίησης

Στους δύο ακριανούς σταθμούς, εγκαταστάθηκε το λειτουργικό σύστημα FreeBSD 3.4 το οποίο υλοποιεί την έκδοση Reno του TCP. Στην έκδοση αυτή προστέθηκε το επίρραμμα (patch) για υποστήριξη του μηχανισμού ECN⁵, το οποίο

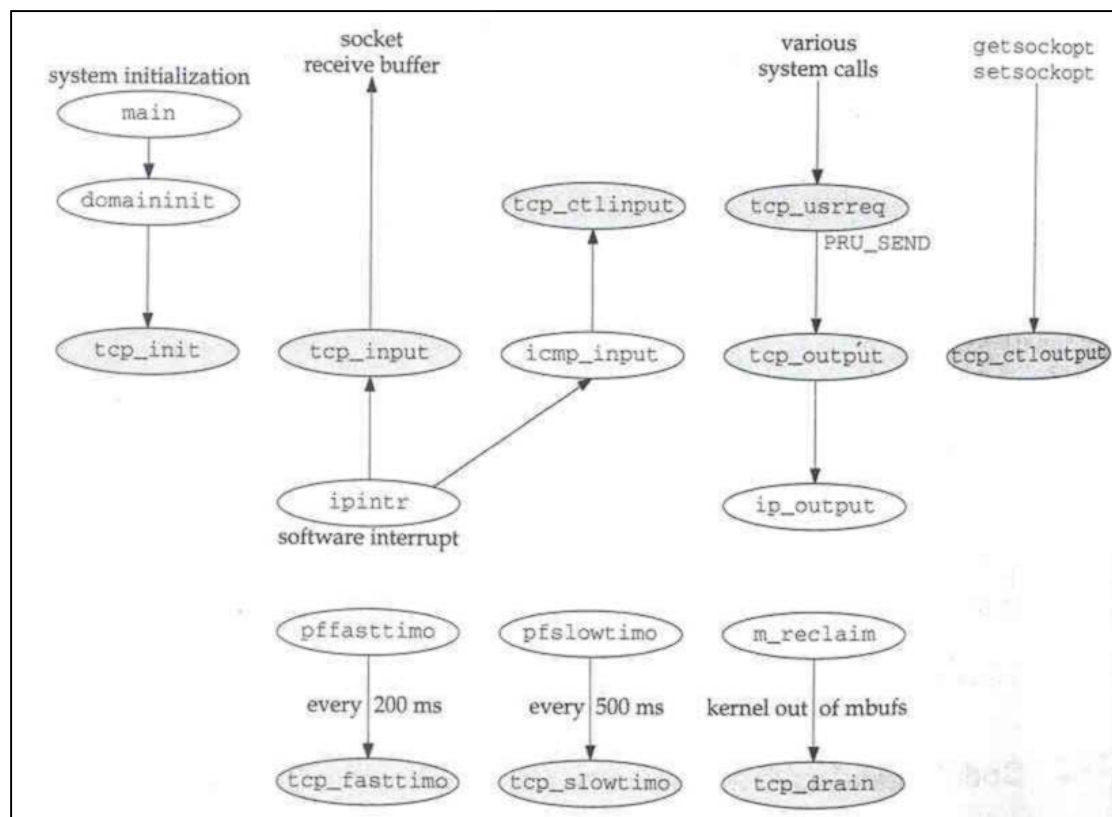
⁵ Τα προγράμματα που χρειάζονται για την υποστήριξη των μηχανισμών του Κεφαλαίου 2 από το λειτουργικό σύστημα FreeBSD καθώς και οδηγίες εγκατάστασης τους μπορούν να βρεθούν στο <http://www.ics.forth.gr/netgroup/publications/wtp.html>

διανέμεται με τη βιβλιοθήκη Alternate Queue (ALTQ), έκδοσης 2.1, την οποία θα περιγράψουμε παρακάτω. Στη συνέχεια τροποποιήσαμε την παραπάνω έκδοση, ώστε να υλοποιείται ο αλγόριθμος WTP.

Στο δρομολογητή (μεσαίος σταθμός εργασίας) εγκαταστάθηκε το λειτουργικό σύστημα FreeBSD 4.0, στο οποίο έχει προστεθεί η βιβλιοθήκη ALTQ. Η βιβλιοθήκη αυτή υλοποιεί διάφορες πολιτικές εξυπηρέτησης ουρών (queuing disciplines), μεταξύ αυτών και το μηχανισμό RED. Παράλληλα υλοποιήσαμε το μηχανισμό VQ, ως επέκταση του μηχανισμού RED, όπως περιγράφουμε στην Ενότητα 3.2.3.2. Ο έλεγχος του χρόνου αποστολής-άφιξης (RTT) εξομοιώθηκε με τη χρήση του προγράμματος Dummynet [16], το οποίο εγκαταστάθηκε στο σταθμό εργασίας του παραλήπτη.

3.2.2. Υλοποίηση WTP στον πυρήνα του λειτουργικού

Στο Σχήμα 7, φαίνονται οι σχέσεις των διάφορων συναρτήσεων του TCP με τις άλλες συναρτήσεις του πυρήνα (kernel) του λειτουργικού FreeBSD. Οι σκιασμένες ελλείψεις αποτελούν τις εννέα βασικές συναρτήσεις του TCP, τις οποίες αναλύσαμε και τροποποιήσαμε ώστε να υποστηρίζεται ο μηχανισμός WTP από το λειτουργικό σύστημα.



Σχήμα 7 Σχέση των συναρτήσεων του TCP με τις υπόλοιπες του πυρήνα

Πιο συγκεκριμένα οι συναρτήσεις οι οποίες έπρεπε να τροποποιηθούν είναι οι:

- **tcp_init:** Η συνάρτηση αυτή καλείται από το λειτουργικό σύστημα κάθε φορά που ξεκινά μία νέα σύνδεση TCP και έχει σκοπό να αρχικοποιήσει τις μεταβλητές του *TCP control block* (TCB). Το TCB αποτελεί μία μεγάλη δομή η οποία είναι μοναδική για κάθε σύνδεση και περιέχει όλη την πληροφορία που χρειάζεται το λειτουργικό για τη σύνδεση που αντιπροσωπεύει. Στην παραπάνω δομή προσθέσαμε τις μεταβλητές, που περιγράφουμε παρακάτω και είναι αναγκαίες για την υλοποίηση του WTP στο πραγματικό σύστημα. Οι μεταβλητές αυτές, οι οποίες αρχικοποιούνται στη συνάρτηση `tcp_init`, είναι:
 - **tcp_ecn:** η οποία δηλώνει αν η σύνδεση χρησιμοποιεί ή όχι τον μηχανισμό ECN όπως τον έχουμε περιγράψει στο Κεφάλαιο 1.
 - **enable_wtp:** η οποία δηλώνει αν η σύνδεση θα χρησιμοποιήσει ή όχι το μηχανισμό WTP για την ενημέρωση της τιμής του παραθύρου συμφόρησης. Αν η μεταβλητή αυτή είναι αληθής τότε αυτόματα γίνεται αληθής και η μεταβλητή `tcp_ecn`.
 - **wtp_w:** η μεταβλητή αυτή είναι ένας ακέραιος αριθμός ο οποίος αντιστοιχεί στην παράμετρο επιθυμίας χρέωσης ανά χρόνο αποστολής-άφιξης, \bar{w} , του WTP.
 - **wtp_invk:** η μεταβλητή αυτή είναι ένας ακέραιος αριθμός ο οποίος αντιστοιχεί στο $\frac{1}{k}$, όπου \bar{k} είναι ο παράγοντας σύγκλισης ανά χρόνο αποστολής-άφιξης του WTP.
- **tcp_ctloutput:** η συνάρτηση αυτή είναι υπεύθυνη για την επικοινωνία των χρηστών με τις συνδέσεις μέσω των κλήσεων συστήματος (system calls) `getsockopt` και `setsockopt`. Με τις συναρτήσεις αυτές ο χρήστης έχει τη δυνατότητα να τροποποιήσει τις παραμέτρους μίας σύνδεσης που επιθυμεί ή να λάβει πληροφορίες γι' αυτή, μέσω του TCB της σύνδεσης. Έτσι τροποποιήσαμε τη συνάρτηση `tcp_ctloutput` ώστε να υπάρχει η δυνατότητα να θέτει ο χρήστης τιμές στις μεταβλητές του WTP που περιγράψαμε παραπάνω.
- **tcp_input:** η συνάρτηση αυτή μαζί με την επόμενη που θα περιγράψουμε περιέχουν το μεγαλύτερο μέρος υλοποίησης του TCP. Πιο συγκεκριμένα, η `tcp_input` καλείται κάθε φορά που φτάνει ένα πακέτο από το δίκτυο (είτε αυτό αποτελεί πακέτο δεδομένων είτε επιβεβαίωση λήψης). Η ενεργοποίηση του WTP σε αυτή τη συνάρτηση γίνεται, θέτοντας τις

Ψευδοκώδικας για WTP

```
if ({packet is marked})
{
    cwnd += (w*(MSS*MSS)/cwnd - MSS)/inv_k;
}
else
{
    cwnd += (w*(Mss*Mss)/cwnd)/inv_k;
}
```

Σχήμα 8 Ψευδοκώδικας για τον αλγόριθμο WTP

μεταβλητές *tcp_ecn* και *enable_wtp* που περιγράψαμε παραπάνω. Οι αλλαγές έγιναν μόνο στο μέρος του κώδικα που αφορούσε τη διαχείριση του παραθύρου συμφόρησης. Θα πρέπει να πούμε στο σημείο αυτό ότι πριν προσθέσουμε τον κώδικα του WTP είχαν επιτελεστεί όλες οι αλλαγές που χρειάζονταν για την υποστήριξη του μηχανισμού ECN με το επίρραμμα που είχαμε εφαρμόσει στον πηγαίο κώδικα του πυρήνα πρωτότερα.

- **tcp_output:** η συνάρτηση αυτή καλείται κάθε φορά που πρέπει να αποσταλεί ένα πακέτο προς το δίκτυο (είτε είναι πακέτο δεδομένων είτε επιβεβαίωση λήψης). Κατά απόλυτη αντιστοιχία με την *tcp_input*, έτσι και σε αυτή τη συνάρτηση οι αλλαγές για την υποστήριξη του WTP επιτελέστηκαν στον ήδη τροποποιημένο κώδικα από το επίρραμμα για την υποστήριξη του μηχανισμού ECN.

3.2.3. Θέματα υλοποίησης του WTP σε πραγματικό περιβάλλον

Στο παρακάτω σχήμα φαίνεται ο ψευδοκώδικας της πραγματικής υλοποίησης του WTP. Παρατηρείστε ότι η σχέση (15) τροποποιήθηκε έτσι ώστε το παράθυρο συμφόρησης να εκφράζεται σε bytes, αφού με τον τρόπο αυτό χρησιμοποιείται στις πραγματικές υλοποιήσεις του TCP. Επίσης, για την αποφυγή πράξεων κινητής υποδιαστολής στον πυρήνα του λειτουργικού, χρησιμοποιούμε τον αντίστροφο του \bar{k} , ο οποίος παίρνει ακέραιες τιμές.

Οι μεταβλητές του ψευδοκώδικα για το WTP έχουν την εξής σημασία:

- **cwnd:** είναι η τρέχουσα τιμή του παραθύρου συμφόρησης

- **MSS**: είναι το μέγιστο μέγεθος πακέτου που μπορεί να μεταδοθεί σε bytes.
- **inv_k**: έχει τιμή ίση με $\frac{1}{k}$
- **w**: έχει τιμή ίση με \bar{w}

Υπενθυμίζουμε στο σημείο αυτό ότι όλες οι μεταβλητές είναι ακέραιοι αριθμοί.

3.2.4. Υλοποίηση αλγορίθμων μαρκαρίσματος

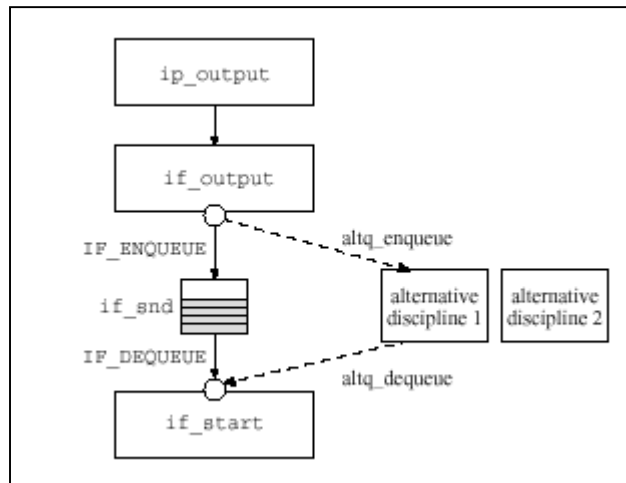
3.2.4.1. *Alternate Queue (ALTQ)*

Η πλατφόρμα Alternate Queue (ALTQ) [19] αποτελεί ένα πλαίσιο (framework) υποστήριξης διάφορων πολιτικών εξυπηρέτησης ουρών (queuing disciplines), για το λειτουργικό σύστημα FreeBSD. Αποτελεί κατάλληλη πλατφόρμα για έρευνα η οποία σχετίζεται με διαχείριση κίνησης δικτύων, ενώ παράλληλα, παρέχει τη δυνατότητα εξομοίωσης σταθμών εργασίας σε δρομολογητές δικτύου. Οι αρχές ουρών που υλοποιεί η πλατφόρμα ALTQ είναι οι: class based queuing (CBQ), weighted fair queuing (WFQ) και random early detect (RED) με υποστήριξη του μηχανισμού ECN.

