

Ατομική Διπλωματική Εργασία

**ΑΞΙΟΛΟΓΗΣΗ ΕΠΙΔΟΣΗΣ ΠΡΩΤΟΚΟΛΛΩΝ
ΜΕΤΑΦΟΡΑΣ ΕΦΑΡΜΟΓΩΝ ΠΟΛΥΜΕΣΩΝ**

Αντρέας Φλωρίδης

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ



ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Ιούνιος 2005

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΑΞΙΟΛΟΓΗΣΗ ΕΠΙΛΟΣΗΣ ΠΡΩΤΟΚΟΛΛΩΝ
ΜΕΤΑΦΟΡΑΣ ΕΦΑΡΜΟΓΩΝ ΠΟΛΥΜΕΣΩΝ

Αντρέας Φλωρίδης

Επιβλέπων Καθηγητής
Αντρέας Πιτσιλλίδης

Η Ατομική αυτή Διπλωματική Εργασία υποβλήθηκε προς μερική εκπλήρωση
των απαιτήσεων απόκτησης του πτυχίου Πληροφορικής
του Πανεπιστημίου Κύπρου

Ιούνιος 2005

Ευχαριστίες

Θα ήθελα να εκφράσω τις βαθιές μου ευχαριστίες στον επιβλέποντα καθηγητή μου, κύριο Ανδρέα Πιτσιλίδη για την βοήθεια που μου προσέφερε καθ' όλη την διάρκεια της ατομικής διπλωματικής εργασίας. Επίσης θα ήθελα να ευχαριστήσω τον κύριο Χρυσόστομο Χρυσοστόμου ο οποίος με τις συμβουλές και το υλικό που μου έχει δώσει βοήθησε στην επιτυχή ολοκλήρωση της έρευνας μου.

Περίληψη

Σκοπός της Ατομικής Διπλωματικής Εργασίας είναι η προσομοίωση των νέων πρωτοκόλλων μεταφοράς δεδομένων με σκοπό την μέτρηση της απόδοσης τους και την ανάλυση του κατά πόσον τα πρωτόκολλα αυτά είναι ικανά να μεταφέρουν πολυμεσικά δεδομένα, δηλαδή εικόνα και ήχο, όταν συναγωνίζονται με ροές πρωτοκόλλων που μεταφέρουν αξιόπιστα δεδομένα όπως για παράδειγμα το TCP.

Συγκεκριμένα γίνεται ανάλυση των πρωτοκόλλων TFRC, SCTP, RTP και UDP σε σχέση με το TCP με χρήση του προσομοιωτή Network Simulator (Version 2.28). Επειδή το κόστος σχεδίασης και υλοποίησης των πρωτοκόλλων είναι πολύ μεγάλο η χρήση προσομοιωτή καθίσταται αναγκαία.

Έχουμε σχεδιάσει και υλοποιήσει τη βασική τοπολογία στον προσομοιωτή και παίρνουμε διάφορες μετρικές όπως τα ποσοστά χρήσης των συνδέσεων, την καθυστέρηση των πακέτων στα άκρα, τον ρυθμό και το ποσοστό με το οποίο χάνονται τα πακέτα σε περίπτωση συμφόρησης, την καθυστέρηση στους δρομολογητές με σκοπό να κατανοήσουμε και να αναλύσουμε την συμπεριφορά των πρωτοκόλλων αυτών.

Μέχρι σήμερα δεν έχει γίνει κάποια προσομοίωση και αξιολόγηση των πρωτοκόλλων αυτών με χρήση πολυμεσικών αρχείων που είναι και το κύριο φορτίο που θα μεταφέρουν αυτά τα πρωτόκολλα. Επιχειρούμε με τις προσομοιώσεις μας να χρησιμοποιήσουμε τα νέα αυτά πρωτόκολλα όπως θα χρησιμοποιούνταν και στην πραγματικότητα. Για το σκοπό αυτό έχουμε καταφέρει να προσομοιώσουμε την χρήση πραγματικής ταινίας μέσα από τον προσομοιωτή.

Η Ατομική Διπλωματική Εργασία χωρίζεται σε τρεις κύριες ενότητες : Στην παρουσίαση των πρωτοκόλλων, την περιγραφή των πειραμάτων και την ανάλυση των αποτελεσμάτων.

