

ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΚΡΗΤΗΣ

**Σχολή Τεχνολογικών Εφαρμογών
Τμήμα Εφαρμοσμένης Πληροφορικής & Πολυμέσων**



Πτυχιακή Εργασία

**“ Συγκριτική μελέτη αλγόριθμων ελέγχου συμφόρησης
πρωτοκόλλου TCP.”**

ΟΝΟΜΑΤΕΠΩΝΥΜΟ: ΜΠΟΜΠΟΚΗΣ ΠΑΝΑΓΙΩΤΗΣ

ΗΜΕΡΟΜΗΝΙΑ: 22/04/2008

ΕΙΣΗΓΗΤΗΣ: ΜΑΡΚΑΚΗΣ ΕΥΑΓΓΕΛΟΣ

Σε αυτούς που στάθηκαν δίπλα μου,
με ιδιαίτερη εκτίμηση και αγάπη

Ευχαριστίες

Με την ολοκλήρωση της πτυχιακής μου εργασίας, η οποία υλοποιήθηκε με τη συνεργασία του Εργαστήριο Τηλεπικοινωνιών και Δικτύων (ΠΑΣΙΦΑΗ) του Α.Τ.Ε.Ι Κρήτης, θα ήθελα να ευχαριστήσω τους ανθρώπους οι οποίοι βοήθησαν στην περάτωση αυτής της εργασίας. Θα ήταν παράλειψη να μην αναφερθώ σε όλους εκείνους που συμπαραστάθηκαν σε αυτήν την προσπάθεια.

Κατά κύριο λόγο, θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου κ. Μαρκάκη Ευάγγελο και τον υποψήφιο διδάκτορα από το Α.Τ.Ε.Ι Κρήτης κ. Μαστοράκη Γεώργιο, οι οποίοι με υποστήριξαν καθ'όλη τη διάρκεια της πτυχιακής εργασίας και μου εξασφάλισαν την παροχή πλούσιας υλικοτεχνικής υποδομής, πολύτιμης για μια μελέτη όπως αυτή και για την καθοδήγηση του προς την επιτυχή ολοκλήρωση της πτυχιακής μου εργασίας.

Ηράκλειο, Απρίλιος 2008

Μπομπόκης Παναγιώτης

Περίληψη

Η διπλωματική αυτή εργασία αναφέρεται σε μία μελέτη η οποία θα πραγματοποιηθεί για τη σύγκριση των αλγόριθμων ελέγχου συμφόρησης του πρωτοκόλλου TCP (Transport Control Protocol) που έχουν προτυποποιηθεί μέχρι σήμερα. Θα μελετηθούν όλες οι ερευνητικές εργαστηριακές μετρήσεις που έχουν δημοσιευθεί σε έγκυρα επιστημονικά περιοδικά και θα συγκριθούν με εργαστηριακές μετρήσεις οι οποίες έχουν πραγματοποιηθεί στον τομέα αυτό στο Εργαστήριο Έρευνας και Ανάπτυξης Τηλεπικοινωνιακών Συστημάτων, “ΠΑΣΙΦΑΗ”. Οι μετρήσεις θα επικεντρωθούν σε αντικειμενικά κριτήρια αξιολόγησης των επιδόσεων δικτύων όπως: throughput, RTT, sequence offset, jitter, one-way delay, delay variation, packet losses, retransmissions.

Στόχος της διπλωματικής αυτής εργασίας αποτελεί η εξέταση των κατάλληλων αρχιτεκτονικών για τη δημιουργία δικτύων με τις μέγιστες αποδόσεις και τις λιγότερες απώλειες ανάλογα με την υλοποίηση που θέλουμε να κάνουμε, οι οποίες υλοποιούν τη ροή μεταφοράς δεδομένων κάνοντας χρήση διαφόρων αλγόριθμων ελέγχου συμφόρησης του πρωτοκόλλου TCP (όπως tcp Tahoe, tcp veno, tcp new Reno, tcp Vegas, tcp Westwood/+, tcp hybla, high speed tcp, scable tcp). Η αξιολόγηση της απόδοσης του δικτύου θα γίνει με αντικειμενικά μέτρα όπως (throughput, RTT, sequence offset, jitter, one-way delay, delay variation, packet losses, retransmissions.) κάτω από διαφορετικές συνθήκες φόρτισης του δικτύου.

ΠΕΡΙΕΧΟΜΕΝΑ

<u>1. ΕΙΣΑΓΩΓΗ</u>	10
<u>1.1. Γενική Περιγραφή</u>	10
<u>1.2. Στόχοι</u>	10
<u>1.3. Δομή</u>	10
<u>2.Γενική περιγραφή αλγορίθμων ελέγχου συμφόρησης πρωτοκόλλου TCP</u>	10
<u>2.1. Εισαγωγή</u>	11
<u>2.2. TCP Extensions</u>	11
<u>2.2.1 TCP Tahoe</u>	11
<u>2.2.2 TCP Reno</u>	11
<u>2.2.3 TCP NewReno</u>	16
<u>2.2.4 TCP Vegas</u>	18
<u>2.2.5 TCP Veno</u>	22
<u>2.2.6 TCP Westwood/+</u>	22
<u>2.2.7TCP Hybla</u>	27
<u>3. Συγκριτική μελέτη ως προς την απόδοση αλγορίθμων ελέγχου συμφόρησης TCP</u>	30
<u>3.1. Εισαγωγή</u>	30
<u>3.2. Η Απόδοση των TCP Tahoe, του Reno, και του SACK σε Διάφορα Σενάρια</u>	30
<u>3.2.1 Γενική Περιγραφή</u>	30
<u>3.2.2 ΠΡΟΣΟΜΟΙΩΣΗ</u>	33
<u>3.3.Σύγκριση Δυναμικής του TCP Veno και Reno</u>	41
<u>3.3.1 Γενική Περιγραφή</u>	41
<u>3.3.2. Ο ΜΗΧΑΝΙΣΜΟΣ ΤΟΥ TCP VENO</u>	43
<u>3.3.3. ΔΙΚΤΥΟ ΠΕΙΡΑΜΑΤΟΣ</u>	46
<u>3.3.4. ΠΕΙΡΑΜΑΤΙΚΑ ΑΠΟΤΕΛΕΣΜΑΤΑ ΚΑΙ ΣΥΖΗΤΗΣΕΙΣ</u>	47
<u>3.4.Σύγκριση Απόδοσης Ανάμεσα σε TCP Reno καιTCP Vegas</u>	56
<u>3.4.1 Γενική Περιγραφή</u>	56
<u>3.4.2 Περιγραφή Reno και Vegas</u>	56
<u>3.4.3 Περιβάλλοντα προσομοίωσης</u>	60
<u>3.4.4 Αποτελέσματα Προσομοίωσης</u>	61
<u>3.4.4.1 Ομογενή περιβάλλοντα</u>	62
<u>3.4.4.2. Ετερογενή περιβάλλοντα</u>	68
<u>3.5 Ένα Τροποποιημένο Σχέδιο Αναμετάδοσης TCP-NewReno για Δίκτυο με</u>	72
<u>Απώλειες</u>	
<u>3.5.1 Γενική Περιγραφή</u>	72
<u>3.5.2. Η Αποτυχία του TCP-NewReno σε δίκτυα με υψηλό BER</u>	73
<u>3.5.3 Η Τροποποίησή μας σε TCP-NewReno</u>	75

<u>3.5.4 Αποτέλεσμα Προσομοίωσης και Συζήτηση</u>	75
<u>3.6 Ανταλλαγές Ενέργειας/Ρυθμοαπόδοσης(throughput) των Στρατηγικών Ελέγχου Λάθους του TCP(tcp error)</u>	79
<u>3.6.1 Γενική Περιγραφή</u>	79
<u>3.6.2 Επισκόπηση TCP</u>	83
<u>3.6.2.1 TCP Tahoe, Reno, και NewReno</u>	83
<u>3.6.3 Περιβάλλον Εξέτασης και Μεθοδολογία</u>	86
<u>3.6.4 Αποτελέσματα και Συζήτηση</u>	87
<u>3.6.4.1 Αποτελεσματική Απόδοση Ρυθμοαπόδοσης (Goodput)</u>	88
<u>3.6.4.2 Ενεργειακά Ζητήματα</u>	90
<u>3.7. ΑΠΩΛΕΙΑ ΠΑΚΕΤΟΥ ΚΑΙ ΚΑΤΑΣΤΑΣΗ ΣΥΜΦΟΡΗΣΗΣ ΣΤΟ TCP VENO</u>	96
<u>3.7.1 Γενική Περιγραφή</u>	96
<u>3.7.2. ΠΕΙΡΑΜΑΤΑ</u>	100
<u>3.7.2.1 Τοπολογία Προσομοίωσης</u>	100
<u>3.7.2.2 Η Απόδοση της Διάκρισης Απώλειας Πακέτου</u>	101
<u>3.8. Αύξηση του TCP Westwood+ σε Μεγάλης Ταχύτητας Ευρέα δίκτυα</u>	107
<u>3.8.1 Γενική Περιγραφή</u>	107
<u>3.8.2 ΣΤΡΑΤΗΓΙΚΕΣ ΕΞΕΛΙΞΗΣ ΠΑΡΑΘΥΡΟΥ ΣΥΜΦΟΡΗΣΗΣ</u>	110
<u>3.8.3 ΑΞΙΟΛΟΓΗΣΗ ΑΠΟΔΟΣΗΣ</u>	115
<u>4. Συμπεράσματα</u>	123
<u>4.1 Εισαγωγή</u>	123
<u>4.2 Συμπεράσματα ως προς την απόδοση των TCP Tahoe, του Reno, και του SACK σε Διάφορα Σενάρια</u>	123
<u>4.3 Συμπεράσματα δυναμικής σύγκρισης ανάμεσα σε TCP Reno και TCP Veno</u>	124
<u>4.4 Συμπεράσματα ως προς την απόδοση ανάμεσα σε TCP Reno και TCP Vegas</u>	125
<u>4.5 Συμπέρασμα για ένα τροποποιημένο σχέδιο αναμετάδοσης TCP-NewReno για δίκτυο με απώλειες</u>	125
<u>4.6 Συμπεράσματα ως προς την ανταλλαγή ενέργειας/ρυθμοαπόδοσης των στρατηγικών ελέγχου λάθους του TCP (tcp error)</u>	126
<u>4.7 Συμπεράσματα ως προς την απώλεια πακέτου και την κατάσταση συμφόρησης στο TCP VENO</u>	127
<u>4.8 Συμπεράσματα ως προς την αύξηση του tcp westwood+ σε μεγάλης ταχύτητας ευρέα δίκτυα</u>	128
<u>ΒΙΒΛΙΟΓΡΑΦΙΑ</u>	129

ΠΕΡΙΕΧΟΜΕΝΑ ΣΧΗΜΑΤΩΝ

<u>Σχήμα 1. Απόδοση διαφορετικών TCP εκδόσεων στην παρούσα συμφόρηση.</u>	25
<u>Σχήμα 2. Εξ ορισμού TCP (χωρίς γρήγορη αναμετάδοση και γρήγορη αποκατάσταση) παράθυρο συμπεριφοράς της συμφόρησης μετά από πτώση πακέτου</u>	34
<u>Σχήμα 3. Σύγκριση μιας συμφόρησης πτώσης παραθύρου.</u>	37
<u>Σχήμα 4. Σύγκριση του μεγέθους συμφόρησης παραθύρου για 2 πτώσεις.</u>	36
<u>Σχήμα 5. Σύγκριση χρόνου ανταπόκρισης για 2 πτώσεις.</u>	37
<u>Σχήμα 6. Σύγκριση του μεγέθους συμφόρησης παραθύρου για 4 πτώσεις.</u>	38
<u>Σχήμα 7. Σύγκριση της ανταπόκρισης για 4 πτώσεις.</u>	39
<u>Σχήμα 8. Ο αλγόριθμος ρύθμισης του ssthresh του TCP Veno</u>	43
<u>Σχήμα 9. Ο αλγόριθμος προσθετικής αύξησης του βελτιωμένου TCP Veno</u>	43
<u>Σχήμα 10. Πείραμα τοπολογίας δικτύου</u>	45
<u>Σχήμα 11. (α) Εξέλιξη παραθύρου σε δύο συνδέσεις Reno. (β) Εξέλιξη παραθύρου σε δύο συνδέσεις Veno</u>	47
<u>Σχήμα 12. Εξέλιξη παραθύρου σε Reno και Veno</u>	49
<u>Σχήμα 13. (α) Εξέλιξη παραθύρου σε συνδέσεις Reno και Veno. (β) Αύξων αριθμός τεσσάρων συνδέσεων</u>	51
<u>Σχήμα 14. Μια σύνδεση Reno ξεκίνησε πρώτη, μια άλλη Reno συνδέεται στα 20sec.</u>	53
<u>Σχήμα 15. Μια σύνδεση Veno ξεκίνησε πρώτη, μια άλλη Reno συνδέεται στα 20sec.</u>	54
<u>Σχήμα 16. Μια σύνδεση Reno ξεκίνησε πρώτη, μια άλλη Veno συνδέεται στα 20sec</u>	54
<u>Σχήμα 17. Τοπολογία δικτύου.</u>	60
<u>Σχήμα 18. Οι cwnd δυναμικές των δυο αντιπροσωπευτικών TCP συνδέσεων (μέγεθος buffer = 5)</u>	63
<u>Σχήμα 19. Οι cwnd δυναμικές των πέντε TCP Vegas συνδέσεων (μέγεθος buffer = 12)</u>	64
<u>Σχήμα 20. Οι cwnd δυναμικές των δυο αντιπροσωπευτικών TCP Reno συνδέσεων bufer500</u>	65

<u>Σχήμα 21. Διαδικασία Διαφορών του TCP – New Reno και των τροποποιήσεών μας.</u>	75
<u>Σχήμα 22. Σενάριο Δικτύου</u>	76
<u>Σχήμα 23 Ρυθμός αναμεταδιδόμενων πακέτων</u>	77
<u>Σχήματα 24,25,26 η επίπτωση από την παράδοση του εκτός λειτουργίας πακέτου είναι κυρίως σε χαμηλά BER</u>	78
<u>Σχήμα 27: Αποτελεσματική Ρυθμοαπόδοση (Goodput) με μέση On/Off φάση 1 second.</u>	87
<u>Σχήμα 28: Αποτελεσματική Ρυθμοαπόδοση (Goodput) με μέση On/Off φάση 10 seconds.</u>	88
<u>Σχήμα 29: Overhead με μέση On/Off φάση 1 second.</u>	90
<u>Σχήμα 30: Overhead με μέση On/Off φάση 10 second.</u>	91
<u>Σχήμα 31: Χρόνος Σύνδεσης με μέση On/Off φάση 10 seconds.</u>	92
<u>Σχήμα 32. Δικτυακό Μοντέλο</u>	99
<u>Σχήμα 33. Παραθύρο συμφόρησης Veno/Reno έναντι της επιβράδυνσης του μήκους της σειράς αναμονής (χωρίς τυχαία απώλεια)</u>	104
<u>Σχήμα 34. Ρυθμοαπόδοση των Veno και Reno έναντι του μέσου όρου ρυθμοαπόδοσης των δυο συνδέσεων Reno.</u>	105
<u>Σχήμα 35: Τάσεις της αύξησης του παραθύρου για διάφορες υπονήφιες ενέργειες.</u>	114
<u>Σχήμα 36: Το προτεινόμενο σενάριο προσομείωσης.</u>	115
<u>Σχήμα 37: Αύξηση παραθύρου των NewReno, Westwood+ και LogWestwood+</u>	116
<u>Σχήμα 38: Αριθμός των αναγνωριζόμενων πακέτων έναντι του χρόνου προσομοίωσης των NewReno, Westwood+ και LogWestwood+</u>	117
<u>Σχήμα 39 Σύγκριση για διάφορες καταστάσεις</u>	119
<u>Σχήμα 40: Σύγκριση εκτέλεσης ρυθμοαπόδοσης με την παρουσία τυχαίων λαθών.</u>	120
<u>Σχήμα 41 Σύγκριση απόδοσης για την περίπτωση πολλαπλών ταυτόχρονων ροών</u>	121

<u>Σχήμα 42 Σύγκριση απόδοσης για την περίπτωση πολλαπλών ταυτόχρονων ροών</u>	122
<u>Πίνακας 1. ΑΝΑΠΤΥΞΗ ΤΟΥ TCP</u>	31
<u>Πίνακας 2. Παράμετροι και αξίες που χρησιμοποιήθηκαν στην προσομοίωση.</u>	59
<u>Πίνακας 3. Μέσος αριθμός των ACKs και των διαφορών τους για 5 TCP συνδέσεις με διαφορετικά μεγέθη buffer στα routers (50sec.)</u>	62
<u>Πίνακας 4. Μέσος αριθμός των ACKs και των διαφορών τους για διαφορετικό αριθμό συνδέσεων TCP (μέγεθος buffer = 50, 50sec.)</u>	65
<u>Πίνακας 5. Μέσος αριθμός των ACKs και των διαφορών τους για διαφορετικό αριθμό συνδέσεων σε κάθε ομάδα (μέγεθος buffer = 50)</u>	67
<u>Πίνακας 6. Μέσος αριθμός των ACKs και των διαφορών τους για 5 Reno και 5 Vegas συνδέσεις με διαφορετικό μέγεθος buffer στα routers (50sec.)</u>	68
<u>Πίνακας 7. Μέσος αριθμός των ACKs και των διαφορών τους για διαφορετικό αριθμό συνδέσεων Vegas με 10 συνδέσεις TCP (μέγεθος buffer = 50, 50sec.)</u>	69
<u>Πίνακας 8. Μέσος αριθμός των ACKs και των διαφορών τους με αριθμό συνδέσεων σε κάθε ομάδα (μέγεθος buffer = 50</u>	71
<u>Πίνακας 9. Αποτελέσματα του test με μέση διάρκεια φάσης On/Off 1 second.</u>	94
<u>Πίνακας 10. Αποτελέσματα του test με μέση διάρκεια φάσης On/Off 10 seconds.</u>	95

1.Εισαγωγή

1.1. Γενική Περιγραφή

Το θέμα με το οποίο ασχολείται η συγκεκριμένη πτυχιακή εργασία αφορά τη μελέτη, τη σύγκριση και την αξιολόγηση των αλγόριθμων ελέγχου συμφόρησης του πρωτοκόλλου TCP (Transport Control Protocol) που έχουν προτυποποιηθεί μέχρι σήμερα . Θα μελετηθούν όλες οι ερευνητικές εργαστηριακές μετρήσεις που έχουν δημοσιευθεί σε έγκριτα επιστημονικά περιοδικά και θα συγκριθούν με εργαστηριακές μετρήσεις οι οποίες έχουν πραγματοποιηθεί στον τομέα αυτό στο Εργαστήριο Έρευνας και Ανάπτυξης Τηλεπικοινωνιακών Συστημάτων, “ΠΑΣΙΦΑΗ”. Οι μετρήσεις θα επικεντρωθούν σε αντικειμενικά κριτήρια αξιολόγησης των επιδόσεων δικτύων με βάση τα εξής χαρακτηριστικά : throughput, RTT, sequence offset, jitter, one-way delay, delay variation, packet losses, retransmissions.

1.2. Στόχοι

Ο βασικός στόχος της πτυχιακής εργασίας αποτελεί η εξέταση των κατάλληλων αρχιτεκτονικών για τη δημιουργία δικτύων με τις μέγιστες αποδόσεις και τις λιγότερες απώλειες κάνοντας χρήση διάφορων αλγόριθμων ελέγχου συμφόρησης του πρωτοκόλλου TCP ανάλογα με την υλοποίηση που θέλουμε να πραγματοποιήσουμε. Έτσι μέσα από αυτό θα βγάλουμε χρήσιμα συμπεράσματα για το ποιος αλγόριθμος είναι καλύτερος ανάλογα με την χρήση που θέλουμε να κάνουμε.

1.3. Δομή

Σε μια προσπάθεια να καλυφθούν ικανοποιητικά τα παραπάνω θέματα, επιχειρείται να εξετασθούν όλες οι παράμετροι που καθορίζουν τη λειτουργία ενός τέτοιου δικτύου με την πραγματοποίηση των κατάλληλων μετρήσεων έτσι ώστε να επιτευχθούν οι απαιτούμενες βελτιστοποιήσεις για την καλύτερη δυνατή απόδοση του συστήματος.

Στην ενότητα 2 παρουσιάζονται θεωρητικά στοιχεία σχετικά με τους αλγορίθμους ελέγχου συμφόρησης που θα χρησιμοποιήσουμε για την μελέτη μας .

Αντικείμενο της ενότητας 3 αποτελεί το κυρίως ήμερος της πτυχιακής εργασίας και έχει να κάνει με πειραματικές μελέτες που έχουν γίνει και βγαίνουν χρήσιμα συμπεράσματα για το ποιος αλγόριθμος είναι ορθότερος να χρησιμοποιήσουμε ανάλογα με την χρήση που θέλουμε να κάνουμε.

Η ολοκλήρωση της εργασίας αυτής πραγματοποιείται με την ενότητα 4 καταγράφοντας τα συμπεράσματα που προκύπτουν από τη συγκριτική μελέτη αλγορίθμων ελέγχου συμφόρησης πρωτοκόλλου TCP .

2.Γενική περιγραφή αλγορίθμων ελέγχου συμφόρησης πρωτοκόλλου TCP

2.1 Εισαγωγή

Σ' αυτό το κεφαλαίο κάνουμε μια γενική περιγραφή για το πώς λειτουργεί ο κάθε αλγόριθμος ελέγχου συμφόρησης TCP, ποια είναι τα κύρια χαρακτηριστικά του τα οποία θα μας είναι πολύ χρήσιμα στο 3 κεφάλαιο όπου πραγματοποιούνται και οι συγκριτικές μελέτες.

2.2 TCP Extensions

2.2.1 TCP Tahoe

Οι πρώτες εφαρμογές TCP ακολούθησαν μια τεχνική την go-back-n χρησιμοποιώντας τη συσσωρευτική θετική αναγνώριση, και απαίτησαν έναν αναμεταδότη λήξης χρονομέτρων για να στέλνει εκ νέου τα στοιχεία που χάνονται κατά τη διάρκεια της πτήσης. Αυτά τα TCPs έκαναν πολύ λίγα για να χειριστούν τη συμφόρηση. Το TCP Tahoe πρόσθεσε διάφορους νέους αλγορίθμους και καθαρισμούς στις προηγούμενες εφαρμογές. Οι νέοι αλγόριθμοι περιλαμβάνουν αργή-έναρξη, αποφυγή συμφόρησης, και γρήγορη-αναμετάδοση. Ένας από τους σημαντικότερους καθαρισμούς ήταν η τροποποίηση του roundtrip

time estimator που χρησιμοποιήθηκε για να θέσει τις retransmission timeout values. Αρχικά, υποτίθετο ότι τα χαμένα πακέτα αντιπροσώπευαν τη συμφόρηση. Επομένως, είχε υποτεθεί από τον Jacobson ότι όταν θα εμφανιζόταν μια απώλεια πακέτων, ο αποστολέας θα έπρεπε να χαμηλώσει το μερίδιό του εύρους ζώνης. Το TCP Tahoe δεν χειρίζεται καλά τις πολλαπλές πτώσεις πακέτων μέσα από ένα ενιαίο παράθυρο στοιχείων. Οι δύο φάσεις στην αύξηση του παραθύρου συμφόρησης, η αργή-έναρξη και οι φάσεις αποφυγής συμφόρησης μπορούν να συνομισθούν με τις ακόλουθες εξισώσεις:

- Φάση αργή-έναρξη:

$$cwnd = cwnd + 1 \quad \text{εάν } cwnd < ssthresh$$

- Φάση αποφυγής συμφόρησης:

$$cwnd = cwnd + 1/cwnd \quad \text{εάν } cwnd \geq ssthresh$$

όπου $ssthresh$ είναι η αξία κατώτατων ορίων στην οποία το TCP αλλάζει τη φάση του από την αργή-έναρξη στην αποφυγή συμφόρησης. Όταν μια απώλεια τμήματος ανιχνεύεται, το $cwnd$ και $ssthresh$ ενημερώνεται ως εξής:

- $ssthresh = cwnd/2$
- $cwnd = 1$

Κατά τη διάρκεια του χρόνου, όταν εμφανίστηκε το TCP Tahoe, το περιβάλλον δικτύων και οι εφαρμογές που χρησιμοποιούνταν δεν απαιτήσαν τις υψηλές συνδέσεις εύρους ζώνης. Ως εκ τούτου, αυτή η παραλλαγή του TCP δεν έπρεπε να αντιμετωπίσει την πρόκληση της διαβάθμισης του υψηλού bandwidth προϊόντων καθυστέρησης δικτύου. Οι μελέτες απεικονίζουν ότι το TCP Tahoe έχει σημαντικά μειονεκτήματα ως μέσο παροχής των υπηρεσιών στοιχείων για ένα δίκτυο πολυμέσων, δεδομένου ότι η τυχαία απώλεια ως αποτέλεσμα των διακυμάνσεων σε πραγματικό χρόνο στη κυκλοφορία μπορεί να οδηγήσει στη σημαντική επιδείνωση ρυθμοαπόδοσης του υψηλού bandwidth προϊόντων καθυστέρησης δικτύου. Τα αποτελέσματα αυτών των μελετών καταλήγουν στο συμπέρασμα ότι η απόδοση

υποβιβάζεται όταν το προϊόν της πιθανότητας απώλειας και του τετραγώνου του υψηλού bandwidth προϊόντων καθυστέρησης είναι μεγάλο. Επίσης, για το υψηλό bandwidth προϊόντων καθυστέρησης δικτύου, το TCP είναι εξαιρετικά άδικο προς τις συνδέσεις με τις υψηλότερες καθυστερήσεις διάδοσης.

2.2.2 TCP Reno

Αυτό είναι το κλασικό πρότυπο που χρησιμοποιείται για τον έλεγχο συμφόρησης. Εκθέτει τη χαρακτηριστική αργή έναρξη των μεταδόσεων. Η ρυθμοαπόδοση αυξάνεται βαθμιαία έως ότου μείνει σταθερή. Μειώνεται μόλις η μεταφορά αντιμετωπίζει τη συμφόρηση, κατόπιν το ποσοστό ανεβαίνει πάλι αργά. Το παράθυρο αυξάνεται με την προσθήκη των σταθερών τιμών. Το TCP Reno χρησιμοποιεί έναν πολλαπλασιαστικό αλγόριθμο μείωσης για τη μείωση του μεγέθους των παραθύρων. Το TCP Reno είναι ο ευρύτερα ανεπτυγμένος αλγόριθμος.

Η εφαρμογή TCP Reno τροποποίησε τον αποστολέα για να ενσωματώσει έναν μηχανισμό αποκαλούμενο fast recovery. Αντίθετα από το Tahoe, το Reno δεν εκκενώνει το σωλήνα, όχι απαραίτητα στην παραλαβή μερικών αριθμών dupacks. Αντ' αυτού, με το μηχανισμό της γρήγορης αποκατάστασης το παράθυρο συμφόρησης τίθεται στη μισή προηγούμενη αξία του. Η ιδέα είναι ότι ο μόνος τρόπος για μια απώλεια να ανιχνευτεί μέσω ενός διαλείμματος και όχι μέσω της παραλαβής ενός dupack είναι όταν η ροή των πακέτων και ACKs έχει σταματήσει εντελώς, το οποίο θα ήταν μια ένδειξη της βαριάς συμφόρησης. Αλλά εάν ο αποστολέας είναι ακόμα ικανός να λάβει ένα ACK, τότε δεν πρέπει να πέσει πίσω στην αργή-έναρξη, όπως κάνει στην περίπτωση του TCP Tahoe. Αυτή η περίπτωση δεν υποδηλώνει βαριά συμφόρηση, δεδομένου ότι η ροή υπάρχει ακόμα, αλλά ο αποστολέας πρέπει να στείλει σχετικά λιγότερο δυναμικά, χρησιμοποιώντας ένα χαμηλότερο ποσό πόρων. Ο μηχανισμός της γρήγορης αποκατάστασης μπαίνει στην εικόνα σε αυτή τη φάση. Αφού ληφθούν ορισμένα dupacks, ο αποστολέας θα αναμεταδώσει το χαμένο πακέτο αλλά, αντίθετα από το Tahoe, δεν θα πέσει

πίσω στο slow-start. Θα εκμεταλλευτεί μάλλον το γεγονός ότι αυτήν την περίοδο η υπάρχουσα ροή πρέπει να συνεχίσει την αποστολή, αν και χρησιμοποιώντας λιγότερους πόρους. Με τη χρησιμοποίηση της γρήγορης αποκατάστασης ο αποστολέας χρησιμοποιεί ένα παράθυρο συμφόρησης που είναι το μισό μέγεθος του παρόντος παραθύρου συμφόρησης αμέσως πριν από την απώλεια. Αυτός ο παράγοντας αναγκάζει το Reno να στείλει λιγότερα πακέτα έξω έως ότου ξέρει ότι είναι εφικτό να σταλούν περισσότερα. Επομένως, έχει μειώσει πράγματι τη χρησιμοποίηση του δικτύου. Αν και το TCP Reno είναι καλύτερο από το Tahoe σε περιπτώσεις ενιαίας απώλειας πακέτων, το TCP Reno δεν είναι πολύ καλύτερο από το Tahoe όταν χάνονται πολλαπλάσια πακέτα μέσα σε ένα παράθυρο στοιχείων (window data). Η γρήγορη αποκατάσταση εξασφαλίζει ότι ο σωλήνας δεν γίνεται κενός. Επομένως, η αργή-έναρξη εκτελείται μόνο όταν ένα πακέτο έχει προσωρινά διακοπεί. Αυτό εφαρμόζεται με τη ρύθμιση ssthresh στο μισό μέγεθος του τρέχοντος παραθύρου συμφόρησης και έπειτα τη ρύθμιση του παραθύρου συμφόρησης σε 1 τμήμα, προκαλώντας την σύνδεση TCP σε αργή-έναρξη έως ότου επιτυγχάνεται το ssthresh, κατόπιν πηγαίνει στη φάση αποφυγής συμφόρησης όπως στην περίπτωση Tahoe. Το TCP Reno που αντιπροσωπεύεται με μορφή εξίσωσης μοιάζει με αυτό:

- Slow-start phase

$$cwnd = cwnd + 1$$

Όταν ένα τμήμα ανιχνεύεται, ο γρήγορος αλγόριθμος αναμετάδοσης μοιράζει το παράθυρο συμφόρησης.

- $ssthresh = cwnd / 2$
- $cwnd = ssthresh$

Το TCP Reno εισάγει έπειτα τη γρήγορη φάση αποκατάστασης. Σε αυτήν την φάση, το μέγεθος των παραθύρων αυξάνεται από το ένα τμήμα όταν παραλαμβάνεται μια διπλή αναγνώριση και το παράθυρο συμφόρησης

αποκαθίσταται στο ssthresh όταν παραλαμβάνεται μια μη-διπλή αναγνώριση που αντιστοιχεί στα τμήματα. Το βασικό πρόβλημα στο TCP Reno είναι ότι η γρήγορη αναμετάδοση υποθέτει ότι μόνο ένα τμήμα χάθηκε. Αυτό μπορεί να οδηγήσει στην απώλεια χρονομέτρησης και διαλειμμάτων ACK εάν περισσότερα από ένα τμήματα χαθούν. Το Reno αντιμετωπίζει διάφορα προβλήματα όταν εμφανίζονται πολλαπλάσιες απώλειες πακέτων σε ένα παράθυρο στοιχείων. Αυτό εμφανίζεται συνήθως όταν επικαλείται γρήγορη αναμετάδοση και γρήγορη αποκατάσταση. Επικαλείται αρκετές φορές σε διαδοχική καθοδήγηση στις πολλαπλασιαστικές μειώσεις του cwnd και ssthresh επιδρώντας στη ρυθμοαπόδοση της σύνδεσης. Ένα άλλο πρόβλημα με το TCP Reno είναι η ACK λιμοκτονία. Αυτό εμφανίζεται λόγω της ασάφειας των διπλών ACKs. Ο αποστολέας μειώνει το παράθυρο συμφόρησης όταν εισάγει γρήγορη αναμετάδοση, λαμβάνει dupacks που διογκώνουν το παράθυρο συμφόρησης έτσι ώστε στέλνει τα νέα πακέτα έως ότου να γεμίσει το διαβιβασμένο παράθυρό του. Λαμβάνει έπειτα non-dupack και βγαίνει σε γρήγορη αποκατάσταση. Εντούτοις, λόγω των πολλαπλών απωλειών στο παρελθόν, το ACK θα ακολουθηθεί από 3 dupacks που επισημαίνουν ότι ένα άλλο τμήμα χάθηκε, αυτός ο τρόπος, γρήγορης αναμετάδοσης εισάγεται πάλι μετά από μια άλλη μείωση του ssthresh και cwnd. Αυτό συμβαίνει αρκετές φορές διαδοχικά και κατά τη διάρκεια αυτής της περιόδου η αριστερή άκρη του διαβιβασμένου παραθύρου προάγει μόνον μετά από κάθε επιτυχή γρήγορη αναμετάδοση, και ο αριθμός των στοιχείων στον αέρα σταδιακά γίνεται περισσότερος από το παράθυρο συμφόρησης. Όταν δεν υπάρχουν άλλα ACKs να παραληφθούν, ο αποστολέας πάει σκόπιμα και ανακτά από αυτό το αδιέξοδο μόνο μέσω της προσωρινής διακοπής, το οποίο προκαλεί την αργή-έναρξη. Υπάρχουν δύο λύσεις διαθέσιμες για τα ανωτέρω προβλήματα: ο Newreno και ο TCP SACK.

Πλεονεκτήματα

1. Απλό σχέδιο εκτίμησης bandwidth.
2. Ο επιθετικός μηχανισμός αποφυγής συμφόρησης εξασφαλίζει bandwidth όταν συνδέεται με τις συνδέσεις TCP Vegas.

3. Ευρύτερα επεκταμένος, πιθανώς λόγω της ωριμότητάς του και της επιθετικότητας του.

Μειονεκτήματα

1. Συνεχώς ενημερώνει το μέγεθος των παραθύρων
 - Μπορείτε να οδηγήσετε σε περιοδική ταλάντωση στο μέγεθος παραθύρων.
 - Μπορείτε να οδηγήσετε στην ταλάντωση σε RTT, προκαλώντας την καθυστέρηση jitter και την ανεπαρκή χρησιμοποίηση bandwidth.
 - Μπορείτε να έχετε πολλές αναμεταδόσεις των ίδιων πακέτων αφότου πέσει ένα πακέτο.
2. Οι συνδέσεις με κοντύτερο RTT μπορούν να ενημερώσουν τα μεγέθη παραθύρων συμφόρησης γρηγορότερα.
 - Τέτοιες συνδέσεις μπορούν έλαβαν ένα άδικο μερίδιο της ικανότητας δικτύων.
 - Το TCP Reno είναι προκατειλημμένο ενάντια στις συνδέσεις με τις πίο μακροχρόνιες καθυστερήσεις.

2.2.3 TCP NewReno

Το TCP Newreno τροποποιεί τη γρήγορη αναμετάδοση και τους γρήγορους μηχανισμούς αποκατάστασης του TCP Reno. Αυτές οι τροποποιήσεις εφαρμόζονται για να καθορίσουν τα μειονεκτήματα του TCP Reno. Εδώ, η αναμονή για το χρονόμετρο αναμετάδοσης αποβάλλεται όταν χάνονται τα

πολλαπλάσια πακέτα από ένα παράθυρο. Το Newreno είναι το ίδιο με Reno αλλά εφαρμόζει περισσότερη νοημοσύνη κατά τη διάρκεια της γρήγορης αποκατάστασης. Χρησιμοποιεί την ιδέα τμηματικών ACKs. Όταν υπάρχουν πολλαπλάσιες απώλειες πακέτων, το ACK για το αναμεταδοθέν πακέτο θα αναγνωρίσει μερικά αλλά όχι όλα τα πακέτα που εστάλησαν προτού την γρήγορη αναμετάδοση. Στο Newreno, ένα τμηματικό ACK λαμβάνεται ως ένδειξη ενός άλλου χαμένου πακέτου και υπό αυτήν τη μορφή ο αποστολέας διαβιβάζει το πρώτο μη αναγνωρισμένο πακέτο. Αντίθετα το Reno, τα τμηματικά ACKs δεν βγάζουν το Newreno από τη γρήγορη αποκατάσταση. Με αυτόν τον τρόπο το Newreno αναμεταδίδει 1 πακέτο ανά RTT έως ότου αναμεταδοθούν όλα τα χαμένα πακέτα, και αποφεύγει ζητώντας πολλαπλές γρήγορες αναμεταδόσεις από ένα ενιαίο παράθυρο στοιχείων. Αυτή η Newreno τροποποίηση του TCP Reno καθορίζει μια γρήγορη διαδικασία αποκατάστασης που αρχίζει όταν παραλαμβάνονται τρία διπλά ACKs και τελειώνει όταν είτε εμφανίζεται ένα διάλειμμα αναμετάδοσης ή φθάνει ένα ACK που αναγνωρίζει όλα τα στοιχεία μέχρι και του συμπεριλαμβανομένου στοιχείου που ήταν σημαντικό όταν άρχισε η γρήγορη διαδικασία αποκατάστασης. Ο αλγόριθμος Newreno μπορεί να εξηγηθεί στα ακόλουθα βήματα:

- Στην παραλαβή του τρίτου dupack, εάν ο αποστολέας δεν είναι ήδη στη γρήγορη διαδικασία αποκατάστασης, κατόπιν θέσει ssthresh λιγότερο από την αξία κατώτερου.

$$ssthresh = \max(\text{flightsize}/2, 2 * \text{MSS})$$

Επίσης, θυμηθείτε τον υψηλότερο αύξοντα αριθμό που διαβιβάζεται σε μια μεταβλητή.

- 1. Αναμεταδώστε το χαμένα πακέτο και το σύνολο cwnd στο ssthresh+3*MSS. Αυτό διογκώνει τεχνητά το παράθυρο συμφόρησης από τον αριθμό

τμημάτων που έχουν αφήσει το δίκτυο και που ο δέκτης είναι αποθηκευμένος.

- 2. Για κάθε πρόσθετα λαμβανόμενο dupack, αυξήστε το παράθυρο συμφόρησης από το MSS.

- 3. Διαβιβάστε ένα τμήμα, εάν επιτρέπεται από τη νέα αξία του cwnd και τους δέκτες του διαφημισμένου παραθύρου.

- 4. Όταν ένα ACK φθάνει να αναγνωρίζει νέα στοιχεία, αυτό το ACK θα μπορούσε να είναι η αναγνώριση αποσπασμένη από την αναμετάδοση από το βήμα 2, ή τη μια αποσπασμένη από μια πιο πρόσφατη αναμετάδοση.

2.2.4 TCP Vegas

Το TCP Vegas εισάγει τη μέτρηση RTT για την αξιολόγηση της ποιότητας συνδέσεων. Χρησιμοποιεί τις πρόσθετες αυξήσεις και τις πρόσθετες μειώσεις για το παράθυρο συμφόρησης. Το 1994, οι Brakmo, O'Malley και Peterson ήρθαν με μια νέα εφαρμογή TCP αποκαλούμενη Vegas που επιτυγχάνει μεταξύ 40% και 70% καλύτερη ρυθμοαπόδοση και του 1/5 έως του 1/2 των απωλειών σε σύγκριση με το TCP Reno. Το TCP Vegas είχε επίσης όλες τις αλλαγές και τις τροποποιήσεις από την πλευρά αποστολέων. Στο Reno, το RTT υπολογίζεται χρησιμοποιώντας ένα μη-λεπτομερές (coarse-grained) χρονόμετρο, το οποίο δεν δίνει μια ακριβή εκτίμηση του RTT. Οι δοκιμές που διευθύνονται καταλήγουν στο συμπέρασμα ότι για τις απώλειες που οδήγησαν σε ένα διάλειμμα, πήρε ο Reno κατά μέσο όρο 1100ms από το χρόνο που έστειλε ένα τμήμα που χάθηκε, μέχρι την προσωρινή παύση και ξαναστέλνει το τμήμα, όπου ότι λιγότερο από 300ms θα ήταν το σωστό διάστημα προσωρινής παύσης με την χρήση ενός ακριβέστερου ρολογιού. Το TCP Vegas καθορίζει αυτό το πρόβλημα χρησιμοποιώντας ένα λεπτότερο μη-λεπτομερές (coarse-grained) χρονόμετρο. Το Vegas άλλαξε επίσης το μηχανισμό αναμετάδοσης. Το ρολόι συστημάτων διαβάζεται και σώζεται κάθε φορά που στέλνεται ένα τμήμα, όταν ένα ACK φθάνει το ρολόι διαβάζεται πάλι και ο υπολογισμός RTT υπολογίζεται χρησιμοποιώντας αυτή τη φορά και το timestamp που καταγράφεται για το σχετικό τμήμα. Με τη χρήση αυτού του ακριβούς RTT, η αναμετάδοση αποφασίζεται ως εξής: Όταν

ένα dupack παραλαμβάνεται, το Vegas ελέγχει για να δει εάν το νέο RTT είναι μεγαλύτερο από το RTO. Εάν είναι, το Vegas αναμεταδίδει το τμήμα χωρίς να πρέπει να περιμένει το 3ο dupack. Εκτιμώντας ότι, όταν παραλαμβάνεται ένα non-dupack, εάν είναι το πρώτο ή το δεύτερο μετά από μια αναμετάδοση, το Vegas ελέγχει πάλι για να δει εάν $RTT > RTO$, εάν είναι έτσι τότε το τμήμα αναμεταδίδεται. Αυτή η διαδικασία πιάνει οποιοδήποτε άλλο τμήμα που μπορεί να είχε χαθεί προηγούμενα στην αναμετάδοση χωρίς απαίτηση μιας περιόδου αναμονής για να παραληφθεί ένα dupack. Το Vegas συμπεριφέρεται στην παραλαβή ορισμένων ACKs ως αφορμή για να ελέγξει εάν έπρεπε να γίνει ένα διάλειμμα, αλλά ακόμα περιέχει τον κώδικα διαλείμματος του Reno σε περίπτωση που αυτός ο μηχανισμός αποτύχει να αναγνωρίσει ένα χαμένο τμήμα. Οι ενέργειες αποφυγής συμφόρησης των Vegas είναι βασισμένες στις αλλαγές στο κατ' εκτίμηση ποσό πρόσθετων στοιχείων στο δίκτυο. Το Vegas καθορίζει το RTT μιας σύνδεσης ως το BaseRTT του όταν η σύνδεση δεν είναι κορεσμένη. Στην πράξη, είναι το ελάχιστο όλων των καταμετρημένων μετ' επιστροφής χρόνων και συνήθως είναι το RTT του πρώτου τμήματος που στέλνεται από τη σύνδεση, πριν από την αύξηση σειρών αναμονής δρομολογητών. Το Vegas χρησιμοποιεί αυτήν την αξία για να υπολογίσει την αναμενόμενη ρυθμοαπόδοση. Αφετέρου, υπολογίζει το τρέχον πραγματικό ποσοστό αποστολής. Αυτό γίνεται με την καταγραφή του χρόνου αποστολής για ένα τμήμα, καταγράφοντας πόσα bytes διαβιβάζονται μεταξύ του χρόνου ότι το τμήμα στέλνεται και της αναγνώρισης παραλαβής, υπολογίζοντας το RTT για το τμήμα όταν φθάνει η αναγνώρισή της, και διαιρώντας τον αριθμό bytes που διαβιβάζονται από το δείγμα RTT. Αυτός ο υπολογισμός γίνεται μία φορά για τον χρόνο κάθε κυκλικού ταξιδιού. Τρίτον το Vegas συγκρίνει την πραγματική με την αναμενόμενη ρυθμοαπόδοση και ρυθμίζει το παράθυρο αναλόγως. Η διαφορά μεταξύ της πραγματικής και της αναμενόμενης ρυθμοαπόδοσης καταγράφεται. Το Vegas καθορίζει δύο κατώτατα όρια, α και β , που αντιστοιχούν κατά προσέγγιση στην κατοχή επίσης λίγων και πάρα πολλών πρόσθετων στοιχείων στο δίκτυο, αντίστοιχα. Παρακάτω είναι ο μηχανισμός του ελέγχου συμφόρησης με μορφή εξίσωσης. Το Diff είναι η διαφορά μεταξύ της πραγματικής και αναμενόμενης ρυθμοαπόδοσης.

$Diff < 0$: αλλαγή BaseRTT στο πιο πρόσφατο επιλεγέν RTT

$Diff < \alpha$: αυξήστε το παράθυρο συμφόρησης γραμμικά

$Diff > \beta$: μειώστε το παράθυρο συμφόρησης γραμμικά

$\alpha < Diff < \beta$: μην κάνετε τίποτα

Για να είναι σε θέση να ανιχνεύσει και να αποφύγει τη συμφόρηση κατά τη διάρκεια της αργής-έναρξης, το Vegas επιτρέπει στην εκθετική αύξηση μόνο κάθε άλλο RTT. Στο μεταξύ, το παράθυρο συμφόρησης μένει σταθερό έτσι μια έγκυρη σύγκριση των αναμενόμενων και πραγματικών ποσοστών μπορεί να γίνει. Όταν το πραγματικό ποσοστό μειώνεται κάτω από το αναμενόμενο ποσοστό από το αντίτιμο ενός απομονωτή δρομολογητών, το Vegas αλλάζει από τον τρόπο slow-start στο γραμμικό τρόπο αύξησης/μείωσης. Μερικά προβλήματα με το TCP Vegas που θα μπορούσαν να ασκήσουν σοβαρή επίδραση στην απόδοσή του, είναι τα ζητήματα της μετακίνησης και της σταθερότητας. Η μετακίνηση μιας πορείας μπορεί να αλλάξει την καθυστέρηση διάδοσης της σύνδεσης. Το Vegas χρησιμοποιεί τη σύνδεση για να ρυθμίσει το μέγεθος παραθύρων και μπορεί να έχει επιπτώσεις στη ρυθμοαπόδοση αρκετά. Ένα άλλο ζήτημα του TCP Vegas είναι η σταθερότητά του. Δεδομένου ότι κάθε σύνδεση TCP προσπαθεί να κρατήσει μερικά πακέτα στο δίκτυο όταν η εκτίμησή τους για την καθυστέρηση διάδοσης είναι κλειστή, αυτό θα μπορούσε να οδηγήσει τη σύνδεση για να κρατήσει ακούσια τα πολλά περισσότερα πακέτα στο δίκτυο προκαλώντας μια επίμονη συμφόρηση. Η έρευνα για το TCP Vegas αποτελείται μέχρι σήμερα πρώτιστα από τις αναλύσεις του πρωτοκόλλου, που βελτιώνει τις τεχνικές της αποφυγής συμφόρησης και τις ανίχνευσης. Οι περισσότερες από τις μελέτες που περιλαμβάνουν το TCP Vegas αποτελούνται από την αξιολόγηση απόδοσής της με σεβασμό στο TCP Reno. Η πρόσφατη έρευνα στο Caltech ερευνά μια νέα έκδοση Vegas, της οποίας οι αξιώσεις για το Caltech είναι μια σταθεροποιημένη έκδοση του Vegas. Αυτή η

σταθεροποιημένη έκδοση Vegas είναι εντελώς μία σταθερή πηγή και δεν απαιτεί καμία υποστήριξη δικτύων. Περαιτέρω προτείνουν να επεκταθεί αυτό το σταθεροποιημένο Vegas σε μια αυξητική μέθοδο όταν ένα δίκτυο περιέχει ένα μίγμα συνδέσεων (μερικές με την ενεργό διαχείριση σειρών αναμονής και μερικές χωρίς). Επίσης, η απόδοση του TCP Vegas συγκρίνεται ενάντια σε αυτήν του TCP Reno στα πλέγματα υπολογισμού υψηλής απόδοσης από Eric Weidge και Wu-chon Feng του κρατικού πανεπιστημίου του Οχάιο. Με τη βοήθεια των πραγματικών διανομών κυκλοφορίας Weidge και Feng δείχνουν ότι το Vegas αποδίδει καλά πέρα από τις σύγχρονες συνδέσεις υψηλής απόδοσης και καλύτερα από το TCP Reno, υπό τον όρο ότι οι παράμετροι α και β του TCP Vegas επιλέγονται κατάλληλα.

Πλεονεκτήματα

1. Δίκαιο σχέδιο εκτίμησης εύρους ζώνης
 - Το ποσοστό αναπροσαρμογών των παραθύρων δεν εξαρτάται μόνο από το RTT όπως σε ένα TCP Reno
2. Ομαλό ποσοστό αποστολής και αποδοτική χρησιμοποίηση συνδέσεων όταν τα μεγέθη σειρών αναμονής είναι μεγάλα (το παράθυρο σταθεροποιεί μεταξύ του α και του β)
3. Το TCP Vegas ανιχνεύει τις απώλειες γρηγορότερα από το TCP Reno και μπορεί να καλύψει από τις πολλαπλάσιες πτώσεις αποτελεσματικότερα

Μειονεκτήματα

1. Δεν μπορείτε να ολοκληρώσετε με τις επιθετικότερες συνδέσεις TCP Reno
2. Το Vegas μπορεί να μην σταθεροποιήσει εάν οι απομονωτές είναι μικροί, οδηγώντας στη συμπεριφορά που είναι παρόμοια με αυτό του TCP Reno

2.2.5 TCP Veno

Το TCP Veno είναι ένα end-to-end σχέδιο ελέγχου συμφόρησης που μπορεί να βελτιώσει την απόδοση του TCP αρκετά σημαντικά πέρα από τα ετερογενή δίκτυα, ιδιαίτερα όταν αποτελούν οι ασύρματες συνδέσεις μέρος τέτοιων δικτύων. Η βασική καινοτομία στο Veno είναι η αύξηση του αλγορίθμου ελέγχου συμφόρησης Reno με τη χρησιμοποίηση της κατ' εκτίμηση κατάστασης μιας σύνδεσης βασισμένης σε Vegas. Αυτό το σχέδιο μειώνει σημαντικά τη «τυφλή» μείωση του παραθύρου TCP ανεξάρτητα από την αιτία της απώλειας πακέτων. Το εμφανές χαρακτηριστικό γνώρισμα του TCP Veno είναι ότι χρειάζεται μόνο την απλή τροποποίηση στην πλευρά αποστολέων της λίστας πρωτοκόλλου Reno. Εξετάζοντας τα πρακτικά ζητήματα - η δυνατότητα ανάπτυξης και η συμβατότητα (προσαρμογή με τις συνδέσεις κληρονομιών), TCP Veno μπορούν να επεκταθούν γρήγορα στη «καυτή» κινητή βιομηχανία Διαδικτύου.

2.2.6 TCP Westwood/+

Το Westwood+ εξετάζει και τις μεγάλες τιμές bandwidth/RTT και την τυχαία απώλεια πακέτων μαζί με τα δυναμικά μεταβαλλόμενα φορτία δικτύων. Αναλύει την κατάσταση της μεταφοράς με την εξέταση τα πακέτα αναγνώρισης. Το Westwood+ είναι μια τροποποίηση του αλγορίθμου TCP Reno.

