



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

**Έλεγχος Συμφόρησης και
Ενεργή Διαχείριση Ουρών
στο Διαδίκτυο**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Σταμάτιος Π. Σταματάκης

Επιβλέπων : Βασίλειος Μάγκλαρης
Καθηγητής ΕΜΠ

Αθήνα, Οκτώβριος 2004



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

**Έλεγχος Συμφόρησης και
Ενεργή Διαχείριση Ουρών
στο Διαδίκτυο**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Σταμάτιος Π. Σταματάκης

Επιβλέπων : Βασίλειος Μάγκλαρης
Καθηγητής ΕΜΠ

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 19^η Οκτωβρίου 2004.

.....
Βασίλειος Μάγκλαρης
Καθηγητής ΕΜΠ

.....
Μιχαήλ Θεολόγου
Καθηγητής ΕΜΠ

.....
Ευστάθιος Συκάς
Καθηγητής ΕΜΠ

Αθήνα, Οκτώβριος 2004

.....
Σταμάτιος Π. Σταματάκης

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

Copyright © Σταμάτιος Π. Σταματάκης, 2004.

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

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

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

Περίληψη

Στόχος της παρούσας εργασίας είναι η ανάλυση και η αξιολόγηση των διαφόρων αλγορίθμων ελέγχου συμφόρησης που χρησιμοποιούνται στο διαδίκτυο σήμερα, καθώς και των αλγορίθμων ενεργής διαχείρισης ουρών που χρησιμοποιούνται από τους σημερινούς δρομολογητές ώστε να υποβοηθήσουν τον έλεγχο συμφόρησης.

Στο πρώτο μέρος γίνεται θεωρητική περιγραφή και ανάλυση δυο εκ των πιο γνωστών αλγορίθμων ελέγχου συμφόρησης, του TCP και του TFRC, καθώς και των τεχνικών ενεργής διαχείρισης ουρών RED, REM και RIO.

Στο δεύτερο γίνονται εκτενή πειράματα με τη χρήση του ειδικού προγράμματος προσομοίωσης δικτύων ns-2 network simulator ώστε να γίνει σύγκριση των ομοειδών αλγορίθμων και εκτίμηση της αλληλεπίδρασης των δυο τεχνικών σε ένα περιβάλλον με τη μορφή του διαδικτύου.

Λέξεις-κλειδιά: Έλεγχος Συμφόρησης, TCP, TFRC, Ενεργή Διαχείριση Ουρών, RED, REM, RIO

Abstract

The purpose of the present diploma thesis is the analysis and evaluation of the various congestion control algorithms that are used in today's internet, as well as the algorithms for active queue management that routers use as a mean to assist congestion control.

The first part consists of the theoretic description and analysis of the two most known congestion control algorithms, TCP and TFRC, and also of the active queue management algorithms RED, REM and RIO.

In the second one we conduct an extensive series of experiments using the specialized network simulator ns-2 in order to compare algorithms of the same type and evaluate the interaction of the two techniques in an internet-like environment.

Keywords: Congestion Control, TCP, TFRC, Active Queue Management, AQM, RED, REM, RIO

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή κ. Βασίλη Μάγκλαρη για τη δυνατότητα που μου έδωσε να ασχοληθώ με αυτή την εργασία. Επίσης θα ήθελα να ευχαριστήσω όλα τα μέλη του εργαστηρίου Διαχείρισης και Βέλτιστου σχεδιασμού δικτύων NETMODE για την υποστήριξη τους. Ιδιαίτερες ευχαριστίες πρέπει να δοθούν στον υποψήφιο διδάκτορα Δημήτρη Βελένη χωρίς τη βοήθεια του οποίου θα ήταν αδύνατο να ολοκληρωθεί αυτή η εργασία.

Στην οικογένειά μου.

Περιεχόμενα

Α. Θεωρητικό Μέρος

1. ΕΙΣΑΓΩΓΗ.....	11
2. ΈΛΕΓΧΟΣ ΣΥΜΦΟΡΗΣΗΣ	13
2.1 ΑΝΑΓΚΑΙΟΤΗΤΑ ΓΙΑ ΑΠΟ ΑΚΡΟ ΣΕ ΑΚΡΟ ΕΛΕΓΧΟ ΣΥΜΦΟΡΗΣΗΣ.....	13
2.2 ΈΛΕΓΧΟΣ ΣΥΜΦΟΡΗΣΗΣ ΤΟΥ TCP	15
2.2.1 Αργή εκκίνηση (<i>Slow Start</i>) και Αποφυγή συμφόρησης (<i>Congestion Avoidance</i>)	18
2.2.2. Γρήγορη Επανεκπομπή (<i>Fast Retransmit</i>) και Γρήγορη Ανάκαμψη (<i>Fast Recovery</i>).....	22
2.2.3 Η τροποποίηση <i>New Reno</i> στον αλγόριθμο της γρήγορης ανάκαμψης του <i>TCP</i>	24
2.3 Ο ΜΗΧΑΝΙΣΜΟΣ ΕΛΕΓΧΟΥ ΣΥΜΦΟΡΗΣΗΣ <i>TFRC</i>	27
2.3.1 Περιεχόμενα των πακέτων	28
2.3.2 Λειτουργία του αποστολέα.....	29
2.3.3 Λειτουργία του δέκτη	31
3. ΕΝΕΡΓΗ ΔΙΑΧΕΙΡΙΣΗ ΟΥΡΩΝ (ACTIVE QUEUE MANAGEMENT).....	35
3.1 ΑΝΑΓΚΑΙΟΤΗΤΑ ΓΙΑ ΤΗΝ ΥΠΑΡΞΗ ΑQM ΣΤΟΥΣ ΔΡΟΜΟΛΟΓΗΤΕΣ.	35
3.2 RANDOM EARLY DETECTION (RED)	37
3.3 RANDOM EXPONENTIAL MARKING (REM)	41
3.3.1 Συνάρτηση κόστους	42
3.3.2 Πιθανότητα απόρριψης	44
3.4 Ο ΑΛΓΟΡΙΘΜΟΣ RIO	46
3.4.1 Κατηγοριοποίηση πακέτων.....	47
3.4.2 Διαχείριση Ουράς.....	49

B. Πειραματικό Μέρος

1. ΓΕΝΙΚΑ.....	52
2. ΜΙΑ ΥΠΟ ΣΥΜΦΟΡΗΣΗ ΖΕΥΞΗ.....	54
2.1 ΔΙΚΑΙΟΣΥΝΗ ΜΕΤΑΞΥ ΤΩΝ ΡΟΩΝ (FAIRNESS).....	56
2.2 ΑΠΟΤΕΛΕΣΜΑΤΙΚΟΤΗΤΑ ΑΠΕΝΑΝΤΙ ΣΕ ΜΗ ΥΠΕΥΘΥΝΕΣ ΡΟΕΣ	62
2.3 ΜΕΣΟ ΜΗΚΟΣ ΟΥΡΑΣ ΚΑΙ ΚΑΘΥΣΤΕΡΗΣΗ.....	65
2.4 ΣΥΓΧΡΟΝΙΣΜΟΣ	67
3. ΕΚΡΗΚΤΙΚΗ ΚΙΝΗΣΗ.....	70
3.1 ΔΙΚΑΙΟΣΥΝΗ ΜΕΤΑΞΥ ΤΩΝ ΡΟΩΝ (FAIRNESS).....	72
3.2 ΜΕΣΟ ΜΗΚΟΣ ΟΥΡΑΣ	74
4. ΠΟΛΛΑΠΛΑ ΒΟΤΤΛΕΝΕΚ ΛΙΝΚΣ.....	77
4.1 ΑΠΟΥΣΙΑ BACKGROUND ΚΙΝΗΣΗΣ (ΔΙΑΦΟΡΟΠΟΙΗΣΗ ΤΩΝ ΡΟΩΝ ΜΕ ΒΑΣΗ ΤΟ RTT).....	80
4.1.1 Δικαιοσύνη μεταξύ των ροών (<i>fairness</i>)	80
4.1.2 Αποκλεισμός.....	85
4.2 ΎΠΑΡΞΗ BACKGROUND ΚΙΝΗΣΗΣ (ΔΙΑΦΟΡΟΠΟΙΗΣΗ ΤΩΝ ΡΟΩΝ ΩΣ ΠΡΟΣ ΤΟΝ ΑΡΙΘΜΟ ΤΩΝ ΥΠΟ ΣΥΜΦΟΡΗΣΗ ΖΕΥΞΕΩΝ ΑΠΟ ΤΙΣ ΟΠΟΙΕΣ ΔΙΕΡΧΟΝΤΑΙ).....	89
4.2.1 Δικαιοσύνη μεταξύ των ροών (<i>fairness</i>)	89
5. ΓΕΝΙΚΑ ΣΥΜΠΕΡΑΣΜΑΤΑ	93
ΠΑΡΑΡΤΗΜΑ : ΒΙΒΛΙΟΓΡΑΦΙΑ	95

Α. Θεωρητικό Μέρος

1. Εισαγωγή

Στην εργασία αυτή θα ασχοληθούμε με την ανάλυση και την αξιολόγηση των διαφόρων αλγορίθμων ελέγχου συμφόρησης που χρησιμοποιούνται στο διαδίκτυο σήμερα, καθώς και των αλγορίθμων ενεργής διαχείρισης ουρών που χρησιμοποιούνται από τους σημερινούς δρομολογητές.

Στα πρώτα βήματα του διαδικτύου η ανταλλαγή δεδομένων μεταξύ των λίγων τότε υπολογιστών που ήταν διασυνδεδεμένοι γινόταν χωρίς τους σημερινούς μηχανισμούς ελέγχου συμφόρησης. Για μικρό αριθμό χρηστών αυτή η τακτική ήταν αποδεκτή. Καθώς όμως το διαδίκτυο εξαπλωνόταν και όλο και περισσότεροι χρήστες προστίθονταν σε αυτό, η κίνηση αυξανόταν σε σημείο που οι φυσικοί πόροι του δικτύου δεν παρορούσαν να καλύψουν την προσφερόμενη κίνηση, είχαμε δηλαδή συμφόρηση (congestion).

Το ευρύτερα χρησιμοποιούμενο πρωτόκολλο μεταφοράς στο διαδίκτυο, το TCP (Transport Control Protocol), διαθέτει μηχανισμούς ελέγχου συμφόρησης που του επιτρέπουν να αντιδρά στη συμφόρηση. Συγκεκριμένα όταν αντιλαμβάνεται την απώλεια κάποιου πακέτου δεδομένων υποθέτει πως αυτή οφείλεται σε συμφόρηση και ξαναστέλνει τα χαμένα πακέτα, μειώνοντας και τον ρυθμό μετάδοσης του. Η εξέλιξη του πρωτοκόλλου όσον αφορά το κομμάτι του ελέγχου συμφόρησης είναι διαρκής, με πολλές εκδόσεις μέχρι σήμερα, ώστε να βελτιώνεται η απόδοση του παράλληλα με την αύξηση των χρηστών που σημαίνει και μεγαλύτερη τάση για εμφάνιση συμφόρησης.

Πολλές εφαρμογές όμως δεν υιοθετούν το TCP ως πρωτόκολλο μεταφοράς, είτε γιατί δεν χρειάζονται την αξιοπιστία που αυτό προσφέρει, είτε για άλλους λόγους όπως η απαίτηση σταθερού ρυθμού μετάδοσης. Χαρακτηριστικές κατηγορίες τέτοιων εφαρμογών είναι η μετάδοση σε πραγματικό χρόνο ήχου και βίντεο, οι διαδραστικές εφαρμογές πολυμέσων και οι εφαρμογές που απαιτούν ταυτόχρονη αποστολή σε πολλούς χρήστες (multicast). Και αυτές όμως θα πρέπει να υιοθετούν κάποιο μηχανισμό για να αντιδρούν σε καταστάσεις συμφόρησης ώστε να μην προκαλείται μείωση της απόδοσης του δικτύου αλλά και οι διαθέσιμοι πόροι να μοιράζονται δίκαια μεταξύ των διαφόρων συνδέσεων. Ένας αλγόριθμος ελέγχου συμφόρησης που τηρεί όλα τα παραπάνω είναι ο TFRC (TCP-Friendly Rate Control ή Φιλικός προς το

TCP Έλεγχος Ρυθμού), ο οποίος επιτυγχάνει σταθερότερο ρυθμό μετάδοσης σε σχέση με το TCP και σχετικά δίκαιη κατανομή του διαθέσιμου εύρους ζώνης.

Ο έλεγχος συμφόρησης που γίνεται στα άκρα του δικτύου μπορεί να υποβοηθηθεί και από το εσωτερικό του. Άλλωστε εκεί είναι που εμφανίζεται η συμφόρηση. Το ρόλο αυτό αναλαμβάνουν να επιτελέσουν οι αλγόριθμοι ενεργής διαχείρισης των ουρών στους ενδιάμεσους δρομολογητές του δικτύου. Αυτοί ανιχνεύουν την επικείμενη συμφόρηση και στέλνουν σήματα στα άκρα ώστε να μειώσουν το ρυθμό μετάδοσης τους.

Η συνδυασμένη χρήση ελέγχου συμφόρησης και ενεργής διαχείρισης ουρών μπορεί να προσφέρει υψηλή απόδοση. Ο τελικός χρήστης την αντιλαμβάνεται σε όρους ρυθμαπόδοσης και καθυστέρησης, ενώ στο εσωτερικό του δικτύου γίνεται αντιληπτή σε όρους συμφόρησης, μεγέθους ουρών και απωλειών πακέτων.

2. Έλεγχος συμφόρησης

2.1 Αναγκαιότητα για από άκρο σε άκρο έλεγχο συμφόρησης.

Η αρχιτεκτονική του διαδικτύου βασίζεται στο πρωτόκολλο IP. Τα δεδομένα μεταδίδονται ως πακέτα χωρίς να απαιτείται η εγκατάσταση σύνδεσης, χωρίς καμία εγγύηση παράδοσης και χωρίς έλεγχο του ρυθμού μετάδοσης. Οι λειτουργίες αυτές επαφίονται στα πρωτόκολλα μεταφοράς. Χωρίς έλεγχο του ρυθμού μετάδοσης είναι δυνατόν η προσφερόμενη κίνηση να υπερβεί τη διαθέσιμη χωρητικότητα. Σε αυτή την περίπτωση οι ενταμιευτές των δρομολογητών γεμίζουν και απορρίπτουν εισερχόμενα πακέτα. Η κατάσταση αυτή ονομάζεται συμφόρηση.

Παρατεταμένη συμφόρηση είναι δυνατό υπό συνθήκες να οδηγήσει το δίκτυο στην κατάσταση που είναι γνωστή ως «κατάρρευση λόγω συμφόρησης». Σε αυτή την περίπτωση αύξηση του προσφερομένου φορτίου στο δίκτυο οδηγεί σε μείωση του ωφέλιμου έργου που εκτελείται. Το φαινόμενο αυτό παρουσιάστηκε για πρώτη φορά κατά την πρώιμη φάση ανάπτυξης του διαδικτύου, στα μέσα της δεκαετίας του '80. Οφειλόταν στις άσκοπες επαναμεταδόσεις πακέτων που είχαν ήδη φτάσει στον τελικό τους προορισμό από την αρχική έκδοση του TCP .

Το πρωτόκολλο μεταφοράς TCP ελέγχει τη ροή δεδομένων μέσω ενός κυλιόμενου παραθύρου, ώστε να αποφεύγεται η υπερχειλίση των ενταμιευτών του παραλήπτη. Τα αποστελλόμενα πακέτα δεδομένων μπορεί να χαθούν είτε λόγω σφαλμάτων κατά τη μετάδοση είτε λόγω της ύπαρξης συμφόρησης. Οι απώλειες ανιχνεύονται μέσω ενός χρονομετρητή, η εκπνοή του οποίου σημαίνει ότι το πακέτο χάθηκε και πρέπει να αποσταλεί ξανά. Στην αρχική έκδοση του πρωτοκόλλου όμως δεν γινόταν προσαρμογή του παραθύρου ή του παραπάνω χρονομετρητή, ως απόκριση στη συμφόρηση. Αυτό είχε ως αποτέλεσμα πολλαπλές αναμεταδώσεις με την αύξηση της καθυστέρησης των πακέτων λόγω συμφόρησης.

Ξεκινώντας το 1986 ο Van Jacobson ανέπτυξε τον μηχανισμό έλεγχου συμφόρησης που είναι πλέον απαραίτητος σε κάθε υλοποίηση του TCP και αποτελεί την λύση στο πρόβλημα της κατάρρευσης. Σύμφωνα με αυτόν κάθε αποστολέας θα πρέπει να μειώνει τον ρυθμό μετάδοσής του όταν ανιχνεύσει συνθήκες συμφόρησης,

δηλαδή όταν ανιχνεύσει απώλεια πακέτων. Έτσι οι ροές TCP αποκρίνονται σε σήματα συμφόρησης.

Όμως ο μηχανισμός έλεγχου συμφόρησης του TCP, αν και απαραίτητος για την αποφυγή της κατάρρευσης και πολύ αποτελεσματικός, δεν τα καταφέρνει σε όλες τις περιπτώσεις. Υπάρχουν εφαρμογές στις οποίες απαιτείται σταθερότερος ρυθμός μετάδοσης από αυτόν που μπορεί να δώσει το TCP, όπως ροές πολυμέσων, ή ταυτόχρονη και αξιόπιστη αποστολή σε πολλούς χρηστές (reliable multicasting). Και αυτές οι εφαρμογές όμως θα πρέπει να υλοποιούν κάποιο σχήμα έλεγχου συμφόρησης, ώστε να αποφευχθούν φαινόμενα κατάρρευσης.

Επιπλέον θα πρέπει όλοι αυτοί οι αλγόριθμοι έλεγχου συμφόρησης να αντιδρούν με τον ίδιο τρόπο με το TCP σε σήματα συμφόρησης. Λόγω της «οπισθοχώρησης» του TCP κατά τη διάρκεια συμφόρησης, πολλαπλές συνδέσεις TCP μπορούν να μοιραστούν μια ζεύξη με τρόπο που το διαθέσιμο εύρος να κατανέμεται ισότιμα ανάμεσα σε αυτές. Για να υπάρχει λοιπόν δίκαιη κατανομή των πόρων του δικτύου ανάμεσα σε όλες τις ροές δεδομένων, ανεξάρτητα από το πρωτόκολλο που κατά περίπτωση χρησιμοποιείται, θα πρέπει τα σχήματα έλεγχου συμφόρησης να είναι συμβατά μεταξύ τους και κατ' επέκταση με το TCP.

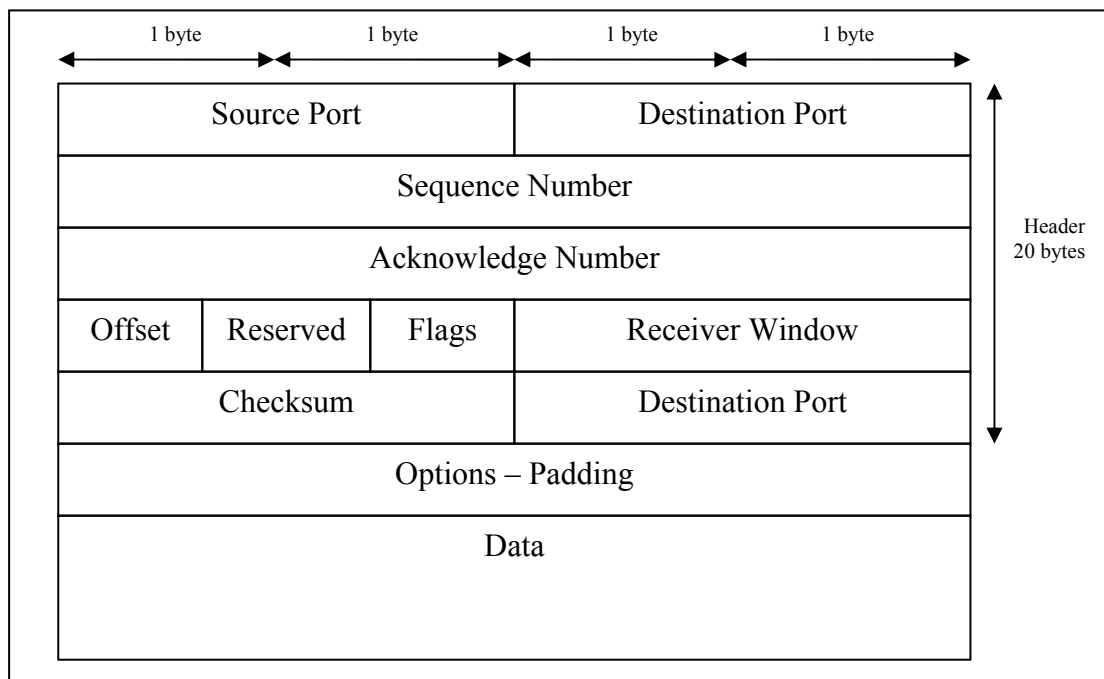
Ροές δεδομένων που δεν υλοποιούν κάποιο σχήμα έλεγχου συμφόρησης (μη υπεύθυνες ροές) μπορούν να μονοπωλήσουν μια ζεύξη, ακόμη κι αν τα πακέτα δεδομένων τους χάνονται σε κάποιο επόμενο σημείο του δικτύου. Έτσι μειώνεται το συνολικό ωφέλιμο έργο του δικτύου, χωρίς οι συγκεκριμένες ροές να ωφελούνται. Η κατάσταση αυτή αποτελεί ένα διαφορετικό είδος «κατάρρευσης λόγω συμφόρησης», που αποκαλείται «κατάρρευση λόγω μη παραδιδόμενων πακέτων» και μπορεί και αυτή να αποφευχθεί μέσω του έλεγχου συμφόρησης.

Από τα παραπάνω γίνεται φανερό ότι η υιοθέτηση κάποιου αλγορίθμου έλεγχου συμφόρησης είναι επιτακτική για όλες τις ροές δεδομένων, τόσο για να επιτυγχάνεται η σταθερότητα του δικτύου σε περιόδους συμφόρησης, όσο και για την εξασφάλιση δίκαιης κατανομής του διαθέσιμου εύρους ζώνης σε κάθε ροή. Αυτό μάλιστα εξασφαλίζεται χωρίς τον διαχωρισμό των ροών (per flow scheduling), ο οποίος έχει στοιχεία μεταγωγής κυκλώματος. Έτσι διατηρείται ο χωρίς σύνδεση χαρακτήρας του διαδικτύου.

2.2 Έλεγχος συμμόρφωσης του TCP

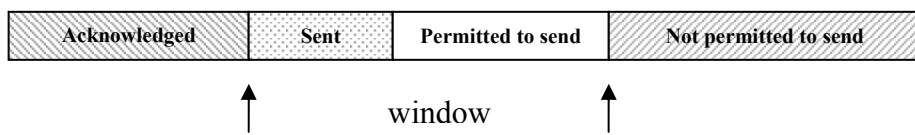
Το πρωτόκολλο TCP είναι σήμερα το κυρίαρχο πρωτόκολλο μεταφοράς στο διαδίκτυο και χρησιμοποιείται από τη συντριπτική πλειοψηφία των εφαρμογών. Δεν είναι τυχαίο το ότι η διαστρωμάτωση των πρωτοκόλλων του διαδικτύου ονομάζεται στοίβα πρωτοκόλλων TCP/IP. Η δυναμική του TCP έγκειται στο ότι προσφέρει αξιόπιστη μεταφορά δεδομένων. Ο αποστολέας σε κάθε πακέτο, πέρα από τις διευθύνσεις της πηγής και του προορισμού, τοποθετεί και έναν αύξοντα αριθμό που εκφράζει τον αριθμό των οκτάδων (bytes) δεδομένων που έχουν μεταδοθεί. Ο παραλήπτης επιβεβαιώνει τα δεδομένα που έχει λάβει, τοποθετώντας στα πακέτα επιβεβαίωσης τον αριθμό που αντιστοιχεί στα δεδομένα που έχει λάβει αυξημένο κατά ένα, δηλαδή τον αριθμό που αντιστοιχεί στο πρώτο byte του επόμενου πακέτου που αναμένεται. Τα δεδομένα που δεν επιβεβαιώνονται εγκαίρως, δηλαδή πριν την εκπνοή ενός χρονομετρητή, επαναμεταδίδονται.

Δομή πακέτων TCP



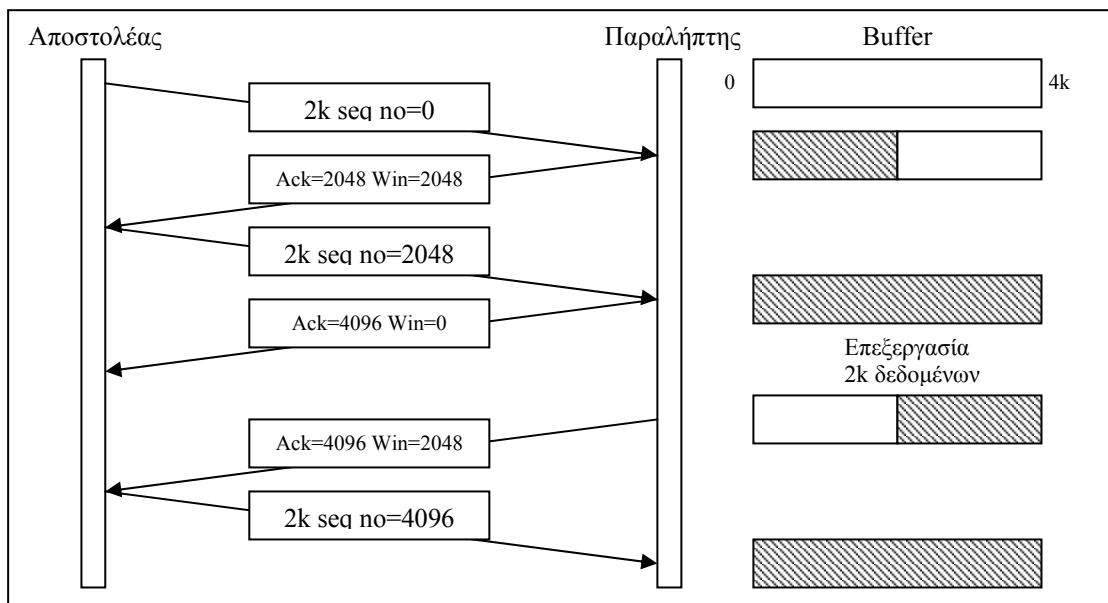
Ένα από τα βασικά κομμάτια του πρωτοκόλλου αυτού είναι ο έλεγχος ροής των αποστελλόμενων δεδομένων. Αυτό επιτυγχάνεται με έναν αλγόριθμο ολισθαίνοντος παραθύρου. Ο αποστολέας επιτρέπεται να στείλει δεδομένα μόνο εάν το πλήθος των δεδομένων που έχουν σταλθεί και δεν έχουν ακόμη επιβεβαιωθεί δεν ξεπερνά το μέγεθος αυτού του παραθύρου. Όταν λαμβάνεται ένα πακέτο που επιβεβαιώνει την λήψη κάποιων δεδομένων που έχουν αποσταλεί, το παράθυρο ολισθαίνει κατά τον αριθμό των δεδομένων που επιβεβαιώθηκαν, επιτρέποντας να σταλθούν νέα δεδομένα. Έτσι η λήψη των επιβεβαιώσεων καθορίζει την αποστολή των πακέτων.

Το παράθυρο του TCP



Κατά την αρχικοποίηση της σύνδεσης ο παραλήπτης στέλνει τον μέγιστο αριθμό δεδομένων που μπορεί να δεχθεί ταυτόχρονα, ο οποίος αντιστοιχεί στο μέγεθος του ενταμιευτή (buffer) που έχει διατεθεί για τη συγκεκριμένη σύνδεση. Στην διάρκεια της σύνδεσης ο διαθέσιμος χώρος αυξομειώνεται ανάλογα με τον αριθμό των δεδομένων που υπάρχουν στο buffer, οπότε ο παραλήπτης «διαφημίζει» το πλήθος των δεδομένων που μπορεί να δεχθεί. Αυτό είναι το «παράθυρο δεκτή» (receiver window) το οποίο περιέχεται στην επικεφαλίδα των επιβεβαιώσεων. Στην πρώτη έκδοση του TCP αυτό το παράθυρο καθόριζε αποκλειστικά τον έλεγχο ροής του αποστολέα.

Παράδειγμα μεταβολής του παραθύρου δέκτη



Στις νεότερες υλοποιήσεις του TCP ο μηχανισμός ελέγχου συμφόρησης που αναπτύχθηκε από τον Van Jacobson προβλέπει την αυξομείωση του παραθύρου του αποστολέα ανάλογα με την συμφόρηση του δικτύου. Συγκεκριμένα ο αποστολέας διατηρεί μια ακόμη μεταβλητή, το παράθυρο συμφόρησης (congestion window), και θέτει ως παράθυρο ελέγχου ροής κάθε φορά το μικρότερο από τα δυο, δηλαδή :

$$window = \min(receiver\ window, congestion\ window)$$

Ο έλεγχος του παραθύρου συμφόρησης πραγματοποιείται με τους αλγόριθμους αργής εκκίνησης (slow start), αποφυγής συμφόρησης (congestion avoidance), γρήγορης επαναμετάδοσης (fast retransmit) και γρήγορης ανάκαμψης (fast recovery).

Για τη εξήγηση των αλγορίθμων αυτών απαιτείται ο ορισμός των παρακάτω όρων οι οποίοι χρησιμοποιούνται στη συνέχεια.

- **Τμήμα (segment)** : Είναι οποιοδήποτε TCP/IP πακέτο δεδομένων ή – και επιβεβαίωσης.
- **Μέγιστο μέγεθος τμήματος αποστολέα (SMSS)** : Είναι το μέγεθος του πιο μεγάλου τμήματος που μπορεί να μεταδώσει ο αποστολέας. Καθορίζεται από τη μέγιστη μονάδα μετάδοσης (MTU) του δικτύου, από το RMSS (βλέπε παρακάτω) ή από άλλους παράγοντες. Δεν περιλαμβάνονται οι επικεφαλίδες.
- **Μέγιστο μέγεθος τμήματος παραλήπτη (RMSS)** : Είναι το μέγεθος του πιο μεγάλου τμήματος που μπορεί να δεχθεί ο παραλήπτης. Καθορίζεται κατά την έναρξη της σύνδεσης, διαφορετικά τίθεται ίσο με 536 bytes. Δεν περιλαμβάνονται οι επικεφαλίδες.
- **Τμήμα πλήρους μεγέθους (full-sized segment)** : Ένα τμήμα που περιέχει τον μέγιστο επιτρεπτό αριθμό δεδομένων, δηλαδή SMSS bytes δεδομένων.
- **Παράθυρο δεκτή (receiver window, rwnd)** : Το πιο πρόσφατα διαφημισμένο παράθυρο δεκτή.
- **Παράθυρο συμφόρησης (congestion window, cwnd)** : Μεταβλητή που περιορίζει το πλήθος των δεδομένων που μπορούν να σταλούν.
- **Αρχικό παράθυρο (Initial Window, IW)** : Η τιμή του παραθύρου συμφόρησης αμέσως μετά την έναρξη της σύνδεσης.

- **Παράθυρο απώλειας (Loss Window, LW)** : Το μέγεθος του παραθύρου συμφόρησης έπειτα από την ανίχνευση χαμένου πακέτου, μέσω της εκπνοής του χρονομετρητή επαναμετάδοσης.
- **Παράθυρο επανεκκίνησης (Restart Window)** : Το μέγεθος του παραθύρου συμφόρησης μετά την επανεκκίνησης της μετάδοσης από μια περίοδο αδράνειας.
- **Flight Size** : Το πλήθος των δεδομένων που έχουν αποσταλεί αλλά δεν έχουν επιβεβαιωθεί ακόμη.

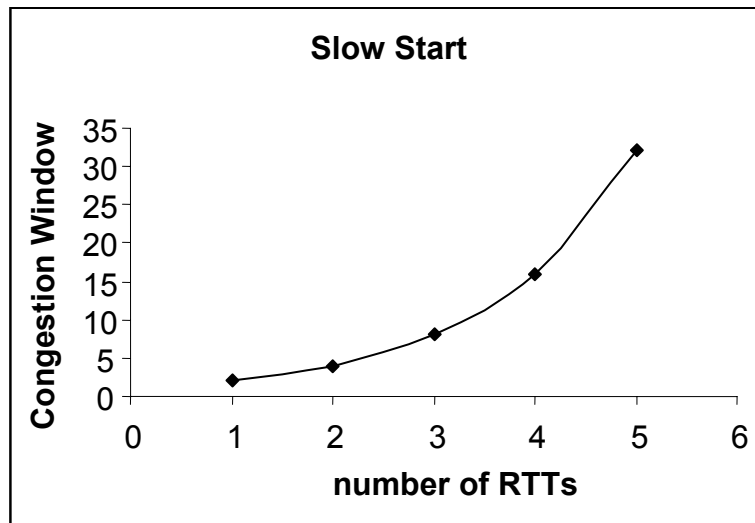
2.2.1 Αργή εκκίνηση (Slow Start) και Αποφυγή συμφόρησης (Congestion Avoidance)

Οι αλγόριθμοι αργής εκκίνησης και αποφυγής συμφόρησης καθορίζουν το πλήθος των δεδομένων που μπορούν να εισαχθούν στο δίκτυο. Για τον καθορισμό του αλγόριθμου που πρέπει να χρησιμοποιείται σε κάθε στιγμή χρειαζόμαστε μια ακόμη μεταβλητή το κατώφλι αργής εκκίνησης (ssthresh).