Περιεχόμενα

Κεφάλαιο 1 : Εισαγωγή	1
1.1 Γενικά.....	1
1.2 Υποκίνηση Έρευνας	2
1.3 Σκιαγράφηση Έρευνας.....	3
Κεφάλαιο 2 : Περιγραφή Πρωτοκόλλων.....	4
2.1 Transport Control Protocol (TCP)	4
2.2 User Datagram Protocol (UDP)	8
2.3 TCP Friendly Rate Control (TFRC)	11
2.4 Streaming Control Transport Protocol (SCTP).....	23
2.5 Real-time Transport Protocol (RTP).....	32
Κεφάλαιο 3 : Περιγραφή και Παρουσίαση Προσομοιώσεων	39
3.1 Ανάλυση Τοπολογίας - Μετρικές	39
3.2 Προσομοιώσεις με χρήση του TFRC	41
3.3 Προσομοιώσεις με χρήση του SCTP	59
3.4 Προσομοιώσεις με χρήση του RTP	76
3.5 Προσομοιώσεις με χρήση του UDP.....	93
Κεφάλαιο 4 : Ανάλυση Αποτελεσμάτων	111
4.1 Ανάλυση Αποτελεσμάτων Πρωτοκόλλου UDP	111
4.2 Ανάλυση Αποτελεσμάτων Πρωτοκόλλου RTP	115
4.3 Ανάλυση Αποτελεσμάτων Πρωτοκόλλου TFRC	116
4.4 Ανάλυση Αποτελεσμάτων Πρωτοκόλλου SCTP	117
Κεφάλαιο 5 : Νέες Τεχνολογίες Πρωτοκόλλων.....	121
5.1 Η προσθήκη του ECN στο IP	121
5.2 ECN με αποστολή μηνυμάτων.....	122
Κεφάλαιο 6 : Συμπεράσματα	127
6.1 Συμπέρασμα Ατομικής Διπλωματικής Εργασίας	127
6.2 Εισηγήσεις για μελλοντική Εργασία.....	130
Βιβλιογραφία	131

Παράρτημα Α	Α-1
Παράρτημα Β.....	Β-2
Παράρτημα Γ.....	Γ-5
Παράρτημα Δ.....	Δ-8
Παράρτημα Ε.....	Ε-10
Παράρτημα ΣΤ.....	ΣΤ-15

ΚΕΦΑΛΑΙΟ 1

Εισαγωγή

1.1 Γενικά	1
1.2 Υποκίνηση Έρευνας	2
1.3 Σκιαγράφηση Έρευνας	3

1.1 Γενικά

Στην σημερινή κοινωνία της Πληροφορίας τα δίκτυα έχουν εξαπλωθεί σε μεγάλο βαθμό σε όλους τους τομείς της ζωής μας. Έχουν εγκατασταθεί σε σχολεία, νοσοκομεία, εταιρείες και οπουδήποτε μπορεί κάποιος να τα χρησιμοποιήσει για την μεταφορά της πληροφορίας. Τα τελευταία χρόνια με την αναγνώριση των δυνατοτήτων που παρέχουν τα δίκτυα έχει δημιουργηθεί μια νέα απαίτηση, αυτή της μεταφοράς πολυμεσικών αρχείων, δηλαδή η δυνατότητα μεταφοράς εικόνας και ήχου με χρήση των δικτύων.

Η δημιουργία νέων εφαρμογών όπως η χρήση τηλεφώνου μέσω ηλεκτρονικού υπολογιστή, η τηλεδιάσκεψη και γενικά εφαρμογές που στέλνουν ήχο και εικόνα στους παραλήπτες θα ήθελαν ένα πρωτόκολλο το οποίο να είναι κατάλληλο για την μεταφορά τέτοιων δεδομένων από το διαδίκτυο.