Ο βασικός σχεδιασμός της ALTQ, είναι αρκετά απλός: η διεπιφάνεια διαχείρισης ουράς σχεδιάστηκε με τρόπο που να παρέχει τη δυνατότητα επιλογής κάποιας αρχής ουρών όπως φαίνεται στο Σχήμα 9. Για την υλοποίηση της πλατφόρμας ALTQ, προστέθηκαν κάποια πεδία στη δομή ifnet του πυρήνα του λειτουργικού. Τα πεδία αυτά είναι μία μεταβλητή που ορίζει τον τύπο της πολιτικής εξυπηρέτησης ουρών, μία μεταβλητή που δηλώνει την κατάσταση της πολιτικής εξυπηρέτησης της ουράς (π.χ. αν είναι έτοιμη προς λειτουργία ή αν χρησιμοποιείται ήδη κτλ.), ένας δείκτης πρόσβασης στην πολιτική εξυπηρέτησης ουρών που χρησιμοποιείται και δείκτες προς τις συναρτήσεις πρόσθεσης / αφαίρεσης πακέτου στην ουρά για κάθε μία πολιτική εξυπηρέτησης.

Στόχος στην υλοποίηση της ALTQ, ήταν να γίνονται όσο το δυνατόν λιγότερες αλλαγές στον πηγαίο κώδικα του πυρήνα. Παρόλα αυτά, ο πυρήνας του λειτουργικού δεν παρέχει αρκετή αφαιρετικότητα όσον αφορά την υλοποίηση πολιτικών εξυπηρέτησης ουρών. Αυτό έχει ως αποτέλεσμα να υπάρχουν αρκετά τμήματα του κώδικα του πυρήνα τα οποία θεωρούν ότι η πολιτική εξυπηρέτησης ουρών που υιοθετείται από το σύστημα είναι η First In First Out (FIFO).

Ειδικότερα, η παραπάνω αντιμετώπιση υιοθετείται και από τους οδηγούς συστήματος, πράγμα που καθιστά απαραίτητη την τροποποίησή τους. Δεν είναι όμως, τόσο απλό να τροποποιηθούν όλοι οι οδηγοί συστήματος που υπάρχουν. Για το λόγο αυτό, υιοθετείται από την ALTQ, μία προσέγγιση η οποία επιτρέπει τη συνύπαρξη των τροποποιημένων οδηγών με αυτούς που δεν έχουν υποστεί αλλαγή. Με τον τρόπο αυτό καθίσταται δυνατή η τροποποίηση μόνο των οδηγών που ενδιαφέρουν κάποιον που χρησιμοποιεί την πλατφόρμα ALTQ.



Σχήμα 9 Αρχιτεκτονική της πλατφόρμας ALTQ

Οι πολιτικές εξυπηρέτησης ουρών ελέγχονται από κλήσεις συστήματος που ανήκουν στην κατηγορία εισόδου / εξόδου (ioctl system.calls) μέσω συσκευών ουρών (queuing devices) . Η ALTQ δηλώνεται στο σύστημα ως μία συσκευή χαρακτήρων και κάθε πολιτική εξυπηρέτησης ουρών ως ελάσσων συσκευή της ALTQ. Υπάρχουν τέσσερις λειτουργίες που διαχειρίζονται τις αρχές ουρών:

- *attach*: δηλώνει ότι μία πολιτική εξυπηρέτησης ουράς θα χρησιμοποιείται από το σύστημα
- *detach*: δηλώνει ότι η πολιτική εξυπηρέτησης ουράς, που είχε δηλωθεί με την *attach* ως χρησιμοποιούμενη, δε χρησιμοποιείται πια από το σύστημα
- *enable*: ενεργοποιεί μία πολιτική εξυπηρέτησης ουράς
- *disable*: απενεργοποιεί μία πολιτική εξυπηρέτησης ουράς

Η λειτουργία *attach* δεν ενεργοποιείται μέχρι να εκπληρωθεί η λειτουργία *enable*. Όταν μία αρχή ουράς παύει να ισχύει, τότε το σύστημα χρησιμοποιεί την πολιτική εξυπηρέτησης FIFO.

3.2.4.2. Υλοποίηση του μηχανισμού VQ

Για την υλοποίηση του μηχανισμού Virtual Queue (VQ), δεν δημιουργήσαμε μία ξεχωριστή πολιτική εξυπηρέτησης ουρών της πλατφόρμας ALTQ, αλλά επεκτείναμε την πολιτική εξυπηρέτησης ουρών RED ώστε να υποστηρίζει το μηχανισμό αυτό.

Πιο συγκεκριμένα, τροποποιήσαμε τον κώδικα της πολιτικής εξυπηρέτησης ουρών, που υλοποιεί το μηχανισμό RED, ώστε αν η μεταβλητή \min_{thr} του RED (βλ. ενότητα 2.4.2) πάρει τιμή μηδέν να εκτελείται ο μηχανισμός VQ. Στην περίπτωση αυτή, οι μεταβλητές \max_{thr} και $\max_{p_inverse}$ ⁶ περιέχουν τις τιμές θ και μεγέθους της ουράς αντίστοιχα.

Η ενεργοποίηση του μηχανισμού VQ, παρακάμπτει τη συνάρτηση προσθήκης πακέτου στην ουρά (`enqueue`) που εκτελείται όταν έχει επιλεγεί ο μηχανισμός RED ως πολιτική εξυπηρέτησης ουράς του συστήματος. Έτσι κάθε φορά που καταφτάνει ένα πακέτο από το δίκτυο και πρέπει να προστεθεί στην ουρά, εκτελείται ο κώδικας που υλοποιεί τη πολιτική εξυπηρέτησης ουρών VQ η οποία παρουσιάζεται στο δεξί μέρος τους Σχήματος 5 της ενότητας 3.1.3.

3.2.5. Θέματα υλοποίησης της πολιτικής εξυπηρέτησης ουρών VQ σε πραγματικό περιβάλλον

Στο παρακάτω σχήμα φαίνεται ο ψευδοκώδικας της πραγματικής υλοποίησης της VQ. Εξαιτίας του περιορισμού απαγόρευσης πράξεων κινητής υποδιαστολής στον πυρήνα του λειτουργικού, για τον υπολογισμό του τρέχοντος μεγέθους του ιδεατού ενταμιευτή στον αλγόριθμο VQ, αναγκαστήκαμε να χρησιμοποιήσουμε τεχνικές κινητού υπολοίπου στην διαίρεση, ώστε να μην υπάρχει σφάλμα λόγω της στρογγυλοποίησης που υφίσταται το αποτέλεσμα όταν μετατρέπεται σε ακέραιο.

Οι μεταβλητές του ψευδοκώδικα για το VQ έχουν την εξής σημασία:

- *virtual_bcount*: η μεταβλητή αυτή διατηρεί το τρέχον μέγεθος του ιδεατού ενταμιευτή.
- *virtual_buffer*: η μεταβλητή αυτή δείχνει το μέγιστο μέγεθος της ουράς του ιδεατού ενταμιευτή.
- *bandwidth*: η μεταβλητή αυτή έχει αποθηκευμένη την τιμή της χωρητικότητας του συνδέσμου.

⁶ Η $\max_{p_inverse}$ είναι ίση με $1/\max_p$ όπως αυτή ορίστηκε στην ενότητα 2.4.2

- *mark_flag*: η μεταβλητή αυτή παίρνει αληθή τιμή στη περίπτωση που υπάρχει συμφόρηση, αλλιώς έχει ψευδή τιμή.

Υπενθυμίζουμε στο σημείο αυτό ότι όλες οι μεταβλητές είναι ακέραιοι αριθμοί.

Ψευδοκώδικας για VQ

```
time_now = { current_time }
virtual_bcount -= bandwidth*(time_now-last_time_vb)
if(virtual_bcount < 0){
    virtual_bcount = 0
    mark_flag = FALSE
}

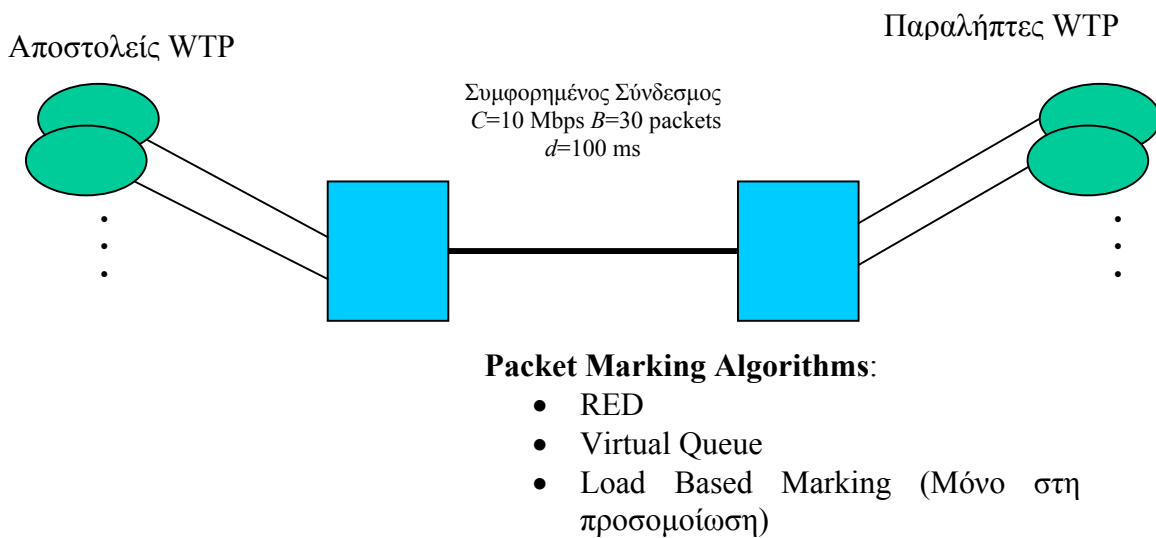
if(virtual_bcount + { current_pkt_size } < virtual_buffer){
    mark_flag = TRUE
}
else
{
    virtual_bcount += { current_pkt_size }
}

if(mark_flag){
    { set ecn bit in IP header }
}
```

Σχήμα 10 Ψευδοκώδικας για τον αλγόριθμο VQ

4. Πειραματικά αποτελέσματα

Τα πειράματα που εκτελέσαμε στα πλαίσια της έρευνάς μας βασίζονται στην παρακάτω τοπολογία δικτύου (Σχήμα 11). Η κίνηση παράγεται από απλοποιημένες πηγές FTP, οι οποίες χρησιμοποιούν όσο περισσότερο εύρος ζώνης τους επιτρέπεται από το TCP. Οι χρόνοι έναρξης των πηγών επιλέγονται τυχαία και τα συγκεντρωτικά αποτελέσματα τα οποία παρουσιάζουμε είναι οι μέσες τιμές δέκα επαναλήψεων του ίδιου πειράματος.



Σχήμα 11 Βασική τοπολογία δικτύου

Στην ενότητα 4.1 σχολιάζουμε τα πειραματικά αποτελέσματα που αφορούν στη φάση της αποφυγής της συμφόρησης, διερευνώντας την επιτευξιμότητα του αλγόριθμου WTP, που αναφέραμε στο Κεφάλαιο 2, όσον αφορά στη διαφοροποίηση υπηρεσίας που προσφέρει και την αλληλεπίδρασή του με τους αλγόριθμους μαρκαρίσματος που χρησιμοποιήσαμε (Σχήμα 11). Παράλληλα παρουσιάζουμε την επίδραση του παράγοντα σύγκλισης \bar{k} του WTP, ως προς την ταχύτητα σύγκλισης των συνδέσεων. Τέλος παρουσιάζουμε μία σύγκριση του αλγόριθμου WTP, με τον αλγόριθμο MulTCP ο οποίος αποτελεί και αυτός μία παραλλαγή του TCP για παροχή διαφοροποίησης υπηρεσιών.