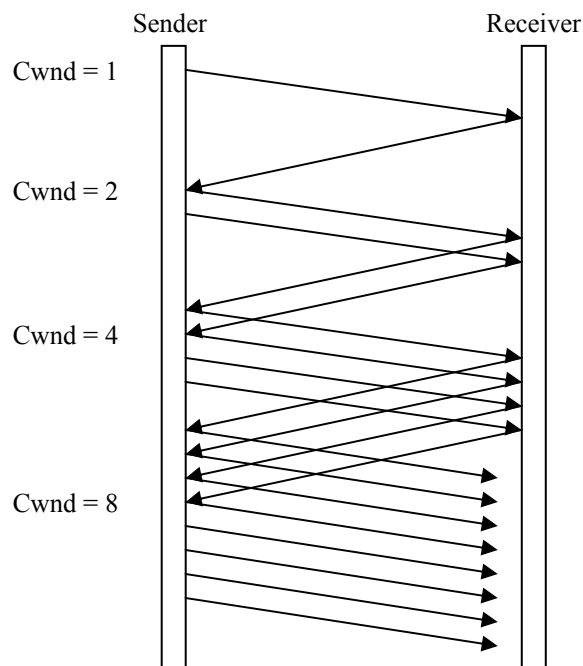
Κατά την εκκίνησης μιας σύνδεσης ή την επανεκκίνηση της μετά από εκπνοή του χρονομετρητή αναμετάδοσης, ο αποστολέας δεν έχει εικόνα για την κατάσταση του δικτύου. Έτσι, ανιχνεύει αργά – αργά το δίκτυο, για να αποφευχθεί η συμφόρησή του με μια μεγάλη ποσότητα δεδομένων. Γι αυτόν το σκοπό χρησιμοποιείται ο αλγόριθμος της αργής εκκίνησης.

Το αρχικό παράθυρο (IW) του αποστολέα είναι το πολύ ίσο με δυο φορές το μέγιστο μέγεθος τμήματος αποστολέα (SMSS) και όχι μεγαλύτερο από δυο τμήματα. Για κάθε επιβεβαίωση νέων δεδομένων το μέγεθος του παραθύρου αυξάνεται κατά SMSS bytes. Αυτό έχει ως αποτέλεσμα σε κάθε Round Trip Time να διπλασιάζεται το παράθυρο συμφόρησης. Η αργή εκκίνηση σταματά όταν το μέγεθος του παραθύρου φτάσει ή ξεπεράσει το κατώφλι αργής εκκίνησης (ssthresh) ή όταν παρατηρηθεί συμφόρηση.

Αύξηση του παραθύρου συμφόρησης σε κάθε RTT κατά το Slow Start



Παράδειγμα λειτουργίας του αλγορίθμου αργής εκκίνησης

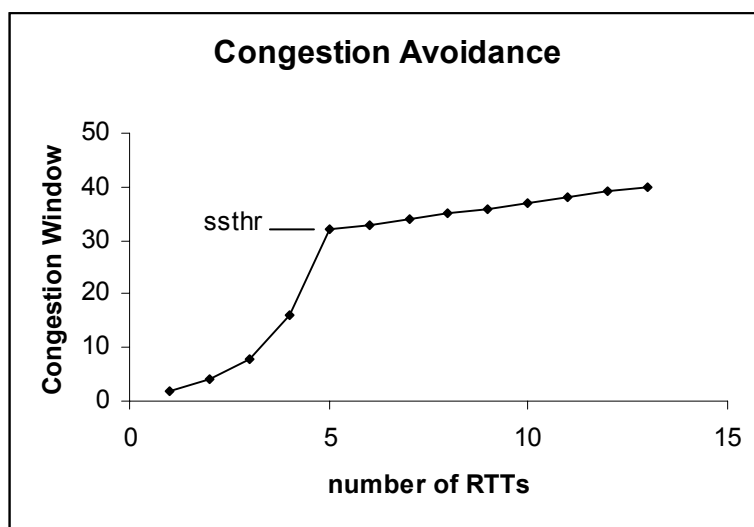


Μετά τη φάση της αργής εκκίνησης ακολουθεί η φάση της αποφυγής συμφόρησης. Σε αυτή την περίοδο το μέγεθος του παραθύρου αυξάνεται κατά ένα τμήμα πλήρους μεγέθους κάθε RTT, μέχρι να παρατηρηθεί συμφόρηση. Στις περισσότερες υλοποιήσεις το παράθυρο αυξάνεται κατά μια σταθερή ποσότητα με κάθε επιβεβαίωση σύμφωνα με τον τύπο :

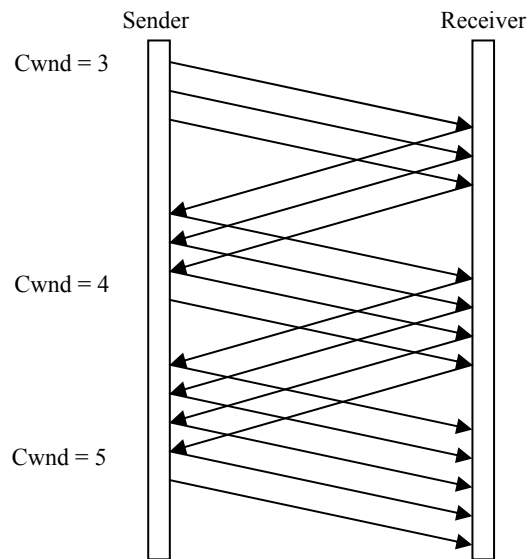
$$cwnd = cwnd + SMSS \cdot \frac{SMSS}{cwnd}$$

Εάν αποστέλλονται επιβεβαιώσεις για κάθε τμήμα που λαμβάνεται η παραπάνω εξίσωση δίνει ως αποτέλεσμα ελαφρώς μεγαλύτερη αύξηση από ένα τμήμα ανά RTT, ενώ αν επιβεβαιώνεται κάθε δεύτερο πακέτο, ελαφρώς μικρότερη.

Αύξηση του παραθύρου συμφόρησης σε κάθε RTT κατά το Congestion Avoidance



Παράδειγμα λειτουργίας του αλγορίθμου αποφυγής συμφόρησης

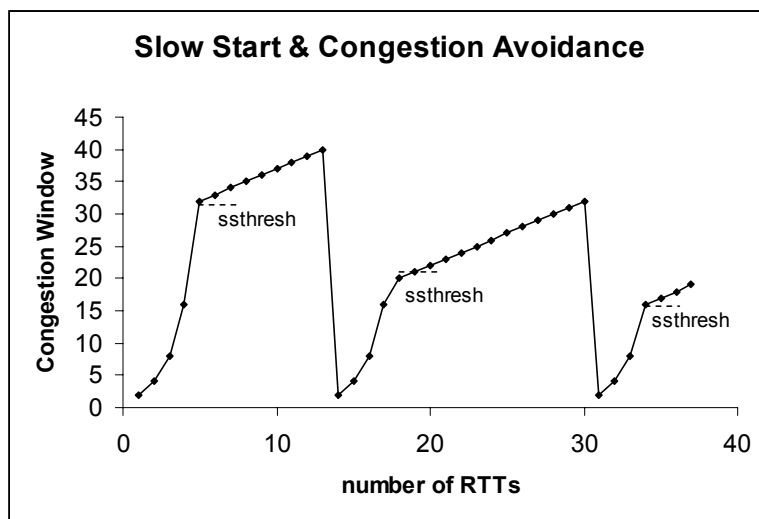


Όταν ανιχνεύεται απώλεια πακέτου μέσω της εκπνοής του χρονομετρητή αναμετάδοσης, τότε το κατώφλι αργής εκκίνησης θα πρέπει να τεθεί το πολύ ίσο με :

$$ssthresh = \max\left(\frac{FlightSize}{2}, 2 \cdot SMSS\right)$$

Επιπλέον θα πρέπει το παράθυρο συμφόρησης να γίνει το πολύ ίσο με το παράθυρο απώλειας (LW) το οποίο ισούται με ένα τμήμα πλήρους μεγέθους. Έπειτα χρησιμοποιείται ο αλγόριθμος της αργής εκκίνησης για την αύξηση του παραθύρου μέχρι τη νέα τιμή του slow start threshold, οπότε και αναλαμβάνει και πάλι το congestion avoidance.

Εναλλαγή Slow Start και Congestion Avoidance και καθορισμός του ssthresh



2.2.2. Γρήγορη Επανεκπομπή (Fast Retransmit) και Γρήγορη Ανάκαμψη (Fast Recovery)

Όταν ο παραλήπτης σε μια TCP σύνδεση λάβει ένα τμήμα εκτός σειράς, δηλαδή ένα τμήμα με sequence number μεγαλύτερο από εκείνο που αναμενόταν, στέλνει άμεσα μια διπλότυπη επιβεβαίωση (duplicate acknowledgement). Σκοπός της επιβεβαίωσης αυτής είναι να ειδοποιηθεί άμεσα ο αποστολέας για την λήψη ενός εκτός σειράς τμήματος και για το ποιο τμήματος αναμενόταν κανονικά. Η λήψη τμημάτων εκτός σειράς από τον παραλήπτη και άρα οι διπλότυπες επιβεβαιώσεις που λαμβάνει ο αποστολέας, μπορεί να οφείλονται σε διάφορα προβλήματα του δικτύου. Πρώτα απ' όλα μπορούν να προκληθούν από κάποιο χαμένο τμήμα, οπότε όλα τα επόμενα τμήματα που θα ληφθούν θα προκαλέσουν διπλότυπες επιβεβαιώσεις. Επίσης μπορούν να προκληθούν από αναδιάρθρωση των τμημάτων στο δίκτυο. Τέλος μπορεί οι διπλότυπες επιβεβαιώσεις να προέρχονται από μια αντεγραμμένη από το δίκτυο επιβεβαίωση. Ο παραλήπτης θα πρέπει επίσης να επιβεβαιώνει άμεσα την λήψη ενός τμήματος που συμπληρώνει ολόκληρο ή τμήμα του κενού στην ακολουθία των δεδομένων. Έτσι ο αποστολέας ενημερώνεται άμεσα για το ποια δεδομένα έχουν ληφθεί σωστά.

Σύμφωνα με τον αλγόριθμο της γρήγορης επανεκπομπής, ο αποστολέας ανιχνεύει την απώλεια τμημάτων μέσω των διπλότυπων επιβεβαιώσεων που λαμβάνει. Συγκεκριμένα η λήψη τριών διπλότυπων επιβεβαιώσεων (τεσσάρων πανομοιότυπων επιβεβαιώσεων χωρίς άλλες ενδιάμεσες επιβεβαιώσεις) ερμηνεύεται από τον δεκτή ως απώλεια του τμήματος που υποδεικνύεται από αυτές. Έτσι το τμήμα αυτό επανεκπέμπεται άμεσα, χωρίς να αναμένεται η λήξη του χρονομετρητή αναμετάδοσης.

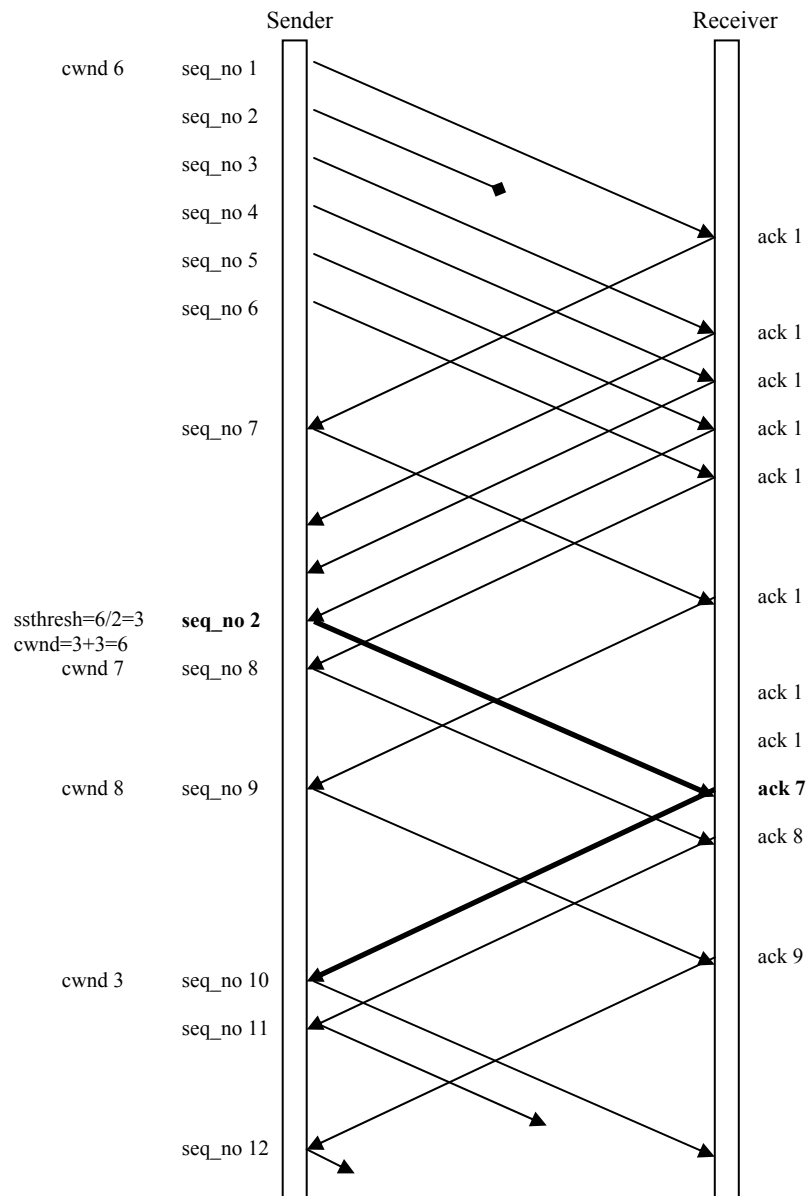
Έπειτα από την αναμετάδοση του τμήματος αυτού, τον έλεγχο της μετάδοσης αναλαμβάνει ο αλγόριθμος της γρήγορης ανάκαμψης. Σε παλαιότερες εκδόσεις του TCP (Tahoe TCP) μετά την αναμετάδοση γινόταν και πάλι αργή εκκίνηση. Όμως η λήψη των διπλότυπων επιβεβαιώσεων, πέρα από την ένδειξη για απώλεια κάποιου τμήματος, υποδηλώνει και την συνέχιση της λήψης τμημάτων από τον παραλήπτη, αφού οι διπλότυπες επιβεβαιώσεις παράγονται μόνο κατά την λήψη δεδομένων. Επομένως για κάθε διπλότυπη επιβεβαίωση ο παραλήπτης έχει λάβει κάποιο τμήμα, που δεν καταλαμβάνει πλέον πόρους του δικτύου. Επιπλέον ο αποστολέας

εξακολουθεί να λαμβάνει επιβεβαιώσεις, επομένως μπορεί να συγχρονίσει την αποστολή νέων δεδομένων, με μειωμένο όμως παράθυρο.

Συνήθως η γρήγορη επανεκπομπή και η γρήγορη ανάκαμψη υλοποιούνται μαζί ως εξής.

- Όταν ληφθεί η τρίτη διπλότυπη επιβεβαίωση το κατώφλι αργής εκκίνησης τίθεται το πολύ ίσο με $ssthresh = \max(\frac{FlightSize}{2}, 2 \cdot SMSS)$
- Επανεκπέμπεται το χαμένο τμήμα και το παράθυρο συμφόρησης γίνεται ίσο με : $cwnd = ssthresh + 3 \cdot SMSS$. Έτσι το παράθυρο διογκώνεται τεχνητά κατά τον αριθμό των τμημάτων που έχουν παραληφθεί σωστά από τον δεκτή
- Για κάθε επιπλέον διπλότυπη επιβεβαίωση που λαμβάνεται το παράθυρο αυξάνεται κατά SMSS. Έτσι το παράθυρο διογκώνεται περαιτέρω για να αντικατοπτρίσει το επιπλέον τμήμα που έχει φτάσει στον δεκτή.
- Εάν η νέα τιμή του παραθύρου το επιτρέπει, μεταδίδεται ένα τμήμα.
- Όταν φτάσει κάποια επιβεβαίωση που επιβεβαιώνει τη λήψη νέων δεδομένων το παράθυρο συμφόρησης γίνεται ίσο με την τιμή του κατωφλίου αργής εκκίνησης που υπολογίστηκε παραπάνω, ώστε να αποδιογκωθεί το παράθυρο και να αποφευχθεί μια ξαφνική ριπή πακέτων, αφού το αριστερό άκρο του ολισθαίνει, εάν δεν υπήρχε και άλλο χαμένο πακέτο, στην τιμή του τελευταίου πακέτου που στάλθηκε πριν ληφθεί η τρίτη επιβεβαίωση. Έπειτα βγαίνουμε από τη φάση της γρήγορης ανάκαμψης και επιστρέφουμε στη φάση της αποφυγής συμφόρησης.

Παράδειγμα λειτουργίας των αλγορίθμων fast retransmit και fast recovery



2.2.3 Η τροποποίηση New Reno στον αλγόριθμο της γρήγορης ανάκαμψης του TCP.

Εάν η επιβεβαίωση νέων δεδομένων κατά τη φάση της γρήγορης αναμετάδοσης δεν καλύπτει όλα τα τμήματα που στάλθηκαν πριν από την έναρξη της, τότε πιθανότατα υπήρξαν και αλλά χαμένα τμήματα από αυτό το «παράθυρο» πακέτων. Η έκδοση του TCP που περιγράφεται παραπάνω (Reno TCP) δεν

ανακάμπτει ιδιαίτερα γρήγορα από πολλαπλές απώλειες σε ένα «παράθυρο» τμημάτων. Όταν φτάσει η πρώτη επιβεβαίωση για νέα δεδομένα και αυτά δεν περιλαμβάνουν όλα τα τμήματα όπως αναφέρθηκε παραπάνω (την οποία ονομάζουμε μερική επιβεβαίωση), τότε οι επόμενες επιβεβαιώσεις θα είναι διπλότυπες και άρα θα ξαναμπούμε στον παραπάνω αλγόριθμο. Έτσι οδηγούμαστε σε πολλαπλές γρήγορες αναμεταδόσεις που έχουν ως αποτέλεσμα την εκ νέου μείωση του παραθύρου και άρα την μείωση του ρυθμού μετάδοσης.

Για να αποφύγουμε αυτό το φαινόμενο, όταν λάβουμε μια μερική επιβεβαίωση δεν βγαίνουμε από τον αλγόριθμο της γρήγορης ανάκαμψης, αλλά υποθέτουμε ότι έχουμε πολλαπλές απώλειες πακέτων και ενεργούμε ανάλογα. Έτσι οι αλγόριθμοι γρήγορης αναμετάδοσης και γρήγορης ανάκαμψης τροποποιούνται ως εξής :

- Όταν ληφθεί η τρίτη διπλότυπη επιβεβαίωση και δεν βρισκόμαστε ήδη στη φάση γρήγορης ανάκαμψης, το κατώφλι αργής εκκίνησης τίθεται το πολύ ίσο με $ssthresh = \max(\frac{FlightSize}{2}, 2 \cdot SMSS)$. Επίσης καταγράφεται ο μέγιστος ακολουθιακός αριθμός δεδομένων που έχουν αποσταλεί σε μια νέα μεταβλητή την μεταβλητή ανάκαμψης «recover».
- Επανεκπέμπεται το χαμένο τμήμα και το παράθυρο συμφόρησης γίνεται ίσο με : $cwnd = ssthresh + 3 \cdot SMSS$. Έτσι το παράθυρο διογκώνεται τεχνητά κατά τον αριθμό των τμημάτων που έχουν παραληφθεί σωστά από τον δεκτή
- Για κάθε επιπλέον διπλότυπη επιβεβαίωση που λαμβάνεται το παράθυρο αυξάνεται κατά SMSS. Έτσι το παράθυρο διογκώνεται περαιτέρω για να αντικατοπτρίσει το επιπλέον τμήμα που έχει φτάσει στον δεκτή.
- Εάν η νέα τιμή του παραθύρου το επιτρέπει, μεταδίδεται ένα τμήμα.
- Όταν ληφθεί μια επιβεβαίωση για νέα δεδομένα εξετάζουμε το πλήθος των δεδομένων που επιβεβαιώνονται :
 - Εάν η επιβεβαίωση καλύπτει όλα τα δεδομένα μέχρι και την τιμή recover, τότε όλα τα ενδιάμεσα τμήματα από το χαμένο πακέτο μέχρι και το τελευταίο που εστάλη πριν τη λήψη της τρίτης διπλότυπης επιβεβαίωσης. Έτσι το παράθυρο συμφόρησης τίθεται ίσο με $ssthresh$ ή $\min(ssthresh, FlightSize + SMSS)$ και επιστρέφουμε στην φάση της αποφυγής συμφόρησης. Με την τελευταία επιλογή για το παράθυρο

εξασφαλίζεται ότι δεν θα υπάρξει μια ξαφνική ριπή τμημάτων, στην περίπτωση που η τιμή του *ssthresh* είναι πολύ μεγαλύτερη από το πλήθος των ανεπιβεβαιωτών δεδομένων *FlightSize*.

- Εάν η επιβεβαίωση δεν καλύπτει όλα τα δεδομένα μέχρι και την τιμή *recover*, τότε πρόκειται για μερική επιβεβαίωση. Σε αυτή την περίπτωση επαναμεταδίδεται το πρώτο μη επιβεβαιωμένο τμήμα. Επιπλέον το παράθυρο μειώνεται κατά το πλήθος των δεδομένων που επιβεβαιώθηκαν και στο αποτέλεσμα προστίθεται ένα SMSS. Έτσι διασφαλίζεται ότι όταν τελειώσει η φάση της γρήγορης ανάκαμψης το πλήθος των ανεπιβεβαιωτών δεδομένων θα είναι περίπου ίσο με την τιμή του *ssthresh*. Τέλος δεν βγαίνουμε από την φάση της γρήγορης ανάκαμψης, αλλά οι τυχόν νέες επιβεβαιώσεις αντιμετωπίζονται με βάση τα παραπάνω βήματα.

2.3 Ο μηχανισμός ελέγχου συμφόρησης TFRC

Ο αλγόριθμος TFRC (TCP Friendly Rate Control) ή Φιλικός προς το TCP Έλεγχος Ρυθμού είναι ένα σχήμα ελέγχου συμφόρησης που επιδιώκει την εξασφάλιση ομαλού ρυθμού μετάδοσης, ο οποίος όμως δεν είναι υπερβολικά πιο μεγάλος από αυτόν του TCP. Η έννοια του φιλικού προς το TCP (TCP Friendly) υποδηλώνει ότι ο υπ' όψιν αλγόριθμος δεν καταλαμβάνει περισσότερους από τους διπλάσιους πόρους ενός δικτύου από αυτούς που καταλαμβάνει μια TCP σύνδεση κάτω από τις ίδιες συνθήκες.

Πρέπει να επισημανθεί ότι το TFRC δεν είναι πρωτόκολλο μεταφοράς αλλά μηχανισμός έλεγχου συμφόρησης, που χρησιμοποιείται από αλλά πρωτοκολλά μεταφοράς και εφαρμογές, τα οποία απαιτούν σταθερό ρυθμό μετάδοσης. Αρχικά σχεδιάστηκε για την μετάδοση ροών πολυμέσων (multimedia streaming), όμως μπορεί να χρησιμοποιηθεί και από άλλες εφαρμογές που απαιτούν ομαλό ρυθμό μετάδοσης όπως είναι για παράδειγμα επικοινωνία φωνής μέσω του διαδικτύου (voice over IP).

Η απαίτηση για ομαλό ρυθμό μετάδοσης επιτυγχάνεται μέσω ελέγχου συμφόρησης όχι βασιζόμενου σε παράθυρο αλλά σε ρυθμό. Το TCP χρησιμοποιεί όπως είδαμε ένα μεταβαλλόμενο παράθυρο για τον έλεγχο των δεδομένων που μεταδίδει. Ο αλγόριθμος έλεγχου συμφόρησης που υλοποιεί όμως προκαλεί την περιοδική αυξομείωση αυτού του παραθύρου με αποτέλεσμα, ακόμη και αν ο μέσος ρυθμός μετάδοσης είναι σταθερός, ο στιγμιαίος να παρουσιάζει σημαντικές διακυμάνσεις. Γι αυτό τον λόγο το TFRC μεταδίδει με συγκεκριμένο ρυθμό, που υπολογίζεται με βάση αναλυτικό μοντέλο για τον ρυθμό μιας TCP σύνδεσης. Ο ρυθμός αυτός μακροπρόθεσμα καθορίζεται από την συμφόρηση που υπάρχει στο δίκτυο, μέσω πακέτων ανάδρασης (feedback) που στέλνει περιοδικά ο δέκτης.

Το άνω φράγμα του ρυθμού μετάδοσης μιας TCP σύνδεσης (και άρα του throughput) δίνεται προσεγγιστικά από τη σχέση :

$$X = \frac{S}{RTT \cdot \sqrt{2 \cdot \frac{B \cdot P_{loss}}{3}} + \left(RTO \cdot \left(3 \cdot \sqrt{3 \cdot \frac{B \cdot P_{loss}}{8}} \cdot P_{loss} \cdot (1 + 32 \cdot P_{loss}^2) \right) \right)}$$

όπου X ο ρυθμός μετάδοσης, S το μέσο μέγεθος των πακέτων, RTT ο χρόνος που μεσολαβεί από την αποστολή ενός πακέτου μέχρι την λήψη της επιβεβαίωσης για αυτό, B ο αριθμός των πακέτων που επιβεβαιώνεται από κάθε επιβεβαίωση, RTO ο χρόνος λήξης του χρονομετρητή αναμετάδοσης και $Ploss$ ο ρυθμός απωλειών πακέτων.

Συνήθως το RTO επιλέγεται να είναι ίσο με $4 \cdot RTT$, ενώ το B είναι σταθερό. Επιπλέον στις περισσότερες εφαρμογές τα μεταδιδόμενα πακέτα είναι σταθερού μεγέθους και ο ρυθμός μετάδοσης καθορίζεται από τα διαστήματα μεταξύ διαδοχικών πακέτων. Έτσι για τη συνεχεία δεχόμαστε ότι και το S είναι σταθερό. Επομένως το TFRC θα πρέπει να μέτρα το RTT και τον ρυθμό απωλειών $Ploss$ ώστε να υπολογίζει το X και ανάλογα να προσαρμόζει τον ρυθμό μετάδοσής του.

2.3.1 Περιεχόμενα των πακέτων

Για να λειτουργήσει το πρωτόκολλο TFRC θα πρέπει τα πακέτα δεδομένων που στέλνονται από τον αποστολέα στον παραλήπτη καθώς και τα πακέτα feedback να περιέχουν τα εξής δεδομένα :

- Πακέτα δεδομένων (DATA)
 - Έναν αύξοντα αριθμό i (sequence number), μοναδικό για κάθε πακέτο που μεταδίδεται.
 - Μια χρονική σφραγίδα (time stamp) t_i που δείχνει την χρονική στιγμή της μετάδοσης κάθε πακέτου i .
 - Την τρέχουσα εκτίμηση του αποστολέα για το RTT R_i . Αυτή τη χρησιμοποιεί ο παραλήπτης μαζί με το timestamp για να διακρίνει αν τα τυχόν χαμένα πακέτα ανήκουν στο ίδιο RTT , όπως θα δούμε παρακάτω.
- Πακέτα FEEDBACK
 - Το time stamp του τελευταίου πακέτου δεδομένων $t_{recvdata}$.
 - Τον χρόνο που πέρασε από την λήψη του τελευταίου πακέτου δεδομένων μέχρι την παραγωγή του feedback t_{delay} .
 - Τον ρυθμό με τον οποίο λαμβάνονται τα δεδομένα μετά την τελευταία αποστολή feedback, όπως τον εκτιμά ο παραλήπτης X_{recv} .
 - Την τρέχουσα εκτίμηση του παραλήπτη για το ρυθμό λαθών p .

2.3.2 Λειτουργία του αποστολέα

Ο αποστολέας στέλνει δεδομένα με έναν ελεγχόμενο ρυθμό. Όταν λάβει κάποιο πακέτο feedback προσαρμόζει τον ρυθμό αυτό με βάση τα δεδομένα που περιέχονται στο συγκεκριμένο πακέτο. Εάν δεν λάβει feedback μέσα σε δυο RTT δηλαδή πριν το μηδενισμό ενός χρονομετρητή no feedback timer ρίχνει τον ρυθμό του στο μισό. Για τη λειτουργία του απαιτείται να διατηρεί κάποιες τοπικές μεταβλητές εκτός των όσων αναφέρθηκαν πριν που είναι :

- Ο χρόνος κατά τον οποίο διπλασιάστηκε τελευταία φορά ο ρυθμός κατά την φάση του slow start tld.
- Ο χρόνος εκπνοής του no feedback timer.
- Ο χρόνος εκπνοής του χρονομετρητή αναμετάδοσης t_RTO. Πρόκειται για την εκτίμηση του t_RTO του TCP που χρησιμοποιείται στον υπολογισμό του X.

Με την έναρξη της σύνδεσης ο αποστολέας μεταδίδει με ρυθμό ενός πακέτου ανά δευτερόλεπτο και θέτει την τιμή του no feedback timer στα δυο δευτερόλεπτα και την τιμή του tld στο -1. Οι τιμές των RTT και t_RTO δεν είναι προσδιορισμένες ακόμη.

Όταν λαμβάνεται ένα πακέτο feedback γίνονται τα εξής :

- Υπολογίζεται η τρέχουσα τιμή του RTT :

$$R_sample = (t_now - t_recvd_data) - t_delay$$

- Ενημερώνεται η μέση τιμή του RTT :

αν είναι το πρώτο feedback $R = R_sample$

διαφορετικά $R = q \cdot R + (1 - q) \cdot R_sample$ συνίσταται $q=0.9$.

- Υπολογίζεται το t_RTO :

$$t_RTO = 4 \cdot R$$

- Γίνεται reset του no feedback timer ώστε να λήξει μετά από :

$$\max\left(4 \cdot R, 2 \cdot \frac{s}{X}\right) \text{ δευτερόλεπτα}$$

- Ενημερώνεται το X ως εξής :
 - Αν $p=0$ βρισκόμαστε σε slow start οπότε

$$\text{αν } t_now - tld \geq R$$

$$X = \max \left[\min(2 \cdot X, 2 \cdot X_recv), \frac{s}{R} \right]$$

$$tld = t_now$$

- Αν $p>0$

Υπολογίζεται το X_calc από την εξίσωση του throughput του TCP

$$X = \max \left[\min(X_calc, 2 \cdot X_recv), \frac{s}{t_mbi} \right] \text{ με } t_mbi = 64 \text{ sec}$$

Όταν βρισκόμαστε σε slow start ο ρυθμός μετάδοσης διπλασιάζεται σχεδόν κάθε RTT, με άνω φράγμα το διπλάσιο του ρυθμού λήψης δεδομένων από τον δέκτη, ώστε να μην στέλνεται υπερβολικά υψηλό πλήθος δεδομένων στο δίκτυο, και κάτω φράγμα ένα πακέτο κάθε RTT. Σε διαφορετική περίπτωση ο ρυθμός είναι ο υπολογιζόμενος με το ίδιο άνω φράγμα και με κάτω φράγμα ένα πακέτο κάθε 64 sec, ώστε να μην σταματάει τελείως η μετάδοση σε περίπτωση επίμονης έλλειψης feedback.

Εάν εκπνεύσει ο no feedback timer τότε γίνονται τα εξής :

- Ο ρυθμός μετάδοσης υποδιπλασιάζεται. Εάν είχε ληφθεί προηγούμενο feedback αυτό γίνεται τροποποιώντας το τοπικά αποθηκευμένο X_recv , το οποίο όπως είδαμε περιορίζει το X . Με αυτόν τον τρόπο έχουμε μείωση του ρυθμού μετάδοσης, αλλά ταυτόχρονα επιτρέπουμε την εκ νέου ανάκαμψη με ένα είδος slow start , αν τα επόμενα πακέτα feedback εξακολουθήσουν να αναφέρουν μηδενικό ρυθμό λαθών. Η τροποποίηση που αναφέραμε είναι η εξής :

$$\text{Αν } X_calc > 2 \cdot X_recv$$

$$X_recv = \max \left(\frac{X_recv}{2}, \frac{s}{2 \cdot t_mbi} \right)$$

Διαφορετικά

$$X_recv = \frac{X_calc}{4}$$

- Υπολογίζεται ξανά ο ρυθμός μετάδοσης, ώστε να υποδιπλασιαστεί τελικά όπως είδαμε παραπάνω.

Σε περίπτωση που δεν είχαμε προηγούμενο feedback παραλείπεται το πρώτο βήμα και υποδιπλασιάζεται άμεσα ο ρυθμός

$$X = \max\left(\frac{X}{2}, \frac{s}{t_mbi}\right)$$

- Επανεκκινεί τον χρονομετρητή με νέο χρόνο εκπνοής :

$$\max\left(4 \cdot R, 2 \cdot \frac{s}{X}\right) \text{ δευτερόλεπτα}$$

Εάν για κάποιο λόγο ο αποστολέας σταματήσει τη μετάδοση για κάποιο διάστημα θα έχουμε διαδοχικά no feedback timeout και το X_recv που κρατά στη μνήμη του ο αποστολέας θα μειωθεί αντίστοιχα. Έτσι εάν επανεκκίνηση τη μετάδοση θα περιοριστεί ο ρυθμός ο οποίος θα αυξηθεί με ένα αντίστοιχο του slow start τρόπο μέχρι να φτάσει στην τιμή του X_calc .

2.3.3 Λειτουργία του δέκτη

Ο δέκτης στέλνει περιοδικά μηνύματα feedback στον αποστολέα ώστε ο τελευταίος να ρυθμίσει το ρυθμό μετάδοσής του σύμφωνα με τις πληροφορίες που περιέχονται σε αυτά. Κανονικά θα πρέπει να στέλνεται ένα feedback ανά RTT (μέσω ενός χρονομετρητή feedback timer), εκτός εάν ο ρυθμός μετάδοσης είναι μικρότερος από ένα πακέτο ανά RTT οπότε στέλνεται feedback για κάθε εισερχόμενο πακέτο. Επίσης θα πρέπει να στέλνεται άμεσα feedback όταν ανιχνευθεί κάποια απώλεια καθώς και όταν κάποιο πακέτο φτάσει εκτός σειράς και αναιρέσει μια απώλεια.