Οι εφαρμογές αυτές έχουν αρκετές διαφορές σε σχέση με τις εφαρμογές που μεταφέρουν απλά αρχεία δεδομένων. Στις εφαρμογές μεταφοράς αρχείων υπάρχουν μηχανισμοί τέτοιοι ώστε να εγγυώνται τη σίγουρη μεταφορά των δεδομένων. Δηλαδή εάν για οποιοδήποτε λόγο ένα κομμάτι ενός αρχείου χαθεί, τότε ο μηχανισμός αυτός θα υποχρεωθεί να ξαναστείλει το δεδομένο αυτό. Με τον τρόπο αυτό είμαστε σίγουροι πως

θα παραλάβουμε όλα τα δεδομένα τα οποία έχει στείλει ο παραλήπτης. Όπως γίνεται κατανοητό ο παράγοντας χρόνος δεν υπάρχει σε αυτές τις εφαρμογές. Είμαστε σίγουροι πως θα παραλάβουμε τα πακέτα μας αλλά δεν γνωρίζουμε το πότε.

Στις πολυμεσικές εφαρμογές τα πράγματα είναι τελείως διαφορετικά. Ο χρόνος μεταφοράς των δεδομένων είναι πολύ σημαντικός και οι εφαρμογές αυτές μπορούν να ανεχθούν ένα ποσοστό χαμένων πακέτων χωρίς τα πακέτα αυτά να παίζουν σημαντικό ρόλο στην απόδοση της εφαρμογής.

Ας πάρουμε για παράδειγμα μια πολυμεσική εφαρμογή η οποία μεταδίδει απευθείας ένα ποδοσφαιρικό αγώνα. Θα ήταν χωρίς νόημα να περιμέναμε με καθυστέρηση να ακούσουμε την περιγραφή γιατί τότε θα χάναμε την έννοια του 'απευθείας'. Εάν από την άλλη χάνονταν κάποια δεδομένα θα είχαμε κάποια αλλοίωση του ήχου αλλά ο ανθρώπινος εγκέφαλος είτε δεν θα καταλάβαινε τη διαφορά είτε θα συμπλήρωνε το περιεχόμενο των χαμένων δεδομένων.

1.2 Υποκίνηση Έρευνας

Τι γίνεται όμως εάν ο αξιόπιστος τρόπος μεταφοράς δεδομένων υπάρχει παράλληλα σε ένα δίκτυο με τον τρόπο μεταφοράς πολυμεσικών αρχείων; Μέχρι σήμερα δεν έχουν γίνει αρκετές προσομοιώσεις ώστε να μπορούμε να εξάγουμε κάποια ασφαλή συμπεράσματα. Σκοπός αυτής της διπλωματικής εργασίας είναι η σύγκριση των δύο αυτών τρόπων μεταφοράς δεδομένων.

Μέχρι σήμερα έχουν γίνει πολλές προσπάθειες για τη δημιουργία νέων πρωτοκόλλων τα οποία θα έχουν ως στόχο τη μεταφορά πολυμεσικών αρχείων. Οι προσομοιώσεις των πρωτοκόλλων αυτών είναι πολύ σημαντικές ώστε να κατανοήσουμε τυχόν λάθη στους μηχανισμούς ελέγχου των πρωτοκόλλων και να δούμε εάν τα πρωτόκολλα αυτά θα ικανοποιούν τον σκοπό για τον οποίο έχουν δημιουργηθεί ή θα πρέπει να σχεδιαστούν και πάλι από την αρχή.

Ο στόχος αυτής της Ατομικής Διπλωματικής Εργασίας είναι τριπλός. Πρώτος στόχος είναι η παρουσίαση των πρωτοκόλλων για μεταφορά δεδομένων που προέρχονται από εφαρμογές πολυμέσων. Δεύτερος στόχος είναι να απαντήσουμε στο ερώτημα, μέσω των προσομοιώσεων, κατά πόσο τα πρωτόκολλα αυτά εξυπηρετούν τον σκοπό για τον οποίο έχουν δημιουργηθεί. Τέλος έχουμε ως στόχο την παρουσίαση και ανάλυση των αποτελεσμάτων από τις προσομοιώσεις βάση του μηχανισμού ελέγχου συμμόρφωσης των πρωτοκόλλων όπου αυτός είναι υπαρκτός.