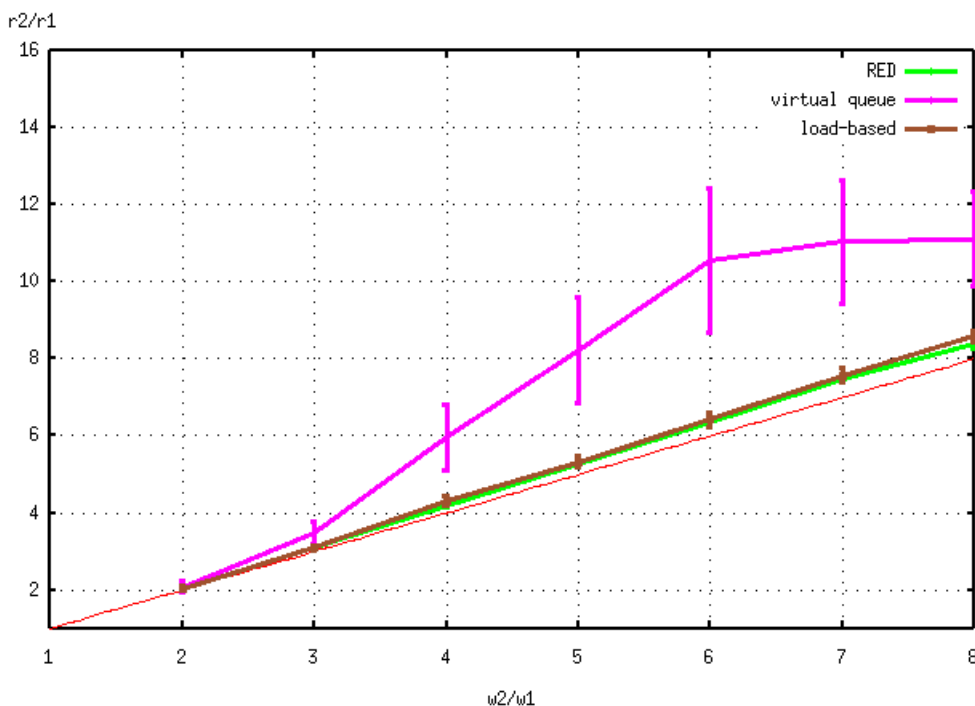
Στην ενότητα 4.2, παρουσιάζουμε τα πειραματικά αποτελέσματα που αφορούν στη φάση της γρήγορης σύγκλισης, συγκρίνοντας την επιτευξιμότητα των αλγορίθμων που αναφέραμε στο Κεφάλαιο 2, όσο αφορά την ταχύτητα σύγκλισης

που επιτυγχάνουν αλλά και την ικανότητα, που παρουσιάζουν, στη διαφοροποίηση υπηρεσιών. Στη συνέχεια διερευνούμε διεξοδικότερα, την συμπεριφορά των αλγόριθμων όσο αφορά στη διαφοροποίηση που μπορούν να προσφέρουν ανάλογα με την επιθυμία χρέωσης του χρήστη. Τέλος, παρουσιάζουμε τα αποτελέσματα που αφορούν τους χρόνους σύγκλισης των αλγόριθμων Slow start του TCP και του Large K.

4.1. Μελέτη αλγόριθμων για τη φάση αποφυγής της συμφόρησης

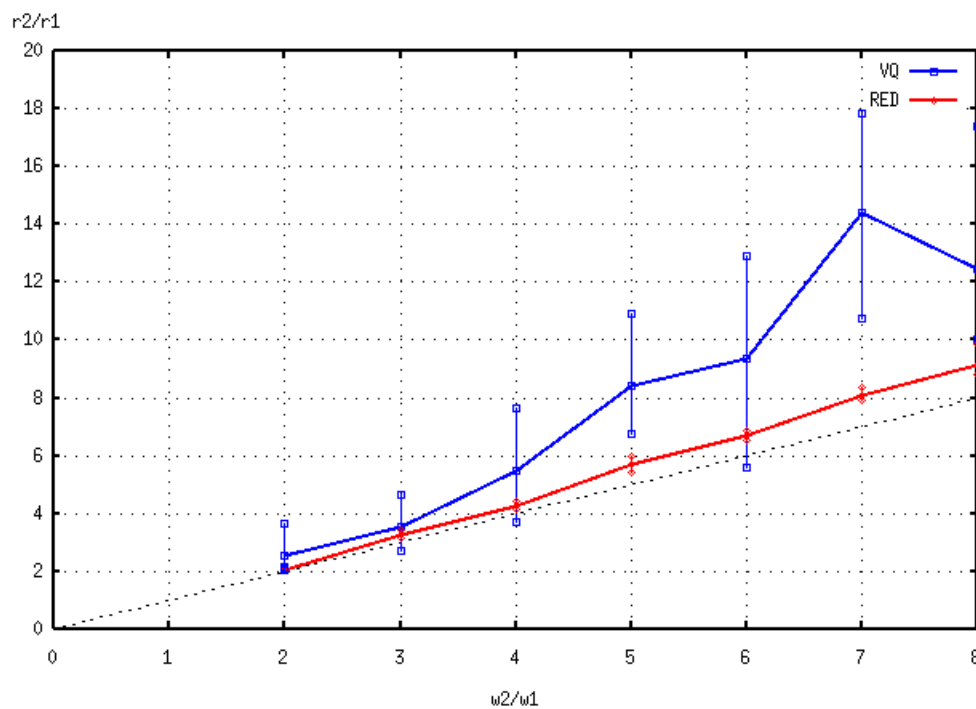
4.1.1. Διαφοροποίηση ως προς την επιθυμία χρέωσης

Στα Σχήματα 12 (αποτελέσματα προσομοίωσης) και 13 (αποτελέσματα πραγματικής υλοποίησης) παρουσιάζεται ο μέσος λόγος των ρυθμών αποστολής για διαφορετικούς λόγους τιμών επιθυμίας χρέωσης. Τα αποτελέσματα αυτά προέκυψαν από δέκα ανεξάρτητες εκτελέσεις πειράματος με τις ίδιες παραμέτρους. Κάθε σύνδεση ήταν τύπου FTP μεγάλης διάρκειας και ο χρόνος έναρξης καθемίας από αυτές επιλέχθηκε τυχαία σε διάστημα $[0, 5]$ δευτερόλεπτα. Τέλος, η διαπερατότητα υπολογίστηκε για χρονική διάρκεια $[60, 180]$ δευτερολέπτων.



Σχήμα 12 Λόγος της διαπερατότητας (προσομοίωση): $k=0.5$, $C=10$ Mbps, $RTT=200$ msec, $N=10$. RED: $\min_{thr}=5$, $\max_{thr}=15$, $\max_p=0.1$, $w_q=0.002$. VQ: $v_b=0.95 \cdot 30$, $v_c=0.95 \cdot 10$ Mbps. LBM: $t_{avg}=0.5$, $p_0=0.6$, $\alpha=0.71$

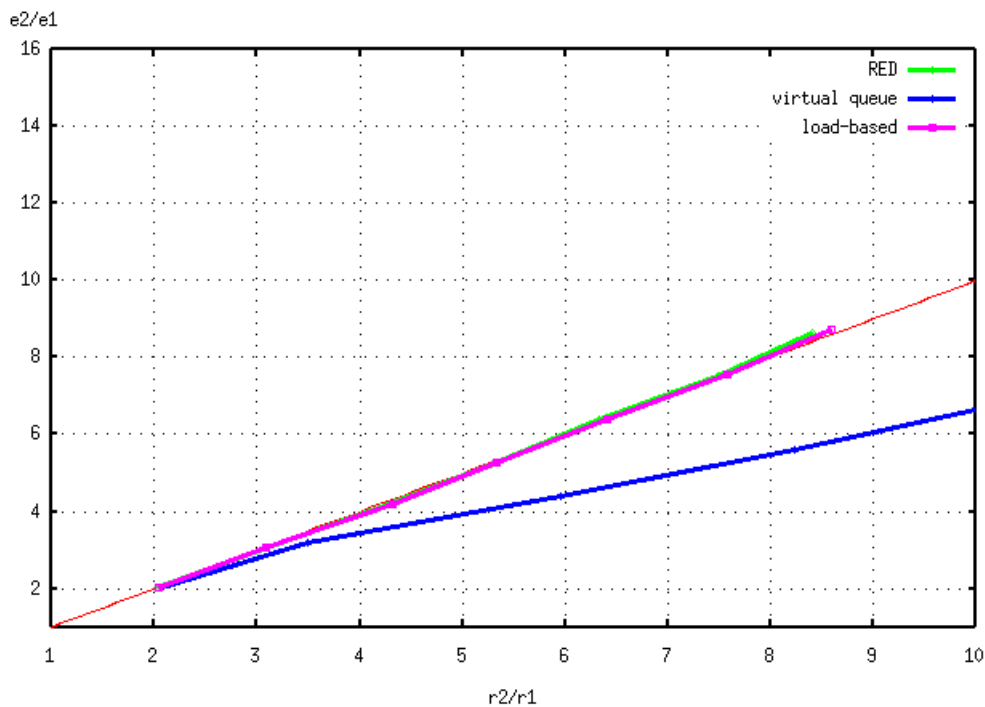
Παρατηρούμε ότι η διαφοροποίηση υπηρεσίας που επιτυγχάνεται για τους μηχανισμούς RED και LBM είναι περίπου η ίδια. Επιπλέον, ο λόγος του μέσου ρυθμού είναι λίγο πιο πάνω από τη διαγώνιο. Από την άλλη μεριά, στην περίπτωση του μηχανισμού VQ παρατηρούμε ότι για μεγάλες τιμές του λόγου των τιμών επιθυμίας χρέωσης, ο λόγος των ρυθμών αποστολής είναι μεγαλύτερος από το λόγο των τιμών επιθυμίας χρέωσης. Επιπλέον, όπως δείχνει το διάστημα εμπιστοσύνης (confidence interval), οι διακυμάνσεις στη διαφοροποίηση υπηρεσίας είναι επίσης πολύ μεγαλύτερες σε σχέση με τις διακυμάνσεις των άλλων δύο μηχανισμών μαρκαρίσματος. Στη συνέχεια ερμηνεύουμε τις παρατηρήσεις αυτές.



Σχήμα 13 Λόγος διαπερατότητας (πραγματική υλοποίηση): $k=0.5$, $C=10$ Mbps, $RTT=200$ msec, $N=10$. RED: $\min_{thr}=5$, $\max_{thr}=15$, $\max_p=0.1$, $w_q=0.002$. VQ: $v_b=0.7\cdot 30$, $v_c=0.7\cdot 10$ Mbps.

Η πρώτη παρατήρηση μπορεί εύκολα να εξηγηθεί με τη βοήθεια του Σχήματος 14, το οποίο παρουσιάζει το λόγο των μαρκαρισμένων πακέτων σαν συνάρτηση του λόγου των μέσων ρυθμών. Το σχήμα δείχνει ότι ένα μικρότερο ποσοστό των πακέτων που ανήκουν σε μία σύνδεση με μεγαλύτερη τιμή επιθυμίας χρέωσης, έχουν μαρκαριστεί. Δηλαδή, η πιθανότητα μαρκαρίσματος είναι μικρότερη για συνδέσεις με μεγαλύτερη τιμή επιθυμίας χρέωσης. Αυτό οφείλεται στο συνδυασμό των επόμενων δύο παραγόντων: α) μία σύνδεση με μικρότερη τιμή επιθυμίας χρέωσης αποστέλλει μικρότερο αριθμό πακέτων σε διάστημα ενός χρόνου αποστολής-άφιξης (RTT) και β) τα πακέτα συνήθως στέλνονται αμέσως το ένα μετά το άλλο (αυτό αποτελεί ιδιότητα οποιουδήποτε παραθυρικού αλγορίθμου ελέγχου συμφόρησης). Το αποτέλεσμα των δύο παραπάνω παραγόντων είναι να παράγεται από τις συνδέσεις με

μικρότερη τιμή επιθυμίας χρέωσης, εκρηκτικότερη κίνηση σε σχέση με τις συνδέσεις που έχουν μεγαλύτερη τιμή επιθυμίας χρέωσης. Οι εκρηκτικότερες κινήσεις όμως, είναι δυσκολότερο για ένα πολυπλέκτη (multiplexer – link) να διαχειριστούν. Το αποτέλεσμα είναι, να απαιτείται περισσότερο εύρος ζώνης από ότι χρειάζεται ο μέσος ρυθμός αποστολής. Ο μηχανισμός VQ, έχει την ιδιότητα να διαφοροποιεί τις συνδέσεις βάσει της εκρηκτικότητας κίνησης που παρουσιάζουν [1]. Έτσι, ο αλγόριθμος αυτός μαρκάρει ένα υψηλότερο ποσοστό πακέτων που ανήκουν σε εκρηκτικότερες συνδέσεις, οι οποίες όπως εξηγήσαμε παραπάνω, είναι αυτές με τη μικρότερη τιμή επιθυμίας χρέωσης.