Για να ανιχνεύσει τις απώλειες πακέτων ο δεκτής κρατάει στη μνήμη του ένα «ιστορικό» με τους αύξοντες αριθμούς των πακέτων που λαμβάνει και τους χρόνους άφιξης τους. Εάν ληφθούν τουλάχιστο τρία πακέτα εκτός σειράς συμπεραίνεται ότι έχει χαθεί το πακέτο που κανονικά αναμενόταν. Η απαίτηση για τρία εκτός σειράς πακέτα είναι αντίστοιχη με αυτή του TCP. Σε αντίθεση με αυτό όμως, αν τελικά ληφθεί το πακέτο που υποτέθηκε ως χαμένο μπορεί να αναιρεθεί αυτή η απώλεια και

να ξαναυπολογιστεί ο σωστός ρυθμός απωλειών p . Η πληροφορία για το χρόνο άφιξης χρησιμοποιείται για την διάκριση τυχόν πολλαπλών χαμένων πακέτων μέσα στο ίδιο RTT τα οποία αντιμετωπίζονται σαν να είχε χαθεί ένα μόνο πακέτο (ένα μόνο loss event), όσον αφορά τον υπολογισμό του ρυθμού απωλειών. Αυτή η θεώρηση είναι αντίστοιχη με τη μείωση του παραθύρου μόνο μια φορά για διαδοχικές απώλειες που κάνει το New Reno TCP.

Ο υπολογισμός του ρυθμού απωλειών p είναι ίσως η σημαντικότερη λειτουργία που εκτελεί ο δεκτής, αφού απ' αυτόν εξαρτάται κατά κύριο λόγο ο ρυθμός μετάδοσης. Ο υπολογισμός γίνεται όχι με βάση τον αριθμό των λαθών σε ένα σταθερού μήκους χρονικό παράθυρο, αλλά με βάση τα διαστήματα μεταξύ των πιο πρόσφατων loss events (loss intervals), ώστε να έχουμε μεταβολή της πιθανότητας μόνο όταν σημειωθεί ένα νέο loss event και όχι με τη πάροδο του χρόνου. Συγκεκριμένα υπολογίζεται ο μέσος χρόνος μεταξύ διαδοχικών loss events με βάση τα n τελευταία loss intervals τα οποία συμβάλουν στον μέσο όρο με διαφορετικά βάρη ως εξής :

$$\hat{S}_{(1,n)} = \frac{\sum_{i=1}^n w_i \cdot S_i}{\sum_{i=0}^n w_i}$$

όπου τα βάρη w_i δίνονται από τις σχέσεις

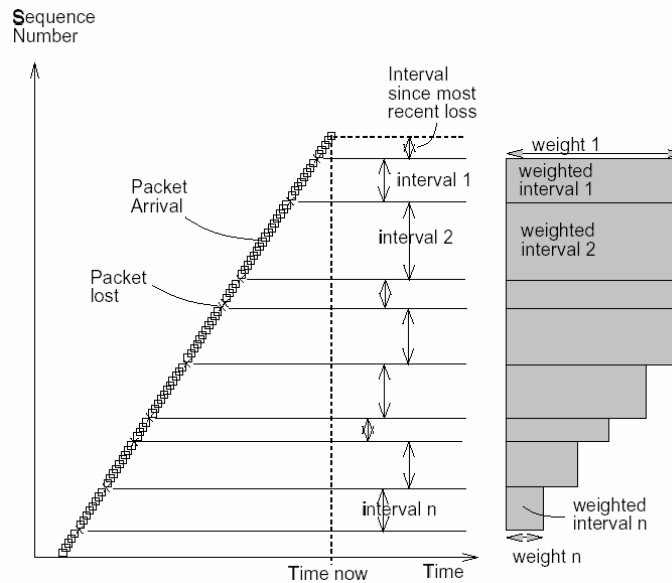
$$w_i = 1, \quad 1 \leq i \leq \frac{n}{2} \quad \text{και} \quad w_i = 1 - \frac{i - \frac{n}{2}}{\frac{n}{2} + 1}, \quad \frac{n}{2} \leq i \leq n$$

Επιλέξαμε τα πρώτα μισά βάρη ίσα, ώστε να αντιμετωπίζονται ισότιμα τα πιο πρόσφατα loss intervals, και τα υπόλοιπα μισά να μειώνονται γραμμικά, ώστε να αποσβένονται παλαιότερες απώλειες. Συνήθως επιλέγεται $n=8$ οπότε τα βάρη κατά σειρά είναι 1, 1, 1, 1, 0.8, 0.6, 0.4 και 0.2. Επίσης δεν υπολογίζεται ως loss interval το διάστημα που πέρασε από τότε που συνέβη το τελευταίο loss event. Έτσι αποφεύγεται το φαινόμενο όπου έχουμε μείωση του p όσο περνά ο χρόνος και ξαφνική αύξηση με ένα νέο loss event ακόμη και αν μακροσκοπικά το p παραμένει σταθερό.

Ο ρυθμός λαθών p με βάση το μέσο loss interval δίνεται απλά ως :

$$p = \frac{1}{\hat{S}}$$

Τα loss intervals και τα αντίστοιχα βάρη που χρησιμοποιούνται για τον υπολογισμό του p



Σύμφωνα και με τα παραπάνω η συμπεριφορά του δεκτή είναι η εξής :

- Όταν λαμβάνει ένα πακέτο δεδομένων
 - Προσθέτει το πακέτο αυτό στο «ιστορικό» των πακέτων που κρατά
 - Υπολογίζει το νέο p
 - Αν το p είναι μεγαλύτερο από την προηγούμενη τιμή του προκαλείται μηδενισμός του feedback timer, αλλιώς δεν ενεργεί περαιτέρω.
- Όταν συμβεί μηδενισμός του feedback timer
 - Εάν έχουν ληφθεί νέα πακέτα δεδομένων μετά το τελευταίο feedback στα οποία θα περιέχεται το RTT όπως υπολογίζεται από τον αποστολέα
 - Υπολογίζεται το p
 - Υπολογίζεται ο ρυθμός X_{recv} με τον οποίο λαμβάνονται δεδομένα μέσα στο τελευταίο RTT
 - Στέλνεται ένα πακέτο feedback
 - Επανεκινείται ο feedback timer ώστε να εκπνεύσει μετά από ένα RTT
 - Εάν δεν έχουν ληφθεί νέα πακέτα δεδομένων μετά το τελευταίο feedback δεν στέλνεται feedback επανεκινείται ο feedback timer με την ίδια τιμή εκπνοής, δηλαδή ενός RTT όπως υπολογίστηκε από το προηγούμενο πακέτο.

- Όταν λάβει το πρώτο πακέτο
 - Θέτει $p = 0$
 - Θέτει $X_{recv} = 0$
 - Ετοιμάζει και στέλνει ένα πακέτο feedback
 - Θέτει το feedback timer ώστε να εκπνεύσει μετά από ένα RTT
- Τέλος όταν ανιχνεύσει το πρώτο loss event μετά το slow start του αποστολέα, υποθέτει πως ο σωστός ρυθμός είναι ίσος με το μισό του ρυθμού μετάδοσης που προκάλεσε το loss και ρυθμίζει κατάλληλα το X_{recv} που επιστρέφει σ' αυτόν. Τότε όμως δεν μπορεί να υπολογιστεί σωστά το πρώτο loss interval, αφού ο ρυθμός μετάδοσης σε αυτό αυξάνεται εκθετικά. Έτσι υπολογίζει το ρυθμό λαθών p που όταν χρησιμοποιηθεί στην εξίσωση υπολογισμού του ρυθμού μετάδοσης θα δώσει ένα αποτέλεσμα ίσο με το X_{recv} (με ένα περιθώριο σφάλματος της τάξης του 5% ώστε να γίνεται εύκολα ο υπολογισμός) και θέτει το πρώτο loss interval ίσο με $\frac{1}{p}$.

3. Ενεργή διαχείριση ουρών (Active Queue Management)

3.1 Αναγκαιότητα για την ύπαρξη AQM στους δρομολογητές.

Στη σημερινή του μορφή το διαδίκτυο αποτελείται από πολλά υποδίκτυα υψηλής ταχύτητας μεταξύ των οποίων όμως μπορεί να υπάρχουν μεγάλες καθυστερήσεις. Οι δρομολογητές του δικτύου, επομένως, θα πρέπει να διαθέτουν ουρές με αρκετά μεγάλο μέγιστο μήκος, ώστε να μπορούν να αποθηκεύσουν παροδικά αυξημένη κίνηση. Τα σχήματα έλεγχου συμφόρησης που χρησιμοποιούνται σήμερα ανιχνεύουν την ύπαρξη συμφόρησης από την απόρριψη κάποιου πακέτου από τους δρομολογητές του δικτύου. Με τις κλασικές ουρές Drop Tail αυτό συμβαίνει μόνο κατά την υπερχειλίση της. Όμως είναι ανεπιθύμητη η ύπαρξη ουρών που είναι πλήρεις για μεγάλο ποσοστό του χρόνου λειτουργίας του δικτύου, για αρκετούς λόγους.

Κατ' αρχήν το μεγάλο μήκος ουράς σημαίνει και μεγάλο χρόνο αναμονής στον δρομολογητή. Σε αυτή την περίπτωση αυξάνεται η μέση καθυστέρηση που βλέπουν οι τελικοί χρηστές του δικτύου. Επιπλέον εάν φτάσουν πολλά πακέτα στον δρομολογητή και η ουρά του είναι σχεδόν πλήρης, τα περισσότερα από αυτά θα χαθούν. Έτσι πηγές των οποίων η κίνηση είναι εκρηκτικής μορφής, δηλαδή αποτελείται από σύντομες περιόδους όπου μεταδίδονται πολλά πακέτα ακολουθούμενες από περιόδους αδράνειας, έχουν υψηλό ρυθμό απωλειών πακέτων και άρα ο ρυθμός μετάδοσής τους μειώνεται δραστικά. Επίσης σε ορισμένες περιπτώσεις οι Drop Tail ουρές μπορούν να οδηγήσουν σε αποκλεισμό ορισμένων συνδέσεων, ως αποτέλεσμα αλληπάλληλων απωλειών του πακέτου εγκαθίδρυσης της σύνδεσης.

Ένα άλλο φαινόμενο που παρατηρείται στις Drop Tail ουρές είναι αυτό του συγχρονισμού των παραθύρων συμφόρησης πολλαπλών συνδέσεων TCP. Πακέτα απορρίπτονται μόνο όταν υπερχειλίσει η ουρά. Λόγω της στατιστικής πολυπλεξίας που παρατηρείται στο διαδίκτυο, τα πακέτα από διάφορες ροές πολυπλέκονται μεταξύ τους. Έτσι στις περιόδους όπου η ουρά είναι γεμάτη, απορρίπτονται πακέτα από πολλές συνδέσεις. Αυτό έχει ως αποτέλεσμα όλες αυτές οι συνδέσεις να ρίχνουν ταυτόχρονα το παράθυρο τους, οπότε σε αυτή την περίοδο υπάρχει μείωση της

συνολικής προσφερόμενης κίνησης στο δίκτυο που μπορεί να οδηγήσει σε άδειασμα των ουρών και υποχρησιμοποίηση των ζεύξεων. Επιπλέον οι συνδέσεις αυτές αυξάνουν το παράθυρο τους με τον ίδιο τρόπο. Έτσι η προσφερόμενη κίνηση αυξάνεται με ρυθμό ανάλογο του αριθμού των συνδέσεων αυτών, γεγονός το οποίο προκαλεί αυξημένη συμφόρηση και υπερχειλίση της ουράς. Οδηγούμαστε έτσι σε μια ατέρμονη εναλλαγή περιόδων υψηλής συμφόρησης και περιόδων χαμηλής χρησιμοποίησης του δικτύου.

Για την αποφυγή αυτών των φαινομένων χρειάζεται η απόρριψη πακέτων από τους δρομολογητές πριν οι ουρές τους υπερχειλίσουν, ώστε να στέλνονται σήματα συμφόρησης στους χρηστές για να μειώσουν τον ρυθμό μετάδοσης τους. Αυτή η τεχνική ονομάζεται Ενεργή Διαχείριση Ουρών (Active Queue Management, AQM). Έτσι οι δρομολογητές έχουν τη δυνατότητα να ελέγχουν το ποια και πόσα πακέτα θα χάνονται.

Η πρόωρη απόρριψη των πακέτων δεν σημαίνει πως αυξάνεται και ο συνολικός αριθμός των χαμένων πακέτων. Η κίνηση που παρατηρείται στο διαδίκτυο είναι συχνά εκρηκτικής μορφής. Εάν οι ουρές των δρομολογητών είναι πλήρεις ή σχεδόν πλήρεις τότε δεν μπορούν να απορροφηθούν τέτοιες ριπές και τα πακέτα χάνονται. Διατηρώντας το μήκος της ουράς χαμηλά, οι αλγόριθμοι AQM εξασφαλίζουν ότι θα υπάρχει χώρος ώστε να απορροφηθούν όλα τα πακέτα χωρίς να χαθούν.

Επιπλέον με τον έλεγχο των πακέτων που απορρίπτονται μπορεί να διασφαλισθεί ότι δεν θα υπάρχει μια ριπή χαμένων πακέτων, από την οποία το TCP ανακάμπτει πιο δύσκολα απ' ότι στην περίπτωση μιας μόνο απώλειας πακέτου.

3.2 Random Early Detection (RED)

Ο αλγόριθμος Τυχαίας Έγκαιρης Ανίχνευσης (RED) είναι ένας αλγόριθμος ενεργής διαχείρισης ουρών για δρομολογητές, που σχεδιάστηκε για να προσφέρει αυξημένη απόδοση στο διαδίκτυο για ένα μεγάλο εύρος περιπτώσεων. Σε αντίθεση με τους κλασσικούς αλγορίθμους διαχείρισης ουρών, στους οποίους απορρίπτονται πακέτα μόνο όταν η ουρά είναι πλήρης, ο αλγόριθμος RED απορρίπτει πακέτα με βάση κάποια πιθανότητα. Η πιθανότητα αυξάνεται όσο αυξάνεται το μέσο μήκος της ουράς.

Χρησιμοποιείται το μέσο μήκος ουράς και όχι το στιγμιαίο, έτσι ώστε ο αλγόριθμος να αντιδρά με βάση την μακροσκοπική κατάσταση του δικτύου. Έτσι εάν η ουρά ήταν σχετικά άδεια, μια ριπή πακέτων δεν προκαλεί απώλειες πακέτων, ενώ αν η ουρά είναι σχετικά γεμάτη για μεγάλο χρονικό διάστημα, τα εισερχόμενα πακέτα θα έχουν μεγάλη πιθανότητα να απορριφθούν, ώστε να ειδοποιηθούν οι χρηστές για την επίμονη συμφόρηση.

Ο ίδιος ο αλγόριθμος RED αποτελείται δυο μέρη : την εκτίμηση του μέσου μήκους ουράς και την απόφαση για το αν κάποιο εισερχόμενο πακέτο θα πρέπει να απορριφθεί ή όχι.

Η εκτίμηση του μέσου μήκους της ουράς γίνεται χρησιμοποιώντας ένα βαθυπερατό φίλτρο με έναν αλγόριθμο Κινουμένου Μέσου Εκθετικού Βάρους (Exponential Weighted Moving Average, EWMA). Συγκεκριμένα με την άφιξη ενός πακέτου γίνεται ο υπολογισμός : $avg = (1 - w_q) avg + w_q \cdot q$, όπου q το στιγμιαίο μήκος ουράς, avg το μέσο μήκος ουράς και w_q το βάρος του αλγορίθμου EWMA.

Αυτός ο υπολογισμός, όμως, θα πρέπει να λαμβάνει, υπ' όψιν τυχόν περιόδους όπου η ουρά είναι άδεια. Έτσι υπολογίζεται ο αριθμός m των πακέτων που θα μπορούσαν να είχαν μεταδοθεί στην περίοδο που η ουρά είναι άδεια :

$m = \frac{time - q_time}{s}$, όπου $time$ ο τρέχον χρόνος, q_time ο χρόνος στον οποίο άδειασε

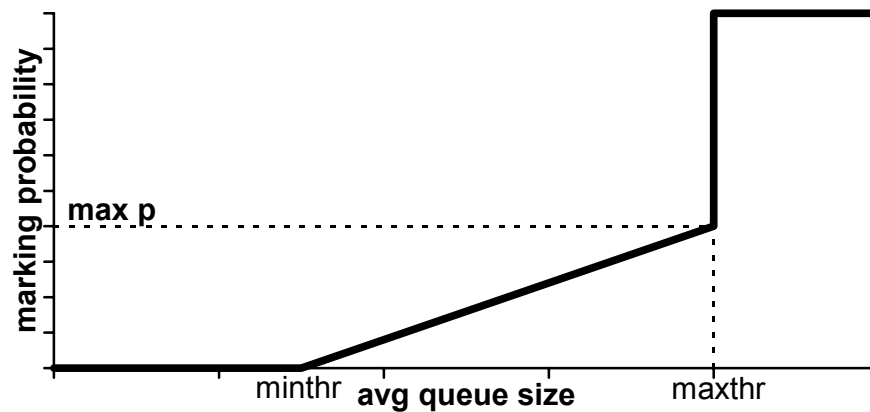
η ουρά και s ο μέσος χρόνος που χρειάζεται για τη μετάδοση ενός πακέτου. Έπειτα το μέσο μήκος ουράς υπολογίζεται σαν να είχαν φτάσει m πακέτα σε άδεια ουρά κατά την διάρκεια αυτής της περιόδου, δηλαδή $avg = (1 - w_q)^m \cdot avg$

Η απόφαση για την απόρριψη ή μη ενός εισερχόμενου πακέτου βασίζεται σε μια υπολογιζόμενη πιθανότητα p_m . Τρεις παράμετροι καθορίζουν κυρίως την υπολογιζόμενη αυτή πιθανότητα το ελάχιστο και το μέγιστο κατώφλι (min_{thr} και max_{thr}) και η μέγιστη πιθανότητα απόρριψης (max_p). Το ελάχιστο κατώφλι είναι το μέσο μήκος ουράς κατώφλι από το οποίο δεν θα απορρίπτεται κανένα πακέτο, ενώ το μέγιστο εκφράζει το μέσο μήκος ουράς πάνω από το οποίο όλα τα εισερχόμενα πακέτα θα απορρίπτονται. Όταν το μέσο μήκος ουράς βρίσκεται μεταξύ των δυο κατωφλίων τότε η πιθανότητα μεταβάλλεται γραμμικά από 0 έως p_{max} . Επομένως η πιθανότητα απόρριψης πακέτου δίνεται από τον τύπο :

$$p = \begin{cases} 0 & q_{ave} < min_{thr} \\ \frac{q_{ave} - min_{thr}}{max_{thr} - min_{thr}} \cdot p_{max} & min_{thr} < q_{ave} < max_{thr} \\ 1 & q_{ave} > max_{thr} \end{cases}$$

Πιθανότητα απόρριψης πακέτου του RED συναρτήσεως του μέσου μήκους ουράς

RED Marking Probability



Η πιο απλή υλοποίηση της απόφαση για την απόρριψη ή μη ενός πακέτου είναι η παραγωγή ενός τυχαίου αριθμού στο διάστημα $[0,1]$ και η σύγκριση του με το p_m , αν είναι μικρότερος από p_m το πακέτο απορρίπτεται, ενώ σε διαφορετική περίπτωση όχι. Επειδή όμως σε ορισμένες περιπτώσεις η γεννήτρια ψευδοτυχαίων αριθμών μπορεί να δώσει αρκετούς συνεχόμενους αριθμούς που να είναι μεγαλύτεροι από το p_m και να καθυστερήσει πολύ να γίνει απόρριψη ενός πακέτου η τελική πιθανότητα αυξάνεται ανάλογα με τον αριθμό των πακέτων count που έχουν φτάσει

μετά την τελευταία απόρριψη ως εξής : $p_m = \frac{p}{1 - count \cdot p}$. Έτσι διασφαλίζεται ότι δεν

θα περιμένουμε υπερβολικά πριν την απόρριψη ενός πακέτου.

Επίσης θα πρέπει να αποφεύγεται η απόρριψη διαδοχικών πακέτων, που πιθανότατα ανήκουν στην ίδια ροή, ώστε ο αλγόριθμος έλεγχου συμφόρησης να μην χρειάζεται να ανακάμψει από πολλαπλές απώλειες, που όπως είδαμε αποτελούν πρόβλημα για συνδέσεις TCP. Έτσι ο δρομολογητής θα πρέπει να περιμένει για κάποιο μικρό διάστημα μεταξύ διαδοχικών απορρίψεων πακέτων.

Τα παραπάνω μπορούν να συνοψισθούν στον εξής αλγόριθμο :

- Αρχικοποίηση
 - avg = 0
 - count = -1
- Με την άφιξη κάθε πακέτου
 - υπολογισμός του μέσου μήκους ουράς
 - Αν η ουρά δεν είναι άδεια : $avg = (1 - w_q)avg + w_q \cdot q$
 - Αν είναι άδεια : $m = \frac{time - q_time}{s}$, $avg = (1 - w_q)^m \cdot avg$
 - Αν $avg > maxth$
 - απόρριψη του πακέτου
 - Μηδενισμός του count
 - Αν $minthr < avg < maxthr$
 - Αύξηση του count
 - υπολογισμός της πιθανότητας απόρριψης
 - $p = \frac{q_ave - min_thr}{max_thr - min_thr} \cdot p_max$
 - $p_m = \frac{p}{1 - count \cdot p}$
 - επιλογή τυχαίου αριθμού $x \in [0.1]$
 - Αν $x > p_m$
 - απόρριψη του πακέτου
 - Μηδενισμός του count
 - Αν $avg < minthr$
 - count = -1

- Όταν αδειάσει η ουρά
 - $q_time = time$

Ο παραπάνω αλγόριθμος υποθέτει ότι το μήκος ουράς μετριέται σε πακέτα. Σύμφωνα με μια διαφορετική προσέγγιση, το μήκος της ουράς μετριέται σε bytes και η πιθανότητα απόρριψης κάθε πακέτου είναι ανάλογη του μεγέθους του. Με αυτόν τον τρόπο ευνοούνται μικρά πακέτα, όπως είναι τα πακέτα έλεγχου και τα πακέτα telnet, έναντι μεγαλύτερων πακέτων όπως είναι τα FTP πακέτα. Σε αυτή την περίπτωση η πιθανότητα απόρριψης για κάθε πακέτο θα υπολογίζεται ως εξής :

$$p = \frac{q_ave - \min_thr}{\max_thr - \min_thr} \cdot p_max$$

$$p = p \frac{Packet\ Size}{Maximum\ Packet\ Size}$$

$$p_m = \frac{p}{1 - count \cdot p}$$

όπου Packet Size το μέγεθος του υπ' όψιν πακέτου και Maximum Packet Size το μέγιστο μέγεθος πακέτου.

Τέλος ο καθορισμός των παραμέτρων του αλγορίθμου RED θα πρέπει να γίνεται σύμφωνα με τα ακόλουθα κριτήρια :

- Το βάρος w_q του αλγορίθμου EWMA για τον υπολογισμό του μέσου μήκους ουράς εκφράζει την χρονική απόκριση του αλγορίθμου. Εάν η τιμή του είναι πολύ χαμηλή τότε το μέσο μήκος ουράς αλλάζει υπερβολικά αργά σε σχέση με το στιγμιαίο μήκος και ο αλγόριθμος δε μπορεί να διακρίνει τα αρχικά στάδια της συμφόρησης. Απ' την άλλη εάν το w_q είναι πολύ μεγάλο τότε μεταβατική συμφόρηση, όπως είναι η κίνηση εκρηκτικής μορφής, θα κόβεται από τον αλγόριθμο.
- Το ελάχιστο κατώφλι \min_{thr} και το μέγιστο κατώφλι \max_{thr} εκφράζουν τα όρια μέσα στα οποία επιτρέπεται να βρίσκεται το μέσο μήκος ουράς. Το \min_{thr} θα πρέπει να είναι αρκετά μεγάλο, ώστε η ουρά να έχει πάντα δεδομένα προς αποστολή, οπότε επιτυγχάνεται υψηλή χρησιμοποίηση. Επίσης το \max_{thr} θα πρέπει να είναι αρκετά χαμηλό ώστε η καθυστέρηση να είναι μικρή, αλλά και να υπάρχει διαθέσιμος χώρος για τυχόν εκρηκτική κίνηση.

3.3 Random Exponential Marking (REM)

Ο αλγόριθμος Τυχαίας Εκθετικής Επισήμανσης (REM) είναι ένα σχήμα ενεργής διαχείρισης ουρών που έχει ως στόχο την ταυτόχρονη επίτευξη υψηλής χρησιμοποίησης και αμελητέας απώλειας πακέτων και καθυστέρησης, με έναν απλό και εύκολα κλιμακούμενο τρόπο. Ως επισήμανση εννοείται η απόρριψη ενός πακέτου ή το να τεθεί το ECN bit της επικεφαλίδας του πακέτου αυτού, ως ένδειξη συμφόρησης. Στο πλαίσιο της εργασίας αυτής δεν χρησιμοποιήθηκε η δυνατότητα του ECN, επομένως ως marking εννοείται η απόρριψη του πακέτου. Η βασική ιδέα είναι η αποσυσχέτιση των μέτρων συμφόρησης από τα μέτρα απόδοσης, όπως το μήκος ουράς ή η καθυστέρηση. Τα μέτρα συμφόρησης δηλώνουν υψηλή απαίτηση για εύρος ζώνης και θα πρέπει να αντικατοπτρίζουν τον αριθμό των χρηστών, σε αντίθεση με τα μέτρα απόδοσης τα οποία θα πρέπει να σταθεροποιούνται στις επιθυμητές τιμές, ανεξάρτητα από τον αριθμό των χρηστών.

Οι βασικές επιδιώξεις του αλγορίθμου REM είναι :

- Η εξίσωση του συνολικού ρυθμού μετάδοσης των χρηστών με την χωρητικότητα του δικτύου ενώσω το μήκος της ουράς παραμένει σταθεροποιημένο γύρω από μια επιθυμητή τιμή, ανεξάρτητα από τον αριθμό των χρηστών.
- Η εξάρτηση της συνολικής πιθανότητας απόρριψης πακέτου σε όλη την διαδρομή, από το άθροισμα του κόστους (μέτρο συμφόρησης) σε κάθε επιμέρους ζεύξη.

Η πρώτη επιδίωξη υποδηλώνει πως, σε αντίθεση με την συμβατική θεώρηση, η υψηλή χρησιμοποίηση δεν επιτυγχάνεται κρατώντας στην ουρά κάποιον σημαντικό αριθμό πακέτων, ώστε να υπάρχουν συνεχώς δεδομένα προς αποστολή, αλλά με την παροχή της κατάλληλης πληροφορίας στους χρηστές, ώστε να προσαρμόσουν κατάλληλα τους ρυθμούς μετάδοσής τους.

Η δεύτερη επιδίωξη είναι απαραίτητη σε δίκτυα όπου οι συνδέσεις διέρχονται από περισσότερες από μια υπό συμφόρηση ζεύξεις, οπότε η πληροφορία που παρέχεται στους χρηστές θα πρέπει να αντικατοπτρίζει την αθροιστική συμφόρηση στο δίκτυο.

Αυτές οι επιδιώξεις είναι σε μεγάλο βαθμό ανεξάρτητες μεταξύ τους. Κατά συνέπεια ο αλγόριθμος REM μπορεί να υλοποιηθεί από δυο ανεξάρτητες μεταξύ τους λειτουργικές μονάδες : την μονάδα υπολογισμού του μέτρου της συμφόρησης και την μονάδα υπολογισμού της τελικής πιθανότητας.

3.3.1 Συνάρτηση κόστους

Η πρώτη επιδίωξη επιτυγχάνεται με την χρήση μιας συνάρτησης κόστους για κάθε ζεύξη (price) ως μέτρο συμφόρησης. Το κόστος αυτό χρησιμοποιείται για τον μετέπειτα υπολογισμό της πιθανότητας απόρριψης πακέτου. Η συνάρτηση κόστους εξαρτάται από τη διάφορα μεταξύ του συνολικού ρυθμού μετάδοσης των χρηστών και από την διάφορα του μήκους της ουράς από την επιθυμητή τιμή αυτού. Το κόστος αυξάνεται όταν το σταθμισμένο άθροισμα των διαφορών αυτών είναι θετικό και μειώνεται σε διαφορετική περίπτωση.

Όσο αυξάνεται ο αριθμός των χρηστών οι διαφορές του ρυθμού και του μήκους ουράς αυξάνονται, οπότε αυξάνεται και το κόστος της ζεύξης. Αυτό προκαλεί την αποστολή ισχυρότερων σημάτων συμφόρησης (δηλαδή περισσότερων χαμένων πακέτων) στους χρηστές, οι οποίοι μειώνουν ανάλογα τους ρυθμούς μετάδοσής τους. Όταν οι ρυθμοί αυτοί είναι χαμηλοί οι παραπάνω διαφορές είναι αρνητικές, μειώνοντας το κόστος και άρα το πλήθος των απορριπτόμενων πακέτων, οπότε αυξάνονται οι ρυθμοί μετάδοσης. Με αυτό τον τρόπο οι διαφορές τείνουν να μηδενιστούν, οπότε επιτυγχάνεται υψηλή χρησιμοποίηση και μικρός αριθμός χαμένων πακέτων. Αν επιπλέον το επιθυμητό μήκος ουράς επιλεγεί να είναι μικρό η καθυστέρηση θα είναι εξίσου μικρή.

Το κόστος της ζεύξης ενημερώνεται περιοδικά και όχι με κάθε άφιξη πακέτων, ώστε να μειώνεται ο αριθμός των απαιτούμενων υπολογισμών και άρα η υπολογιστική επιβάρυνση του δρομολογητή. Συγκεκριμένα για μια ζεύξη l (link) το κόστος p_l υπολογίζεται με βάση τον εξής τύπο :

$$p_l(t+1) = \max \left\{ 0, p_l(t) + \gamma \left[a_l \cdot (b_l(t) - b_l^*) + x_l(t) - c_l(t) \right] \right\}$$

όπου a_i και γ μικρές θετικές σταθερές, $b_i(t)$ το μήκος της ουράς, b_i^* το επιθυμητό μήκος ουράς, $x_i(t)$ ο συνολικός ρυθμός της εισερχόμενης κίνησης και $c_i(t)$ η χωρητικότητα της ζεύξης σε κάθε περίοδο ενημέρωσης.

Σε αυτή τη σχέση η διάφορα $x_i(t) - c_i(t)$ εκφράζει τη διάφορα ρυθμού εισόδου - χωρητικότητας ενώ η διάφορα $b_i(t) - b_i^*$ τη διάφορα τρέχοντος και επιθυμητού μήκους ουράς.

Η σταθερά a_i εκφράζει το βάρος που έχει στην παραπάνω συνάρτηση η διάφορα του μήκους ουράς σε σχέση με τη διάφορα των ρυθμών και κατ' επέκταση μπορεί να χρησιμοποιηθεί για το trade-off μεταξύ χρησιμοποίησης της ζεύξης και καθυστέρησης στην ουρά. Μικρότερες τιμές της παραμέτρου επιτυγχάνουν μεγαλύτερη χρησιμοποίηση. Συνίσταται μια τιμή $a_i=0.1$.

Η σταθερά γ εκφράζει την απόκριση του αλγορίθμου στις μεταβολές στις συνθήκες του δικτύου και την ταχύτητα σύγκλισής του. Μια μεγάλη τιμή του γ σημαίνει ταχύτερη σύγκλιση, όμως μπορεί να προκαλέσει σημαντικές διακυμάνσεις στο μήκος της ουράς. Πολύ μικρές τιμές του γ όμως μπορεί να μην επιτρέψουν στον αλγόριθμο την έγκαιρη παρακολούθηση των αλλαγών. Συνίσταται το γ να κυμαίνεται μεταξύ 0.001 και 0.01.

Επειδή το υπό βάρη άθροισμα των διαφορών είναι πιθανόν να είναι αρνητικό και η προηγούμενη τιμή του κόστους της ζεύξης να είναι μικρότερη κατ' απόλυτη τιμή από αυτό το άθροισμα, μπορεί να προκύψει αρνητικό κόστος. Για να αποφύγουμε λοιπόν αυτή την κατάσταση είναι απαραίτητο να θέσουμε ως κάτω φράγμα του κόστους το μηδέν.

Στην πράξη δεν είναι τόσο εύκολη η μέτρηση των ρυθμών εισερχόμενων και εξερχόμενων πακέτων, όσο είναι η μέτρηση του μήκους της ουράς. Όμως η διάφορα $x_i(t) - c_i(t)$ εκφράζει τον ρυθμό αύξησης του μήκους της ουράς και επομένως μπορεί να προσεγγιστεί από την σχέση $b_i(t+1) - b_i(t)$. Μπορούμε, λοιπόν, να υπολογίσουμε το κόστος της ζεύξης βασιζόμενοι μόνο στην τρέχουσα και την προηγούμενη τιμή του μήκους ουράς σύμφωνα με τη σχέση :

$$p_i(t+1) = \max \left\{ 0, p_i(t) + \gamma \left[b_i(t+1) - (1 - a_i) \cdot b_i(t) - a_i \cdot b_i^* \right] \right\}$$

Ο υπολογισμός του κόστους κάθε ζεύξης βασίζεται σε πληροφορίες που αφορούν μόνο τη ζεύξη αυτή και όχι το υπόλοιπο δίκτυο. Επίσης χρησιμοποιείται η εκτίμηση της συνολικής κίνησης, χωρίς ανά σύνδεση πληροφορία. Επομένως η πολυπλοκότητα του είναι ανεξάρτητη από το μέγεθος του δικτύου ή τον αριθμό των χρηστών και άρα μπορεί να χρησιμοποιηθεί εύκολα σε κάθε κλίμακα.