1.3 Σκιαγράφιση Έρευνας

Η διπλωματική εργασία ξεκινά με μια περιγραφή των πρωτοκόλλων τα οποία θα χρησιμοποιήσουμε στις προσομοιώσεις. Ακολούθως παρουσιάζονται οι προσομοιώσεις με τη χρήση πινάκων και γραφικών παραστάσεων για τις μετρήσεις τις οποίες έχουμε κάνει. Τέλος υπάρχουν τα συμπεράσματα τα οποία έχουμε εξάξει και τεχνικές οι οποίες μπορούν να χρησιμοποιηθούν στο μέλλον μαζί με τα πρωτόκολλα τα οποία έχουμε αναλύσει. Συγκεκριμένα η δομή της Ατομικής Διπλωματικής Εργασίας έχει ως ακολούθως :

- Κεφάλαιο 1 : Εισαγωγή
- Κεφάλαιο 2 : Περιγραφή Πρωτοκόλλων
- Κεφάλαιο 3 : Περιγραφή και Παρουσίαση Προσομοιώσεων
- Κεφάλαιο 4 : Ανάλυση Αποτελεσμάτων
- Κεφάλαιο 5 : Νέες Τεχνολογίες Πρωτοκόλλων
- Κεφάλαιο 6 : Συμπεράσματα

Στο τέλος της εργασίας υπάρχουν η βιβλιογραφία και ενδεικτικός κώδικας ο οποίος έχει χρησιμοποιηθεί κατά την διάρκεια των προσομοιώσεων.

ΚΕΦΑΛΑΙΟ 2

Περιγραφή Πρωτοκόλλων Μεταφοράς Δεδομένων

2.1 Transport Control Protocol (TCP)	4
2.2 User Datagram Protocol (UDP)	8
2.3 TCP Friendly Rate Control (TFRC)	11
2.4 Streaming Control Transport Protocol (SCTP)	23
2.5 Real Time Protocol (RTP)	32

2.1 Transport Control Protocol (TCP)

Το TCP [1] είναι το αξιόπιστο πρωτόκολλο μεταφοράς δεδομένων στο σημερινό διαδίκτυο. Για να εξασφαλίσει την αξιόπιστη μεταφορά δεδομένων το TCP βασίζεται σε διάφορες αρχές όπως την ανίχνευση λάθους, τις επαναληπτικές μεταδόσεις χαμένων πακέτων τις συνεχόμενες αναγνωρίσεις πακέτων, τα χρονόμετρα καθώς και τις επικεφαλίδες στα επιπρόσθετα πεδία των πακέτων.

Το TCP λέγεται ότι είναι στηριζόμενο σε σύνδεση γιατί μια εφαρμογή πριν ξεκινήσει να στέλνει πακέτα δεδομένων, πρέπει να έρθει σε συμφωνία με τον παραλήπτη, δηλαδή να στείλει προκαταρκτικά πακέτα ελέγχου τα οποία θα καθορίζουν κάποιες παραμέτρους του πρωτοκόλλου τα οποία θα εξασφαλίζουν την αξιόπιστη μεταφορά των δεδομένων. Το TCP είναι κατάλληλο για μεταφορά δεδομένων και προς τις δύο κατευθύνσεις. Επίσης η μεταφορά δεδομένων γίνεται μόνο μεταξύ δύο ακραίων κόμβων και όχι περισσοτέρων. Επίσης οι ενδιάμεσοι κόμβοι, δηλαδή οι δρομολογητές δεν γνωρίζουν καμιά πληροφορία για τις συνδέσεις αυτές.

Το πακέτο του TCP αποτελείται από δύο κύρια κομμάτια. Το πρώτο κομμάτι που συνήθως έχει μέγεθος 20 ψηφιολέξεις αποτελεί το μέρος των πεδίων της επικεφαλίδας. Τα κυριότερα από αυτά τα πεδία είναι η πηγή και ο προορισμός του πακέτου, ο μοναδικός κωδικός αριθμός του πακέτου και το μέγεθος του παραθύρου στον παραλήπτη. Το πεδίο αυτό χρησιμεύει για τον έλεγχο ροής δεδομένων, ώστε ο ρυθμός αποστολής των πακέτων να μειώνεται σε περίπτωση συμφόρησης στο δίκτυο. Υπάρχουν και κάποια άλλα πεδία τα οποία δεν χρησιμεύουν πάντα όπως το πεδίο της χρονοσφραγίδας και του μέγιστου μεγέθους δεδομένων. Το δεύτερο κομμάτι αποτελεί το χώρο στον οποίο θα τοποθετηθούν τα δεδομένα της εφαρμογής.