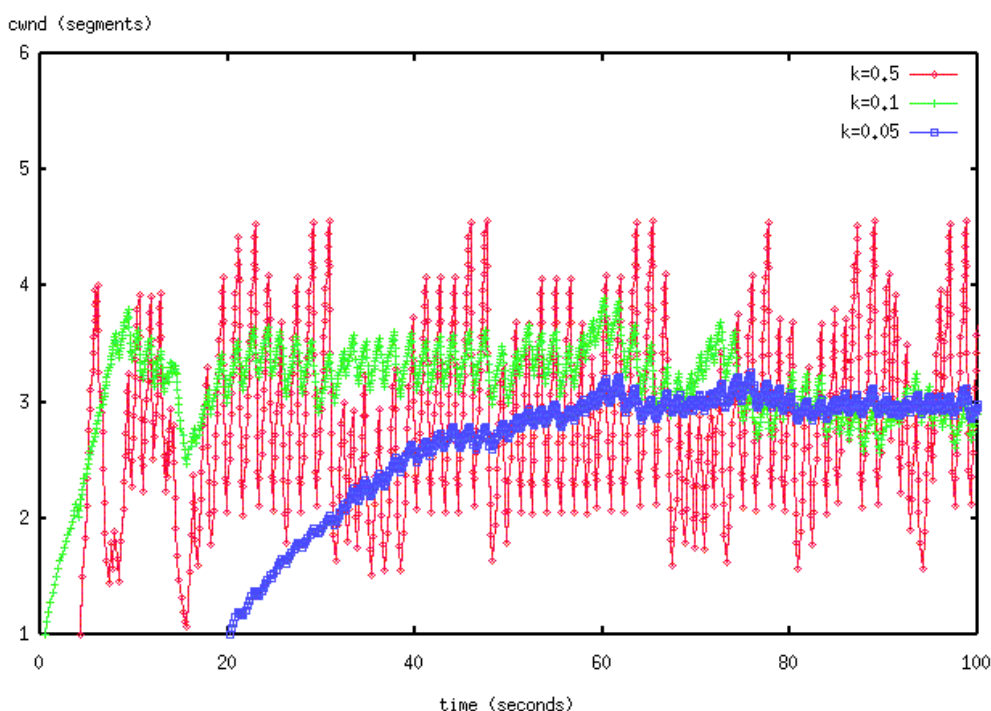


Σχήμα 14 Λόγος μαρκαρισμένων πακέτων (προσομοίωση): $k=0.5$, $C=10$ Mbps, $RTT=200$ msec, $N=10$. RED: $\min_{thr}=5$, $\max_{thr}=15$, $\max_p=0.1$, $w_q=0.002$. VQ: $v_b=0.95 \cdot 30$, $v_c=0.95 \cdot 10$ Mbps. LBM: $t_{avg}=0.5$, $\rho_0=0.6$, $\alpha=0.71$

Στη συνέχεια εξηγούμε τη δεύτερη παρατήρηση που προέκυψε από τα Σχήματα 12 και 13, δηλαδή ότι παρουσιάζονται μεγαλύτερες διακυμάνσεις στη διαφοροποίηση υπηρεσίας στο μηχανισμό VQ σε σχέση με τους αλγόριθμους RED και LBM. Στους δύο τελευταίους αλγόριθμους, η πιθανότητα μαρκαρίσματος εξαρτάται από κάποια μέση τιμή. Πιο συγκεκριμένα, εξαρτάται από το μέσο μέγεθος ουράς για το RED και από το μέσο φόρτο για το LBM. Το γεγονός αυτό έχει ως αποτέλεσμα, ομαλότερες αλλαγές της πιθανότητας μαρκαρίσματος και έτσι ομαλότερες αλλαγές του παραθύρου συμφόρησης.

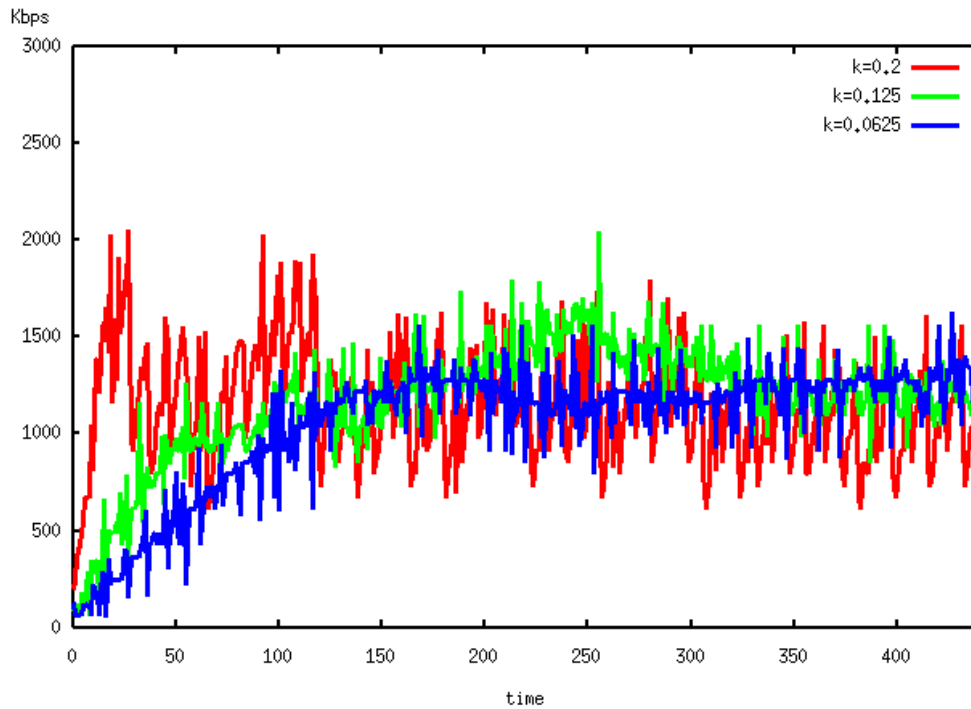
4.1.2. Επίδραση του παράγοντα k στην ταχύτητα σύγκλισης

Στο Σχήμα 15, βλέπουμε τις τιμές του παράθυρου συμφόρησης, σε περιβάλλον προσομοίωσης, συνδέσεων WTP με διαφορετικές τιμές του παράγοντα \bar{k} . Στο Σχήμα 16, παρουσιάζεται ο ρυθμός αποστολής συνδέσεων WTP για διαφορετικές τιμές του παράγοντα \bar{k} , σε πραγματικό περιβάλλον. Παρατηρούμε ότι οι ροές με μεγαλύτερες τιμές του \bar{k} συγκλίνουν γρηγορότερα από τις ροές με μικρότερες τιμές του παράγοντα \bar{k} . Παρόλα αυτά, οι ροές με μικρότερες τιμές του \bar{k} επιδεικνύουν μικρότερες διακυμάνσεις στο σημείο ισοροπίας.



Σχήμα 15 Επίδραση του παράγοντα \bar{k} (αποτελέσματα προσομοίωσης): $C=10$ Mbps, $RTT=200$ msec $w=1$. VQ: $v_b=0.9\cdot 30$, $v_c=0.9\cdot 10$ Mbps.

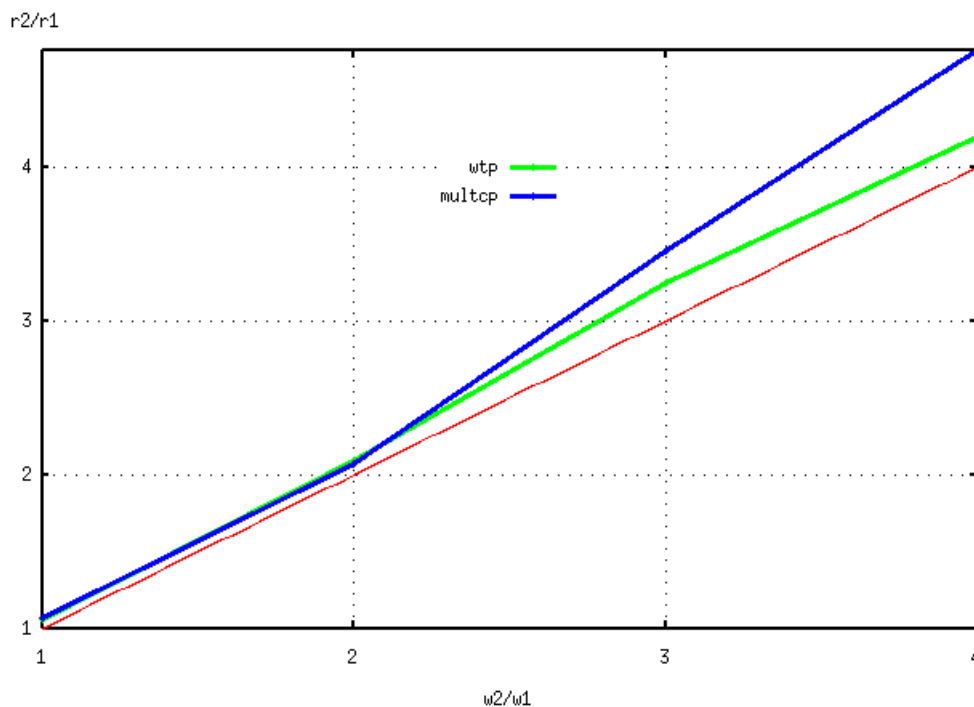
Για παράδειγμα, για $\bar{k}=0.05$ ο χρόνος σύγκλισης είναι περίπου σαράντα δευτερόλεπτα (Σχήμα 15), μία χρονική διάρκεια στην οποία συνήθως οι περισσότερες συνδέσεις έχουν ολοκληρωθεί: π.χ. οι κίνηση ενός εξυπηρετητή Internet (web server) έχει μέσο μέγεθος περί τα 8 – 10 Kbytes και μέση διάρκεια 2 – 9 δευτερόλεπτα. Αυτή η παρατήρηση καθιστά αναγκαία την ύπαρξη μίας φάσης για γρήγορη σύγκλιση (μία περιοχή που μελετάμε στην ενότητα 4.2).



Σχήμα 16 Επίδραση του παράγοντα \bar{k} (αποτελέσματα πραγματικής υλοποίησης): $C=10$ Mbps, $RTT=200$ msec $w=1$. VQ: $v_b=0.9 \cdot 30$, $v_c=0.9 \cdot 10$ Mbps.

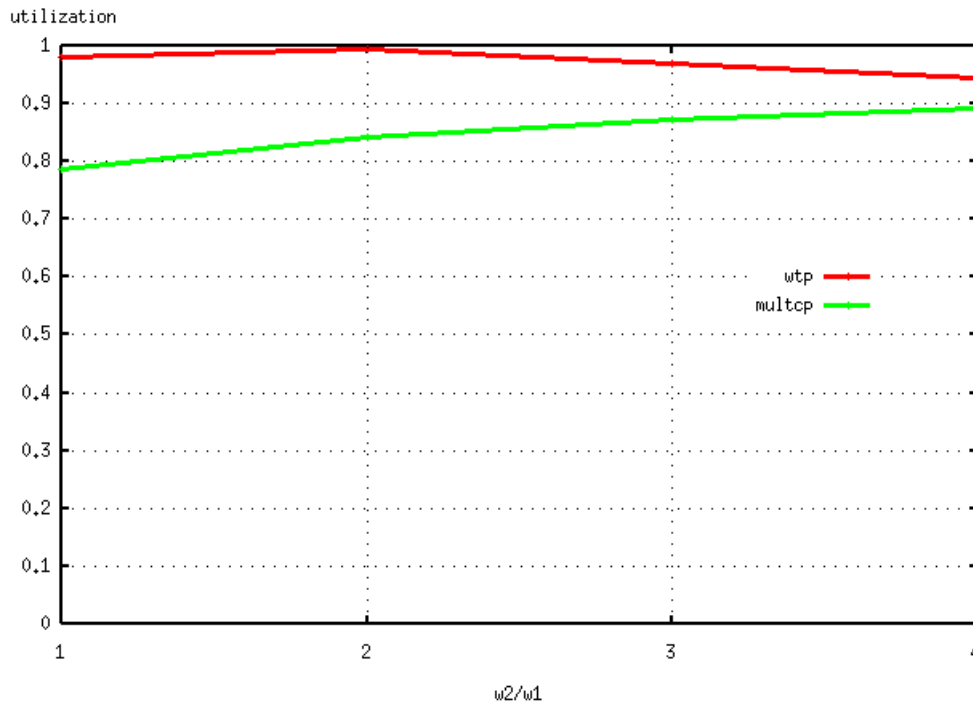
4.1.3. Σύγκριση WTP και MultTCP

Τα Σχήματα 17 και 18, συγκρίνουν τον μηχανισμό WTP με το μηχανισμό MultTCP. Ο δεύτερος μηχανισμός ακολουθεί μία διπλά πολλαπλασιαστική μείωση (double multiplicative decrease), αλλά παρέχει υποστήριξη για διαφοροποίηση υπηρεσιών. Το Σχήμα 17 δείχνει ότι ο λόγος της μέσης διαπερατότητας είναι μεγαλύτερος από το λόγο των βαρών για το MultTCP, επισημαίνοντας έτσι ότι στο μηχανισμό αυτό ευνοούνται οι συνδέσεις με μεγαλύτερα βάρη.



Σχήμα 17 Διαφοροποίηση υπηρεσίας για τους μηχανισμούς MultTCP και WTP: $C=10$ Mbps, $RTT=200$ ms, $k=0.5$, $N=10$, $N1=5$ ($w1=1$)
RED: $minthr=5$, $maxthr=15$, $maxp=0.1$, $wq=0.002$

Στο Σχήμα 18, συγκρίνεται η χρησιμοποίηση (utilization) του δικτύου που παρουσιάζουν οι μηχανισμοί MulTCP και WTP, όταν ο ίδιος αριθμός συνδέσεων πολυπλέκεται σε ένα σύνδεσμο που χρησιμοποιεί ως μηχανισμό μαρκαρίσματος τον RED. Παρατηρούμε ότι για μικρά βάρη ο WTP απολαμβάνει μεγαλύτερη χρησιμοποίηση σε σύγκριση με το MulTCP. Καθώς όμως τα βάρη μεγαλώνουν, η διαφορά μεταξύ των δύο μηχανισμών ελέγχου συμφόρησης γίνεται μικρότερη.