Το TCP Westwood είναι ένα σχέδιο που υιοθετείται από την πηγή TCP για να υπολογίσει το διαθέσιμο εύρος ζώνης και να χρησιμοποιήσει την εκτίμηση εύρους ζώνης που ανακτεί γρηγορότερα, επιτυγχάνοντας κατά συνέπεια την υψηλότερη ρυθμοαπόδοση. Είναι βασισμένο σε δύο έννοιες: η end-to-end εκτίμηση του διαθέσιμου εύρους ζώνης και του τρόπου που μια τέτοια εκτίμηση χρησιμοποιείται για να θέσει το κατώτατο όριο αργής-έναρξης και το παράθυρο συμφόρησης. Επίσης, είναι σημαντικό να σημειωθεί ότι η ανατροφοδότηση είναι μόνο end-to-end και δεν εξαρτάται από οποιουδήποτε

ενδιάμεσους κόμβους στο επίπεδο δικτύων. Η πηγή TCP Westwood (TCPW) υπολογίζει συνεχώς το ποσοστό πακέτων της σύνδεσης με κατάλληλο υπολογισμό του μέσου όρου στο ποσοστό επιστροφής ACKs. Αυτή η εκτίμηση χρησιμοποιείται για να υπολογίσει το επιτρεπόμενο κατώτατο όριο παραθύρων και αργής-έναρξης συμφόρησης που χρησιμοποιείται αφότου ένα επεισόδιο συμφόρησης είναι ανιχνευμένο μετά από τρεις διπλές αναγνωρίσεις ή ένα διάλειμμα. Αντίθετα από το TCP Reno, το οποίο απλά χωρίζει κατά το ήμισυ το παράθυρο συμφόρησης μετά από τρία dupacks, τα TCPW προσπαθούν για να λάβουν μια ευφύστερη απόφαση. Επιλέγει ένα κατώτατο όριο αργής-έναρξης και ένα παράθυρο συμφόρησης που είναι σύμφωνα με το αποτελεσματικό ποσοστό σύνδεσης κατά την διάρκεια της συμφόρησης.

Αυτοί οι τύποι τεχνικών για την εκτίμηση εύρους ζώνης έχουν προταθεί από πριν, (packet pair and TCP Vegas) αλλά, λόγω των τεχνικών λόγων που δεν έχουν επεκταθεί επάνω στο δίκτυο. Το βασικό πράγμα για το TCPW είναι ότι εξετάζει το δίκτυο για το πραγματικό ποσοστό που μια σύνδεση επιτυγχάνει κατά τη διάρκεια της μεταφοράς στοιχείων, όχι το διαθέσιμο εύρος ζώνης προτού να αρχίσει η σύνδεση. TCPW προσφέρει διάφορα χαρακτηριστικά γνωρίσματα που δεν είναι διαθέσιμα στο TCP Reno ή TCP SACK. Η γνώση του διαθέσιμου εύρους ζώνης μπορεί να χρησιμοποιηθεί για να ρυθμίσει το ποσοστό μιας μεταβλητής πηγής ποσοστού. Στο TCPW ο αποστολέας υπολογίζει συνεχώς την εκτίμηση εύρους ζώνης σύνδεσης (BWE) που ορίζεται ως το μερίδιο του εύρους ζώνης δυσχερειών που χρησιμοποιείται από τη σύνδεση. Μετά από μια ένδειξη απώλειας πακέτων, ο αποστολέας επαναρυθμίζει το παράθυρο συμφόρησης και το κατώτατο όριο αργής-έναρξης βασισμένα στο BWE ως $cwnd = BWE \times RTT$.

Ένα άλλο σημαντικό στοιχείο αυτής της διαδικασίας είναι η εκτίμηση RTT. Η RTT πρέπει για να υπολογίσει το παράθυρο που υποστηρίζει το κατ' εκτίμηση ποσοστό BWE. Ιδανικά, το RTT πρέπει να μετρηθεί όταν η δυσχέρεια είναι κενή. Στην πράξη, τίθεται ίσο με τη γενική ελάχιστη μετ' επιστροφής καθυστέρηση (RTTmin) που μετριέται μέχρι τώρα σχετικά με εκείνη την σύνδεση. Στο TCPW, οι αυξήσεις παραθύρων συμφόρησης κατά τη διάρκεια της αργής αποφυγής έναρξης και συμφόρησης παραμένουν οι ίδιες όπως στο Reno □ που είναι είναι εκθετικές και γραμμικές, αντίστοιχα.

Σε περίπτωση 3 dupacks, το TCPW θέτει το παράθυρο συμφόρησης και το κατώτατο όριο αργής-έναρξης ως εξής:

```
ssthresh = (BWE*RTTmin)/MSS
```

```
if(cwnd > ssthresh) /*congestion avoidance*/
```

```
cwnd = ssthresh;
```

```
endif
```

Στην περίπτωση μιας απώλειας πακέτων που υποδεικνύεται από το διάλειμμα η λήξη, cwnd και ssthresh είναι σύνολο ως εξής:

```
cwnd = 1;
```

```
ssthresh = (BWE*RTTmin)/MSS;
```

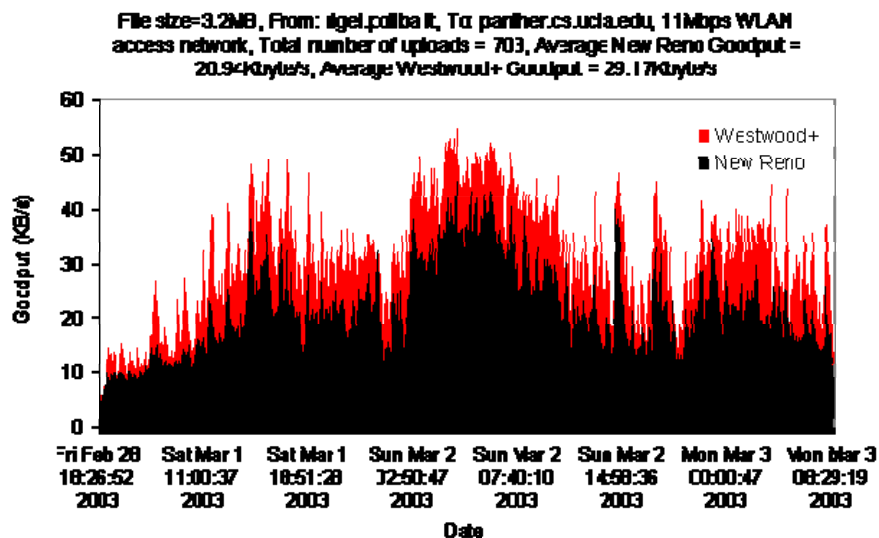
```
if(ssthresh < 2)
```

```
ssthresh = 2;
```

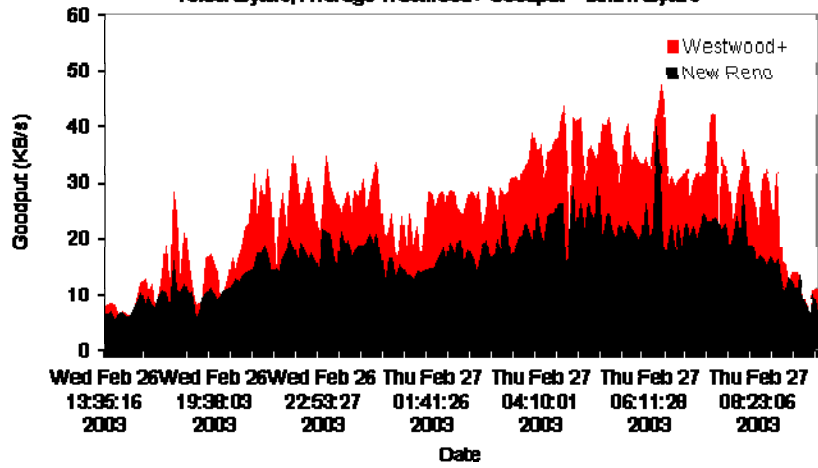
```
endif;
```

Η πρόσφατη έρευνα για την απόδοση του TCPW πέρα από τους μεγάλους σωλήνες εύρους ζώνης περιλαμβάνει τις τροποποιήσεις TCPW στο TCP Westwood με Bulk Repeat (TCPW BR). Το TCPW BR έχει τρεις sender-side τροποποιήσεις, ονομαστικά Bulk Repeat, σταθερό Retransmission time-out, και έξυπνη ρύθμιση παραθύρου για να βοηθήσει έναν αποστολέα να ανακτήσει από τις πολλαπλάσιες απώλειες στο ίδιο παράθυρο συμφόρησης και για να κρατήσει το μέγεθος των παραθύρων εύλογα μεγάλο όταν δεν υπάρχει καμία συμφόρηση κατά μήκος της πορείας. Το TCPW BR χρησιμοποιεί επίσης έναν αλγόριθμο διαφοροποίησης της απώλειας (LDA), ο οποίος είναι βασισμένος σε δύο σχέδια: την ακίδα και το ποσοστό gap του κατώτατου ορίου, το οποίο χρησιμοποιείται για τη διαφοροποίηση μεταξύ των απωλειών λόγω της συμφόρησης και των απωλειών λόγω του λάθους. Στις

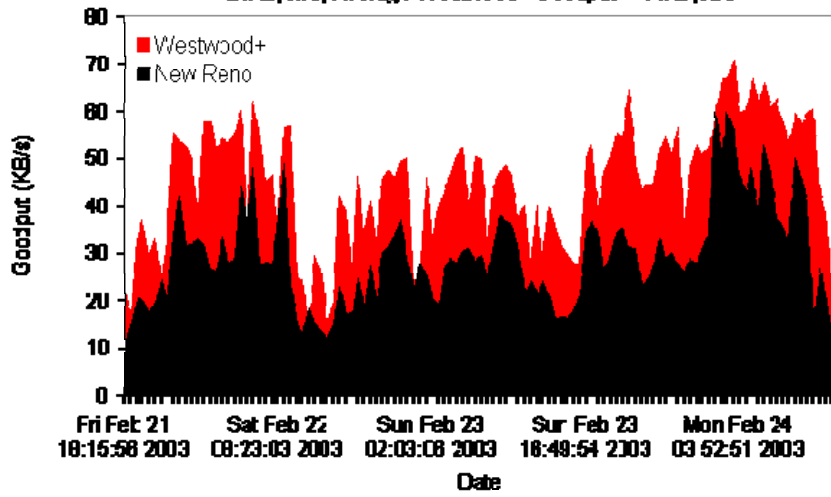
απώλειες λόγω της συμφόρησης, “congestion loss mode” το TCPW BR λειτουργεί με τον ίδιο τρόπο όπως το TCPW. Σε περιπτώσεις απώλειας λαθών, “error loss mode” το TCPW BR στηρίζεται στις τρεις sender-side τροποποιήσεις που συζητούνται παραπάνω. Αυτό το πρωτόκολλο έχει παρουσιάσει σημαντική βελτίωση απόδοσης στα βαριά περιβάλλοντα απώλειας. Το TCPW έχει τροποποιηθεί επίσης για την απόδοσή του πέρα από τα μεγάλα δίκτυα εύρους ζώνης. Οι τεχνικές όπως το προσαρμοστικό καινούριο ξεκίνημα (Astart), paced-Westwood, TCP-Westwood με το easy-RED, και το TCP Westwood με τα ποσοστά εκτιμήσεων που ερευνούν τα ζητήματα αμεροληψίας, αποδοτικότητας, φιλικής διάθεσης, και απόδοσης του TCPW πέρα από τα υψηλού εύρους ζώνης- προϊόντα καθυστέρησης δικτύων τα οποία έχουν μικρούς απομονωτές.



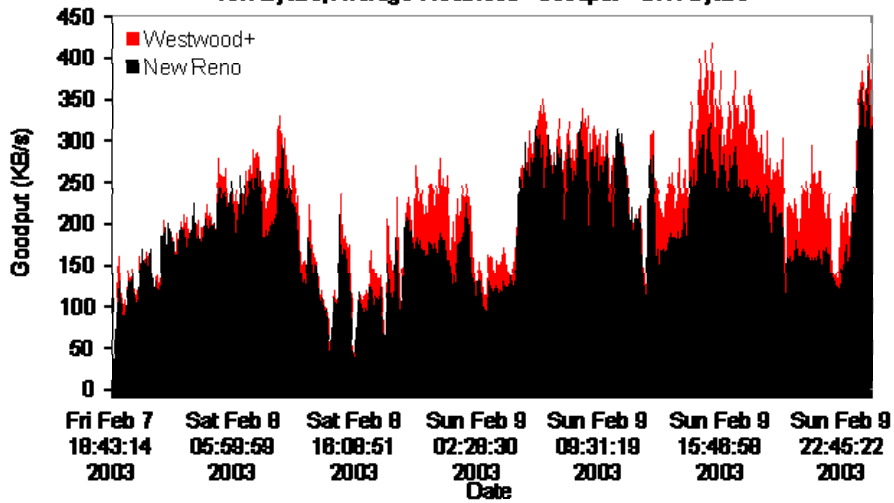
File size=3.2MB, From: rigel.poliba.it, To: panther.cs.ucla.edu, 11Mbps WLAN access network, Total number of uploads = 197, Average New Reno Goodput = 16.86Kbyte/s, Average Westwood+ Goodput = 25.21Kbyte/s



File size=32MB, From: rigel.poliba.it, To: panther.cs.ucla.edu, 11Mbps WLAN access network, Total number of uploads = 115, Average New Reno Goodput = 29Kbyte/s, Average Westwood+ Goodput = 44Kbyte/s



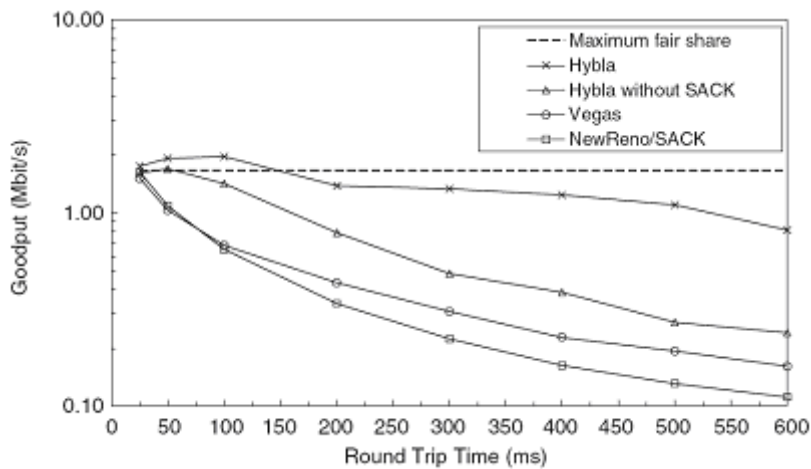
File size=32MB, From: rigel.poliba.it, To: signserv.signal.uu.se, 11Mbps WLAN access network, Total number of uploads = 500, Average New Reno Goodput = 187Kbyte/s, Average Westwood+ Goodput = 217Kbyte/s



2.2.7 TCP Hybla

Το TCP Hybla προτάθηκε προκειμένου να διαβιβαστούν τα στοιχεία αποτελεσματικά πέρα από τις δορυφορικές συνδέσεις και «να υπερασπιστεί» η μετάδοση ενάντια στις ροές TCP από άλλη προέλευση.

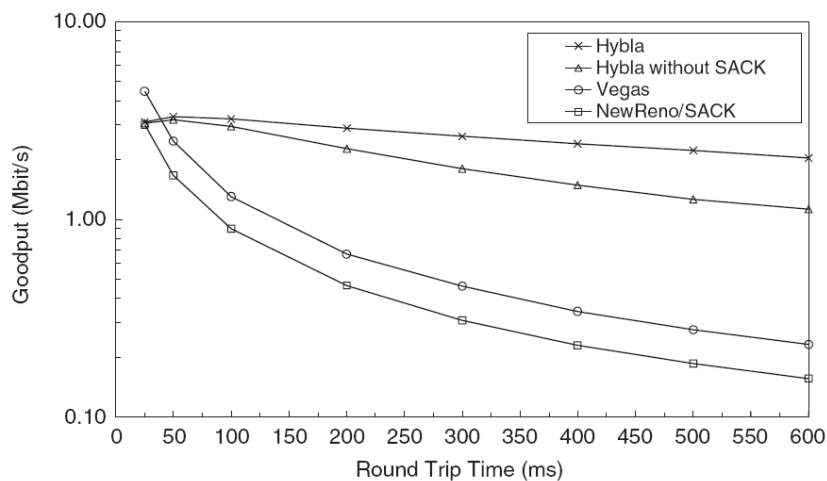
Απόδοση στην παρουσία συμφόρησης



Σχήμα 1. Απόδοση διαφορετικών TCP εκδόσεων στην παρούσα συμφόρηση.

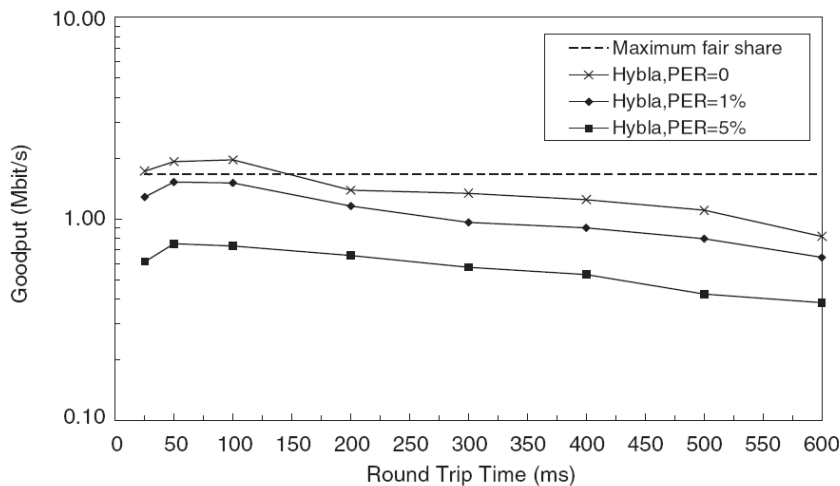
1. Όσο πιο μακροχρόνια η ασύρματη σύνδεση RTT, τόσο χειρότερη η απόδοσή του στα πρότυπα TCP (NewReno), με ένα πολύ γρήγορο ποσοστό υποβάθμισης goodput.
2. Το TCP Reno με την επιλογή TCP SACK παρουσιάζει την ίδια απόδοση του TCP NewReno, κατά συνέπεια η επιλογή TCP SACK από μόνη της δεν είναι αποτελεσματική στην παρουσία συμφόρησης.
3. Το TCP Vegas λειτουργεί μόνον ελαφρώς καλύτερα από το TCP NewReno.
4. Το TCP Hybla χωρίς την επιλογή TCP SACK παρουσιάζει εμφανή βελτίωση απόδοσης εντούτοις, η απόδοση εξαρτάται ακόμα από το RTT, αν και πολύ λιγότερο από πριν.
5. Το TCP Hybla (με την επιλογή TCP SACK) αποδεικνύεται πολύ αποτελεσματικό στην αντίδραση των αποτελεσμάτων των μακροχρόνιων RTT και απωλειών συμφόρησης, η ουσιαστική απόδοση ανεξαρτησίας του RTT επιτυγχάνεται και η ασύρματη σύνδεση δεν είναι πλέον μειονέκτημα.

Απόδοση στην παρουσία Link Losses



1. Αν και η προέλευση του προβλήματος είναι απολύτως διαφορετική, τα αποτελέσματα είναι ποιοτικά τα ίδια με εκείνους που αναφέρονται στην περίπτωση συμφόρησης που μόλις συζητήθηκε. Η ανωτερότητα του TCP Hybla είναι προφανής (κέρδος απόδοσης υψηλότερο από δέκα για $RTT > 300ms$ όσον αφορά NewReno).
2. Η βελτίωση απόδοσης λόγω του συνυπολογισμού της επιλογής TCP SACK στο TCP Hybla είναι λιγότερο εμφανής από, την μόνη περίπτωση συμφόρησης.
3. Αν και το TCP Hybla παρέχει το α περισσότερο από ικανοποιητικό goodput (3-2 MBIT/s) η θεωρητική ικανότητα του καναλιού $\pi 10$ Mbit/s είναι ακόμα μακριά από το να επιτευχθεί.

Απόδοση στην παρουσία και των δύο Congestion and Link Losses



1. Η εισαγωγή της απώλειας πακέτων λόγω των σφαλμάτων στις συνδέσεις προκαλεί μια σημαντική υποβάθμιση της απόδοσης, όσον αφορά μόνο την περίπτωση συμφόρησης.
2. Η επιδείνωση στην απόδοση λόγω της ταυτόχρονης παρουσίας της συμφόρησης και επειδή τα σφάλματα στις συνδέσεις είναι χαμηλότερα

από το άθροισμα των δύο αποσυνδεδεμένων μειωνεκτιμάτων, πιθανές επειδή οι πλαστές μειώσεις cwnd που προκαλούνται από τα λάθη συνδέσεων, στην παρουσία συμφόρησης, παρουσιάζουν τη θετική παρενέργεια του χαμηλώματος της πιθανότητας της υπερχείλισης των πακέτων.

3. Συγκριτική μελέτη ως προς την απόδοση αλγορίθμων ελέγχου συμφόρησης TCP

3.1 Εισαγωγή

Το κεφάλαιο αυτό είναι και το κυρίως μέρος της πτυχιακής μου εργασίας όπου παρουσιάζονται συγκριτικές μελέτες ανάμεσα στους διάφορους αλγόριθμους που αναφέραμε στο προηγούμενο κεφάλαιο . Μέσα από αυτά θα βγάλουμε χρήσιμα συμπεράσματα για το ποιος αλγόριθμος είναι καλύτερος να χρησιμοποιήσουμε ανάλογα με την χρήση που θέλουμε να κάνουμε.

3.2 Η Απόδοση των TCP Tahoe, του Reno, και του SACK σε Διάφορα Σενάρια

3.2.1 Γενική Περιγραφή

Αυτή η εργασία χρησιμοποιεί προσομοιώσεις για να εξερευνήσει την απόδοση των διαφορετικών εκδόσεων TCP. Συγκρίνουμε το TCP Tahoe, Reno, και SACK στα διάφορα σενάρια. Η ανάλυση απόδοσης στηρίχτηκε στην προσομοίωση με διαφορετικές πτώσεις πακέτου από ένα παράθυρο δεδομένων και των διαφορετικά φορτία κυκλοφορίας. Βρήκαμε από τα αποτελέσματα προσομοίωσης ότι το TCP Reno πάσχει από τα προβλήματα απόδοσης όταν πέφτουν πολλαπλά πακέτα ενώ αποδίδει τόσο καλά όσο εργάζεται το SACK όταν πέφτει μόνο ένα πακέτο. Το SACK TCP ξεπερνά σε απόδοση το TCP Tahoe και Reno όταν πέφτουν πολλαπλά πακέτα.

Δεδομένου ότι το Διαδίκτυο συνεχίζει να επεκτείνεται σε μέγεθος, ποικιλομορφία και η προσιτότητα, οι μηχανισμοί ελέγχου συμφόρησης παίζουν έναν συνεχώς αυξανόμενο ρόλο στο σημερινό Διαδίκτυο. Τα πρωτόκολλα ελέγχου συμφόρησης TCP έχουν υποβληθεί σε τεράστιες βελτιώσεις. Ένα δίκτυο θα είναι κορεσμένο κάτω από υψηλό φορτίο δικτύου. Σε αυτή τη κατάσταση οι buffers αλληλοεπαφής του οικοδεσπότη και του δρομολογητή μπορεί να ξεχειλίσουν αναγκάζοντας τα πακέτα για να πέσουν. Εφαρμόζοντας τον έλεγχο συμφόρησης με αυτόν τον τρόπο εξασφαλίζεται η σταθερότητα δικτύου και αποτρέπεται η κατάρρευση συμφόρησης. Στο TCP, οι στρατηγικές αποκατάστασης λάθους είναι συσχετισμένες με το σχέδιο ελέγχου συμφόρησης στον ακόλουθο τρόπο. Το TCP μετρά συνεχώς το (Round Trip Time)-RTT και υπολογίζει την αναμενόμενη απόκλιση στην καθυστέρηση από το μέσο όρο. Εάν η τρέχουσα καθυστέρηση είναι μεγαλύτερη από το μέσο όρο περισσότερο από τέσσερις φορές της αναμενόμενης απόκλισης, το TCP υιοθετεί την αποφυγή συμφόρησης ως ένα τρόπο για να ασχοληθεί με τα χαμένα πακέτα. Υποτίθεται ότι η απώλεια πακέτου προκαλείται από τη συμφόρηση κάπου στο δίκτυο μεταξύ της πηγής και του προορισμού. Η εφαρμογή του σχεδίου ελέγχου συμφόρησης είναι συνυφαινόμενος με το παράθυρο του TCP βασισμένο στο σχέδιο ελέγχου ροής μέσω της χρήσης των δύο από πλευράς αποστολέα μεταβλητών κατάστασης, το παράθυρο συμφόρησης (cwnd) και το κατώτατο όριο αργής-εκκίνησης (sssthresh) [1], [2], [3].

Το TCP χρησιμοποιεί ένα αναλογικό μηχανισμό παραθύρου που καταφέρνει τον αξιόπιστο, με εντάξει παράδοση και ροή έλεγχο συμφόρησης. Η βάση του ελέγχου συμφόρησης του TCP βρίσκεται στους ακόλουθους αλγόριθμους: αργή εκκίνηση, αποφυγή συμφόρησης, γρήγορη αναμετάδοση και γρήγορη αποκατάσταση.

Η αργή εκκίνηση χρησιμοποιείται στην αρχή της μεταφοράς, ή μετά από την επιδιόρθωση της απώλειας που ανιχνεύεται από το χρονόμετρο αναμετάδοσης. Κατά τη διάρκεια της αργής εκκίνησης, το παράθυρο συμφόρησης αυξάνεται το μέγιστο κατά ένα MSS (maximum segment size : μέγιστο μέγεθος τμήματος) στη πλευρά του αποστολέα για κάθε λαμβανόμενο ACK, το οποίο αναγνωρίζει νέα δεδομένα. Αυτό ανέρχεται αποτελεσματικά σε διπλασιασμό του παραθύρου συμφόρησης για κάθε χρόνο στρογγυλού ταξιδιού. Η αργή εκκίνηση τελειώνει, όταν το παράθυρο συμφόρησης φθάνει την αρχικά ορισμένη αξία του κατώτατου ορίου, ή όταν εμφανίζεται η συμφόρηση, και μετά αναλαμβάνει η αποφυγή συμφόρησης. Στην αποφυγή συμφόρησης, το παράθυρο συμφόρησης ρυθμίζεται δυναμικά, για να

προσαρμόσκει στις καταστάσεις δικτύου. Επάνω στην ανίχνευση μιας απώλειας πακέτου, η αξία κατώτατων ορίων τίθεται στο ήμισυ της αρχικής αξίας του κατώτατου ορίου. Κατά συνέπεια το TCP εφαρμόζει την Προσθετική Αύξηση και την Πολλαπλασιαστική μείωση (AIMD) για να υπερνικήσει τη συμφόρηση στα δίκτυα. Υπάρχουν δύο ενδείξεις μιας απώλειας πακέτου: η μία είναι το διάλειμμα αναμετάδοσης (RTO). Η άλλη είναι η άφιξη τριών διπλότυπων αναγνωρίσεων (DUPACKs). Εάν μια απώλεια πακέτων είναι υποδειγμένη από ένα διάλειμμα, το TCP είναι αρκετά σίγουρο ότι το δίκτυο είναι κορεσμένο και ως εκ τούτου υιοθετεί τον αλγόριθμο αργής εκκίνησης για να ανακτήσει από αυτόν τον κορεσμό και μετά αναλαμβάνει η αποφυγή συμφόρησης. Αφ' ενός η παραλαβή τριών διπλότυπων ACKs λέει περισσότερα από μόνο μια απώλεια πακέτου. Δεδομένου ότι ο δέκτης μπορεί να παραγάγει τα διπλά ACKs μόνο όταν παραλαμβάνεται ένα άλλο τμήμα, αυτό το τμήμα έχει αφήσει το δίκτυο και είναι στο buffer του δέκτη το οποίο δείχνει ότι υπάρχουν ακόμα δεδομένα που ρέουν μεταξύ των δύο άκρων και έτσι προκειμένου να αποφύγει την απότομη μείωση ροής, το TCP υιοθετεί τον αλγόριθμο γρήγορης αναμετάδοσης, γρήγορης αποκατάστασης και αποφυγής συμφόρησης.

Η γρήγορη αναμετάδοση και η γρήγορη αποκατάσταση χρησιμοποιείται όπου υπάρχει απώλεια μερικών τμημάτων. Ο αποστολέας λαμβάνει το ACK για το κάθε τμήμα που παραλαμβάνεται από το δέκτη. Εάν υπάρχει μια απώλεια ενός ή δύο τμημάτων, ο αποστολέας λαμβάνει ένα αντίγραφο αναγνώρισης (DUPACK) που περιέχει το ίδιο αριθμό ACK όπως το τελευταίο σταλμένο ACK. Επάνω στη λήψη τριών τέτοιων DUPACKs, ο αποστολέας, χωρίς να περιμένει το χρονόμετρο αναμετάδοσης να κάνει διάλειμμα, στέλνει το χαμένο τμήμα/τμήματα και αρχίζει πάλι την αποστολή των μη αναγνωρισμένων δεδομένων, κρατά τον αγωγό γεμάτο και περιμένει για τα ACKs.

Το TCP Tahoe αναφέρεται στο TCP που εφαρμόζεται με τους αλγόριθμους της αργής εκκίνησης, της αποφυγής συμφόρησης και γρήγορης αναμετάδοσης, και αυτό εφαρμόστηκε αρχικά στο 4.3 BSD Tahoe TCP το 1988. Για το TCP Reno, κρατά τους αλγόριθμους του TCP σε Tahoe (αργής εκκίνησης, αποφυγής συμφόρησης και γρήγορης αναμετάδοσης) προστιθέμενους στο TCP, επιπλέον, πρόσθεσε τη γρήγορη αποκατάσταση στις εφαρμογές και εφαρμόστηκε αρχικά στο 4.3 BSD TCP Reno το 1990. Το TCP SACK επιτρέπει στους δέκτες να ACK εκτός της ακολουθίας δεδομένων και στοχεύει στην εξάλειψη των διαλειμμάτων που

προκύπτουν στο TCP Reno λόγω των πολλαπλών απωλειών από το ίδιο παράθυρο. Η ανάπτυξη παρατίθεται στον [πίνακα 1](#).

TCP version	Difference
Tahoe	Added slow-start, congestion avoidance and fast retransmit
Reno	Base on Tahoe, added fast recovery
SACK	Base on Reno, added SACK (select acknowledge)

ΠΙΝΑΚΑΣ 1. ΑΝΑΠΤΥΞΗ ΤΟΥ TCP

Ακολουθώντας την ανάπτυξη του TCP, υπάρχουν πολλές διαφορετικές εκδόσεις TCP. Θεωρώντας τη σημασία που εκτελεί το TCP, σκεφτόμαστε ότι είναι κερδοφόρο να ρίξουμε μια στενή ματιά στη διαφορά μερικών σημαντικών TCP εκδόσεων.

Ακολουθώντας τις προσπάθειες για να βελτιωθεί το προηγούμενο TCP και τη διαφορά μεταξύ αυτού του TCP, μπορούμε να πάρουμε μια γεύση του πώς οι άλλοι ερευνητές προσπαθούν να καταστήσουν την απόδοση καλύτερη. Σημαντικότερα, από τις προηγούμενες ερευνητικές εργασίες, ακούμε διαφορετικές απόψεις για την απόδοση του TCP Tahoe, Reno και SACK στα διάφορα σενάρια [3], [4].

Προσπαθούμε να ανακαλύψουμε την αλήθεια και ενδεχομένως μπορούμε τελικά να πάρουμε τον περιεκτική κατανόηση αυτών των αποδόσεων των διαφορετικών TCPs κάτω από διαφορετικά σενάρια.

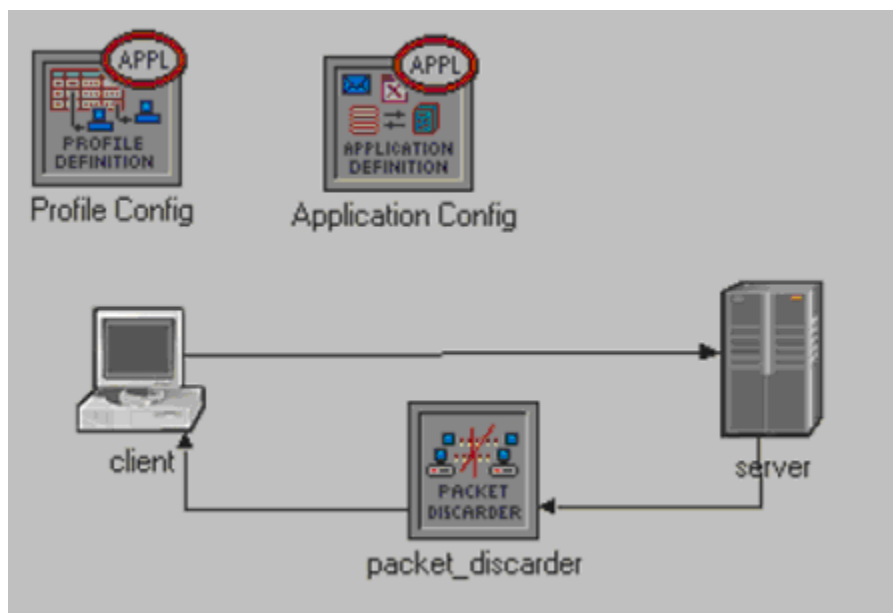
3.2.2 ΠΡΟΣΟΜΟΙΩΣΗ

Το σύστημα παρέχει την υπηρεσία της μεταφοράς των πακέτων ημερομηνίας από τους αποστολείς στους δέκτες μέσω του δικτύου, όταν συμβαίνει η συμφόρηση, το TCP θα προσπαθήσει να ανακτηθεί από αυτή γρήγορα και ελπίζουμε ότι δεν θα πάρει μεγάλη περίοδο στους αποστολείς να αρχίσουν να αναμεταδίδουν τα πακέτα που πέφτουν ή να διαβιβάσουν τα νέα πακέτα πάλι, δηλαδή μετά από μια συμφόρηση που συμβαίνει και ανακτάται, δεν θέλουμε επίσης να δούμε για μια μακρά χρονική περίοδο, να μην υπάρχει κανένας αποστολέας να διαβιβάζει πακέτο και η σύνδεση να είναι ελεύθερη.

Ο στόχος της προσομοίωσης είναι να συγκριθεί η απόδοση του ελέγχου συμφόρησης του TCP Tahoe, Reno και SACK όταν συμβαίνει η συμφόρηση οπουδήποτε στο δίκτυο. Για μία περιεκτική σύγκριση, πρέπει να κατασκευάσουμε διαφορετικά σενάρια, όπως οι διαφορετικοί τύποι φόρτου εργασίας, συσχετισμένες ή ανεξάρτητες πτώσεις πακέτου, αξιόπιστα ή αναξιόπιστα δίκτυα, κ.λπ.

A. Πείραμα

Προκειμένου να επιτευχθεί ο στόχος μας της σύγκρισης της απόδοσης χρησιμοποιούμε το λογισμικό προσομοίωσης OPNET του εμπορίου για να τρέξει τη προσομοίωσή μας. Η τοπολογία προσομοίωσης που παρουσιάζεται στο [σχήμα 1](#) χρησιμοποιείται για όλα τα σενάρια και όλους τους διαφορετικούς φόρτους εργασίας. Τα διαφορετικά σενάρια παράγονται με τον διαφορετικό αριθμό απώλεια πακέτου μεταξύ του αποστολέα και του δέκτη. Ο κόμβος απόρριψης πακέτου χρησιμοποιείται για να οδηγήσει στην απόρριψη πακέτου σε αυτή τη διασύνδεση. Παρέχει ιδιότητες για τον έλεγχο του εύρους-χρόνου και του αριθμού πακέτων μέσα σε αυτό το φάσμα που χρειάζεται να είναι απορριμμένα. Σύμφωνα με το φόρτο εργασίας που μπορεί να αντιμετωπίσουμε στην πραγματικότητα, επιλέγουμε να εκτελέσουμε την προσομοίωση και σε ανομοιόμορφο και σε ομοιόμορφα διανεμημένο φόρτο εργασίας. Το HTTP και το FTP εφαρμόστηκαν σαν εκρηκτικό και ομοιόμορφα διανεμημένο φόρτο εργασίας. Το «*Application Config*» χρησιμοποιείται για να διευκρινίσει τις εφαρμογές του HTTP και του FTP. Ο κόμβος «*Profile Config*» χρησιμοποιείται για να δημιουργήσει τα προφίλ του χρήστη σε διαφορετικούς κόμβους στο δίκτυο για να παραγάγει στρώμα εφαρμογής κυκλοφορία. Το μέγεθος του buffer στον πελάτη και στον server είναι επιλεγμένο σε 64kbyte αρκετά μεγάλο για να μην υπερχειλίσει. Από τώρα θα εξετάζουμε μόνο τη συμφόρηση της διασύνδεσης αλλά όχι την υπερχειλίση του buffer. Το εύρος ζώνης των διασυνδέσεων μεταξύ των κόμβων τίθεται σε DS1 (1.45Mbps). Προκειμένου να αξιολογηθεί η απόδοση των διαφορετικών εκδόσεων TCP επιλέγουμε τις μετρικές του μεγέθους συμφόρησης παραθύρου, το χρόνος απόκρισης και τη ρυθμοαπόδοση(throughput). Το μέσο μέγεθος του παραθύρου συμφόρησης αποκαλύπτει άμεσα την απόδοση του αλγόριθμου ελέγχου και αποφυγής συμφόρησης του TCP.



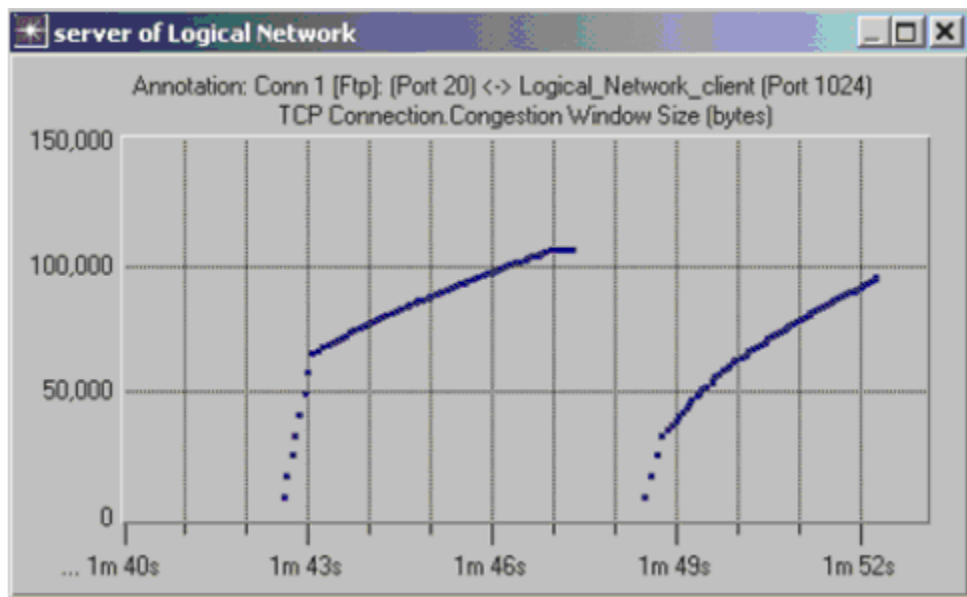
Σχήμα 1. Εξομίωση τοπολογίας

B. Αποτελέσματα Προσομοίωσης

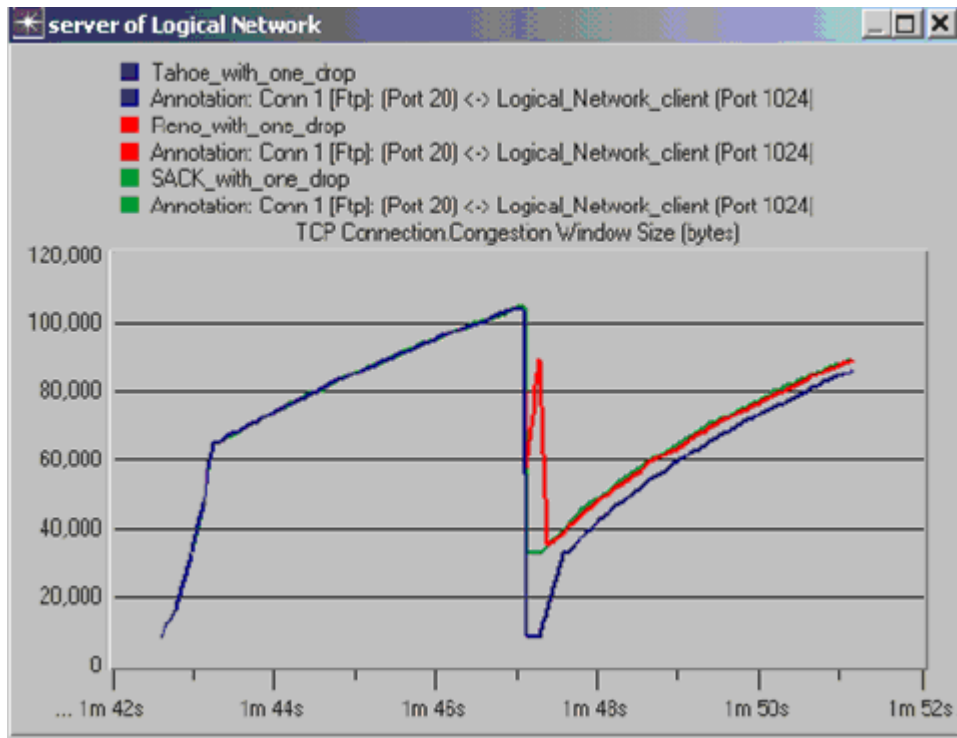
Το [σχήμα 2](#) παρουσιάζει τη συμπεριφορά του παραθύρου συμφόρησης του TCP προεπιλογής (χωρίς γρήγορη αναμετάδοση και γρήγορη αποκατάστασης) μετά από μία πτώση πακέτου ενώ το [σχήμα 3](#) παρουσιάζει τη συμπεριφορά του παραθύρου συμφόρησης του TCP Tahoe, Reno και SACK μετά από μια πτώση πακέτου.

Συγκρίνοντας το [σχήμα 3](#) και το [σχήμα 2](#), βρήκαμε ότι το TCP Tahoe, Reno και SACK είναι όλα καλύτερα από το TCP προεπιλογής (χωρίς γρήγορη αναμετάδοση και γρήγορη αποκατάστασης). Στην αρχή της μεταφοράς, το TCP Tahoe, Reno, και SACK όλα εκλιπαρούν την αργή εκκίνηση. Κατά τη διάρκεια της αργής εκκίνησης, το παράθυρο συμφόρησης αυξάνεται το μέγιστο κατά ένα MSS (maximum segment size : μέγιστο μέγεθος τμήματος) στη πλευρά του αποστολέα για κάθε λαμβανόμενο ACK. Όταν το παράθυρο συμφόρησης φθάνει την αρχικά ορισμένη αξία του κατώτατου ορίου, η αργή εκκίνηση τελειώνει. Πριν από την πτώση πακέτου το TCP Tahoe, Reno και SACK έχουν σχεδόν την ίδια συμπεριφορά. Στο χρόνο 107 δευτερόλεπτα που ένα πακέτο πέφτει, οι αντιδράσεις από το TCP Tahoe, Reno και SACK είναι διαφορετικές. Από το [σχήμα 2](#) βρήκαμε καθαρά ότι το

Tahoe χωρίς τη γρήγορη αποκατάσταση απαιτεί μία αργή εκκίνηση για να ανακτήσει από την πτώση πακέτου, ενώ το Reno και SACK είναι σε θέση να ανακτήσουν τη χρησιμοποιώντας τη γρήγορη αποκατάσταση. Το Tahoe επικαλέστηκε τη γρήγορη αναμετάδοση και την αργή εκκίνηση και το κατώτατο όριο αργής εκκίνησης είναι μειωμένο στο μισό από το προηγούμενο. Το TCP αποστολής επαναρυθμίζει το παράθυρο συμφόρησής του σε ένα. Το Reno και SACK ενεργούν σχεδόν το ίδιο. Όλα επικαλέστηκαν τη γρήγορη αποκατάσταση με το παράθυρο συμφόρησης του αποστολέα να μειώνεται στη μέση, το οποίο δίνει πολύ βελτιωμένη απόδοση και αποφεύγει την αργή εκκίνηση.

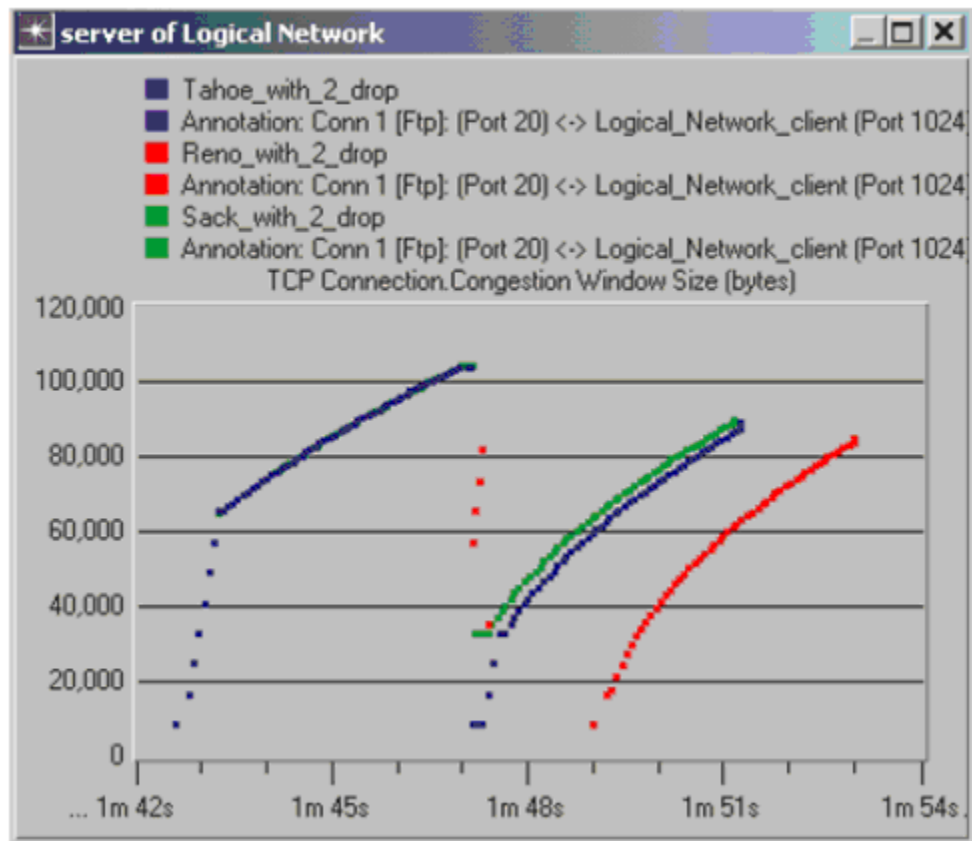


Σχήμα 2. Εξ ορισμού TCP (χωρίς γρήγορη αναμετάδοση και γρήγορη αποκατάσταση) παράθυρο συμπεριφοράς της συμφόρησης μετά από πτώση πακέτου.

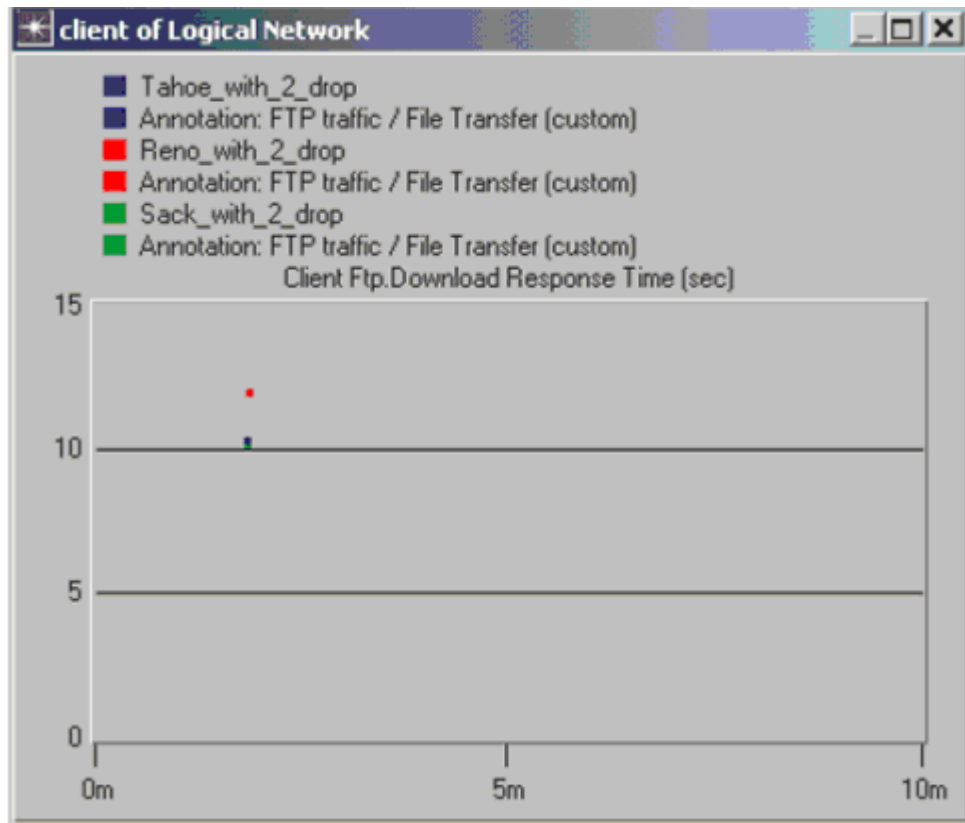


Σχήμα 3. Σύγκριση μιας συμφόρησης πτώσης παραθύρου.