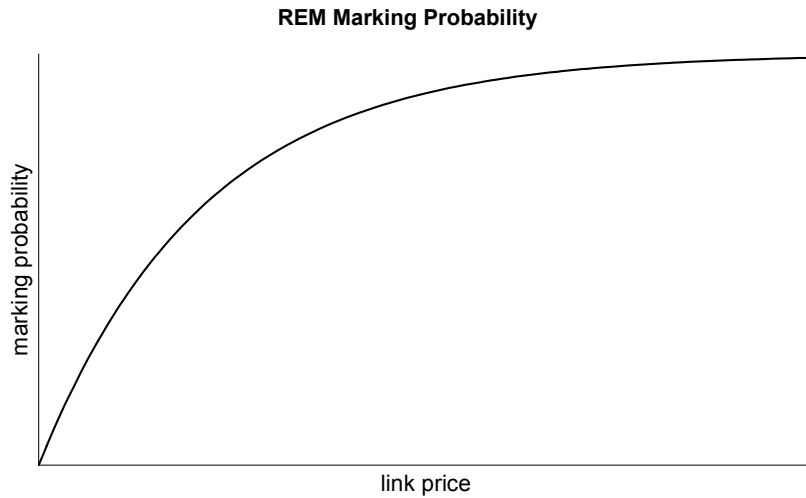
3.3.2 Πιθανότητα απόρριψης

Ο δεύτερος στόχος του REM επιτυγχάνεται με την κατάλληλη επιλογή της συνάρτησης για την πιθανότητα απόρριψης. Θέλουμε η συνολική πιθανότητα απόρριψης πακέτου να εξαρτάται από το άθροισμα του κόστους όλων των ζεύξεων της διαδρομής που ακολουθούν τα πακέτα, δηλαδή : $m_{TOTAL} = f(\sum p_i)$, όπου m η πιθανότητα απόρριψης πακέτου και p_i το κόστος ζεύξης. Η απόρριψη των πακέτων σε κάθε ζεύξη όμως γίνεται ανεξάρτητα από τις υπόλοιπες. Επομένως πρόκειται για ανεξάρτητες μεταξύ τους στοχαστικές διαδικασίες και άρα η συνολική πιθανότητα απόρριψης είναι το γινόμενο των επιμέρους πιθανοτήτων, δηλαδή : $m_{TOTAL} = \prod m_i$. Θέλουμε επομένως μια συνάρτηση $m_i = f(p_i)$, με την ιδιότητα : $\prod f(x) = f(\sum x)$.

Αυτή την ιδιότητα την έχουν μόνο οι εκθετικές συναρτήσεις και έτσι επιλέγουμε :

$$m_i = 1 - \phi^{-p_i} \quad \text{με } \phi > 1$$

πιθανότητα απόρριψης πακέτου συναρτήσει του κόστους ζεύξης



Η παράμετρος ϕ καθορίζει το διάστημα τιμών της πιθανότητας απόρριψης, σε σχέση βέβαια και με τις τιμές του κόστους της ζεύξης. Ιδανικά το ϕ θα πρέπει να επιλέγεται έτσι ώστε η πιθανότητα που παρατηρείται στη συνολική διαδρομή να κυμαίνεται γύρω στο 0.5 % . Όμως ο αλγόριθμος AIMD του TCP και των σύμφωνων με αυτό αλγορίθμων έλεγχου συμφόρησης απαιτεί η πιθανότητα να είναι μικρή, γύρω στο 0.1 % . Γι αυτό το λόγο επιλέγεται μια τιμή $\phi=1.001$.

Όταν οι πιθανότητες είναι χαμηλές και άρα και το κόστος κάθε ζεύξης είναι χαμηλό, η συνολική πιθανότητα απόρριψης πακέτων είναι κατά προσέγγιση ανάλογη με το άθροισμα του κόστους των επιμέρους ζεύξεων και συγκεκριμένα :

$$m_{TOTAL} \approx (\ln \phi) \sum p_l$$

3.4 Ο αλγόριθμος RIO

Οι κλασσικοί αλγόριθμοι ενεργής διαχείρισης ουρών προϋποθέτουν ότι τα πακέτα που καταφθάνουν στις ουρές προέρχονται από ροές που αποκρίνονται στα χαμένα πακέτα μειώνοντας το ρυθμό μετάδοσής τους. Όταν όμως υπάρχουν και μη αποκρινόμενες ροές, που μεταδίδουν με σταθερό ρυθμό ανεξάρτητα από την κατάσταση του δικτύου, δεν λειτουργούν σωστά. Οι μη αποκρινόμενες ροές πλημμυρίζουν την ουρά, προκαλώντας την απόρριψη των περισσότερων από τα εισερχόμενα πακέτα ανεξάρτητα από την προέλευση τους. Αυτό μπορεί να συμβεί όμως και με ροές που χρησιμοποιούν κάποιον έλεγχο συμφόρησης, όμως αυτός είναι περισσότερο επιθετικός στην αναζήτηση του διαθέσιμου εύρους ζώνης απ' ό,τι οι κλασσικοί αλγόριθμοι όπως το TCP. Έτσι χάνονται πολλά πακέτα και από τις ροές που υλοποιούν έλεγχο συμφόρησης, με αποτέλεσμα αυτές να ρίχνουν πολύ χαμηλά τον ρυθμό μετάδοσής τους και να παίρνουν πολύ μικρό ποσοστό της διαθέσιμης χωρητικότητας. Υπάρχει δηλαδή έλλειψη δικαιοσύνης.

Επίσης σε πολλές περιπτώσεις είναι επιθυμητό ορισμένες ροές να έχουν κάποιον εξασφαλισμένο ρυθμό μετάδοσης ακόμη και σε συνθήκες υψηλής συμφόρησης. Τέτοιες ροές μπορεί να προέρχονται είτε από κρίσιμες για το δίκτυο εφαρμογές, όπως είναι διαχειριστικά εργαλεία, είτε από χρηστές που θέλουν εξασφαλισμένη ποιότητα υπηρεσίας από το δίκτυο και χρεώνονται ανάλογα γι αυτή. Με την συμβατική σχεδίαση βέλτιστης προσπάθειας (best effort) του διαδικτύου δεν είναι δυνατό να προσφερθούν τέτοιες υπηρεσίες, αφού η κάθε ροή καταλαμβάνει το βέλτιστο εύρος ζώνης που μπορεί να επιτύχει, το οποίο μπορεί να είναι πολύ χαμηλό σε συνθήκες συμφόρησης.

Για την επίλυση του θέματος της κατά προβλέψιμο τρόπο κατανομής των πόρων του δικτύου χρησιμοποιούνται μέθοδοι που κάνουν ανά ροή προγραμματισμό της μετάδοσής των πακέτων στους δρομολογητές (per flow scheduling). Τα πακέτα κατηγοριοποιούνται με βάση την ροή στην οποία ανήκουν και αντιμετωπίζονται με διαφορετικό τρόπο από τον δρομολογητή. Ένα παράδειγμα τέτοιας μεθόδου είναι η διατήρηση μιας ξεχωριστής ουράς για κάθε ροή και η μετάδοση των πακέτων κάθε ουράς με βάση κάποιο σχήμα καταμερισμού χρόνου (π.χ. round robin). Για να λειτουργήσουν αυτές οι μέθοδοι, όμως, απαιτείται μεγάλος αριθμός υπολογισμών

καθώς και μέγεθος μνήμης στους δρομολογητές, ανάλογο του αριθμού των διαφορετικών ροών. Σε μεγάλη κλίμακα επομένως, όπως είναι το διαδίκτυο, οι μέθοδοι αυτές είναι ασύμφορες.

Σε αντίθεση με το per flow scheduling, ο αλγόριθμος RIO (RED with IN/OUT bit) αποτελεί μια εύκολα κλιμακούμενη λύση στο πρόβλημα της έλλειψης fairness. Συγκεκριμένα μεταθέτει το πρόβλημα της κατηγοριοποίησης της εισερχόμενης κίνησης σε ξεχωριστές οντότητες του δικτύου και αφήνει για τον δρομολογητή μόνο το θέμα της διαχείρισης της μιας και μοναδικής ουράς του με τρόπο που βασίζεται στον αλγόριθμο RED, όπως δηλώνεται και από το όνομα του .

3.4.1 Κατηγοριοποίηση πακέτων

Σύμφωνα με τον σχεδιασμό του RIO απαιτείται η διάκριση των πακέτων που διακινούνται μέσα σε ένα δίκτυο σε πακέτα υψηλής και σε πακέτα χαμηλότερης προτεραιότητας (IN και OUT packets αντίστοιχα). Για τον σκοπό αυτό απαιτείται ένα και μόνο bit το οποίο τίθεται στην τιμή «1» για τα πακέτα υψηλής προτεραιότητας ενώ για τα υπόλοιπα έχει την τιμή «0». Όταν τα πακέτα είναι TCP μπορούμε να χρησιμοποιήσουμε το priority bit της επικεφαλίδας κάθε πακέτου, ενώ αντίστοιχα bit χρησιμοποιούνται και για διαφορετικά πρωτόκολλα μεταφοράς. Επομένως η διάκριση μεταξύ των πακέτων IN και OUT γίνεται με απλό τρόπο και με μηδενική επιβάρυνση στις επικεφαλίδες.

Σε κάθε ροή δεδομένων ανατίθεται κάποιος συγκεκριμένος Επιθυμητός Ρυθμός Μετάδοσης με τον οποίο θα πρέπει να συμμορφώνονται σε περιόδους συμφόρησης. Τα πακέτα που μεταδίδονται σύμφωνα με αυτό τον ρυθμό χαρακτηρίζονται ως IN ενώ τυχόν επιπλέον πακέτα χαρακτηρίζονται ως OUT. Σε περιόδους όπου παρατηρείται συμφόρηση στο δίκτυο τα OUT πακέτα απορρίπτονται με μεγαλύτερη πιθανότητα από τα IN. Έτσι εξασφαλίζεται ότι όταν υπάρχει αυξημένο φορτίο στο δίκτυο, οι διάφορες ροές θα παίρνουν σε γενικές γραμμές το ποσοστό της χωρητικότητας που αντιστοιχεί στον επιθυμητό ρυθμό μετάδοσης που τους έχει ανατεθεί.

Ο μηχανισμός σύμφωνα με τον οποίο γίνεται η επισήμανση των πακέτων δεν βασίζεται στον στιγμιαίο αλλά στον μέσο ρυθμό μετάδοσης. Εάν χρησιμοποιούσαμε τον στιγμιαίο ρυθμό τότε στις TCP συνδέσεις των οποίων το παράθυρο συμφόρησης αυξομειώνεται περιοδικά, ακόμη και αν ο μέσος ρυθμός τους είναι σύμφωνος με τον προδιαγραφόμενο, θα χαρακτηρίζονταν ως OOT σχεδόν τα μισά πακέτα. Έτσι χρησιμοποιείται ένας αλγόριθμος χρονικά κυλιόμενου παραθύρου (time-sliding window, TSW) για την εκτίμηση του μέσου ρυθμού μετάδοσης. Υπολογίζεται ο μέσος ρυθμός μετάδοσης με κάθε εισερχόμενο πακέτο με βάση με ένα παράθυρο των προηγούμενων τιμών του, το οποίο δεν έχει σταθερό μέγεθος σε πακέτα αλλά σε χρόνο. Με αυτόν το τρόπο εξασφαλίζεται ότι μια γρηγορότερη σύνδεση δεν «ξεχνά» τις προηγούμενες τιμές πιο γρήγορα από μια σύνδεση χαμηλότερης ταχύτητας και το ότι μια σύνδεση που δεν στέλνει πακέτα θα τις «ξεχνά» με το χρόνο, το οποίο δε μπορεί να επιτευχθεί με έναν απλό EWMA αλγόριθμο.

Είναι σημαντικό το ότι τα πακέτα πλέον του επιθυμητού ρυθμού δεν απορρίπτονται εξ' αρχής από το δίκτυο. Έτσι σε περίπτωση που δεν υπάρχει συμφόρηση οι ροές μπορούν να μεταδώσουν με αρκετά υψηλότερο ρυθμό από αυτόν που τους έχει ανατεθεί. Με αυτό τον τρόπο διασφαλίζεται αφ' ενός ότι δεν θα αποτρέπονται άσκοπα οι ροές από το να μεταδώσουν όλα τα διαθέσιμα πακέτα τους όταν δεν υπάρχει συμφόρηση και αφ' εταίρου ότι δεν θα υποχρησιμοποιείται το δίκτυο, λόγω που περιορισμού των ρυθμών μετάδοσης ακόμη και αν το άθροισμα τους είναι μικρότερο της διαθέσιμης χωρητικότητας.

Οι agents που εκτελούν τον διαχωρισμό των πακέτων σε IN και OOT μπορούν να βρίσκονται σε οποιοδήποτε σημείο του δικτύου. Μπορούν για παράδειγμα να βρίσκονται στα σημεία όπου συνδέονται οι χρηστές στο δίκτυο δηλαδή στους παροχείς δικτύου (ISPs), ώστε να ελέγχεται η κίνηση για κάθε χρηστή ξεχωριστά, ή ακόμη και στα όρια αυτόνομων συστημάτων (Autonomous Systems), για να καθορίζεται η κίνηση μεταξύ τους. Επίσης μπορεί ο έλεγχος για την συμμόρφωση με τον επιθυμητό ρυθμό μετάδοσης να γίνεται είτε από τον αποστολέα της κίνησης είτε από τον παραλήπτη της. Στο διαδίκτυο σήμερα το μεγαλύτερο ποσοστό της κίνησης είναι από εξυπηρετητές (servers) προς τους τελικούς χρήστες (end users), επομένως ως αποστολείς εννοούνται οι εξυπηρετητές ενώ ως παραλήπτες οι απλοί χρήστες του δικτύου. Επομένως, όταν έχουμε έλεγχο από τον αποστολέα, η ανάθεση πόρων σε κάθε χρήστη γίνεται από μια ανεξάρτητη αρχή σύμφωνα με προκαθορισμένο τρόπο,

ενώ όταν ο έλεγχος γίνεται από τον χρήστη μπορεί να έχουμε δυναμική ανάθεση πόρων, ανάλογα με τις εκάστοτε ανάγκες του χρήστη και με αντίστοιχη χρέωση.

3.4.2 Διαχείριση Ουράς

Οι δρομολογητές του δικτύου επιφορτίζονται μόνο με το κομμάτι της διαχείρισης της ουράς τους. Όπως είδαμε η διαφοροποίηση δεν γίνεται ανά ροή δεδομένων, αλλά ανά κατηγορία πακέτων. Έτσι χρειάζεται να έχουμε διαχωρισμό των εισερχομένων πακέτων μόνο ως προς το αν είναι IN ή OUT. Αυτό ο αλγόριθμος του RIO το επιτυγχάνεται με μια και μοναδική FIFO ουρά, δηλαδή μια ουρά στην οποία η εξυπηρέτηση γίνεται με την σειρά εισόδου, απορρίπτοντας τα πακέτα με διαφορετική πιθανότητα ανάλογα με το αν είναι IN ή OUT.

Η δομή του RIO στηρίζεται σε δυο παράλληλες υλοποιήσεις του αλγορίθμου RED, μια για κάθε είδος πακέτων. Συγκεκριμένα όταν κάποιο πακέτο φτάνει στον δρομολογητή ελέγχεται το bit προτεραιότητάς του για να διαπιστωθεί εάν πρόκειται για πακέτο IN ή OUT.

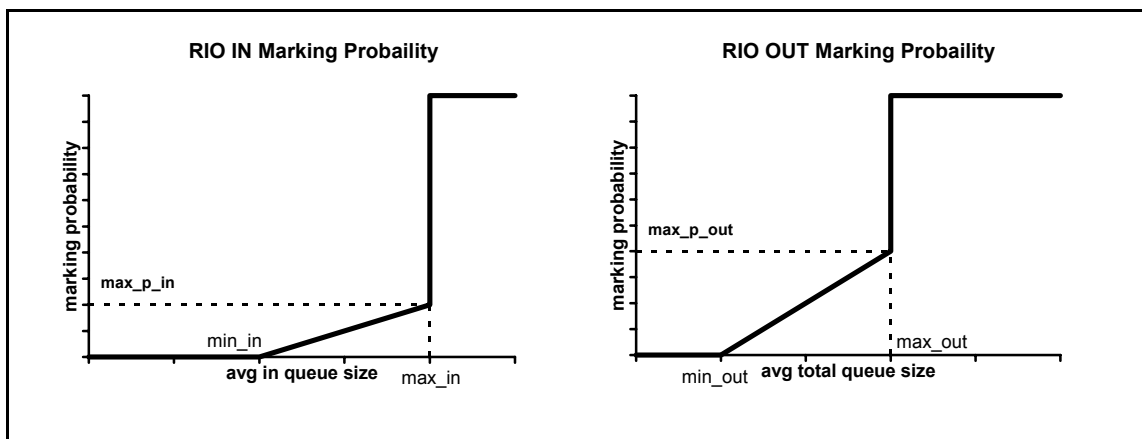
- Εάν το πακέτο είναι IN τότε υπολογίζεται το μέσο μήκος ουράς, σύμφωνα με τον αλγόριθμο EWMA που χρησιμοποιείται και από το RED, σαν να είχαμε στην ουρά μόνο τα IN πακέτα. Αυτό δεν σημαίνει πως τα IN πακέτα οδηγούνται σε ξεχωριστή ουρά, αλλά απλά διατηρείται μια μεταβλητή που παρακολουθεί το πλήθος των πακέτων αυτών που βρίσκονται στην ουρά. Στην συνέχεια υπολογίζεται η πιθανότητα απόρριψης πακέτου σύμφωνα με τον αλγόριθμο RED ως συνάρτηση του μέσου μήκους της «εικονικής» ουράς των IN πακέτων και ανάλογα απορρίπτεται ή όχι το συγκεκριμένο πακέτο.
- Εάν πρόκειται για OUT πακέτο τότε υπολογίζεται το πραγματικό μέσο μήκος ουράς και το πακέτο απορρίπτεται με πιθανότητα σύμφωνα με τον αλγόριθμο RED για το συνολικό πλέον μήκος ουράς.

Η επιλογή για την απόρριψη των IN πακέτων σύμφωνα με τον αριθμό μόνο αυτού του είδους πακέτων στην ουρά διασφαλίζει ότι ακόμη και αν η ουρά είναι

πλημμυρισμένη με OUT πακέτα τα IN πακέτα θα απορρίπτονται με μικρή ή και μηδενική πιθανότητα. Η χρήση του συνολικού αριθμού των πακέτων στην ουρά για την πιθανότητα απόρριψης των OUT πακέτων εξασφαλίζει ότι τα OUT πακέτα θα απορρίπτονται με μεγαλύτερη πιθανότητα από τα IN, καθώς και ότι η ουρά θα παραμένει σε κάθε περίπτωση εντός κάποιων συγκεκριμένων ορίων.

Για καθέναν από τους δύο ξεχωριστούς αλγορίθμους RED που χρησιμοποιούνται επιλέγεται και διαφορετικό σεντ παραμέτρων min_{thresh} , max_{thresh} , max_p . Έτσι για τα IN πακέτα έχουμε τις παραμέτρους min_{in} , max_{in} , $p_{max_{in}}$ και για τα OUT τις min_{out} , max_{out} , $p_{max_{out}}$. Το ποσοστό της διάκρισης εναντίον των OUT πακέτων καθορίζεται από την επιλογή των παραμέτρων αυτών. Συγκεκριμένα επιλέγοντας $min_{out} < min_{in}$ τα OUT πακέτα απορρίπτονται πολύ νωρίτερα από τα IN. Επίσης με την επιλογή $p_{max_{out}} > p_{max_{in}}$ τα OUT πακέτα έχουν μεγαλύτερη πιθανότητα απόρριψης. Τέλος όταν είναι $max_{out} < max_{in}$ διασφαλίζεται ότι αν η ουρά βρίσκεται σε υπερβολικά υψηλά επίπεδα θα απορρίπτονται όλα τα OUT πακέτα ώστε να υπάρχει διαθέσιμος χώρος για τα IN.

Πιθανότητες απόρριψης IN και OUT πακέτων στο RIO



Β. Πειραματικό Μέρος

1. Γενικά

Στην εργασία αυτή εξετάζονται τρεις τοπολογίες δικτύου με διάφορους συνδυασμούς κίνησης, ώστε να αναδειχθούν οι διαφορές και τα πλεονεκτήματα τόσο των πρωτοκόλλων ελέγχου συμφόρησης όσο και των αλγορίθμων ενεργής διαχείρισης ουρών.

Τα μέτρα απόδοσης που χρησιμοποιούνται κατά περίπτωση για την αξιολόγηση των εξεταζόμενων αλγορίθμων είναι :

- Η στιγμιαία και μέση ρυθμαπόδοση (throughput) που επιτυγχάνουν οι συνδέσεις και κατά πόσο η κατανομή του διαθέσιμου εύρους ζώνης (fairness) ανάμεσα στις διάφορες ροές είναι δίκαιη.
- Η αποτελεσματικότητα των αλγορίθμων ενεργής διαχείρισης ουρών απέναντι σε μη υπεύθυνες σε σήματα συμφόρησης ροές (UDP).
- Το χαμηλό και όσο το δυνατόν σταθερό μήκος ουράς, που εκφράζει την καθυστέρηση στην ουρά και την διασπορά αυτής, η οποία καθορίζει την τελική καθυστέρηση που συναντούν τα πακέτα δεδομένων που εδώ εκφράζεται μέσω του Round Trip Time (RTT).
- Η αποφυγή του συγχρονισμού ανάμεσα στις TCP ροές, ο οποίος προκαλεί περιοδικότητα στο μήκος ουράς.
- Η αποφυγή του αποκλεισμού συγκεκριμένων συνδέσεων.

Όλα τα πειράματα διεξάγονται με τη χρήση του εξομοιωτή δικτύων *ns-2 network simulator* του *Lawrence Berkeley National Laboratory*. Το πρόγραμμα αυτό προσφέρει έναν εύκολο και αποδοτικό τρόπο για τη δημιουργία και την προσομοίωση πολλών τοπολογιών δικτύων. Στις βιβλιοθήκες του περιέχεται ένας μεγάλος αριθμός αντικειμένων που προσομοιώνουν τη λειτουργία βασικών στοιχείων ενός δικτύου. Μεσώ μιας scripting language, της TCL καλούνται στιγμιότυπα αυτών των αντικειμένων, με παραμέτρους που τίθενται από το χρήστη, ώστε να συντεθεί το επιθυμητό δίκτυο. Οι μεταβλητές που περιέχονται σε αυτά, δίνουν τη κατάσταση του δικτύου σε κάθε χρονική στιγμή. Η παρακολούθηση των τιμών αυτών των μεταβλητών και η κατάλληλη επεξεργασία τους δίνει όλες τις πληροφορίες που

χρειάζονται για την αξιολόγηση του εξεταζόμενου δικτύου όπως throughput μήκη ουρών κλπ.

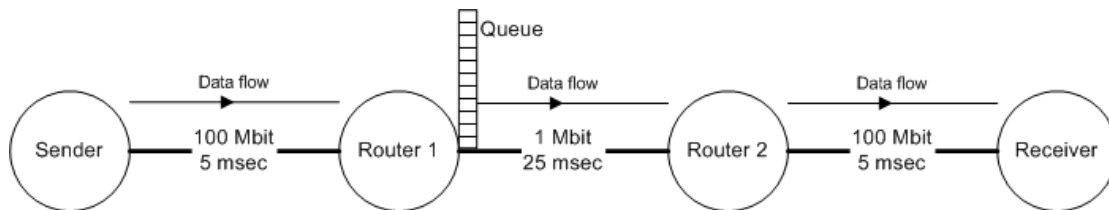
Στην παρούσα εργασία χρησιμοποιείται μια τροποποιημένη έκδοση του κώδικα του ns-2. Οι τροποποιήσεις που έγιναν δεν αφορούν τους αλγόριθμους από τους οποίους προσομοιώνεται η λειτουργία των στοιχείων δικτύου. Προσφέρουν μόνο την παρακολούθηση κάποιων μεταβλητών που η γνώση τους απαιτείται για την εξαγωγή των επιθυμητών συμπερασμάτων. Συγκεκριμένα έγιναν οι εξής τροποποιήσεις :

- Κώδικας ουράς RED :
 - Τροποποίηση της συνάρτησης παρακολούθησης των μεταβλητών (trace) για αυτοματοποιημένη επεξεργασία των αποτελεσμάτων. Ομοίως και για τα υπόλοιπα είδη ουρών.
- Κώδικας ουράς REM :
 - Προσθήκη συνάρτησης υπολογισμού του μέσου μήκους ουράς μέσω αλγορίθμου EWMA.
 - Παρακολούθηση (trace) του κόστους ζεύξης (link price)
- Κώδικας ουράς RIO :
 - Παρακολούθηση μήκους IN ουράς και πιθανοτήτων απόρριψης IN και OUT πακέτων
- Κώδικας πρωτοκόλλου TFRC :
 - Παρακολούθηση ρυθμού μετάδοσης και RTT.

Επίσης η ουρά Drop Tail προσομοιώθηκε από μια RED ουρά με τα δυο κατώφλια ίσα με το όριο της ουράς, ώστε να είναι δυνατή η αποδοτικότερη παρακολούθηση της .

2. Μια υπό συμφόρηση ζεύξη

Στην πρώτη τοπολογία έχουμε έναν κόμβο ο οποίος στέλνει πακέτα (αποστολέας), έναν ο οποίος τα λαμβάνει (παραλήπτης) και δυο ενδιάμεσους κόμβους – δρομολογητές. Μεταξύ των δυο δρομολογητών υπάρχει αμφίδρομη σύνδεση με ταχύτητα 1Mbit/sec και καθυστέρηση διάδοσης 25msec ενώ μεταξύ του αποστολέα και του πρώτου δρομολογητή καθώς και μεταξύ του δεύτερου δρομολογητή και του παραλήπτη υπάρχουν αμφίδρομες συνδέσεις ταχύτητας 100Mbit/sec και καθυστέρησης 5msec. Το δίκτυο αυτό φαίνεται στο παρακάτω σχήμα .



Ως κίνηση χρησιμοποιούμε συνδέσεις TCP ή-και TFRC από τον αποστολέα στον παραλήπτη. Οι συνδέσεις αυτές ξεκινούν ταυτόχρονα και στέλνουν πακέτα των 1000 bytes σύμφωνα με τους αλγορίθμους ελέγχου ροής του κάθε πρωτοκόλλου για διάρκεια 150 sec. χρησιμοποιούμε την παραλλαγή New Reno για τις συνδέσεις TCP, αφού όπως είδαμε στη θεωρία είναι η πλέον αποδοτική για τον αλγόριθμο ελέγχου συμφόρησης του TCP, με καθυστερούμενες επιβεβαιώσεις (delayed acknowledgments) και θέτουμε το παράθυρο του παραλήπτη αρκετά μεγάλο ώστε να προκαλείται συμφόρηση στην αργή ζεύξη μεταξύ των δυο δρομολογητών ακόμη και με μια και μοναδική σύνδεση. Στις συνδέσεις TFRC θέτουμε $\beta=2$ στην εξίσωση ρυθμαπόδοσης του TCP (TCP throughput equation) αφού όπως αναφέραμε οι συνδέσεις TCP χρησιμοποιούν delayed acknowledgments. Εκτός από αυτές τις συνδέσεις σε ορισμένα σενάρια χρησιμοποιούμε κίνηση UDP ως κίνηση υποβάθρου (background traffic). Τα UDP πακέτα ξεκινούν να παράγονται 30 sec έπειτα από την έναρξη των υπολοίπων συνδέσεων με ρυθμό ίσο με το 50% της χωρητικότητας της σύνδεσης δηλαδή 500 kbit/sec , έπειτα από άλλα 30 sec ο ρυθμός αυτός γίνεται ίσος

με 750 kbit/sec ή 75% της χωρητικότητας , μετά από άλλα 30 sec ο ρυθμός ξαναπέφτει στο 50% για να τερματιστεί τελειώς η σύνδεση μετά από ακόμη 30sec.

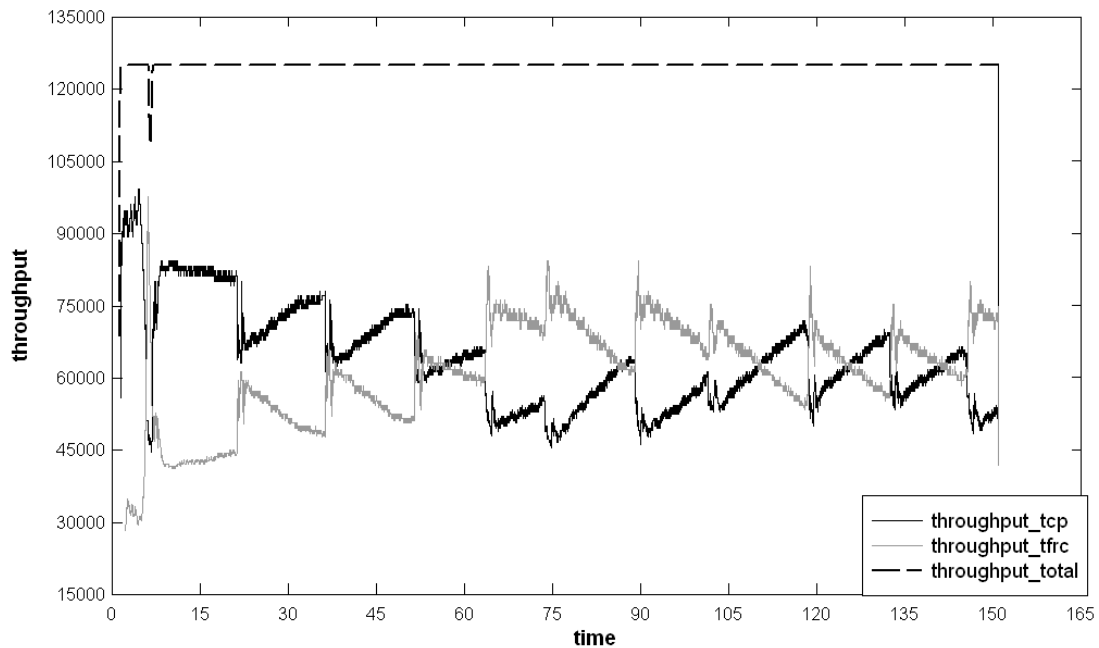
Στην ουρά της σύνδεσης μεταξύ των δυο δρομολογητών ο χώρος ενταμίευσης των πακέτων έχει μέγεθος 50 πακέτα. Στην περίπτωση που ο αλγόριθμος ενεργής διαχείρισης της ουράς (AQM) είναι ο RED έχουμε την εξής διαμόρφωση των παραμέτρων : ελάχιστο κατώφλι min threshold 15 πακέτα, μέγιστο κατώφλι max threshold 30 πακέτα, μέγιστη πιθανότητα απόρριψης πακέτου max dropping probability 0.1 και βάρος στον αλγόριθμο Exponential Weighted Moving Average για τον υπολογισμό του μέσου μήκους ουράς queue weight 0.005. Όταν χρησιμοποιούμε τον αλγόριθμο REM για AQM έχουμε τις εξής τιμές παραμέτρων : χρόνος ενημέρωσης του κόστους σύνδεσης update time 0,002 sec , τιμή του γ για τον υπολογισμό του κόστους gamma 0.001 , τιμή του ϕ για τον υπολογισμό της πιθανότητας απόρριψης πακέτων phi 1.001 , επιθυμητό μήκος ουράς b_0 20 πακέτα. Στην περίπτωση που ο αλγόριθμος AQM είναι ο RIO εξετάσαμε δυο σετ παραμέτρων. Στο πρώτο έχουμε : ελάχιστο κατώφλι για τα σύμφωνα πακέτα min in threshold 10 πακέτα , μέγιστο κατώφλι για τα σύμφωνα πακέτα max in threshold 30 πακέτα , μέγιστη πιθανότητα απόρριψης σύμφωνου πακέτου max in dropping probability 0.05 , ελάχιστο κατώφλι για τα μη σύμφωνα πακέτα min out threshold 15 πακέτα , μέγιστο κατώφλι για τα μη σύμφωνα πακέτα max out threshold 25 πακέτα , μέγιστη πιθανότητα απόρριψης μη σύμφωνου πακέτου max out dropping probability 0.5 , βάρος αλγορίθμου EWMA για τον υπολογισμό του μέσου μήκους ουράς queue weight 0.005. Στο δεύτερο σετ έχουμε min out threshold 5 πακέτα και max out threshold 15 πακέτα έτσι ώστε να έχουμε πιο «σκληρή» αντιμετώπιση των μη σύμφωνων πακέτων. Επίσης στην περίπτωση του RIO χρησιμοποιήσαμε την εξής πολιτική κατηγοριοποίησης των πακέτων σε σύμφωνα και μη : αναθέτουμε σε κάθε σύνδεση εύρος ίσο με την χωρητικότητα της ζεύξης προς τον συνολικό αριθμό των συνδέσεων και τα πακέτα πάνω από αυτό τον ρυθμό είναι μη σύμφωνα, ενώ όλα τα πακέτα UDP είναι μη σύμφωνα.