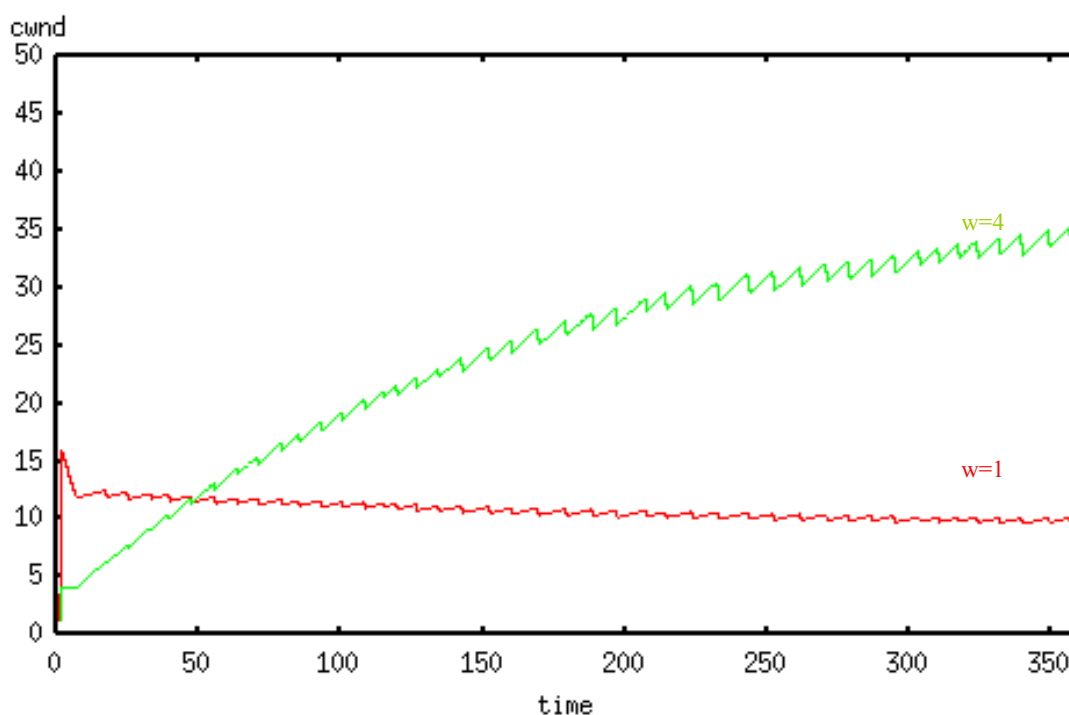
Σχήμα 18 Μέση χρησιμοποίηση δικτύου για τους μηχανισμούς MulTCP και WTP: $C=10$ Mbps, $RTT=200$ ms, $k=0.5$, $N=10$, $N1=5$ ($w_1=1$) RED: $minthr=5$, $maxthr=15$, $maxp=0.1$, $wq=0.002$

Εικάζουμε ότι αυτό οφείλεται στο εξής: Για το MulTCP, οι συνδέσεις είναι εκρηκτικές ακόμα και για μικρά βάρη, λόγω της ανάλογης μείωσης του παραθύρου συμφόρησης όταν ληφθούν μαρκαρισμένα πακέτα. Επιπλέον, καθώς τα βάρη αυξάνονται, αυξάνεται και η επιθετικότητα (aggressiveness) των συνδέσεων βοηθώντας τις έτσι να απολαμβάνουν μεγαλύτερη μέση διαπερατότητα. Από την άλλη μεριά, για μικρά βάρη οι συνδέσεις WTP παράγουν ομαλή κίνηση, και επομένως η μέση χρησιμοποίηση είναι μεγάλη. Καθώς τα βάρη αυξάνουν, όμως, οι συνδέσεις γίνονται εκρηκτικότερες και σαν αποτέλεσμα η μέση χρησιμοποίηση μειώνεται.

4.2. Μελέτη αλγορίθμων για τη φάση γρήγορης σύγκλισης

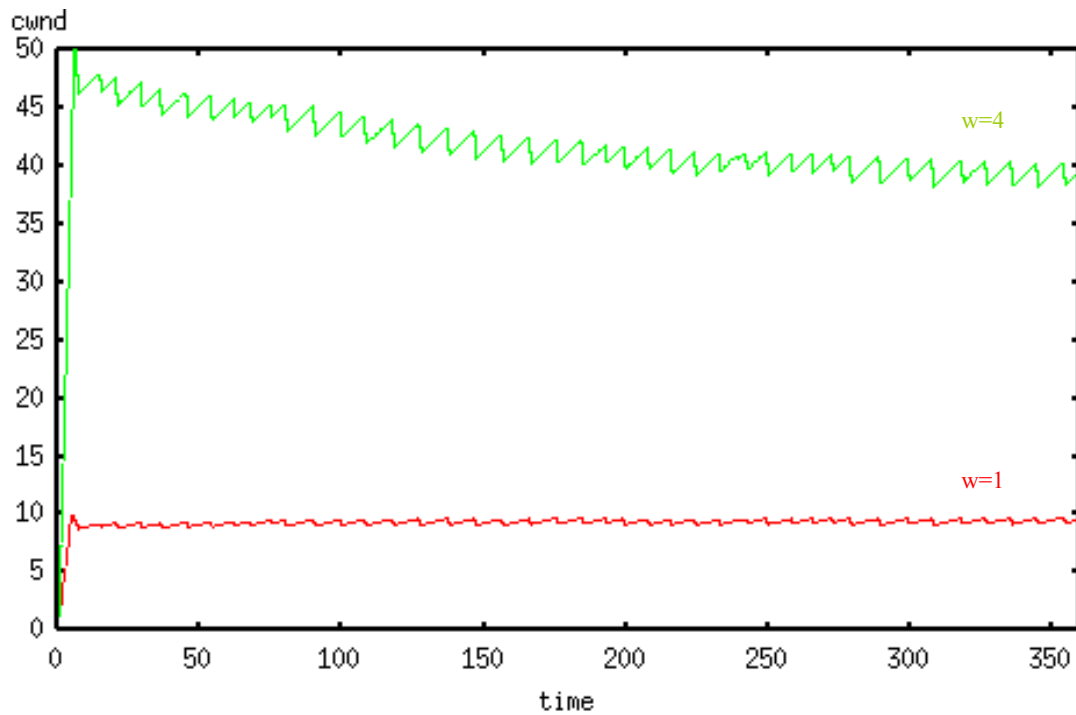
4.2.1. Ταχύτητα σύγκλισης και διαφοροποίηση

Στην ενότητα αυτή θα παρουσιάσουμε τη συμπεριφορά των αλγορίθμων που προτείναμε για γρήγορη σύγκλιση σε συνδυασμό με τον αλγόριθμο WTP. Στο Σχήμα 19, παρουσιάζονται οι τιμές του παραθύρου συμφόρησης συνδέσεων WTP με διαφορετική επιθυμία χρέωσης, που χρησιμοποιούν ως αλγόριθμο γρήγορης σύγκλισης το Slow start του TCP.

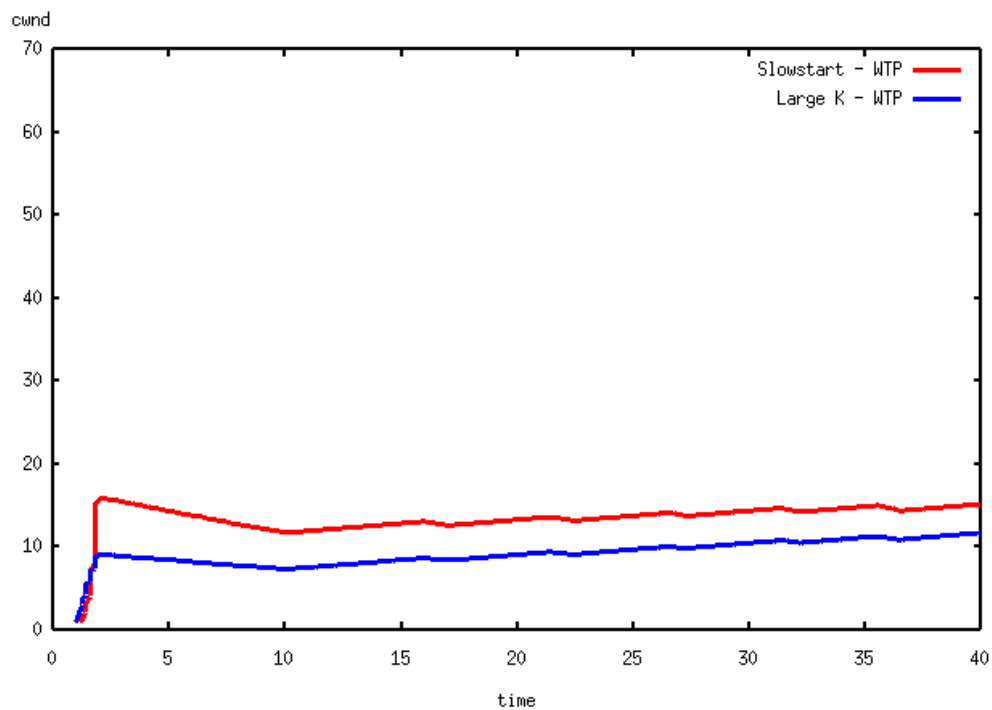


Σχήμα 19 Σύγκλιση συνδέσεων WTP με τη χρήση του μηχανισμού Slow start

Πιο συγκεκριμένα από το Σχήμα 19, παρατηρούμε ότι μπορεί να υπάρξουν περιπτώσεις κατά τις οποίες, στο τέλος της διάρκειας εφαρμογής του μηχανισμού Slow start (δηλαδή όταν λήφθηκε μαρκαρισμένο πακέτο) οι τιμές του παραθύρου συμφόρησης για συνδέσεις με μικρότερη επιθυμία χρέωσης, να είναι μεγαλύτερες από αυτές των παραθύρων συμφόρησης συνδέσεων με μεγαλύτερη επιθυμία χρέωσης. Το αποτέλεσμα του γεγονότος αυτού είναι ότι ξεκινώντας τη φάση αποφυγής της συμφόρησης, με αυτές τις τιμές, οι συνδέσεις παρουσίαζαν μεγάλους χρόνους σύγκλισης στο σημείο ισορροπίας, αφού οι τιμές των παραθύρων συμφόρησης αποκλίνουν κατά πολύ από αυτό.



Σχήμα 20 Σύγκλιση συνδέσεων WTP με τη χρήση του μηχανισμού Large K



Σχήμα 21 Σύγκλιση συνδέσεων WTP για τους μηχανισμούς Slow start και Large K

Το γεγονός αυτό μπορεί να εξηγηθεί εύκολα, αν το συνδυάσουμε με την λογική ισότιμης κατανομής του εύρους ζώνης, που ακολουθείται από τους μηχανισμούς του TCP. Αναλυτικότερα από τη στιγμή που ο μηχανισμός Slow start έχει την ίδια συμπεριφορά για συνδέσεις με μικρά και μεγάλα βάρη (αυξάνοντας και στις δύο περιπτώσεις το παράθυρο συμφόρησης κατά ένα για κάθε επιβεβαίωση λήψης), είναι αναμενόμενο να μην υπάρχει αναλογική δικαιοσύνη στο ρυθμό που απολαμβάνουν οι συνδέσεις με αποτέλεσμα οι τιμές παραθύρου συμφόρησης που αποκτούνται να μη είναι ανάλογες της επιθυμίας χρέωσης των συνδέσεων.

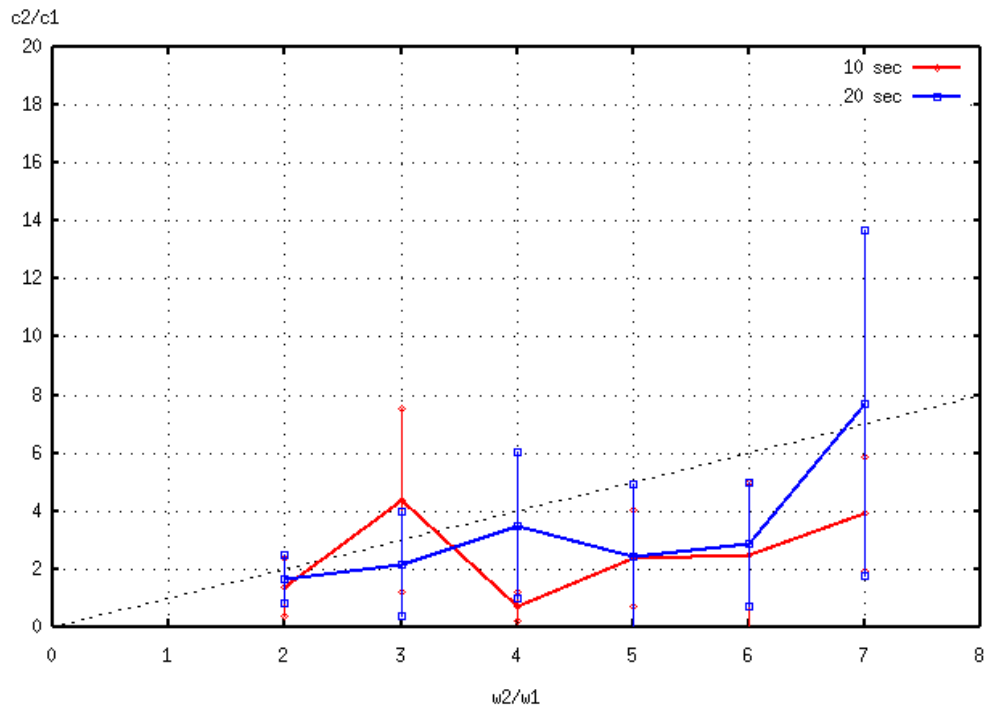
Ανάλογες συμπεριφορές με εκείνες του Σχήματος 19 συναντήσαμε και στην περίπτωση χρησιμοποίησης του μηχανισμού weighted Slow start, που περιγράψαμε στο Κεφάλαιο 2. Πιο συγκεκριμένα η διαφοροποίηση του μηχανισμού αυτού σε σχέση με το Slow start, έγκειται στο γεγονός ότι οι συνδέσεις με μεγάλα βάρη συνεχίζουν το μηχανισμό αυτό και μετά από την κατάσταση συμφόρησης στο δίκτυο, μεγαλώνοντας με τον εκθετικό ρυθμό του Slow start, το παράθυρο συμφόρησης τους, σε μία τιμή ανάλογη του βάρους τους. Παρόλα αυτά, ο μηχανισμός συνεχίζει να είναι αυτός του Slow start με τη μόνη διαφορά ότι οι συνδέσεις με μεγαλύτερα βάρη το χρησιμοποιούν λίγο περισσότερο από αυτές με μικρότερα βάρη. Έτσι και στη περίπτωση του weighted Slow start παρουσιάζονται καταστάσεις ανάλογες με αυτές του απλού Slow start.