Το [σχήμα 4](#) δείχνει τη συμπεριφορά του παραθύρου συμφόρησης του TCP Tahoe, Reno και SACK με δύο πεσμένα πακέτα. Το Tahoe ακόμα η επικαλείται την αργή εκκίνηση για να ανακτήσει από τις πτώσεις πακέτου ενώ το SACK ανακτά ομαλά και γρήγορα με τη διχοτόμηση του παραθύρου συμφόρησης όπως στο προηγούμενο σενάριο προσομοίωσης. Αλλά το Reno ενεργεί εντελώς διαφορετικά. Αντί της γρήγορης αποκατάστασης, το Reno υπέφερε από το τεράστιο πρόβλημα απόδοσης. Καταρχάς το Reno επικαλέσθηκε τον αλγόριθμο γρήγορης αποκατάστασης με το παράθυρο συμφόρησης διχοτομημένο για να ανακτήσει από την πτώση πακέτου. Αλλά λόγω της δεύτερης πτώσης πακέτου αυτός ο αλγόριθμος αποτυγχάνει και το Reno πρέπει να περιμένει γιατί μια αναμετάδοση λήξης χρονομέτρου για να ανακτήσει από το πεσμένο πακέτο με την αργή εκκίνηση. Επίσης από το [σχήμα 5](#), βρήκαμε ότι ο χρόνος απόκρισης του Reno είναι ο μακρύτερος μεταξύ των τριών εκδόσεων TCP ενώ ο χρόνος απόκρισης του SACK είναι ο γρηγορότερος.



Σχήμα 4. Σύγκριση του μεγέθους συμφόρησης παραθύρου για 2 πτώσεις.



Σχήμα 5. Σύγκριση χρόνου ανταπόκρισης για 2 πτώσεις.

Το [σχήμα 6](#) παρουσιάζει τη συμπεριφορά του παραθύρου συμφόρησης του TCP Tahoe, Reno και SACK με τέσσερα πεσμένα πακέτα. Όπως στις προηγούμενες προσομοιώσεις, το Tahoe εκλιπαρεί ακόμα την αργή εκκίνηση για να συνέλθει από τις πτώσεις πακέτου. Το SACK ανακτά ομαλά και γρήγορα με τη διχοτόμηση του παραθύρου συμφόρησης επίσης όπως το προηγούμενο σενάριο προσομοίωσης. Αντί της γρήγορης αποκατάστασης, το Reno υπέφερε από τεράστιο πρόβλημα απόδοσης και πρέπει να περιμένει γιατί μια αναμετάδοση λήξης χρονομέτρου για να ανακτήσει από το πεσμένο πακέτο με την αργή εκκίνηση. Συγκρίνοντας το [σχήμα 5](#) και το [σχήμα 7](#) βρήκαμε ότι ο χρόνος απόκρισης του Reno με τέσσερις απώλειες πακέτων είναι μακρύτερος από το χρόνο απόκρισης του Reno στο σενάριο με δύο απώλειες πακέτων. Αυτό σημαίνει ότι το Reno αποδίδει χειρότερα με την αύξηση της συμφόρησης ενώ η απόδοση του TCP Tahoe και SACK παραμένει σχεδόν αμετάβλητη σε σύγκριση με το σενάριο με δύο απώλειες πακέτων.

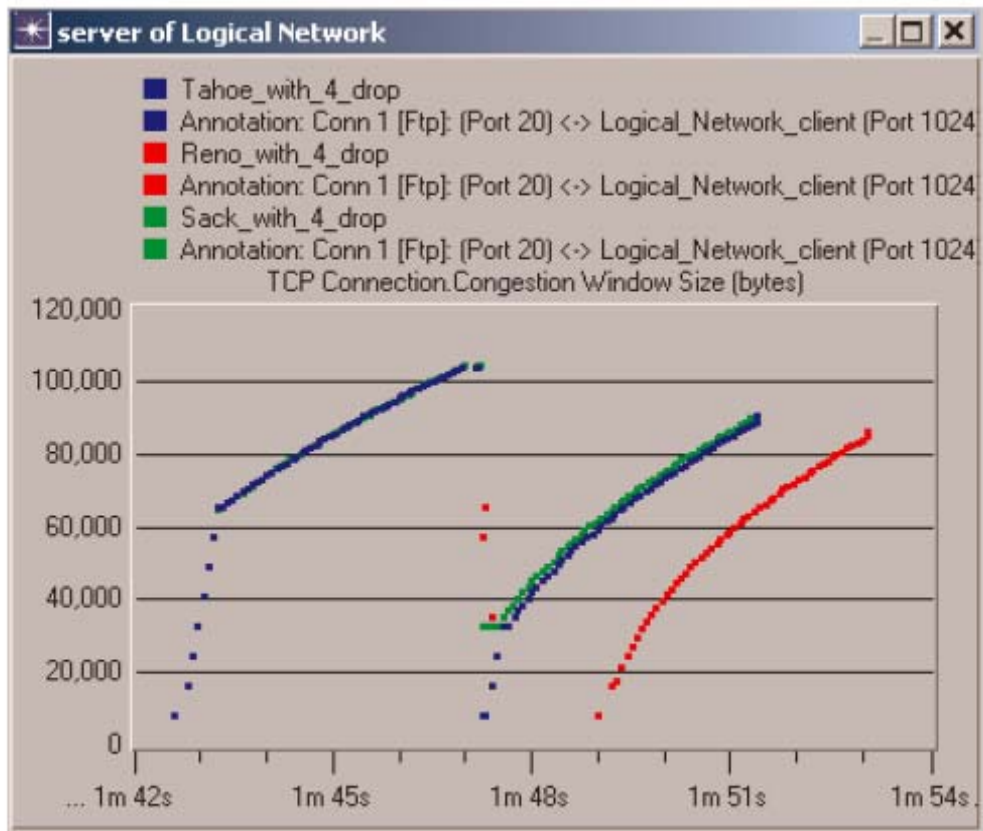
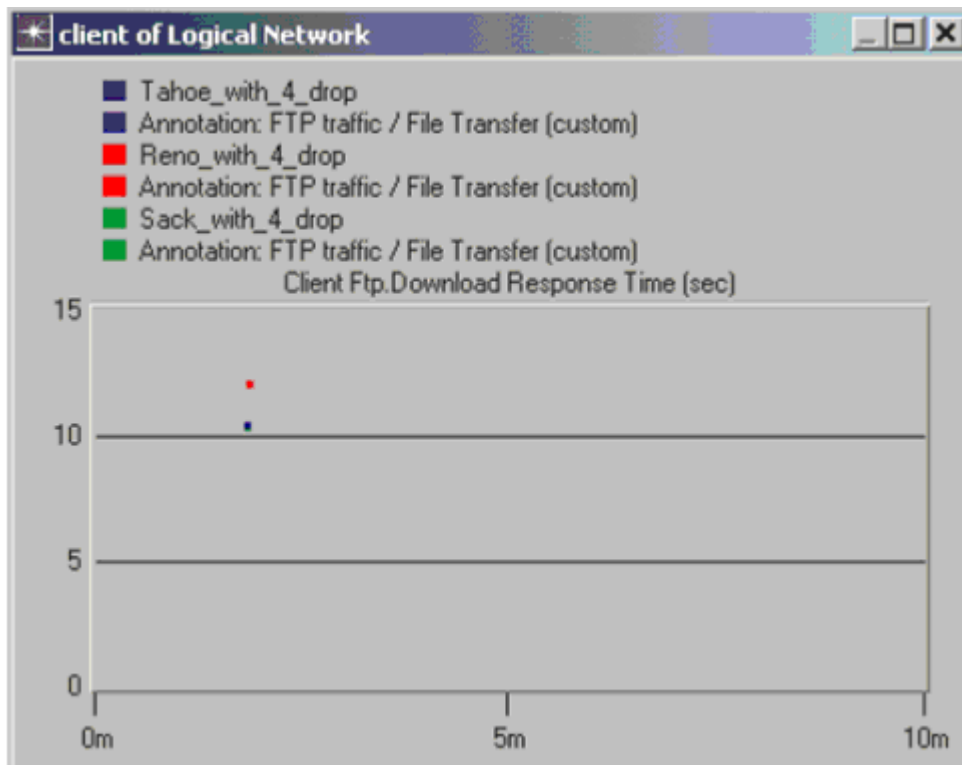


Figure 6. Comparison of congestion window size for 4 drops

Σχήμα 6. Σύγκριση του μεγέθους συμφόρησης παραθύρου για 4 πτώσεις.



Σχήμα 7. Σύγκριση της ανταπόκρισης για 4 πτώσεις.

Για HTTP φόρτο εργασίας, ως μία διασύνδεση που διαρκεί μόνο για μια πολύ μικρή χρονική περίοδο, είναι λιγότερο πιθανό ότι αρκετά πακέτα είναι πεσμένα στο ίδιο παράθυρο. Αλλά η γενική απόδοση του TCP SACK είναι λίγο καλύτερη από του Reno και του Tahoe. Για να μελετήσουμε την περίπτωση περισσότερων από μιας μείωσης πακέτου στο ίδιο παράθυρο, αυτό είναι καλύτερα να χρησιμοποιηθεί η FTP κυκλοφορία, καθώς ο χρόνος σύνδεσης για μία τέτοιου είδους κυκλοφορία είναι μακρύτερος από τη HTTP κυκλοφορία και οι πτώσεις είναι πιθανότερο να συνέβησαν σε μια παράθυρο.

3.3. Σύγκριση Δυναμικής του TCP Veno και Reno

3.3.1 Γενική Περιγραφή

Το TCP Veno εισήχθηκε πρόσφατα για να εξαλείψει την απόδοση του TCP που υποφέρει από τις ασύρματες διασυνδέσεις. Πραγματικές μετρήσεις δικτύου και ζωντανά αποτελέσματα Διαδικτύου έχουν επικυρώσει τη σημαντική βελτίωση ρυθμοαπόδοσης του Veno στα ασύρματα δίκτυα και την αρμονική του συνύπαρξη με τις συνδέσεις TCP Reno σε ενσύρματα δίκτυα. Σε αυτό το έγγραφο, καταδεικνύουμε τον έξω-από-τη φάση (*out-of-phase*) συγχρονισμό του Veno σε μονής διαδρομής κυκλοφορία, σε αντιδιαστολή με τον μέσα-στη φάση (*in-phase*) συγχρονισμό του Reno. Εκθέτονται οι λεπτομερείς μελέτες αυτών των συμπεριφορών και η αλληλεπίδρασή τους με το Reno. Επιπλέον, η προσεκτική μελέτη μας δείχνει ότι αυτός ο έξω-από-τη φάση συγχρονισμός ωφελεί τη χρησιμοποίηση συνδέσεων δικτύων, και μειώνει το περιστατικό της απώλεια συμφόρησης.

Το TCP είναι ένα αξιόπιστο προσανατολισμένο πρωτόκολλο σύνδεσης όπου εφαρμόζει τον έλεγχο ροής με τη βοήθεια ενός αλγόριθμου κανόνα παραθύρου [9]. Το TCP Reno, το οποίο χρησιμοποιεί τους αλγόριθμους αργή εκκίνηση και αποφυγής συμφόρησης για να ρυθμίσει το μέγεθος παραθύρου, είναι ευρέως επεκταμένος στο Διαδίκτυο. Κατά τη διάρκεια της φάσης αργής εκκίνησης, το παράθυρο του αυξάνεται για κάθε ACK που παραλαμβάνεται μέχρι η απώλεια πακέτου να είναι δοκιμασμένη, στο σημείο όπου το παράθυρο διχοτομείται και έπειτα ένας γραμμικής αύξησης αλγόριθμος αναλαμβάνει έως ότου η πρόσθετη απώλεια πακέτου να είναι δοκιμασμένη. Αυτός ο μηχανισμός προσθετικής αύξησης και πολλαπλασιαστικής μείωσης οδηγεί σε περιοδικές ταλαντώσεις στο παράθυρο συμφόρησης, καθυστέρηση πλήρους ταξιδιού και μήκος σειρών αναμονής της δυσχερούς buffer στην διαδρομή.

Εντούτοις, η υπόθεση στο TCP Reno ότι η απώλεια πακέτων υπονοεί τη συμφόρηση δικτύου μπορεί να μην ισχύσει για τα ασύρματα δίκτυα, στα οποία η απώλεια πακέτου μπορεί να προκληθεί από το θόρυβο, τα λάθη διασύνδεσης ή από άλλους λόγους εκτός της συμφόρησης δικτύου. Μην κάνοντας προσπάθεια να διακρίνει μεταξύ τυχαίας και λόγω συμφόρησης απώλειας, το TCP Reno είναι εξίσου ευαίσθητο και στις δύο. Αυτό μπορεί να οδηγήσει στη σημαντική αλλά περιττή *end-to-end* υποβάθμιση ρυθμοαπόδοσης.

Πρόσφατα, μια νέα παραλλαγή του TCP αποκαλούμενη ως TCP Venο [10, 11, 12] προτάθηκε να εξαλείψει τη σοβαρή μάζιγα από τις ασύρματες διασυνδέσεις. Ενσωματώνει τα πλεονεκτήματα δύο αντιτιθέμενων στρατοπέδων, του TCP Reno και Vegas [14]. Ένα ιδιαίτερο χαρακτηριστικό στοιχείο του TCP Venο είναι ότι η σημαντική βελτίωση απόδοσης του πέρα από αυτή του Reno επιτυγχάνεται από τη καλύτερη εκμετάλλευση του διαθέσιμου εύρους ζώνης που αφήνεται αχρησιμοποίητο από τις άλλες υπάρχουσες συνδέσεις, παρά να αρπάζει επιθετικά το πρόσθετο εύρος ζώνης από τις άλλες συνδέσεις. Εκτός αυτού, το Venο απαιτεί μόνο την τροποποίηση της λίστας πρωτοκόλλου από πλευράς αποστολέα, κάνοντάς το ευκολότερο να αναπτυχθεί πάνω στο τρέχον Διαδίκτυο.

Νωρίτερα το 1990, Shenker και λοιποί [7-8] έχει μελετήσει το φαινόμενο συγχρονισμού μέσα-στη φάση (in-phase) του Reno στη μονόδρομη κυκλοφορία, και παρατήρησε ότι οι συνυπάρχοντες συνδέσεις ρίχνουν τα πακέτα σχεδόν ταυτόχρονα όταν φθάνουν την χωρητικότητα διαδρομής. Σε αυτό το έγγραφο, μελετάμε τη συμπεριφορά δυναμικής του αλγόριθμου ελέγχου συμφόρησης του Venο και τις λεπτομερείς αλληλεπιδράσεις του με συνδέσεις TCP Reno. Τα πειραματικά αποτελέσματα καταδεικνύουν ένα ενδιαφέρον χαρακτηριστικό του Venο – τον έξω-από-τη φάση συγχρονισμό, ο οποίος είναι αρκετά διαφορετικός από τη δυναμική του παραθύρου του Reno που παρατηρήθηκε πριν. Επιπλέον, η εκτενής μελέτη μας στη συνύπαρξη κάτω από διαφορετικές καταστάσεις επιβεβαιώνει περαιτέρω ότι το Venο λειτουργεί πράγματι αρμονικά με τις σε ανταγωνισμό συνδέσεις του Reno πάνω σε ενσύρματα δίκτυα. Η ανεπαίσθητη βελτίωση της, πετυχαίνεται από αυτό τον έξω-από-τη φάση συγχρονισμό.

3.3.2. Ο ΜΗΧΑΝΙΣΜΟΣ ΤΟΥ TCP VENO

Το TCP Venο χρησιμοποιεί την ιδέα του σχεδίου ελέγχου συμφόρησης από το TCP Vegas, και το ενσωματώνει στο σχέδιο εξέλιξης παραθύρου του Reno. Στο TCP Vegas, αυτός ο έλεγχος είναι υπολογισμένος από τη διαφορά μεταξύ της μετρημένης και αναμενόμενης ρυθμοαπόδοσης, δηλαδή,

$$DIFF = (\text{Αναμενόμενη} - \text{Πραγματική}) \quad (1)$$

(Expected – Actual)

με την *Αναμενόμενη* = $cwnd/BaseRTT$ και το *BaseRTT* είναι το ελάχιστο από όλα τα μετρημένα RTT (χρόνοι στρογγυλού ταξιδιού)¹.

¹ Στο Vegas, το *BaseRTT* ενημερώνεται συνεχώς σε όλη τη διάρκεια του ζωντανού χρόνου των TCP συνδέσεων με τον ελάχιστο μετ'επιστροφής χρόνο που συλλέχθηκε μέχρι τότε. Στο TCP Veno, εντούτοις, το *BaseRTT* επαναρυθμίζεται όποτε ανιχνεύεται η απώλεια πακέτων, είτε λόγω του διαλείμματος είτε των διπλών ACK. Το *BaseRTT* ενημερώνεται έπειτα όπως στον αρχικό αλγόριθμο του Vegas μέχρι να προκληθεί η επόμενη γρήγορη-αποκατάσταση ή η αργή-εκκίνηση. Αυτό γίνεται για να λάβει υπόψη την μεταβαλλόμενη κυκλοφορία από τις άλλες συνδέσεις και ότι το εύρος ζώνης που αποκτήθηκε από μια μονή σύνδεση μεταξύ πολλών συνδέσεων μπορεί να αλλάξει από καιρό σε καιρό, αναγκάζοντας το *BaseRTT* να αλλάξει επίσης.

Η *Πραγματική* είναι η μετρημένη ρυθμοαπόδοση στον αποστολέα που δίνεται από $cwnd/RTT$, όπου το RTT είναι ο πραγματικός μετ'επιστροφής χρόνος ενός πακέτου με ετικέτα. Για να κυριολεκτήσουμε, η *Αναμενόμενη* όπως καθορίζεται είναι η καλύτερη δυνατή ρυθμοαπόδοση, δεδομένου ότι το *BaseRTT* είναι το ελάχιστο όλων των μετρημένων RTT.

Στο Veno, η διαφορά $DIFF * BaseRTT$ χρησιμοποιείται για να υπολογίσει τον αριθμό πακέτων που συσσωρεύονται στον δυσχερή buffer. Εάν υπάρχουν περισσότερα από ένα ανώτερο όριο (β) πακέτων που περιμένουν στη σειρά για επεξεργασία, η TCP σύνδεση λέγεται ότι έχει εξελιχθεί σε μία συμφορητική κατάσταση. Διαφορετικά, είναι σε μία μη-συμφορητική. Όπως στο TCP Reno, η απώλεια πακέτου στη κατάσταση συμφορητική (*συμφορητική πτώση*) θα αναγκάσει το παράθυρο να διχοτομηθεί. Εντούτοις, η απώλεια πακέτων στη μη-συμφορητική κατάσταση (*μη-συμφορητική πτώση*) θα προκαλέσει μόνο μείωση κατά 1/5 στο μέγεθος παραθύρου.

Επιπλέον, το Veno καθαρίζει την φάση προσθετικής αύξησης του Reno με τον καταναγκασμό της TCP σύνδεσης να μείνει πιο μακροχρόνια στη περιοχή λειτουργίας. Πιο αναλυτικά, οι αλγόριθμοι περιγράφονται στα [σχήματα 8-9](#).

Όταν η απώλεια πακέτου ανιχνεύεται από τη γρήγορη αναμετάδοση:

εάν ($DIFF * BaseRTT < \beta$) //πολύ πιθανό αυτό να είναι μια τυχαία απώλεια

$$ssthresh = cwndloss * (4/5)$$

διαφορετικά //πολύ πιθανό αυτό να είναι μια απώλεια συμφόρησης

$$ssthresh = cwndloss / 2$$

Όταν η απώλεια πακέτου ανιχνεύεται από το χρονόμετρο αναμετάδοσης διαλείμματος:

το $ssthresh$ τίθεται στο μισό του τρέχοντος παραθύρου

η αργή εκκίνηση εκτελείται //εκτελεί την ίδια ενέργεια όπως στο Reno

Σχήμα 8. Ο αλγόριθμος ρύθμισης του $ssthresh$ του TCP Veno

Κατά τη διάρκεια της περιόδου προσθετικής αύξησης:

εάν ($DIFF * BaseRTT < \beta$) //το διαθέσιμο εύρος ζώνης είναι υποχρησιμοποιούμενο

$$cwnd = cwnd + 1 / cwnd \text{ όταν κάθε νέο ack λαμβάνεται}$$

διαφορετικά //το διαθέσιμο εύρος ζώνης χρησιμοποιείται πλήρως

$$cwnd = cwnd + 1 / cwnd \text{ όταν κάθε άλλο νέο ack λαμβάνεται}$$

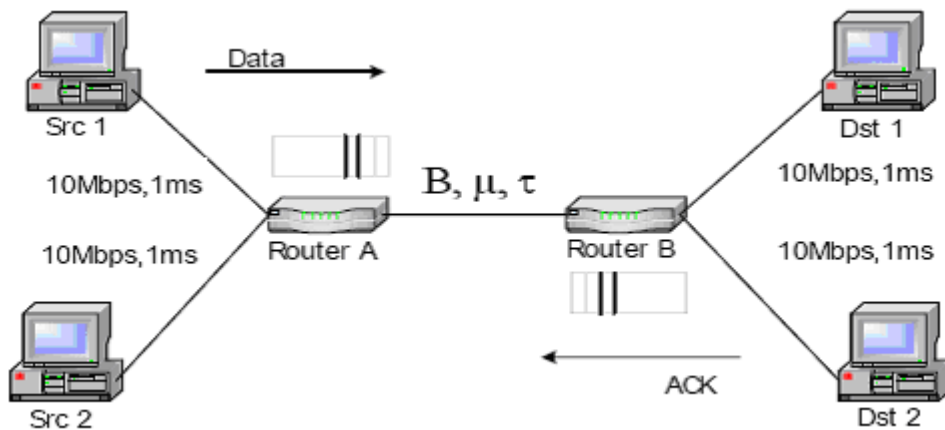
Σχήμα 9. Ο αλγόριθμος προσθετικής αύξησης του βελτιωμένου TCP Veno

Παρόμοια με [7-8], καθορίζουμε μια *εποχή* (epoch) μιας TCP σύνδεσης να είναι το χρονικό διάστημα κατά τη διάρκεια του οποίου η ποσότητα πακέτων ενός ολόκληρου παραθύρου να έχει αναγνωριστεί. Θα εστιάσουμε σε εκείνες τις ιδιαίτερες *εποχές* στις οποίες εμφανίζεται η απώλεια πακέτου, και τις αναφέρουμε ως *εποχές συμφόρησης*. Επομένως, το cwnd του Venο αυξάνει κατά ένα πακέτο σε κάθε *εποχή* όταν $DIFF * BaseRTT < \beta$, ή αυξάνει κατά ένα πακέτο μόνο κάθε δύο *εποχές* όταν $DIFF * BaseRTT \geq \beta$. Σε αντίθεση, το Reno αυξάνει πάντα ένα πακέτο σε κάθε *εποχή*. Το γενικό ποσοστό αύξησης του παραθύρου συμφόρησης του Venο είναι πιο αργό από αυτό του Reno.

3.3.3. ΔΙΚΤΥΟ ΠΕΙΡΑΜΑΤΟΣ

Σε αυτό το έγγραφο, χρησιμοποιούμε τον προσομοιωτή δικτύου (ns-2.26) [17] από Εθνικό Εργαστήριο Lawrence Berkley για να μελετήσουμε το TCP Venο. Το πειραματικό δίκτυο (δείτε το [σχήμα 10](#)) αποτελείται από δύο ζευγάρια πηγών και προορισμών. Οι συνδέσεις ζευγαριού μοιράζονται την ίδια διασύνδεση δυσχέρειας χρησιμοποιώντας μια σειρά αναμονής *droptail*. Η ταχύτητα κάθε διασύνδεσης για τη σύνδεση των κόμβων με τους ανάλογους δρομολογητές είναι 10Mbps με 1ms καθυστέρηση διάδοσης. Η δυσχερής διασύνδεση που ενώνει τους δρομολογητές A και B έχει μ πακέτα/s και τ δευτερόλεπτα καθυστέρηση διάδοσης, και το μέγεθος της σειράς αναμονής κατά μήκος αυτής της διασύνδεσης είναι B πακέτα και για τις δύο κατευθύνσεις. Το τυπικό μέγεθος πακέτου δεδομένων είναι 1kBytes και το τυπικό μέγεθος ack είναι 40 Bytes.

Όπως στο [8], καθορίζουμε την χωρητικότητα της διαδρομής να είναι ο μέγιστος αριθμός των πακέτων (πακέτα δεδομένων και πακέτα βεβαίωσης λήψης) που εκκρεμούν κατά μήκος αυτής της πορείας. Αυτά τα εκκρεμή πακέτα - στοιχείων ή acks - θα μπορούσαν να διαδοθούν κατά μήκος της γραμμής μετάδοσης, ή να περιμένουν στη σειρά στο buffer του δρομολογητή δυσχέρειας. Αναφερόμενοι στο [σχήμα 10](#), το μέγεθος αγωγού (C) μεταξύ μιας πηγής και ενός προορισμού είναι ίσοι με $B + 2P$, όπου $P = \mu \cdot (\tau + 1ms + 1ms) \square \mu \cdot \tau$.



Σχήμα 10. Πείραμα τοπολογίας δικτύου.

Υποθέτοντας ότι η πηγή έχει άπειρα δεδομένα για να στείλει, και το δημοσιευμένο παράθυρο του δέκτη έχει τεθεί ως αρκετά μεγάλο, κατά συνέπεια, εναλλακτικά χρησιμοποιούμε το wnd (παράθυρο) και $cwnd$ (παράθυρο συμφόρησης) στο ακόλουθο τμήμα.

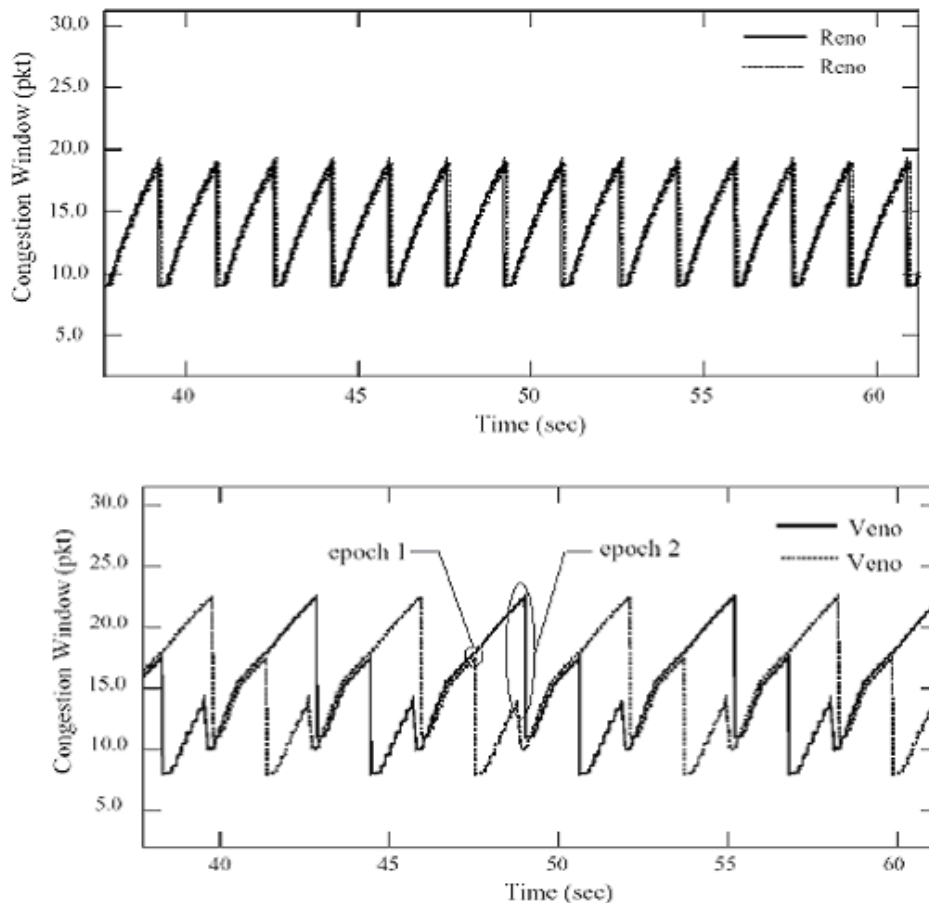
3.3.4. ΠΕΙΡΑΜΑΤΙΚΑ ΑΠΟΤΕΛΕΣΜΑΤΑ ΚΑΙ ΣΥΖΗΤΗΣΕΙΣ

Αυτή η παράγραφος πραγματεύεται μερικά ενδιαφέροντα αποτελέσματα για τη δυναμική συμπεριφορά του TCP Veno και την συνύπαρξή του με το TCP Reno με κυριότερα σημεία τους ρυθμούς συγχρονισμού της *έξω-από-τη φάση* που παρατηρούνται στο TCP Veno. Επιπλέον δείχνουμε το ο συγχρονισμός *έξω-από-τη φάση* οφελεί τη συμπεριφορά του Veno.

A. Η Δυναμική του TCP Veno και η αλληλεπίδρασή του με το Reno TCP

Δύο συνδέσεις TCP Reno ή Veno διευθύνονται μεταξύ των ζευγαριών των πηγών και των προορισμών στο σχήμα 1 με διαμόρφωση του $B = 15$, $\mu = 1.6$ Mbps

και $\tau = 50\text{ms}$. Όπως πραγματεύτηκε σε [7-8], δύο Reno έχουν τους ρυθμούς συγχρονισμού *μέσα-στη φάση*, δηλαδή, όταν μια σύνδεση ρίχνει το παράθυρό της στο μισό, μια άλλη σύνδεση θα ακολουθήσει την ίδια ποινική ρήτρα παραθύρου αμέσως. Το [σχήμα 11\(a\)](#) παρουσιάζει την περίπτωση δύο συνδέσεων TCP Reno. Το φαινόμενο του *μέσα-στη φάση* συγχρονισμού συμβαίνει κυρίως επειδή όταν χρησιμοποιείται πλήρως η κοινή σύνδεση, οι δύο TCP Reno αποστολείς συνεχίζουν να αυξάνουν το *cwnd* προκαλώντας υπερχειλίση και ένα πακέτο από κάθε σύνδεση θα πέσει κατά τη διάρκεια αυτής της *εποχής* συμφόρησης. Μετά από τότε, τα μεγέθη παραθύρων και των δύο συνδέσεων θα διχοτομηθούν.



Σχήμα 11. (α) Εξέλιξη παραθύρου σε δύο συνδέσεις Reno. (β) Εξέλιξη παραθύρου σε δύο συνδέσεις Veno.

Το Veno καθαρίζει την Πολλαπλασιαστική Μείωση στο Reno για να βελτιώσει την απόδοση στα ασύρματα περιβάλλοντα. Αυτό ο καθαρισμός παράγει

επίσης ένα ενδιαφέρον φαινόμενο – τον έξω-από-τη φάση συγχρονισμό. Μπορεί να γίνει κατανοητό ως εξής. Στο [σχήμα 11\(b\)](#), οι δύο συνδέσεις Venο, κατά την παραμονή τους στη κατάσταση συμφόρησης, παίρνουν πιο συντηρητική αύξηση παραθύρου από το Reno, δηλαδή, κάθε *cwnd* αυξάνεται κατά ένα πακέτο για κάθε δύο χρόνους στρογγυλού ταξιδιού αντί για κάθε ένα χρόνο στρογγυλού ταξιδιού. Στην «εποχή 1», όταν η διασύνδεση χρησιμοποιηθεί πλήρως, η μια σύνδεση αυξάνει το παράθυρό της κατά ένα πακέτο ενώ η άλλη όχι. Αυτό αναγκάζει μόνο ένα πακέτο να πέσει από το δρομολογητή και ως εκ τούτου μόνο μια σύνδεση υφίσταται την ποινική ρήτρα διχοτόμησης παραθύρου. Η ποινική ρήτρα διχοτόμησης παραθύρου έχει τότε απελευθερώσει πολύ από το εύρος ζώνης για μια άλλη σύνδεση για να συνεχίσει στην αρπαγή περισσότερου εύρους ζώνης με τη συντηρητική αύξηση παραθύρου.

Εντούτοις, όταν και οι δύο συνδέσεις Venο αυξάνουν τα μεγέθη παραθύρων τους κατά τη διάρκεια του ίδιου χρόνου στρογγυλού ταξιδιού όταν η διασύνδεση είναι πλήρως χρησιμοποιημένη, θα υποστούν πτώσεις πακέτων και θα μειώσουν τα μεγέθη παραθύρων τους και οι δύο συνδέσεις λόγω της συμφόρησης δικτύου. Αυτό απεικονίζεται στο [σχήμα 11\(b\)](#) στην «εποχή 2». Είναι επίσης αξιοσημείωτο στην «εποχή 2» που η σύνδεση με ένα χαμηλότερο μέγεθος παραθύρου μειώνεται λιγότερο από αυτή της σύνδεσης με έναν υψηλότερο μέγεθος παραθύρων. Ο λόγος είναι ότι η σύνδεση με το χαμηλό μέγεθος παραθύρου εξελίσσεται σε κατάσταση μη-συμφόρησης (με αύξηση ενός πακέτου για ένα χρόνο στρογγυλού ταξιδιού), ενώ άλλος παραμένει στην κατάσταση συμφόρησης².

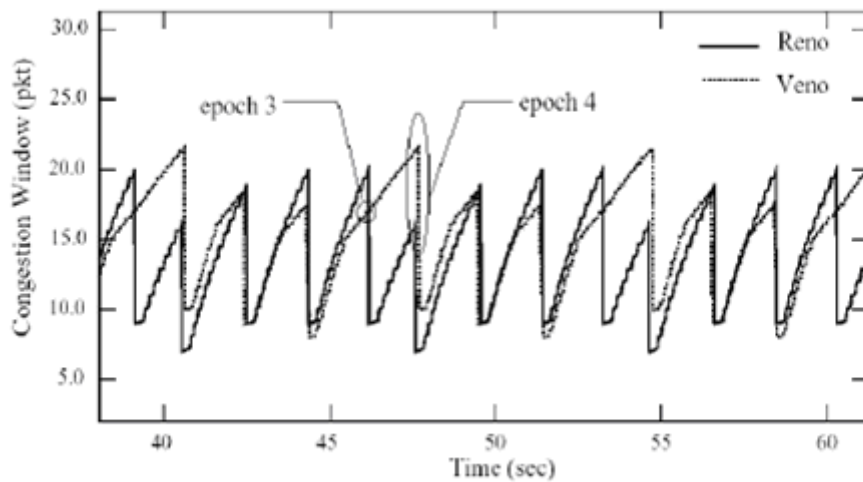
² Στο Venο, το παράθυρο εκτελεί μόνο τη μισή μείωση σε κατάσταση συμφόρησης όταν η απώλεια πακέτου ανιχνεύεται από τρία αναπαραχθέντα ACK, διαφορετικά, αυτό θα υποστεί μείωση παραθύρου μόνο κατά 1/5. Αυτή η απώλεια πακέτων εδώ, ενώ προκαλείται από την υπερχείλιση του *buffer*, μπορεί να θεωρηθεί ως συμφόρηση παροδικής απώλειας, όπως συζητείται στο [\[19\]](#).

Σημειώνεται εδώ ότι ενώ δύο οι συνδέσεις ανταγωνίζονται για τους πόρους δικτύου, η σύνδεση που καταλαμβάνει περισσότερο πόρο, θα υποστεί την αυστηρή ποινή και η σύνδεση που καταλαμβάνει το λιγότερο πόρο, θα έχει λιγότερο αυστηρή μείωση παραθύρου. Κατά συνέπεια, η λιγότερο επιθετική σύνδεση έχει μια ευκαιρία να αρπάξει περισσότερο πόρο όταν εμφανίζεται η απώλεια, εν τω μεταξύ, η επιθετικότερη σύνδεση θα παρατήρει μερικό πόρο που είχε καταλάβει.

Πιο κατά τρόπο ενδιαφέροντα, και οι δύο συνδέσεις ανταλλάσσουν τους ρόλους τους του να υπάρχει ένα περισσότερο και ένα λιγότερο επιτιθέμενο από άποψη χρήσης του εύρους ζώνης καθώς εξελίσσεται το *cwnd*, το οποίο είναι μοναδικό στο TCP Veno. Αυτό ο μηχανισμός θα επιφέρει σίγουρα το δίκαιο ανταγωνισμό για περιορισμένο πόρο στο Διαδίκτυο. Βλέποντας την εποχή συμφόρησης 1 και 2, αυτό το *συνυφασμένο* σχέδιο επαναλαμβάνεται επ' αόριστο, το σχήμα περιέχει μόνο μια ή μερικές από αυτές τις επαναλαμβανόμενες περιόδους για να επιτρέψει στον αναγνώστη να δει τις λεπτομέρειες των εξελίξεων του παραθύρου.

Μελετάται επίσης η περίπτωση ενός Reno και ενός Veno που μοιράζονται μια διασύνδεση όπως δείχνεται στο [σχήμα 10](#). Η εξέλιξη παραθύρου και των δύο συνδέσεων παρουσιάζονται στο σχήμα 5, με τα αποδεικτικά στοιχεία του *έξω-από-τη φάση* φαινόμενου συγχρονισμού του Veno. Τα περιστατικά στην «εποχή 3» και «εποχή 4» στο [σχήμα 12](#) αντιστοιχούν σε αυτά της «εποχής 1» και «εποχής 2» στο [σχήμα 11\(b\)](#). Σε αυτήν την περίπτωση, η σύνδεση Reno είναι αυτή που πάντα είναι η επιθετικότερη σύνδεση στο μοίρασμα του εύρους ζώνης.

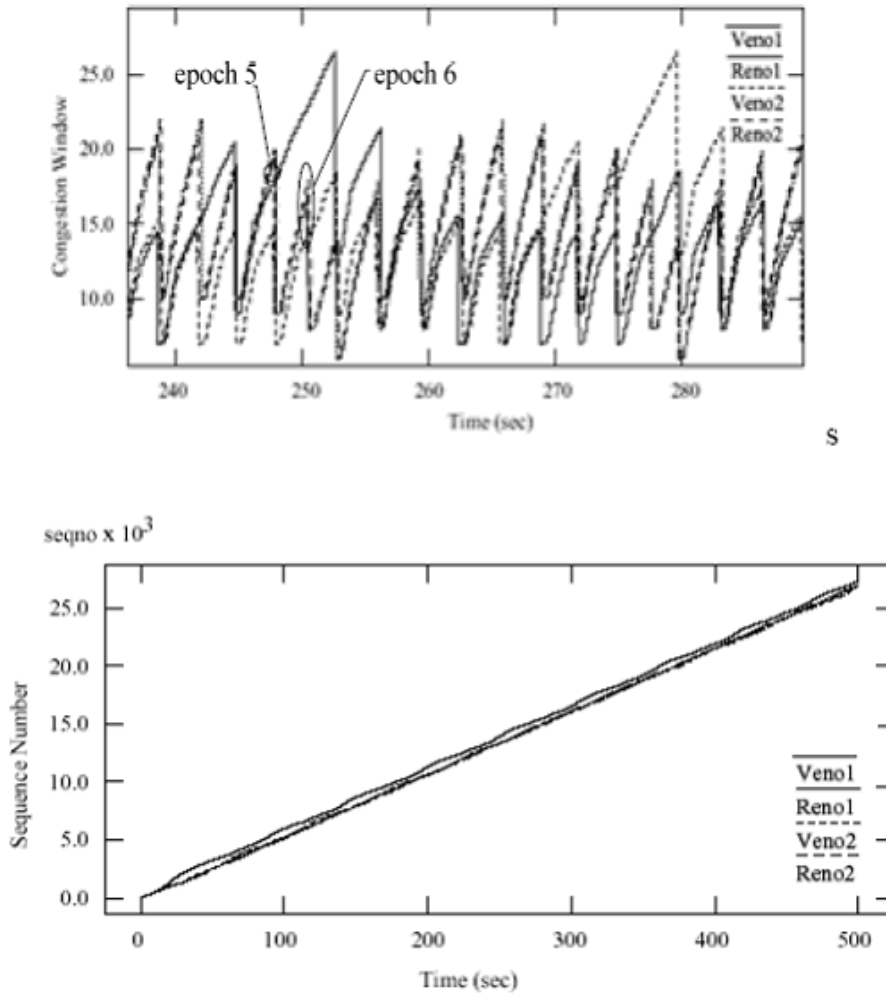
Η μοιρασιά του Reno και Veno εμφανίζει ένα ενδιαφέρον αποτέλεσμα που δείχνει την εγκυρότητα στο μοίρασμα του εύρους ζώνης μεταξύ Reno και Veno. Συνήθως, ένα λιγότερο επιτιθέμενο στο μοίρασμα του εύρους ζώνης, σαν το TCP Vegas, κατέχει λιγότερο εύρος ζώνης κατά μία μακροπρόθεσμη λειτουργία όταν μοιράζεται μια διασύνδεση με το επιθετικότερο Reno. Εντούτοις, το Veno, μολονότι είναι το λιγότερο επιτιθέμενο, είναι σε θέση να μοιραστεί το εύρος ζώνης δίκαια με το Reno. Βλέποντας το [σχήμα 12](#), η συνεχόμενη γραμμή είναι εξέλιξη του Reno, και η διακεκομμένη γραμμή είναι της σύνδεση Veno. Ας προσέξουμε στους δύο χαρακτηριστικούς χρόνους που η απώλεια πακέτου εμφανίζεται - «εποχή 3» και «εποχή 4».



Σχήμα 12. Εξέλιξη παραθύρου σε Reno και Veno.

Στην «εποχή 3», η σύνδεση Reno αυξάνει το παράθυρό της κατά ένα πακέτο ενώ η σύνδεση Veno όχι (το Veno, αφού μπει στην κατάσταση συμφόρησης, δεν αυξάνει πάντα το *cwnd* του μετά από το ένα χρόνο στρογγυλού ταξιδιού, το παράθυρό του αυξάνεται κατά ένα πακέτο για κάθε άλλο χρόνο στρογγυλού ταξιδιού στην κατάσταση συμφόρησης). Εξαιτίας του γεμάτου buffer (ήδη κατειλημμένος από αυτές τις δύο συνδέσεις) του δυσχερούς δρομολογητή, ένα πρόσφατα πρόσθετο πακέτο στο παράθυρο του Reno θα οδηγήσει σε μια απώλεια πακέτου, ενώ το Veno, λόγω του αμετάβλητου παραθύρου του, αποφεύγει την απώλεια πακέτου. Προφανώς, αυτή τη στιγμή, η επιθετική πολιτική παραθύρου του Reno επιφέρει μια περικοπή παραθύρου (αμεσότητα που διχотоμείται από την ανίχνευση αυτής της απώλειας πακέτου) ενώ το Veno επιτρέπεται να αυξηθεί και να λάβει περισσότερο πόρο εύρους ζώνης μέχρι την «εποχή 4», όπου και το Reno και το Veno υφίστανται μια απώλεια πακέτου λόγω της ταυτόχρονης αύξησης των παραθύρων τους αυτή τη στιγμή. Κατά συνέπεια, και τα δύο παράθυρα κόβονται στα μισά. Εν τούτοις, στην ακόλουθη φάση εξέλιξης παραθύρου (από 47.5s έως 53s), το Reno βαθμιαία υπερβαίνει το παράθυρο του Veno και αρπάζει περισσότερο πόρο του εύρους ζώνης λόγω της λίγο επιθετικότητάς του. Στο χρόνο 53s, αυτό θα επαναλάβει την περίπτωση της «εποχής 3». Αυτό το *συνυφασμένο* σχέδιο επαναλαμβάνεται επ' αόριστο. Κατά συνέπεια, κατά μία μακροπρόθεσμη λειτουργία, η εγκυρότητα-δικαιότητα μεταξύ του Veno και Reno είναι σε θέση να διατηρηθεί. Παρατηρήθηκε ότι μέσα σε αυτές τις δύο ανταγωνιστικές συνδέσεις, το Veno έχει υποστεί λιγότερη απώλεια πακέτου λόγω συμφόρησης (περίπου 9 φορές στο [σχήμα 12](#)), ενώ το Reno έχει περισσότερη (περίπου 13 φορές). Αυτός ο μειωμένος αριθμός συμφόρησης σίγουρα αυξάνει κάπως τη χρησιμοποίηση του εύρους ζώνης.

Ένα πιο ενδιαφέρον σενάριο των πολλαπλών συνδέσεων TCP με δύο Venο και δύο Reno συνδέσεων δυναμικής εξέλιξης παραθύρου παρουσιάζονται στο [σχήμα 13\(a\)](#). Με τις ρυθμίσεις πειράματος $B = 44$, $\mu = 1.8\text{Mbps}$ και $\tau = 50\text{ms}$ με την τοπολογία δικτύου που περιγράφηκε στο [σχήμα 10](#).



Σχήμα 13. (α) Εξέλιξη παραθύρου σε συνδέσεις Reno και Venο.

(β) Αύξων αριθμός τεσσάρων συνδέσεων.

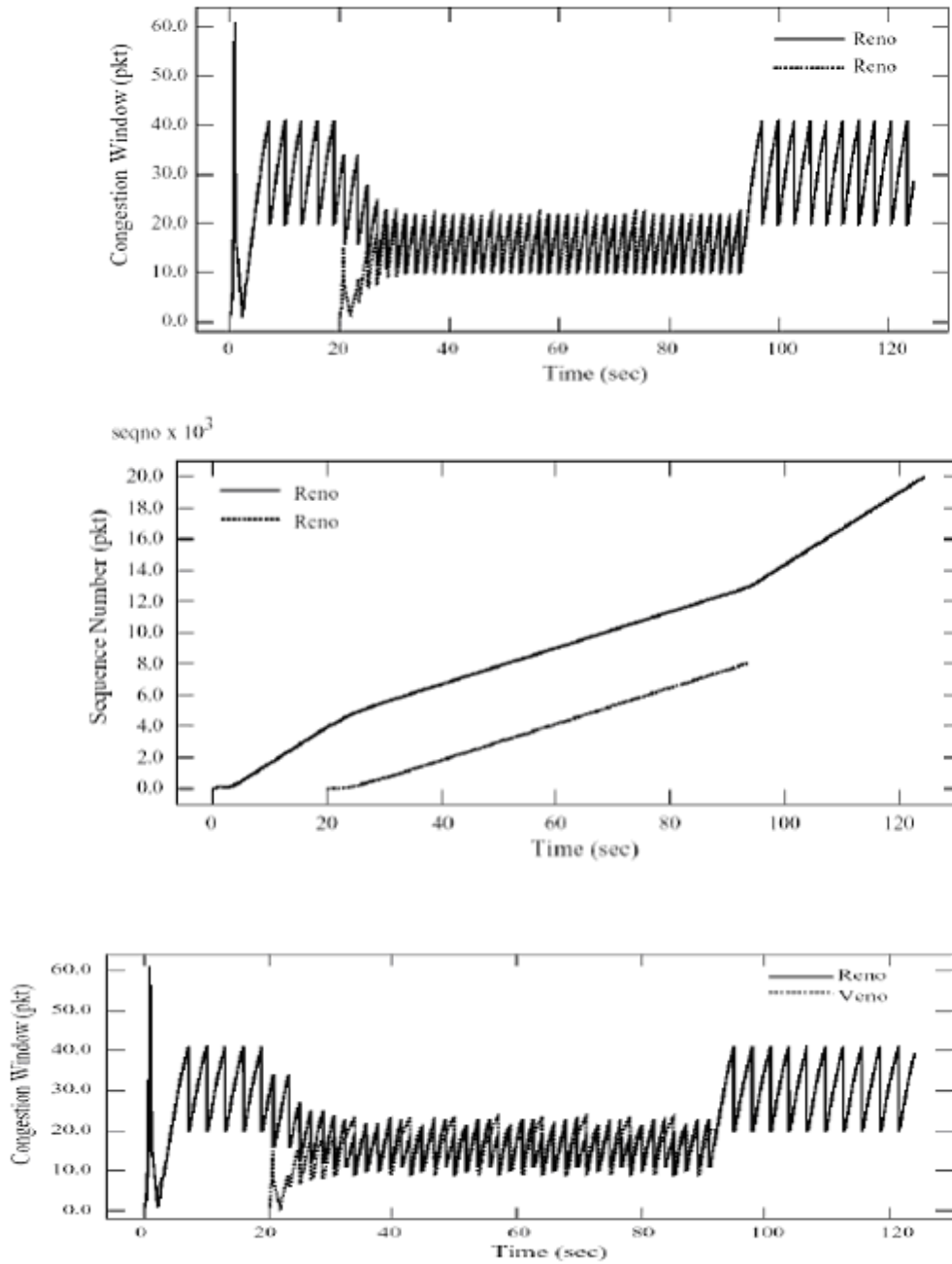
Τα γεγονότα στην «εποχή 5» και την «εποχή 6» είναι παρόμοια με αυτά από την «εποχή 4» στο σχήμα 5. Στην «εποχή 5» μόνο το Venο 1 δεν εκτίθεται σε απώλεια, στην «εποχή 6» οι Venο 1 και Venο 2 συνδέσεις δεν εκτίθενται σε απώλεια. Ακολουθώντας μια σειρά εμφάνισης συγχρονισμού συμφορήσεων (όλες οι Reno και Venο συνδέσεις ρίχνουν ένα πακέτο ακριβώς σε κάθε εποχή), οι συνδέσεις Reno παρουσιάζουν περισσότερη επιθετικότητα σε αυτούς τους ακόλουθους κύκλους. Μετά από αυτό, το *συνυφασμένο* σχέδιο ανακύπτει με τη Venο 2 σύνδεση.

Παρατηρούμε ότι η εξέλιξη παραθύρου γίνεται πιο περίπλοκη ενώ υπάρχουν πολλαπλές συνδέσεις, ειδικά όταν εισάγεται η κυκλοφορία υποβάθρου, αλλά το *έξω-από-τη φάση* φαινόμενο υπάρχει ακόμα κάπως. Το [σχήμα 13](#) δίνει την εξέλιξη των αριθμών ακολουθίας εναντίον του χρόνου και το διάγραμμα δείχνει ότι και οι τέσσερις συνδέσεις μοιράζονται το εύρος ζώνης της δυσχερής διασύνδεσης εξίσου χωρίς προκατάληψη.

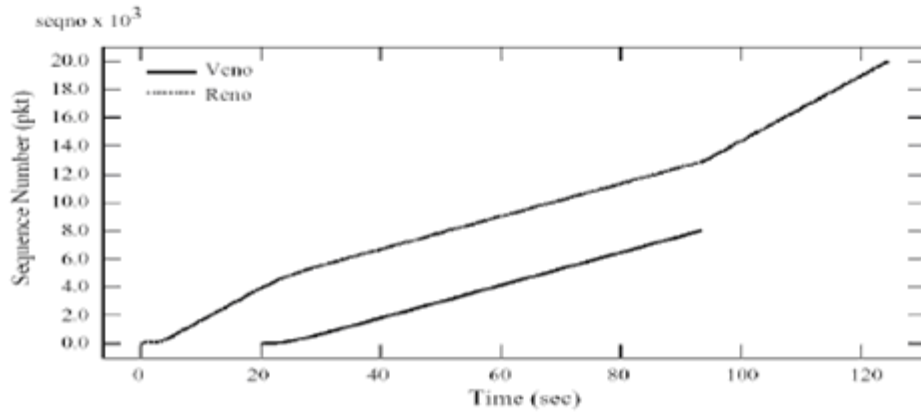
B. Εγκυρότητα(Δικαιότητα) του TCP Reno και Veno

Το έγγραφο [\[10\]](#) έχει μελετήσει ότι το TCP Veno μπορεί να επιτύχει υψηλή βελτίωση ρυθμοαπόδοσης πέρα του Reno στα ασύρματα δίκτυα. Σε αυτή τη παράγραφο, ερευνάμε τα ζητήματα δυναμικής της εγκυρότητας του TCP Reno και Veno πάνω σε ενσύρματα δίκτυα.