Εκτελέσαμε προσομοιώσεις χρησιμοποιώντας το ns-2 network simulator με διάφορους συνδυασμούς ροών. Μόνο συνδέσεις TCP (μια ή τρεις μαζί), μόνο συνδέσεις TFRC (μια ή τρεις μαζί), ή και τους δυο τύπους μαζί (μια από κάθε είδος, τρεις από κάθε είδος, μια με τρεις και τρεις με μια), με ή χωρίς background UDP traffic. από τα αποτελέσματα των προσομοιώσεων παρατηρούμε τα εξής :

2.1 Δικαιοσύνη μεταξύ των ροών (fairness)

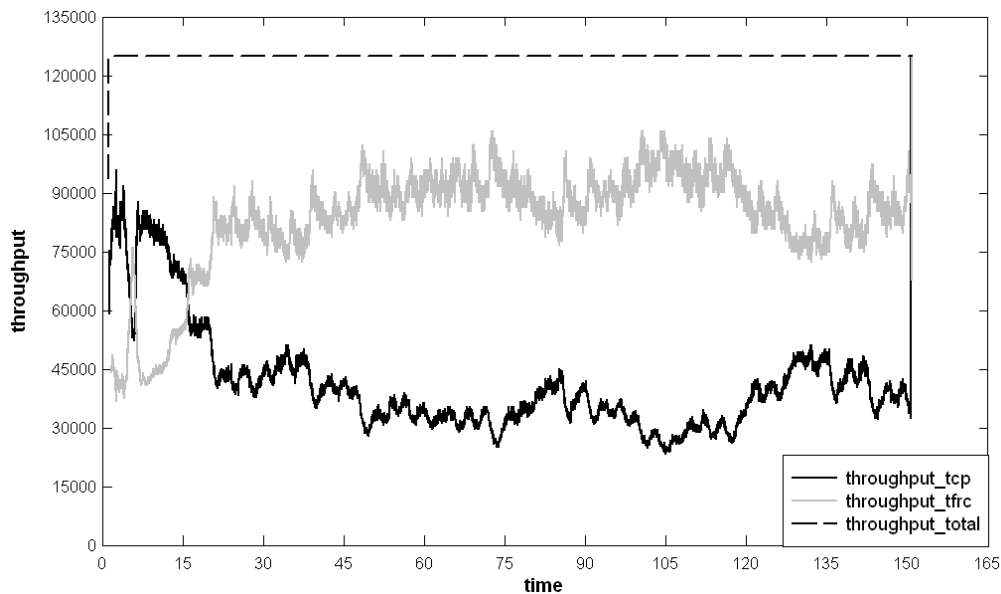
- Όταν έχουμε πολλαπλές ροές είτε μόνο TCP είτε μόνο TFRC παρατηρούμε ότι στιγμιαία κάποιες συνδέσεις μπορεί να καταλαμβάνουν περισσότερο εύρος ζώνης από τις υπόλοιπες, όμως οι μέσες τιμές για όλες τις συνδέσεις είναι σχεδόν ίδιες. Σε αυτή την περίπτωση χρησιμοποιώντας κάποιον αλγόριθμο ενεργής διαχείρισης ουρών μπορούμε να επιτύχουμε μια σχετική μείωση των βραχυπρόθεσμων αυτών διαφορών.
- Στην περίπτωση όπου συνυπάρχουν τα δυο είδη ροών και δεν χρησιμοποιούμε κάποιον αλγόριθμο ενεργής διαχείρισης ουρών αλλά η ουρά στη σύνδεση μεταξύ των δυο δρομολογητών είναι Drop Tail, οι συνδέσεις TFRC διεκδικούν πιο αργά από τις TCP συνδέσεις το διαθέσιμο εύρος ζώνης, με το πέρασμα του χρόνου όμως καταλαμβάνουν μεγαλύτερο ποσοστό της χωρητικότητας της ζεύξης από αυτό που παίρνουν οι TCP συνδέσεις.

Throughput ως συνάρτηση του χρόνου ανά τύπο σύνδεσης
με μια σύνδεση από κάθε είδος



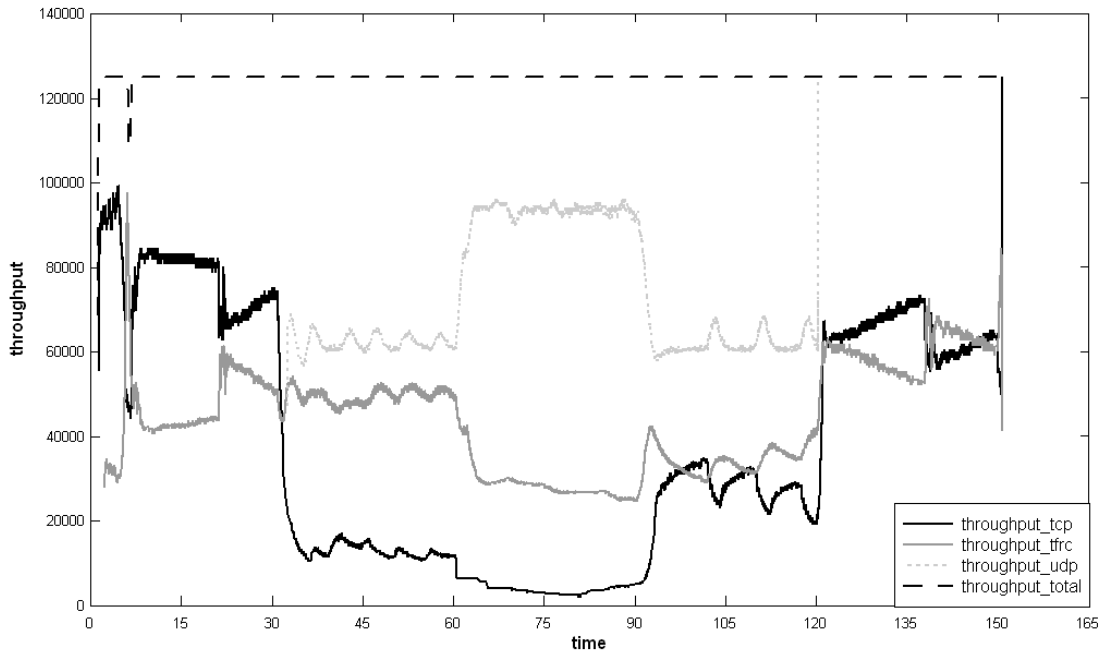
- Η ανισότητα στην κατανομή των πόρων του δικτύου αυξάνεται όταν αυξάνονται οι συνδέσεις από κάθε είδος ροής, αφού μεγαλύτερος αριθμός συνδέσεων σημαίνει και μεγαλύτερη επιθετικότητα για το TFRC.

Throughput ως συνάρτηση του χρόνου ανά τύπο σύνδεσης
με τρεις συνδέσεις από κάθε είδος



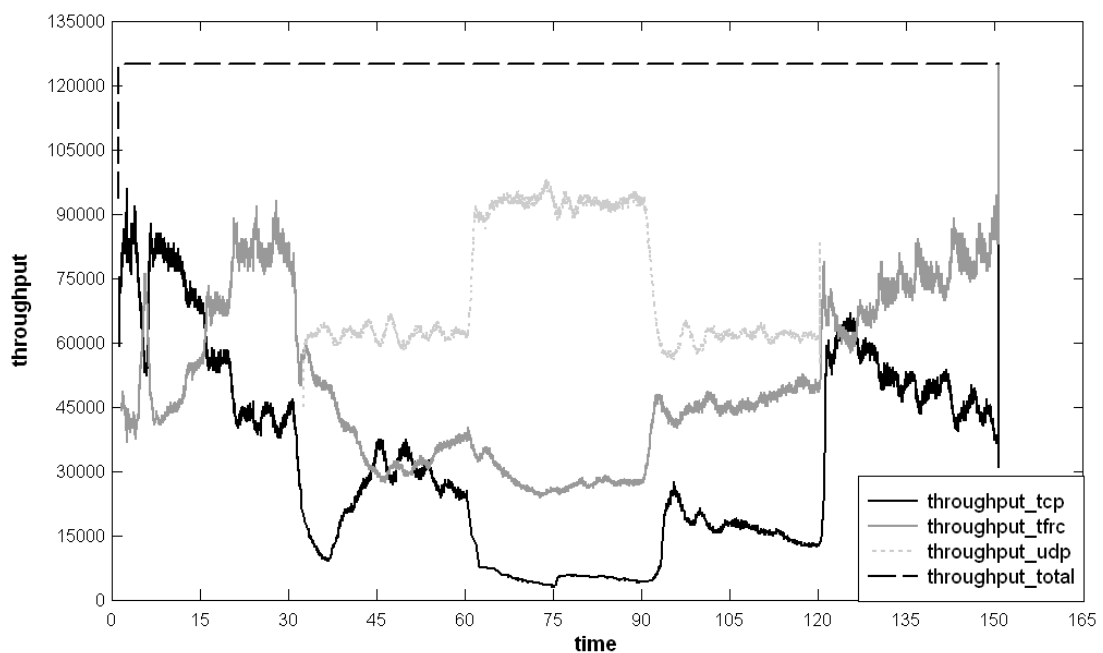
- Όταν η διαθέσιμη χωρητικότητα της ζεύξης μειώνεται λόγω της παρουσίας background traffic υπό τη μορφή μιας UDP ροής, ή οποία ως γνωστόν δεν αντιδρά σε σήματα συμφόρησης, οι TFRC ροές μειώνουν τον ρυθμό μετάδοσης τους πιο αργά από τις ροές TCP με αποτέλεσμα οι τελευταίες να παίρνουν δυσανάλογα μικρό μερίδιο του διαθέσιμου εύρους ζώνης. Όταν όμως αυξάνεται η διαθέσιμη χωρητικότητα λόγω μείωσης του ρυθμού μετάδοσης της UDP ροής το TCP διεκδικεί γρηγορότερα τους περισσιους πόρους του δικτύου.

Throughput ως συνάρτηση του χρόνου ανά τύπο σύνδεσης
 παρουσία background UDP traffic με μια σύνδεση από κάθε είδος



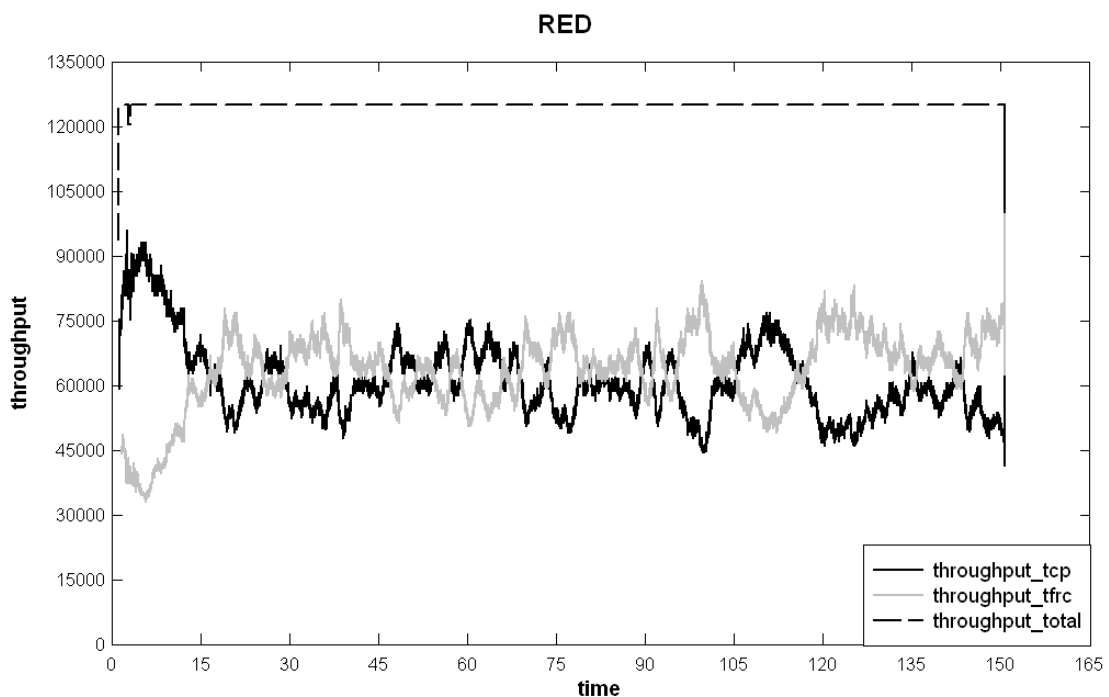
- Στην περίπτωση πολλαπλών συνδέσεων από κάθε είδος (οπότε και μεγαλύτερη επιθετικότητα του TFRC) οι ροές TFRC διεκδικούν γρηγορότερα το διαθέσιμο εύρος ζώνης.

Throughput ως συνάρτηση του χρόνου ανά τύπο σύνδεσης
 παρουσία background UDP traffic με τρεις συνδέσεις από κάθε είδος

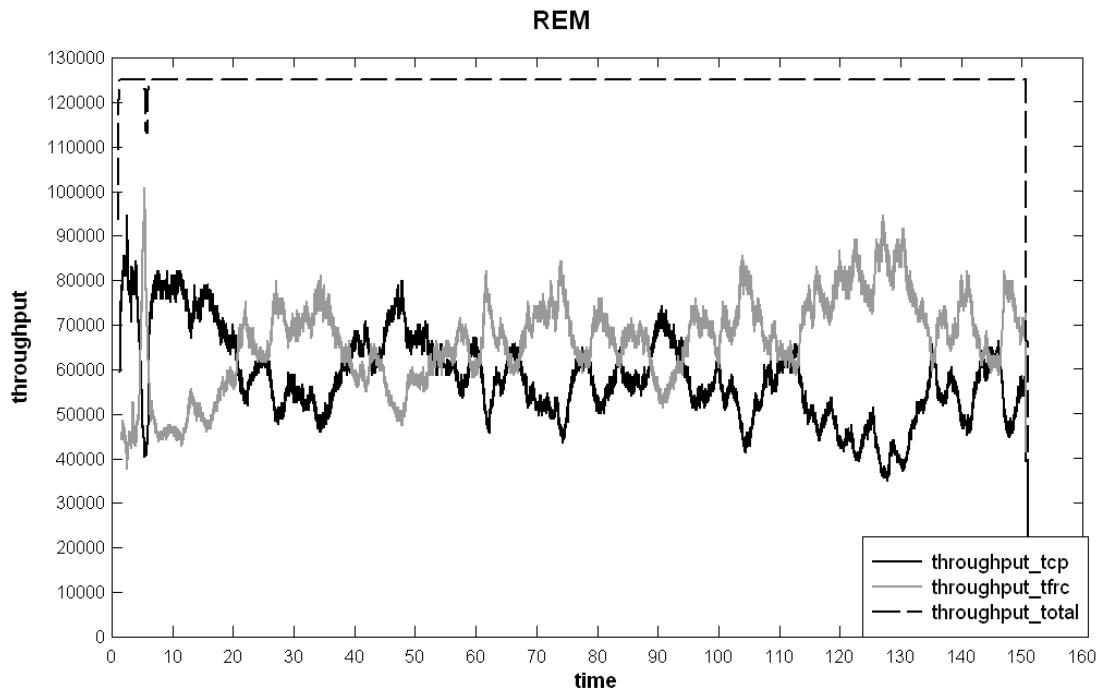


- Όταν χρησιμοποιούμε κάποιον αλγόριθμο ενεργής διαχείρισης ουρών και έχουμε και τα δυο είδη ροών μαζί οι ανισότητες που παρατηρούνταν στην περίπτωση Drop Tail ουράς αμβλύνονται. Επομένως με τους αλγορίθμους AQM επιτυγχάνουμε μεγαλύτερο fairness μεταξύ ανόμοιων ροών. Αυτό οφείλεται στον μηχανισμό απόρριψης πακέτων που χρησιμοποιείται από αυτούς σύμφωνα με τον οποίο έχουμε απόρριψη με βάση κάποια ορισμένη πιθανότητα, οπότε πιο επιθετικές ροές χάνουν περισσότερα πακέτα και άρα δέχονται ισχυρότερα σήματα συμφόρησης με αποτέλεσμα να ρίχνουν τον ρυθμό μετάδοσης τους.

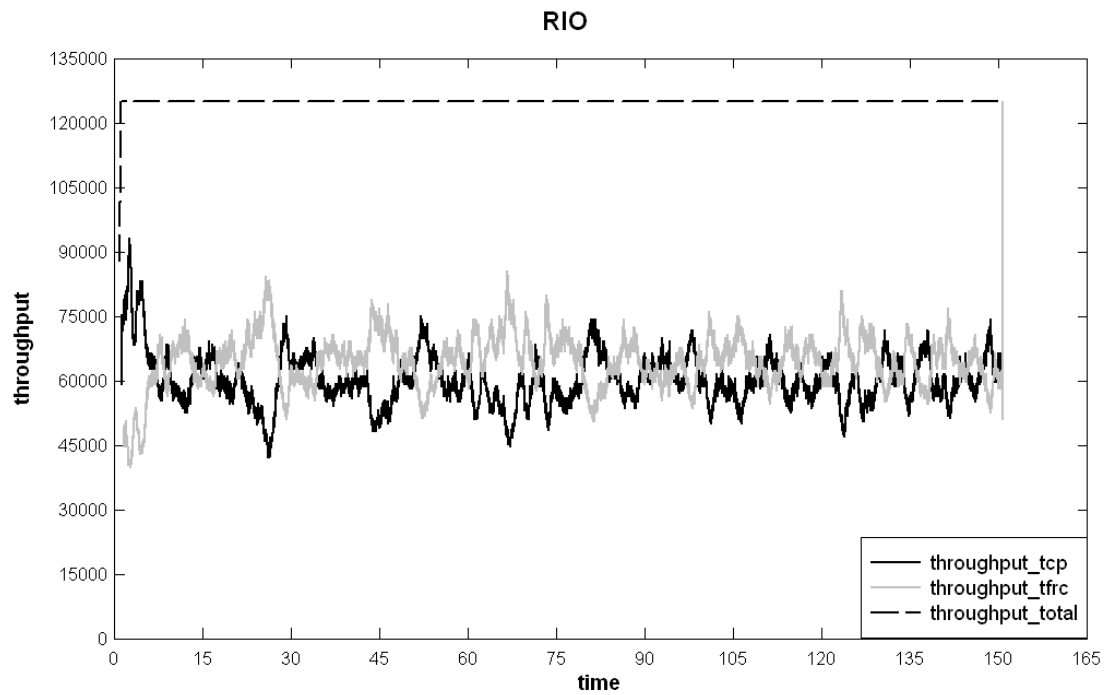
Throughput ως συνάρτηση του χρόνου ανά τύπο σύνδεσης
με τρεις συνδέσεις από κάθε είδος και RED ουρά



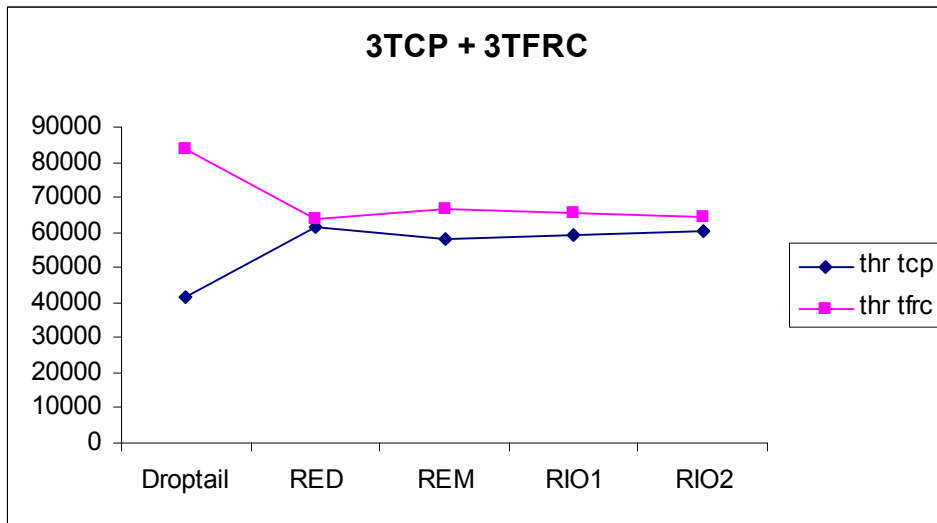
Throughput ως συνάρτηση του χρόνου ανά τύπο σύνδεσης
με τρεις συνδέσεις από κάθε είδος και REM ουρά



Throughput ως συνάρτηση του χρόνου ανά τύπο σύνδεσης
με τρεις συνδέσεις από κάθε είδος και RIO ουρά



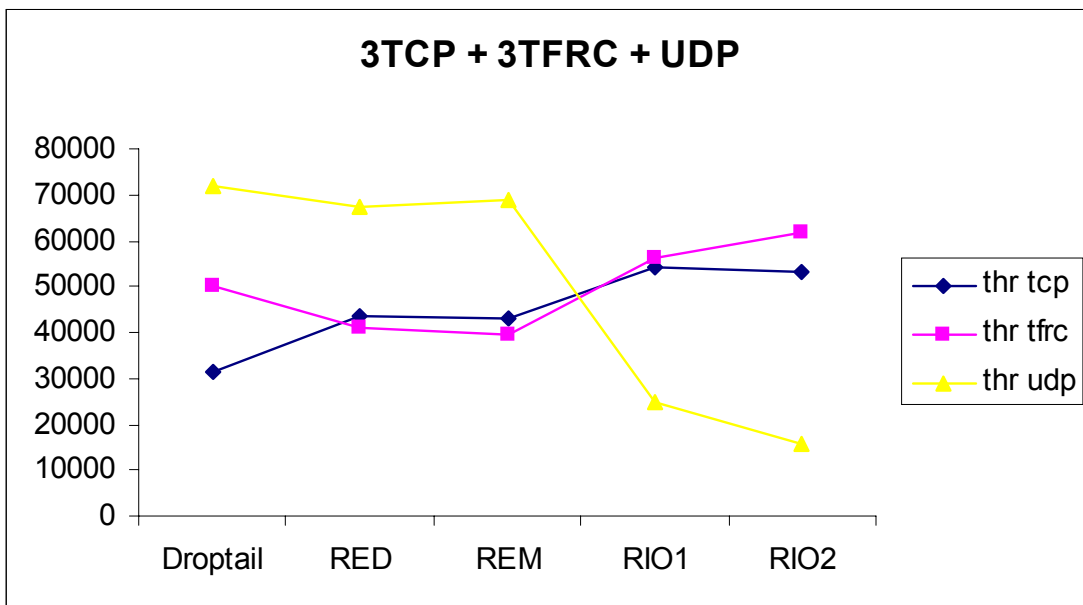
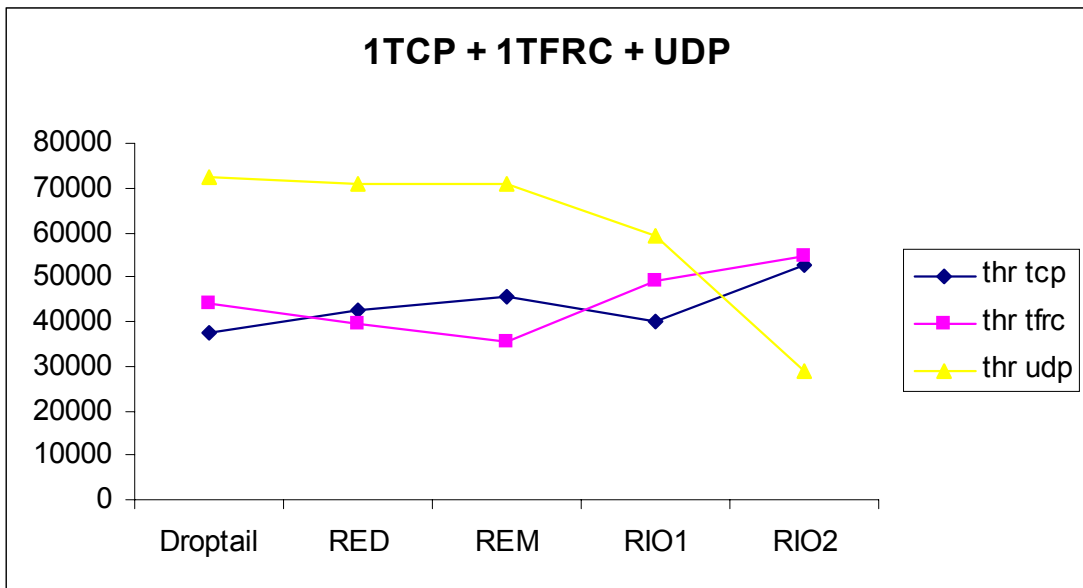
Μέσο Throughput ως συνάρτηση του αλγορίθμου AQM ανά τύπο σύνδεσης
με τρεις συνδέσεις από κάθε είδος

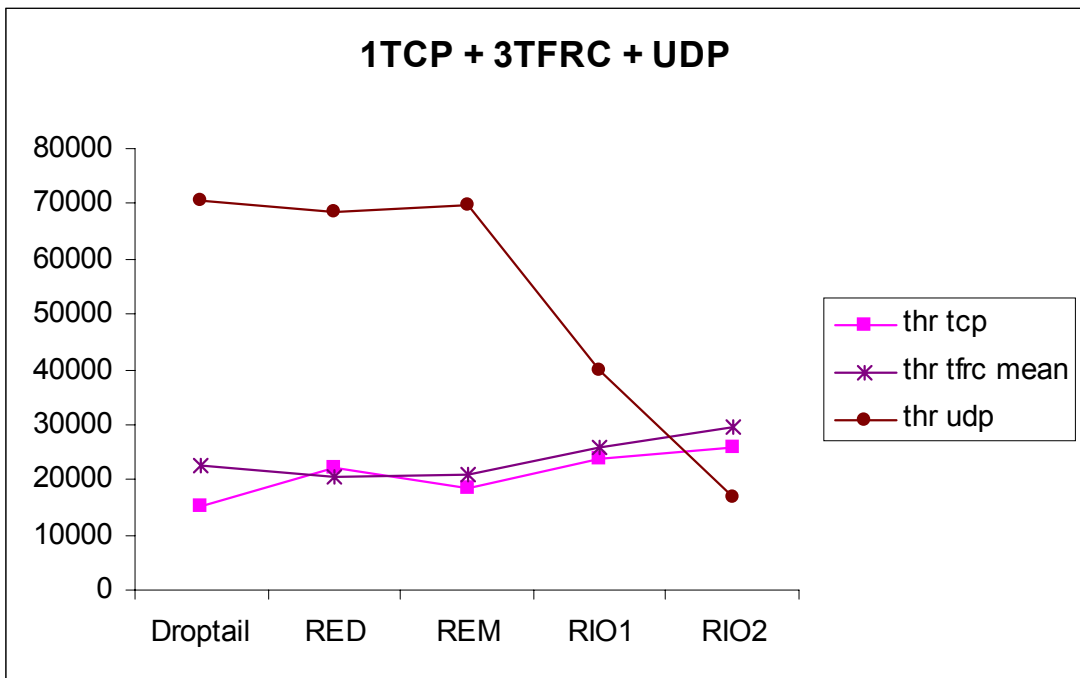
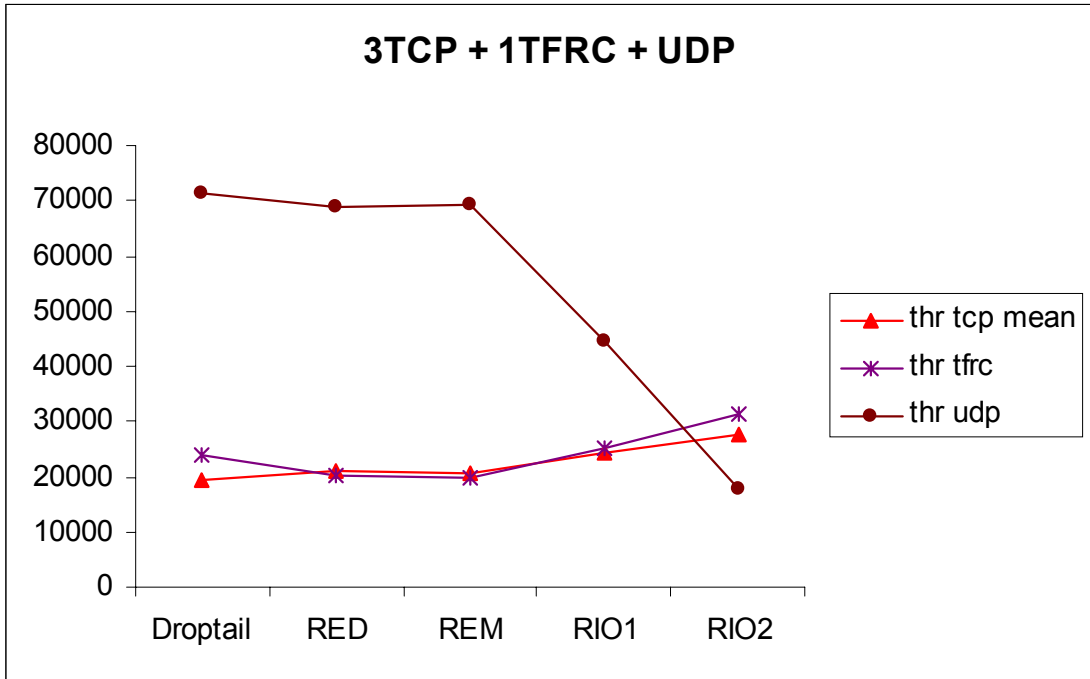


2.2 Αποτελεσματικότητα απέναντι σε μη υπεύθυνες ροές

- Οι αλγόριθμοι AQM RED και REM δεν κατορθώνουν να μειώσουν το εύρος ζώνης που καταλαμβάνει μια μη υπεύθυνη σε σήματα συμφόρησης ροή, παρά μόνο μοιράζουν δίκαια το εύρος ζώνης που περισσεύει στις υπόλοιπες ροές.
- Ο αλγόριθμος RIO μπορεί να καταστείλει τις μη υπεύθυνες ροές και να δώσει μεγαλύτερο εύρος ζώνης στις υπόλοιπες απ' όσο θα έπαιρναν με Drop Tail ουρά ή με ουρά που υλοποιεί αλγορίθμους RED ή REM. αυτό οφείλεται στο ότι γίνεται ανάθεση του ρυθμού μετάδοσης δεδομένων ανά ροή και τα μη σύμφωνα με τον προδιαγραφόμενο ρυθμό πακέτα απορρίπτονται πρώτα. έτσι, όταν στα πειράματα μας αναθέτουμε μηδενικό ρυθμό στην UDP ροή, το αποτέλεσμα είναι να χάνονται τα περισσότερα από τα πακέτα UDP, ενώ η ουρά περιέχει σχεδόν αποκλειστικά πακέτα από τις υπόλοιπες ροές. Εάν αναθέσουμε κάποιον μεγαλύτερο ρυθμό στο UDP τότε μειώνεται η αποτελεσματικότητα του αλγορίθμου αυτού. Επίσης όταν οι παράμετροι του αλγορίθμου είναι πιο «σκληροί» απέναντι στα μη σύμφωνα με τον προδιαγραφόμενο ρυθμό πακέτα (out packets) μειώνεται ακόμη περισσότερο το ποσοστό του εύρους ζώνης που παίρνει η UDP ροή και κατά συνέπεια αυξάνεται το ποσοστό για τις υπόλοιπες ροές. Τέλος όσες περισσότερες υπεύθυνες ροές έχουμε τόσο περισσότερο μειώνεται το throughput της μη υπεύθυνης, αφού αυτή έχει να ανταγωνιστεί περισσότερες συνδέσεις.