Αντίθετα η χρησιμοποίηση του μηχανισμού Large K για γρήγορη σύγκλιση των συνδέσεων, οδηγεί το μηχανισμό WTP σε καλύτερη απόδοση, όσο αφορά στην επιτευξιμότητα διαφοροποίησης υπηρεσιών που προσφέρει. Αναλυτικότερα, επειδή η λογική της αναλογικής δικαιοσύνης (proportional fairness) ισχύει στον αλγόριθμο Large K (το παράθυρο συμφόρησης αυξάνει ανάλογα της τιμής επιθυμίας χρέωσης που έχει δηλώσει ο χρήστης), ο ρυθμός αποστολής των συνδέσεων είναι πάντοτε ανάλογος της επιθυμίας χρέωσης αυτών (βλ. Σχήμα 20). Έτσι αποφεύγονται οι καταστάσεις που συναντήσαμε με τους άλλους δύο αλγόριθμους.

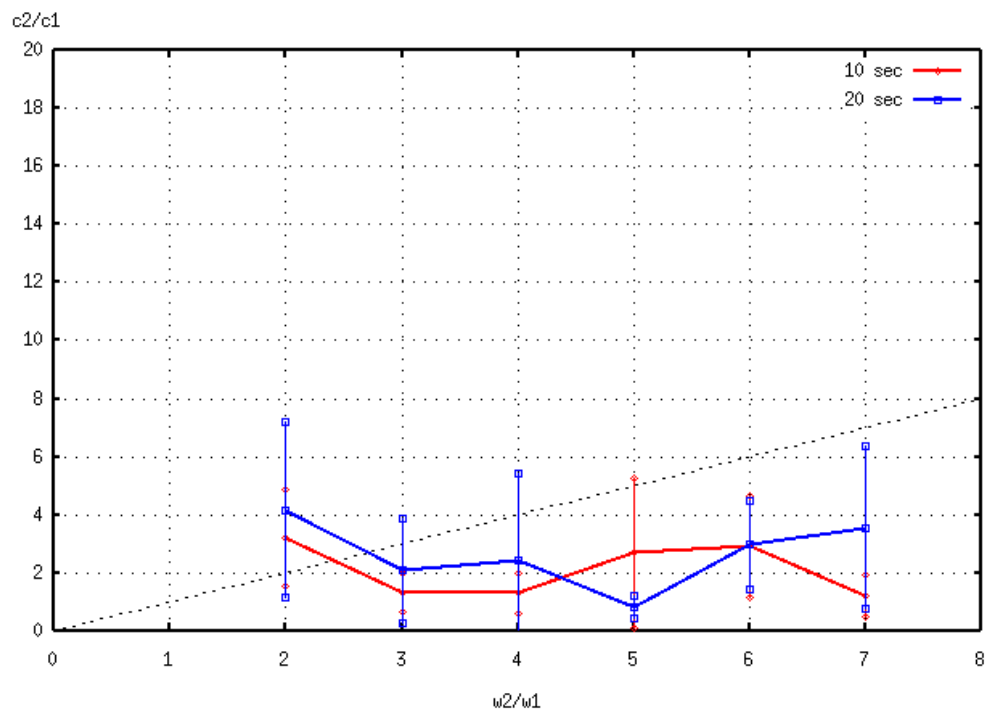
Στο Σχήμα 21, παρατηρούμε τις τιμές των παραθύρων συμφόρησης δύο συνδέσεων WTP οι οποίες χρησιμοποιούν τους αλγόριθμους Slow start και Large K για γρήγορη σύγκλιση. Παρατηρούμε ότι και στη περίπτωση του Large K παρουσιάζεται ανάλογη αύξηση των παραθύρου συμφόρησης, με αυτής του Slow start (σχεδόν συγκλίνουν την ίδια χρονική στιγμή), αποδεικνύοντας με αυτόν τον τρόπο την επιτευξιμότητα του δεύτερου αλγορίθμου όσο αφορά τη γρήγορη σύγκλιση των συνδέσεων.

4.2.2. Διαφοροποίηση ως προς την επιθυμία χρέωσης

Στην ενότητα αυτή θα παρουσιάσουμε τα πειραματικά αποτελέσματα που αφορούν στη διαφοροποίηση υπηρεσίας που παρουσιάζουν οι αλγόριθμοι για γρήγορη σύγκλιση που μελετάμε.

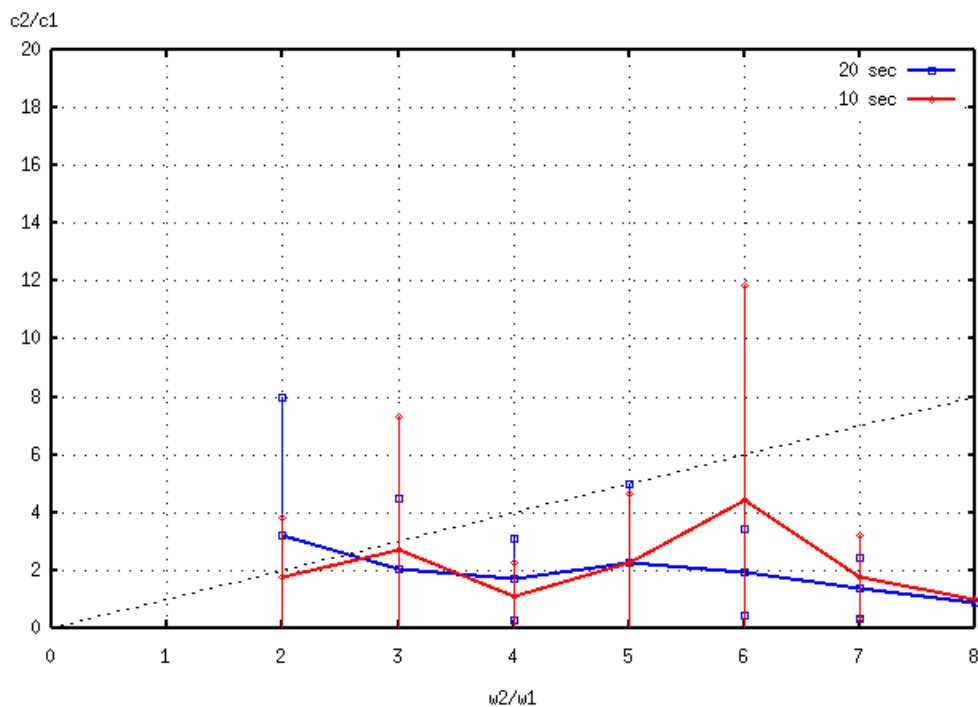


Σχήμα 22 Λόγος της διαπερατότητας συνδέσεων WTP με χρήση μηχανισμού γρήγορης σύγκλισης Slow start



Σχήμα 23 Λόγος της διαπερατότητας συνδέσεων WTP με χρήση μηχανισμού γρήγορης σύγκλισης weighted Slow start

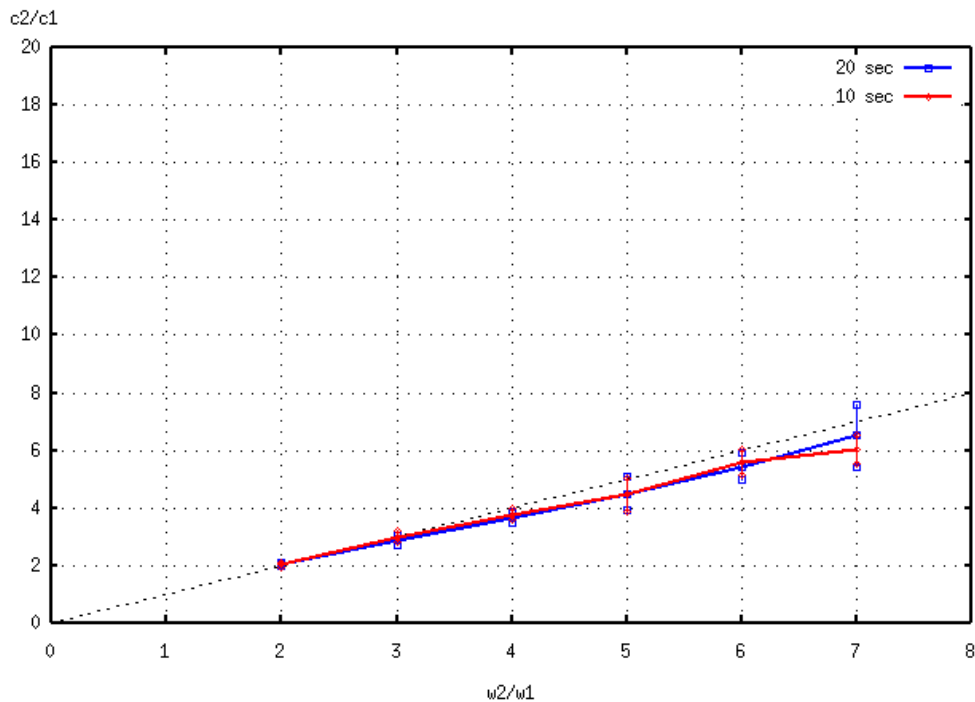
Στο Σχήμα 22, βλέπουμε το λόγο των παράθυρων συμφόρησης ως συνάρτηση του λόγου των τιμών επιθυμίας χρέωσης για συνδέσεις WTP που χρησιμοποιούν ως μηχανισμό για γρήγορη σύγκλιση το Slow start. Τα αποτελέσματα αφορούν συνδέσεις που είχαν μέση διάρκεια 10 και 20 δευτερόλεπτα. Παρατηρούμε, ότι στην περίπτωση του Slow start δεν παρουσιάζεται καμία αναλογία μεταξύ του ρυθμού αποστολής μίας σύνδεσης και της τιμής επιθυμίας χρέωσης της σύνδεσης αυτής. Το γεγονός αυτό οφείλεται, όπως έχουμε εξηγήσει στην προηγούμενη ενότητα, ότι σε μηχανισμούς όπως ο Slow start δεν υπάρχει αναλογική δικαιοσύνη (proportional fairness) στις συνδέσεις. Στο Σχήμα 23, παρουσιάζεται ο λόγος των παραθύρων συμφόρησης ως συνάρτηση του λόγου των τιμών επιθυμίας χρέωσης για συνδέσεις WTP που χρησιμοποιούν ως μηχανισμό για γρήγορη σύγκλιση το weighted Slow start. Ήταν αναμενόμενο ότι, όπως και στην περίπτωση του Slow start, δεν υπάρχει καμία αντιστοιχία μεταξύ του ρυθμού αποστολής των συνδέσεων και των τιμών επιθυμίας χρέωσης αυτών.



Σχήμα 24 Λόγος της διαπερατότητας συνδέσεων WTP με χρήση μηχανισμού γρήγορης σύγκλισης MultTCP Slow start

Παράλληλα, στο Σχήμα 24, βλέπουμε ότι ο λόγος της διαπερατότητας συνδέσεων WTP, που χρησιμοποιούν το μηχανισμό MulTCP Slow start για γρήγορη σύγκλιση, δεν είναι ανάλογος του λόγου επιθυμίας χρέωσης. Και σε αυτή την περίπτωση, ο μηχανισμός που χρησιμοποιείται δεν διαφοροποιεί τις συνδέσεις με κάποιο τρόπο αναλογικής δικαιοσύνης (βέβαια ο μηχανισμός αποφυγής της συμφόρησης που χρησιμοποιεί ο MulTCP, προσφέρει αναλογική δικαιοσύνη).

Αντίθετα όπως φαίνεται στο Σχήμα 25, οι λόγοι διαπερατότητας των συνδέσεων WTP που χρησιμοποιούν το μηχανισμό Large K, για γρήγορη σύγκλιση, είναι ανάλογοι της επιθυμίας χρέωσης. Το αποτέλεσμα ήταν και το αναμενόμενο αφού όπως αναφέραμε και στην προηγούμενη ενότητα ο μηχανισμός Large K, υποστηρίζει αναλογική δικαιοσύνη.