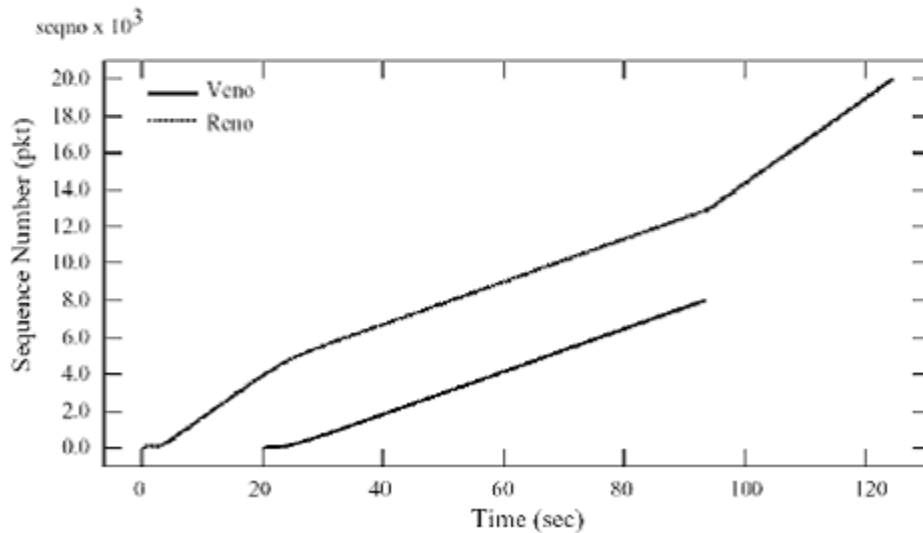
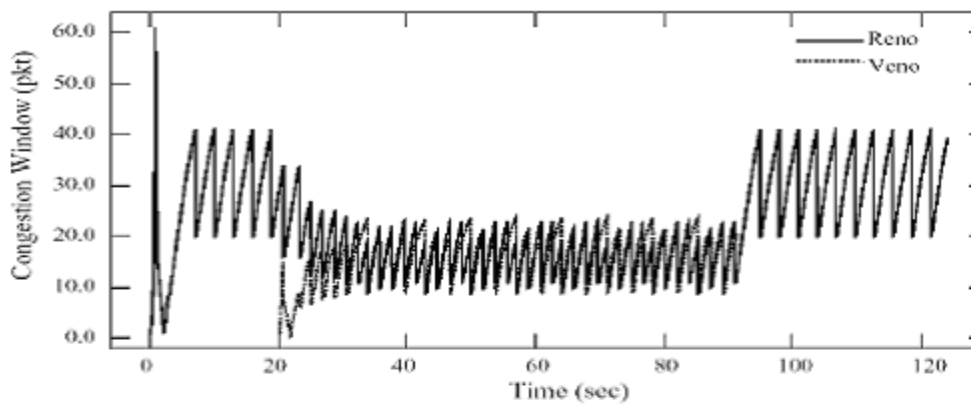
Όπως παρατηρήσαμε, δύο συνδέσεις Reno πάντα συγχρονίζονταν σε *μέσα-στη φάση* ρυθμούς. Ενώ μια σύνδεση Veno συνυπάρχει με μια σύνδεση Reno, υπάρχουν μερικές περιπτώσεις όπου το Veno δεν εκτίθεται σε απώλεια πακέτου σε συγκεκριμένη *εποχή*. Σε μερικό βαθμό, αυτή η συμπεριφορά θα αποφύγει κάποια απώλεια συμφόρησης, και μπορεί να βελτιώσει τη χρησιμοποίηση της διασύνδεσης δικτύου. Σχεδιάζουμε το πείραμά μας ως εξής: είτε η Reno είτε η σύνδεση Veno ξεκινάει πρώτη να διαβιβάζει τα αρχεία 20Mbyte, η άλλη σύνδεση - Reno ή Veno - συμμετέχει στα 20s και αρχίζει να μεταφέρει αρχεία 8Mbyte. Η διάταξη της τοπολογίας δικτύου είναι: ποσοστό μετάδοσης 2Mbps, μια μετ'επιστροφής καθυστέρηση 100ms και το μέγεθος buffer της δυσχερής διασύνδεσης είναι 15. Παριστάνουμε γραφικά το παράθυρο συμφόρησης καθώς επίσης και τον αριθμό ακολουθίας ενάντια στο χρόνο στα [σχήματα 14](#), [15](#) και [16](#). Από τρία διαφορετικά σενάρια παρατηρούμε ότι η σύνδεση που ξεκίνησε αργότερα δεν θα προκαταλαμβάνονταν από την προηγούμενη σε λειτουργία σύνδεση, θα μοιραστούν το εύρος ζώνης ομοιόμορφα, αυτό φαίνεται επίσης σε κάθε διάγραμμα αριθμού ακολουθίας εναντίον του χρόνου, όπου η κλίση των δύο συνδέσεων είναι σχεδόν ίδια.



Σχήμα 14. Μια σύνδεση Reno ξεκίνησε πρώτη, μια άλλη Reno συνδέεται στα 20sec.



Σχήμα 15. Μια σύνδεση Veno ξεκίνησε πρώτη, μια άλλη Reno συνδέεται στα 20sec.



Σχήμα 16. Μια σύνδεση Reno ξεκίνησε πρώτη, μια άλλη Veno συνδέεται στα 20sec.

Στον ακόλουθο πίνακα 1, υπολογίζουμε τη ρυθμοαπόδοσης χρησιμοποιώντας το μέγεθος αρχείων συν τα συνολικά αναμεταδιδόμενα bytes δια του χρόνου σε δευτερόλεπτα. Από τον πίνακα, πήραμε τρία αποτελέσματα, 1) αναφορικά με τη

μακρύτερη μετάδοση πάνω σε καλωδίωση, το Veno παίρνει ελαφρώς υψηλότερη ρυθμοαπόδοση ανεξάρτητα από οτιδήποτε συνδυασμοί και αν υιοθετούνται 2) όσον αφορά στη σύντομη μετάδοση, το Veno επίσης αποκτά ελαφρώς υψηλότερη ρυθμοαπόδοση 3) αναφερόμενοι στη 3^η σειρά αυτού του πίνακα, ξέρουμε ότι αυτή η ελαφριά βελτίωση επέρχεται από λιγότερα αναμεταδιδόμενα πακέτα, με άλλα λόγια, οι απώλειες συμφόρησης στο Veno μειώνονται σε σύγκριση με του Reno.

3.4. Σύγκριση Απόδοσης Ανάμεσα σε TCP Reno και TCP Vegas

3.4.1 Γενική Περιγραφή

Μετά από την ανάπτυξη του TCP Vegas, η σύγκριση απόδοσης μεταξύ του TCP Reno και TCP Vegas δεν έχει συζητηθεί λεπτομερώς. Η συζήτηση της αναθεωρημένης έκδοσης TCP, TCP Vegas, παραμένει ανεπαρκής για να αποφασίσουμε εάν πρέπει να τη χρησιμοποιήσουμε ή όχι. Αυτό το έγγραφο προσπαθεί να συγκρίνει την απόδοση, ρυθμοαπόδοση και εγκυρότητα, των Reno και Vegas στα περιβάλλοντα δικτύου με ομοιογενείς και ετερογενείς εκδόσεις του TCP. Τα αποτελέσματα σε αυτήν την μελέτη δείχνουν ότι ενώ το TCP Vegas λαμβάνει καλύτερη ρυθμοαπόδοση και εγκυρότητα σε ομοιογενείς περιπτώσεις, αυτό αποτυγχάνει να ξεπεράσει το Reno σε ετερογενείς περιπτώσεις. Αυτό το φαινόμενο αποτρέπει τους χρήστες του TCP Vegas να το υιοθετήσουν παρά την καλύτερη απόδοσή του.

Το TCP είναι το πιο διαδεδομένο πρωτόκολλο από τότε που έφθασε το Διαδίκτυο (Internet). Οι νέες εκδόσεις του TCP ήταν προτεινόμενες για να βελτιώσουν την απόδοση μετάδοσης δεδομένων. Η πρώτη έκδοση του TCP, που τυποποιήθηκε σε RFC 793, καθόρισε τη βασική δομή του TCP, δηλαδή το σχέδιο ελέγχου ροής βασισμένο σε μέγεθος παραθύρου και ένα τραχύ χρονόμετρο διαλείμματος. Η δεύτερη έκδοση, TCP Tahoe, πρόσθεσε το σχέδιο αποφυγής συμφόρησης προτεινόμενο από τον Van Jacobson [20]. Η τρίτη έκδοση, TCP Reno, που ακολούθησε δύο έτη αργότερα [21]. Αυτή η έκδοση επέκτεινε το σχέδιο ελέγχου συμφόρησης με το να περιλάβει γρήγορη αναμετάδοση και γρήγορα σχέδια αποκατάστασης. Το TCP Reno έχει γίνει η δημοφιλέστερη έκδοση σήμερα. Εντούτοις αυτή η έκδοση προβλέπει το διαθέσιμο εύρος ζώνης ανιχνεύοντας την απώλεια

πακέτων, που προκαλεί τη συμφόρηση δικτύων και την άδικη χρήση εύρους ζώνης. Οι Brakmo και Peterson [22] πρότειναν ένα νέο αλγόριθμο ελέγχου συμφόρησης που μεταγλώττισε το TCP Vegas με την τροποποίηση του σχεδίου αποφυγής συμφόρησης του TCP Reno. Αυτή η έρευνά τους, ανέφερε ότι το TCP Vegas πέτυχε τη 37%-71% καλύτερη ρυθμοαπόδοση απ' ό,τι το TCP Reno.

Εκτός από την καλύτερη ρυθμοαπόδοση, το πιο δίκαιο μερίδιο εύρους ζώνης είναι ένα άλλο πλεονέκτημα του TCP Vegas. Πολλές μελέτες έχουν παραχωρήσει το δικαίωμα στη TCP Reno σύνδεση ότι μια πιο μακροχρόνια καθυστέρηση διάδοσης παίρνει λιγότερο εύρος ζώνης από τη σύνδεση TCP Reno με μια πιο σύντομη καθυστέρηση [23, 24]. Αυτή η αδικία μειώνεται εντυπωσιακά όταν η έκδοση του TCP αλλάζει από Reno σε Vegas.

Με την πρώτη ματιά, φαίνεται ότι η κυριαρχία του TCP Vegas πρέπει να αντικαταστήσει το TCP Reno. Εντούτοις, αυτή η αλλαγή ακόμα δεν έχει συμβεί. Ο κύριος λόγος για τον οποίο το Reno παραμένει σε λειτουργία είναι λόγω του προβλήματος του αδικίου όταν οι συνδέσεις TCP Reno και Vegas υπάρχουν μαζί [23, 24]. Το TCP Reno είναι ένα επιθετικό σχέδιο ελέγχου το οποίο επεκτείνει περισσότερο το εύρος ζώνης του μέχρι τα διαβιβασθέντα πακέτα να χαθούν. Εν τω μεταξύ, το TCP Vegas είναι ένα συντηρητικό σχέδιο το οποίο απαιτεί ένα κατάλληλο εύρος ζώνης για κάθε σύνδεση. Ως εκ τούτου, όταν οι συνδέσεις TCP Reno και TCP Vegas υπάρχουν μαζί, οι συνδέσεις TCP Reno θα πάρουν εύρος ζώνης από τις συνδέσεις TCP Vegas.

3.4.2 Περιγραφή Reno και Vegas

Το TCP χρησιμοποιεί την ανασύζευξη από τον προορισμό για να υπολογίσει το διαθέσιμο εύρος ζώνης. Τα Reno και Vegas διαφέρουν κυρίως στο πως υπολογίζουν το εύρος ζώνης. Το Reno μεταχειρίζεται την απώλεια πακέτων ως ένδειξη της συμφόρησης δικτύου. Η πηγή κυκλοφορίας ξέρει ότι καταχράται το εύρος ζώνης όταν τα διαβιβασθέντα πακέτα χάνονται. Το Vegas υπολογίζει τις αναμενόμενες και τις πραγματικές πηγαίες τιμές βασισμένο στο χρόνο κάθε πακέτου για ένα πλήρες ταξίδι επιστροφής και ελέγχει το μέγεθος παραθύρου χρησιμοποιώντας τη διαφορά μεταξύ αναμενόμενων και πραγματικών τιμών. Αυτή η

ενότητα περιγράφει αρχικά όλο το σχέδιο ελέγχου του Reno [25] και έπειτα, παρουσιάζει τα μέρη Vegas που ενισχύουν αυτό το σχέδιο [22].

Reno: Μια σύνδεση TCP Reno περιλαμβάνει τρεις πρωταγωνιστές: τον αποστολέα, το δρομολογητή και το δέκτη. Ένας δρομολογητής προωθεί τα πακέτα δεδομένων. Ένας δέκτης TCP επιστρέφει την παραχθείσα πληροφορία αναγνώρισης (ACK) των πακέτων δεδομένων στον αποστολέα. Εάν οποιαδήποτε πακέτα εκτός λειτουργίας παραληφθούν, ο δέκτης θα επιστρέψει ένα πανομοιότυπο ACK στον αποστολέα. Ένας TCP αποστολέας ελέγχει τη ροή των πακέτων δεδομένων με τέσσερις παραμέτρους: (1) *cwnd* (congestion window: παράθυρο συμφόρησης), τη ποσότητα πακέτων που μπορεί να σταλεί (2) *mwnd* (maximum window), τη μέγιστη αξία του *cwnd*, (3) *ssthresh* (slow start threshold), τη μονάδα για την παύση της κατάστασης “slow start” (αργής εκκίνησης) και την έναρξη της κατάστασης “congestion avoidance” (αποφυγής συμφόρησης) και (4) *coarse grain timer granularity*. Η προκαθορισμένη αξία για αυτό το χρονόμετρο είναι 500ms, δηλαδή για κάθε 500ms, ο αποστολέας θα ελέγχει εάν έχει γίνει διάλειμμα. Εάν έχει γίνει, ο αποστολέας θα επαναρυθμίσει το *cwnd* στο 1 και θα ξανά ξεκινήσει τη κατάσταση αργής εκκίνησης. Ο αποστολέας TCP χρησιμοποιεί αυτές τις τέσσερις παραμέτρους για να επιτύχει τα ακόλουθα τέσσερα επίπεδα του σχεδίου ελέγχου ροής.

(1) Slow Start (αργή εκκίνηση): Μετά από μια δημιουργία σύνδεσης ή ένα διάλειμμα, η κατάσταση αργής εκκίνησης αρχίζει. Η αρχική τιμή του *cwnd* οριοθετείται στο ένα στην αρχή αυτής της κατάστασης και κατά συνέπεια, μόνο ένα πακέτο μπορεί να σταλεί. Όταν ο αποστολέας λαμβάνει το ACK αυτού του πακέτου, το *cwnd* αυξάνεται κατά ένα, και ο αποστολέας μπορεί έπειτα να στείλει δύο πακέτα. Το *cwnd* αυξάνει κατά ένα κάθε φορά που ο αποστολέας λαμβάνει ένα ACK. Όταν το *cwnd* γίνεται ίσο με το *ssthresh*, η κατάσταση αποφυγής συμφόρησης αρχίζει.

(2) Congestion Avoidance (αποφυγή συμφόρησης): Αφότου το μέγεθος παραθύρου στην κατάσταση αργής εκκίνησης επεκτείνεται εκθετικά, η αποστολή των πακέτων γρήγορα θα συμφορήσει το δίκτυο. Καθώς το *cwnd* υπερβαίνει το *ssthresh*, το *cwnd* αυξάνεται κατά $1/cwnd$ για κάθε ACK που παραλαμβάνεται. Το μέγεθος παραθύρου θα αυξηθεί γραμμικά στην κατάσταση αποφυγής συμφόρησης.

(3) Fast Retransmission (γρήγορη αναμετάδοση): Ο διπλασιασμός του ACK μπορεί να είναι προκαλούμενος από ένα χαμένο πακέτο ή από μία επαναπαραγγελία

πακέτων. Εάν τρία ή περισσότερα διπλασιασμένα *ACKs* παραληφτούν στη σειρά, η απώλεια πακέτων είναι πολύ πιθανή. Το TCP εκτελεί την αναμετάδοση αμέσως χωρίς να περιμένει για ένα διάλειμμα αναμετάδοσης.

(4) Fast Recovery (γρήγορη ανάκτηση): Όταν το τρίτο διπλασιασμένο *ACK* λαμβάνεται, το *ssthresh* τίθεται κατά το ήμισυ του *cwnd* και έπειτα το *cwnd* τίθεται στη τιμή του *ssthresh* συν τρεις φορές το μέγεθος του πακέτου. Το *cwnd* προστίθεται από το μέγεθος του πακέτου σε κάθε λήψη ενός διπλασιασμένου *ACK*. Όταν το *ACK* του αναμεταδιδόμενου πακέτου παραλαμβάνεται, το *cwnd* τίθεται στη τιμή του *ssthresh*. Καθορισμένο εκ νέου, το *cwnd* τίθεται κατά το ήμισυ της προηγούμενης του τιμής.

Vegas: Το σχέδιο ελέγχου ροής του TCP Vegas είναι προερχόμενο από το Reno με τρεις τροποποιημένες τεχνικές. Το Vegas απαιτεί από τον αποστολέα να καταγράφει τον *round trip* χρόνο και τον χρόνο αποστολής του κάθε πακέτου. Ο ελάχιστος *round trip* χρόνος, *Basertt*, πρέπει επίσης να καταγραφεί.

(1) New Congestion Avoidance (νέα αποφυγή συμφόρησης): Όταν ο αποστολέας λαμβάνει ένα *ACK*, υπολογίζει τη διαφορά μεταξύ της αναμενόμενης ρυθμοαπόδοσης και της πραγματικής ρυθμοαπόδοσης. (*αναμενόμενη* = $cwnd / Basertt$, *πραγματική* = $\text{data bytes μεταξύ του χρόνου που το πακέτο στέλνεται και του ACK που λαμβάνεται} / \text{το διάστημα μεταξύ του χρόνου που το πακέτο στέλνεται και του ACK που λαμβάνεται}$). Το Vegas καθορίζει δύο κατώτατα όρια, α και β . Επίσης υπολογίζει $Diff = (\text{Αναμενόμενη} - \text{Πραγματική}) \times Basertt$. Το *cwnd* αυξάνεται γραμμικά εάν το $Diff < \alpha$ και μειώνεται γραμμικά εάν $Diff > \beta$. Η αξία των (α , β) προτείνεται για να είναι (1, 3) μέσα στο πακέτο. Η αξία *Basertt* ενημερώνεται εάν $Diff < 0$ αφού η *Αναμενόμενη* τιμή πρέπει να υπερβεί την *Πραγματική*. Η αξία των (α , β) επιτρέπει στην πηγή να ελέγξει την διαφορά μεταξύ των *Αναμενόμενων* και *Πραγματικών* ποσοστών από ένα σε τρία πακέτα ανά *round trip* χρόνο.

(2) New Retransmission (νέα αναμετάδοση): Όταν ο αποστολέας λαμβάνει ένα αναπαραχθέν *ACK*, ελέγχει εάν το διάστημα μεταξύ του τρέχοντος χρόνου και του χρόνου αποστολής συν το *Basertt* του σχετικού πακέτου υπερβαίνει της τιμής του διαλείμματος (timeout). Εάν το διάστημα υπερβαίνει αυτήν την τιμή, το Vegas αναμεταδίδει το χαμένο πακέτο χωρίς να αναμένει τα τρία αναπαραχθέντα *ACKs*.

Όταν το πρώτο ή το δεύτερο *ACK* φθάσει μετά την αναμετάδοση του τελευταίου πακέτου, το Vegas ελέγχει πάλι για διάλειμμα.

(3) Modified Slow Start (τροποποιημένη αργή εκκίνηση): Το Vegas χρησιμοποιεί το διάστημα ανάμεσα στα *ACKs* κατά τη διάρκεια της κατάστασης της αργής έναρξης για να εκτιμήσει, κρίνει το διαθέσιμο εύρος ζώνης. Το εκτιμημένο διαθέσιμο εύρος ζώνης χρησιμοποιείται για να θέσει το *ssthresh* κατάλληλα, το οποίο κάνει το Vegas λιγότερο πιθανό να υπερβεί το διαθέσιμο εύρος ζώνης.

3.4.3 Περιβάλλοντα προσομοίωσης

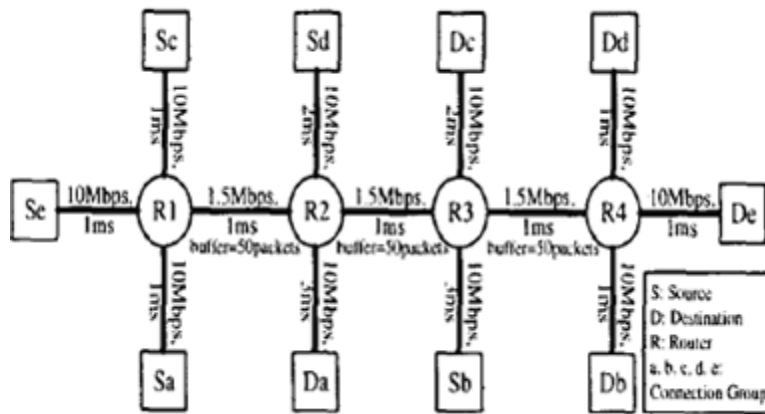
Ο προσομοιωτής δικτύων (ns) [26], όπως αναπτύχθηκε από το Lawrence Berkley Laboratory, χρησιμοποιείται σε αυτή την έρευνα για να “τρέξει” τις προσομοιώσεις μας. Αυτός ο προσομοιωτής έχει χρησιμοποιηθεί συχνά στις προηγούμενες εισηγητικές μελέτες του TCP [23, 24]. Ο [πίνακας 2](#) απαριθμεί τις παραμέτρους ελέγχου του TCP σε αυτό το έγγραφο.

Parameter	Value
timer granularity	500ms
ssthresh	64KB
mwnd	128KB
cwnd	1KB
α	1
β	3
Basertt	0

Πίνακας 2. Παράμετροι και αξίες που χρησιμοποιήθηκαν στην προσομοίωση.

Η διενεργηθείσα προσομοίωση εν τω παρόντι χρησιμοποίησε έναν απλό σχηματισμό τεσσάρων δρομολογητών, όπως φαίνεται στο [σχήμα 17](#), σαν τοπολογία των δικτύων μας, όπου S, D, και R δείχνουν “Source”, “Destination”, και “Router”, αντίστοιχα. Μερικές συνδέσεις TCP είναι ταξινομημένες σε πέντε ομάδες (a, b, c, d, e) σύμφωνα με τους δρομολογητές ταξιδιού τους. Επαναδιατυπωμένες, αυτές οι πέντε ομάδες συνδέσεων TCP διαβιβάζονται από το Sa στο Da, από το Sb στο Db, από το Sc στο Dc, από το Sd στο Dd, και από το Se στο De. Ειδικότερα, οι ομάδες a και b είναι συνδέσεις TCP που ταξιδεύουν μέσω δύο δρομολογητών, οι ομάδες c και d

είναι συνδέσεις TCP που ταξιδεύουν μέσω τριών δρομολογητών, και της ομάδας e είναι οι TCP συνδέσεις που ταξιδεύουν μέσω τεσσάρων δρομολογητών.



Σχήμα 17. Τοπολογία δικτύου.

Η σύνδεση μεταξύ δύο γειτονικών δρομολογητών έχει 1ms καθυστέρηση με 1.5Mbps δυνατότητα μεταφοράς δεδομένων. Το μέγεθος του buffer τίθεται σε 50 πακέτα, και κάθε πακέτο καθορίζεται σε ένα μήκος των 1000 bytes. Μερικές ομάδες έχουν μια πρόσθετη καθυστέρηση σύνδεσης τοποθετημένη ανάμεσα στους δρομολογητές και τις πηγές ή τους προορισμούς για να ισορροπήσει την καθυστέρηση διάδοσης για όλες τις ομάδες. Αυτό γίνεται για να εξαλειφτεί η αδικία που έχει προκληθεί από διαφορετικές καθυστερήσεις διάδοσης.

Αυτή η μελέτη μετρά τον αριθμό των *ACKs* και τη διαφορά τους για 50 δευτερόλεπτα FTP (File Transfer Protocol) μεταφέροντας στοιχεία σε κάθε σύνδεση. Ο μέσος αριθμός των *ACKs* μπορεί να αντιπροσωπεύσει τη ρυθμοαπόδοση και η διαφορά τους μπορεί να απεικονίσει το βαθμό αμεροληψίας. Τα αποτελέσματα της προσομοίωσης μπορούν να καθοδηγήσουν τους χρήστες στην επιλογή TCP Reno ή Vegas.

3.4.4 Αποτελέσματα Προσομοίωσης

Οι προσομοιώσεις διεξάγονται σε δύο διαφορετικά περιβάλλοντα, ομοιογενή και ετερογενή. Τα ομοιογενή περιβάλλοντα περιέχουν μια απλή έκδοση από τις συνδέσεις TCP, ενώ τα ετερογενή περιβάλλοντα περιέχουν διαφορετικές εκδόσεις

των TCP συνδέσεων. Τα αποτελέσματα αποσαφηνίζουν πως χρησιμοποιώντας μαζί το TCP Reno και Vegas επηρεάζουν την απόδοση. Τρεις περιπτώσεις εξετάζονται σε κάθε περιβάλλον.

Στην πρώτη περίπτωση του ομοιογενούς περιβάλλοντος, αυτή η εργασία χρησιμοποιεί την ομάδα e για να τρέξει 5 συνδέσεις TCP με διάφορα μεγέθη buffer να παρέχονται στους δρομολογητές. Στη δεύτερη περίπτωση, υπάρχουν διαφορετικοί αριθμοί από TCP συνδέσεις της ομάδας e με κάθε δρομολογητή να έχει μέγεθος buffer πενήντα πακέτων. Αυτές οι δύο περιπτώσεις αποκαλύπτουν την προϋπόθεση για αμεροληψία του TCP Vegas. Στην τρίτη περίπτωση, πέντε ομάδες χρησιμοποιούνται για να ερευνήσουμε την επίδραση των συνδέσεων που ταξιδεύουν σε διαφορετικούς αριθμούς δρομολογητών.

Στην πρώτη περίπτωση ετερογενούς περιβάλλοντος, αυτή η εργασία χρησιμοποιεί την ομάδα e για να τρέξει 5 TCP Reno και 5 TCP Vegas συνδέσεις μαζί, με διάφορα μεγέθη buffer να παρεχόμενα στους δρομολογητές. Στη δεύτερη περίπτωση, υπάρχουν διαφορετικοί αριθμοί συνδέσεων TCP Vegas της ομάδας e μέσα σε 10 TCP συνδέσεις, με κάθε δρομολογητή να έχει μέγεθος buffer πενήντα πακέτων. Οι συνδέσεις εκτός από TCP Vegas εκτελούν και το TCP Reno. Αυτή η περίπτωση διευκρινίζει τις επιδράσεις στα ποσοστά του TCP Vegas σε 10 TCP συνδέσεις. Στην τρίτη περίπτωση, χρησιμοποιούνται πέντε ομάδες και κάθε ομάδα εκτελεί τον ίδιο αριθμό TCP Reno και TCP Vegas συνδέσεων ταυτόχρονα.

3.4.4.1 Ομογενή περιβάλλοντα

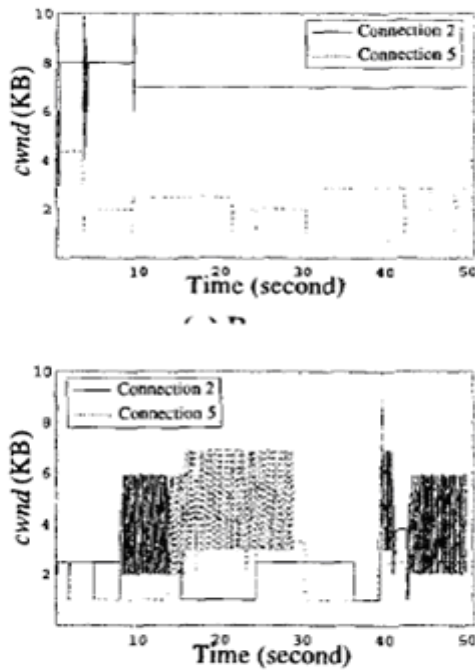
A. Μεγέθη Buffer

Σε αυτήν την περίπτωση, εστιάζουμε στον πως τα μεγέθη του buffer στους δρομολογητές επηρεάζουν την απόδοση του TCP Reno και Vegas. Αυτή η προσομοίωση χρησιμοποιεί την ομάδα e μόνο. Ο [πίνακας 3](#) παρουσιάζει αποτελέσματα της προσομοίωσης. Μερικές παρατηρήσεις αποκαλύπτουν τη διαφορά απόδοσης μεταξύ του TCP Reno και Vegas.

Buffer size	Reno		Vegas	
	Mean	Variance	Mean	Variance
5	1783	1534400	1862	16491400
7	1809	1771800	1868	5420840
9	1814	1372350	1871	3070490
11	1816	114206	1872	118129
13	1817	1061910	1873	12
14	1817	429598	1873	9
15	1815	212	1873	2
16	1812	113	1873	0
17	1813	40049	1873	0
20	1814	30581	1873	0
50	1810	8439	1873	0
315	1873	0.18	1873	0
316	1873	813	1873	0
500	1873	813	1873	0

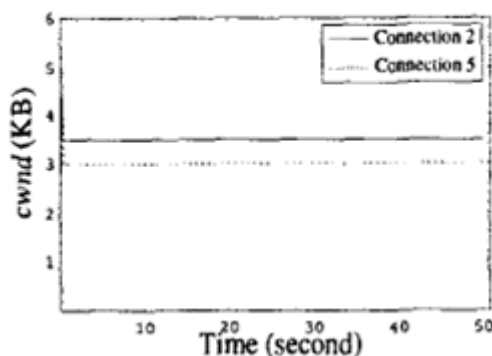
Πίνακας 3. Μέσος αριθμός των ACKs και των διαφορών τους για 5 TCP συνδέσεις με διαφορετικά μεγέθη buffer στα routers (50sec.)

(1) Το TCP Vegas επιτυγχάνει καλύτερη ρυθμοαπόδοση από το TCP Reno. Το [σχήμα 18](#) παρουσιάζει την δυναμική *cwnd* δύο αντιπροσωπευτικών συνδέσεων από πέντε συνδέσεις με το μέγεθος του buffer στα πέντε πακέτα σε κάθε δρομολογητή. Και οι δύο Reno και Vegas χρειάζεται να παλέψουν για περισσότερους buffer αφότου αυτοί είναι ανεπαρκείς για 5 συνδέσεις. Όπως το [σχήμα 18\(a\)](#) παρουσιάζει, το *cwnd* περιστασιακά δονείται για τις TCP Reno συνδέσεις λόγω του σχεδίου ελέγχου γρήγορης αποκατάστασης. Αυτή η δόνηση στο *cwnd* δημιουργεί χάσιμο χρόνου. Ως εκ τούτου, για τις Reno συνδέσεις, όσο μεγαλύτερο το μέγεθος buffer στο δρομολογητή, τόσο μικρότερος γίνεται ο ρυθμός εμφάνισης της αργής εκκίνησης, κατά συνέπεια βελτιώνεται η ρυθμοαπόδοση. Αντιθέτως, το TCP Vegas είναι πολύ σταθερό λόγω της μεθόδου της νέας αποφυγής συμφόρησής.



Σχήμα 18. Οι cwnd δυναμικές των δυο αντιπροσωπευτικών TCP συνδέσεων (μέγεθος buffer = 5)

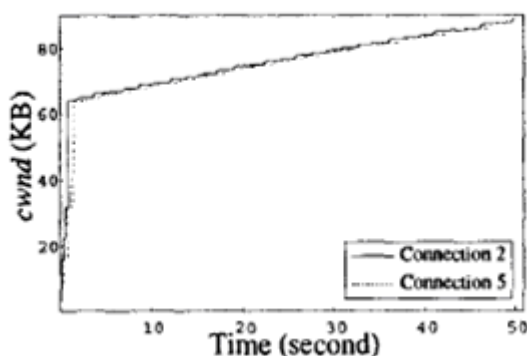
(2) Το TCP Vegas πετυχαίνει κακή εγκυρότητα όταν το μέγεθος του buffer είναι ανεπαρκές και καλή εγκυρότητα όταν το μέγεθος του buffer είναι ικανοποιητικό. Όπως φαίνεται στο [σχήμα 18](#), όταν το μέγεθος του buffer είναι ανεπαρκές, μερικές συνδέσεις Vegas καταλαμβάνουν το εύρος ζώνης πρώτα και αυξάνουν το cwnd τους, και οι άλλες Vegas συνδέσεις δεν έχουν καμία ευκαιρία να κερδίσουν εύρος ζώνης από αυτές. Συνδέσεις με εύρος ζώνης επιτυγχάνουν υψηλή ρυθμοαπόδοση, αλλά οι άλλες συνδέσεις επιτυγχάνουν χαμηλή ρυθμοαπόδοση. Αν και το Vegas έχει μεγαλύτερη ρυθμοαπόδοση από το Reno, εν τούτοις πάσχει από την υψηλή διαφορά.



Σχήμα 19. Οι cwnd δυναμικές των πέντε TCP Vegas συνδέσεων (μέγεθος buffer = 12)

Η διαφορά του TCP Vegas συρρικνώνεται (σε λιγότερο από 20) καθώς τα μεγέθη buffer υπερβαίνουν τα 12 πακέτα. Το [σχήμα 19](#) επιδεικνύει τη δυναμική *cwnd* πέντε TCP Vegas συνδέσεων με ένα μέγεθος buffer δώδεκα πακέτων. Δύο συνδέσεις έχουν μια στατική τιμή στο *cwnd* 3K και τρεις συνδέσεις έχουν την τιμή του *cwnd* 3.5K. Οι πέντε συνδέσεις έχουν ρυθμοαπόδοση κλεισίματος από τη στιγμή που οι πηγές μεταχειρίζονται το *cwnd* 3.5K ως 3K, το οποίο σημαίνει τρία πακέτα. Το Vegas μπορεί να επιτύχει ένα δίκαιο μέρος εύρος ζώνης όταν οι buffer είναι επαρκείς. Ως εκ τούτου, κατά τη χρησιμοποίηση του TCP Vegas, είναι σημαντικό να υπάρχουν αρκετοί buffer.

(3) Το TCP Reno χρειάζεται πολύ μεγαλύτερους buffer, από το TCP Vegas για να διατηρήσει τη σταθερότητα. Το TCP Reno αποδίδει σταθερά όταν τα μεγέθη buffer υπερβαίνουν τα 316 πακέτα, και το TCP Vegas επιτυγχάνει μια σταθερή κατάσταση όταν τα μεγέθη buffer υπερβαίνουν τα 16 πακέτα. Στο [σχήμα 20](#) καταρτίζεται η δυναμική του *cwnd* δύο αντιπροσωπευτικών TCP Reno συνδέσεων με ένα μέγεθος buffer 500 πακέτων μέσα στους δρομολογητές. Η σύγκριση των [σχημάτων 19](#) και [20](#) αποκαλύπτει ότι το TCP Vegas επιτυγχάνει τη στατική τιμή του *cwnd*, η οποία είναι 3K, αλλά το TCP Reno απαιτεί 60K του *cwnd* για να τελειώσει την αργή εκκίνησή του, κατόπιν αρχίζει η κατάσταση αποφυγής συμφόρησης. Ως εκ τούτου, το TCP Reno απαιτεί πολύ μεγαλύτερους buffer για να διατηρήσει τη σταθερότητα απ' ό,τι το TCP Vegas.



Σχήμα 20. Οι *cwnd* δυναμικές των δυο αντιπροσωπευτικών TCP Reno συνδέσεων (μέγεθος buffer = 500)

B. Αριθμός Συνδέσεων

Σε αυτήν την περίπτωση, εξετάζουμε πώς ο αριθμός των συνδέσεων έχει επιπτώσεις στην απόδοση του TCP Reno και Vegas. Αυτή η προσομοίωση χρησιμοποιεί μόνο την ομάδα e και τα μεγέθη buffer καθορίζονται σε πενήντα πακέτα. Ο [πίνακας 4](#) απαριθμεί αποτελέσματα που επιτυγχάνονται από τους διαφορετικούς αριθμούς των συνδέσεων TCP Reno και TCP Vegas.

Number of connections	Reno		Vegas	
	Mean	Variance	Mean	Variance
2	4512	12640	4679	12.50
4	2258	695	2341	0.00
6	1506	434	1560	0.66
8	1131	47461	1170	2.14
10	901	1789	936	2.00
12	755	10608	779	2.15
14	651	5128	668	5.29
15	608	3378	623	7.95
16	568	8724	584	6.91
17	534	7153	550	5232.50
18	507	39480	518	73660.20
20	456	43399	465	99821.80

Πίνακας 4. Μέσος αριθμός των ACKs και των διαφορών τους για διαφορετικό αριθμό συνδέσεων TCP (μέγεθος buffer = 50, 50sec.)

(1) Η ρυθμοαπόδοση διαφέρει ανάμεσα στα TCP Reno και το Vegas μειώνεται με τον αυξανόμενο αριθμό συνδέσεων. Το TCP Vegas έχει σαφώς μια καλύτερη χρήση των buffer από την προηγούμενη υποενότητα επειδή το TCP Vegas επιτυγχάνει τη σταθερότητα πολύ νωρίτερα από το TCP Reno. Η ρυθμοαπόδοση του TCP Vegas βελτιώνεται όταν υπάρχουν λίγες συνδέσεις.

(2) Υπολογίστε την κατάσταση εγκυρότητας για τις συνδέσεις TCP Vegas. Στην προηγούμενη περίπτωση, πέντε συνδέσεις Vegas πέτυχαν την εγκυρότητα όταν τα μεγέθη των buffer υπερέβησαν τα 12 πακέτα. Σε αυτήν την περίπτωση, buffer μεγέθους των 50 πακέτων απαιτούνται για να κρατήσουν την εγκυρότητα όταν ο αριθμός των συνδέσεων είναι κάτω από 16. Αυτό το αποτέλεσμα μπορεί να βρεθεί με έναν τραχύ τύπο εγκυρότητας. Όταν ο τύπος:

$$\frac{\text{Buffer Size in Packets}}{\text{Number of Vegas Connections}} \geq \beta$$

στέκει, το TCP Vegas μπορεί να επιτύχει ένα δίκαιο κομμάτι εύρους ζώνης. Εάν αυτή η αξία βρίσκεται κάτω από το β , είναι αβέβαιο εάν η εγκυρότητα ισχύει ακόμα ή όχι. Έχουμε πιστοποιήσει την ακρίβεια αυτού του τύπου για διαφορετικά μεγέθη buffer, διαφορετικούς αριθμούς συνδέσεων και διαφορετικές τοπολογίες δικτύων. Ο τύπος μπορεί να χρησιμοποιηθεί για να ελέγξει εάν ένα περιβάλλον δικτύων είναι κατάλληλο για το TCP Vegas.

Γ. Αριθμός Hops

(Στις τηλεπικοινωνίες ο όρος *Hop* έχει τις εξής σημασίες: ένα βήμα, από τον ένα router στον επόμενο, στην πορεία ενός πακέτου σε ένα δίκτυο πρωτοκόλλου Διαδικτύου)

Αυτή η έρευνα χρησιμοποιεί τώρα πέντε ομάδες, a, b, c, d, και e, για να μελετήσει πώς ο αριθμός *Hops* έχει επιπτώσεις στην απόδοση των συνδέσεων Reno και Vegas TCP. Οι πέντε ομάδες συνδέσεων έχουν την ίδια πηγή στην απόσταση προορισμού. Οι ομάδες a και b ταξιδεύουν μέσω δύο δρομολογητών, οι ομάδες c και d ταξιδεύουν μέσω τριών δρομολογητών και η ομάδα e ταξιδεύει μέσω τεσσάρων δρομολογητών. Τα μεγέθη buffer καθορίζονται σε πενήντα πακέτα. Ο [πίνακας 5](#) συνοψίζει τα αποτελέσματα προσομοίωσης.

(1) Όσο μεγαλύτερος ο αριθμός των *Hops* που η σύνδεση ακολουθεί το TCP Vegas αποφέρει μια καλύτερη ρυθμοαπόδοση από αυτό του TCP Reno. Η ρυθμοαπόδοση για τις ομάδες το a και b των συνδέσεων Reno είναι ελαφρώς καλύτερη από ότι για τις συνδέσεις Vegas, ενώ η ρυθμοαπόδοση για τις ομάδες c, d, και e των συνδέσεων Reno είναι ελαφρώς χειρότερες από ότι για Vegas συνδέσεις. Δηλαδή τα περισσότερα *Hops* που μια σύνδεση ακολουθεί αποφέρουν μια καλύτερη ρυθμοαπόδοση του TCP Vegas από αυτό του TCP Reno. Λόγω αυτού, στα παρόντα έγγραφα, το TCP Vegas μπορεί να επιτύχει 37-71% καλύτερη ρυθμοαπόδοση από το TCP Reno όταν δεκαεπτά *Hops* ταξιδεύουν [22], 4-20% καλύτερη όταν εννέα *Hops* ταξιδεύουν [27], και 0-10% καλύτερη όταν τέσσερις *Hops* ταξιδεύουν στην προσομοίωσή μας.

(2) Το TCP Vegas επιτυγχάνει ένα μικρότερο χάσμα ρυθμοαπόδοσης μεταξύ των ομάδων που ταξίδεψαν με διαφορετικούς αριθμούς *Hops*. Η ρυθμοαπόδοση για τις

ομάδες συνδέσεων του TCP Vegas κλείνει, το οποίο σημαίνει ότι ο αριθμός των *Hops* έχουν τη λιγότερη επιρροή στην απόδοση του TCP Vegas. Αντίθετα, η ρυθμοαπόδοση για τις συνδέσεις TCP Reno αλλάζει σημαντικά με μια αλλαγή στον αριθμό των *Hops*. Όταν η σύνδεση Reno ταξιδεύει ένα περισσότερος *Hop*, η σύνδεση παίρνει χιλιάδες λιγότερα *ACKs*.

(3) Το TCP Vegas επιτυγχάνει καλύτερη αλληλο-ομαδική εγκυρότητα από το TCP Reno.

Number of connections in each group	Group a		Group b		Group c		Group d		Group e		
	Mean	Variance	Mean	Variance	Mean	Variance	Mean	Variance	Mean	Variance	
Reno	1	5288	-	6084	-	3202	-	2577	-	374	-
	2	2969	29768	2635	90738	1288	46512	1624	288	304	3872
	3	1705	219242	2205	152674	1224	98551	779	404363	95	316
	4	1246	13545	1613	81305	939	72248	583	11467	116	6562
	5	1217	1447750	1156	4069	581	22680	614	39847	57	201
	6	1056	17347	960	66692	401	5678	506	5342	62	513
Vegas	1	4086	-	4570	-	3360	-	2879	-	1918	-
	2	1970	1.00	1971	0.50	1729	0.50	1729	0.50	982	0.00
	3	1278	1.33	1285	0.33	1185	0.33	1180	0.00	655	1.33
	4	964	2.25	966	0.25	894	1.00	894	2.25	479	2.00
	5	764	2.30	765	1.80	723	2.00	723	0.250	382	0.20
	6	604	1.76	606	0.66	578	4.66	578	2.26	373	37920

Πίνακας 5. Μέσος αριθμός των ACKs και των διαφορών τους για διαφορετικό αριθμό συνδέσεων σε κάθε ομάδα (μέγεθος buffer = 50)

Οι διαφορές στη ρυθμοαπόδοση ανάμεσα στις ομάδες a και b, και στις ομάδες c και d για τις TCP Vegas συνδέσεις είναι μικρότερες από εκείνες για το TCP Reno. Πιστεύουμε ότι το πέρασμα συνδέσεων μέσω του ίδιου αριθμού *Hops* πρέπει να έχει ίδια απόδοση. Προφανώς, το Reno αποτυγχάνει να επιτύχει αυτή την ιδιότητα.

(4) Το TCP Vegas επιτυγχάνει μια καλύτερη εσω-ομαδική εγκυρότητα από το TCP Reno. Η παραλλαγή για κάθε ομάδα συνδέσεων Vegas είναι πολύ μικρή εκτός από έξι συνδέσεις στην ομάδα e. Η μεγάλη διαφορά της εξαίρεση οφείλεται στην απουσία buffer. Αντιθέτως, το TCP Reno έχει πάντα μια μεγάλη διαφωνία σε κάθε ομάδα.

3.4.4.2. Ετερογενή περιβάλλοντα

A. Μεγάλη Buffer

Σε αυτήν την περίπτωση, ερευνάμε πώς το μέγεθος buffer έχει επιπτώσεις στην απόδοση 5 Reno και 5 Vegas συνδέσεων. Η χρησιμοποίηση των συνδέσεων Reno και Vegas μαζί, μας επιτρέπει να προσομοιώσουμε την αλληλεπίδραση στην απόδοσή τους. Αυτή η προσομοίωση χρησιμοποιεί μόνο την ομάδα e. [Ο πίνακας 6](#) επιδεικνύει τα αποτελέσματα προσομοίωσης.

Buffer size	Reno		Vegas	
	Mean	Variance	Mean	Variance
10	123	19085	1744	5748420
20	435	178166	1434	149313
30	800	275376	1066	82432
40	1042	41	823	15154
50	1177	39046	687	10888
60	1247	446	616	8592
70	1314	617	548	6845
80	1364	475	495	6681
90	1404	3.8	453	4776
100	1430	3346	425	4929
200	1559	43076	288	1479
300	1756	422392	108	1.8
331	1772	80583	99	1.8
332	1778	649	93	1.8
400	1778	649	93	1.8
500	1778	649	93	1.8

Πίνακας 6. Μέσος αριθμός των ACKs και των διαφορών τους για 5 Reno και 5 Vegas συνδέσεις με διαφορετικό μέγεθος buffer στα routers (50sec.)

(1) Στην ετερογενή περίπτωση, το TCP Reno διατηρεί μια καλύτερη ρυθμοαπόδοση με ένα μεγαλύτερο μέγεθος buffer. Όταν το μέγεθος του buffer αυξάνεται, η ρυθμοαπόδοση για το TCP Reno αυξάνεται ενώ για το TCP Vegas μειώνεται. Η προοδευτική αύξηση της ρυθμοαπόδοσης του TCP Reno οφείλεται στην υποστήριξη πρόσθετων buffer καθώς και στη συντηρητική συμπεριφορά του TCP Vegas. Το TCP Reno προσπαθεί για μεγαλύτερο αριθμό buffer που τελικά αποσπά εύρος ζώνης από το TCP Vegas. Το TCP Vegas αποδίδει καλά όταν τα μεγέθη buffer είναι μικρά επειδή το TCP Reno έχει υψηλότερη πιθανότητα να υποστεί τα διαλείμματα ή τις απώλειες πακέτων. Σε κάθε διάλειμμα, η τιμή $cwnd$ του TCP Reno αναστοιχειοθετείται στο 1, ενώ κάθε πακέτο χάνει στα μισά την τιμή $cwnd$. Οι συνδέσεις TCP Vegas ιδιοποιούνται το εύρος ζώνης όταν οι συνδέσεις TCP Reno χάνουν το εύρος ζώνης. Εντούτοις, η σύνδεση TCP Vegas δεν έχει καμία πιθανότητα να υπερτερήσει της TCP Reno η οποία αποδίδει καλά όταν τα μεγέθη buffer είναι μεγάλα.

(2) Η σταθερότητα στην ετερογενή περίπτωση είναι κυρίως εξαρτώμενη από τη συμπεριφορά των συνδέσεων TCP Reno. Θυμηθείτε ότι το TCP Reno είναι σταθερό όταν τα μεγέθη buffer υπερβαίνουν τα 316 πακέτα και το TCP Vegas είναι σταθερό όταν τα μεγέθη buffer υπερβαίνουν τα 16 πακέτα στα ομοιογενή περιβάλλοντα. Προφανώς, η εξέλιξη απόδοσης εξαρτάται σημαντικά από το TCP Reno όταν το TCP

Reno και Vegas εκτελούνται από κοινού. Ως εκ τούτου μέσα σε ετερογενές περιβάλλον, το Vegas δε μπορεί να κρατήσει την ίδια καλή απόδοση που έχει μέσα σε ομοιογενές περιβάλλον.

(3) Ο τύπος εγκυρότητας του TCP Vegas είναι ακατάλληλος για ετερογενή περιβάλλοντα. Επειδή η απόδοση του TCP Vegas καθορίζεται από το TCP Reno στα ετερογενή περιβάλλοντα, ο τύπος της εγκυρότητας ισχύει μόνο για τις ανόθευτες συνδέσεις TCP Vegas.

B. Αριθμός Συνδέσεων

Number of Vegas connections	Reno		Vegas	
	Mean	Variance	Mean	Variance
0	901	1789	-	-
1	933	863	908	-
2	940	1251	878	17848
3	995	11324	781	25576
4	1067	5236	729	14965
5	1177	39046	687	10888
6	1364	41	645	14.2
7	1628	997	634	7.2
8	2169	1200	625	5.4
9	3716	-	625	1.2
10	-	-	936	2.0

Πίνακας 7. Μέσος αριθμός των ACKs και των διαφορών τους για διαφορετικό αριθμό συνδέσεων Vegas με 10 συνδέσεις TCP (μεγέθος buffer = 50, 50sec.)

Σε αυτήν την περίπτωση, ερευνάμε πώς ο αριθμός σύνδεσεων επηρεάζει την απόδοση όταν TCP Reno και Vegas εκτελούνται από κοινού. Αυτή η προσομοίωση χρησιμοποιεί μόνο την ομάδα e, τα μεγέθη buffer καθορίζονται σε πενήντα πακέτα, και ο αριθμός συνδέσεων τίθεται σε δέκα. Αλλάζουμε τον αριθμό συνδέσεων Vegas μέσα σε δέκα TCP συνδέσεις για να ανακαλύψουμε την επιρροή της αναλογίας των συνδέσεων TCP Vegas. Ο [πίνακας 7](#) παρουσιάζει την επίπτωση των διαφορετικών αριθμών του TCP Vegas συνδέσεων στη μέση τιμή και τη διαφορά των ACKs.

(1) Η ρυθμοαπόδοση του TCP Reno βελτιώνεται πολύ με μεγαλύτερη ποσότητα TCP Vegas συνδέσεων. Διαισθητικά, όσο μεγαλύτερος ο αριθμός του TCP Reno συνδέσεων σημαίνει μια χειρότερη ρυθμοαπόδοση των TCP Vegas συνδέσεων. Εντούτοις, αυτό δεν ισχύει. Όσο μεγαλώνει ο αριθμός των συνδέσεων TCP Reno τόσο πιο εντατικός ανταγωνισμός για το εύρος ζώνης μεταξύ των ίδιων των

συνδέσεων. Οι ανταγωνισμοί για το εύρος ζώνης μπορούν να προκαλέσουν διαλείμματα, μειώνοντας κατά συνέπεια τη ρυθμοαπόδοση του TCP Reno και δίνουν στο TCP Vegas το εύρος ζώνης. Μόλις μειωθεί ο αριθμός συνδέσεων Reno, η ρυθμοαπόδοση TCP Reno αυξάνεται και η ρυθμοαπόδοση TCP Vegas συρρικνώνεται.