Μέσο Throughput ως συνάρτηση του αλγορίθμου AQM ανά τύπο σύνδεσης
(TCP, TFRC, UDP) για διάφορους συνδυασμούς ροών

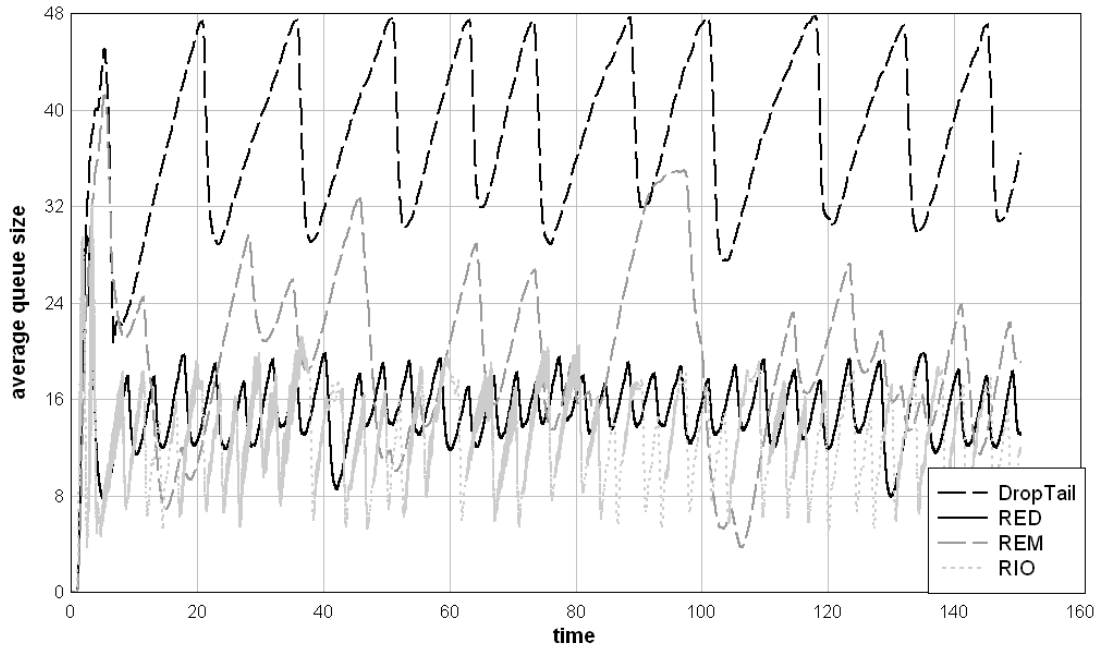




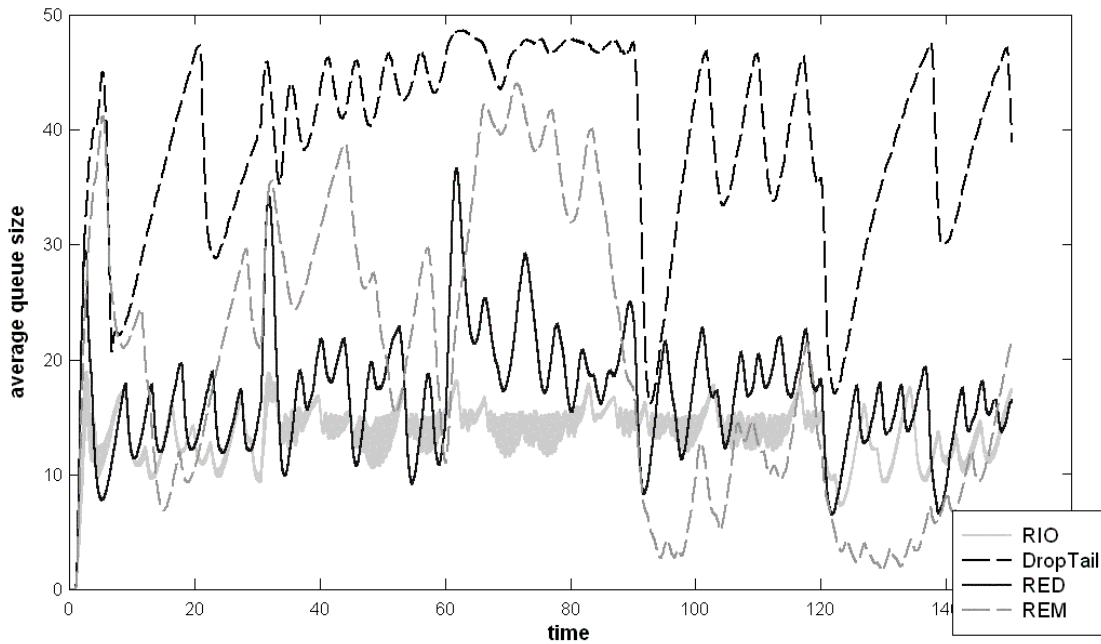
2.3 Μέσο μήκος ουράς και καθυστέρηση

- Όταν δεν χρησιμοποιείται κάποιος αλγόριθμος AQM, αλλά η ουρά είναι Drop Tail το μέσο μήκος ουράς σε περιόδους συμφόρησης παραμένει κοντά στο όριο της ουράς, αφού πακέτα χάνονται μόνο όταν η ουρά είναι γεμάτη. αυτό έχει ως αποτέλεσμα η καθυστέρηση των διαφόρων πακέτων να είναι υψηλή.
- Με την χρησιμοποίηση των αλγορίθμων AQM η ουρά παραμένει σε χαμηλότερα αλλά όχι υπερβολικά χαμηλά επίπεδα. έτσι εξασφαλίζεται τόσο η υψηλή χρησιμοποίηση (utilization) της ζεύξης, αφού υπάρχουν πάντοτε πακέτα προς αποστολή στην ουρά, όσο και μικρότερη συνολική καθυστέρηση αφού η καθυστέρηση στην ουρά (queuing delay) είναι ανάλογη του μήκους της.
- Οι αλγόριθμοι RED και RIO εξασφαλίζουν επιπλέον μικρότερη διακύμανση του μήκους της ουράς, γεγονός που συμβάλει στην μείωση της διασποράς της καθυστέρησης από πακέτο σε πακέτο. αυτό το χαρακτηριστικό είναι ιδιαίτερα σημαντικό για ροές με ευαισθησία στη διασπορά της καθυστέρησης, όπως είναι για παράδειγμα το Voice over IP.
- Όταν έχουμε πλέον των υπολοίπων και ροή UDP ο αλγόριθμος REM παρουσιάζει αστάθεια με μεγάλες αυξομειώσεις του μήκους ουράς η οποία γεμίζει ή αδειάζει ανάλογα με το εάν υπάρχει κίνηση UDP. υπό τις ίδιες συνθήκες ο αλγόριθμος RED παρουσιάζει μια μικρή αυξητική τάση όσον αφορά το μήκος της ουράς σε σχέση με την περίπτωση που δεν έχουμε UDP κίνηση, ενώ με τον αλγόριθμο RIO το μήκος της ουράς κυμαίνεται στα ίδια σχεδόν επίπεδα με πριν, αφού το μεγαλύτερο ποσοστό των πακέτων UDP απορρίπτονται την ουρά.

μέσο μήκος ουράς σε συνάρτηση με τον χρόνο για τους διάφορους αλγορίθμους AQM
με μια TCP και μια TFRC σύνδεση



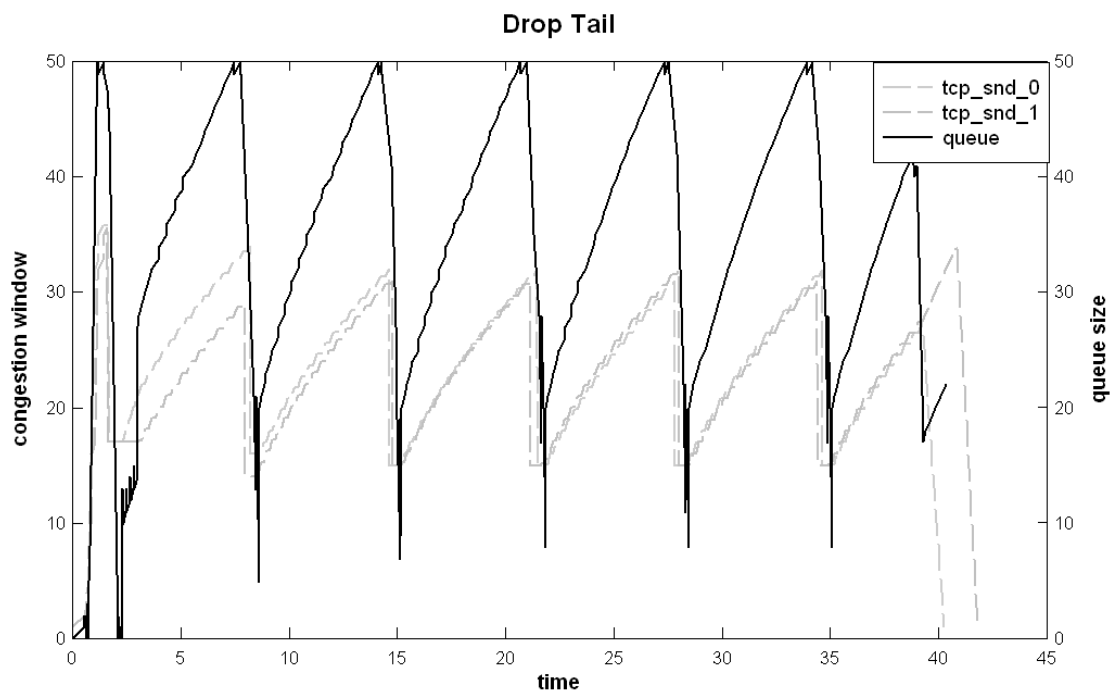
μέσο μήκος ουράς σε συνάρτηση με τον χρόνο για τους διάφορους αλγορίθμους AQM
με μια TCP και μια TFRC σύνδεση παρουσία UDP ροής



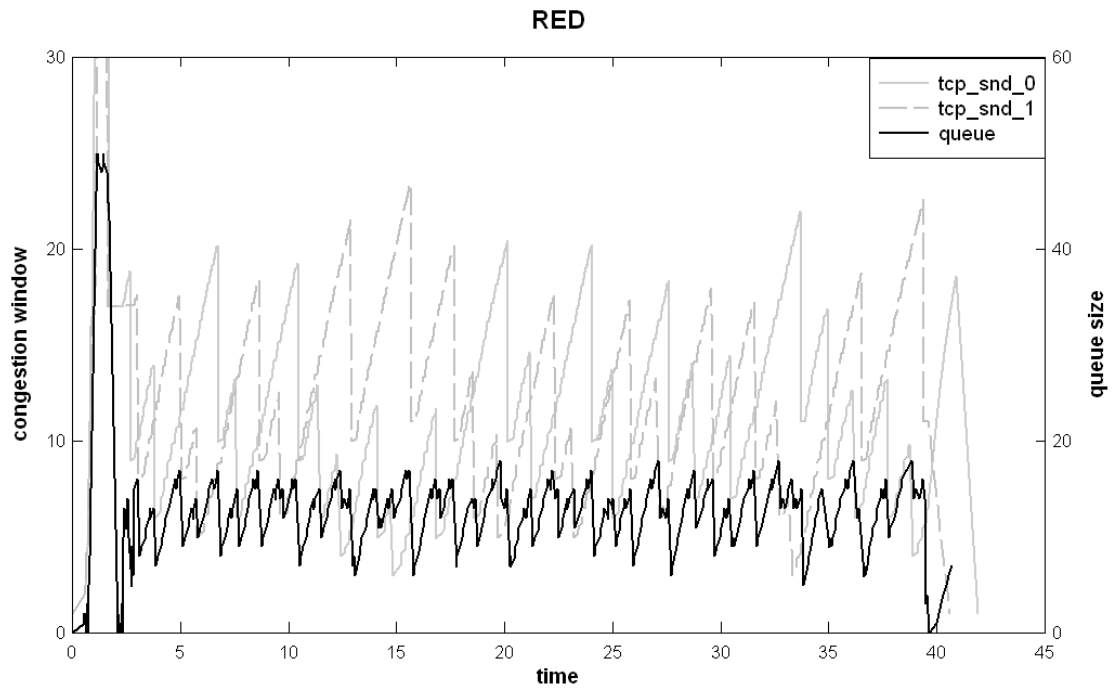
2.4 Συγχρονισμός

- Όταν η ουρά στη ζεύξη μεταξύ των δυο δρομολογητών είναι Drop Tail και έχουμε συνδέσεις TCP παρατηρείται συγχρονισμός μεταξύ των διάφορων ροών όσον αφορά την αυξομείωση του παραθύρου συμφόρησης το οποίο άλλοτε είναι πολύ μεγάλο σε όλες τις ροές, οπότε έχουμε υψηλή συμφόρηση, και άλλοτε πολύ μικρό σε όλες τις ροές, οπότε έχουμε υποχρησιμοποίηση της ζεύξης.
- Οι αλγόριθμοι AQM αποσυγχρονίζουν τα μεγέθη των παραθύρων έτσι ώστε όταν το ένα είναι πολύ μεγάλο τα υπόλοιπα να είναι πιο μικρά και το αντίθετο, οπότε σε όλες τις περιπτώσεις έχουμε μικρότερη συμφόρηση.
- Ο αλγόριθμος REM δεν έχει τόσο καλή απόδοση όσο ο RED και ο REM.

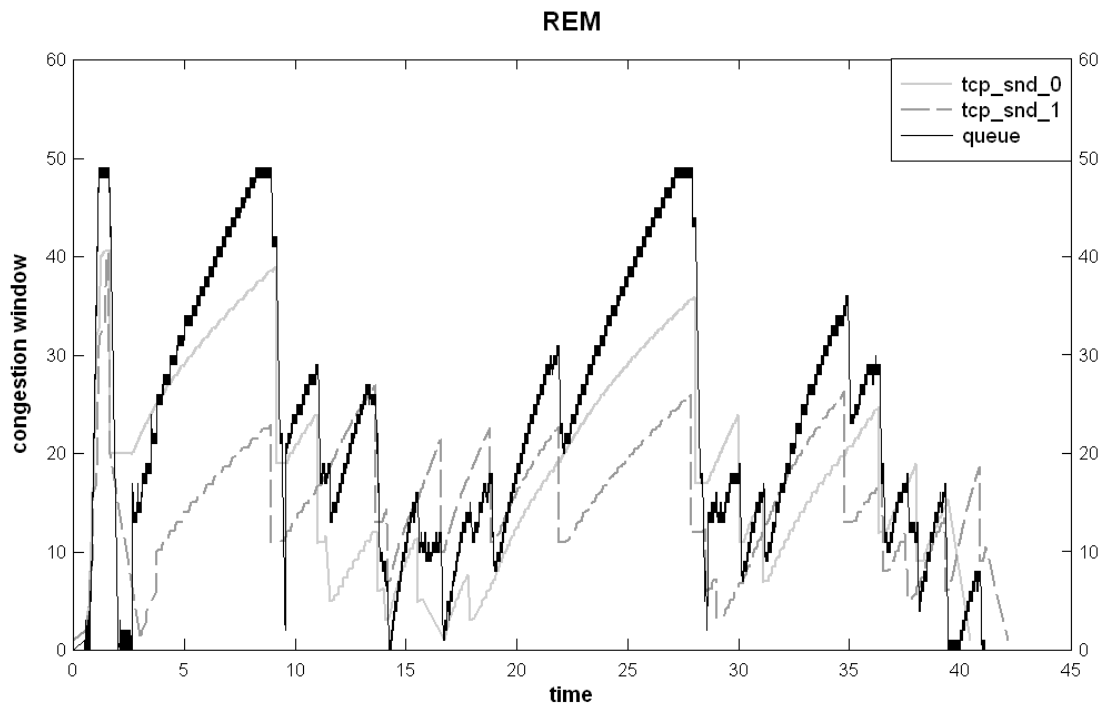
μέγεθος παραθύρου συμφόρησης και μήκος ουράς συναρτήσει του χρόνου
για 2 ταυτόχρονες TCP συνδέσεις με Drop Tail ουρά



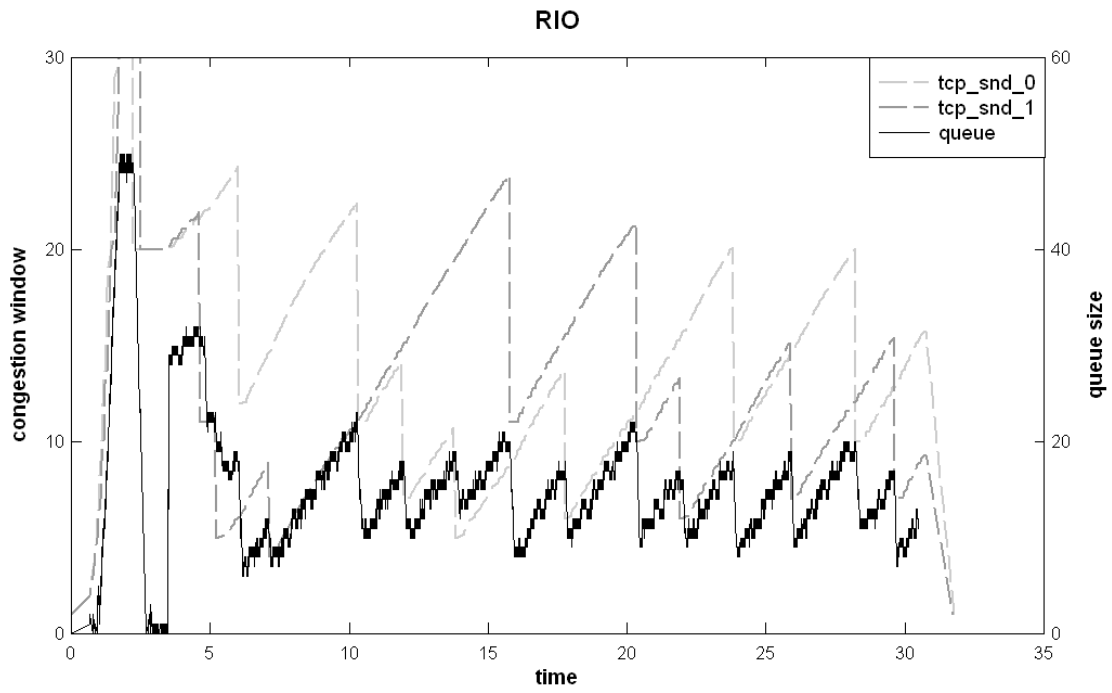
μέγεθος παραθύρου συμφόρησης και μήκος ουράς συναρτήσει του χρόνου
για 2 ταυτόχρονες TCP συνδέσεις με RED ουρά



μέγεθος παραθύρου συμφόρησης συναρτήσει του χρόνου
για 2 ταυτόχρονες TCP συνδέσεις με REM ουρά



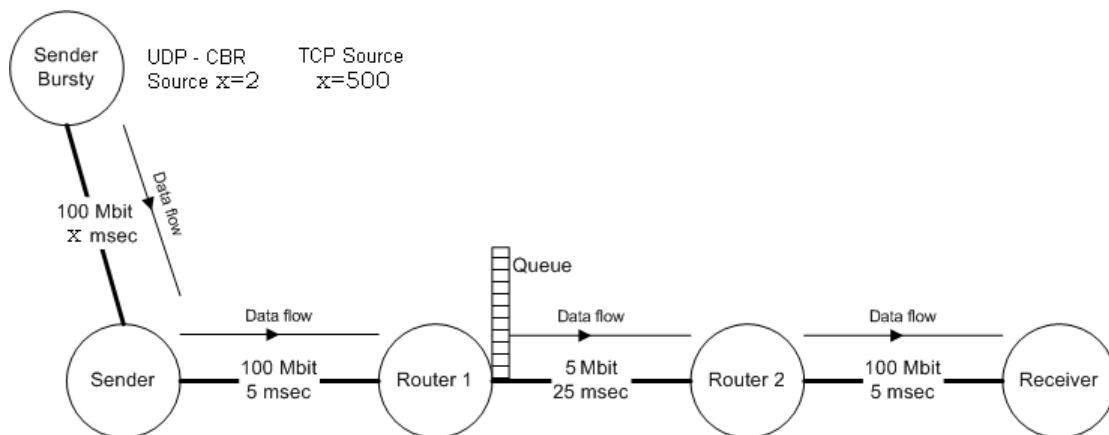
μέγεθος παραθύρου συμφόρησης συναρτήσει του χρόνου
για 2 ταυτόχρονες TCP συνδέσεις με RIO ουρά



3. Εκρηκτική κίνηση

Η δεύτερη τοπολογία είναι μια παραλλαγή της προηγούμενης που σκοπό έχει να μελετήσει την συμπεριφορά των αλγορίθμων ενεργής διαχείρισης ουρών απέναντι σε κίνηση εκρηκτικής μορφής, δηλαδή σύντομες ριπές πακέτων που επαναλαμβάνονται σε αραιά χρονικά διαστήματα. Έχουμε επομένως τους ίδιους κόμβους και ζεύξεις με την προηγούμενη τοπολογία, με την διάφορα ότι η χωρητικότητα της σύνδεσης μεταξύ των δυο δρομολογητών έχει γίνει 5 Mbit/sec, και επιπλέον έναν κόμβο στον οποίο βρίσκεται η πηγή εκρηκτικής κίνησης, ο οποίος συνδέεται με τον κόμβο – αποστολέα με μια ζεύξη χωρητικότητας 100 Mbit/sec.

Διακρίνουμε δυο σενάρια. Στο πρώτο η εκρηκτική κίνηση είναι μια TCP σύνδεση με μικρό παράθυρο συμφόρησης (10 πακέτα) και μεγάλο Round Trip Time, η οποία παράγει σχετικά εκρηκτική κίνηση. Σε αυτή την περίπτωση η καθυστέρηση στη ζεύξη μεταξύ εκρηκτικής πηγής και αποστολέα είναι 500 msec. Στο δεύτερο σενάριο η εκρηκτική κίνηση είναι UDP κίνηση που παράγεται από μια πηγή on – off που παράγει 10 πακέτα δεδομένων κάθε 1.2 sec, επομένως πρόκειται για απόλυτα εκρηκτική κίνηση. Στην περίπτωση αυτή η καθυστέρηση στη ζεύξη μεταξύ εκρηκτικής πηγής και αποστολέα είναι 2 msec. Το δίκτυο αυτό φαίνεται στο παρακάτω σχήμα.



Ως κίνηση υποβάθρου (background traffic) έχουμε συνδέσεις TCP New Reno με Delayed Acknowledgements από τον αποστολέα στον παραλήπτη, των οποίων το παράθυρο συμφόρησης επιλέγεται αρκετά υψηλό ώστε να μπορεί να προκληθεί

συμφόρηση στη ζεύξη μεταξύ των δυο δρομολογητών. όλες οι συνδέσεις ξεκινούν ταυτόχρονα, με εξαίρεση την εκρηκτική η οποία ξεκινά ελαφρώς νωρίτερα ώστε να ανοίξει το παράθυρο της, και έχουν όλες διάρκεια 30 sec. Επίσης πρέπει να αναφέρουμε ότι ο μέγιστος ρυθμός μετάδοσης της εκρηκτικής πηγής είναι κατά πολύ μικρότερος από την χωρητικότητα της ζεύξης μεταξύ των δυο δρομολογητών, και κατά συνέπεια μικρότερος από αυτόν των background συνδέσεων.

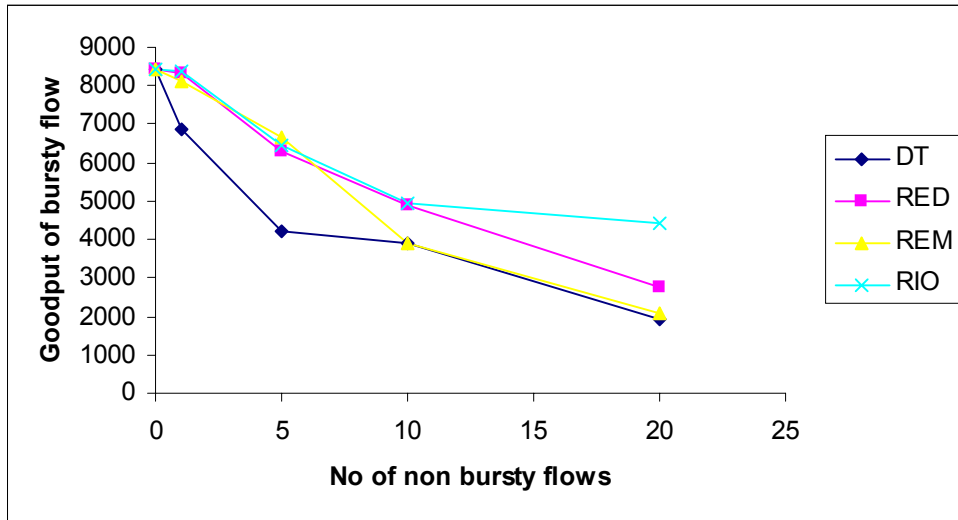
Στην ουρά της σύνδεσης μεταξύ των δυο δρομολογητών ο χώρος ενταμίευσης των πακέτων έχει μέγεθος 50 πακέτα. Στην περίπτωση που ο αλγόριθμος ενεργής διαχείρισης της ουράς (AQM) είναι ο RED έχουμε την εξής διαμόρφωση των παραμέτρων : min threshold 15 πακέτα, max threshold 30 πακέτα, max dropping probability 0.1 και queue weight 0.005 στον αλγόριθμο Exponential Weighted Moving Average. Όταν χρησιμοποιούμε τον αλγόριθμο REM για AQM έχουμε τις εξής τιμές παραμέτρων : update time 0,002 sec , gamma 0.001 , phi 1.001 , b_0 20 πακέτα. Στην περίπτωση που ο αλγόριθμος AQM είναι ο RIO έχουμε : min in threshold 10 πακέτα , max in threshold 30 πακέτα , max in dropping probability 0.05 , min out threshold 15 πακέτα , max out threshold 25 πακέτα , max out dropping probability 0.5 και βάρος αλγορίθμου EWMA queue weight 0.005. Επίσης στην περίπτωση του RIO χρησιμοποιήσαμε την εξής πολιτική κατηγοριοποίησης των πακέτων σε in και out : αναθέτουμε σε κάθε σύνδεση εύρος ίσο με την χωρητικότητα της ζεύξης προς τον συνολικό αριθμό των συνδέσεων και τα πακέτα πάνω από αυτό τον ρυθμό είναι μη σύμφωνα, ενώ όλα τα πακέτα της εκρηκτικής πηγής είναι σύμφωνα.

Εκτελέσαμε και για τα δυο σενάρια προσομοιώσεις με καμία, μία, πέντε, δέκα και είκοσι background TCP συνδέσεις για να δούμε την συμπεριφορά του δικτύου σε διαφορετικές συνθήκες κίνησης και για όλους τους υπό εξέταση αλγορίθμους ενεργής διαχείρισης ουρών. από τα αποτελέσματα που λάβαμε παρατηρούμε τα εξής :

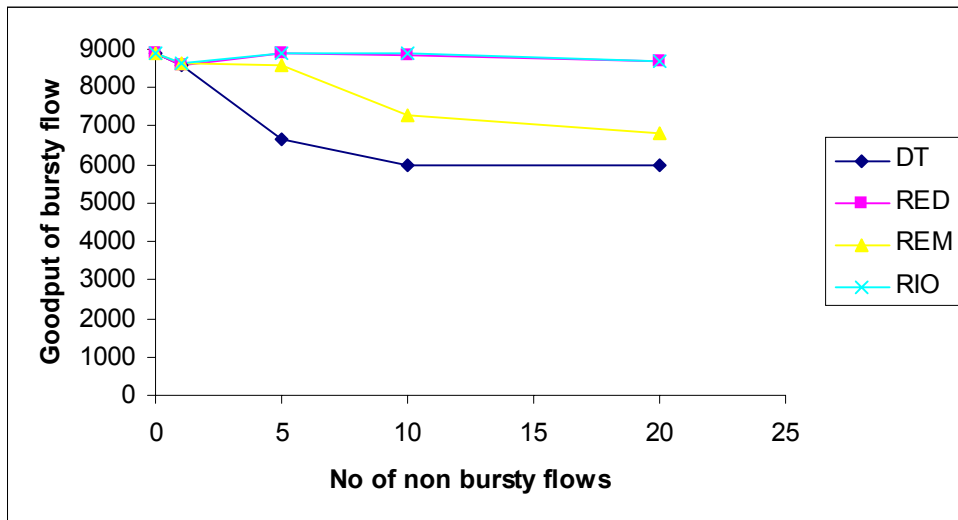
3.1 Δικαιοσύνη μεταξύ των ροών (fairness)

- Όταν η ουρά της ζεύξης μεταξύ των δυο δρομολογητών είναι Drop Tail το throughput της εκρηκτικής πηγής μειώνεται όσο αυξάνονται οι background TCP συνδέσεις. αυτό οφείλεται στο γεγονός ότι όσο αυξάνεται ο αριθμός των συνδέσεων αυξάνεται και το μήκος της ουράς, οπότε έχοντας σχεδόν πλήρη ουρά τα πακέτα της εκρηκτικής πηγής τα οποία έρχονται πολλά μαζεμένα την υπερχειλίζουν και ένας μεγάλος αριθμός από αυτά χάνονται.
- χρησιμοποιώντας κάποιον από τους αλγορίθμους AQM κρατάμε την ουρά σε χαμηλότερο επίπεδο, έτσι ώστε να μπορεί να απορροφήσει σποραδικές ριπές πακέτων. έτσι το throughput που επιτυγχάνει η εκρηκτική πηγή σε αυτή την περίπτωση είναι αρκετά μεγαλύτερο από όσο ήταν με Drop Tail ουρά.
- Όταν έχουμε απόλυτα εκρηκτική πηγή, δηλαδή όταν αυτή είναι πηγή on – off που παράγει UDP πακέτα, το throughput που αυτή επιτυγχάνει σε συνθήκες συμφόρησης είναι μεγαλύτερο αυτό που επιτυγχάνει μια σχετικά εκρηκτική κίνηση (TCP με μεγάλο RTT και μικρό παράθυρο). αυτό οφείλεται σε μεγάλο ποσοστό στο ότι η TCP πηγή μειώνει το παράθυρο της όταν ανιχνεύσει κάποιο χαμένο πακέτο, ενώ η πηγή UDP διατηρεί τον ίδιο ρυθμό μετάδοσης ανεξάρτητα από το ποσά πακέτα χάνονται.
- Στην περίπτωση UDP εκρηκτικής κίνησης οι αλγόριθμοι RED και RIO δίνουν σχεδόν σταθερό throughput ανεξάρτητα από το αν υπάρχει background κίνηση και τον αριθμό των πηγών αυτών.
- Ο αλγόριθμος REM δίνει μεγαλύτερο throughput στην εκρηκτική σύνδεση από αυτό που παίρνει στην περίπτωση της Drop Tail ουράς, αλλά μικρότερο από όσο δίνουν οι αλγόριθμοι RED και RIO. Μάλιστα όταν υπάρχει πολύ έντονη συμφόρηση (μεγάλος αριθμός background πηγών) η απόδοση του REM πέφτει και πλησιάζει στα επίπεδα του Drop Tail.
- Με τον αλγόριθμο RIO η εκρηκτική πηγή παίρνει ικανοποιητικό μερίδιο της χωρητικότητας της ζεύξης ακόμη και σε μεγάλη συμφόρηση, αφού όλα τα πακέτα της είναι in packets και έτσι έχουν μικρή πιθανότητα να απορριφθούν.

Throughput που επιτυγχάνει η εκρηκτική πηγή συναρτήσει του αριθμού των background πηγών
 ανά αλγόριθμο AQM στην περίπτωση TCP εκρηκτικής πηγής



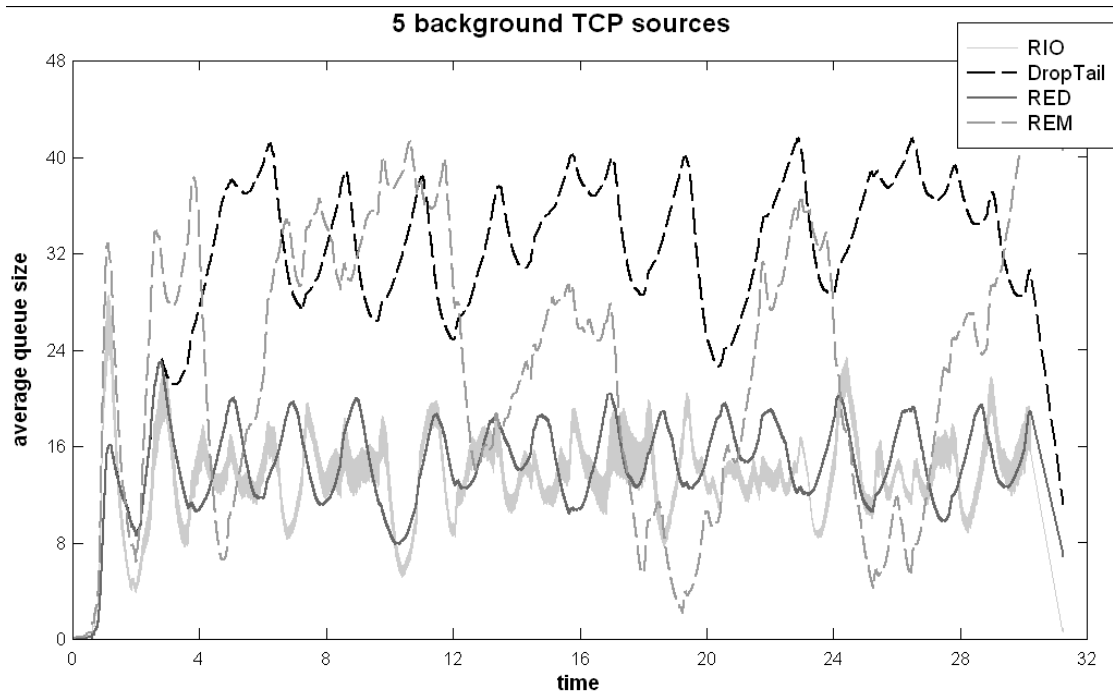
Throughput που επιτυγχάνει η εκρηκτική πηγή συναρτήσει του αριθμού των background πηγών
 ανά αλγόριθμο AQM στην περίπτωση UDP εκρηκτικής πηγής



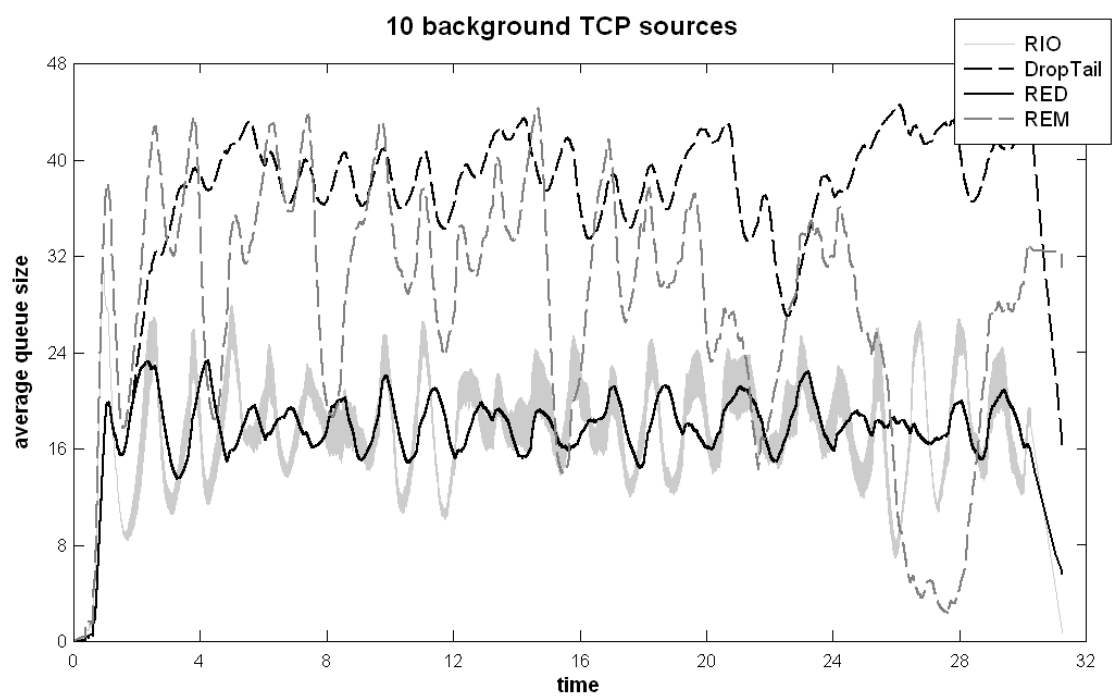
3.2 Μέσο μήκος ουράς

- Όταν έχουμε εκρηκτική κίνηση, δηλαδή πολλά πακέτα να καταφθάνουν μέσα σε μικρό χρονικό διάστημα, θέλουμε να υπάρχει χώρος στην ουρά του δρομολογητή ώστε να απορροφηθούν όλα αυτά τα πακέτα. επομένως θέλουμε το μέσο μήκος ουράς να κινείται μέσα σε κάποια συγκεκριμένα όρια, χαμηλότερα από το μέγιστο μήκος αυτής.
- Δεν παρατηρούμε διαφορές στο μέσο μήκος ουράς σε σχέση με το είδος της εκρηκτικής κίνησης, αφού το μεγαλύτερο ποσοστό των πακέτων που διακινούνται προέρχεται από τις background συνδέσεις.
- Στην περίπτωση που η ουρά είναι Drop Tail το μέσο μήκος της πλησιάζει το μέγιστο, επομένως δεν υπάρχει ελεύθερος χώρος για απορροφηθούν ριπές πακέτων.
- Στην περίπτωση που χρησιμοποιείται κάποιος από τους RED ή RIO αλγορίθμους ενεργής διαχείρισης ουρών, το μέσο μήκος ουράς είναι χαμηλότερο από το μέγιστο, μεταξύ των κατωφλιών που έχουν δοθεί ως παράμετροι στους αλγορίθμους.
- Όταν ο χρησιμοποιούμενος αλγόριθμος AQM είναι ο REM το μέσο μήκος ουράς είναι σε χαμηλότερα επίπεδα από την περίπτωση που η ουρά μας είναι Drop Tail, όμως παρουσιάζει αρκετή διακύμανση και σε ορισμένα διαστήματα, κατά τη διάρκεια της προσομοίωσης, πλησιάζει το μέγιστο μήκος ουράς. Μάλιστα όσο αυξάνεται η συμφόρηση, δηλαδή ο αριθμός των background συνδέσεων, τόσο το μέσο μήκος ουράς πλησιάζει το μέγιστο, χειροτερεύοντας την απόδοση του αλγορίθμου.