Σχήμα 25 Λόγος της διαπερατότητας συνδέσεων WTP με χρήση μηχανισμού γρήγορης σύγκλισης Large K

4.2.3. Χρόνοι σύγκλισης

Ο πίνακας του Σχήματος 26, παρουσιάζει το μέσο χρόνο διαφορετικών αλγόριθμων σύγκλισης συνδέσεων WTP. Σαν μέσο χρόνο σύγκλισης μίας σύνδεσης, θεωρούμε τη χρονική διάρκεια από την έναρξη αποστολής δεδομένων μέχρι τη στιγμή που το παράθυρο συμφόρησης παίρνει τιμές σε διάστημα που η απόκλιση από τη μέση τιμή του παράθυρου συμφόρησης, είναι α ($\alpha=10\%$).

Τα αποτελέσματα του πίνακα στο Σχήμα 26, δείχνουν το μέσο όρο δέκα ανεξάρτητων επαναλήψεων του ίδιου πειράματος και αποτελούν το μέσο όρο. Στο πείραμα έλαβαν μέρος δέκα συνδέσεις WTP, εκ των οποίων οι πέντε χρησιμοποίησαν για τη γρήγορη σύγκλισή τους, το μηχανισμό Slow start και οι υπόλοιπες το Large K.

Παρατηρούμε ότι οι χρόνοι σύγκλισης για το μηχανισμό Slow start είναι πάντοτε μεγαλύτεροι από αυτούς του Large K, γεγονός κατανοητό αφού ο Slow start δεν αυξάνει το παράθυρο συμφόρησης ανάλογα της επιθυμίας χρέωσης όπως γίνεται με τον Large K. Έτσι, στην περίπτωση του Slow start οι συνδέσεις, μπορεί να αποκτήσουν τιμές του παράθυρου συμφόρησης πολύ μεγαλύτερες ή πολύ μικρότερες από αυτές του που παίρνει το παράθυρο συμφόρησης στο σημείο ισορροπίας. Αυτό όμως που φαίνεται πιο ενδιαφέρον, είναι το γεγονός ότι καθώς μεγαλώνουν οι τιμές των επιθυμιών χρέωσης η διαφορά των χρόνων μεταξύ των δύο αλγόριθμων μειώνεται. Εικάζουμε, ότι το γεγονός αυτό οφείλεται στο ότι, όσο μεγαλώνουν οι τιμές των επιθυμιών χρέωσης των συνδέσεων, στην περίπτωση του Large K, τόσο μεγαλύτερες είναι οι τιμές του παράθυρου συμφόρησης στην ισορροπία, οπότε χρειάζεται και περισσότερο χρόνο ο Large K να αυξήσει το παράθυρο συμφόρησης σε αυτές τις τιμές.

Χρόνος Σύγκλισης ($\alpha = 0.5$)	w=1	w=4
Large K	1.74	18.62
Slowstart	34.81	22.95

Σχήμα 26 Χρόνοι σύγκλισης

5. Συμπεράσματα και μελλοντική εργασία

Στα πλαίσια της παρούσας μεταπτυχιακής εργασίας, ερευνήσαμε, χρησιμοποιώντας προσομοίωση αλλά και πραγματική υλοποίηση, αλγόριθμους διαφοροποίησης υπηρεσίας τόσο για τη φάση αποφυγής της συμφόρησης, όσο και για τη φάση γρήγορης σύγκλισης, συνδέσεων Internet⁷.

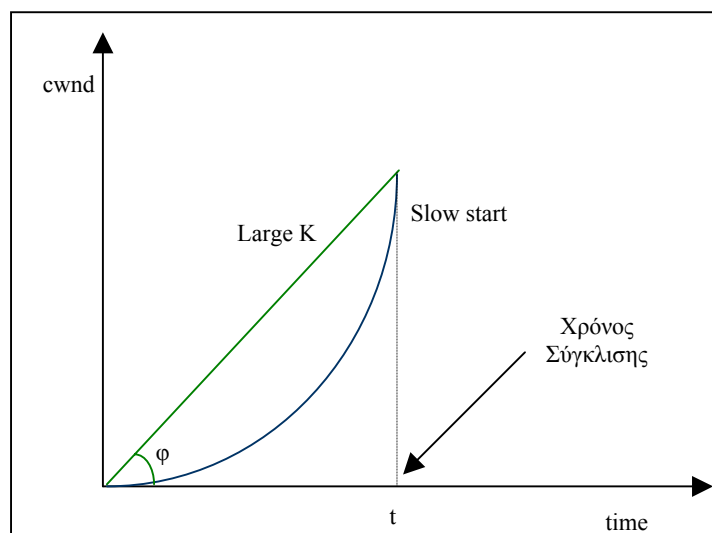
Πιο συγκεκριμένα στην περιοχή αλγορίθμων αποφυγής της συμφόρησης (Congestion Avoidance), διερευνήσαμε τη διαφοροποίηση υπηρεσίας και την επίδοση, που πετυχαίνεται από ένα αλγόριθμο ελέγχου συμφόρησης (WTP), ο οποίος στηρίζεται σε τεχνικές παραθύρου (window based) όπως και το TCP και ο οποίος παρέχει διαφοροποίηση στο ρυθμό αποστολής των χρηστών με τη χρήση βαρών. Τα πειραματικά μας αποτελέσματα έδειξαν ότι για δίκτυα στα οποία δεν παρουσιάζονται απώλειες πακέτων και υποστηρίζουν τον μηχανισμό Explicit Congestion Notification (ECN), ο αλγόριθμος WTP μπορεί να είναι δίκαιος και να οδηγεί σε μεγαλύτερη χρησιμοποίηση του δικτύου, σε σχέση με το MulTCP, ο οποίος είναι ένας άλλος αλγόριθμος που στηρίζεται στο TCP και παρέχει διαφοροποίηση υπηρεσίας. Οι αλγόριθμοι μαρκαρίσματος που ερευνήθηκαν είναι ο RED, Virtual Queue και Load Based Marking. Όσον αφορά τους αλγόριθμους αυτούς, τα αποτελέσματα των ερευνών μας έδειξαν, ότι η διαφοροποίηση υπηρεσίας στις ουρές των δρομολογητών μπορεί να είναι χειρότερες για τον αλγόριθμο Virtual Queue, όπου το μαρκάρισμα των πακέτων προκύπτει σε χρονικές κλίμακες υπερχειλίσης ενός ιδεατού ενταμιευτή, σε αντιδιαστολή με τους αλγόριθμους RED και Load Based Marking όπου η πιθανότητα μαρκαρίσματος εξαρτάται σε μέσες τιμές μεγεθών (στην περίπτωση του RED αναφερόμαστε στο μέσο μέγεθος ουράς, ενώ στην περίπτωση του Load Based Marking στο φόρτο κίνησης). Από την άλλη μεριά, ο αλγόριθμος Virtual Queue μπορεί και διαφοροποιεί τις ροές βάσει της εκρηκτικότητας τους.

Παράλληλα, μελετήσαμε αλγόριθμους για τη φάση γρήγορης σύγκλισης των συνδέσεων που χρησιμοποιούν τον αλγόριθμο WTP, διαπιστώνοντας ότι μηχανισμοί παρόμοιοι του Slow start του TCP δεν επέφεραν αναλογική δικαιοσύνη όσον αφορά τη διαφοροποίηση υπηρεσίας που προσφέρεται σε συνδέσεις μικρής διάρκειας. Αντίθετα παρατηρήσαμε ότι ο ίδιος ο αλγόριθμος WTP μπορεί να χρησιμοποιηθεί ως αλγόριθμος για τη γρήγορη σύγκλιση συνδέσεων (την παραμετροποίηση αυτή του WTP για γρήγορη σύγκλιση την ονομάσαμε, Large K), χρησιμοποιώντας μεγάλες

⁷ Μέρος της μεταπτυχιακής εργασίας πραγματοποιήθηκε στα πλαίσια του έργου της Ευρωπαϊκής Ένωσης Market Managed Multiservice Internet – M3I (<http://www.m3i.org>). Επίσης μέρος της παρούσας εργασίας αποτελούν οι δημοσιεύσεις [20] και [22].

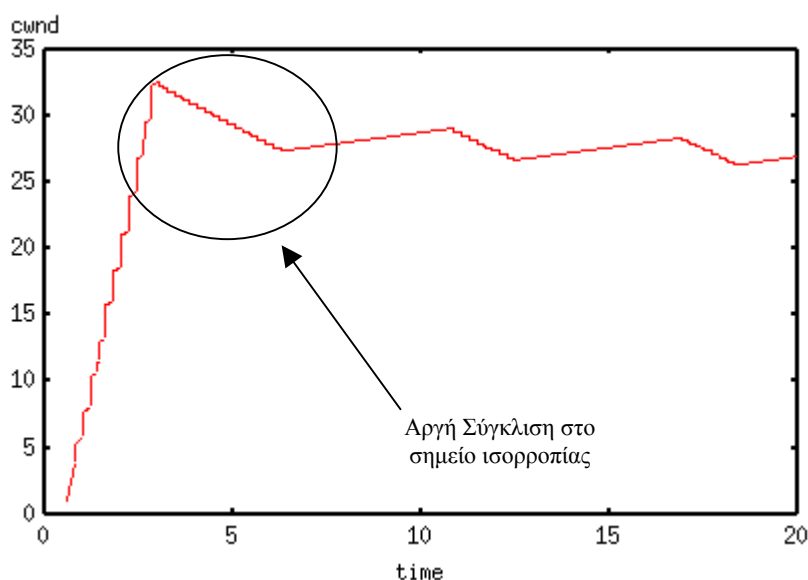
τιμές του παράγοντα \bar{k} , ο οποίος είναι υπεύθυνος για τη σύγκλιση και σταθερότητα του αλγόριθμου. Από τα πειραματικά μας αποτελέσματα, διαπιστώσαμε ότι η χρήση του Large K, εκτός από τη γρήγορη σύγκλιση των συνδέσεων, προσέφερε διαφοροποίηση υπηρεσίας ανάλογα την επιθυμία χρέωσης των χρηστών.

Από την άλλη μεριά, ένα θέμα που προκύπτει από τη λειτουργία του αλγόριθμου Large K, για μελλοντική διερεύνηση είναι η επιλογή της τιμής του παράγοντα \bar{k} , ώστε να έχουμε ταχύτητα σύγκλισης ανάλογης του Slow start του TCP. Όπως διαπιστώσαμε και στα πειράματα, η επιλογή της τιμής του \bar{k} , εξαρτάται από τα χαρακτηριστικά του δικτύου κάθε φορά. Πιο συγκεκριμένα, όπως φαίνεται στο παρακάτω σχήμα, η αύξηση του παραθύρου συμφόρησης στην περίπτωση του Slow start παρουσιάζει εκθετική μορφή. Από την άλλη μεριά, η αύξηση του παραθύρου συμφόρησης στην περίπτωση του Large K είναι γραμμική και επομένως, για να μπορούμε να πούμε ότι οι αλγόριθμοι Slow start και Large K συγκλίνουν με την ίδια ταχύτητα, πρέπει η γωνία φ (Σχήμα 27), η οποία εξαρτάται από το παράγοντα \bar{k} , να έχει κατάλληλη τιμή ώστε τη χρονική στιγμή t , που θεωρούμε ότι οι αλγόριθμοι συγκλίνουν, το παράθυρο συμφόρησης και για τους δύο αλγόριθμους να είναι το ίδιο. Παρόλα αυτά, ο χρόνος σύγκλισης t , μίας σύνδεσης εξαρτάται, κάθε φορά, από τα χαρακτηριστικά του δικτύου (π.χ. καθυστέρηση, εύρος ζώνης κτλ), με αποτέλεσμα η τιμή του παράγοντα \bar{k} του Large K να χρειάζεται να επιλέγεται κάθε φορά, σε αντίθεση με το μηχανισμό Slow start ο οποίος είναι ο ίδιος για κάθε είδος δικτύου.