(2) Κατά την παρούσα περίοδο οι χρήστες δεν θα υιοθετήσουν το TCP Vegas. Όταν όλες οι συνδέσεις είναι TCP Vegas, κάθε μία αποκτά καλύτερη ρυθμοαπόδοση και μικρότερη διαφορά. Ως εκ τούτου εάν «όλοι» οι χρήστες ταυτόχρονα αλλάξουν από Reno σε Vegas, η απόδοση θα αναβαθμιστεί πραγματικά. Εντούτοις, η μετάβαση είναι προοδευτική. Το πρώτο άτομο που θα χρησιμοποιήσει το TCP Vegas θα πάρει μια βελτιωμένη ρυθμοαπόδοση από 901 έως 908. Η δεύτερη μετατροπή στο TCP Vegas δεν βελτιώνει τη ρυθμοαπόδοση επειδή η ρυθμοαπόδοση ελαττώνεται από 933 έως 878. Επιπλέον, η ρυθμοαπόδοση του πρώτου χρήστη για να κάνει την αλλαγή θα μειωθεί από 908 έως 878. Η ρυθμοαπόδοση του τελευταίου ατόμου που κάνει την αλλαγή θα ελαττωθεί από 3716 σε 936, αν και αυτή η αλλαγή μπορεί σημαντικά να ενισχύσει την ρυθμοαπόδοση άλλων. Ως εκ τούτου, ενεργώντας χωριστά οι χρήστες δεν θα υιοθετήσουν την έκδοση Vegas προς το παρόν.

Γ. Αριθμός Hops

Σε αυτήν την περίπτωση, χρησιμοποιούνται πέντε ομάδες a, b, c, d, και e για να ερευνηθούν πώς ο αριθμός Hops επηρεάζει την απόδοση των Reno και Vegas TCP συνδέσεων. Τα μεγέθη buffer καθορίζονται σε πενήντα πακέτα. Ο αριθμός συνδέσεων Reno και Vegas TCP σε κάθε ομάδα αλλάζουν. Ο [πίνακας 8](#) συνοψίζει τα αποτελέσματα της προσομοίωσης.

(1) Το TCP Reno παίρνει εύρος ζώνης από το TCP Vegas. Η σύγκριση του [πίνακα 8](#) με τον [πίνακα 5](#) αποκαλύπτει το αποτέλεσμα ότι το TCP Reno παίρνει πολύ εύρος ζώνης από το TCP Vegas. Για παράδειγμα, η ρυθμοαπόδοση για 2 οι ομοιογενείς συνδέσεις TCP Reno είναι 2969, ενώ η ρυθμοαπόδοση για 2 το ομοιογενές TCP Vegas συνδέσεις είναι 1970. Εντούτοις, η ρυθμοαπόδοση για 1 TCP Reno και 1 TCP Vegas συνδέσεις είναι 4274 και 830, αντίστοιχα.

Number of connections in each group	Group a		Group b		Group c		Group d		Group e		
	Mean	Variance	Mean	Variance	Mean	Variance	Mean	Variance	Mean	Variance	
Reno	1	4274	-	4517	-	2609	-	2593	-	297	-
Vegas		830	-	776	-	925	-	742	-	371	-
Reno	2	2237	1250	1764	480	784	61600	1464	2664	206	544
Vegas		669	0.5	558	7200	540	13944	465	5512	189	2048
Reno	3	1091	69436	895	256697	431	5249	785	12562	73	684
Vegas		858	15700	564	10876	505	13540	652	23209	124	856

Πίνακας 8. Μέσος αριθμός των ACKs και των διαφορών τους με αριθμό συνδέσεων σε κάθε ομάδα (μεγέθος buffer = 50

3.5 Ένα Τροποποιημένο Σχέδιο Αναμετάδοσης TCP-NewReno για Δίκτυο με Απώλειες

3.5.1 Γενική Περιγραφή

Μ'αυτήν την μελέτη θέλουμε να προτείνουμε ένα τροποποιημένο TCP-NewReno για τα δίκτυα με απώλειες όπου εμφανίζεται απώλεια πακέτων λόγω του αναξιόπιστου στρώματος συνδέσεων στοιχείων ή λόγω απωλειών στο φυσικό στρώμα μεταφορών. Προηγούμενη έρευνα έχει αποδείξει ότι το σχέδιο του TCP-NewReno ή το σχέδιο για τη μείωση γρήγορης-αναμετάδοσης στα κατώτατα όρια μπορεί αποτελεσματικά να βελτιώσει την απόδοση του TCP-NewReno στα δίκτυα με απώλειες. Εντούτοις, μέσω της προσομοίωσης, δείχνουμε ότι αυτά τα δύο σχέδια δεν μπορούν να ανταπεξέλθουν θετικά λόγω μερικών δυσμενών παραγόντων όπως την παράδοση των εκτός λειτουργίας πακέτων και τις απώλειες αναμετάδοσης. Κατόπιν, δίνουμε μερικούς κανόνες τροποποίησης για το TCP-NewReno. Σύμφωνα με τα αποτελέσματα της προσομοίωσης θα δείξουμε ότι η τροποποίησή μας λειτουργεί καλά τόσο για το περιβάλλον LAN όσο και για το WAN.

Το TCP έχει σχεδιαστεί για τα δίκτυα μέσα στα οποία οι απώλειες/φθορές πακέτων οφείλονται κυρίως στη συμφόρηση. Επάνω σε εκείνη την υπόθεση, το σχέδιο αναμετάδοσης του TCP είναι συντηρητικό προκειμένου να αποφευχθεί η περιττή αναμετάδοση. Αλλά, με τη γρήγορη ανάπτυξη του Διαδικτύου, πολλά δίκτυα συγκλίνουν με το Διαδίκτυο, όπως το ATM δίκτυο και τα ασύρματα δίκτυα. Προηγούμενη έρευνα έδειξε ότι σύνδεση με απώλειες υπάρχει σε αυτά τα δίκτυα, και η απώλεια πακέτων εμφανίζεται λόγω του αναξιόπιστου στρώματος δεδομένων

σύνδεσης όπως το δίκτυο του ATM ή με απώλειες στο φυσικό στρώμα μεταφορών όπως στα ασύρματα δίκτυα. Επιπλέον, στο μέλλον η κυκλοφορία δεδομένων του TCP θα μοιράζεται το δίκτυο με πολυμέσα όπως τη φωνή και το βίντεο, τα οποία μπορεί να οδηγήσουν σε παροδικές απώλειες στην κυκλοφορία του TCP. Επομένως το τρέχον TCP δεν είναι προσαρμοστικό στο τρέχον περιβάλλον δικτύου, και η βελτιστοποίηση του πρωτοκόλλου μέσα στο στρώμα μεταφοράς είναι απαραίτητη και σημαντική.

Πολλή έρευνα έχει δείξει ότι, προκειμένου να βελτιωθεί η ρυθμοαπόδοση του TCP στα δίκτυα με απώλειες, είναι σημαντικότερο να αποφευχθεί το διαδοχικό διάλειμμα και οι δραστικές μειώσεις παραθύρου λόγω των τυχαίων απωλειών. Για αυτό, χρειαζόμαστε έναν αποτελεσματικό σχέδιο αναμετάδοσης του TCP. Η περισσότερη εργασία γίνεται πριν ολοκληρωθεί το σχέδιο TCP-NewReno ή το σχέδιο μείωσης της γρήγορης αναμετάδοσης αφού στο τέλος μπορεί αποτελεσματικά να βελτιώσει την απόδοση του TCP στα δίκτυα με απώλειες. Αλλά, αυτά τα σχέδια δεν εξετάζουν μερικούς δυσμενείς παράγοντες όπως την παράδοση πακέτων εκτός λειτουργίας καθώς και τις απώλειες αναμετάδοσης.

Σε αυτό το έγγραφο, μελετάμε την επίπτωση στις απώλειες αναμετάδοσης. Μέσω της προσομοίωσης, συμπεραίνουμε ότι η ρυθμοαπόδοση TCP σημαντικά υποβιβάζεται λόγω των απωλειών αναμετάδοσης. Επομένως, δίνουμε κάποια τροποποίηση σε TCP-NewReno, και συγκρίνουμε την τροποποίησή μας με το τρέχον TCP στη ρυθμοαπόδοση σε υψηλά BER.

3.5.2. Η Αποτυχία του TCP-NewReno σε δίκτυα με υψηλό BER

Στο [30], οι συντάκτες αποκαλύπτουν ότι στους λόγους υποβάθμισης της ρυθμοαπόδοσης του TCP πάνω στα δίκτυα με απώλειες συγκαταλέγεται,

- 1) ο ψεύτικος έλεγχος συμφόρησης που προκαλείται από τις τυχαίες απώλειες πληροφοριών,
- 2) οι πολλαπλές απώλειες/φθορές σε ένα ενιαίο παράθυρο,
- 3) η απώλεια αναμεταδιδόμενων πακέτων λόγω των υψηλών BER.

Οι συντάκτες στο [30] δείχνουν επίσης ότι ο τρέχον αλγόριθμος γρήγορης αναμετάδοσης/γρήγορης επαναφοράς του TCP-Reno δεν δουλεύει καλά όταν

συμβαίνουν πολλαπλές απώλειες μέσα το ίδιο παράθυρο. Επομένως, μόνο μερικά από τα χαμένα πακέτα μπορούν να ανακτηθούν από αλγόριθμο γρήγορης αναμετάδοσης. Τα υπόλοιπα είναι συνήθως ανακτημένα από την διαδικασία αργής εκκίνησης μετά από αναμετάδοση διαλείμματος, που προκαλεί μια μεγάλη αδρανή περίοδο (λόγω ξεχρονισμένου χρονομέτρου) που οδηγεί σε σημαντική υποβάθμιση της απόδοσης.

Το TCP-NewReno είναι μια καλή λύση σε πολλαπλές απώλειες σε ένα ενιαίο παράθυρο. Εάν μερικά από τα πακέτα μεταδοθούν πριν από την γρήγορη αναμετάδοση απώλειας, τα ACK's για το γρήγορα-αναμεταδιδόμενο πακέτο θα αναγνωρίσει μερικά αλλά όχι όλα τα μεταδιδόμενα πακέτα. Αυτό το πακέτο ACK καλείται πακέτο μερικής αναγνώρισης. Το TCP-NewReno θα αναμεταδώσει το υποδεικνυμένο πακέτο χωρίς καθυστέρηση κατά τη λήψη του πακέτου μερικής αναγνώρισης.

Στο [\[29\]](#), οι συντάκτες συμπεραίνουν ότι το TCPNewReno μπορεί να δουλέψει καλά μόνο όταν υπάρχουν πολλαπλές απώλειες/σφάλματα σε ένα ενιαίο παράθυρο και οι αναμεταδόσεις είναι συνήθως επιτυχείς. Αλλά όταν η αναμετάδοση απωλειών εμφανίζεται λόγω των υψηλών BER σε χαμηλότερο στρώμα, τα χαμένα πακέτα που διαβιβάζονται μετά την γρήγορη αναμετάδοση δεν μπορούν να ανακτηθούν γρήγορα από το TCP-NewReno, και δεν μπορούν επίσης να αναμεταδοθούν γρήγορα λόγω μη αρκετού αναπαραχθέντος ACK. Επομένως το διαδοχικό διάλειμμα είναι αναπόφευκτο. Αυτός οδηγεί σε υποεκμετάλλευση των πόρων του δικτύου. Προκειμένου να εξηγηθεί αυτό σαφώς, δίνουμε ένα παράδειγμα ([δείτε σχήμα 21](#)).

Από το διάγραμμα ροής ([σχήμα 21](#)), μπορούμε να δούμε ότι το διαδοχικό διάλειμμα οδηγεί σε έναν μακρύ αδρανή χρόνο και σε χαμηλή τιμή και της αργής εκκίνησης και του μεγέθους παραθύρου συμφόρησης, τα οποία υποβιβάζουν σημαντικά την ρυθμοαπόδοση του TCP.

3.5.3 Η Τροποποίησή μας σε TCP-NewReno.

Από το προαναφερθέν TCP σχέδιο αναμετάδοσης, διαπιστώνουμε ότι επιταχύνοντας την αντίδραση του TCP στις απώλειες πακέτων είναι αποφασιστικά σημαντικό. Αλλά το τρέχον TCP είναι συντηρητικό σε τρέχοντα δίκτυα με απώλειες με υψηλά BER.

Στην τροποποίησή μας σε TCP-NewReno, η γρήγορη αναμετάδοση και η αναμετάδοση διαλείμματος ονομάζονται και οι δύο καθυστερημένη-αναμετάδοση (*delay-retransmit*). Όταν το καθυστερημένο-αναμεταδιδόμενο πακέτο αναγνωρίζεται, το πακέτο που στάλθηκε πριν από αυτό πρέπει επίσης να αναγνωρισθεί εάν καμία απώλεια δεν συμβαίνει. Έτσι, εκλαμβάνουμε τα πακέτα που δεν αναγνωρίστηκαν ως χαμένα, και αναμεταδίδονται αμέσως. Εάν η καθυστερημένη-αναμετάδοση (*delay-retransmit*) προκαλείται ψεύτικα όταν δεν συμβαίνει καμία απώλεια λόγω της υποτίμησης του RTT, η τροποποίησή μας μπορεί να οδηγήσει σε περιττή αναμετάδοση έτσι ώστε η αποτελεσματική ρυθμοαπόδοση να υποβιβασθεί λίγο. Εντούτοις, το θεωρούμε ως γεγονός με μικρή πιθανότητα επειδή η κατά προσέγγιση εκτίμηση του διαλείμματος είναι συχνά μεγαλύτερη από την ακριβή περίοδο του πραγματικού διαλείμματος.

Όταν η τροποποίησή μας χρησιμοποιείται, χρειαζόμαστε μία μεταβλητή για να καταγράψει τον υψηλότερο αριθμό ακολουθίας πριν από οποιαδήποτε καθυστερημένη-αναμετάδοση στην πηγαία TCP. Εδώ, δείχνουμε το «H» σε αυτήν την μεταβλητή. Εάν η αναγνώριση του πακέτου καθυστερημένης-αναμετάδοσης δεν καλύπτει το «H», αναμεταδίδουμε το συγκεκριμένο πακέτο αμέσως. Δεν αλλάζουμε άλλους αλγόριθμους του TCP-NewReno. Χρησιμοποιούμε το παραπάνω παράδειγμα για να εξηγήσουμε το τροποποιημένο TCP-NewReno μας ([δείτε το σχήμα 21](#)).

Συγκρίνοντας το TCP-NewReno, το τροποποιημένο μας TCP-NewReno μπορεί να επιταχύνει την αντίδραση του TCP στις απώλειες πακέτου και μειώνει συχνότητα του διαλείμματος προκειμένου να αποφευχθεί η δραστική μείωση παραθύρου και η μεγάλη περίοδος αδράνειας, οι οποίες οδηγούν σε δίκτυα με βαριές απώλειες.

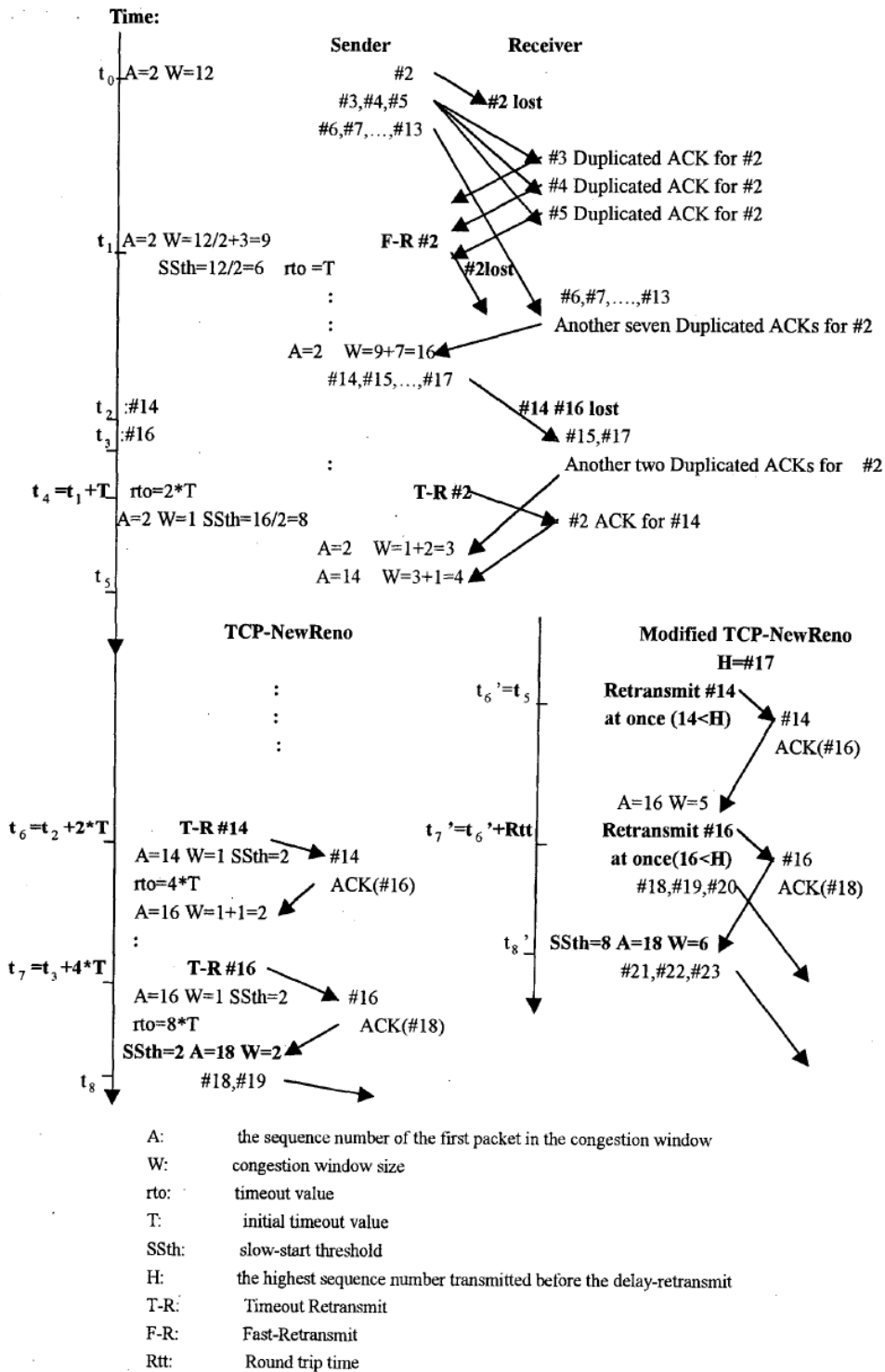
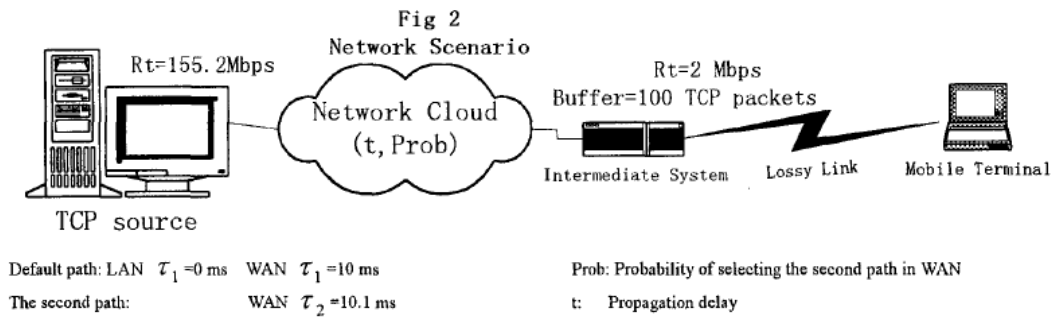


Fig. 1 Process Difference of TCP-NewReno and Our Modification

Σχήμα 21. Διαδικασία Διαφορών του TCP – New Reno και των τροποποιήσεών μας.



Σχήμα 22. Σενάριο Δικτύου

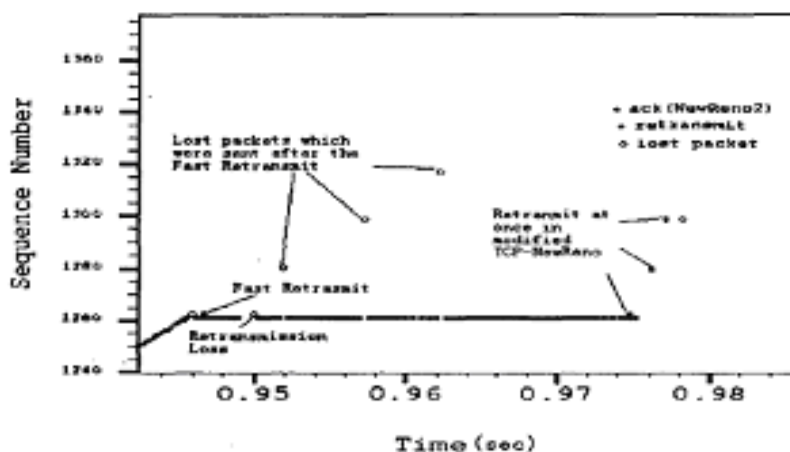
3.5.4 Αποτέλεσμα Προσομοίωσης και Συζήτηση

Το [σχήμα 22](#) παρουσιάζει το σενάριο του δικτύου μας. Εμείς διαμορφώνουμε την ασύρματη σύνδεση σαν μία απλή χαμένη σύνδεση και χρησιμοποιούμε την άπληστη πηγή που παράγει δεδομένα όταν απαιτείται. Τα TCP πακέτα είναι καθορισμένα στο μήκος των 512 bytes. Το ποσοστό μετάδοσης της πηγής είναι 155.2 Mbps. Το μέγεθος buffer του ενδιάμεσου συστήματος είναι 100 TCP πακέτα. Το ποσοστό μετάδοσης του ενδιάμεσου συστήματος είναι 2 Mbps. Όταν η κατάσταση ενός τοπικού δικτύου θεωρείται, η καθυστέρηση διάδοσης υποτίθεται ότι είναι αμελητέα, η παράδοση χαλασμένων πακέτων επίσης αγνοείται και όλα τα πακέτα από τη ροή στοιχείων θεωρείται ότι ακολουθούν την προεπιλεγθείσα πορεία. Από την άλλη μεριά, στο δίκτυο ευρείας περιοχής, θεωρούμε ότι μπορούν να ακολουθήσουν δύο διαφορετικές πορείες με τη διαφορετική καθυστέρηση διάδοσης ($\tau_1=10\text{ms}$, $\tau_2=10.1\text{ms}$) προκειμένου να μελετήσουμε την επίπτωση από την παράδοση των εκτός λειτουργίας πακέτων στη ρυθμοαπόδοση του TCP. Επίσης υποθέτουμε ότι τα πακέτα ACK είναι διαβιβασθέντα χωρίς απώλεια δεδομένου ότι γενικά έχουν ορισθεί σε υψηλότερη προτεραιότητα. Δεδομένου ότι το ACK είναι σχετικά μικρότερο από το πακέτο δεδομένων (40-byte του ACK's κατά 512-byte του πακέτου), η καθυστερημένη του μετάδοση είναι πολύ πιο σύντομη. Για την απλοποίηση, αγνοούμε την καθυστέρηση μετάδοσής του.

Στην προσομοίωση, το τροποποιημένο TCP-NewReno ονομάζεται NewReno2. Μελετάμε αρχικά την επίπτωση στην ρυθμοαπόδοση του TCP στο LAN και των πολλαπλών απωλειών και των απωλειών αναμεταδόσης. Το τροποποιημένο μας

TCP-NewReno μπορεί να αναμεταδώσει τα χαμένα πακέτα αμέσως για να βελτιώσει την ρυθμοαπόδοση (σχήμα 23). Το σχήμα 24a δείχνει ότι, χωρίς την παράδοση των εκτός λειτουργίας πακέτων που μειώνει το *fast-retransmit* στο κατώτατο όριο K μπορεί να βελτιώσει τη ρυθμοαπόδοση. Συμπεραίνουμε επίσης από το σχήμα 24b ότι, συγκρινόμενο με το τρέχον TCP το τροποποιημένο μας TCP-NewReno καταφέρνει καλύτερη απόδοση στο LAN με μία σύνδεση απώλειας.

Σε ένα WAN χωρίς λάθη, μελετάμε κυρίως τη επίπτωση της παράδοσης εκτός λειτουργίας πακέτων στη ρυθμοαπόδοση. Από το σχήμα 25 βλέπουμε ότι, αυξάνοντας την πιθανότητα της επιλογής της δεύτερης πορείας που οδηγεί σε υψηλή πιθανότητα παράδοσης εκτός λειτουργίας πακέτων, η ρυθμοαπόδοση υποβιβάζεται λόγω των συχνών περιττών αναμεταδόσεων. Το TCP-NewReno(K=1) υποβιβάζεται σε μεγαλύτερο βαθμό, τη στιγμή που και το τροποποιημένο μας TCP-NewReno και το TCP-NewReno(K=3) δουλεύουν καλά. Σε ένα WAN με μια σύνδεση απώλειας, συγκρίνουμε το τροποποιημένο μας TCP-NewReno, το TCP-NewReno(K=1), και το TCP-NewReno(K=3). Το σχήμα 26 δείχνει πότε η επίπτωση από την παράδοση του εκτός λειτουργίας πακέτου είναι κυρίως σε χαμηλά BER. Το τροποποιημένο TCP-NewReno και το TCP-NewReno(K=3) εργάζονται καλύτερα από το TCPNewReno(K=1). Με την αύξηση του BER, η επίπτωση των τυχαίων απωλειών γίνεται όλο και περισσότερο σημαντική, η ρυθμοαπόδοση του TCP-NewReno(K=3) υποβιβάζεται σημαντικά έτσι ώστε η ρυθμοαπόδοση του TCP-NewReno(K=1) να την υπερβαίνει, και το τροποποιημένο μας TCP-NewReno δουλεύει ακόμα καλά.



Σχήμα 23 ρυθμός αναμεταδιδόμενων πακέτων

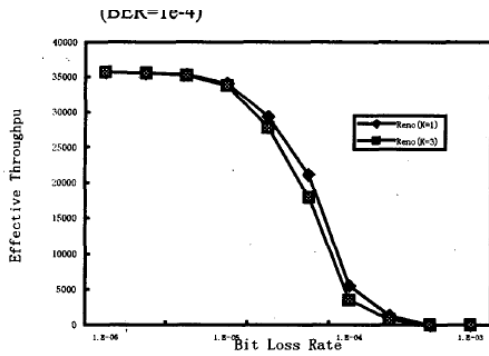


Fig. 4a Throughput of versions of TCP versus bit loss rate in LAN with a lossy link. (K is the fast-retransmit threshold)

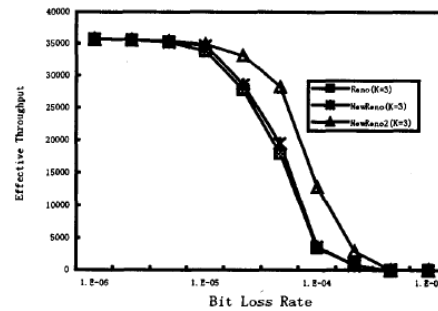


Fig. 4b Throughput of versions of TCP versus bit error rate in LAN with a lossy link. (K is the fast-retransmit threshold)

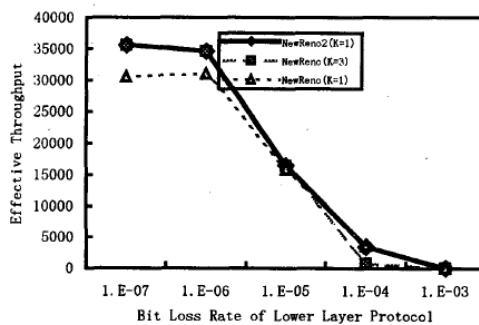


Fig. 6 Throughput of versions of TCP versus bit error rate in WAN with a lossy link. (K is the fast-retransmit threshold)

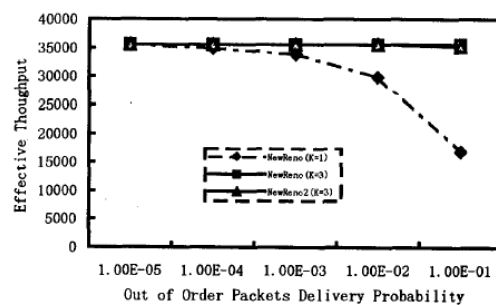


Fig 5 Simulation results which demonstrate the effect of out of order packet delivery on TCP throughput in a error free WAN.

Σχήματα 24,25,26 η επίπτωση από την παράδοση του εκτός λειτουργίας πακέτου είναι κυρίως σε χαμηλά BER

3.6 Ανταλλαγές Ενέργειας/Ρυθμοαπόδοσης(throughput) των Στρατηγικών Ελέγχου Λάθους του TCP(tcp error)

3.6.1 Γενική Περιγραφή

Στις σημερινές παγκόσμιες επικοινωνίες όλο και περισσότερο εμπλέκονται κινητές και με μπαταρίες συσκευές (π.χ. laptop, IP-phone) στα ενσύρματα και ασύρματα δίκτυα. Η ενεργειακή αποδοτικότητα, καθώς επίσης και η ρυθμοαπόδοση, γίνονται χαρακτηριστικά υπηρεσιών με κυρίαρχη σπουδαιότητα στα πρωτόκολλα επικοινωνίας. Η ευρεία δυνατότητα εφαρμογής των IP-networks/συσκευών, και το ευρύ φάσμα σε εφαρμογές βασισμένες στο TCP, έχουν καταστήσει το TCP ως “de facto” πλέον αξιόπιστο πρωτόκολλο μεταφορών όχι μόνο για τις ενσύρματες επικοινωνίες, αλλά και για τις ασύρματες και τις μικτές (ενσύρματες/ασύρματες) επικοινωνίες. Οι αλγόριθμοι ελέγχου συμφόρησης του TCP έχουν πρόσφατα

βελτιωθεί για να επιτύχουν την υψηλότερη ρυθμοαπόδοση. Ακόμη και με αυτές τις μετατροπές, εντούτοις, οι εκδόσεις TCP δεν περιλαμβάνουν μία ευέλικτη στρατηγική ανάκτησης λάθους που να ανταποκρίνεται στα ευδιάκριτα χαρακτηριστικά του περιβάλλοντος και στη συσκευή περιορισμών. Έχουμε συγκρίνει την ενέργεια και ρυθμοαπόδοση-αποδοτικότητα των TCP στρατηγικών ελέγχου λάθους με βάση τα αποτελέσματα που μαζεύονται από την εφαρμογή μας στο TCP Tahoe, Reno, και New Reno. Δείχνουμε ότι, ανάλογα με τη συχνότητα και τη διάρκεια του λάθους, κάθε μια καταδεικνύει πρέπουσα συμπεριφορά κάτω από συγκεκριμένες περιστάσεις, με το Tahoe να έχει πάνω - κάτω την πλέον ενεργειακή συντήρηση από τα τρία. Κανένα από αυτά, εντούτοις, δεν κατέχει ένα ευδιάκριτο γενικό πλεονέκτημα που να μας καθιστά σαφές την επιλογή της έκδοσης για ενσύρματα/ασύρματα ετερογενή δίκτυα.

Η αποδοτικότητα ρυθμοαπόδοσης του TCP υπήρξε θέμα εντατικής προσοχής κατά τα τελευταία χρόνια. Εντούτοις, η ανταλλαγή ενέργειας/ρυθμοαπόδοσης δεν ήταν επαρκώς μελετημένη στη σχετική βιβλιογραφία. Αυτή η ανταλλαγή γίνεται περισσότερο σημαντική λόγω της ευρείας δυνατότητας εφαρμογής σε κινητά δίκτυα, η οποία περιλαμβάνει συχνά τη χρήση συσκευών με μπαταρίες (π.χ. lap-top, IP-τηλέφωνο). Η ικανότητα στην ενεργειακή-προστασία των πρωτοκόλλων επικοινωνίας μπορεί να διαδραματίσει έναν σημαντικό ρόλο στον καθορισμό στη διάρκεια της λειτουργικής ζωής τέτοιων συσκευών. Ειδικότερα, η λειτουργικότητα στο στρώμα μεταφορών μπορεί να ασκήσει σημαντική επίδραση στην ενεργειακή κατανάλωση.

Το κλειδί για τις αποδοτικές ανταλλαγές ενέργειας/ρυθμοαπόδοσης στα αξιόπιστα πρωτόκολλα μεταφορών, όπως καταδεικνύεται από τις δοκιμές που παρουσιάζονται εδώ, είναι ο μηχανισμός ελέγχου λάθους. Ο έλεγχος λάθους είναι συνήθως μια διαδικασία ελέγχου σε δύο στάδια: α) ανίχνευση λάθους β) ακολουθούμενη από την αποκατάσταση λάθους. Τα περισσότερα πρωτόκολλα μεταφορών όπως το TCP ανιχνεύει τα λάθη με παρακολούθηση της ακολουθίας τμηματικών δεδομένων που παρελήφθησαν ή αναγνωρίστηκαν. Όταν τα διαλείμματα διαμορφώνονται σωστά, ένα ελλείπον τμήμα λαμβάνεται για να δείξει το λάθος, δηλαδή ότι το τμήμα χάνεται. Αξιόπιστα πρωτόκολλα συνήθως εφαρμόζουν μια στρατηγική αποκατάστασης λάθους με βάση δύο τεχνικές: α) την αναμετάδοση των χαμένων τμημάτων και β) την προς τα κάτω ρύθμιση του μεγέθους παραθύρου του αποστολέα και αναπροσαρμογή της περιόδου διαλείμματος. Αναμετάδοση των

τμημάτων που προσπάθησαν ανεπιτυχώς, βεβαίως, συνεπάγεται τις πρόσθετες ενεργειακές δαπάνες και οδηγεί σε χαμηλότερα αποτελεσματική ρυθμοαπόδοση. Κατά συνέπεια, η ενεργειακή αποδοτικότητα της στρατηγικής ελέγχου λάθους δεν μπορεί να μελετηθεί χωρίς να ληφθούν υπόψη οι σχετικοί μηχανισμοί για την αποκατάσταση. Για παράδειγμα, τα διαλείμματα και οι ρυθμίσεις του παραθύρου συμφόρησης και το κατώτατο όριό του, διαδραματίζουν έναν σημαντικό ρόλο στη συνολική ρυθμοαπόδοση και στις επιδόσεις ενεργειακής-δαπάνης. Ενώ η καθαρή έκβαση της διαδικασίας ανάκτησης πρέπει να είναι η αναμετάδοση των ελλειπόντων τμημάτων, η φύση του λάθους πρέπει πραγματικά να παίζει έναν καθοριστικό ρόλο καθορίζοντας τη στρατηγική ανάκτησης που χρησιμοποιείται.

Μέχρι πρόσφατα, το TCP μελετήθηκε στα πλαίσια της αποδοτικότητάς της ρυθμοαπόδοσης του επί περίπου «ομοιογενή» περιβάλλοντα (π.χ. ενσύρματου έναντι ασύρματου) συνεπώς, η ομοιογένεια απεικονίστηκε επίσης στη φύση των λαθών τα οποία εξετάστηκαν (π.χ. λάθη συμφόρησης/μετάδοσης έναντι τυχαίων/βίαιων/εξασθενίσεις διαύλου λαθών). Ο Jacobson στο [35] ήταν ο πρώτος που μελέτησε τον αντίκτυπο της αναμετάδοσης στη ρυθμοαπόδοση, βασισμένος σε πειράματα με ενσύρματα δίκτυα με συμφόρηση. Πιο πρόσφατα, άλλοι έχουν αφιερώσει επίσης την προσοχή τους στην ρυθμοαπόδοση του TCP και προτείνουν τροποποιήσεις προκειμένου να ενισχυθεί η απόδοσή του. Για παράδειγμα, οι Floyd & Henderson, έχουν δείξει ότι η ρυθμοαπόδοση του TCP σε καλωδιακό περιβάλλον μπορεί να γίνει για να βελτιώσει τη χρησιμοποίηση της Μερικής Αναγνώρισης και Γρήγορης Ανάκτησης στο [34]. Από την άλλη, οι Balakrishnan κ.α. στο [32], Lakshman & Madhow στο [37], και άλλοι (π.χ. [36]), έχουν δείξει ότι η ρυθμοαπόδοση του TCP υποβιβάζεται όταν το πρωτόκολλο χρησιμοποιείται από δορυφόρο ή ασύρματες συνδέσεις. Εντούτοις, μπορεί να προκύψουν προβλήματα στον κοντινό μέλλον:

- Οι σημερινές εφαρμογές TCP αναμένονται για να τρέξουν σε φυσικά ετερογενή, αλλά λειτουργικά ολοκληρωμένα, ενσύρματα/ασύρματα περιβάλλοντα. Οι προτεινόμενες τροποποιήσεις δεν ικανοποιούν αυτήν την καθολική λειτουργία δεδομένου ότι δεν ρυθμίζουν ευέλικτα τη στρατηγική ανάκτησης στη μεταβλητή φύση των λαθών.
- Η κινητήριος δύναμη που οδηγεί αυτές τις τροποποιήσεις αγνοεί την ανταλλαγή ενέργειας/ρυθμοαπόδοσης, η οποία γίνεται ένα κύριο ζήτημα απόδοσης.

Οι μόνες δημοσιευμένες μελέτες της κατανάλωση ενέργειας του TCP είναι από Zorzi κ.α στα [33, 42]. Οι συντάκτες παρουσιάζουν αποτελέσματα, τα οποία έχουν ευρέως αναφερθεί στην πρόσφατη βιβλιογραφία, βασισμένη σε ένα πιθανολογικό πρότυπο συμπεριφοράς του TCP. Ενώ το πρότυπο κάνει μερικές δραστικές απλοποιημένες υποθέσεις προκειμένου να διατηρηθεί η αναλυτική ευπείθεια (analytic tractability), καμία κατάλληλη επικύρωση της ακρίβειάς της δεν έχει αναφερθεί, έτσι παραμένει ασαφές πώς αυτές οι απλουστευμένες υποθέσεις ίσως ανατρέψουν τα αποτελέσματα. Επιπλέον, τα συμπεράσματα των συντακτών είναι βασισμένα στη πρόσθετη δαπάνη ενέργειας που προκαλείται από την εξασθένιση διαύλου μόνο στο πλαίσιο των αναμεταδομένων στοιχείων. Ο εκτεταμένος χρόνος επικοινωνίας καθώς επίσης και συγκεκριμένα λειτουργικά χαρακτηριστικά δεν έχουν ληφθεί υπόψη¹.

¹ Σε αυτό το έγγραφο είμαστε σε θέση να τεκμηριώσουμε μόνο ένα από τα σημαντικά τους συμπεράσματα, δηλαδή, τη σχετική ενεργειακή αποδοτικότητα από το Tahoe.

Σε αυτό το έγγραφο παρουσιάζουμε αποτελέσματα πάνω στην απόδοση σε ενέργεια και ρυθμοαπόδοση των TCP Tahoe, Reno, και NewReno. Ιστορικά, το TCP Tahoe ήταν η πρώτη τροποποίηση του TCP [38]. Το νεώτερο TCP Reno περιέλαβε το αλγόριθμο γρήγορης αποκατάστασης [31]. Αυτό ακολουθήθηκε από το NewReno [34] και το μηχανισμό Μερικής Αναγνώρισης για τις πολλαπλές απώλειες στο ενιαίο παράθυρο δεδομένων. Στο πλαίσιο ετερογενών ενσύρματων/ασύρματων περιβαλλόντων, τα αποτελέσματά μας δείχνουν ότι, υπάρχουν λίγα για να επιλέξεις μεταξύ αυτών των τριών εκδόσεων από άποψη ανταλλαγής ενέργειας/ρυθμοαπόδοσης. Σε εξεζητημένα σενάρια των μάλλον επιτακτικών συνθηκών λάθους, το Tahoe, αν και η παλαιότερη έκδοση, εκτελεί καλύτερα από το Reno και το NewReno με σεβασμό στην ενέργεια και στη ρυθμοαπόδοση. Το Reno και το NewReno, εντούτοις, μπορούν να αποδώσουν μικρότερα επιτεύγματα ρυθμοαπόδοσης όταν αμβλύνονται οι συνθήκες λάθους.

3.6.2 Επισκόπηση TCP

Ο μηχανισμός ελέγχου λάθους TCP έχει κάποιες ικανότητες ενεργειακής διατήρησης. Σε απάντηση στις πτώσεις τμήματος, το TCP μειώνει το μέγεθος παραθύρου του και επομένως διατηρεί την προσπάθεια μετάδοσης. Ο στόχος εδώ δεν είναι μόνο να ανακουφιστούν οι κορεσμένοι διακόπτες, αλλά και να αποφευχθεί η περιττή αναμετάδοση που υποβιβάζει την απόδοση του πρωτοκόλλου. Τα TCP Tahoe, Reno και NewReno, όλα χρησιμοποιούν ουσιαστικά τον ίδιο αλγόριθμο στο δέκτη, αλλά εφαρμόζουν διαφορετικές παραλλαγές στη διαδικασία μετάδοσης στον αποστολέα. Ο δέκτης δέχεται τα τμήματα από την ακολουθία, αλλά τα παραδίδει σύμφωνα με τα ανωτέρω πρωτόκολλα. Δημοσιεύει ένα μέγεθος παραθύρου, και ο αποστολέας εξασφαλίζει ότι ο αριθμός των μη αναγνωρισμένων bytes να μην υπερβαίνει αυτό το μέγεθος. Για κάθε σωστά λαμβανόμενο τμήμα, ο δέκτης στέλνει μία αναγνώριση που περιλαμβάνει τον αριθμό ακολουθίας προσδιορίζοντας το επόμενο στη σειρά τμήμα. Ο αποστολέας εφαρμόζει ένα παράθυρο συμφόρησης που καθορίζει το μέγιστο αριθμό διαβιβασθέντων αλλά μη αναγνωρισμένων τμημάτων που επιτρέπονται. Αυτό το προσαρμοστικό παράθυρο μπορεί να αυξηθεί και να μειωθεί, αλλά δεν υπερβαίνει ποτέ του το παράθυρο που δημοσιεύει ο δέκτης. Το TCP εφαρμόζει διαβαθμισμένες πολλαπλασιασμένες και η προσθετικές αυξήσεις στο παράθυρο συμφόρησης του αποστολέα. Οι εκδόσεις πρωτοκόλλου διαφέρουν μεταξύ τους ουσιαστικά στο ότι το παράθυρο συμφόρησης ελέγχεται ανάλογα με την απόκριση των αναγνωρίσεων και των διαλειμμάτων, και στη μέθοδο με την οποία τα παραδοθέντα ή ελλείποντα τμήματα αναγνωρίζονται.

3.6.2.1 TCP Tahoe, Reno, και NewReno

Ο μηχανισμός ελέγχου λαθών του TCP προσανατολίζεται πρώτα στον έλεγχο της συμφόρησης. Έλεγχος συμφόρησης μπορεί να είναι ευεργετικός επίσης στη ροή που τον δοκιμάζει, αφού η αποφυγή της περιττής αναμετάδοσης μπορεί να οδηγήσει σε καλύτερη ρυθμοαπόδοση [35]. Η βασική ιδέα είναι για την κάθε πηγή να καθορίσει πόση χωρητικότητα είναι διαθέσιμη στο δίκτυο, έτσι ώστε να ξέρει πόσα τμήματα μπορεί ακίνδυνα να έχει προς διακίνηση. Το TCP χρησιμοποιεί τις

αναγνωρίσεις για να δώσει το ρυθμό στη μετάδοση των τμημάτων και να ερμηνεύσει τα γεγονότα διαλείμματος ως ένδειξη της συμφόρησης. Σε απάντηση, ο αποστολέας TCP μειώνει το ποσοστό μετάδοσης με τη συρρίκνωση του παράθυρό του. Το Tahoe και το Reno είναι οι δύο πιο κοινές εφαρμογές αναφοράς για το TCP. Το NewReno είναι μια τροποποιημένη έκδοση Reno που προσπαθεί να λύσει μερικά από τα προβλήματα απόδοσης του Reno όταν πολλαπλά πακέτα πέφτουν από ένα ενιαίο παράθυρο δεδομένων. Έως τώρα αυτές οι τρεις εκδόσεις του TCP μοιράζονται το ίδιο πρόβλημα όσον αφορά την αναμετάδοση. Αυτοί είτε αναμεταδίδουν το πολύ ένα πεταγμένο πακέτο ανά RTT (Round Trip Time), είτε αναμεταδίδουν τα πακέτα που μπορεί να έχουν ήδη παραδοθεί επιτυχώς.

Ο αλγόριθμος ελέγχου συμφόρησης του **TCP Tahoe** περιλαμβάνει αργή εκκίνηση, αποφυγή συμφόρησης, και γρήγορη αναμετάδοση [31, 35]. Αυτό επίσης εφαρμόζει μια εκτίμηση, βασισμένη στο RTT, της αναμετάδοσης διαλείμματος. Στο μηχανισμό γρήγορης αναμετάδοσης, ο αριθμός διαδοχικών (το κατώτατο όριο είναι συνήθως στο τρία), πολλαπλών αναγνωρίσεων (dacks) μεταφέρει τον ίδιο αριθμό ακολουθίας που προκαλείται από μια αναμετάδοση χωρίς να περιμένει το σχετικό γεγονός του διαλείμματος να εμφανιστεί. Η στρατηγική ρύθμισης του παραθύρου για αυτό το «πρόωρο διάλειμμα» είναι όπως για το κανονικό διάλειμμα δηλαδή εφαρμόζεται η αργή. Το πρόβλημα, εντούτοις, είναι ότι η αργή εκκίνηση δεν είναι πάντα αποδοτική, ειδικά εάν το λάθος ήταν καθαρά προσωρινό ή τυχαίο στη φύση, και μη ανυποχώρητος. Σε τέτοια περίπτωση η σμίκρυνση του συμφορημένου παραθύρου είναι, στην ουσία, περιττή, και καθιστά το πρωτόκολλο πλήρως ανίκανο να χρησιμοποιήσει το διαθέσιμο εύρος ζώνης του καναλιού επικοινωνίας κατά τη διάρκεια της επόμενης φάσης της επανα-επέκτασης παραθύρου.

Το **TCP Reno** εισάγει τη γρήγορη αποκατάσταση από κοινού με τη γρήγορη αναμετάδοση. Η ιδέα πίσω από τη γρήγορη αποκατάσταση είναι ότι το *dack* είναι μια ένδειξη του διαθέσιμου εύρους ζώνης του καναλιού από τότε που ένα τμήμα έχει παραδοθεί επιτυχώς. Αυτό, στη συνέχεια, σημαίνει ότι το παράθυρο συμφόρησης (cwnd) πρέπει στην πραγματικότητα να αυξηθεί. Έτσι, για κάθε *dack*, το cwnd είναι αυξανόμενο κατά ένα. Λαμβάνοντας τον αριθμό κατώτατων ορίων των *dacks* προκαλείται η γρήγορη αποκατάσταση: ο αποστολέας αναμεταδίδει ένα τμήμα, μειώνει στο μισό το cwnd, και θέτει το κατώτατο όριο συμφόρησης στην τιμή του cwnd. Κατόπιν, αντί της εισόδου στη διαδικασία της αργής εκκίνησης όπως στο Tahoe, ο αποστολέας αυξάνει το cwnd του και αυτή η αύξηση είναι ίση με από τον

αριθμό του κατώτατου ορίου του *dack*. Έκτοτε, και για όσο χρονικό διάστημα παραμένει ο αποστολέας στη διαδικασία της γρήγορης αποκατάστασης, το *cwnd* αυξάνεται κατά ένα για κάθε ένα πρόσθετο *dack* που λαμβάνεται. Αυτή η διαδικασία καλείται «διογκώνοντας» το *cwnd*. Το στάδιο γρήγορης αποκατάστασης ολοκληρώνεται όταν παραλαμβάνεται μια αναγνώριση (*ack*) για τα νέα δεδομένα. Τότε ο αποστολέας θέτει την τιμή του *cwnd* στην ίδια τιμή με την κατώτατη τιμή της τρέχουσας συμφόρησης («ξεφουσκώνοντας» το παράθυρο), και ρυθμίζει εκ νέου το μετρητή του *dack*. Στη γρήγορη αποκατάσταση, το *cwnd* είναι κατά τέτοιο τρόπο αποτελεσματικά ρυθμισμένο στη μισή από την προηγούμενη τιμή του με την παρουσία των *dacks*, παρά να εκτελέσει την αργή εκκίνηση όπως κάνει για έναν γενικό διάλειμμα αναμετάδοσης. Η γρήγορη αποκατάσταση που έχει το TCP Reno μπορεί να είναι αποτελεσματική όταν υπάρχει μόνο μία πτώση τμήματος από το ένα παράθυρο δεδομένων, λαμβάνοντας υπόψη το γεγονός ότι το Reno αναμεταδίδει το ανώτερο ένα πεταγμένο τμήμα ανά RTT. Το πρόβλημα με το μηχανισμό είναι ότι δεν βελτιστοποιείται για το πολλαπλές πτώσεις πακέτων από ένα ενιαίο παράθυρο, και αυτό θα μπορούσε να έχει αρνητική επίπτωση στην απόδοση.

Το **NewReno** εξετάζει το πρόβλημα των πτώσεων πολλαπλών τμημάτων από ένα ενιαίο παράθυρο. Στην πραγματικότητα, μπορεί να αποφύγει πολλές από τις αναμεταδόσεις διαλειμμάτων του Reno. Η τροποποίηση του NewReno εισάγει μια στρατηγική μερικής αναγνώρισης στη γρήγορη αποκατάσταση. Μια *μερική αναγνώριση* ορίζεται ως ένα *ack* για το νέο δεδομένο το οποίο δεν αναγνωρίζει όλα τα τμήματα που ήταν σε τροχιά τη στιγμή που η γρήγορη αποκατάσταση άρχιζε. Είναι κατά συνέπεια μια ένδειξη ότι δεν σταλθήκαν όλα τα δεδομένα που είχαν παραληφτεί πριν μπει στη διαδικασία γρήγορης αποκατάστασης. Στο Reno, το μερικό *ack* προκαλεί την έξοδο από τη γρήγορη αποκατάσταση. Στο NewReno, αυτό είναι μια ένδειξη ότι (τουλάχιστον) ένα τμήμα λείπει και πρέπει να αναμεταδοθεί. Κατά συνέπεια, όταν πολλαπλά πακέτα χάνονται από ένα παράθυρο δεδομένων, το NewReno μπορεί να αποκατασταθεί χωρίς να περιμένει την αναμετάδοση διαλείμματος. Σημειώστε ότι ο αλγόριθμος αναμεταδίδει ακόμα το πιο πολύ ένα τμήμα ανά RTT. Το NewReno μπορεί να βελτιώσει τη ρυθμοαπόδοση κάτω από πολλαπλές πτώσεις τμημάτων από ένα ενιαίο παράθυρο. Εντούτοις, η αναμετάδοση που προκαλείται από ένα μερικό *ack* πιθανώς να είναι για ένα καθυστερημένο παρά χαμένο τμήμα. Κατά συνέπεια, η στρατηγική διακινδυνεύει κάνοντας για το τμήμα

πολλαπλές επιτυχείς μεταδόσεις, το οποίο μπορεί να έχει σοβαρή επίπτωση στην ενεργειακή του αποδοτικότητα χωρίς αντισταθμιστικό κέρδος σε ρυθμοαπόδοση.