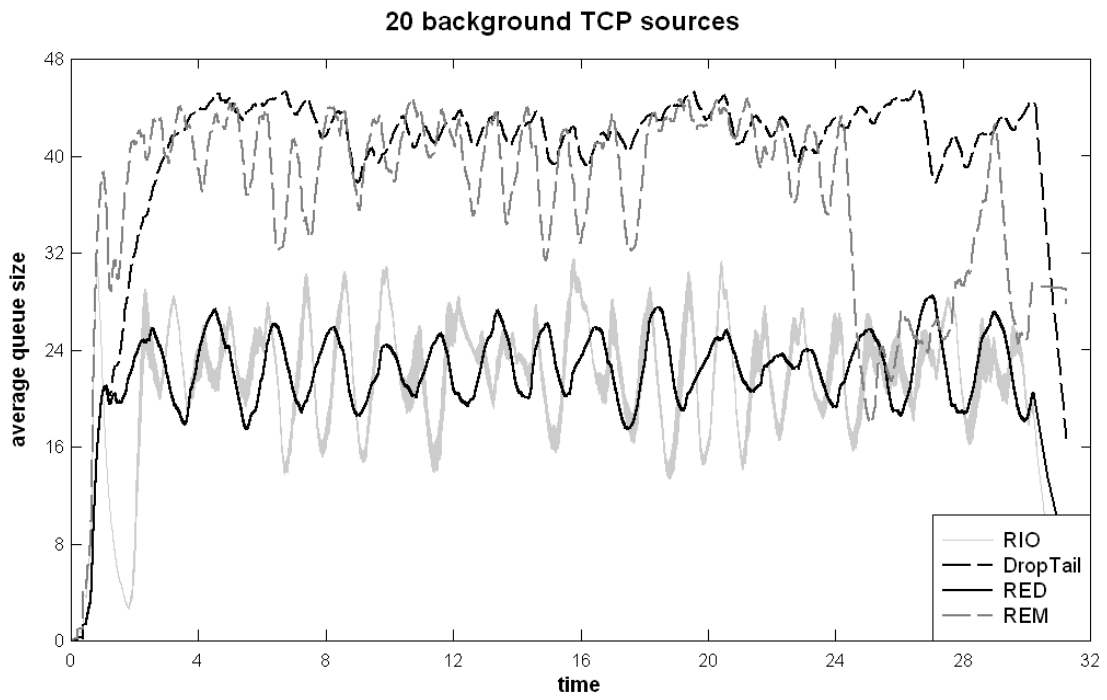
μέσο μήκος ουράς συναρτήσει του χρόνου ανά αλγόριθμο AQM
με 5 background συνδέσεις



μέσο μήκος ουράς συναρτήσει του χρόνου ανά αλγόριθμο AQM
με 10 background συνδέσεις

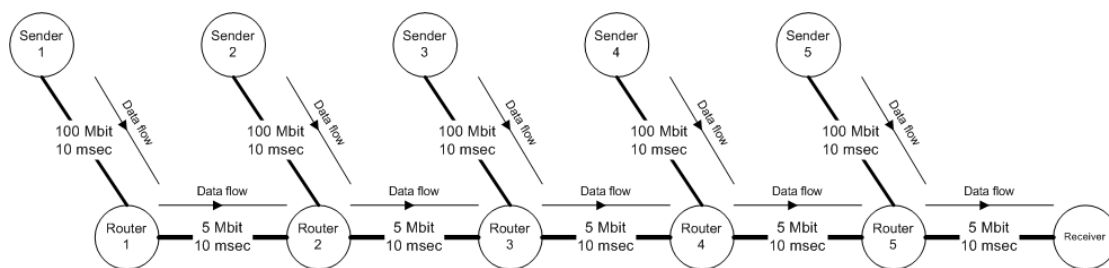


μέσο μήκος ουράς συναρτήσει του χρόνου ανά αλγόριθμο AQM
με 20 background συνδέσεις



4. Πολλαπλά Bottleneck Links

Στην τρίτη τοπολογία εξετάζουμε ένα ελαφρώς πιο σύνθετο δίκτυο. Αποτελείται από πέντε δρομολογητές, οι οποίοι συνδέονται σε σειρά ο ένας με τον άλλο ($1^{\text{ος}}$ με $2^{\text{ο}}$, $2^{\text{ος}}$ με $3^{\text{ο}}$ κλπ) με ζεύξεις ταχύτητας 5 Mbit/sec και καθυστέρησης 10 msec η καθεμιά. Σε κάθε έναν από αυτούς συνδέεται από ένας κόμβος – αποστολέας μέσω γραμμής ταχύτητας 5 Mbit/sec και καθυστέρησης 10 msec, από όπου ξεκινούν συνδέσεις δεδομένων. Ο πέμπτος δρομολογητής (και μέσω αυτού όλοι οι υπόλοιποι) συνδέεται με έναν κόμβο – παραλήπτη, στον οποίο καταλήγουν όλες οι συνδέσεις, με ζεύξη ταχύτητας 5 Mbit/sec και καθυστέρησης 10 msec. Το δίκτυο αυτό φαίνεται στο παρακάτω σχήμα :

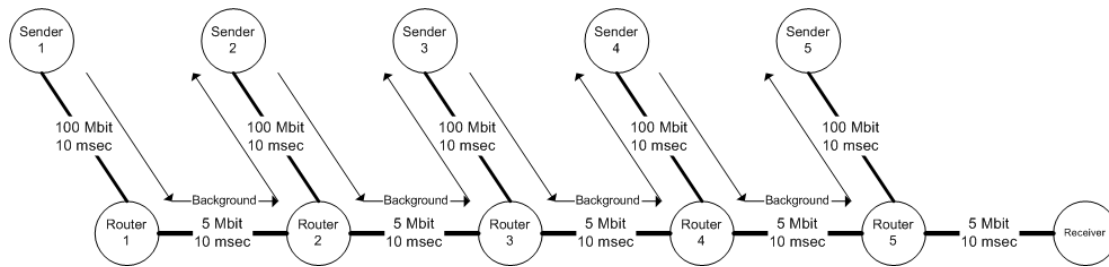


Οι συνδέσεις από τους κόμβους – αποστολείς στον παραλήπτη ακολουθούν την διαδρομή που δείχνουν τα βέλη στο παραπάνω σχήμα. Οι συνδέσεις αυτές που είναι συνδέσεις TCP ή-και TFRC ξεκινούν ταυτόχρονα και στέλνουν πακέτα των 1000 bytes για διάρκεια 30 sec. χρησιμοποιούμε την παραλλαγή New Reno για τις συνδέσεις TCP με καθυστερούμενες επιβεβαιώσεις (delayed acknowledgments) και θέτουμε το παράθυρο του παραλήπτη αρκετά μεγάλο ώστε να μπορεί να προκληθεί συμφόρηση στις αργές ζεύξεις μεταξύ των δρομολογητών και προς τον παραλήπτη. Στις συνδέσεις TFRC θέτουμε $\beta=2$ στην εξίσωση ρυθμαπόδοσης του TCP (TCP throughput equation) αφού όπως αναφέραμε οι συνδέσεις TCP χρησιμοποιούν delayed acknowledgments.

Στην ουρά κάθε δρομολογητή ο χώρος ενταμίευσης των πακέτων έχει μέγεθος 50 πακέτα. Όταν χρησιμοποιείται κάποιος αλγόριθμος ενεργής διαχείρισης της ουράς (AQM) χρησιμοποιείται σε όλους τους δρομολογητές με τις ίδιες παραμέτρους. Στην περίπτωση που ο αλγόριθμος AQM είναι ο RED έχουμε την εξής διαμόρφωση των

παραμέτρων : ελάχιστο κατώφλι min threshold 20 πακέτα, μέγιστο κατώφλι max threshold 35 πακέτα, μέγιστη πιθανότητα απόρριψης πακέτου max dropping probability 0.1 και βάρος στον αλγόριθμο Exponential Weighted Moving Average για τον υπολογισμό του μέσου μήκους ουράς queue weight 0.005. Στην περίπτωση που ο αλγόριθμος AQM είναι ο REM εξετάσαμε δυο σετ παραμέτρων. Στο πρώτο έχουμε : χρόνος ενημέρωσης του κόστους σύνδεσης update time 0,002 sec , τιμή του γ για τον υπολογισμό του κόστους gamma 0.001 , τιμή του ϕ για τον υπολογισμό της πιθανότητας απόρριψης πακέτων phi 1.001 , επιθυμητό μήκος ουράς b_0 20 πακέτα.. Στο δεύτερο σετ (REM 2) χρησιμοποιούμε gamma 0.01 και phi 1.01 , ώστε ο αλγόριθμος να αντιδρά ταχύτερα στην συμφόρηση. Όταν χρησιμοποιούμε τον αλγόριθμο RIO για AQM έχουμε τις εξής τιμές παραμέτρων : ελάχιστο κατώφλι για τα σύμφωνα πακέτα min in threshold 10 πακέτα , μέγιστο κατώφλι για τα σύμφωνα πακέτα max in threshold 30 πακέτα , μέγιστη πιθανότητα απόρριψης σύμφωνου πακέτου max in dropping probability 0.05 , ελάχιστο κατώφλι για τα μη σύμφωνα πακέτα min out threshold 5 πακέτα , μέγιστο κατώφλι για τα μη σύμφωνα πακέτα max out threshold 15 πακέτα , μέγιστη πιθανότητα απόρριψης μη σύμφωνου πακέτου max out dropping probability 0.5 , βάρος αλγορίθμου EWMA για τον υπολογισμό του μέσου μήκους ουράς queue weight 0.005. Επίσης στην περίπτωση του RIO χρησιμοποιήσαμε την εξής πολιτική κατηγοριοποίησης των πακέτων σε σύμφωνα και μη : αναθέτουμε σε κάθε σύνδεση εύρος ίσο με την χωρητικότητα της ζεύξης μεταξύ του 5^{ου} δρομολογητή και του παραλήπτη προς τον συνολικό αριθμό των συνδέσεων. Τα πακέτα κάτω από αυτό τον ρυθμό είναι μη σύμφωνα, ενώ τα πακέτα πάνω από αυτόν είναι μη σύμφωνα.

Διακρίνουμε δυο διαφορετικά σενάρια. Στο πρώτο έχουμε μόνο τις συνδέσεις από τους κόμβους – αποστολείς προς τον κόμβο παραλήπτη. έτσι η ζεύξη μεταξύ του 5^{ου} δρομολογητή και του παραλήπτη είναι εκείνη στην οποία παρουσιάζεται το μέγιστο φορτίο και η οποία καθορίζει τον ρυθμό μετάδοσης των συνδέσεων. Η διάφορα μεταξύ των συνδέσεων από διαφορετικούς κόμβους είναι το Round Trip Time, το οποίο μεγαλώνει όσο απομακρυνόμαστε από τον παραλήπτη. Στο δεύτερο σενάριο εκτός των προαναφερθέντων συνδέσεων έχουμε και συνδέσεις μεταξύ των κόμβων – αποστολέων (1^{ος} με 2^ο , 2^{ος} με 3^ο , 3^{ος} με 4^ο και 4^{ος} με 5^ο), όπως φαίνεται στο σχήμα που ακολουθεί.



Αυτές οι συνδέσεις (background traffic) έχουν ως σκοπό να δημιουργήσουν συμφόρηση και στις ζεύξεις μεταξύ των δρομολογητών. Έτσι οι συνδέσεις από κόμβους που βρίσκονται πιο μακριά από τον παραλήπτη διέρχονται μέσω περισσότερων υπό συμφόρηση ζεύξεων και άρα έχουν μεγαλύτερες απώλειες πακέτων. Χρησιμοποιούμε συνδέσεις TCP ή-και TFRC, ανάλογα με το τι κίνηση έχουμε από τους αποστολείς στον παραλήπτη, οι οποίες έχουν τα ίδια χαρακτηριστικά με τις υπόλοιπες, ώστε να ανταγωνίζονται επί ίσοις όροις με αυτές.

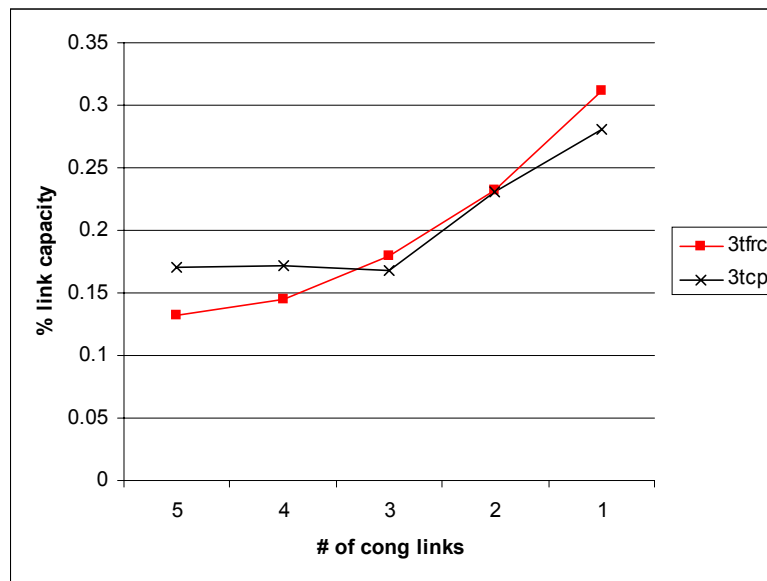
Εκτελέσαμε προσομοιώσεις με διάφορους συνδυασμούς ροών, με τον ίδιο αριθμό και είδος συνδέσεων από κάθε κόμβο – αποστολέα. Μόνο συνδέσεις TCP (μια ή τρεις μαζί), μόνο συνδέσεις TFRC (μια ή τρεις μαζί), ή και τους δυο τύπους μαζί (μια από κάθε είδος, τρεις από κάθε είδος, μια με τρεις και τρεις με μια). Από τα αποτελέσματα των προσομοιώσεων παρατηρούμε τα εξής :

4.1 Απουσία *background* κίνησης (διαφοροποίηση των ροών με βάση το RTT)

4.1.1 Δικαιοσύνη μεταξύ των ροών (fairness)

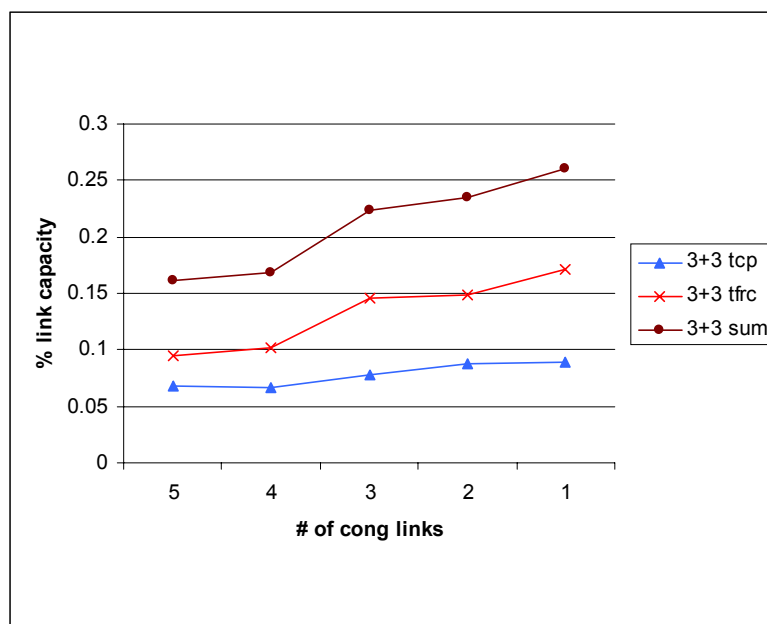
- Οι συνδέσεις με μικρό RTT έχουν την τάση να παίρνουν μεγαλύτερο μερίδιο της χωρητικότητας της ζεύξης από αυτό που καταλαμβάνουν συνδέσεις με μεγαλύτερο RTT. Μάλιστα όσο μεγαλύτερο είναι το RTT μιας σύνδεσης τόσο μικρότερο μερίδιο της χωρητικότητας παίρνει. αυτό οφείλεται στον μηχανισμό με τον οποίο ο αλγόριθμος ελέγχου συμφόρησης αυξάνει τον ρυθμό μετάδοσης. Στις TCP συνδέσεις το παράθυρο συμφόρησης, όταν βρισκόμαστε στην φάση της αποφυγής συμφόρησης (congestion avoidance), αυξάνεται με ρυθμό ενός τμήματος πλήρους μεγέθους (full size segment) ανά RTT, έτσι οι συνδέσεις με μεγαλύτερη καθυστέρηση αυξάνουν πιο αργά τον ρυθμό μετάδοσης τους. Με την ανίχνευση όμως κάποιου χαμένου πακέτου το παράθυρο πέφτει στο μισό ανεξάρτητα του RTT. Επομένως οι TCP συνδέσεις με μικρότερο RTT είναι πιο επιθετικές και καταλαμβάνουν μεγαλύτερο εύρος ζώνης. Στις TFRC συνδέσεις ο αποστολέας μεταβάλλει τον ρυθμό μετάδοσης του κάθε φορά που λαμβάνει κάποιο πακέτο ανάδρασης (feedback packet). Ο παραλήπτης, όταν δεν υπάρχουν απώλειες και άρα ο αποστολέας θα πρέπει να αυξήσει τον ρυθμό του, στέλνει ένα πακέτο ανάδρασης κάθε RTT. Σε περίπτωση απώλειας πακέτου όμως, στέλνεται άμεσα πακέτο ανάδρασης ώστε να μειωθεί ο ρυθμός μετάδοσης. άρα και οι TFRC συνδέσεις με μικρότερο RTT είναι πιο επιθετικές και έτσι καταλαμβάνουν μεγαλύτερο εύρος ζώνης.

Ποσοστό της διαθέσιμης χωρητικότητας που καταλαμβάνει κάθε είδος σύνδεσης συναρτήσει του αριθμού των ενδιάμεσων αργών ζεύξεων με κίνηση μόνο TCP ή μόνο TFRC



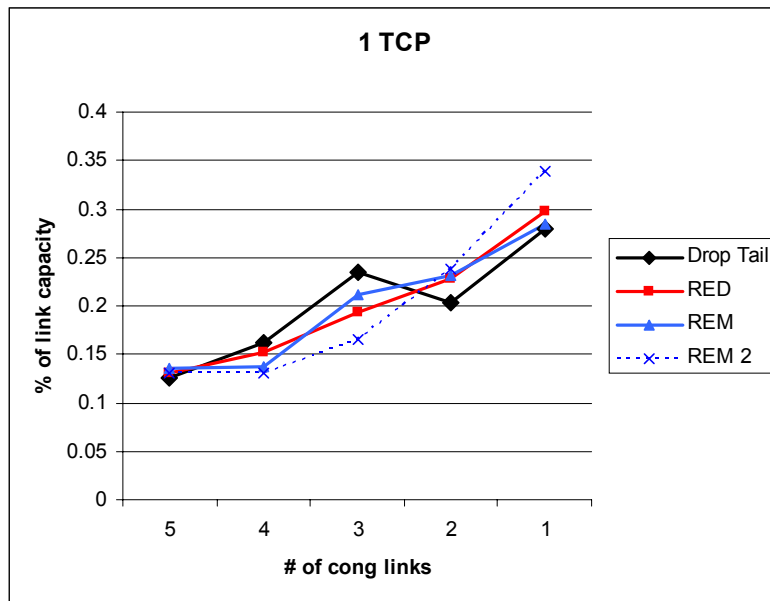
- Οι συνδέσεις TFRC όταν ανταγωνίζονται με συνδέσεις TCP επιτυγχάνουν μεγαλύτερη ρυθμαπόδοση (throughput) από τις συνδέσεις TCP που προέρχονται από τον αντίστοιχο κόμβο και άρα έχουν το ίδιο RTT.

Ποσοστό της διαθέσιμης χωρητικότητας που καταλαμβάνει κάθε είδος σύνδεσης συναρτήσει του αριθμού των ενδιάμεσων αργών ζεύξεων με κίνηση TCP και TFRC μαζί

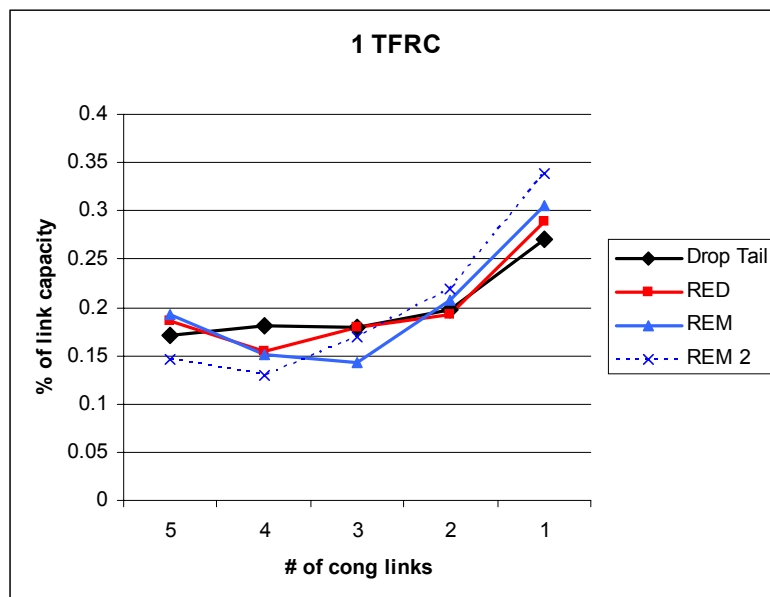


- Οι αλγόριθμοι ενεργής διαχείρισης ουρών RED και REM δεν παρουσιάζουν αξιοσημείωτες διαφορές σε σχέση με μια Drop Tail ουρά, αφού όπως αναφέρθηκε η άνιση κατανομή της χωρητικότητας μεταξύ συνδέσεων διαφορετικού RTT οφείλεται στους μηχανισμούς ελέγχου συμφόρησης και όχι στη δυναμική κατάσταση της ουράς.

Ποσοστό της διαθέσιμης χωρητικότητας που καταλαμβάνει κάθε είδος σύνδεσης συναρτήσει του αριθμού των ενδιάμεσων αργών ζεύξεων με κίνηση TCP ανά αλγόριθμο AQM

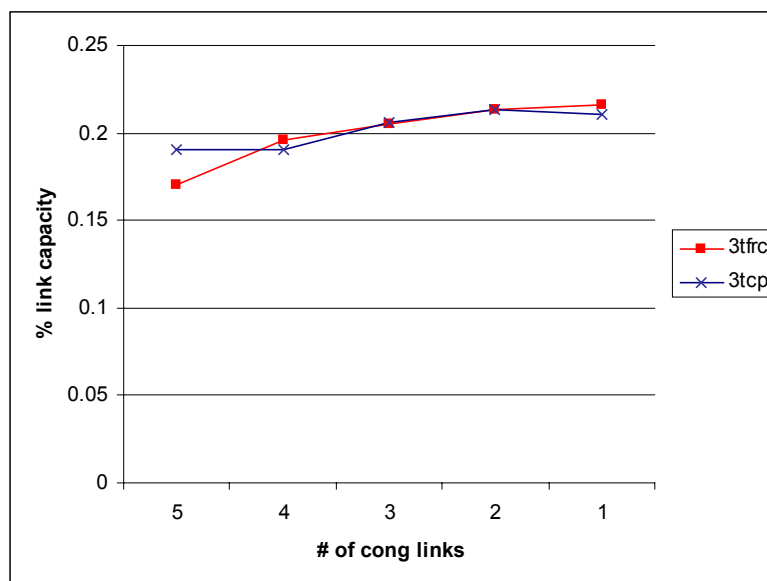


Ποσοστό της διαθέσιμης χωρητικότητας που καταλαμβάνει κάθε είδος σύνδεσης συναρτήσει του αριθμού των ενδιάμεσων αργών ζεύξεων με κίνηση TFRC ανά αλγόριθμο AQM

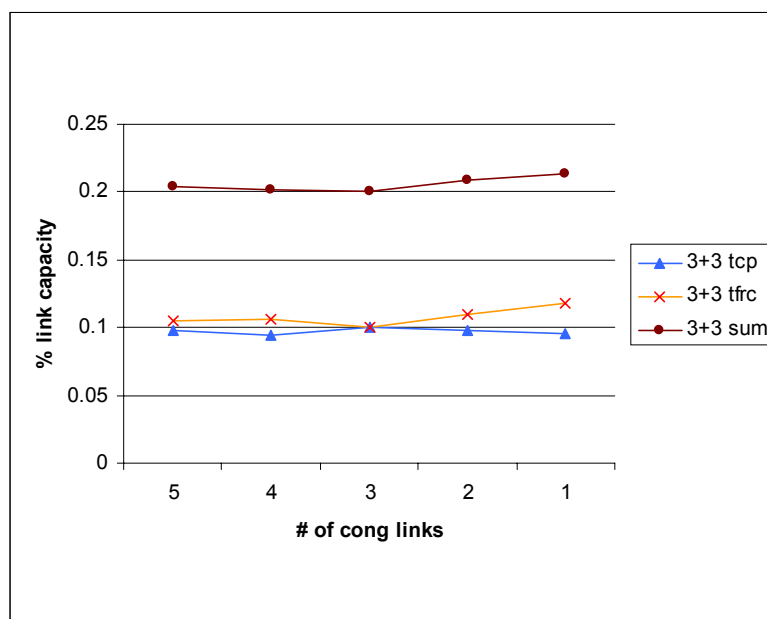


- Ο αλγόριθμος RIO επιτυγχάνει την σχετικά δίκαιη κατανομή του διαθέσιμου εύρους ζώνης τόσο μεταξύ συνδέσεων με διαφορετικό RTT όσο και μεταξύ των συνδέσεων TCP και TFRC. Το γεγονός αυτό οφείλεται στο ότι αναθέτουμε το ίδιο εύρος ζώνης σε κάθε σύνδεση και μάλιστα ίσο με το δίκαια μοιρασμένο εύρος ζώνης της αργής ζεύξης.

Ποσοστό της διαθέσιμης χωρητικότητας που καταλαμβάνει κάθε είδος σύνδεσης με RIO AQM συναρτήσει του αριθμού των ενδιάμεσων αργών ζεύξεων με κίνηση TCP ή TFRC

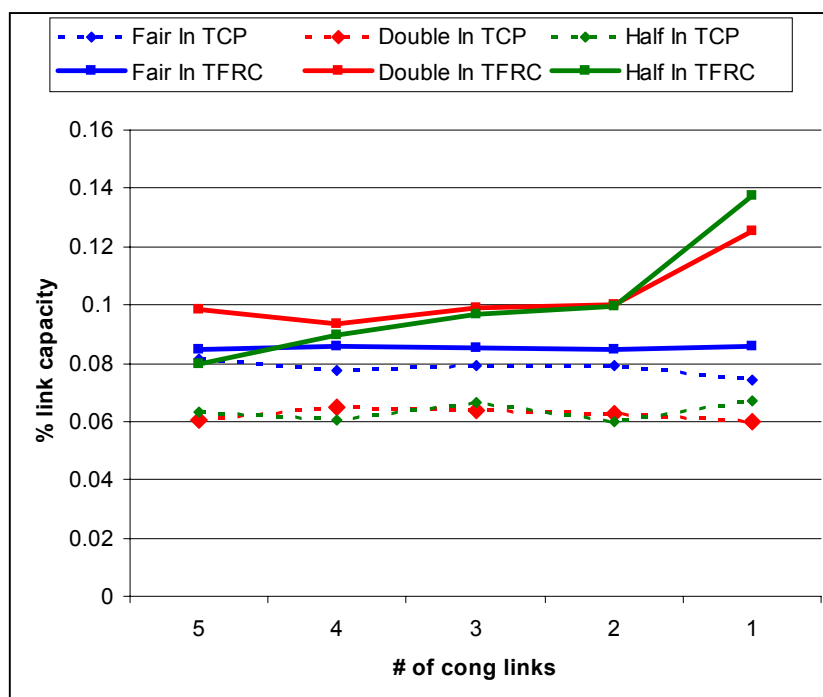


Ποσοστό της διαθέσιμης χωρητικότητας που καταλαμβάνει κάθε είδος σύνδεσης με RIO AQM συναρτήσει του αριθμού των ενδιάμεσων αργών ζεύξεων με κίνηση TCP και TFRC μαζί



- Στην περίπτωση που αναθέσουμε σε κάθε σύνδεση ως επιθυμητό ρυθμό μετάδοσης το μισό από τον ρυθμό που αναλογεί με βάση την ίση κατανομή των πόρων, όλες οι συνδέσεις καταλαμβάνουν την προβλεπόμενη χωρητικότητα και διεκδικούν τους υπόλοιπους πόρους δυναμικά, με αποτέλεσμα αυτοί να κατανέμονται άνισα και ανάλογα με το RTT και το είδος της σύνδεσης.
- Στην περίπτωση που αναθέτουμε τον διπλάσιο από τον ισοκατανεμημένο ρυθμό οι συνδέσεις με μικρότερο RTT καταλαμβάνουν μεγαλύτερο ποσοστό της χωρητικότητας, όπως και στην περίπτωση των υπολοίπων αλγορίθμων AQM, με άνω φράγμα τον ρυθμό που αναθέσαμε σε αυτές.