Όπως έχουμε αναφέρει το πρωτόκολλο αυτό είναι αξιόπιστο. Δηλαδή σε περίπτωση που έχουν χαθεί κάποια πακέτα, το πρωτόκολλο είναι αναγκασμένο να τα ξαναστείλει. Για να γνωρίζει πότε ένα πακέτο μπορεί να έχει χαθεί το πρωτόκολλο διατηρεί ένα χρονόμετρο του χρόνου μεταφοράς (round trip time RTT), δηλαδή του χρόνου που ένα πακέτο κάνει να φτάσει στον παραλήπτη και να έρθει πίσω η επιβεβαίωση στον αποστολέα. Εάν αυτό το χρονόμετρο λήξει, τότε υπάρχει η υπόνοια πως το πακέτο έχει χαθεί και γι' αυτό στέλνεται ξανά. Όποτε έχουμε επαναποστολή λόγω λήξης του χρονόμετρου, αρχικοποιούμε το χρονόμετρο αυτό στον διπλάσιο από τον προηγούμενο χρόνο.

Παρατηρούμε πως με την αύξηση του χρόνου στο χρονόμετρο, υπάρχει η πιθανότητα να χαθεί ένα πακέτο αλλά να περάσει αρκετή ώρα πριν αυτό εντοπιστεί. Για το σκοπό αυτό γίνεται χρήση του πεδίου του μοναδικού αύξοντα αριθμού στα πακέτα. Κάθε πακέτο που στέλνεται έχει ένα κωδικό αύξοντα αριθμό ο οποίος αυξάνεται κατά ένα με κάθε αποστολή πακέτων. Μόλις ένα πακέτο φτάσει στον προορισμό του, ο παραλήπτης φτιάχνει μια επιβεβαίωση παραλαβής την οποία και αποστέλλει πίσω στον αποδέκτη όπου πάνω υπάρχει ο μοναδικός κωδικός του πακέτου.

Εάν ένα πακέτο φτάσει στον παραλήπτη με ένα κωδικό αριθμό μεγαλύτερο από αυτό που θα περίμενε τότε αντί να στείλει επιβεβαίωση με τον αύξοντα αριθμό του πακέτου αυτού, στέλνει επιβεβαίωση με τον αύξοντα αριθμό του τελευταίου πακέτου που έφτασε στην σειρά. Αποτέλεσμα αυτής της διαδικασίας είναι, εάν ένα πακέτο έχει χαθεί, ο παραλήπτης να στέλνει επιβεβαιώσεις του τελευταίου πακέτου που έφτασε σε

σειρά. Εάν ο αποστολέας πάρει τρεις διαδοχικές επιβεβαιώσεις του ιδίου πακέτου, τότε αναγνωρίζει πως το πακέτο έχει χαθεί και το ξαναστέλνει ακόμα κι αν το χρονόμετρο του χρόνου μεταφοράς δεν έχει λήξει. Η διαδικασία αυτή ονομάζεται γρήγορη επανεκκίνηση και είναι ένα από τα πλεονεκτήματα του πρωτοκόλλου αυτού.

Ο παραλήπτης κάθε σύνδεσης έχει ένα προσωρινό χώρο αποθήκευσης (buffer) στον οποίο αποθηκεύει τα πακέτα τα οποία έχουν φτάσει και βρίσκονται στη σωστή σειρά. Η εφαρμογή που σχετίζεται με την σύνδεση αυτή διαβάζει από αυτό τον αποθηκευτικό χώρο όποτε το θεωρήσει αυτή αναγκαίο και όχι με ένα σταθερό ρυθμό. Εάν ο ρυθμός αυτός είναι σχετικά αργός τότε ο αποθηκευτικός χώρος θα γεμίσει σχετικά γρήγορα και τα πακέτα τα οποία θα φτάνουν από εκείνη την στιγμή και μετά θα ρίχνονται αναγκαστικά από τον παραλήπτη.