3.6.3 Περιβάλλον Εξέτασης και Μεθοδολογία

Οι τρεις εκδόσεις του TCP εφαρμόστηκαν χρησιμοποιώντας το πλαίσιο πρωτοκόλλου x-πυρήνων (x-kernel protocol framework) [41]. Κάναμε τις δοκιμές προσομοιώνοντας ένα περιβάλλον αρκετά χαμηλού εύρους ζώνης, αφού πρώτιστα μας ενδιαφερόμαστε για ενσύρματα/ασύρματα ετερογενή περιβάλλοντα. Οι δοκιμές πραγματοποιήθηκαν σε μια ενιαία σύνοδο, με τον πελάτη και τον server να τρέχουν σε δύο άμεσα συνδεδεμένους αφοσιωμένους οικοδεσπότες, ώστε να αποφευχθούν οι απρόβλεπτες καταστάσεις με τη διαστρέβλωση των αποτελεσμάτων στην απόδοση του πρωτοκόλλου.

Προκειμένου να εξομοιώσουμε τις συνθήκες λάθους, αναπτύξαμε ένα «εικονικό πρωτόκολλο», VDELDROP, το οποίο ήταν διαμορφωμένος μεταξύ του TCP και της IP. Ο κεντρικός μηχανισμός του VDELDROP αποτελείται από έναν συνεχή χρόνο 2-επίπεδων σε κλίμακα Markov. Κάθε επίπεδο έχει έναν μέσο χρόνο παραμονής m_i , και ένα ποσοστό πτώσης r_i , ($i=1, 2$) οι τιμές του οποίου τίθενται από το χρήστη. Το ποσοστό πτώσης r_i , παίρνει μια αξία μεταξύ 0 και 1, και καθορίζει το ποσοστό των τμημάτων που πέφτουν κατά τη διάρκεια του i επιπέδου. Κατά συνέπεια, όταν αυτό επισκέπτεται το i επίπεδο, ο μηχανισμός παραμένει εκεί για ένα μέγεθος χρόνου ραγδαία κατανομημένο με μέσο m_i , κατά τη διάρκεια του οποίου ρίχνει μια αναλογία r_i από τα τμήματα που διαβιβάζονται, και έπειτα μεταβαίνει στο άλλο επίπεδο.

Στα πειράματά μας εναρμονίσαμε τα δύο επίπεδα ώστε να έχουν ίσο μέσο χρόνο παραμονής. Φυσικά, η τιμή αυτού του μέσου χρόνου ποικίλει από πείραμα σε πείραμα αλλά πάντα τέθηκε ίση για τα δύο επίπεδα. Επιπλέον, το ένα επίπεδο πάντα διαμορφωνόταν με μηδενικό ποσοστό πτώσης. Κατά συνέπεια, μιμούμενοι καταστάσεις λάθους κατά τη διάρκεια ενός δεδομένου πειράματος εναλλάχτηκαν μεταξύ τους οι «On» και «Off» φάσεις κατά τη διάρκεια των οποίων οι ενέργειες που προκαλούν πτώση ήταν σε ισχύ και ανεστάλησαν, αντίστοιχα. Καταστάσεις λάθους των διαφόρων εντάσεων, της διατήρησης και της διάρκειας θα μπορούσαν με αυτό

τον τρόπο να μιμηθούν, εξαρτώμενες από την επιλογή του μέσου χρόνου παραμονής στο επίπεδο και από το ποσοστό πτώσης της φάσης «On».

3.6.4 Αποτελέσματα και Συζήτηση

Όλες οι αναφερόμενες δοκιμές έγιναν χρησιμοποιώντας στοιχεία συνόλων 5MByte (5.242.880 bytes) για τη μετάδοση. Ο σκοπός των δοκιμών ήταν να αξιολογηθεί η συμπεριφορά των πρωτοκόλλων κατά τη διάρκεια των αλλαγών στο περιβάλλον δικτύου, όπως είναι η προσπονημένη συμφόρηση, και τα λάθη μετάδοσης λόγω της διαφορετικής έντασης και διάρκειας. Πήραμε μετρήσεις από το συνολικό χρόνο σύνδεσης και το συνολικό αριθμό bytes που διαβιβαστήκαν (δηλ. συμπεριλαμβανομένων και των υπερυψωμένων μεταδόσεων του πρωτοκόλλου ελέγχου και της αναμετάδοσης τμήματος δεδομένων, κλπ). Και οι δύο παράγοντες έχουν σημαντικές επιπτώσεις στη δαπάνη ενέργειας καθώς επίσης και στη ρυθμοαπόδοση. Στον [πίνακα 9](#) παρουσιάζουμε τα αποτελέσματα για το VDELDROP με μέση διάρκεια φάσης On/Off 1 δευτερόλεπτο, ενώ για τον [πίνακα 10](#) αυτή η μέση διάρκεια είναι 10 δευτερόλεπτα. Προκειμένου να αναπαρουσιασθεί η (μετάδοση) δαπάνη σε ενέργεια που απαιτείται για την ολοκλήρωση μια αξιόπιστης μετάδοσης κάτω από διαφορετικές συνθήκες, χρησιμοποιούμε το **Overhead** (επιβάρυνση λειτουργίας) ως μετρικό σύστημα. Αυτός είναι ο συνολικός *επιπλέον* αριθμός των bytes που το πρωτόκολλο διαβιβάζει, εκφρασμένος σε επί τοις εκατόν ποσοστό, επιπλέον των 5 MByte που παραδίδονται κατά τη διαδικασία στο δέκτη, από την έναρξη έως τη λήξη της σύνδεσης. Το **Overhead** υπολογίζεται από τον τύπο: **Overhead = 100*(Total- Base)/5Mbytes**, όπου,

- **Base** είναι ο αριθμός bytes που παραδίδονται στο υψηλού επιπέδου πρωτόκολλο του δέκτη. Είναι ρυθμισμένος στα 5Mbyte για όλες τις δοκιμές.
- **Total** είναι το σύνολο όλων των bytes που διαβιβάστηκαν από τον αποστολέα και το δέκτη μέσα από τα στρώματα μεταφορών, και δίνονται στη στήλη **Total Bytes**. Αυτό περιλαμβάνει το πρωτόκολλο ελέγχου επιβάρυνσης λειτουργίας, την αναμετάδοση κομματιών δεδομένων, όπως επίσης και τα παραδοθέντα δεδομένα.

Ο χρόνος που απαιτείται για το overhead να ολοκληρώσει την αξιόπιστη μετάδοση υπό διαφορετικές συνθήκες δίνεται στη στήλη **Time**. Η μετρημένη απόδοση των πρωτοκόλλων δίνεται στη στήλη **Goodput** που χρησιμοποιεί τον τύπο: **Goodput =**

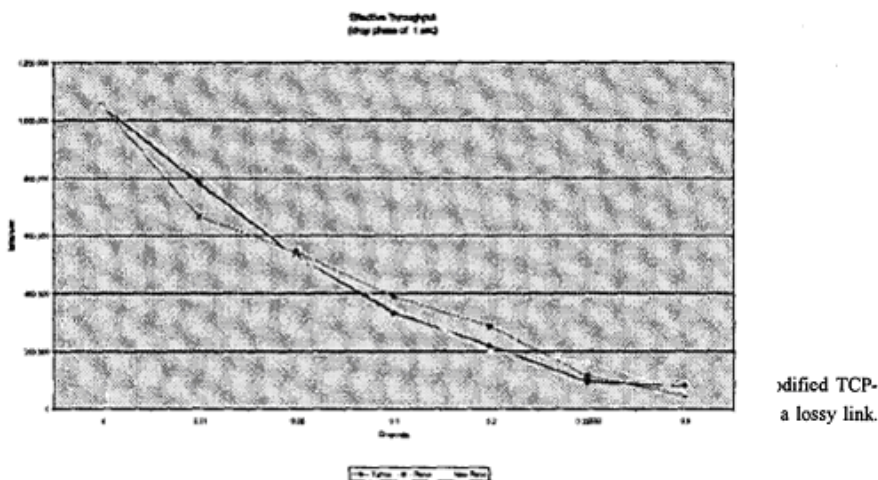
Original Data/Connection Time, όπου,

- **Original Data** (αρχικά δεδομένα) είναι το σύνολο δεδομένων των 5MBytes.
- **Connection Time** (χρόνος σύνδεσης) είναι το χρονικό διάστημα που απαιτείται για την ολοκλήρωση της παράδοσης δεδομένων, από την έναρξη έως τη λήξη της σύνδεσης.

Για το VDELDROP, το **DROP Rate** (ποσοστό πτώσης) που αναφέρθηκε είναι το ποσοστό πτώσης των τμημάτων κατά τη διάρκεια των «On» φάσεων, και όχι το υπολογισμένο κατά μέσο όρο, ολικό ποσοστό πτώσης στις φάσεις «On/Off». Η καταχώριση **0** στη **DROP Rate** στήλη υποδηλώνει κατάσταση χωρίς λάθη.

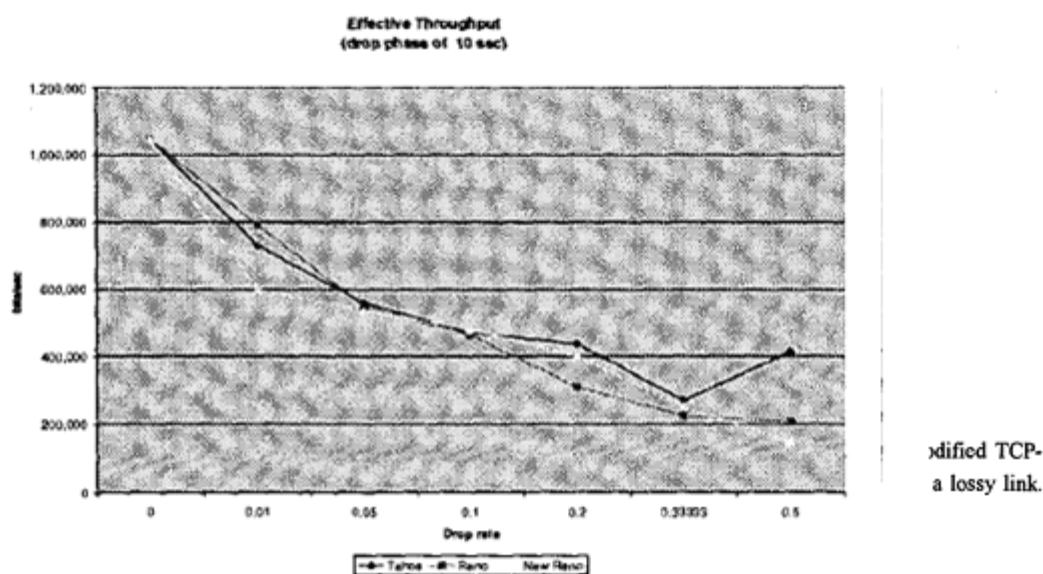
3.6.4.1 Αποτελεσματική Απόδοση Ρυθμοαπόδοσης (Goodput)

Από το παρακάτω [σχήμα 27](#) μπορεί να παρατηρηθεί ότι για συχνές On/Off αλλαγές φάσης (εδώ, με 1 δευτερόλεπτο μέση διάρκεια φάσης) το Tahoe, το Reno, και το NewReno παρουσιάζουν παρόμοια συμπεριφορά. Οι διαφορές είναι ελάχιστες αφού τα πρωτόκολλα δεν έχουν την ευκαιρία να καταδείξουν τα χαρακτηριστικά τη συμπεριφοράς τους επεκτείνοντας τα μεγέθη του παραθύρου τους και εφαρμόζοντας τους αλγορίθμους ελέγχου συμφόρησής τους για μία ικανοποιητική χρονική περίοδο (το [σχήμα 27](#) παρακάτω & ο [πίνακα 9](#) στο Παράρτημα).



Σχήμα 27: Αποτελεσματική Ρυθμοαπόδοση (Goodput) με μέση On/Off φάση 1 second.

Γενικά μία παρατήρηση για τον έλεγχο συμφόρησης είναι ότι δεν επιτρέπει τις γρήγορες αυξήσεις παραθύρων ως στρατηγική αποκατάστασης, αλλά μόνο τις μειώσεις. Η ιδέα είναι να ανακουφιστούν οι συμφορημένοι δρομολογητές και να αποφευχθεί η πλημμύρα του δικτύου. Το αποτέλεσμα είναι ότι η γενική απόδοση ρυθμοαπόδοσης υποβιβάζεται σημαντικά, και αντιστρόφως ανάλογα για το ποσοστό λάθους. Σε μια ιδανική κατάσταση, η στρατηγική αποκατάστασης λάθους θα ήταν να αυξηθεί αμέσως μετά από τη φάση λάθους, ή ακόμα και κατά τη διάρκεια της ίδιας αυτής της φάσης εφ' όσον το ποσοστό λάθους είναι χαμηλά [9], [10]. Από το [σχήμα 28](#) μπορούμε να δούμε ότι για τα χαμηλά ποσοστά λάθους (0.01, ή 1%), το Reno βελτιώνει την απόδοσή του σχετικά με τις άλλες δύο εκδόσεις καθώς η διάρκεια φάσης αυξάνεται. Με μέση διάρκεια φάσης στα 10 δευτερόλεπτα, το Reno έχει τον αρκετό χρόνο να επεκτείνει το παράθυρο στο μέγιστο βαθμό, όμως, αντίθετα από Tahoe, όταν εμφανίζεται ένα λάθος αυτό δεν ακολουθεί πάντα την αργή εκκίνηση, αλλά αντ' αυτού μερικές φορές εξασκεί την γρήγορη αποκατάσταση.



Σχήμα 28: Αποτελεσματική Ρυθμοαπόδοση (Goodput) με μέση On/Off φάση 10 seconds.

Αυτή η συμπεριφορά του επιτρέπει να εξερευνησει τα καλύτερα σε ευκαιρίες παράθυρα για τις μεταδόσεις χωρίς λάθη, και ως εκ τούτου να επιτύχει ελαφρώς καλύτερη *goodput* (αποτελεσματική απόδοση ρυθμοαπόδοσης) από το Tahoe.

Αυτή η ελαφρώς ευνοϊκή συμπεριφορά του Reno

βαθμιαία εξασθενίζει. Δεν είναι καλύτερο από τις άλλες δύο εκδόσεις όταν το ποσοστό λάθους είναι στο 10%, και ξεπερνιέται από το Tahoe και το NewReno όταν είναι στο 20%. Σε αυτήν την περίπτωση του ποσοστού λάθους στο 20%, το Reno επιτυγχάνει μόνο 70% της *goodput* που επιτυγχάνεται από το Tahoe ([σχήμα 28](#) & Ρύθμιση Δοκιμής 4² του [πίνακα 10](#)).

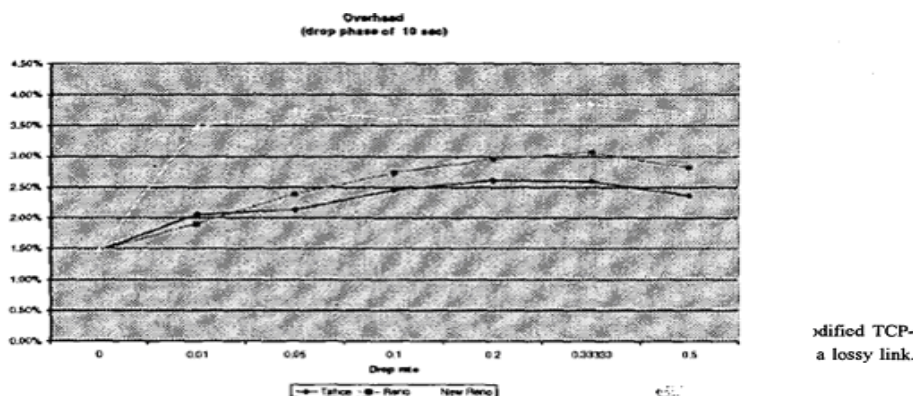
² δηλ. το σύνολο των τριών δοκιμών αριθμείται ως 4.1, 4.2 & 4.3

Έκτοτε το Tahoe επιδεικνύει καλύτερη απόδοση, με συνέπεια το σημαντικό κέρδος στη σχετική *goodput* κατά την περίπτωση με το υψηλότερο ποσοστό λάθους που αναφέρθηκε στις δοκιμές ([σχήμα 28](#) & Ρύθμιση Δοκιμής 6 του [πίνακα 10](#)). Αυτό επιβεβαιώνει τον ισχυρισμό μας ότι ένας συντηρητικός τρόπος μετάδοσης συμπεριφέρεται ευνοϊκά προς τη *goodput* όταν οι συνθήκες λάθους είναι μάλλον εντατικοί, ενώ μια επιθετικότερη συμπεριφορά υπό τους ίδιες συνθήκες έχει ως αποτέλεσμα τη χειρότερη *goodput* απόδοση λόγω της αναμετάδοσης και της ρύθμισης διαλείμματος. Για την περίπτωση που το ποσοστό λάθους είναι στο 50% και η φάση «On/Off» έχει μέση διάρκεια 10 δευτερόλεπτα, το Tahoe αμέσως συρρικνώνει το παράθυρό του και με παθητικότητα δεν έχει την πιθανότητα να το επανα-επεκτείνει. Το Reno εισέρχεται σε φάση γρήγορης αποκατάστασης, συνεχίζοντας τη μετάδοση δεδομένων μέχρι να εμφανισθεί ένα διάλειμμα. Το NewReno ερμηνεύει τις αναγνωρίσεις κατά τη διάρκεια της γρήγορης αποκατάστασης ως μερικές αναγνωρίσεις, και ως εκ τούτου αναμεταδίδει προκειμένου να ανακτήσει πιο γρήγορα από τις πολλαπλές απώλειες. Κατά συνέπεια, παρόλα τα περισσότερα bytes που εγχέονται στο δίκτυο από το Reno και το New Reno, η *goodput* που επιτυγχάνεται από το Tahoe είναι σχεδόν διπλάσια ([σχήμα 28](#) & Ρύθμιση Δοκιμής 6 του [πίνακα 10](#)).

3.6.4.2 Ενεργειακά Ζητήματα

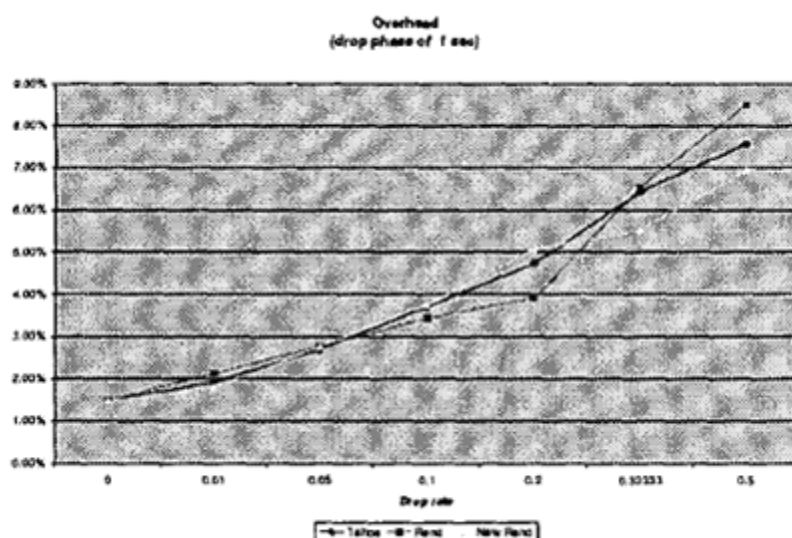
Ο υπολογισμός συγκεκριμένων ενεργειακών λειτουργιών θα μπορούσε να παρέχει μέτρα αποταμίευση της ενέργειας για τις εκδόσεις TCP πιθανόν αμφίβολης ακρίβειας. Η δαπάνη ενέργειας προσδιορίζεται από το μέσο, τον τρόπο, και τη χρησιμότητα. Μπορεί να μετρηθεί με ακρίβεια μόνο σε μια συγκεκριμένη συσκευή,

τρέχοντας κάθε μια έκδοση πρωτοκόλλου χωριστά, και καταγράφοντας την ενέργεια της μπαταρίας που καταναλώνεται ανά έκδοση. Η ενέργεια καταναλώνεται σε διαφορετικά ποσοστά κατά τη διάρκεια των διάφορων σταδίων επικοινωνίας, και το μέγεθος δαπανών εξαρτάται από τη κείμενη λειτουργία. Παραδείγματος χάριν, ο μεταδότης/δέκτης χρησιμοποιούν διαφορετικά ποσά ενέργειας όταν, αντίστοιχα, στέλνουν ή λαμβάνουν τμήματα δεδομένων, περιμένουν αυτά τα τμήματα να φτάσουν, και είναι αδρανή. Ως εκ τούτου, ο ρυθμός [13], μια εκτίμηση των πρόσθετων ενεργειακών δαπανών δεν μπορεί να βασιστεί απλώς στη *overhead* των byte λόγω της αναμετάδοσης, αφού ο συνολικός χρόνος σύνδεσης μπορεί να έχει επεκταθεί περιμένοντας ή προσπαθώντας για μία μέτρια σε ρυθμό μετάδοσης. Από την άλλη, η εκτίμηση δεν μπορεί να βασιστεί ούτε στο συνολικό χρόνο σύνδεσης, αφού οι ευδιάκριτες διαδικασίες που εκτελούνται κατά τη διάρκεια εκείνου του χρόνου (π.χ. μετάδοση εναντίον αδράνειας) καταναλώνουν διαφορετικά επίπεδα ενέργειας. Εντούτοις, η δυνατότητα για αποταμίευση ενέργειας μπορεί να μετρηθεί από το *συνδυασμό* του χρόνου και byte επιτυγχάνοντας αποταμίευση *overhead*. Με βάση αυτά τα μετρικά μεγέθη, μπορούμε να υπολογίσουμε τα χαμηλότερα όρια για την κατανάλωση ενέργειας. Δεδομένου ότι δεν εμπλέκεται μία κατάσταση αδράνειας στις διαδικασίες του TCP (ο αποστολέας/δέκτης είτε αποστέλλει τμήμα/αναγνώριση, ή περιμένει αναγνώριση/τμήμα), ένα χαμηλότερο όριο μπορεί να καθορισθεί βασισμένο στην υπόθεση ότι ο πρόσθετος χρόνος που χρησιμοποιείται από ένα πρωτόκολλο ξοδεύεται σε κατάσταση αναμονής. Όπως με τη *goodput*, η διαφορά μεταξύ των τριών εκδόσεων TCP όσον αφορά τα γενικά έξοδα είναι επίσης ασήμαντη κάτω από γρήγορες μεταβαλλόμενες καταστάσεις με μέση διάρκεια φάσης «On/Off» στο 1 δευτερόλεπτο (δείτε το [σχήμα 29](#)).



Σχήμα 29: Overhead με μέση On/Off φάση 1 second.

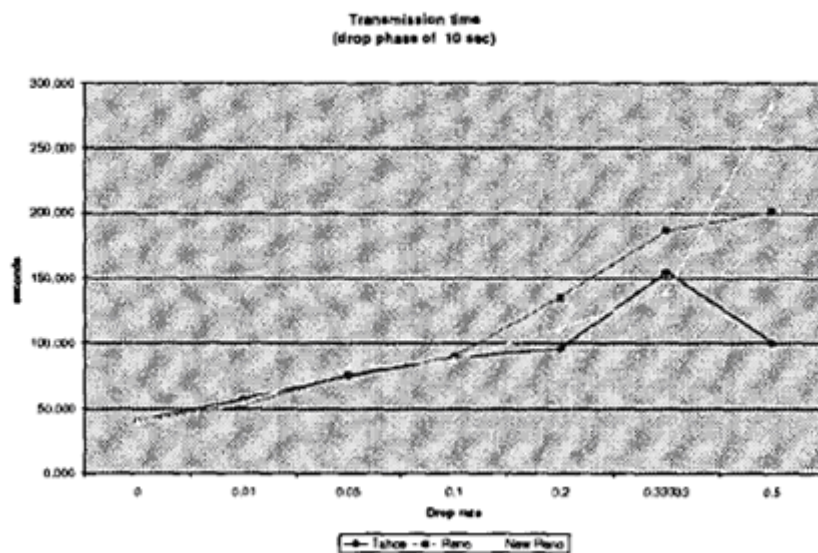
Αυτό παραμένει κατά ένα μεγάλο μέρος αληθινό για το Tahoe και το Reno σε μεγαλύτερες φάσεις «On/Off» με μέση διάρκεια 10 δευτερόλεπτα, ειδικά όταν πρόκειται για ποσοστά ήπιων λαθών (σχήμα 30), αν και πρέπει να σημειωθεί ότι τα σχετικά γενικά έξοδα λόγω της αναμετάδοσης είναι πολύ χαμηλότερα για διάρκεια φάσης 10 δευτερολέπτων από ότι για διάρκεια φάσης 1 δευτερόλεπτου (συγκρίνετε τα σχήματα 29 & 30, και από τα γενικά έξοδα Ρύθμισης Δοκιμής 6 των πινάκων 9 & 10). Ενδιαφέρουσες παρατηρήσεις μπορούν να γίνουν σχετικά με την κατανάλωση ενέργειας των τριών εκδόσεων πρωτοκόλλου από τις δοκιμές με μέση διάρκεια 10 δευτερόλεπτα. Καταρχήν, το NewReno σπαταλά ένα αρκετά μεγαλύτερο ποσό ενέργειας στην αναμετάδοση και το χρόνο εκτεταμένης επικοινωνίας σε όλα τα ποσοστά λάθους που αναφέρονται στον [πίνακα 10](#). Κατά τη διάρκεια αυτών των παρατεταμένων «On» φάσεων, προσπαθεί να ανακτήσει από τις πολλαπλές απώλειες όπως έκανε προηγουμένως κατά τη διάρκεια της φάσης του 1 δευτερολέπτου, αλλά τώρα η επιθετική αναμετάδοση η οποία προκύπτει από τις μερικές αναγνωρίσεις δεν είναι πάντα επιτυχής όταν η φάση λάθους εμμένει (σχήμα 30).



Σχήμα 30: Overhead με μέση On/Off φάση 10 second.

Αφετέρου, στην περίπτωση που το ποσοστό λάθους είναι στο 20% το Reno καταβάλει μεγαλύτερη προσπάθεια στην μετάδοση και επιτυγχάνει χειρότερη ρυθμοαπόδοση από το Tahoe. Σε αυτή τη περίπτωση, το παράθυρο του Reno επεκτάθηκε σημαντικά και η σχετική επιθετική αναμετάδοση πραγματοποιείται για μια στιγμή, αν και βρισκόμαστε σε μια συνεχή επιρρεπή σε λάθη φάση. Το Tahoe,

από την άλλη, χρησιμοποιεί την αργή εκκίνηση αντί της γρήγορης αποκατάστασης, συρρικνώνει το παράθυρό του αμέσως, ως εκ τούτου αποφεύγει τη μείωση τμημάτων που ειδάλλως θα συνεπαγόταν την αναμετάδοση τους, και συνεχίζει την επέκταση παραθύρου όταν το επιτρέπουν οι συνθήκες. Ο χρόνος επικοινωνίας παρατείνεται για το Reno κατά 40 δευτερόλεπτα (δηλ. κατά περίπου 42%) περισσότερο απ' ό τι για το Tahoe (σχήμα 31 και Ρύθμιση Δοκιμής 4 του πίνακα 10), και ως εκ τούτου περισσότερη ενέργεια χρησιμοποιείται κατά τη διάρκεια της πρόσθετων μετάδοσης και της κατάστασης αναμονής. Εντούτοις, εάν υποθέσουμε ότι το Reno παραμένει σε μία κατάσταση αναμονής σε όλο το πρόσθετο χρόνο των 40 δευτερολέπτων, καταναλώνοντας ενέργεια σε ένα ποσοστό του $1/10^{00}$ της καταναλωθείσας από το Tahoe κατά τη διάρκεια του χρόνου σύνδεσής του, αυτό θα οδηγούσε σε ένα χαμηλότερο όριο πρόσθετης ενεργειακής δαπάνης κατά 4.2% έναντι του Tahoe.



Σχήμα 31: Χρόνος Σύνδεσης με μέση On/Off φάση 10 seconds.

Σε μία κατάσταση ποσοστού λάθους 50%, η υπερψωμένη ενέργεια του Reno είναι σχετική με του Tahoe μπορεί να μετρηθεί από τη πρόσθετη κατά 0.50% της υπερψωμένης αναμετάδοσης και τα πρόσθετα 100 δευτερόλεπτα από τον παραταθέντος χρόνου επικοινωνίας (σχήματα 30 & 31, και Ρύθμιση Δοκιμής 6 του πίνακα 10). Εάν πάλι υποθέτουμε ότι το Reno ξοδεύει μόνο το ίδιο μέγεθος χρόνου στη κατάσταση μετάδοσης όπως κάνει το Tahoe, και έτσι καταναλώνει ενέργεια κατά τη διάρκεια όλου του πρόσθετου χρόνου των 100 δευτερολέπτων σε μία κατάσταση αναμονής, ένα μικρότερο μέγεθος πρόσθετης ενεργειακής δαπάνης της τάξης του

10% περισσότερο από την ενέργεια που καταναλώνεται από το Tahoe μπορεί να υπολογιστεί όπως έγινε παραπάνω (δείτε τη Ρύθμιση Δοκιμής 6 του [πίνακα 10](#)).

Στην περίπτωση του NewReno είναι ακόμα χειρότερη. Συμπεριφερόμενο περισσότερο επιθετικά σε περίπτωση πολλαπλών απωλειών, αυτό προσωρινά (δηλ. μέχρι ένα περιστατικό διαλείμματος να εμφανιστεί) μεγιστοποιεί την προσπάθεια αναμετάδοσης, αν και οι πιθανότητες να παραλάβει κατευθείαν τμήματα είναι ελάχιστες κάτω από μία επίμονη φάση λάθους μέσης διάρκειας 10 δευτερολέπτων με ποσοστό πτώσης 50%.

Appendix: Tables

Test	Protocol	Drop rate	Total Bytes	Time	Goodput	Overhead
0.1	Tahoe	0	5,320,228	40.000	1,048,576	1.48%
0.2	Reno	0	5,320,228	40.000	1,048,576	1.48%
0.3	New Reno	0	5,320,228	40.000	1,048,576	1.48%
1.1	Tahoe	0.01	5,343,790	53.591	782,650	1.92%
1.2	Reno	0.01	5,354,292	62.970	666,079	2.13%
1.3	New Reno	0.01	5,337,030	51.003	822,369	1.80%
2.1	Tahoe	0.05	5,384,054	79.198	529,598	2.69%
2.2	Reno	0.05	5,387,814	77.127	543,820	2.76%
2.3	New Reno	0.05	5,387,814	79.734	526,037	2.76%
3.1	Tahoe	0.1	5,436,706	126.023	332,822	3.70%
3.2	Reno	0.1	5,423,210	107.944	388,561	3.44%
3.3	New Reno	0.1	5,440,152	118.079	355,213	3.76%
4.1	Tahoe	0.2	5,491,350	192.219	218,205	4.74%
4.2	Reno	0.2	5,449,024	146.059	287,164	3.93%
4.3	New Reno	0.2	5,509,490	210.476	199,277	5.09%
5.1	Tahoe	0.33333	5,580,730	445.242	94,203	6.44%
5.2	Reno	0.33333	5,585,810	374.421	112,021	6.54%
5.3	New Reno	0.33333	5,533,606	257.884	162,643	5.55%
6.1	Tahoe	0.5	5,639,912	532.897	78,708	7.57%
6.2	Reno	0.5	5,688,910	870.334	48,192	8.51%
6.3	New Reno	0.5	5,606,390	697.130	60,165	6.93%

Table 1. Test Results with mean On/Off phase duration 1 second

Πίνακας 9. Αποτελέσματα του test με μέση διάρκεια φάσης On/Off 1 second.

Test	Protocol	Drop rate	Bytes sent	Time (s)	Goodput	Overhead
0.1	Tahoe	0	5,320,228	40.00	1,048,576	1.48%
0.2	Reno	0	5,320,228	40.00	1,048,576	1.48%
0.3	New Reno	0	5,320,228	40.00	1,048,576	1.48%
1.1	Tahoe	0.01	5,350,756	57.52	729,215	2.06%
1.2	Reno	0.01	5,342,788	53.02	791,020	1.91%
1.3	New Reno	0.01	5,425,165	70.02	598,996	3.48%
2.1	Tahoe	0.05	5,354,740	75.06	558,766	2.13%
2.2	Reno	0.05	5,368,254	76.42	548,857	2.39%
2.3	New Reno	0.05	5,439,765	78.18	536,524	3.76%
3.1	Tahoe	0.1	5,371,442	89.16	470,415	2.45%
3.2	Reno	0.1	5,386,152	90.05	465,783	2.73%
3.3	New Reno	0.1	5,432,178	87.35	480,162	3.61%
4.1	Tahoe	0.2	5,379,740	95.81	437,796	2.61%
4.2	Reno	0.2	5,397,880	135.55	309,421	2.96%
4.3	New Reno	0.2	5,437,248	104.56	401,155	3.71%
5.1	Tahoe	0.33333	5,378,856	154.67	271,176	2.59%
5.2	Reno	0.33333	5,403,632	186.83	224,498	3.07%

5.3	New Reno	0.33333	5,445,038	166.89	251,326	3.86%
6.1	Tahoe	0.5	5,366,798	101.01	415,247	2.36%
6.2	Reno	0.5	5,390,914	201.18	208,488	2.82%
6.3	New Reno	0.5	5,437,543	283.44	147,977	3.71%

Πίνακας 10. Αποτελέσματα του test με μέση διάρκεια φάσης On/Off 10 seconds.

3.7. ΑΠΩΛΕΙΑ ΠΑΚΕΤΟΥ ΚΑΙ ΚΑΤΑΣΤΑΣΗ ΣΥΜΦΟΡΗΣΗΣ ΣΤΟ TCP VENO

3.7.1 Γενική Περιγραφή

Μια ανατίμηση στη μεριά αποστολής του TCP, αποκαλούμενη ως Veno, εισήχθη πρόσφατα κυρίως για να εξαλείψει την ταλαιπωρία του TCP στα ασύρματα περιβάλλοντα. Τα πραγματικά πειράματα επικύρωσαν ότι το TCP Veno πράγματι λειτουργεί αποτελεσματικά και αρμονικά μέσα σε ενσύρματα/ασύρματα

δίκτυα. Σε αυτό το έγγραφο, ξαναεπισκεπτόμαστε το ξεχωριστό σχέδιο απώλειας πακέτου, το οποίο δανείζεται από το TCP Vegas και υιοθετείται από το TCP VenO. Τα αποτελέσματα προσομοίωσής μας καταδεικνύουν ότι σε δίκτυα με ελαφριά ή μέτρια διασταύρωση κυκλοφορίας, η απώλεια που διακρίνεται μεταξύ απώλειας συμφόρησης και τυχαίας απώλειας δουλεύει καλά, ενώ στα δίκτυα με τη βαριά διασταύρωση κυκλοφορίας, η παρερμηνεία της απώλειας συμφόρησης ως τυχαίας απώλειας είναι πολύ υψηλή. Εντούτοις, αυτός ο τύπος κρίσης ανώτερης λανθασμένης απώλειας δεν έχει δυσμενείς επιπτώσεις στη συμπεριφορά του VenO, η οποία είναι αρκετά διαφορετική από η διαίσθησή μας. Η συνολική εξέλιξη παραθύρου του VenO είναι ακόμα συγκρίσιμη με αυτή του Reno και συνεχίζει έτσι να εργάζεται αρμονικά με το Reno στα ενσύρματα δίκτυα. Τέτοιο παρατηρηθέν δίλημμα μας ωθεί να ξανασκεφτούμε δύο θεμελιώδεις πτυχές στη θεωρία ελέγχου συστήματος – το σήμα συμφόρησης ελέγχεται και που ανάλογες ενέργειες διενεργούνται. Η περαιτέρω μελέτη πάνω στο VenO εξερευνιέται ακόμα.

Το TCP έχει γίνει το κυρίαρχο πρωτόκολλο μεταφοράς για υπηρεσία παράδοσης αξιόπιστων δεδομένων στο Διαδίκτυο. Το TCP συμπεραίνει ότι όλες οι απώλειες το πακέτου προκαλούνται από τη συμφόρηση. Επάνω σε όλες τις απώλειες πακέτου, το TCP ρίχνει το παράθυρο συμφόρησης για να μειώσει το ποσοστό αποστολής. Καθώς η ασύρματη επικοινωνία έχει αυξηθεί γρήγορα, ο λόγος για τις απώλειες πακέτου μπορεί να είναι η λάθος μετάδοση εκτός από τη συμφόρηση. Κατά συνέπεια, η χρησιμοποίηση του TCP χωρίς καμία τροποποίηση οδηγεί σε σοβαρή υποβάθμιση της απόδοσης.

Μέχρι σήμερα, πολλή εργασία έχει εξερευνηθεί για να στραφεί με ακρίβεια στην αντίληψη των ποιών από τις απώλειες πακέτου οφείλονται στη συμφόρηση και ποιες οφείλονται στα bit-λάθη ή σε άλλους μη-συμφόρησης λόγους, αλλά, που περιορίζονται σε πάρα πολλούς αβέβαιους παράγοντες (δηλ., η κυκλοφορία υποβάθρου που αλλάζει κατά μήκος της διαδρομής της σύνδεσης) σε πραγματικά δίκτυα, η πρόοδος σε αυτό το είδος κρίσης μοιάζει πολύ αργή. Το TCP VenO [\[45\]](#) εισάγει τη διαφοροποίηση επιπέδου συμφόρησης, που στοχεύει να παρακάμψει τη διάκριση τύπου-απώλειας πακέτου.

Στο TCP Reno, υπάρχουν δύο σήματα για να εμπλέξουν την ύπαρξη απώλειας πακέτου: το διάλειμμα και τρία αναπαραχθέντα acks (*dupacks*), όταν εμφανίζεται το διάλειμμα, το Reno δέχεται μια αυστηρή ποινή παραθύρου – διχοτόμηση-παραθύρου και εκτελείται η αργή εκκίνηση. Όταν τρία *dupacks* παραλαμβάνονται από τον

αποστολέα, το Reno θα διχοτομήσει το παράθυρό του, και το επιθετικότερο σχέδιο της αποκατάστασης αναλαμβάνει από εκείνη την δράση που αντιστοιχεί στην ύπαρξη διαλείμματος. Γενικά, το διάλειμμα ερμηνεύεται ως αυστηρό σήμα κατάστασης συμφόρησης και τρία *dupacks* ερμηνεύονται ως σήμα κατάστασης συμφόρησης. Το TCP Veno εξαλείφει περαιτέρω την κατάσταση συμφόρησης του Reno σε δύο επίπεδα συμφόρησης – την *ελαφριά συμφόρηση*, την *μετριοπαθή συμφόρηση* όταν παραλαμβάνονται τρία *dupacks*. Στην ελαφριά συμφόρηση, το παράθυρο μειώνεται κατά $1/5$, παρά του $1/2$ σε Reno. Στη μέτρια συμφόρηση, η ποινή παραθύρου είναι ίδια με τη δράση του Reno. Ένεκα τέτοιας ύπαρξης διαλείμματος, το Veno κληρονομεί την ερμηνεία του Reno ως *αυστηρή συμφόρηση* και κατά συνέπεια εκτελεί την αυστηρό ποινή παραθύρου, όπως το Reno. Συνολικά, η ρύθμιση παραθύρου του Veno είναι βασισμένη στη εξαλειμμένη κατάσταση συμφόρησης χαρακτηρίζοντας σε - αυστηρή κατάσταση συμφόρησης, μέτρια κατάσταση συμφόρησης και ελαφριά κατάσταση συμφόρησης, διαφορετικά απλώς στην αφαίρεση απώλειας πακέτου από το διάλειμμα ή τα τρία *dupacks*.

Το TCP Veno είναι μια παραλλαγή του TCP Reno από πλευράς αποστολής, στοχεύοντας να βελτιώσει την απόδοση του TCP στο ενσύρματο/ασύρματο δίκτυο. Η κύρια ιδέα προκύπτει με τον ευφυή συνδυασμό του TCP Reno και του Vegas. Σαν περίπτωση, το Veno δανείζεται το σχέδιο¹ ανίχνευσης κατάστασης του Vegas για να συμπληρώσει την εκτίμηση της κατάστασης ενώ μία TCP σύνδεση εξελίσσεται στην φάση προσθετικής αύξησης.

¹ Υπάρχει μια ασήμαντη τροποποίηση στην *BaseRRT* εκτίμηση του Veno, διαφορετική από την αυθεντική του Vegas.

Από αυτές τις συμπληρωματικές πληροφορίες, το Venο μπορεί κατά συνέπεια να μετατρέψει την φάση προσθετικής αύξησης σε *ελαφριά* συμφόρηση και *μέτρια* συμφόρηση. Όπως δείχνεται στο [45], εκτός από το σχέδιο ανίχνευσης συμφόρησης του Vegas, το Venο «θα μπορούσε επίσης να χρησιμοποιήσει άλλες προφητικές ανιχνεύσεις συμφόρησης (π.χ., PBM [58], Ζευγάρι Πακέτου [57]) ή/και άλλα σχέδια καλύτερης προφητικής ανίχνευσης συμφόρησης» για να συμπληρώσει την εκτίμηση της κατάστασης ενώ μία σύνδεση TCP εμπλέκεται στη φάση προσθετικής αύξησης.

Σε αυτό το έγγραφο, μελετάμε τις συμπεριφορές του Venο χρησιμοποιώντας το ns-2 εργαλείο, και λαμβάνονται δύο παρατηρήσεις, 1) το σχέδιο διάκρισης απώλειας που δανείζεται από το TCP Vegas από το Venο μπορεί να λειτουργήσει καλά μέσα σε δίκτυα με ελαφριά ή μέτρια διασταύρωση κυκλοφορίας, ενώ μέσα σε δίκτυα με τη βαριά διασταύρωση κυκλοφορίας, η αναγνώριση απώλειας συμφόρησής του είναι πολύ χαμηλή αν και η τυχαία αναγνώριση απώλειας είναι ακόμα ακριβής. Το σχέδιο ανίχνευσης συμφόρησης του Vegas δεν μπορεί να εφαρμοστεί καλά στις καταστάσεις βαριάς συμφόρησης. 2) Δεν υπάρχει καμία προφανή επιθετική συμπεριφορά για το TCP Venο σε όλες τις διαφορετικές καταστάσεις δικτύου. Στην πραγματικότητα, το Venο συνεχίζει να εργάζεται αρμονικά και αποτελεσματικά με το Reno πάνω σε ενσύρματα δίκτυα ανεξάρτητα από την ορθότητα της διάκριση τύπου απώλειας. Με άλλα λόγια, η παρερμηνευμένη απώλεια συμφόρησης ως τυχαία απώλεια δεν φέρνει οποιοδήποτε αρνητικό αντίκτυπο στη γενική συμπεριφορά του TCP Venο. Αυτά τα παρατηρημένα διλήμματα μας ωθούν να ξανασκεφτούμε δύο θεμελιώδεις πτυχές στη θεωρία ελέγχου συστήματος, 1) πώς να καταλάβει τελείως τα ελεγχόμενα σήματα συμφόρησης, 2) πώς να εκτελέσει τις ανάλογες ενέργειες σαν απάντηση στα ελεγχόμενα σήματα. Στο Venο TCP, και η απώλεια πακέτου και η κατάσταση σύνδεσης ελέγχονται ως σήματα συμφόρησης, παρά σαν απώλεια πακέτου. Καθένα από αυτά θα μπορούσε να είναι ανακριβή σε κάποιες συγκεκριμένες καταστάσεις, παραδείγματος χάριν, μέσα σε ασύρματα δίκτυα, η απώλεια μπορεί να μην οφείλεται λόγω της συμφόρησης. Στα δίκτυα με διασταύρωση κυκλοφορίας, η εκτίμηση κατάστασης σύνδεσης μπορεί να μην είναι τόσο ακριβής. Κατά συνέπεια, η σωστή κρίση πρέπει να βασίζεται στη σύνθεση αυτών των δύο ελεγχόμενων σημάτων, παρά στο ένα μόνο σήμα. Ειδικά, πρέπει να επισημανθεί ότι αν και αυτή η κρίση σύνθεσης μπορεί να είναι λάθος όπως σαν παρερμηνευμένη

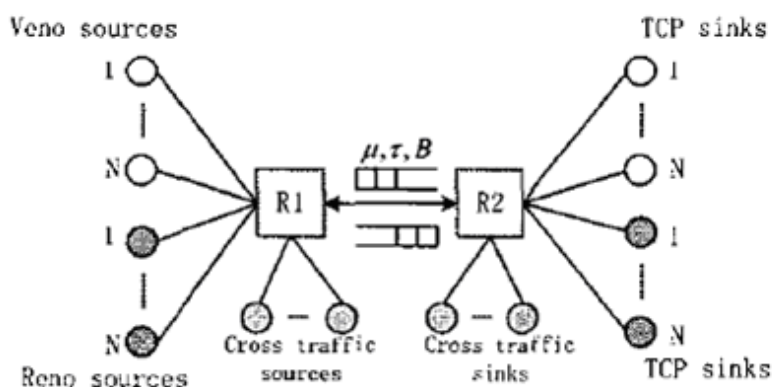
απώλεια συμφόρησης σαν τυχαία απώλεια, το παράθυρό του ακόμα μειώνεται κατά 1/5 (εκτός αν παραμένει αμετάβλητο) επειδή το Veno ακόμα συνάγει ότι η σύνδεση εξελίσσεται μέσα σε μια κατάσταση συμφόρησης - ελαφριά συμφόρηση. αυτή η σύνθεση σημάτων και το σχέδιο ευφυής μείωσης παραθύρου είναι που οδηγούν στα χαρακτηριστικά καλής-συμπεριφοράς του Veno.

Τέλος, δίνουμε έμφαση «ποιο TCP Veno προτείνει πρόκειται να εξαλείψει την AIMD (προσθετική αύξηση και πολλαπλασιαστική μείωση: **additive increase and multiplicative decrease**) εξέλιξη του Reno πάνω σε ετερογενή δίκτυα με τη χρησιμοποίηση την πλήρης κρίση της εκτίμησης κατάστασης δικτύου - κατάστασης συμφόρησης ή κατάστασης λιγότερης συμφόρησης, παρά μόνο ανάλογα με το γεγονός απώλειας πακέτου». Η γενική απόδοση του Veno λειτουργεί αρκετά καλά σε ενσύρματα και ασύρματα, ιδιαίτερα σε ασύρματα περιβάλλοντα.

3.7.2. ΠΕΙΡΑΜΑΤΑ

3.7.2.1 Τοπολογία Προσομοίωσης

Χρησιμοποιούμε το ns-2 [56] για να διαχειρισθεί την προσομοίωσή μας. Το [σχήμα 32](#) παρουσιάζει το πρότυπο που χρησιμοποιείται για την προσομοίωση.



Σχήμα 32. Δικτυακό Μοντέλο

Στην προσομοίωση, έχουμε τις συνδέσεις TCP Veno και Reno να μοιράζονται τη δυσχερή διασύνδεση με διασταύρωση κυκλοφορίας μεταξύ R1(δρομολογητής1)

και R2. Η δυσχερή διασύνδεση χρησιμοποιεί την πειθαρχία FIFO αναμονής. Οι διασυνδέσεις μεταξύ R2 και όλοι οι οχετοί TCP συμπεριφέρονται ως ασύρματες διασυνδέσεις. Η συσσωμάτωση των πηγών Pareto διασταύρωσης κυκλοφορίας με σχηματισμένες παραμέτρους στο 1.7 (<2) παράγουν τη μακροχρόνια εξαρτώμενη σειρά (Long Range Dependent) (LRD) κυκλοφορία [53], το μ δείχνει το μέγεθος buffer της δυσχερής διασύνδεσης μεταξύ του R1 και R2, το τ δείχνει την μονής διαδρομής καθυστέρηση διάδοσης (δεν περιλαμβάνει το χρόνο μετάδοσης και την καθυστέρηση αναμονής) της δυσχερής διασύνδεσης. Το B αντιπροσωπεύει την χωρητικότητα δυσχέρειας. Κάθε παράμετρος εφαρμόζεται και για τις προς τα εμπρός και για τις προς τα πίσω κατευθύνσεις.

Έστω ότι το CI δείχνει το φορτίο της με διασταύρωση κυκλοφορίας (δηλ., η χρησιμοποίηση της κατά προσέγγιση χωρητικότητας της δυσχέρειας που καταλαμβάνεται από τη με διασταύρωση κυκλοφορία). Ορίζουμε το γ να είναι το ποσοστό τυχαίας απώλειας πάνω σε ασύρματες διασυνδέσεις. Εφαρμόζουμε διαφορετικά CI και γ για να εκτελέσουμε την αξιολόγηση κάτω από διαφορετικές συνθήκες δικτύου. Κάθε πείραμα διαρκεί για 500 δευτερόλεπτα.

3.7.2.2 Η Απόδοση της Διάκρισης Απώλειας Πακέτου

Στο Venet, η κρίση, του πότε μια απώλεια πακέτου οφείλεται σε συμφόρηση δικτύου ή σε λάθος της ασύρματης μετάδοσης συνάγεται από την εκτίμηση της κατάστασης δικτύου. Ενώ μερικές εξαιρέσεις που δεν μπορούμε να αντιληφθούμε μπορεί να υποστούν και να οδηγήσουν σε λάθος διάγνωση: η πραγματική κατάσταση δικτύου θα μπορούσε να σχετίζεται με μια άσχετη απώλεια πακέτου ή μια απώλεια πακέτου είναι συνδεδεμένη με κατάσταση δικτύου που εσφαλμένα εκτιμήθηκε.

Καθορίζουμε δύο μετρικές για να αξιολογήσουμε την ακρίβεια της διάκρισης απώλειας πακέτου:

- $PN \geq \beta | C$: Αυτός ο μετρικός είναι η υπό όρους πιθανότητα προσδιορισμού μιας απώλειας πακέτου ως απώλεια συμφόρησης δεδομένου ότι πράγματι οφείλεται στη συμφόρηση. Δείχνει την ακρίβεια της διάκρισης της απώλειας συμφόρησης.

- $PN < \beta | C$: Αυτός ο μετρικός είναι η υπό όρους πιθανότητα προσδιορισμού μιας απώλειας πακέτου ως τυχαία απώλεια δεδομένου ότι πράγματι οφείλεται σε τυχαία απώλεια. Δείχνει την ακρίβεια της διάκρισης της τυχαίας απώλειας.