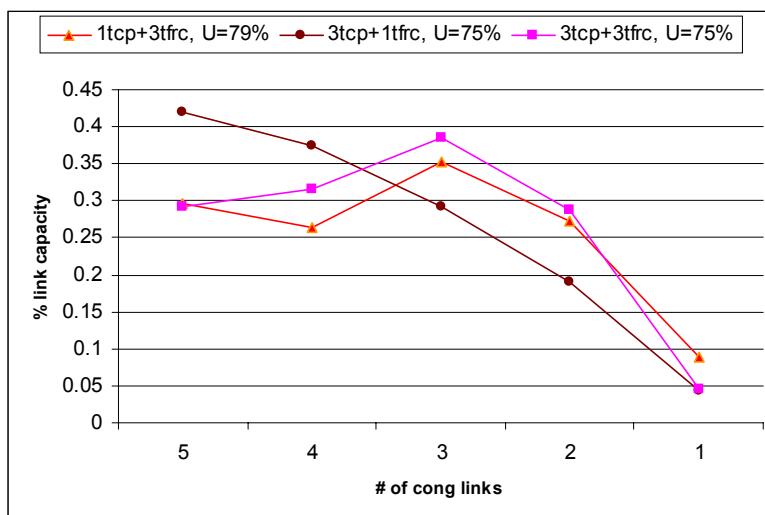
Ποσοστό της διαθέσιμης χωρητικότητας που καταλαμβάνει κάθε είδος σύνδεσης με RIO AQM συναρτήσει του αριθμού των ενδιάμεσων αργών ζεύξεων με κίνηση TCP και TFRC μαζί για διαφορετικές αναθέσεις επιθυμητών ρυθμών μετάδοσης



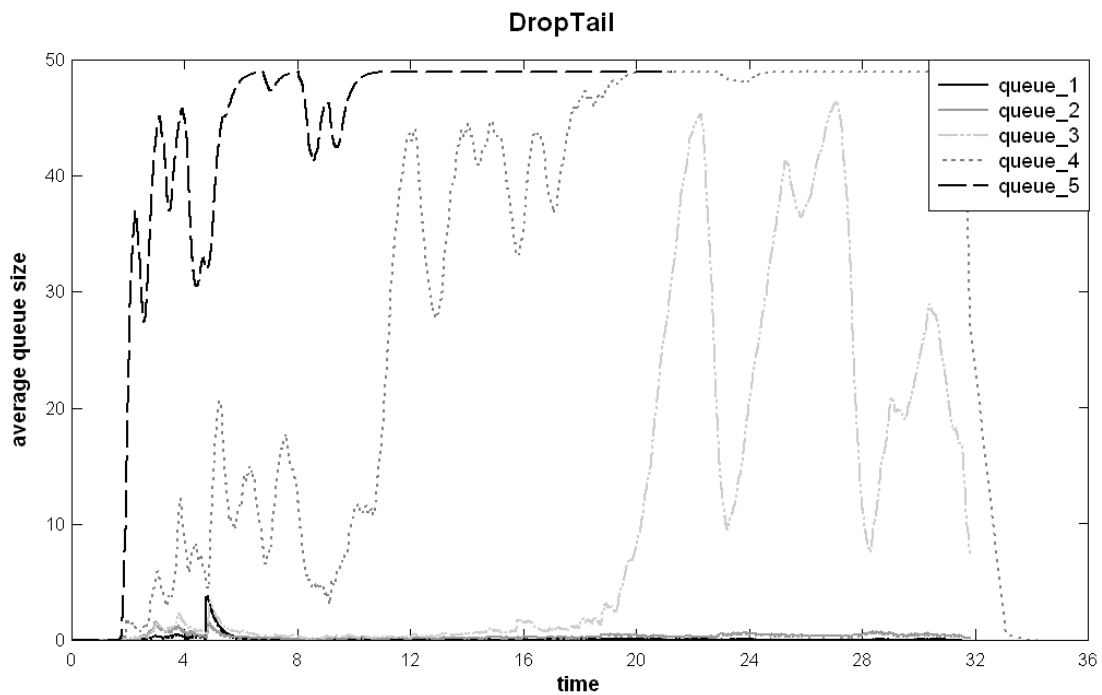
4.1.2 Αποκλεισμός

- Με Drop Tail ουρά παρατηρείται αποκλεισμός ροών που προέρχονται κυρίως από τους πιο κοντινούς στον παραλήπτη κόμβους. Αυτό οφείλεται στο ότι οι ροές αυτές, υποφέρουν από πολλές απορρίψεις πακέτων μέσα σε μικρά χρονικά διαστήματα, με αποτέλεσμα να ρίχνουν το ρυθμό μετάδοσης τους. Πιο συγκεκριμένα η ουρά του τελευταίου δρομολογητή είναι συνεχώς γεμάτη. Τα πακέτα του τελευταίου αποστολέα φτάνουν στον δρομολογητή αυτό μέσω πιο γρήγορης ζεύξης απ' ότι τα υπόλοιπα (100Mbit έναντι 5Mbit). Αυτό έχει ως αποτέλεσμα η κίνηση αυτή να είναι εκρηκτικής μορφής και επομένως να έχει περισσότερες απώλειες, όποτε ο ρυθμός μετάδοσής της γίνεται υπερβολικά χαμηλός. Ως επακόλουθο αυτού προκύπτει συμφόρηση και στον αμέσως προηγούμενο δρομολογητή. Έτσι η κίνηση που προέρχεται από τον κόμβο που συνδέεται άμεσα και με γρηγορότερη ζεύξη σε αυτόν αντιμετωπίζεται ως εκρηκτική. Με αυτόν τον τρόπο δημιουργείται μια αλυσιδωτή αντίδραση που έχει ως επακόλουθο ροές με μικρότερο RTT να έχουν μικρότερο throughput, παρ' όλο που υπό άλλες συνθήκες θα έπρεπε να είναι οι πλέον γρήγορες. Ο μικρός ρυθμός μετάδοσης των ροών αυτών μπορεί να οδηγήσει σε υποχρησιμοποίηση του δικτύου.

Ποσοστό της διαθέσιμης χωρητικότητας που καταλαμβάνουν οι σύνδεσης κάθε κόμβου και ολική χρησιμοποίηση της ζεύξης συναρτήσει του αριθμού των ενδιάμεσων αργών ζεύξεων με κίνηση TCP και TFRC μαζί σε διάφορους συνδυασμούς στην περίπτωση Drop Tail ουράς

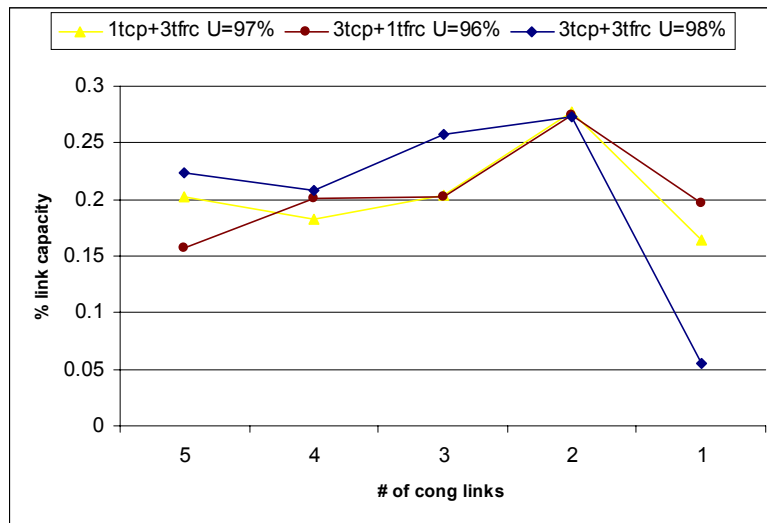


Μέσο μήκος ουράς κάθε δρομολογητή σε συνάρτηση με το χρόνο
με κίνηση TCP και TFRC μαζί στην περίπτωση Drop Tail ουράς

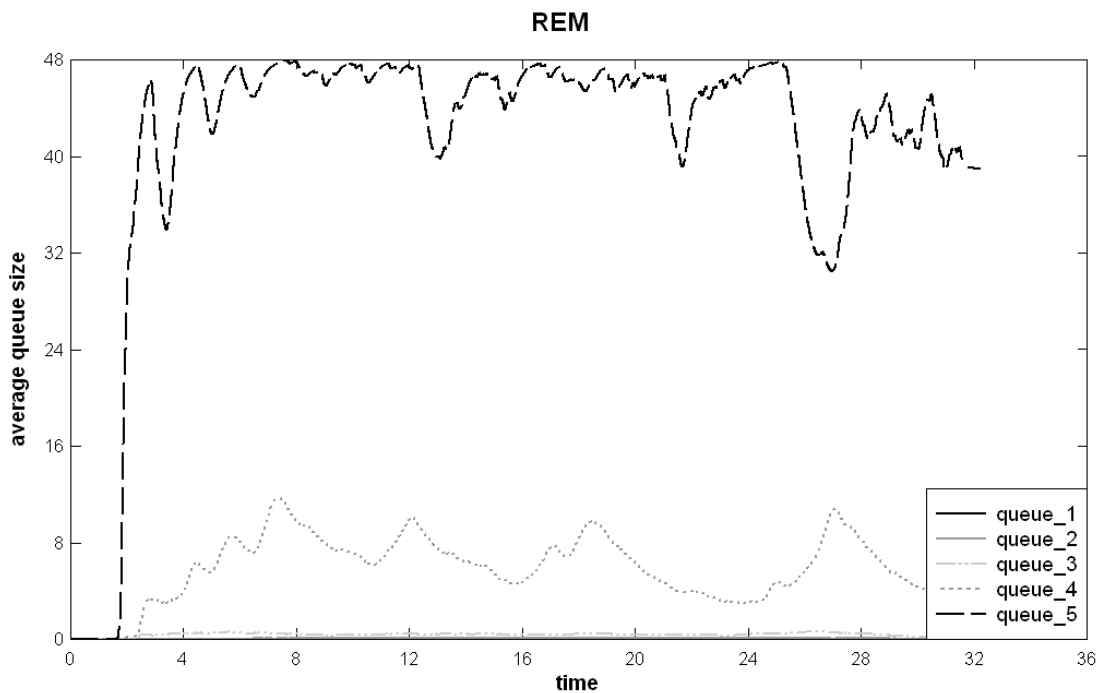


- Όταν χρησιμοποιείται ο αλγόριθμος REM για την διαχείριση των ουρών παρατηρούμε ότι και πάλι οι συνδέσεις που προέρχονται από τον πιο κοντινό στο παραλήπτη κόμβο παίρνουν μικρότερο μερίδιο των πόρων του δικτύου, όμως η ολική χρησιμοποίηση είναι υψηλή. αυτό οφείλεται στο ότι τα πακέτα από τον κόμβο αυτό φτάνουν στον τελευταίο δρομολογητή με γρηγορότερο ρυθμό απ' ότι τα υπόλοιπα, αφού η γραμμή που συνδέει τον υπ' όψιν κόμβο με τον δρομολογητή είναι πιο γρήγορη από τις γραμμές μεταξύ των δρομολογητών. έτσι η πιθανότητα απόρριψης για αυτά τα πακέτα είναι μεγαλύτερη, αφού η τελευταία εξαρτάται από τον ρυθμό των εισερχομένων πακέτων.

Ποσοστό της διαθέσιμης χωρητικότητας που καταλαμβάνουν οι σύνδεσης κάθε κόμβου και ολική χρησιμοποίηση της ζεύξης συναρτήσει του αριθμού των ενδιάμεσων αργών ζεύξεων με κίνηση TCP και TFRC μαζί σε διάφορους συνδυασμούς στην περίπτωση REM ουράς με $\gamma=0.001$ και $\phi=1.001$

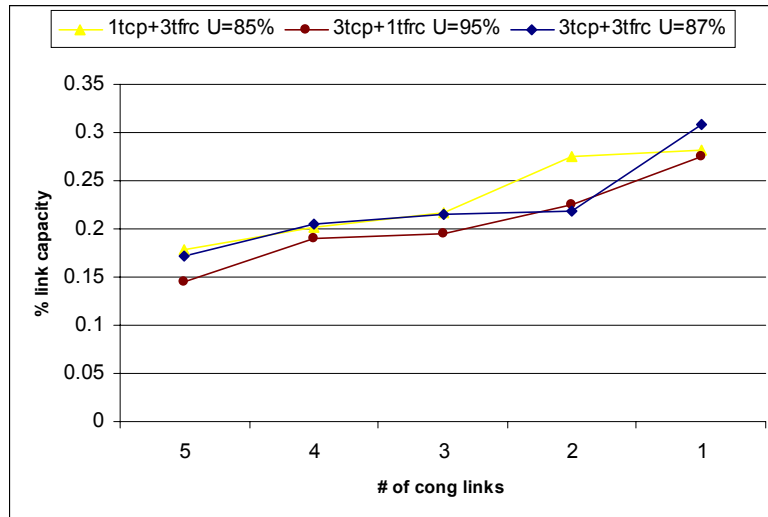


Μέσο μήκος ουράς κάθε δρομολογητή σε συνάρτηση με το χρόνο με κίνηση TCP και TFRC μαζί στην περίπτωση REM ουράς με $\gamma=0.001$ και $\phi=1.001$

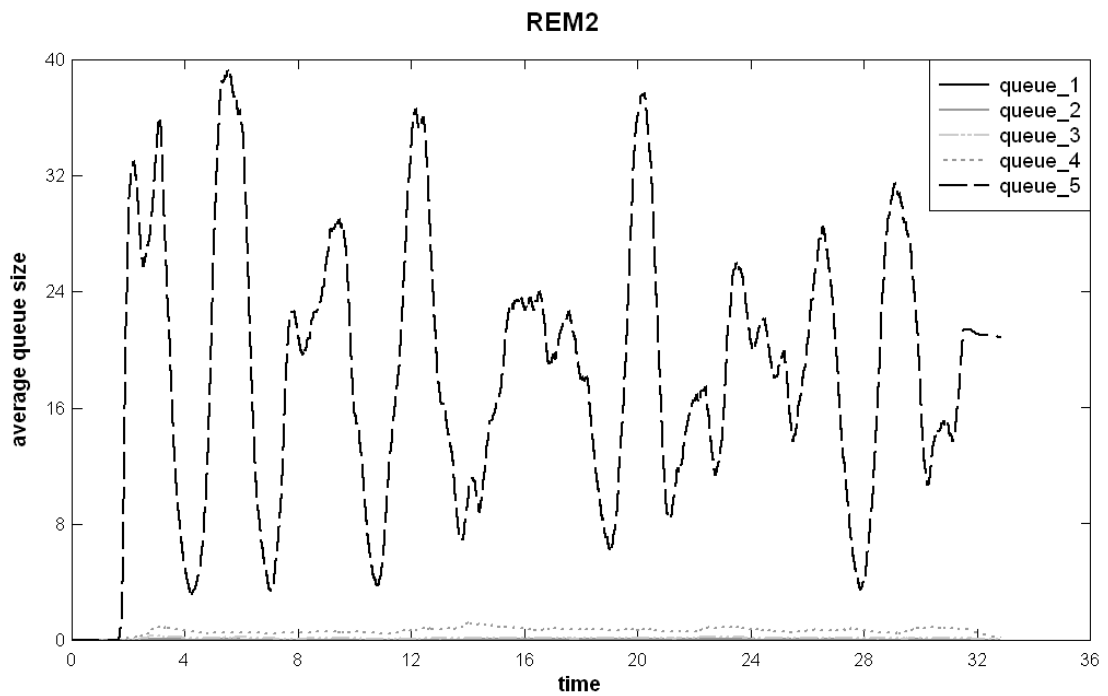


- Για την αντιμετώπιση του αποκλεισμού συγκεκριμένων ροών μπορεί να επιλεγεί ένα διαφορετικό σετ παραμέτρων για τον αλγόριθμο REM, με κόστος μια μικρή μείωση της συνολικής χρησιμοποίησης.

Ποσοστό της διαθέσιμης χωρητικότητας που καταλαμβάνουν οι σύνδεσεις κάθε κόμβου και ολική χρησιμοποίηση της ζεύξης συναρτήσει του αριθμού των ενδιάμεσων αργών ζεύξεων με κίνηση TCP και TFRC μαζί σε διάφορους συνδυασμούς στην περίπτωση REM ουράς με $\gamma=0.001$ και $\phi=1.001$



Μέσο μήκος ουράς κάθε δρομολογητή σε συνάρτηση με το χρόνο με κίνηση TCP και TFRC μαζί στην περίπτωση REM ουράς με $\gamma=0.001$ και $\phi=1.001$



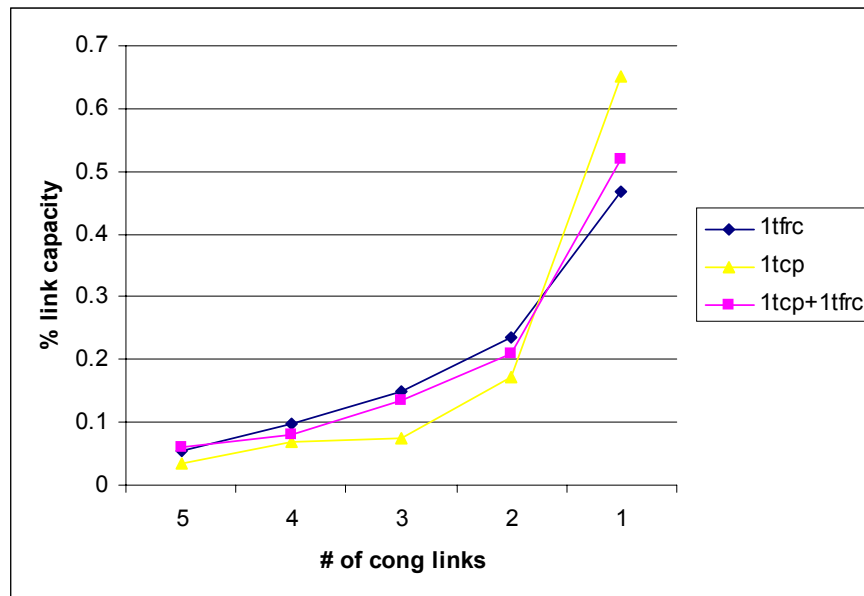
4.2 Ύπαρξη background κίνησης (διαφοροποίηση των ροών ως προς τον αριθμό των υπό συμφόρηση ζεύξεων από τις οποίες διέρχονται)

4.2.1 Δικαιοσύνη μεταξύ των ροών (fairness)

- Ροές οι οποίες διέρχονται από περισσότερες υπό συμφόρηση ζεύξεις παίρνουν μικρότερο μερίδιο της χωρητικότητας της ζεύξης μεταξύ του πέμπτου δρομολογητή και του παραλήπτη. αυτό οφείλεται στο ότι οι ροές αυτές συναντούν συμφόρηση και στις ζεύξεις μεταξύ των προηγούμενων δρομολογητών. έτσι πακέτα από αυτές απορρίπτονται και στις ζεύξεις αυτές, με αποτέλεσμα να πέφτει περισσότερο ο ρυθμός μετάδοσής τους, ως απόκριση στην αυξημένη συμφόρηση που συναντούν.
- Το παραπάνω φαινόμενο δεν συνεπάγεται απαραίτητα και απουσία δικαιοσύνης. Σύμφωνα με μια διαφορετική προσέγγιση, κάθε ροή θα πρέπει να αντιμετωπίζεται σύμφωνα με τους πόρους του δικτύου που καταλαμβάνει. έτσι ροές που απαιτούν λιγότερους πόρους του δικτύου θα πρέπει να αντιμετωπίζονται ευνοϊκότερα από άλλες που απαιτούν περισσότερους πόρους από το δίκτυο. Η ιδανική περίπτωση κατανομής των πόρων στο σενάριο που εξετάζουμε, σύμφωνα με την παραπάνω τακτική, θα ήταν το εύρος ζώνης κάθε σύνδεσης να ήταν αντιστρόφως ανάλογο του αριθμού των υπό συμφόρηση ζεύξεων που συναντά. Επομένως το διάγραμμα της χωρητικότητας συναρτήσει του αριθμού των υπό συμφόρηση ζεύξεων θα πρέπει να προσεγγίζει μια ευθεία υπό κλίση.
- Στην περίπτωση που οι ουρές στους δρομολογητές είναι Drop Tail παρατηρούμε ότι η κατανομή του εύρους ζώνης δεν είναι γραμμική αλλά εκθετική, με τις ροές που διέρχονται από μια μόνο υπό συμφόρηση ζεύξη να παίρνουν πολύ μεγάλο μερίδιο του διαθέσιμου εύρους ζώνης.
- Στην περίπτωση που η κίνηση είναι αποκλειστικά TCP έχουμε πιο άνιση κατανομή, από την περίπτωση που έχουμε κίνηση μόνο TFRC ή μικτή κίνηση. αυτό είναι αποτέλεσμα του αλγόριθμου μεταβολής του ρυθμού μετάδοσης του

TFRC. όταν ανιχνευθεί κάποια απώλεια πακέτου από το TCP, το παράθυρο συμφόρησης πέφτει στο μισό, ρίχνοντας τον ρυθμό μετάδοσης στιγμιαία στο ήμισυ, σε αντίθεση με το TFRC, το οποίο αντιδρά στα χαμένα πακέτα με μικρή μείωση του ρυθμού του. έτσι οι συνδέσεις TCP ρίχνουν τον ρυθμό μετάδοσης τους γεωμετρικά σε σχέση με τον αριθμό των χαμένων πακέτων (αποτέλεσμα του AIMD αλγόριθμου), ενώ οι συνδέσεις TFRC προσεγγίζουν περισσότερο τη γραμμική μείωση. Αυτή η διαφορά απεικονίζεται και στον τρόπο κατανομής του διαθέσιμου εύρους ζώνης.

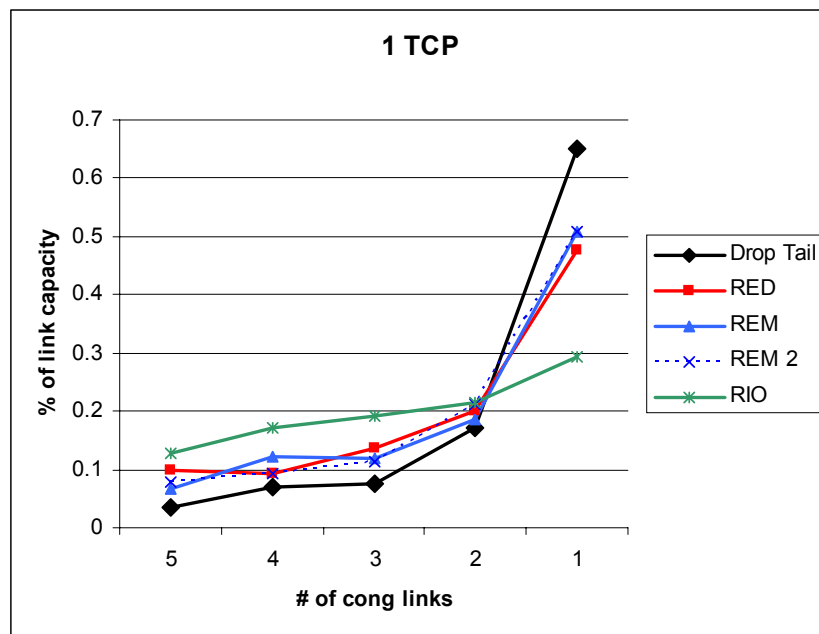
Ποσοστό της διαθέσιμης χωρητικότητας που καταλαμβάνουν οι σύνδεσης κάθε κόμβου συναρτήσει του αριθμού των ενδιάμεσων αργών ζεύξεων με ουρά Drop Tail ανάλογα με το είδος των συνδέσεων.



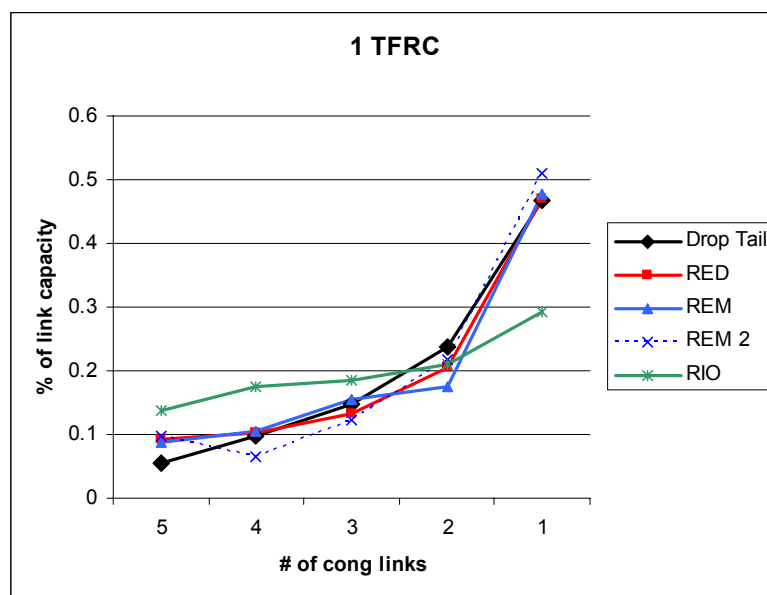
- Όταν στους δρομολογητές γίνεται ενεργή διαχείριση των ουρών με τους αλγόριθμους RED ή REM (με οποιοδήποτε από τα δυο σει παραμέτρων) και έχουμε κίνηση αποκλειστικά TCP οι ανισότητες στην κατανομή της διαθέσιμης χωρητικότητας αμβλύνονται, αλλά δεν εξαλείφονται.
- Όταν υπάρχει αποκλειστικά TFRC κίνηση ή συνδυασμός TCP και TFRC κίνησης, δεν παρατηρούνται σημαντικές διαφορές μεταξύ Drop Tail και RED ή REM, αφού οι διαφορές είναι ήδη μικρότερες από την περίπτωση που η κίνηση είναι μόνο TCP.

- Με τον συνδυασμό ενεργής διαχείρισης ουρών και per-flow scheduling που προσφέρει ο αλγόριθμος RIO μπορεί να επιτευχθεί δίκαια κατανομή του διαθέσιμου εύρους ζώνης, που σύμφωνα με την θεώρηση των ροών σε σχέση με τους πόρους δικτύου που απαιτούν προβλέπει γραμμικό διάγραμμα κατανομής της διαθέσιμης χωρητικότητας.

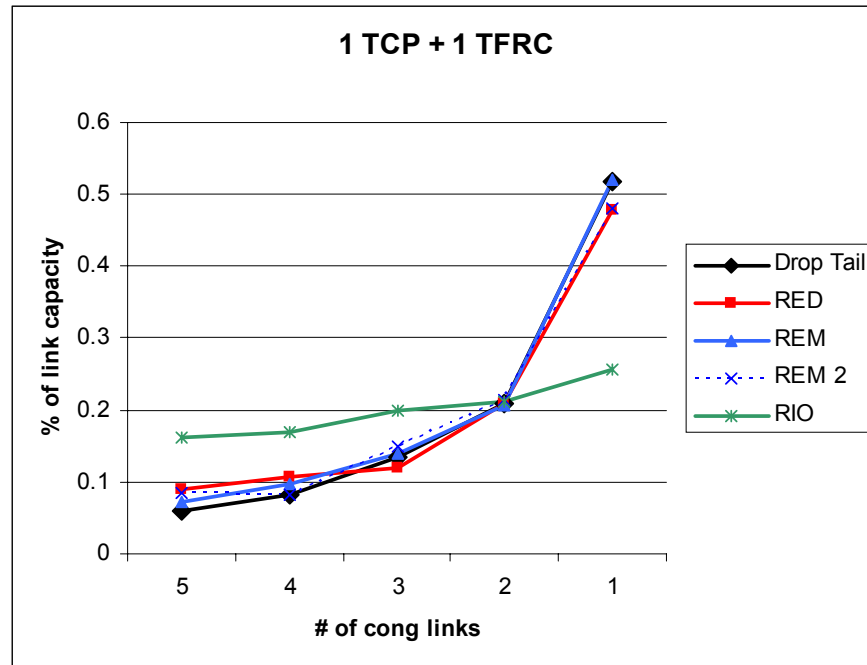
Ποσοστό της διαθέσιμης χωρητικότητας που καταλαμβάνουν οι σύνδεσης κάθε κόμβου συναρτήσει του αριθμού των ενδιάμεσων αργών ζεύξεων με κίνηση μόνο TCP ανάλογα με το είδος της ουράς.



Ποσοστό της διαθέσιμης χωρητικότητας που καταλαμβάνουν οι σύνδεσης κάθε κόμβου συναρτήσει του αριθμού των ενδιάμεσων αργών ζεύξεων με κίνηση μόνο TFRC ανάλογα με το είδος της ουράς.



Ποσοστό της διαθέσιμης χωρητικότητας που καταλαμβάνουν οι σύνδεσης κάθε κόμβου συναρτήσει του αριθμού των ενδιάμεσων αργών ζεύξεων με συνδυασμό κίνησης TCP και TFRC ανάλογα με το είδος της ουράς



5. Γενικά Συμπεράσματα

Από το σύνολο των προσομοιώσεων που πραγματοποιήθηκαν και τα αποτελέσματα που πήραμε από αυτές οδηγούμαστε στα εξής συμπεράσματα.

- Μη υπεύθυνες σε σήματα συμφόρησης συνδέσεις (π.χ. UDP ροές) μπορούν να μονοπωλήσουν τους πόρους του δικτύου και υπό συνθήκες να μειώσουν την απόδοση του.
- Ο αλγόριθμος έλεγχου ροής TFRC προσφέρει ικανοποιητικό επίπεδο δικαιοσύνης απέναντι σε TCP συνδέσεις, διατηρώντας παράλληλα σταθερότερο ρυθμό μετάδοσης από αυτές που είναι κατάλληλος για μετάδοση ροών πολυμέσων σταθερής ροής (multimedia streaming).
- Η χρήση αλγορίθμων ενεργής διαχείρισης ουρών στους δρομολογητές μπορεί να προσφέρει δίκαιη κατανομή των πόρων του δικτύου ανάμεσα στους χρήστες σε συνδυασμό με υψηλή χρησιμοποίηση.
- Η μεροληψία κατά της κίνησης εκρηκτικής μορφής μπορεί να μειωθεί σε μεγάλο ποσοστό με την χρήση AQM.
- Φαινόμενα συγχρονισμού μεταξύ των ροών, που μπορούν να οδηγήσουν σε εναλλαγή περιόδων έντονης συμφόρησης, με υψηλά ποσοστά χαμένων πακέτων, και περιόδων χαμηλής κίνησης, όπου το δίκτυο υποχρησιμοποιείται, εξαλείφονται με την χρήση ενεργής διαχείρισης ουρών.
- Με τους αλγόριθμους αυτούς το μέσο μήκος ουράς των δρομολογητών παραμένει μέσα σε καθορισμένα όρια, με αποτέλεσμα το Queuing Delay και κατά συνέπεια το RTT των συνδέσεων να κρατείται σε χαμηλά επίπεδα.
- Επίσης το μήκος της ουράς παραμένει σχετικά σταθερό, επομένως η διασπορά της καθυστέρησης των πακέτων είναι μικρή, ευνοώντας εφαρμογές που απαιτούν σταθερή καθυστέρηση (π.χ. Voice over IP).
- Αποφεύγεται επίσης ο αποκλεισμός ροών που οδηγεί σε μείωση της συνολικής απόδοσης του δικτύου.
- Ο συνδυασμός AQM με ανά ροή έλεγχο στους δρομολογητές μπορεί να επιτύχει δίκαιη κατανομή των πόρων ή, αν το επιθυμούμε, συγκεκριμένο τρόπο ανάθεσης εύρους ζώνης σε κάθε χρηστή.

- Με ανάθεση συγκεκριμένου εύρους ζώνης στις μη υπεύθυνες ροές μπορούμε να περιορίσουμε τους πόρους που καταλαμβάνουν και έτσι να εξασφαλίσουμε καλύτερη αντιμετώπιση για τις υπόλοιπες ροές.

Συνίσταται επομένως η χρήση του από άκρο σε άκρο ελέγχου συμφόρησης για όλα τα πρωτοκολλά μεταφοράς, ανάλογα με τις ανάγκες κάθε εφαρμογής. Τα διάφορα σχήματα έλεγχου συμφόρησης, όμως, θα πρέπει να είναι συμβατά με αυτό του TCP, ώστε να γίνεται δίκαιη κατανομή των πόρων του δικτύου ανάμεσα στις διαφορετικού τύπου ροές.

επίσης συνίσταται η υιοθέτηση κάποιου σχήματος ενεργής διαχείρισης ουρών στους δρομολογητές, αφού με την χρήση τους επιτυγχάνεται αυξημένη δικαιοσύνη και υψηλότερη ποιότητα υπηρεσίας σε σχέση με τις απλές Drop Tail ουρές, χωρίς σημαντική επιβάρυνση σε υπολογιστικό φόρτο και απαίτηση μνήμης.

Είναι επιπλέον επιθυμητό η ενεργή διαχείριση των ουρών να συνδυάζεται με ανά ροή έλεγχο των πακέτων που φτάνουν στον δρομολογητή. Η σχεδίαση αυτή μπορεί να προσφέρει ακόμη δικαιότερη κατανομή των πόρων, αλλά και περιορισμό των μη συμβατών με το TCP ροών. Επίσης μπορούν, αν το θελήσουμε, να προσφέρουν κατανομή των πόρων του δικτύου ανάλογα με την επιθυμητή ποιότητα υπηρεσίας κάθε χρηστή και την τιμή που προτίθεται να πληρώσει για αυτή.

Παράρτημα : Βιβλιογραφία

- [1] Andrew S. Tanenbaum, "Δίκτυα Υπολογιστών", μετάφραση Β. Στυλιανάκης, επιμέλεια Ε. Συκάς και Μ. Θεολόγου, Παπασωτηρίου 2000
- [2] Stevens, W., "TCP/IP Illustrated, Volume 1: The Protocols", Addison-Wesley, 1994
- [3] Douglas E. Comer, "Internetworking with TCP/IP" Prentice Hall, 2000
- [4] Braden, R., Clark, D., Crowcroft, J., Davie, B., Deering, S., Estrin, D., Floyd, S., Jacobson, V., Minshall, G., Partridge, C., Peterson, L., Ramakrishnan, K.K., Shenker, S., Wroclawski, J., and L. Zhang, "Recommendations on Queue Management and Congestion Avoidance in the Internet", RFC 2309, April 1998
- [5] Floyd, S., "Congestion Control Principles", BCP 41, RFC 2914, September 2000
- [6] Stevens, W., Allman, M. and V. Paxson, "TCP Congestion Control", RFC 2581, April 1999
- [7] Floyd, S. and T. Henderson, "The NewReno Modification to TCP's Fast Recovery Algorithm", RFC 2582, April 1999
- [8] Fall, K. and S. Floyd, "Simulation-based Comparisons of Tahoe, Reno and SACK TCP", Computer Communication Review, July 1996.
<ftp://ftp.ee.lbl.gov/papers/sacks.ps.Z>
- [9] Handley, M., Floyd, S., Padhye, J. and J. Widmer, "TCP Friendly Rate Control (TFRC): Protocol Specification", RFC 3448, January 2003
- [10] J. Padhye, V. Firoiu, D. Towsley, J. Kurose, "Modeling TCP Throughput: A Simple Model and its Empirical Validation", Tech Report TF 98-008, U. Mass, February 1998
- [11] S. Floyd, M. Handley, J. Padhye, J. Widmer, "Equation- Based Congestion Control for Unicast Applications", ACM SIGCOMM 2000
- [12] S. Floyd, M. Handley, J. Padhye, "A comparison of Equation-Based and AIMD Congestion Control", ACIRI May 2000

- [13] S. Floyd and V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance",
IEEE/ACM Trans. on Networking, vol. 1 no 4 pp 397-413 August 1993
- [14] Sanjeeva Athuraliya, Victor H. Li, Steven H. Low and Qinghe Yin,
"REM: Active Queue Management", October 2000
- [15] Qinghe Yin and Steven H. Low, "Convergence of REM Flow Control at a Single Link", August 2000
- [16] Clark D. and Fang W. "Explicit Allocation of Best-Effort Packet Delivery Service",
IEEE/ACM Trans. on Networking, vol. 6 no 4 pp 362-373 August 1998
- [17] Floyd S., "Connections with Multiple Congested Gateways in Packet-Switched Networks Part 1: One-way Traffic". Computer Communications Review, Vol.21, No.5, October 1991, pp. 30-47.
URL <http://ftp.ee.lbl.gov/floyd/>
- [18] The ns-2 network simulator web page
<http://www.isi.edu/nsnam/ns/>