Το TCP για να αποφύγει αυτό το πρόβλημα παρέχει ένα μηχανισμό ελέγχου ροής έτσι ώστε να εμποδίζει τον αποστολέα να στέλνει δεδομένα πιο γρήγορα από τον ρυθμό που ο παραλήπτης μπορεί να τα παραλαμβάνει [17]. Για τον σκοπό αυτό ο παραλήπτης διατηρεί ένα παράθυρο το οποίο αντιπροσωπεύει τον ελεύθερο αποθηκευτικό χώρο που υπάρχει. Το μέγεθος αυτό στέλνεται στον αποστολέα με κάθε επιβεβαίωση που στέλνεται για πακέτα που έχουν παραληφθεί. Ο αποστολέας χρησιμοποιεί το δεδομένο αυτό ως εξής : Ο αποστολέας γνωρίζει το μέγεθος των δεδομένων που έχει στείλει στον παραλήπτη καθώς και το ποσοστό των δεδομένων αυτών που έχουν επιβεβαιωθεί ότι έφτασαν κανονικά στον παραλήπτη. Ο αποστολέας τότε γνωρίζει πως τα νέα δεδομένα που θα στείλει προστιθέμενα στα δεδομένα που δεν έχουν ακόμα επιβεβαιωθεί, δεν πρέπει να ξεπερνούν συνολικά τον ελεύθερο διαθέσιμο αποθηκευτικό χώρο του παραλήπτη.

Ο αλγόριθμος συμφόρησης του TCP χωρίζεται συνολικά σε τρεις φάσεις : Στην φάση της γραμμικής αύξησης και εκθετικής μείωσης , στην φάση της αργής εκκίνησης και στην φάση της αντίδρασης από γεγονότα λήξης χρόνου. Όπως έχουμε αναφέρει το πρωτόκολλο αντιδρά στην ύπαρξη συμφόρησης με το να μειώνει το παράθυρο του ελεύθερου αποθηκευτικού χώρου. Πιο συγκεκριμένα εάν το πρωτόκολλο στέλνει πακέτα με ένα ρυθμό αποστολής X και παρατηρηθεί συμφόρηση, τότε ο νέος ρυθμός αποστολής θα μοιραστεί και θα γίνει $X / 2$. Ο ρυθμός αποστολής θα συνεχίσει να

μοιράζεται εφόσον παρατηρείται συμφόρηση και την επόμενη φορά που ο αποστολέας θα στείλει πακέτα. Το ελάχιστο που μπορεί να στείλει ένα αποστολέας μετά από διαδοχική ανίχνευση συμφόρησης είναι ένα πακέτο σε κάθε αποστολή (ένα πακέτο ανά RTT).

Επίσης εάν μετά από διαδοχικές αποστολές στις οποίες υπήρχε συμφόρηση ο αποστολέας αναγνωρίσει την μη ύπαρξη συμφόρησης, τότε δεν συνεχίζει να στέλνει με τον μέγιστο επιτρεπτό ρυθμό αλλά στέλνει ένα πακέτο περισσότερο από τις προηγούμενες φορές για κάθε χρόνο που δεν παρατηρείται συμφόρηση. Αυτή η διαδικασία ονομάζεται αποφυγή συμφόρησης (congestion avoidance) και είναι χρήσιμη για να μην παρατηρούνται πολλές συνεχόμενες συμφορήσεις. Αν δεν υπήρχε ο μηχανισμός αυτός τότε θα παρατηρούσαμε μια συνεχή αυξομείωση στον ρυθμό αποστολής των πακέτων.

Ένα άλλο κομμάτι του μηχανισμού ελέγχου συμφόρησης είναι η γρήγορη εκκίνηση. Με την αρχικοποίηση της σύνδεσης ο ρυθμός αποστολής ορίζεται στο ένα πακέτο ανά χρόνο αποστολής. Επειδή ο ρυθμός αυτός είναι σχετικά μικρός σε σχέση με το διαθέσιμο εύρος ζώνης της σύνδεσης, θα ήταν κρίμα κάθε φορά να αυξάνουμε τον ρυθμό αυτό κατά ένα. Αντίθετα διπλασιάζουμε τον ρυθμό αποστολής κάθε φορά που ο παραλήπτης επιβεβαιώνει ότι έχει παραλάβει τα πακέτα που έχουμε στείλει. Ο ρυθμός αυτός συνεχίζει να διπλασιάζεται μέχρι να φτάσουμε σε ένα σημείο στο οποίο ο αποστολέας θα καταλάβει πως υπάρχει πιθανή συμφόρηση στο δίκτυο. Η διαδικασία αυτή γίνεται μόνο στην έναρξη της σύνδεσης ενώ στις υπόλοιπες περιπτώσεις όπου δεν υπάρχει συμφόρηση, ο ρυθμός αυξάνεται γραμμικά όπως έχουμε περιγράψει πιο πάνω.