Σε αυτό το σενάριο, το μέγεθος πακέτου μιας Veno και μιας Reno σύνδεσης είναι 1460bytes. Το μ , το B , και το τ είναι 60ms, 4Mbps και 28, αντίστοιχα. Το μέγεθος buffer όλων των άλλων διασυνδέσεων είναι 50 κατά προεπιλογή. Οι διασυνδέσεις μεταξύ των πηγών TCP και του R1 ορίζονται στα 10Mbps και 0.1ms μονής διαδρομής καθυστέρηση διάδοσης και στις δύο κατευθύνσεις. Οι ασύρματες διασυνδέσεις μεταξύ του R2 και των οχητών TCP έχουν 2Mbps χωρητικότητα και 0.01ms μονής διαδρομής καθυστέρηση διάδοσης και στις δύο κατευθύνσεις.

Ο [πίνακας 11](#), ο [πίνακας 12](#) και ο [πίνακας 13](#) συνοψίζουν τα αποτελέσματα του CI για 30%, 50% και 70%¹, αντίστοιχα.

¹ Τα αποτελέσματα του CI για 0 και 10% έχουν παρόμοιες ροπές σαν τις παραπάνω παρατηρήσεις. Για να εξοικονομήσουμε χώρο στην εργασία, δεν παρουσιάσαμε αυτά τα αποτελέσματα,

Αποκτάμε τρεις παρατηρήσεις 1) το $PN \geq \beta | C$ μειώνεται και το $PN < \beta | C$ αυξάνεται, όταν το CI αυξάνεται. 2) το $PN \geq \beta | C$ αυξάνεται και το $PN < \beta | C$ εμφανίζει διακύμανση με την αύξηση του γ , ενώ το CI είναι σταθερό. 3) Όταν το δίκτυο είναι βαριά κορεσμένο ($CI=70\%$), η ικανότητα διάκρισης της απώλειας της συμφόρησης του Vegas ($PN \geq \beta | C$) είναι σημαντικά αποδυναμωμένη.

Ας ερευνήσουμε την πρώτη παρατήρηση. Αναφερόμενοι στον ορισμό στο [\[45\]](#), η σειρά προτεραιότητας σε αναμονή μπορεί να γραφτεί ως όρος $cwnd * (1 - BaseRTT/RTT)$. Όταν μια σύνδεση TCP Veno εγκαθίσταται, εάν το δίκτυο έχει φορτωθεί βαριά από τη με διασταύρωση κυκλοφορία, το $BaseRTT$ μπορεί να υπερεκτιμηθεί και έτσι ο όρος $(1 - BaseRTT/RTT)$ μειώνεται. Συνεπώς, η σειρά προτεραιότητας σε αναμονή είναι υποτιμημένη, το οποίο οδηγεί στην υποτίμηση της κατάστασης δικτύου. Επιπλέον, αφού η σύνδεση Veno καταλαμβάνει μόνο ένα μικρό μέρος της χωρητικότητας δικτύου - περίπου το 15% της χωρητικότητας δυσχέρειας, ο αντίκτυπος αυτής της σύνδεσης στο επίπεδο συμφόρησης είναι ασήμαντος. Αυτό υποδηλώνει ότι το $BaseRTT$ μπορεί ακόμα να είναι υπερεκτιμημένο ακόμα κι αν το Veno μειώνει το παράθυρο συμφόρησης ώστε να μειώσει το ποσοστό, επειδή το

δίκτυο δεν μπορεί ακόμα να βγει από την κατάσταση συμφόρησης. Ο λόγος για τις αυξανόμενες τάσεις του $PN < \beta | C$ μπορεί να εξηγηθεί όπως ακολουθεί. Εάν μία τυχαία απώλεια εμφανισθεί όταν το δίκτυο βρίσκεται σε κατάσταση συμφόρησης (περιμένουν στη σειρά του buffer δυσχέρειας το λιγότερο β πακέτα και το buffer δεν είναι υπερχειλισμένο), η τυχαία απώλεια θα θεωρηθεί σαν απώλεια συμφόρησης. Θυμηθείτε ότι από τη μία πλευρά, καθώς το CI αυξάνει, η υποτίμηση της κατάστασης συμφόρησης δικτύου χειροτερεύει την ακρίβεια της εκτίμησης της απώλειας συμφόρησης.

Στην παράγραφο παρατήρησης, το $PN < \beta | C$ αυξάνει και το $PN \geq \beta | C$ επιδεικνύει διακύμανση με την αύξηση του γ , καθώς το CI είναι σταθερό. Στο συχνότερο περιστατικό της τυχαίας απώλειας που το Venet πρέπει να αντιδράσει, είναι λιγότερο πιθανό για το Venet να αυξήσει την αποστολή του μειώνει τις πιθανότητες εξακρίβωσης την τυχαία απώλεια ως απώλεια συμφόρησης. Μπορούμε να δούμε ότι κάτω από σχετικά μικρό CI (30%, 50%), αυτή η τάση είναι σημαντικότερη από την τάση κάτω από εξαιρετικά μεγάλο CI (70%), επειδή μια ροή TCP που χρησιμοποιεί ένα πολύ μικρό μέρος της χωρητικότητας δικτύου έχει ελάχιστη επιρροή πάνω στην κατάσταση δικτύου, έτσι όπως έχουμε συζητήσει ανωτέρω. Το βαρύ φορτίο της με διασταύρωση κυκλοφορίας και της τυχαίας απώλειας οδηγούν σε μικρό $PN \geq \beta | C$ και μεγάλος $PN < \beta | C$.

TABLE 1
Experiment results for loss distinguishing in Venet, $C_l = 30\%$

γ	Th(Mbps)		Actual Type of Packet Loss						$P_{N < \beta W}$	$P_{N \geq \beta C}$
	Venet	Reno	Random Loss			Congestion Loss				
			NUM_W	$NUM_{W, N \geq \beta}$	$NUM_{W, N < \beta}$	NUM_C	$NUM_{C, N \geq \beta}$	$NUM_{C, N < \beta}$		
0	1.288	1.242	0	0	0	231	208	13	N/A	0.941
0.002	1.373	1.169	100	33	67	151	136	15	0.67	0.901
0.004	1.369	1.111	188	61	127	105	99	6	0.676	0.943
0.006	1.380	1.06	311	65	246	70	64	6	0.79	0.914
0.008	1.31	1.003	350	57	307	46	38	8	0.86	0.826
0.01	1.328	0.888	525	49	476	17	14	3	0.907	0.824

TABLE 2
Experiment results for loss distinguishing in Venet, $C_l = 50\%$

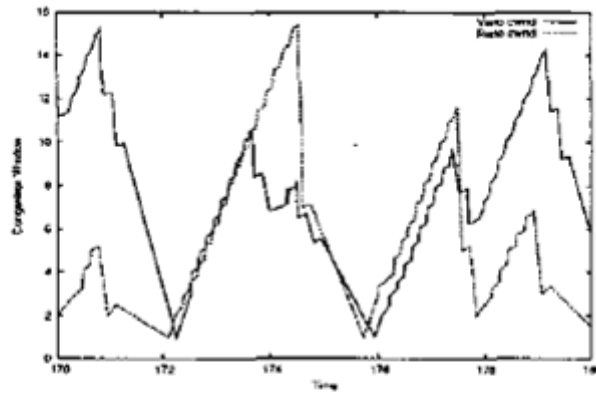
γ	Th(Mbps)		Actual Type of Packet Loss						$P_{N < \beta W}$	$P_{N \geq \beta C}$
	Venet	Reno	Random Loss			Congestion Loss				
			NUM_W	$NUM_{W, N \geq \beta}$	$NUM_{W, N < \beta}$	NUM_C	$NUM_{C, N \geq \beta}$	$NUM_{C, N < \beta}$		
0	1.041	0.909	0	0	0	278	173	105	N/A	0.622
0.002	0.967	0.820	62	17	45	248	157	91	0.726	0.633
0.004	1.007	0.822	115	22	93	241	122	119	0.809	0.506
0.006	1.027	0.781	217	25	192	188	73	115	0.885	0.388
0.008	1.032	0.777	326	28	298	158	81	77	0.914	0.513
0.01	1.041	0.69	365	31	334	141	56	85	0.915	0.392

TABLE 3
Experiment results for loss distinguishing in Venet, $C_l = 70\%$

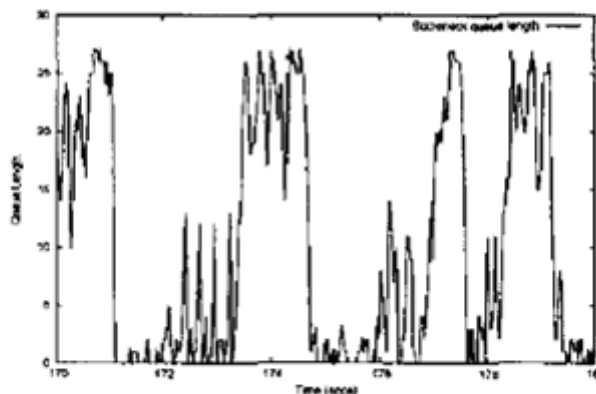
γ	Th(Mbps)		Actual Type of Packet Loss						$P_{N < \beta W}$	$P_{N \geq \beta C}$
	Venet	Reno	Random Loss			Congestion Loss				
			NUM_W	$NUM_{W, N \geq \beta}$	$NUM_{W, N < \beta}$	NUM_C	$NUM_{C, N \geq \beta}$	$NUM_{C, N < \beta}$		
0	0.714	0.570	0	0	0	387	58	329	N/A	0.15
0.002	0.746	0.556	44	3	41	386	45	341	0.931	0.117
0.004	0.708	0.560	83	4	79	386	22	344	0.952	0.109
0.006	0.748	0.527	131	6	125	343	22	321	0.954	0.064
0.008	0.692	0.531	179	8	171	334	23	311	0.955	0.069
0.01	0.681	0.488	203	8	195	257	32	255	0.961	0.111

Πίνακας 11, 12, 13 συνοπτικά τα αποτελέσματα του CI για 30%, 50% και 70%

Η τρίτη παρατήρηση είναι ότι το $PN \geq \beta | C$ μειώνεται με την αύξηση του CI . Σε ακραία περίπτωση που το CI είναι 70% και το γ είναι 0, το $PN \geq \beta | C$ είναι 0.15. Αυτό υπονοεί ότι μεταξύ κάθε 7 απωλειών συμφόρησης, μόνο μια μπορεί να ταξινομηθεί με ακρίβεια ενώ οι άλλες 6 έχουν διαγνωσθεί λάθος ως τυχαίες απώλειες και αντίστοιχα, τείνουν να οδηγήσουν το δίκτυο σε συμφορητική κατάσταση και έτσι, το παράθυρο συμφόρησης μειώνεται κατά 1/5. Σε τέτοια περίπτωση, κάποιος μπορεί να σκεφτεί ότι μια Venο σύνδεση πρέπει να είναι πολύ επιθετική λόγω της χαμηλής ακρίβεια στη διάκριση της απώλειας συμφόρησης και θα αποσπάσει πολύ πόρο εύρους ζώνης από τη Reno σύνδεση που συνυπάρχει. Εντούτοις, βρήκαμε ότι σε τέτοια ακραία περίπτωση, το Venο μπορεί ακόμα να κρατήσει το χαρακτηριστικό της συμβατότητάς του – το Venο συνεχίζει να δουλεύει φιλικά με το Reno χωρίς «να κλέβει» εύρος ζώνης από το Reno. Έκτοτε το παράθυρο συμφόρησης του Venο έχει περιοριστεί βαριά από τις συχνές απώλειες συμφόρησης στις περίπτωση του ακραίου βαρύ φορτίου. Έτσι η διαφορά μεταξύ του παραθύρου συμφόρησης που μειώνεται κατά 1/2 και 1/5 είναι ασήμαντη. Παραδείγματος χάριν, στο [σχήμα 33](#), όταν το CI είναι 70%, ο μέσος όρος του παραθύρου συμφόρησης της Venο σύνδεσης είναι μόνο περίπου 5.1 ενώ ο μέσος όρος του παραθύρου συμφόρησης της Reno σύνδεσης είναι 4.9. Το [σχήμα 33](#) επεξηγεί τη γενική εικόνα της εξέλιξης του παραθύρου συμφόρησης για μία Reno και μια Venο σύνδεση και τη διακύμανση του μήκους της σειρά αναμονής για το CI του 70% χωρίς τεχνητά εισαγμένη τυχαία απώλεια.



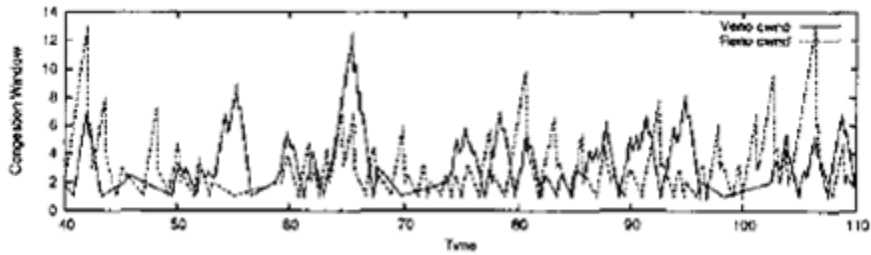
(a) Veno/Reno congestion window $C_i = 70\%$



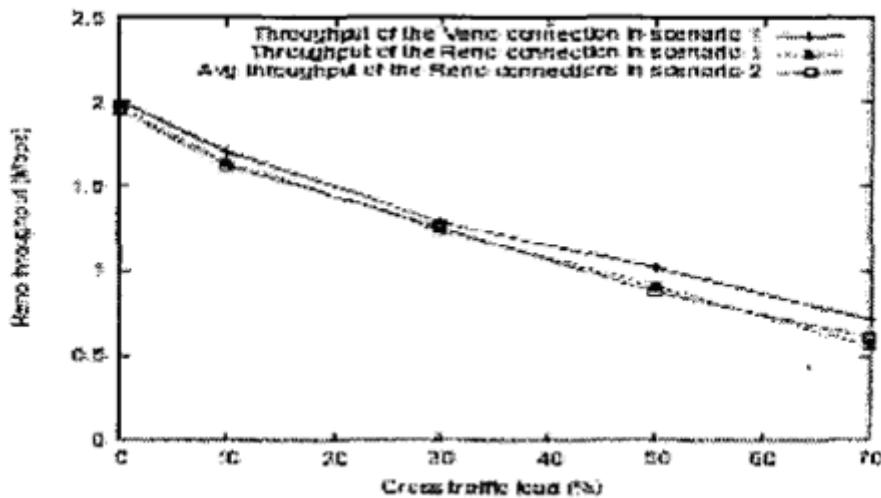
(b) Bottleneck queue length $C_i = 70\%$

Σχήμα 33. Παράθυρο συμφόρησης Veno/Reno έναντι της επιβράδυνσης του μήκους της σειράς αναμονής (χωρίς τυχαία απώλεια)

Από την άλλη, όπως απεικονίζεται στο [σχήμα 33](#), το Reno έχει ευκαιρία να αυξήσει το μέγεθος παραθύρου του (π.χ., γύρω από τη χρονική στιγμή των 174s) πέραν του Veno ([σχήμα 33](#) (a)), αν και κάποια στιγμή (π.χ. γύρω από τη χρονική στιγμή των 178s) το Veno μπορεί να διατηρήσει υψηλότερο μέγεθος παραθύρου λόγω της λάθος διάγνωσης πακέτου όταν το δίκτυο είναι βαριά κορεσμένο ([σχήμα 33](#) (a) και [σχήμα 33](#) (b)). Για παράδειγμα, βρήκαμε ότι γύρω από τη χρονική στιγμή των 174s όταν συμβαίνει μια απώλεια συμφόρησης αλλά εντοπίζεται ως τυχαία απώλεια, το Veno προσπάθησε να πετύχει αποδοτικότερη χρησιμοποίηση με το μικρότερο παράγοντα πτώσης παραθύρου του 1/5. Αυτό με τη σειρά του επέστρεψε τον ερχομό της επόμενης απώλεια συμφόρησης, η οποία θα διαγνωστεί λάθος και πάλι. Επιπλέον, στο [σχήμα 3](#), μπορούμε να δούμε ότι το Διάλειμμα εμφανίζεται με υψηλή συχνότητα, που κάνει είτε τη Veno είτε τη Reno σύνδεση να ρίχνουν το παράθυρο συμφόρησής τους σε 1 και εν τω μεταξύ δίνουν άλλες πιθανότητες να προφθάσουν τη ρυθμοαπόδοσή τους. Όλα αυτά κάνουν το μέσο παράθυρο συμφόρησης του Veno συγκρίσιμο με το μέσο παράθυρο συμφόρησης του Reno.



Σχήμα 33. Εξέλιξη παραθύρου συμφόρησης Venet και Reno (χωρίς τυχαία απώλεια).



Σχήμα 34. Ρυθμοαπόδοση των Venet και Reno έναντι του μέσου όρου ρυθμοαπόδοσης των δυο συνδέσεων Reno.

Για να επικυρώσουμε τη ανάλυσή μας, κάνουμε επίσης το πείραμα (σενάριο 2) των δύο συνδέσεων Reno παρά από μία Venet και μία Reno σύνδεση, ενώ κρατάμε αμετάβλητες τις άλλες οργανώσεις που είναι προαναφερμένες στο Πείραμα (σενάριο 1). Η μέση ρυθμοαπόδοση των δύο συνδέσεων Reno κάτω από διαφορετικά CI (0, 10%, 30%, 50% 70%) συγκρίνονται με τη ρυθμοαπόδοση της Reno και Venet σύνδεσης στο σενάριο 1, το γ είναι 0. Όπως δείχνει το [σχήμα 34](#), η Venet σύνδεση είναι συμβατή ακόμη και όταν η διαδρομή είναι βαριά φορτωμένη - η απόδοση του Reno δεν συγκρούεται σοβαρά με του Venet, όπως συγκρίνεται με μέση ρυθμοαπόδοση των δύο συνδέσεων Reno.

3.8.Αύξηση του TCP Westwood+ σε Μεγάλης Ταχύτητας Ευρέα δίκτυα

3.8.1 Γενική Περιγραφή

Σε αυτό το έγγραφο, μελετούνται οι μηχανισμοί για να ενισχύσουν την απόδοση του TCP Westwood+ παρουσία μιας μεγάλης παραγόμενης καθυστέρησης-εύρους ζώνης (Bandwidth - Delay Product). Συγκεκριμένα, προτείνεται η απασχόληση μιας λογαριθμικής λειτουργίας για την αύξηση του παραθύρου συμφόρησης στην απουσία των απωλειών πακέτου. Εκτενής αριθμητικές προσομοιώσεις, που πραγματοποιήθηκαν με τη χρησιμοποίηση του ns-2, δείχνουν ότι η χρήση τέτοιας λογαριθμικής λειτουργίας μπορεί να οδηγήσει σε σχετικές βελτιώσεις απόδοσης με σεβασμό στο τυποποιημένο πρωτόκολλο Westwood+.

Το πρότυπο αναφοράς TCP/IP καθορίζει ένα σύνολο πρωτοκόλλων για να καταστήσουν ικανές τις end-to-end αξιόπιστες επικοινωνίες μέσω του Διαδικτύου. Σύμφωνα με αυτό το πρότυπο, κανένα υπόστρωμα δεν έχει ολοκληρώσει και οι πληροφορίες πραγματικού-χρόνου για τους διαθέσιμους πόρους δικτύου πάνω σε διαδρομή με πολλά *hop* όπου οι επικοινωνίες πραγματοποιούνται. Η ανικανότητα να εντοπίσει μια δυσχέρεια παρούσα στην διαδρομή μετάδοσης στοιχείων παρακινεί την ανάγκη του να ελέγχεται το ποσό δεδομένων κυκλοφορίας που παραδίνεται στο δίκτυο προκειμένου να αποφευχθεί η συμφόρηση δικτύου και να αποτραπεί η κατάρρευση δικτύου.

Το Πρωτόκολλο Ελέγχου Μετάδοσης (TCP) [59] είναι το *de facto* πρότυπο για την εξασφάλιση της *end-to-end* αξιόπιστης παράδοσης των πακέτων στο Διαδίκτυο και χρησιμοποιείται από μια μεγάλη ποικιλία εφαρμογών [60], [61]. Ο αλγόριθμος ελέγχου συμφόρησης του TCP που πρωτοπαρουσιάστηκε από τον Van Jacobson [62] είναι βασισμένος σε ένα μηχανισμό κινητού παραθύρου και υιοθετεί ένα αλγόριθμο Προσθετικής Αύξησης Πολλαπλασιαστικής Μείωσης (AIMD) για να ενημερώσει το ποσοστό μετάδοσης των διαθέσιμων πόρων δικτύου. Ο δέκτης παρέχει μία αναγνώριση (ACK) ανάδρασης, βασισμένος στην οποία ο αποστολέας ανιχνεύει τις απώλειες πακέτων και συνεπώς ενημερώνει το ποσοστό μετάδοσης. Δεδομένου ότι το TCP σχεδιάστηκε αρχικά για ενσύρματα με χαμηλό ποσοστό λάθους δίκτυα, αυτό υποθέτει ότι όλες οι απώλειες πακέτου είναι προκαλούμενες από τη συμφόρηση δικτύου.

Το παράθυρο συμφόρησης είναι η βασική παράμετρος στη συμπεριφορά του TCP, και καθορίζει το ποσό των μη αναγνωρισμένων δεδομένων (σε αντιστοιχία με TCP segments) το οποίο μπορεί να παραδοθεί στο δίκτυο. Ο μηχανισμός AIMD καθορίζει μια γραμμική αύξηση του παραθύρου συμφόρησης με την πάροδο του χρόνου. Επάνω στην ανίχνευση απώλειας μέσω της υποδοχής των διπλών ACKs*

* Υποθέτουμε εδώ ότι η ανίχνευση απώλειας πακέτου πραγματοποιείται αφότου λάβει ο αποστολέας τρεις διπλές αναγνωρίσεις από το δέκτη, παραμελώντας τον αντίκτυπο των περιστατικών διαλείμματος.

Ο αποστολέας διχοτομεί την τιμή του παράθυρο συμφόρησης. Το γενικό αποτέλεσμα της στρατηγική της συντηρητικής αύξησης και της επιθετική μείωσης είναι ότι οι μηχανισμοί που βασισμένοι σε AIMD υποχρησιμοποιεί το διαθέσιμο εύρος ζώνης σε μεγάλης ταχύτητας δίκτυα παρουσίας μεγάλων καθυστερήσεων [63], [64]. Ο κύριος λόγος για την κακή ποιότητα απόδοσης του AIMD σε τέτοια σενάρια είναι ριζωμένος μέσα στο γεγονός ότι ο μηχανισμός προσθετικής αύξησης του τυποποιημένου TCP είναι ιδιαίτερα ανεπαρκής παρουσίας της παραγόμενης καθυστέρησης-εύρους ζώνης (BDP). Με άλλα λόγια, είναι απαιτημένος ένας μεγάλο ποσό χρόνου για «να αποκαταστήσει» το παράθυρο συμφόρησης μετά από μια ανίχνευση πτώσης πακέτου. Όσο μεγαλύτερο είναι το BDP, τόσο περισσότερος χρόνος χρειάστηκε για την αποκατάσταση του παραθύρου. Μια λεπτομερής μελέτη αυτού του προβλήματος είναι παρουσιασμένη στο [64], όπου ο συντάκτης προτείνει ένα νέο πρωτόκολλο, αποκαλούμενο Scalable (εξελικτικό) TCP, βασισμένο στην υιοθέτηση ενός αλγόριθμος πολλαπλασιαστικής αύξησης για να υπερνικηθεί ένας τέτοιος περιορισμός. Η εκθετική αύξηση παραθύρου του Scalable TCP οδηγεί σε αξιοσημείωτη βελτίωση της απόδοσης στις μεγάλες BDP διασυνδέσεις, με εύρος ζώνης μεγαλύτερο από 100 Mb/s και το χρόνο στρογγυλού ταξιδιού καθυστέρηση διάδοσης πάνω από 50ms. Εντούτοις, σε χαμηλά BDP δίκτυα, ο αλγόριθμος αύξησης παραθύρου που προτείνεται στο Scalable TCP είναι λιγότερο επιθετικός από το μηχανισμό AIMD, που οδηγεί σε κακής ποιότητας απόδοση εάν συγκριθεί με το TCP NewReno.

Άλλες προτεινόμενες τροποποιήσεις στους μηχανισμούς ελέγχου συμφόρησης του τυποποιημένου TCP οι περιλαμβάνουν το High Speed TCP (HSTCP) προτεινόμενο στο [63], FAST TCP [65], και Binary Increase Congestion Control (Ελεγχος Δυαδικής Αύξησης Συμφόρησης) (BIC) [66]. Συγκεκριμένα, το τελευταίο

υιοθετεί μια λογαριθμική λειτουργία για την αύξηση παραθύρου στην αποφυγή συμφόρησης.

Στο [67], οι συντάκτες προτείνουν ένα εναλλακτικό πρωτόκολλο μεταφορών, αποκαλούμενο TCP Westwood, το οποίο στοχεύει στην ενίσχυση της απόδοσης του TCP στα με απώλειες περιβάλλοντα. Η κύρια ιδέα, όσο αναφορά το Westwood, είναι να κρατήσει μια εκτίμηση της διαθέσιμης end-to-end χωρητικότητας (που λαμβάνεται από ένα κατάλληλο φιλτράρισμα της επιστρεφόμενης ροής ACK) και να εκμεταλλευτεί τέτοιες πληροφορίες προκειμένου να μειώσει το ποσοστό μετάδοσης, αντί της τυφλής διχοτόμησης, προσέγγιση που εφαρμόζεται στο τυποποιημένο TCP. Οι διαδικασίες του TCP Westwood είναι ως εκ τούτου βασισμένες σε ένα μηχανισμό Προσθετικής Αύξησης Προσαρμοστικής Μείωσης (AIAD) εξέλιξης παραθύρου. Τα προβλήματα που αντιμετωπίστηκαν από το TCP Westwood παρουσία συμπίεσμένων/καθυστερημένων ACKs παρακίνησε μια τροποποίηση στο ACK φίλτρο που χρησιμοποιείται στην εκτίμηση εύρους ζώνης, που οδηγώντας στην εισαγωγή του πρωτοκόλλου Westwood+ [68].

Ενώ το Westwood+ είναι σε θέση να ξεπεράσει το τυποποιημένο TCP σε ένα ευρύ φάσμα σεναρίων εφαρμογής, ακόμα υποφέρει στην παρουσία ενός μεγάλου BDP, λόγω της χαμηλής ανταπόκρισης μετά από μία απώλεια πακέτου. Ακόμα κι αν μερικές πρόσφατες εργασίες έχουν προσπαθήσει να ενισχύσουν την απόδοση Westwood σε τέτοια σεσάρια [69], [70], η κύρια λειτουργία του αλγορίθμου αύξησης παραθύρου αφήνεται για να είναι προσθετική με τη συντηρητική συμπεριφορά αύξησής του.

Σε αυτό το έγγραφο, ερευνάμε τις πιθανές αυξήσεις του TCP Westwood+ στη παρουσία των υψηλής ταχύτητας μεγάλης-καθυστερήσης δικτύων. Αφήνοντας την εκτίμηση εύρους ζώνης και τη προσαρμοστική μείωση παραθύρου άθικτα, μελετάμε τον αντίκτυπο διαφορετικών αλγορίθμων εξέλιξης παραθύρου, και δείχνουμε πόσες αξιοσημείωτες βελτιώσεις πάνω στο Westwood+ μπορούν να επιτευχθούν στα δίκτυα που χαρακτηρίζονται από ένα μεγάλο BDP με τη βοήθεια μιας λογαριθμικής λειτουργίας αύξησης.

3.8.2 ΣΤΡΑΤΗΓΙΚΕΣ ΕΞΕΛΙΞΗΣ ΠΑΡΑΘΥΡΟΥ ΣΥΜΦΟΡΗΣΗΣ

Η σύνδεση TCP αρχίζει με το παράθυρο συμφόρησης (W) ίσο με ένα πακέτο. Κατόπιν, το W αυξάνεται κατά ένα για κάθε λαμβανόμενο μη-αντίγραφο ACK μέχρι το Κατώτατο όριο Αργής Εκκίνησης να είναι φτασμένο. Η βασική ιδέα πίσω από την φάση αργής εκκίνησης είναι να παρέχει μια γρήγορη (απότομα εκθετική με την πάροδο του χρόνου) αύξηση παραθύρου μέχρι η χωρητικότητα του αγωγού μετάδοσης να επιτευχθεί.

Μόλις ξεπεραστεί το κατώτατο όριο της αργής εκκίνησης, το TCP περνά στη φάση της αποφυγής συμφόρησης. Αυτή η φάση, η οποία στοχεύει στην ήπια διερεύνηση του δικτύου για το διαθέσιμο εύρος ζώνης, είναι χαρακτηρισμένη από μια απότομη γραμμική αύξηση του W με την πάροδο του χρόνου. Στην περίπτωση που τρία διαδοχικά διπλά ACKs παραλαμβάνονται, η τιμή του παραθύρου συμφόρησης W διχοτομείται. Συνοψίζοντας, οι ενέργειες που γίνονται κατά την υποδοχή ενός ACK και κατά την ανίχνευση απώλειας πακέτου μπορεί να αναπαρασταθεί ως:

$$W \leftarrow W + \frac{1}{W}; \quad (1)$$

$$W \leftarrow \frac{W}{2}. \quad (2)$$

Συνολικά, αυτό καθορίζει ένα μηχανισμό Προσθετικής Αύξησης Πολλαπλασιαστικής Μείωσης (AIMD).

Το Scalable TCP [64] προσπαθεί να υπερνικήσει τις ανεπάρκειες του μηχανισμού AIMD του TCP σε μεγάλα BDP δίκτυα με την υιοθέτηση ενός αλγόριθμου Πολλαπλασιαστικής αύξησης πολλαπλασιαστικής μείωσης (MIMD), οι λειτουργίες του οποίου μπορούν να συνοψιστούν ως:

$$W \leftarrow W + \alpha; \quad (3)$$

$$W \leftarrow \beta * W. \quad (4)$$

Η ταχύτητα αύξησης παραθύρου ελέγχεται από την παράμετρο α , ενώ το β είναι αρμόδιο για τη δυναμική μείωσης παραθύρου. Στο [64], ο συντάκτης προτείνει τη ρύθμιση του $\alpha=0.01$ και $\beta=0.875$ ως προεπιλεγμένες τιμές για τη βαθμονόμηση της προτεινόμενης στρατηγικής MIMD.

Από την άλλη μεριά, το TCP Westwood+ χρησιμοποιεί τη λειτουργία γραμμικής προσθετικής αύξησης παραθύρων πλήρως σύμφωνη με την εξίσωση (1). Εντούτοις, σε περίπτωση που μια απώλεια πακέτου ανιχνεύεται, η τιμή του παραθύρου συμφόρησης W , ενημερώνεται ως εξής:

$$W \leftarrow \max\left(2, \frac{BWE \cdot RTT_{min}}{seg_size}\right). \quad (5)$$

όπου BWE είναι το κατ' εκτίμηση διαθέσιμο από άκρη-σε-άκρη εύρος ζώνης, το RTT_{min} είναι ο ελάχιστος χρόνος στρογγυλού ταξιδιού που μετριέται κατά τη διάρκεια της σύνδεσης και το seg_size είναι το μέγεθος ενός τμήματος του TCP, σε bits.

Ένας συνδυασμός του TCP Westwood+ με τον αλγόριθμο πολλαπλασιαστικής αύξησης που προτείνεται στο Scalable TCP δημιουργεί τον αλγόριθμο Πολλαπλασιαστικής Αύξησης προσαρμοστικής Μείωσης (MIAD) για τον έλεγχο συμφόρησης. Κατά κανόνα, θα έπρεπε να φέρει την εξελικτική δυνατότητα στο TCP Westwood+ με την μείωση του χρόνου που ξοδεύτηκε στην αποκατάσταση απώλειας για δίκτυα με μεγάλη καθυστέρηση στο εύρος ζώνης. Εντούτοις, αυτό απαιτεί (όπως με το Scalable TCP) δυναμικό επαναδιαμόρφωση των παραμέτρων α και β για δίκτυα με χαμηλό BDP ή στην περίπτωση εμφάνισης διαλείμματος, προκειμένου να το καταστήσει τουλάχιστον τόσο επιθετικό, όσο η παραδοσιακή στρατηγική προσθετικής αύξησης.

Μια άλλη δυσμενής ιδιοκτησία που παρουσιάζεται από τους αλγόριθμους MIAD έγκειται σε μεγάλα βήματα αύξησης παραθύρου καθώς προσεγγίζουν το κατώτατο όριο της διαθέσιμης χωρητικότητας. Πράγματι, όσο μεγαλύτερη η βάση για τον εκθέτη (δηλ., η τρέχουσα τιμή παραθύρου), τόσο μεγαλύτερο το βήμα αύξησης. Αυτό σημαίνει ότι οι πολλαπλές πτώσεις πακέτου είναι πιθανό να συμβούν όταν το πρωτόκολλο πλησιάζει τη διαθέσιμη χωρητικότητα, ως αποτέλεσμα της υπερχειλίσης buffer στη άκρη δυσχερούς δρομολογητή.

Οι πολλαπλές πτώσεις πακέτων σπαταλούν τους πόρους δικτύου στη διαδρομή μεταξύ της πηγής και της δυσχερής διασύνδεσης, ενώ την ίδια στιγμή έχουν μια αρνητική επίπτωση στον αλγόριθμο εκτίμησης εύρους ζώνης του Westwood+. Το γενικό αποτέλεσμα είναι μια υποτίμηση του εύρους ζώνης που είναι διαθέσιμο για τη σύνδεση, με μια γενική αρνητική επίπτωση στην απόδοση του πρωτοκόλλου. Αυτό το φαινόμενο έχει επιβεβαιωθεί από εκτενής αριθμητικές προσομοιώσεις (που δεν αναφέρονται στην εργασία λόγω έλλειψης χώρου).

Λαμβάνοντας υπόψη τέτοιες εκτιμήσεις, είναι δυνατές δύο προσεγγίσεις. Η πρώτη θα έγκειται στο σχέδιο ενός νέου φίλτρου για την εκτίμηση του εύρους ζώνης, ικανή να ανταπεξέρθει με τη συμπεριφορά που προκαλείται από τους αλγορίθμους MIAD. Εντούτοις, πιστεύουμε ότι αυτός δεν είναι ο σωστός τρόπος να πάμε, δεδομένου ότι θα οδηγούσε στην εισαγωγή ενός λιγότερο ανταποκρινόμενου μηχανισμού εκτίμησης εύρους ζώνης. Η άλλη προσέγγιση, αυτή που θα ακολουθήσουμε στα επόμενα τμήματα, έγκειται στην εισαγωγή μιας διαφορετικής λειτουργίας για την αύξηση του παραθύρου συμφόρησης, ικανή να ταιριάζει καλά τα χαρακτηριστικά γνωρίσματα του μηχανισμού εκτίμησης εύρους ζώνης του Westwood+.

Με βάση τις προαναφερθείσες παρατηρήσεις, προτείνουμε ένα μηχανισμό αύξησης παραθύρου συμφόρησης ικανό να:

- (1) είναι επιθετικότερος από τον αλγόριθμο παραδοσιακής προσθετικής αύξησης σε κάθε στιγμή του χρόνου, προκειμένου εγγυηθεί ίση ή καλύτερη απόδοση της ρυθμοαπόδοσης.
- (2) παρέχει μια γρήγορη αύξηση παραθύρου για τις χαμηλές τιμές του W , ενώ όντας ακριβής στην προσέγγιση της διαθέσιμης χωρητικότητας αγωγού
- (3) παρουσιάζει μια μικρή ευαισθησία όσον αφορά την τιμή του RTT.

Σε αυτό το έγγραφο, ερευνάμε τη λογαριθμική λειτουργία σαν υποψήφια για την αντικατάσταση της γραμμικής εξέλιξης του παραθύρου συμφόρησης. Η προτεινόμενη λειτουργία λειτουργεί κατά τη διάρκεια της φάσης του TCP της αποφυγής συμφόρησης, ενώ η φάση αργής εκκίνησης αφήνεται αμετάβλητη. Ο

αλγόριθμος διευκρινίζει ότι, για κάθε ένα λαμβανόμενο πακέτο αναγνώρισης, το παράθυρο συμφόρησης θα πρέπει να ενημερωθεί σύμφωνα με:

$$W \leftarrow W + \frac{W_{max} - W}{\alpha \cdot W} . \quad (6)$$

Το μέγιστο μέγεθος παραθύρων W_{max} ορίζεται ως η τιμή του παραθύρου συμφόρησης στην οποία ανιχνεύθηκε το τελευταίο γεγονός απώλειας πακέτου. Αντιπροσωπεύει βασικά μια (κατά προσέγγιση) εκτίμηση του γενικού διαθέσιμου μεγέθους αγωγού³.

³ Η μελέτη πιο αποτελεσματικών αλγόριθμων για να εκτιμηθεί μια τέτοια τιμή είναι έξω από το βασικό πεδίο δράσης αυτής της εργασίας και επομένως αφήνεται για μελλοντικές εργασίες.

Η εξίσωση (6) επιλέγει την επόμενη αξία του W επιλέγοντας ένα σημείο της τιμής του τρέχοντος W και του W_{max} διαβαθμισμένο από την παράμετρο α . Η παράμετρος α ελέγχει το επίπεδο επιθετικότητας του αλγορίθμου αύξησης δυναμικά: το α είναι αυξανόμενο από ένα συντελεστή δύο σε περίπτωση που μια πτώση πακέτου ανιχνεύεται προτού να φθάσει το W στην τιμή W_{max} , ενώ μειώνεται από τον ίδιο συντελεστή για κάθε αλάνθαστη αύξηση στο ίδιο διάστημα. Εμείς περιορίζουμε χαμηλότερα το α κατά 2 (αρχική τιμή) από τις εκτιμήσεις όπου τιμές μικρότερες από 2 θα καταστήσουν την αύξηση πάρα πολύ επιθετική.

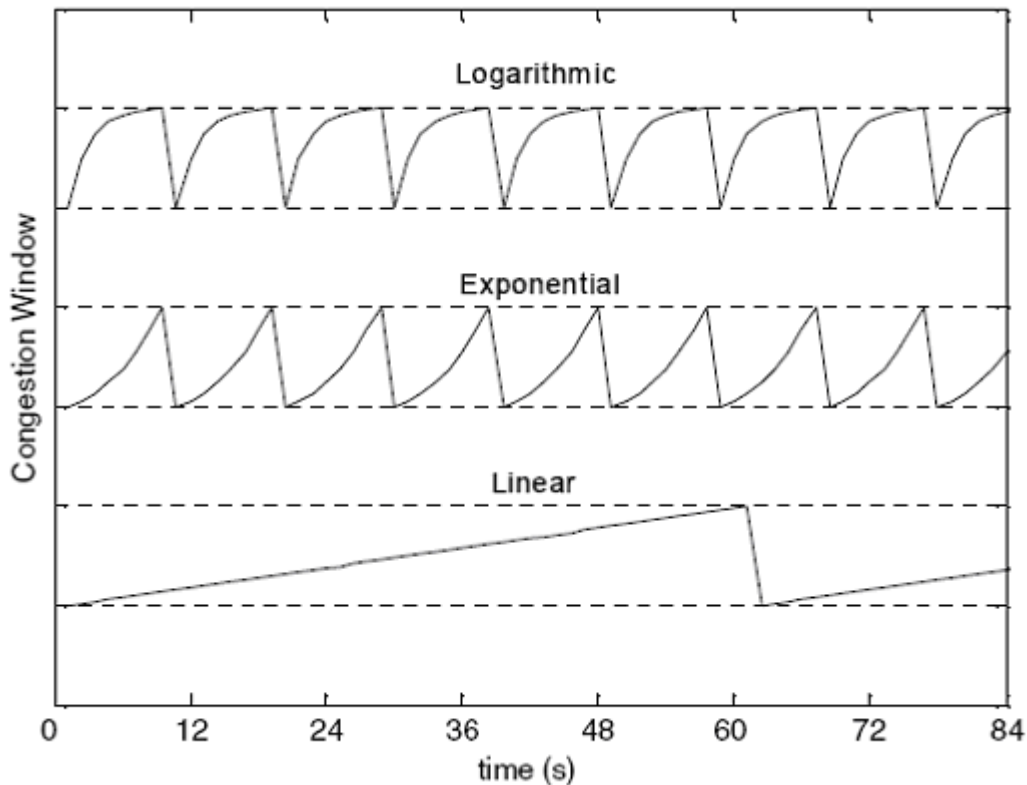
Σύμφωνα με την εξίσωση (6) η αύξηση της τιμής του παραθύρου συμφόρησης είναι επιθετική παρουσίας των χαμηλών τιμών του W , ενώ γίνεται συνετότερη κατά τη προσέγγιση του W_{max} , κατά συνέπεια ικανοποιώντας τους στόχους 2 και 3 του σχεδίου. Εντούτοις, προκειμένου εξασφαλίσουμε ότι η σχεδιασμένη λειτουργία είναι πάντα τόσο επιθετική όσο τουλάχιστον ο τυποποιημένος μηχανισμός TCP, δεσμεύσαμε την ελάχιστη αύξηση παραθύρου σε ένα πακέτο ανά παράθυρο:

$$W \leftarrow W + \max\left(\frac{W_{max} - W}{\alpha \cdot W}, \frac{1}{W}\right) . \quad (7)$$

Η προτεινόμενη προσέγγιση μπορεί να υπενθυμίσει τη Λειτουργία Δυναδικής Αύξησης Ελέγχου Συμφόρησης (BIC) που παρουσιάζεται στο [6], η οποία έχει μια εννοιολογική ομοιότητα με την προτεινόμενη λειτουργία (αφήνοντας το $\alpha = 2$). Εντούτοις, σε αντίθεση με τη BIC, η προτεινόμενη αύξηση παραθύρου ακολουθεί πάντα μια λογαριθμική συμπεριφορά με $W < W_{max}$, ενώ το επίπεδο επιθετικότητας είναι ελεγχόμενο από το διαβαθμιζόμενο α . Ως αποτέλεσμα συμπεριφέρεται εξίσου καλά και για μικρές και για μεγάλες τιμές παραθύρου.

Στο [σχήμα 35](#) εκθέσαμε την τάση αύξησης του παραθύρου συμφόρησης για τους μηχανισμούς προσθετικής, πολλαπλασιαστικής και λογαριθμικής αύξησης, αντίστοιχα. Αφού το μέρος μείωσης του μηχανισμού ελέγχου συμφόρησης του Westwood+ αφήνεται απείραχτο, τέτοιες λειτουργίες αύξησης θα πρέπει (στην περίπτωση μιας εκτίμησης τέλει εύρους ζώνης) να είναι ενεργές μόνο όταν το W είναι στο εμπεριεχόμενο εύρος μεταξύ του BDP που πραγματικά δόθηκε από την άκρη-σε-άκρη διασύνδεση και του BDP που είναι συνολικά διαθέσιμο (συμπεριλαμβανομένης και της απομόνωσης)

Το επίπεδο επιθετικότητας των πραγματευόμενων στρατηγικών αντιστοιχεί στην ταχύτητα αύξησης παραθύρου. Η απόδοση της TCP ροής είναι ανάλογη του μεγέθους παραθύρου σε κάθε χρονική στιγμή, δηλαδή, μεγαλύτερα χρονικά διαστήματα που ξοδεύονται σε υψηλότερη τιμή παραθύρου καταλήγουν σε αύξησης της απόδοσης ρυθμοαπόδοσης. Έτσι, διαφορετικές στρατηγικές παραθύρου συμφόρησης μπορούν να συγκριθούν από τα ακέραια τετράγωνα των λειτουργιών αύξησης.

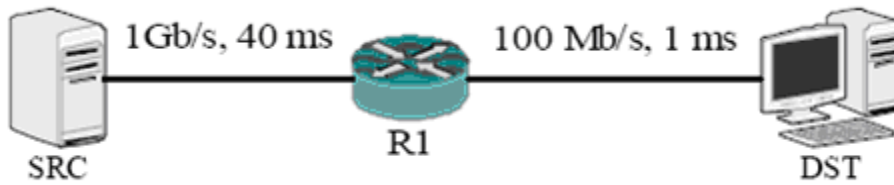


Σχήμα 35: Τάσεις της αύξησης του παραθύρου για διάφορες υποψήφιες ενέργειες.

3.8.3 ΑΞΙΟΛΟΓΗΣΗ ΑΠΟΔΟΣΗΣ

Η απόδοση των προτεινόμενων αυξήσεων του TCP Westwood+ έχει αναλυθεί χρησιμοποιώντας τον ns-2 προσομοιωτή δικτύου [10] στην τωρινή του (2.28) εκδοχή. Τα πρότυπα συναρτησιακά στοιχεία του TCP Westwood+ αποκτήθηκαν από [72], στην κορυφή των οποίων η προτεινόμενη λειτουργία λογαριθμικής αύξησης πραγματοποιήθηκε με α ίσο με 4.

Προκειμένου αξιολογήσουμε την απόδοση του προτεινόμενου αλγόριθμου, αποφασίσαμε να περιοριστούμε σε ένα απλό σενάριο, περιλαμβάνοντας μία μόνο δυσχέρεια. Το προτεινόμενο σενάριο προσομοίωσης παρουσιάζεται στο [σχήμα 36](#). Αυτό περιλαμβάνει τρεις κόμβους συνδεδεμένοι με δύο διασυνδέσεις με διαφορετικά χαρακτηριστικά χωρητικότητας. Ο κόμβος πηγής (SRC) είναι συνδεδεμένος με κόμβο προορισμού (DST) μέσω μιας διαδρομής δύο-hop.



Σχήμα 36: Το προτεινόμενο σενάριο προσομοίωσης.

Η διασύνδεση μεταξύ SRC και του δρομολογητή R1 παρουσιάζει χωρητικότητα 1Gb/s και μία καθυστέρηση των 40ms. Από την άλλη, η διασύνδεση μεταξύ DST και του δρομολογητή R1 παρουσιάζει εύρος ζώνης 100 Mb/s και μία καθυστέρηση του 1ms, αναπαριστά χοντρικά, ανάλογα της χωρητικότητας ένα δίκτυο LAN Ethernet. Τέτοια διαμόρφωση επιτρέπει σε 4.1 Mb δεδομένων να είναι σε εκκρεμότητα στο δίκτυο, που αντιστοιχούν σε μία τιμή του BDP 683 πακέτα TCP των 1500-byte. Το μέγεθος της εξερχόμενης σειράς αναμονής των κόμβων δικτύου είναι ρυθμισμένο στα 50 πακέτα, που αντιστοιχούν σε 75 Kbytes. Μία επίμονη TCP σύνδεση προσομοιώνεται, που αντιστοιχεί στο download (από το SRC στο DST) ενός απείρωσ μεγάλου αρχείου. Κάθε προσομοίωση σταματά μετά από 2000 δευτερόλεπτα. Προκειμένου καταλάβουμε τις βελτιώσεις απόδοσης που μπορούν να επιτευχθούν με την προτεινόμενη λύση, το Ποσοστό Λάθους του Πακέτου (Packet Error Rate: PER) είναι κυμαινόμενο στη δυσχερή διασύνδεση.

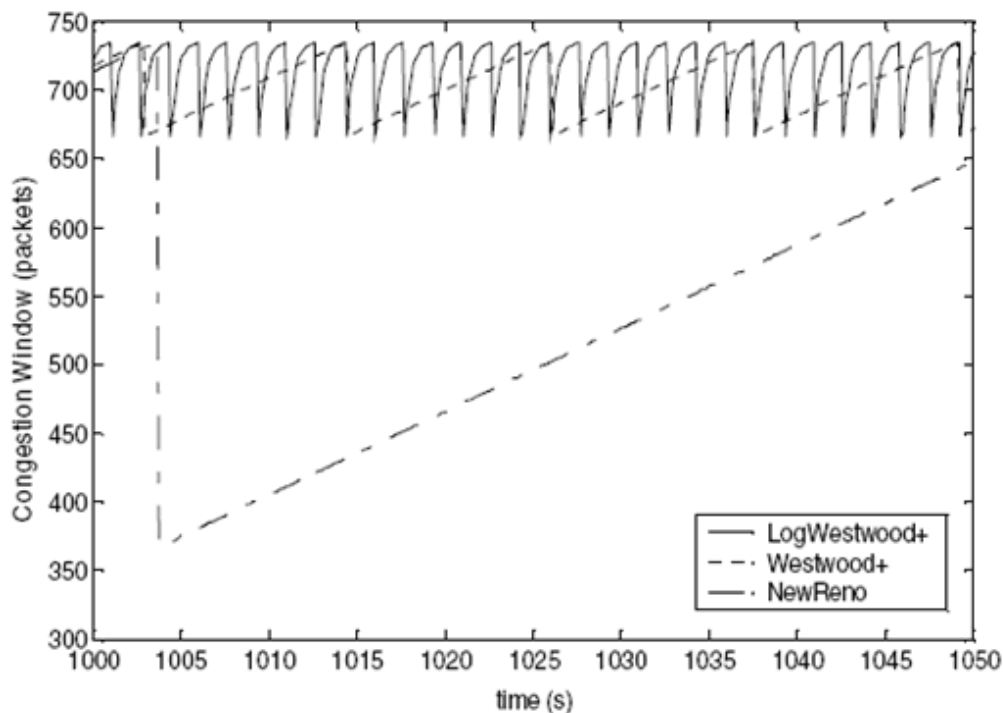
Το προτεινόμενο πρωτόκολλο, που ονομάζεται *LogWestwood+*, είναι συγκρινόμενο και με τους δύο προγόνους του, το *Westwood+* καθώς επίσης και το πρότυπο TCP NewReno με επιλεγμένες αναγνωρίσεις (SACK), η πιο ευρέως διαδεδομένη εκδοχή του TCP πρόσφατα απασχολούμενη στο Διαδίκτυο.

Η ποσότητα δεδομένων που παραδίδονται κατά την συνολική διάρκεια της προσομοίωσης, η οποία αντιστοιχεί στον αριθμό ακολουθίας του TCP, διαλέγεται ως το βασικό μετρικό για τη σύγκριση της απόδοσης των τριών πρωτοκόλλων.

Στο [σχήμα 37](#), εκθέτονται οι τιμές της δυναμικής του παραθύρου συμφόρησης, που παρατηρήθηκαν κατά ένα χρονικό διάστημα 50s. Από το γράφημα είναι εύκολο να δούμε ότι το *LogWestwood+* και το *Westwood+* ξεπερνούν σε απόδοση το πρότυπο TCP. Επιπλέον, το NewReno διχοτομεί την τιμή του παραθύρου συμφόρησης έως ότου υπερχειλίζει το δίκτυο, το TCP *Westwood+* όπως και το *LogWestwood+* μειώνονται στην εκτίμηση του εύρους ζώνης (το οποίο αντιστοιχεί

στην πραγματική χωρητικότητα διασύνδεσης, εξαιρώντας τους πόρους του buffer που είναι παρόντες στη διασύνδεση).