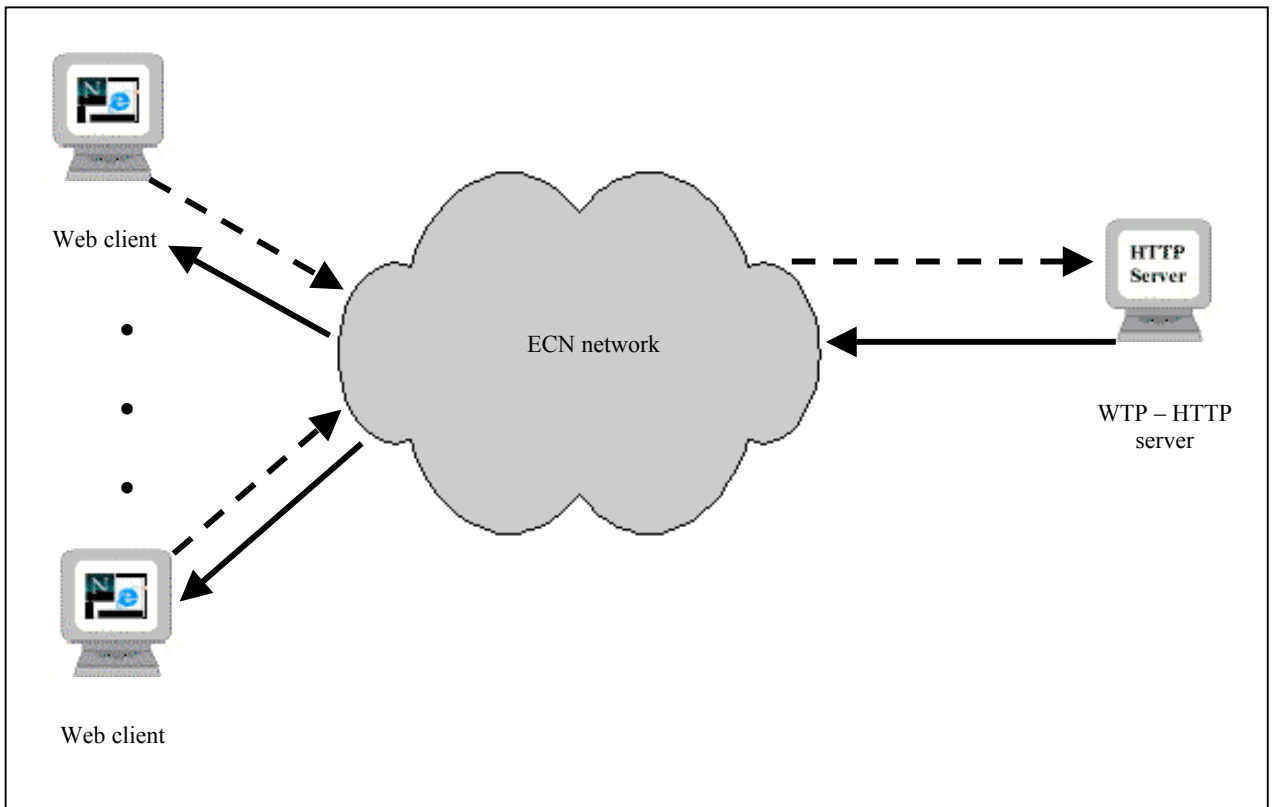
Σχήμα 27 Αναπαράσταση σύγκλισης για τους αλγόριθμους Slow start και Large K

Επίσης ένα άλλο θέμα μελλοντικής διερεύνησης για το μηχανισμό Large K, είναι η ανάγκη ύπαρξης ενδιάμεσων τιμών του παράγοντα \bar{k} , κατά τη μετάβαση των συνδέσεων από τη φάση της γρήγορης σύγκλισης, στη φάση αποφυγής της συμφόρησης (Σχήμα 28). Πιο συγκεκριμένα, από τα πειραματικά αποτελέσματα που αφορούσαν στο μηχανισμό Large K, διαπιστώσαμε κατά την μετάβαση από τη φάση γρήγορης σύγκλισης στη φάση αποφυγής της συμφόρησης, το παράθυρο συμφόρησης παρουσίαζε αργή προσαρμογή στο σημείο ισορροπίας. Αυτό οφείλεται στο γεγονός ότι κατά τη διαπίστωση συμφόρησης (με τη λήψη μαρκαρισμένης επιβεβαίωσης λήψης), χρησιμοποιούνται αυτόματα μικρές τιμές του \bar{k} , ώστε να διασφαλίζεται η σύγκλιση των συνδέσεων. Έτσι ένα θέμα μελλοντικής διερεύνησης θα ήταν η επέκταση του αλγόριθμου Large K ώστε η μετάβαση από τη φάση της γρήγορης σύγκλισης στη φάση αποφυγής της συμφόρησης να γίνεται με ενδιάμεσες τιμές του παράγοντα \bar{k} .



Σχήμα 28 Παράδειγμα αργής σύγκλισης στο σημείο ισορροπίας κατά τη μετάβαση της σύνδεσης από τη φάση γρήγορης σύγκλισης στη φάση αποφυγής της συμφόρησης για το μηχανισμό Large K.

Από τη μελέτη των μηχανισμών που αναλύσαμε στα πλαίσια της μεταπτυχιακής μας εργασίας, προέκυψαν αρκετά ενδιαφέροντα θέματα για μελλοντική εργασία. Για παράδειγμα, η χρήση του μηχανισμού ECN, ώστε να γίνεται αντιληπτή η συμφόρηση στο δίκτυο χωρίς να χρειάζεται να υπάρξει απώλεια πακέτων, θα μπορούσε να εφαρμοστεί επιτυχημένα στα ασύρματα δίκτυα. Στη περίπτωση των ασύρματων δικτύων, κάποια απώλεια πακέτου δεν είναι απαραίτητα απόρροια κατάστασης συμφόρησης αλλά πιθανής εμφάνισης θορύβου κατά τη μετάδοση. Έτσι ένας μηχανισμός μαρκαρίσματος όπως ο ECN, θα αποτελούσε λύση στο παραπάνω πρόβλημα αφού θα γινόταν ξεκάθαρο, τότε θα υπήρχε συμφόρηση στο δίκτυο. Από τη στιγμή όμως που υιοθετηθεί ο μηχανισμός ECN σε ασύρματα δίκτυα, ανοίγει ο δρόμος εφαρμογής των αλγόριθμων που μελετήσαμε στα δίκτυα αυτού του είδους, καθιστώντας ικανή την προσφορά διαφοροποίησης υπηρεσίας και σε ασύρματα δίκτυα.



Σχήμα 29 Παράδειγμα αξιοποίησης αλγόριθμων διαφοροποίησης υπηρεσιών στο Internet

Ένα άλλο πεδίο που θα μπορούσε να αποτελέσει θέμα για μελλοντική εργασία, θα ήταν η σύγκριση των αλγόριθμων που μελετήσαμε με άλλους αλγόριθμους διαφοροποίησης υπηρεσιών στο Internet, όπως η αρχιτεκτονική DiffServ.

Αντικείμενο μελλοντικής εργασίας, είναι και η ανάπτυξη μίας εφαρμογής επίδειξης διαφοροποιημένων υπηρεσιών, που βασίζεται στους αλγόριθμους που μελετήσαμε. Για παράδειγμα μία τέτοια εφαρμογή (Σχήμα 29) θα μπορούσε να είναι ένας εξυπηρετητής Internet (web browser), ο οποίος θα διατηρούσε κάποιο προφίλ των χρηστών που θα είχαν πρόσβαση σε αυτόν και θα διαφοροποιούσε τις υπηρεσίες του ανάλογα την επιθυμία χρέωσης που θα του είχαν δηλώσει οι χρήστες. Πιο συγκεκριμένα κάθε χρήστης δηλώνει την επιθυμία χρέωσης στον εξυπηρετητή Internet (διακεκομμένες γραμμές στο σχήμα), διαμορφώνοντας ένα προφίλ. Έτσι κάθε φορά, που υπάρχει συμφόρηση στο δίκτυο, ο εξυπηρετητής στέλνει δεδομένα σε κάθε χρήστη ανάλογα με την επιθυμία χρέωσης που έχει δηλωθεί στο προφίλ του.

Μία εναλλακτική εφαρμογή, επίδειξης παροχής διαφοροποιημένης υπηρεσίας, με τη χρήση των αλγόριθμων που μελετήσαμε, θα μπορούσε να είναι ένας εξυπηρετητής video on demand. Στην περίπτωση αυτή, στη θέση του εξυπηρετητή Internet του Σχήματος 26, θα υπήρχε ένας εξυπηρετητής που ανάλογα με την επιθυμία χρέωσης των χρηστών θα έστελνε ροές video (video streams) με διαφορετικό ρυθμό. Έτσι σε περιπτώσεις συμφόρησης, οι χρήστες με μεγαλύτερες τιμές επιθυμίας χρέωσης, θα μπορούσαν να απολαμβάνουν καλύτερη ποιότητα σε εικόνα και ήχο σε σχέση με τους χρήστες που έχουν μικρότερη επιθυμία χρέωσης, αφού μεγαλύτερες τιμές επιθυμίας χρέωσης σημαίνουν και μεγαλύτερη διαπερατότητα για τις συνδέσεις. Σε μία τέτοια περίπτωση, όμως υπάρχει ένας επιπλέον περιορισμός, ο οποίος καθορίζεται από την ελαστικότητα σε καθυστερήσεις που παρουσιάζουν οι ροές video. Πιο συγκεκριμένα, οι ροές πραγματικού χρόνου, όπως οι ροές video, παρουσιάζουν πολύ μικρή ελαστικότητα όσον αφορά την καθυστέρηση που συναντούν τα πακέτα στο δίκτυο. Έτσι στην περίπτωση, του σεναρίου παροχής διαφορετικής ποιότητας υπηρεσίας video, το δίκτυο πρέπει να ικανοποιεί τους περιορισμούς σε καθυστέρηση που θέτουν οι ροές video, ρυθμίζοντας τις παραμέτρους των αλγόριθμων μαρκαρίσματος ώστε να τηρούνται οι περιορισμοί αυτοί. Σε μια τέτοια περίπτωση, όμως, όλες οι ροές που περνούν από τους δρομολογητές, από όπου προωθούνται τα πακέτα των ροών πραγματικού χρόνου, θα αντιμετωπίζουν την ίδια μικρές καθυστερήσεις, παρόλο που μερικές από αυτές μπορεί να είναι αρκετά ελαστικές (π.χ. συνδέσεις FTP) με αποτέλεσμα να υπάρχουν περιπτώσεις που να μη γίνεται καλή χρησιμοποίηση των πόρων του δικτύου.

6. Αναφορές

- [1] R. J. Gibbens, F. P. Kelly. Resource pricing and congestion control. *Automatica*, 39:1969-1985, 1999.
- [2] F. P. Kelly. Models for a self-managed Internet. *European Transactions on Telecommunications*, 8:33-37, January 1997.
- [3] P. B. Key, D. R. McAuley. Differential QoS and pricing in network: Where flow control meets game theory. *IEE Proceedings Software*, 146(2):39-43, March 1999.
- [4] K. K. Ramakrishnan, S. Floyd. An addition of Explicit Congestion Notification (ECN) to IP. *RFC 3168*, September 2001.
- [5] S. Floyd. TCP and explicit congestion notification. *ACM Computer Communication review*, 24(5):10-23, 1994.
- [6] S. Floyd, V. Jacobson. Random early detection gateways for congestion avoidance. *IEEE/ACM Transactions on Networking*, 1(4):397-413, 1993.
- [7] J. Crowcroft, P. Oechslin. Differentiated end-to-end services using a weighted proportional fair share TCP. *Computer Communication Review*, 28(3):53-69, 1998.
- [8] R. J. Gibbens, P. B. Key. A note on resource pricing and congestion control. *Automatica*, 1999.
- [9] V. Jacobson. Congestion Avoidance and control. *In Proceedings of SIGCOMM '88*, (Standford, CA, August 1988), ACM.
- [10] K. K. Ramakrishnan, D. M. Chiu. Congestion avoidance in computer networks with connectionless network layer. *Tech. Rep. DEC-TR-506*, Digital Equipment Corporation, August 1987.
- [11] L. Kleinrock. *Queuing Systems. Vol II. John Wiley & Sons*, 1976.
- [12] F. P. Kelly. Charging and rate control for elastic traffic. *European Transactions on Telecommunications*, 8:33-37, January 1997.
- [13] F. P. Kelly, A. Mauloo, D. Tan. Rate control in communication networks: shadow prices, proportional fairness and stability. *Journal of the Operational Research Society*, 49, 1998.
- [14] J. M. Harrison. *Brownian Motion and Stochastic Flow Systems. Krieger*, 1985.

- [15] <http://www.freebsd.org>
- [16] L. Rizzo. Dummynet: a simple approach to the evaluation of network protocols. *ACM Computer Communication Review*, 27:31-41, 1997.
- [17] P. Key, D. McAuley, P. Barham, K. Laevens. Congestion pricing for congestion avoidance. *Microsoft Research report MSR-TR-9-15*.
- [18] UCB/LBNL/VINT Network Simulator – ns (version 2). <http://www-mash.cs.berkeley.edu/ns/ns.html>
- [19] K. Cho. A framework for Alternate Queuing: Towards traffic management by PC-Unix based routers. *In Proceedings of USENIX 1998 Annual Technical Conference*, June 1998.
- [20] P. Antoniadis, C. Courcoubetis, G. Margetis, V. Siris, G. Stamoulis. Efficient adaptation to dynamic pricing communicated by ECN marks: Scenarios for experimental assessment. *SPIE International Symposium on Information Technologies 2000 (Program on Internet Performance and Control)*, Boston, USA, November 2000.
- [21] S. Floyd. Recommendation on using the “gentle_” variant of RED. 2000, <http://www.aciri.org/floyd/red/gentle.html>
- [22] V. Siris, C. Courcoubetis, G. Margetis. Service differentiation in ECN networks using weighted window-based congestion control for various packet marking algorithms. *QoIS'01, 2nd International Workshop on Quality of future Internet Services*, 2001.
- [23] M. Mathis, J. Semke, J. Mahdavi, T. Ott. The macroscopic behavior of the TCP congestion avoidance algorithm. *Computer Communications Review*, 27(3), 1997.
- [24] T. Ott, J. Kemperman, and M. Mathis. The stationary behavior of ideal TCP congestion avoidance. *Pre-print*, 1997.
- [25] J. Padhye, V. Firoiu, D. Towsley, J. Kurose. Modeling TCP throughput: A simple model and its empirical validation. *In Computer Communication Review, Proceedings Sigcomm 98*, 28:303-315, 1998.
- [26] K. Fall, S. Floyd. Simulation-based Comparisons of Tahoe, Reno, and Sack TCP. *Computer Communication Review*, V. 26 N. 3, pp. 5-21, July 1996.
- [27] W. R. Stevens. TCP/IP Illustrated, Volume 1: The protocols. *Addison Wesley*, 1994.