Όπως έχουμε αναφέρει, ο αποστολέας λαμβάνει την ένδειξη για συμφόρηση, είτε μέσω τριπλών επιβεβαιώσεων είτε με την λήξη του χρονομέτρου για την παραλαβή επιβεβαίωσης. Το πρωτόκολλο έχει την δυνατότητα να αντιδράσει διαφορετικά σε αυτές τις περιπτώσεις. Εάν έχουμε παραλάβει τριπλή επιβεβαίωση σημαίνει πως ο παραλήπτης έχει την δυνατότητα να λαμβάνει πακέτα ενώ σε περίπτωση που δεν παίρνουμε επιβεβαιώσεις δεν μπορούμε να συμπεράνουμε κάτι τέτοιο. Επομένως σε περίπτωση τριπλής επιβεβαίωσης ο ρυθμός αποστολής μειώνεται στην μέση και συνεχίζουμε με την φάση της αποφυγής συμφόρησης.

Σε περίπτωση όμως που έχουμε λήξη του χρονομέτρου, αντί να μειώσουμε τον ρυθμό αποστολής στην μέση, θέτουμε τον ρυθμό αποστολής στο ένα πακέτο ανά χρόνο αποστολής. Όμως αντί να μπαίνουμε στην φάση της αποφυγής συμφόρησης, μπαίνουμε στην φάση την γρήγορης εκκίνησης και αυξάνουμε τον ρυθμό αποστολής εκθετικά μέχρι ενός σημείου που ονομάζουμε κατώφλι.

Το κατώφλι αυτό είναι αρχικά ίσο με το μισό του παραθύρου του παραλήπτη και σε κάθε εμφάνιση συμφόρησης το κατώφλι αυτό μοιράζεται στην μέση. Εάν ο ρυθμός αποστολής φτάσει το κατώφλι τότε μπαίνουμε στην φάση της αποφυγής συμφόρησης και ο ρυθμός αποστολής αυξάνεται γραμμικά.

Το TCP είναι κατασκευασμένο για να παρέχει δικαιοσύνη (fairness) ανάμεσα στις ροές του ίδιου τύπου. Δηλαδή εάν έχουμε ροές TCP που περνούν από το ίδιο μονοπάτι που έχει εύρος X και έχουμε Ψ ροές, τότε ο μέσος ρυθμός αποστολής δεδομένων για κάθε ροή είναι X/Ψ .

Το πρωτόκολλο TCP χρησιμοποιείται στο σημερινό διαδύκτιο γιατί παρέχει αξιόπιστη μεταφορά δεδομένων. Υπάρχουν πολλοί παράμετροι που ακόμα δεν έχει αξιοποιήσει αλλά το πρωτόκολλο αυτό συνεχίζει να εξελίσσεται ώστε να ανταποκρίνεται στις ανάγκες της σημερινής τεχνολογίας. Ένα από τα μεγαλύτερα προβλήματα που αντιμετωπίζει είναι ο συναγωνισμός με ροές διαφορετικού τύπου όπως το UDP που μεταφέρει πολυμεσικά πακέτα στα οποία δεν υπάρχει έλεγχος συμφόρησης.

2.2 User Datagram Protocol (UDP)

Το UDP [1] είναι ένα πρωτόκολλο μεταφοράς δεδομένων το οποίο κάνει όσο το δυνατό λιγότερα πράγματα μπορεί ένα πρωτόκολλο να κάνει. Εκτός από την πολυπλεξία και την αποπολυπλεξία και κάποιο μικρό έλεγχο για σφάλματα δεν κάνει τίποτα άλλο. Με τον όρο πολυπλεξία εννοούμε την δυνατότητα του πρωτοκόλλου στον αποστολέα, να βάζει κομμάτια δεδομένων που προέρχονται από διαφορετικές εφαρμογές στο ίδιο πακέτο και αντίστοιχα με τον όρο αποπολυπλεξία εννοούμε την δυνατότητα του

παραλήπτη να αναγνωρίζει σε ποια εφαρμογή θα προωθηθούν τα κομμάτια δεδομένων που βρίσκονται στα πακέτα.