Στο σενάριο αξιολόγησής μας, ο AIMD αλγόριθμος του NewReno απαιτεί 60 δευτερόλεπτα να συνέλθει από το γεγονός της απώλειας. Ο μηχανισμός AIAD εισάγεται από το TCP Westwood+ το οποίο παίρνει 12 δευτερόλεπτα για να αυξήσει το παράθυρο από την εκτίμηση του εύρους ζώνης στη συνολική χωρητικότητα δικτύου, ενώ το LogWestwood+ απαιτεί μόνο 1.6 δευτερόλεπτα για να εκτελέσει την ίδια λειτουργία.



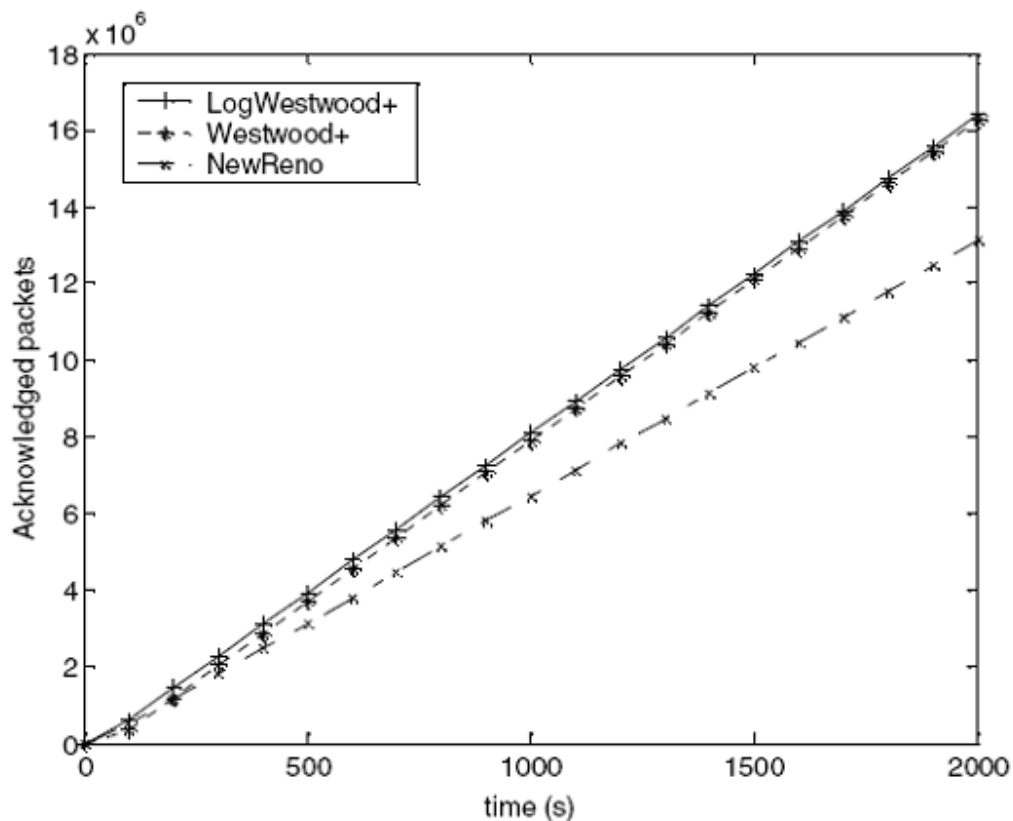
Σχήμα 37: Αύξηση παραθύρου των NewReno, Westwood+ και LogWestwood+

Το LogWestwood+ και το Westwood+ διατηρούν τα παράθυρα συμφόρησης τους κατά το διάστημα μεταξύ της εκτιμώμενης χωρητικότητας της διασύνδεσης και της συνολικής χωρητικότητας (το οποίο περιλαμβάνει την απομόνωση κατά μήκος της διαδρομής δεδομένων). Ωστόσο, το γεγονός ότι η ελάχιστη τιμή του παραθύρου συμφόρησης δεν πέφτει κάτω από τη χωρητικότητα διασύνδεσης κάνει αυτά να αποδίδουν ίσα σε περίπτωση που δεν υπάρχουν λάθη στη διασύνδεση.

Στο [σχήμα 38](#), παριστάναμε γραφικά τις αποδόσεις που επιτεύχθηκαν από τις τρεις λύσεις απουσίας απωλειών πακέτου, μετρημένες σε επίπεδα αριθμού

αναγνωρισμένων πακέτων εναντίον χρόνου προσομοίωσης. Ο αριθμός των αναγνωρισμένων πακέτων αντιστοιχεί στον αριθμό συχνότητας της TCP ροής στην προσομοίωσή μας η οποία, σε αντίθεση με τις πραγματικές εφαρμογές, πάντα ξεκινά από το μηδέν.

Η χαμηλή απόδοση που πετυχαίνεται από το TCP NewReno οφείλεται στο γεγονός ότι η «τυφλή» συμφόρηση μειώνει διχοτομώντας το παράθυρο συμφόρησης κάτω από το κατώτατο όριο της διαθέσιμης χωρητικότητας δικτύου. Κατόπιν, υποχρησιμοποιεί το εύρος ζώνης για όσο χρόνο χρειάζεται ο αλγόριθμος συντηρητικής προσθετικής αύξησης για να φτάσει ξανά τη χωρητικότητα διασύνδεσης.



Σχήμα 38: Αριθμός των αναγνωριζόμενων πακέτων έναντι του χρόνου προσομοίωσης των NewReno, Westwood+ και LogWestwood+

Προκειμένου να αξιολογήσουμε το κέρδος απόδοσης που πετυχαίνεται από τη προτεινόμενη λογαριθμική λειτουργία παραθύρου, μια συσκευή παραγωγής λάθους συνδέθηκε στη δυσχερή διασύνδεση μεταξύ του δρομολογητή R1 και του κόμβου DST (βλέπε [σχήμα 39](#)). Τα εισαγόμενα λάθη είναι σε επίπεδο πακέτου και

ακολουθούν ένα χωρίς μνήμη μοτίβο, κάθε πακέτο χάνεται με πιθανότητα $PER=10^{-7}$, 10^{-5} και 10^{-3} , αντίστοιχα. Τα αποτελέσματα που επετεύχθησαν εκθέτονται στο σχήμα 39 σε περιόδους με διάστημα εμπιστοσύνης 95% υπολογισμένες κατά μέσο όρο από 10 λειτουργίες. Στα σενάρια με λίγα ή καθόλου λάθη τα διαστήματα εμπιστοσύνης είναι ασήμαντα μικρά και έτσι δεν αναφέρονται.

Τα αποτελέσματα που αποκτήθηκαν δείχνουν το κέρδος απόδοσης του LogWestwood+ να αυξάνει με τις μεγαλύτερες PERs (πιθανότητες) το οποίο είναι η συνέπεια της πιο γρήγορης αποκατάστασης παραθύρου από την παρουσιαζόμενη απώλεια πακέτου.

Λεπτομερέστερα, καθώς το καθιερωμένο TCP NewReno υποθέτει ότι όλες οι απώλειες πακέτου προκαλούνται από τη συμφόρηση δικτύου, το TCP Westwood+ στηρίζεται σε μία εκτίμηση του διαθέσιμου εύρους ζώνης για τη μείωση παραθύρου μετά τη παραλαβή των διπλών ACKs. Ο αλγόριθμος για την εκτίμηση του εύρους ζώνης είναι βασισμένος στο ρυθμό των εισερχόμενων ACKs, το οποίο σημαίνει ότι δε λαμβάνει υπόψη τα πακέτα που δεν μεταδόθηκαν με επιτυχία, αλλά για ποια δίοδο μετάδοσης σπαταλήθηκαν οι πόροι.

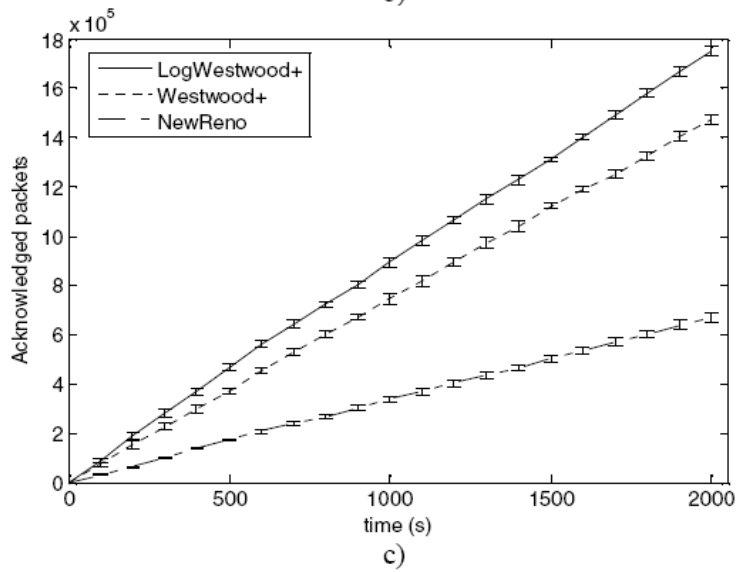
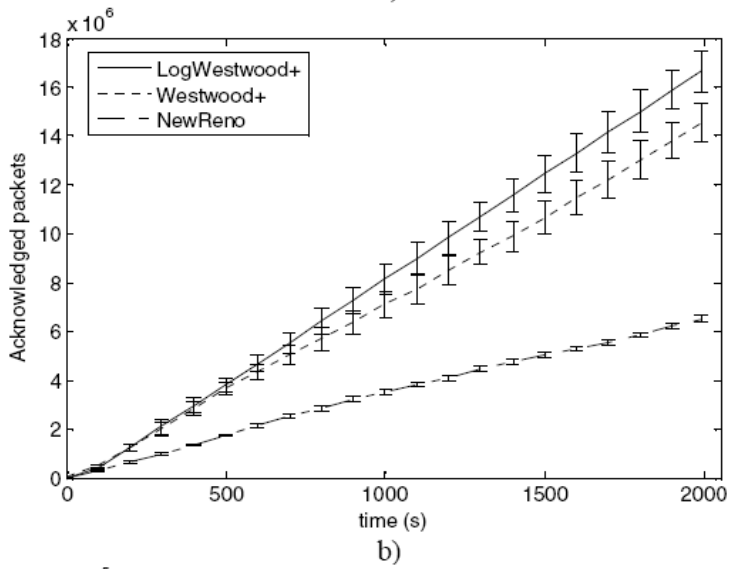
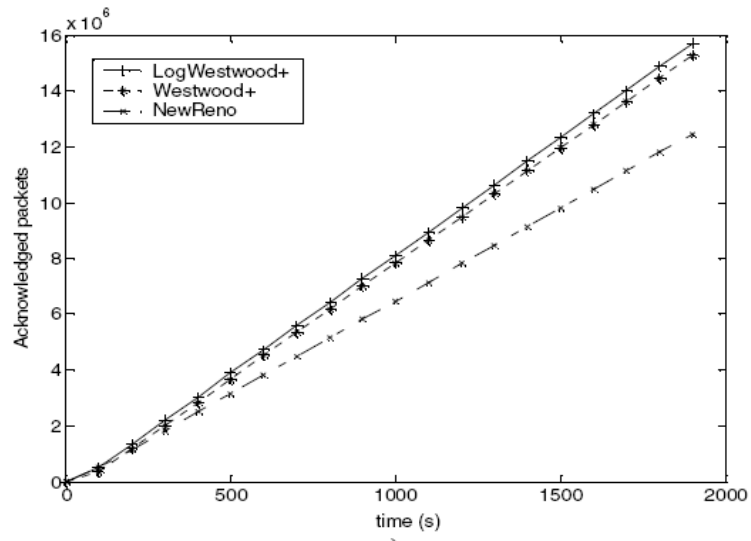
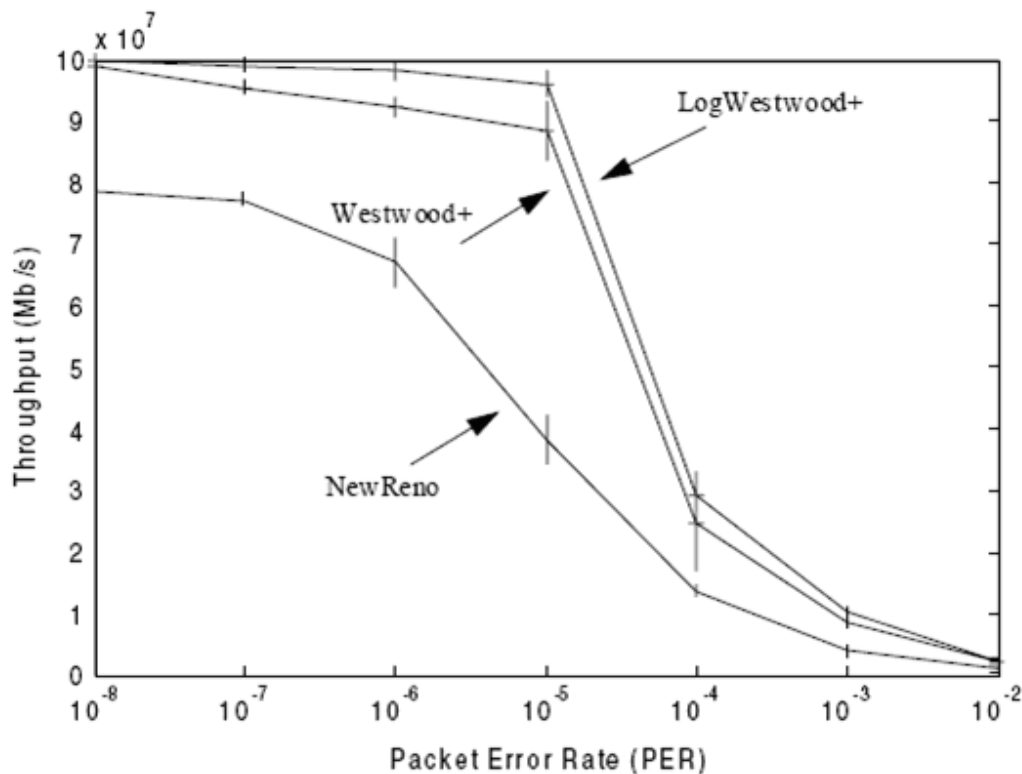


Fig. 5: Performance comparison in the presence of random errors, PER equal to a) 10^{-7} , b) 10^{-5} , and c) 10^{-3} .

Σχήμα 39 Σύγκριση για διάφορες καταστάσεις

Ως αποτέλεσμα, το TCP Westwood+ μειώνει το παράθυρό του σε μία τιμή η οποία είναι χαμηλότερη από τους διαθέσιμους πόρους δίοδου. Έτσι, η δίοδος είναι υποχρησιμοποιημένη για το χρόνο που σπαταλήθηκε στη φάση προσθετικής αύξησης παραθύρου μέχρι το εύρος ζώνης της διασύνδεσης.



Σχήμα 40: Σύγκριση εκτέλεσης ρυθμοαπόδοσης με την παρουσία τυχαίων λαθών.

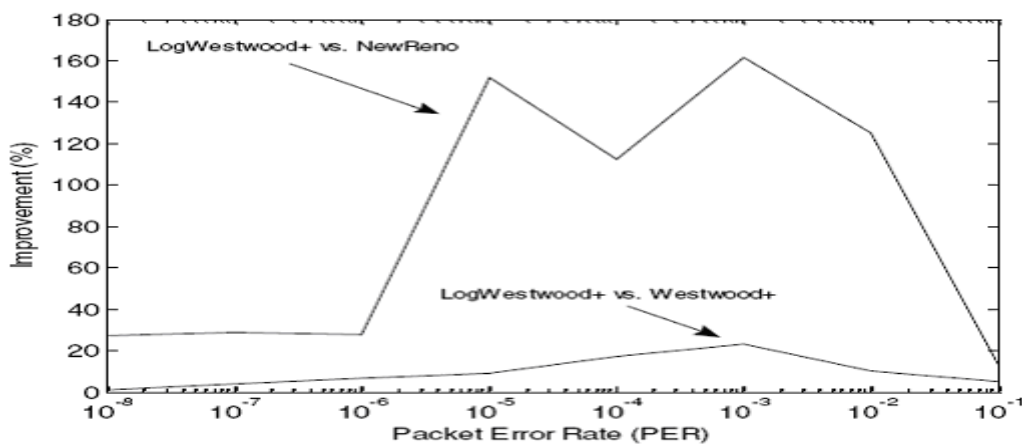
Στο [σχήμα 40](#), εκθέσαμε το μέσο όρο ρυθμοαπόδοσης που επιτεύχθηκε από τις τρεις εξεταζόμενες εφαρμογές TCP λειτουργώντας σε διαφορετικά ποσοστά λάθους. Η απόδοση του LogWestwood+ με αμεσότητα πλησιάζει το αυθεντικό αλγόριθμο του Westwood+ δείχνοντας ένα καλύτερο επίπεδο χρησιμοποίησης της από άκρη-σε-άκρη διασύνδεσης δεδομένων, ενώ το TCP NewReno πετυχαίνει χαμηλότερη ρυθμοαπόδοση σε όλες τις εξεταζόμενες ρυθμίσεις προσομοίωσης, χρησιμοποιώντας το πιο πολύ το 80% της χωρητικότητας της διασύνδεσης.

Στο [σχήμα 41,42](#), απεικονίζουμε τις αυξήσεις της απόδοσης, από άποψη αναλογίας ρυθμοαπόδοσης, που επιτεύχθηκαν από το LogWestwood+ απέναντι στο

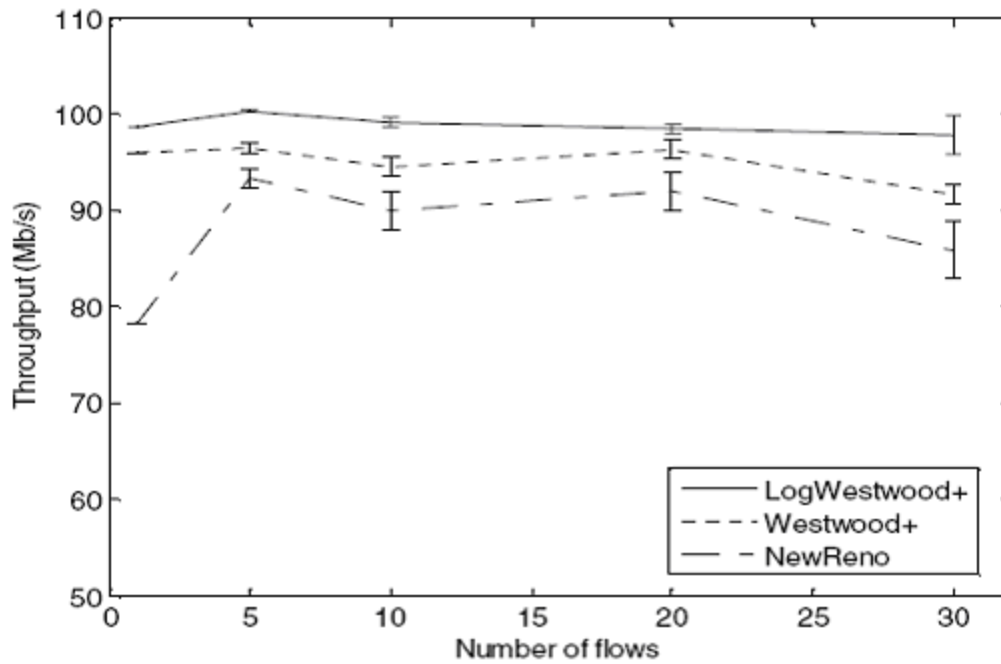
Westwood+ και στο NewReno. Τα αποτελέσματα δείχνουν ότι το επίπεδο βελτίωσης απέναντι στο Westwood+ είναι εξαιρετικά εξαρτώμενο από την PER, όντας ίσο με μηδέν (για σενάριο μονής ροής χωρίς σφάλματα) μέχρι 20% (για μία PER μέσα στο διάστημα $10^{-5} - 10^{-2}$). Εάν συγκριθεί με το NewReno, το προτεινόμενο LogWestwood+ πετυχαίνει μία βελτίωση μέχρι 160% για μία PER μέσα στο διάστημα $10^{-5} - 10^{-2}$.

Για υψηλό PER ($>10^{-1}$), και οι τρεις εκτιμημένες λύσεις πετυχαίνουν ένα παρόμοιο επίπεδο ρυθμοαπόδοσης. Αυτό μπορεί να εξηγηθεί θεωρώντας ότι, σε τέτοια κατάσταση, οι περισσότερες απώλειες πακέτου ανιχνεύονται από τη λήξη διαλείμματος. Σε αυτήν την περίπτωση, τα πρωτόκολλα ξοδεύουν τον περισσότερο χρόνο στην φάση αργή εκκίνησης, η οποία εξηγεί τα παρόμοια σχήματα απόδοσης που αποκτήθηκαν.

Το [σχήμα 41.42](#) εκθέτει μια σύγκριση απόδοσης για την περίπτωση πολλαπλών ταυτόχρονων ροών. Εξετάσαμε την αθροισμένη ρυθμοαπόδοση ως μετρική απόδοσης, και ποικίλαμε τον αριθμό ροών (που αρχίζουν στην αρχή της προσομοίωσης) από 1 έως 30. Από τα αποτελέσματα εκεί μέσα, μπορούμε να δούμε ότι το LogWestwood+ κρατά τη ρυθμοαπόδοση κοντά στη χωρητικότητα δυσχέρειας δείχνοντας καλή (95 - 99%) χρησιμοποίηση διασύνδεσης, ξεπερνώντας τις εφαρμογές και του Westwood+ και του TCP NewReno.



Σχήμα 41 σύγκριση απόδοσης για την περίπτωση πολλαπλών ταυτόχρονων ροών



Σχήμα 42 σύγκριση απόδοσης για την περίπτωση πολλαπλών ταυτόχρονων ροών

4. Συμπεράσματα

4.1 Εισαγωγή

Στο κεφάλαιο αυτό παρουσιάζονται συμπεράσματα στα οποία έχουμε καταλήξει από τις μελέτες που αναλύσαμε εκτενώς στο κεφάλαιο 3.

4.2 Συμπεράσματα ως προς την απόδοση των TCP Tahoe, του Reno, και του SACK σε Διάφορα Σενάρια

Σε αυτή την εργασία έχουμε εξετάσει το TCP Tahoe, Reno, και SACK υπό διαφορετικά σενάρια. Από τα αποτελέσματα προσομοίωσης, το TCP SACK είναι το καλύτερο μεταξύ του TCP Tahoe, Reno και SACK κάτω από οποιαδήποτε σενάρια όπως προσδοκούσαμε λόγω της ενσωμάτωσης των εκλεκτικών αναγνώρισεων μέσα

στο TCP. Για μία απώλεια πακέτου, το TCP Tahoe είναι καλύτερο από το TCP Reno αλλά με δύο και τέσσερις απώλειες πακέτων το TCP Tahoe είναι χειρότερο από το TCP Reno. Καταλήγουμε στο συμπέρασμα ότι η απόδοση του TCP Tahoe και του Reno είναι ίση και το SACK ξεπερνά σε απόδοση το TCP Tahoe και Reno.

Με τις όλο και περισσότερες κινητές συσκευές, όπως PDA και laptop διαθέσιμες στους χρήστες, τα ασύρματα δίκτυα θα γίνουν πιο δημοφιλή και θα παίξουν έναν σημαντικό ρόλο στα ενδο-δίκτυα στο μέλλον. Όταν το TCP χρησιμοποιείται για ασύρματο δίκτυο όπου οι απώλειες πακέτων προκαλούνται συχνά από τα σποραδικά υψηλά ποσοστά bit-errors και διακεκομμένη συνδεσιμότητα λόγω των handoffs, η απόδοση του TCP υποβιβάζεται λόγω της ενεργοποίησης του έλεγχος συμφόρησης. Πρόσφατα, αρκετά βελτιωμένα σχέδια TCP έχουν προταθεί συμπεριλαμβανομένου του TCP Westwood [5] και Freeze-TCP [6] για να βελτιώσουν την απόδοση του TCP για μη σχετικές με συμφόρηση απώλειες πάνω σε ασύρματα δίκτυα. Με το ανωτέρω συμπέρασμα που το SACK ξεπερνά σε απόδοση το TCP Tahoe και Reno, θα συγκρίνουμε την απόδοση του TCP SACK, TCP Westwood και Freeze -TCP πάνω σε ασύρματα δίκτυα σε διαφορετικά σενάρια.

4.3 Συμπεράσματα δυναμικής σύγκρισης ανάμεσα σε TCP Reno και TCP Veno

Το TCP Veno [10, 12, 13] εισήχθη πρόσφατα για να εξαλείψει την απόδοση του TCP που υποφέρει από τις ασύρματες διασυνδέσεις. Πραγματικές μέτρα δικτύου και ζωντανά αποτελέσματα Διαδικτύου έχουν επικυρώσει τη σημαντική βελτίωση ρυθμοαπόδοσης του Veno στα ασύρματα δίκτυα και την αρμονική συνύπαρξή του με τις συνδέσεις TCP Reno μέσα σε ενσύρματα δίκτυα.

Σε αυτό το έγγραφο, μελετήσαμε τη δυναμική του TCP Veno και την λεπτομερή αλληλεπίδραση του με το TCP Reno μέσα σε ενσύρματα δίκτυα. Παρατηρήσαμε την εμφάνιση του φαινομένου συγχρονισμού *έξω-από-τη φάση* στην εξέλιξη παραθύρου του Veno, αυτή η επιθυμητή φύση οδηγεί στην καλύτερη χρησιμοποίηση του Veno ενσύρματων διασυνδέσεων. Η περαιτέρω μελέτη μας επικυρώνει ότι είτε το Veno ή το Reno δεν θα επέφερε καμία προκατάληψη για οποιαδήποτε σύνδεση που αρχίζει αργότερα, και μπορούν δίκαια να μοιραστούν το εύρος ζώνης της δυσχερής

διασύνδεσης στα ενσύρματα δίκτυα. Από πρακτική άποψη, αυτό το πλεονέκτημα θα είναι σίγουρα χρήσιμο στην ανάπτυξη από πλευράς αποστολής του TCP Veno στο τρέχον Διαδίκτυο με υβριδικές ενσύρματες και ασύρματες διασυνδέσεις

4.4 Συμπεράσματα ως προς την απόδοση ανάμεσα σε TCP Reno και TCP Vegas

Στην ομοιογενή περίπτωση, το TCP Vegas ξεπερνά το TCP Reno λόγω του υψηλού βαθμού ρυθμοαπόδοσης και εγκυρότητας, εντούτοις, το TCP Vegas θα είναι σε δύσκολη κατάσταση όταν το TCP Reno και Vegas εκτελούνται από κοινού. Επειδή οι δύο εκδόσεις TCP μοιράζονται τους buffer στους δρομολογητές, η προοδευτική συμπεριφορά του TCP Reno και η συντηρητική συμπεριφορά του TCP Vegas δημιουργούν την αδικία όταν χρησιμοποιούνται από κοινού. Αυτήν την περίοδο οι χρήστες δεν θα υιοθετήσουν το TCP Vegas παρά την αναθεωρημένη έκδοση που αποδίδει καλύτερα από το TCP Reno μέσα στο ομοιογενές περιβάλλον.

4.5 Συμπέρασμα για ένα τροποποιημένο σχέδιο αναμετάδοσης TCP-NewReno για δίκτυο με απώλειες

Μέσω της προσομοίωσης και της σύγκρισης, δείξαμε ότι όταν η παράδοση του εκτός λειτουργίας πακέτου γίνεται σε ένα WAN το σχέδιο της μείωσης της γρήγορης αναμετάδοσης στο κατώτατο όριο K μπορεί να οδηγήσει στη υποβάθμιση της ρυθμοαπόδοσης. Ερευνούμε επίσης την επίπτωση των απωλειών αναμετάδοσης σε TCP-NewReno και συμπεραίνουμε ότι το TCP-NewReno δεν μπορεί να ανακτηθεί αποτελεσματικά από τις απώλειες αναμετάδοσης. Τέλος, καταλήγουμε στο συμπέρασμα ότι το τροποποιημένο μας TCP-NewReno λειτουργεί καλύτερα από το τρέχον TCP σε WAN/LAN με μια σύνδεση απώλειας.

4.6 Συμπεράσματα ως προς την ανταλλαγή ενέργειας/ρυθμοαπόδοσης των στρατηγικών ελέγχου λάθους του TCP (tcp error)

Τα αποτελέσματά μας για τις ανταλλαγές ενέργειας/ρυθμοαπόδοσης του TCP Tahoe, Reno και NewReno καταδεικνύουν ότι η απόδοσή τους είναι σε γενικές γραμμές σχεδόν όμοιες. Το Tahoe αποδίδει ευδιάκριτα καλύτερα όταν τα λάθη είναι εντατικά και επίμονα, εντούτοις, κανένα τους δεν επέδειξε την απαιτούμενη προσαρμοστικότητα που απαιτείται για έναν αλγόριθμο καθολικού ελέγχου λάθους στο ενσύρματο/ασύρματο ετερογενές περιβάλλον.

Υπό τις σχετικά επίμονες συνθήκες λάθους (π.χ. ξέσπασμα λαθών) οι όποιες, εντούτοις, δεν εμφανίζονται πολύ συχνά, υποχωρώντας άμεσα, όπως κάνει το Tahoe, φαίνεται να είναι η σωστή στρατηγική αφού η σταδιακή μείωση του μεγέθους παραθύρου μπορεί να έχει τις ακόλουθες ανεπιθύμητες συνέπειες:

1. Τα μεγάλα παράθυρα δεδομένων θα μπορούσαν να συνεχίσουν να διαβιβάζονται για μια χρονική περίοδο (μέχρι το παράθυρο να συρρικνωθεί) παρά την επικρατούσα φάση λάθους, και επομένως η αποτελεσματική ρυθμοαπόδοση να υποβιβάζεται και η ενεργειακή δαπάνη να αυξάνεται.
2. Τα διαλείμματα επεκτείνονται και ως εκ τούτου όταν η χωρίς λάθη φάση εμφανίζεται η παρουσία τους δεν μπορεί να ανιχνευτεί γρήγορα.
3. Χρησιμοποιώντας τη γρήγορη αποκατάσταση (Reno), πηγαίνουμε σε μια φάση αποφυγής συμφόρησης που εφαρμόζει μια γραμμική αύξηση στο παράθυρο, που ενδεχομένως να αρχίζει από μάλλον μικρό μέγεθος παραθύρου. Η αργή εκκίνηση (το Tahoe αντιδρά στις απώλειες, ακόμη και κατά τη διάρκεια της αποφυγής συμφόρησης) μπορεί να ανακτήσει γρηγορότερα μόλις περάσει μια παρατεταμένη φάση λάθους.

Η σχετική αποδοτικότητα του Reno μπορεί να παρατηρηθεί κάτω από επίμονες φάσεις με σχετικά χαμηλά ποσοστά λάθους. Αντίθετα, το NewReno μπορεί να είναι το επιλεγμένο πρωτόκολλο μόνο για περιβάλλοντα με σχετικά σποραδικά, μικρά και περιορισμένα λάθη. Σύμφωνα με τα αποτελέσματά μας, όχι μόνο οι αποδόσεις τους σε ρυθμοαπόδοσή μειώνονται κάτω από πιο επίμονες φάσεις λάθους, αλλά τα πρόσθετα γενικά έξοδα λόγω της αναμετάδοσης είναι επίσης πολύ υψηλότερα.

4.7 Συμπεράσματα ως προς την απώλεια πακέτου και την κατάσταση συμφόρησης στο TCP VENO

Σε αυτό το έγγραφο, η διάκριση της απώλειας πακέτου στο TCP Veno είναι ερευνείται κάτω από διαφορετικές συνθήκες δικτύου. Χρησιμοποιούμε το πρότυπο του Pareto κατανομής on/off για να μιμηθεί τη με διασταύρωση κυκλοφορία. Τα αποτελέσματα προσομοίωσης δείχνουν ότι η διάκριση της απώλειας πακέτου στο Veno μπορεί να αποδώσει καλά στα δίκτυα με ελαφριά και μέτρια διασταυρωμένη κυκλοφορία, αλλά σε περιβάλλον με τη βαριά διασταυρωμένη κυκλοφορία, το Veno δεν μπορεί να δανειστεί το σχέδιο ανίχνευσης συμφόρησης του Vegas και πρέπει να ανακαλύψει έναν καλύτερο για να το αντικαταστήσει εάν είναι πιθανό. Εν τούτοις, η γενική απόδοση του Veno ακόμα δουλεύει καλά όταν ανταγωνίζεται με άλλη σύνδεση που συνυπάρχει. Αυτό το ευδιάκριτο χαρακτηριστικό γνώρισμα αναμφίβολα είναι αποδιδόμενο στα ολόκληρα ελεγχόμενα σήματα - απώλεια πακέτου (που ανιχνεύεται από τα αναπαραχθέντα acks) και στη συμπληρωματική εκτίμηση κατάστασης - και η μείωση παραθύρου ενεργεί σύμφωνα με τα συντεθειμένα σήματα στο Veno. Στην πραγματικότητα, στο Διαδίκτυο, η κυκλοφορία υποβάθρου κατά μήκος της διαδρομής της σύνδεσης είναι αρκετά αβέβαιη από καιρό σε καιρό. Ένα μονό σήμα που καταγράφεται σαν απώλεια πακέτου δεν είναι αρκετό, όσα περισσότερα σήματα πρόκειται να καταγραφούν, τόσο το καλύτερο. Βασικά πρέπει να χρησιμοποιήσουμε τα καταγεγραμμένα στοιχεία στατιστικής για να συνθέσουμε τη κατάσταση συμφόρησης στην οποία μια σύνδεση TCP εξελίσσεται, ιδιαίτερα στην φάση προσθετικής αύξησης.

Κατά γενική ομολογία, μπορούμε να δούμε ότι η TCP Veno σύνδεση δανείζεται την ιδέα του σχεδίου ανίχνευσης συμφόρησης του Vegas και έξυπνα το ενσωματώνει στη φάση προσθετικής αύξησης του Reno. Η εκτίμηση της κατάστασής του χρησιμοποιείται ως συμπληρωματική πληροφορία αναφοράς για να αποφασίσει πώς να εξαλείψει την προσθετική αύξηση στο επόμενο βήμα και πόσο το παράθυρο πρόκειται να μειωθεί όταν είναι προκαλούμενη η γρήγορη αναμετάδοση. Φυσικά, κυρίως, θα μπορούσαμε επίσης να χρησιμοποιήσουμε άλλες προφητικές ανιχνεύσεις συμφόρησης (π.χ., PBM' [\[58\]](#), Ζευγάρι Πακέτου [\[57\]](#)) ή/και άλλα σχέδια καλύτερης

προφητικής ανίχνευσης συμφόρησης, ή συνδυασμούς αυτών των σχεδίων για να βελτιωθεί η εξέλιξη του Reno. Στη μελλοντική εργασία, ψάχνουμε ένα καλύτερη σχέδιο ανίχνευσης κατάστασης να αντικατασταθεί το σχέδιο ανίχνευσης του Vegas που είναι ενσωματωμένο στο VenO.

4.8 Συμπεράσματα ως προς την αύξηση του tcp westwood+ σε μεγάλης ταχύτητας ευρέα δίκτυα

Σε αυτή την εργασία, έχουμε παρουσιάσει αλγόριθμους έλεγχου συμφόρησης για τη βελτίωση της απόδοσης του TCP Westwood+ σε μεγάλα BDP δίκτυα. Η ανάλυση των λύσεων που είναι διαθέσιμες στη λογοτεχνία μας παρακίνησε για να εξετάσουμε το σχέδιο μιας μη γραμμικής λειτουργίας για την αντικατάσταση του τυποποιημένου μηχανισμού προσθετικής αύξησης. Οι ακόλουθοι στόχοι σχεδίου έχουν καθορισθεί για τη στοχευόμενη λειτουργία: (i) για είναι επιθετικότερος από τον αλγόριθμο παραδοσιακής προσθετικής αύξησης (ii) για να παρέχει μια γρήγορη αύξηση παραθύρου για τις χαμηλές τιμές του W , ενώ όντας ακριβής στην προσέγγιση της διαθέσιμης χωρητικότητας αγωγού και (iii) για να εξασφαλίσει μικρή ευαισθησία όσον αφορά το RTT. Επιδείξαμε πώς η λογαριθμική λειτουργία είναι σε θέση να επιτύχει τους προαναφερθέντες στόχους σχεδίου, οδηγώντας μας στην εισαγωγή μιας στρατηγικής που εμείς ονομάσαμε LogWestwood+.

Οι εκτενείς ns-2 προσομοιώσεις εκτελέστηκαν για να εξετάσουμε την απόδοση και την ευρωστία του προτεινόμενου αλγορίθμου, τα Αριθμητικά αποτελέσματα δείχνουν ότι η προτεινόμενη προσέγγιση είναι σε θέση να ξεπεράσει τα πρωτόκολλα Westwood+ και τυποποιημένο NewReno σε ένα ευρύ φάσμα των ρυθμίσεων προσομοίωσης.

Δύο ερευνητικές κατευθύνσεις εμφανίζουν σημαντικό ενδιαφέρον για περαιτέρω έρευνες πάνω στο θέμα: η πρώτη αφορά τη μελέτη των μηχανισμών που είναι ικανοί να προσαρμόσουν δυναμικά την αξία της παραμέτρου α στις τρέχουσες λειτουργούντες συνθήκες δικτύου. Η δεύτερη έγκειται στην εφαρμογή του προτεινόμενου μηχανισμού στους κεντρικούς υπολογιστές δικτύου και στην εκτέλεση των δοκιμών σε ένα πραγματικό περιβάλλον Διαδικτύου.

Βιβλιογραφία

- [1] S. H. Low, F. Paganini, and J. C. Doyle, "Internet Congestion Control", *IEEE Control Systems Magazine*, Vol22, No 1, pp. 28-43, 2002.
- [2] A. Kumar, "Comparative Performance Analysis of Versions of TCP in a Local Network with a Lossy Link", *IEEE/ACM Trans. Networking*, Vol6, No 4, pp. 485-498, Aug. 1998
- [3] C. Barakat, E. Altman, and W. Dabbous, "On TCP Performance in a Heterogeneous Network: A Survey", *IEEE Communications Magazine*, Vol38, No 1, pp. 40-46, Jan. 2000.
- [4] B. Sikdar, S. Kalyanaraman, and K. S. Vastola, "Analytic Models for the Latency and Steady-State Throughput of TCP Tahoe, Reno and SACK", *Proceedings of IEEE GLOBECOM*, San Antonio, TX, 2001.
- [5] A. Zanella, G. Procissi, M. Gerla, and M. Sanadidi, "TCP Westwood: Analytic Model and Performance Evaluation", *Proceedings of IEEE Globecom 2001*, Volume: 3, pp 1703- 1707, November, 2001.
- [6] T. Goff, J. Moronski, D. Phatak, and V. Gupta, "Freeze-TCP: A true end-to-end TCP enhancement mechanism for mobile environments", *Proceedings of the IEEE INFOCOM' 2000*, pp 1537- 1545
- [7] Scott Shenker, Lixia Zhang, David D. Clark, "Some Observations on the Dynamics of a Congestion Control Algorithm" *ACM SIGCOMM Computer Communication Review*, Volume 20, Issue 5 Oct. 1990
- [8] Lixia Zhang, Scott Shenker, David D. Clark, "Observations on the Dynamics of a Congestion Control Algorithm: The Effects of Two-Way Traffic". *Proceedings of the conference on Communications Architecture & Protocols*, Zurich, Switzerland Pages: 133 - 147, 1991
- [9] Cheng Peng Fu, and Soung Chang Liew, "A remedy for performance degradation of TCP Vegas in asymmetric networks," *IEEE Communications Letters*, January 2003.
- [10] Cheng Peng Fu, and Soung Chang Liew, "TCP Veno: TCP enhancement for wireless access networks," *IEEE Journal of Selected Areas in Communications*, vol. 21, no. 2, February 2003.
- [11] Cheng Peng Fu, and Soung Chang Liew, "A remedy for performance degradation of TCP Vegas in asymmetric networks," *IEEE Comm. Letters*, January 2003
- [12] Q. X. Pang, S. C. Liew, C. P. Fu, W. Wang, "Performance Study of TCP Veno over wireless LAN and RED router," *IEEE Globecom 2003*.
- [13] C. P. Fu, W. Lu, B. S. Lee, "TCP Veno Revisited," *IEEE Globecom 2003*

- [14] L.Brakmo, S.O'Malley, and L.Peterson. TCP Vegas: New techniques for congestion detection and avoidance. In Proceedings of the SIGCOMM' 94 Symposium (Aug. 1994) pages 24-35.
- [15] T.V Lakshman, Upamanyu Madhow, "The Performance of TCP/IP for Networks with High Bandwidth-Delay Products and Random Loss," *IEEE/ACM Trans. Networking*, Vol5, No. 3, June 1997
- [16] J. Mo, R. J. La, V. Anantharam, and J. Walrand. Analysis and Comparison of TCP Reno and TCP Vegas. *IEEE INFOCOM 1999*, March 1999.
- [17] Ns. Network Simulator, version 2.26. <http://www.isi.edu/nsnam/ns>.
- [18] A. Veres and M. Boda, "The Chaotic Nature of TCP Congestion Control," *IEEE INFOCOM 2000*, March 2000.
- [19] Dhiman Barman, Ibrahim Matta, "Effectiveness of Loss Labeling in Improving TCP Performance in Wired/Wireless Networks," *IEEE International Conference on Network Protocols*, Paris, France, November 12-15, 2002.
- [20] V. Jacobson, "Congestion Avoidance and Control," *ACM SIGCOMM '88*, pp. 273-288, 1988.
- [21] V. Jacobson, "Modified TCP Congestion Avoidance Algorithm," mailing list, end2end-interest, April 30, 1990.
- [22] L. S. Brakmo and L.L. Peterson, "Peterson, "TCP Vegas: End to End Congestion Avoidance on a Global Internet," *IEEE JSAC*, vol. 13, pp. 1465-1480, 1995.
- [23] O. Ait-Hellal and E. Altman, "Analysis of TCP Vegas and TCP Reno," *IEEE ICC '97*, pp. 495-499, 1997.
- [24] J. Mo, R. J. La, V. Anantharam, and J. Walrand, "Analysis and Comparison of TCP Reno and Vegas," *IEEE INFOCOM '99*, pp. 1556-1563, 1999.
- [25] W. Stevens, "TCP Slow Start, Congestion Avoidance , Fast Retransmit, and Fast Recovery Algorithms," *RFC 2001*, January 1997.
- [26] K. Fall and S. Floyd , ns-Network Simulator, <http://www-mesh.cs.berkeley>.
- [27] J. S. Ahn, P. B. Danzig, Z. Liu, and L. Yan, "Evaluation of TCP Vegas: Emulation and Experiment," <http://excalibur.usc.edu/research/vegas/doc/vegas.html>.
- [28] T. V. Lakshman, Upamanyu Madhow "The Performance of TCP/IP for Networks with High Bandwidth-Delay Products and Random Loss" in *IEEE/ACM Transaction on Networking*. vol. 5. No. 3. June 1997.
- [29] Anurag Kumar "Comparison Performance Analysis of Versions of TCP in a Local Network with a Lossy Link" in *IEEE/ACM Ttransaction on Networking*. vol. 6. no. 4. June 1998.

- [30] Aldar C. F. Chan, Danny H. K Tsang, Sanjay Gupta “Performance Analysis of TCP in the Presence of Random Losses/Errors” IEEE Globe Com '98, Sydney, Australi, 1998.
- [31] M. Allman, V. Paxson, W. Stevens, “TCP Congestion Control”, RFC 2581, April 1999.
- [32] H. Balakrishnan, V. Padmanabhan, S. Seshan, and R. Katz, “A comparison of mechanisms for improving TCP performance over wireless links”, ACM/IEEE Transactions on Networking, December 1997.
- [33] A. Chockalingam, M. Zorzi, R. Rao, “Performance of TCP on Wireless Fding Links with memory”, in Proc. of IEEE ICC'98, June 1998.
- [34] S. Floyd, T. Henderson, “The New Reno Modification to TCP’s Fast Recovery Algorithm”, RFC 2582, April 1999.
- [35] V. Jacobson, “Congestion avoidance and control” in Proc. of ACM SIGCOMM '88, August 1988.
- [36] A. Kumar, “Comparative performance analysis of versions of TCP in a local network with a lossy link”, ACM/IEEE Transactions on Networking, August 1998.
- [37] T. Lakshman, U. Madhow, “The performance of TCP/IP for networks with high bandwidth-delay products and random loss”, IEEE/ACM Transactions on Networking, pp. 336-350, June 1997.
- [38] J. Postel, Transmission Control Protocol, RFC 793, September 1981.
- [39] V. Tsaoussidis, H. Badr, R. Verma “Wave and Wait: An Energy-saving Transport Protocol for Mobile IP-Devices”, in Proc. of IEEE ICNP '99, Toronto, October 1999.
- [40] V. Tsaoussidis, H. Badr, “Energy/Throughput Tradeoffs of Reliable Transport Protocols”, in Proc. PDPTA 2000, Las Vegas, June 2000.
- [41] The X-kernel: www.cs.arizona.edu/xkernel
- [42] M. Zorzi, R. Rao, “Energy Efficiency of TCP” MoMUC '99, San Diego, California, 1999
- [43] W. Richard Stevens, “TCP/IP illustrated,” Addison-Wesley, 1994.
- [44] M. Allman and V. Paxson, and W. Stevens, RFC 2581: TCP Congestion Contol, April 1999.
- [45] Cheng Peng Fu, Soung C. Liew TCP Veno: TCP Enhancement for Transmission over Wireless Access Networks,” IEEE (JSAC) Journal of Selected Areas in Communications, Feb. 2003.

- [46] Dhiman Barman, Ibrahim Matta, “Effectiveness of Loss Labeling in Improving TCP Enhancement for Transmission over Wireless Access Networks,” *IEEE (JSAC) Journal of Selected Areas in Communications*, Feb. 2003.
- [47] J. Martins, A. Nillson and I. Rhee. “Delay-based congestion avoidance for TCP”. *IEEE/ACM Transactions on Networking* June 2003.
- [48] Saad Biaz, Hitin H. Vaidya. “Is the Round-trip Time Correlated with the Number of Packets in Flight?” *IMC '03* October 27-29, 2003.
- [49] Cheng Peng Fu, Willium Lu, Bu Sung Lee. “TCP Venno Revisited” *IEEE Globecom* 2003.
- [50] L. S. Brakmo, S. W. O’Malley, and L. Peterson. TCP Vegas: New Techniques for Congestion Detection and Avoidance. In *Proc. of ACM SIGCOMM’94*, pages 24-35, London, October 1994.
- [51] Saad Biaz, Nitin H. Vaidya. “Distinguishing Congestion Losses from Wireless Transmission Losses: A Negative Result” *Seventh International Conference on Computer Communications and Networks (IC3N)*, New Orleans, October 1998.
- [52] U. Hengartner, J. Bolliger, Th. Gross, “TCP Vegas revisited” in *Proceedings of the IEEE INFOCOM’2000*, Tel Aviv, Israel, March 2000.
- [53] M. S. Taqqu, W. Willinger, and R. Sherman, “Proof of a Fundamental Result in Self-Similar Traffic Modeling,” *ACM Computer Communications Review*, pp. 5-23, Apr. 1997.
- [54] J. Mo, R. J. La, V. Anantharam, and J. Walrand, Analysis and Comparison of TCP. Reno and Vegas. In *Proceeding of INFOCOM’99*, pages 1556-1563, March 1999.
- [55] Richard J. La, V. Anantharam, Jean Walrand, and Venkat Anantharam , “Issues in TCP Vegas”.
<http://www.eecs.berkeley.edu/~ananth/1999-2001/Richard/IssuesInTCPVegas.pdf>
- [56] ns2: network simulator. <http://www-mash.cs.berkeley.edu/ns>.
- [57] S. Keshav, “A Control – Theoretic Approach to Flow Control,” in *Proceeding ACM SIGCOMM*. Sept. 1991
- [58] V. Paxson, “Measurements and Analysis of End-to-End Internet Dynamics”, PhD thesis, University of California, Berkeley, April 1997
- [59] J. B. Postel, “Transmission Control Protocol,” *RFC 793*, September 1981.
- [60] G. Miller, K. Thompson, and R. Wilder, “Wide-area Internet traffic patterns and characteristics,” *IEEE Network*, pp. 10 – 23, November/December 1997.
- [61] N. Brownlee and K. Claffy, “Understanding Internet traffic streams: Dragonflies and tortoises,” *IEEE Comm. Mag.*, vol. 40, pp. 110 – 117, Oct. 2002.

- [62] V. Jacobson, "Congestion Avoidance and Control", in *Proc. of ACM SIGCOMM*, pp. 314 – 329, August 1988, Standford, CA.
- [63] S. Floyd, "HighSpeed TCP for large congestion windows," *RFC 3649*, December 2003.
- [64] Tom Kelly, " Scalable TCP: improving performance in highspeed wide area networks," *ACM Comp. Comm. Rev.*, vol. 33 , pp. 83 – 91, April 2003.
- [65] Cheng Jin, D.X. Wei, S.H. Low, "FAST TCP: motivation, architecture, algorithms, performance," in *Proc. of IEEE INFOCOM*, vol. 4, pp. 2490 – 2501, March 2004, Hong Kong.
- [66] Lisong Xu; K. Harfoush, and Injong Rhee, "Binary increase congestion control (BIC) for fast long-distance networks", in *Proc. of IEEE INFOCOM*, vol. 4, pp. 2514 – 2524, March 2004, Hong Kong.
- [67] S. Mascolo, C. Casetti, M. Gerla, M. Sanadidi, and R. Wang, "TCP Westwood: End-to-end Bandwidth Estimation of Efficient Transport over Wired and Wireless Networks", in *Proc. of ACM Mobicom*, July, 2001, Rome, Italy.
- [68] L. A. Grieco and S. Mascolo, "Performance evaluation and comparison of Westwood+, New Reno, and Vegas TCP congestion control", *ACM Comp. Comm. Rev.*, vol. 34, pp. 25 – 38, April 2004.
- [69] R. Wang, G. Pau, K. Yamada, M. Y. Sanadidi, M. Gerla "TCP Startup Performance in Large Bandwidth Delay Networks", in *Proc. Of INFOCOM 2004*, vol. 2, pp. 796 – 805, March 2004, Hong Kong.
- [70] R.Wang, K. Yamada, M. Y. Sanadidi, and M. Gerla "TCP with senderside intelligence to handle dynamic, large, leaky pipes," *IEEE J. Sel. Ar.Comm.*, vol. 23, n. 2, pp. 235-248, 2005.
- [71] NS-2 Simulator. <http://www.isi.edu/nsnam/ns>
- [72] TCP Westwood+ modules for ns2,
<http://193.204.59.68/mascolo/tcp%20westwood/modules.htm>