Το πρωτόκολλο αυτό δεν προσφέρει ουσιαστικά τίποτα στο IP. Συγκεκριμένα εάν ένας προγραμματιστής επιλέξει να χρησιμοποιήσει UDP αντί TCP, τότε η εφαρμογή ουσιαστικά είναι σαν να επικοινωνεί απευθείας με το IP αγνοώντας το επίπεδο μεταφοράς δεδομένων.

Το πρωτόκολλο παίρνει τα μηνύματα από την εφαρμογή, προσθέσει τον αριθμό καναλιού της πηγής καθώς και τον αριθμό καναλιού του προορισμού, προσθέτει άλλα δύο μικρά πεδία και στέλνει το πακέτο στο επίπεδο δικτύου. Το IP προσθέτει αυτό το πακέτο σε ένα διάγραμμα IP και κάνει μια προσπάθεια να στείλει το τελικό πακέτο στον αποδέκτη.

Εάν το πακέτο φτάσει τελικά στον προορισμό του, το UDP κοιτάζει τον αριθμό καναλιού του αποδέκτη ώστε να στείλει το πακέτο στην σωστή εφαρμογή. Σημειώνουμε πως το UDP δεν καθιστά σύνδεση από τον αποστολέα στον παραλήπτη όπως κάνει το TCP πριν να αρχίσει να στέλνει πακέτα.

Το ερώτημα είναι γιατί κάποιος να χρησιμοποιήσει UDP που δεν είναι αξιόπιστο σε αντίθεση με την χρήση του TCP. Η απάντηση είναι ότι πολλές εφαρμογές είναι καλύτερα να χρησιμοποιήσουν το UDP για τους εξής λόγους :

- Δεν υπάρχει αρχική σύνδεση : Σε αντίθεση με το TCP που αρχικά κάνει σύνδεση τριών βημάτων (όπως είδαμε στο προηγούμενο τμήμα) το UDP απλά ξεκινά να στέλνει τα πακέτα χωρίς οποιαδήποτε προειδοποίηση. Αυτός είναι ο λόγος που δεν υπάρχει και καθυστέρηση κατά την δημιουργία μιας σύνδεσης.
- Δεν κρατά πληροφορίες σύνδεσης : Το TCP κρατά πληροφορίες για την σύνδεση στα άκρα. Αυτές οι πληροφορίες περιλαμβάνουν τα μεγέθη των αποθηκευτικών χώρων του αποστολέα και του παραλήπτη, παραμέτρους για τον έλεγχο συμφόρησης καθώς και παραμέτρους για την σειρά αποστολής των πακέτων. Από την άλλη μεριά το UDP δεν κρατά τέτοιες πληροφορίες. Για αυτό το λόγο

ένας εξυπηρετητής που είναι αφοσιωμένος σε μια εφαρμογή μπορεί να υποστηρίξει περισσότερους τελικούς χρήστες εάν η εφαρμογή τρέχει σε UDP παρά σε TCP.

- Μικρότερες Επικεφαλίδες Πακέτων: Σε αντίθεση με το TCP που έχει κόστος επικεφαλίδων 20 ψηφιολέξεις, το UDP έχει κόστος μόνο 8 ψηφιολέξεων.
- Εκλεπτυσμένος έλεγχος στην εφαρμογή για το τι δεδομένα στέλνονται και πότε: Εάν κάνουμε χρήση του UDP τότε μόλις το πρωτόκολλο πάρει κάποια δεδομένα από την εφαρμογή δημιουργεί τα πακέτα και τα στέλνει στο επίπεδο δικτύου. Από την άλλη το TCP έχει μηχανισμό ελέγχου συμφόρησης, ο οποίος καθυστερεί την αποστολή πακέτων μόλις διαπιστώσει ότι υπάρχει μεγάλη συμφόρηση σε ένα ή περισσότερα μονοπάτια αποστολής των πακέτων. Επίσης το TCP συνεχίζει να στέλνει πακέτα εφόσον παίρνει επιβεβαιώσεις από τον παραλήπτη, ασχέτως του πόσο χρόνο χρειάζονται οι επιβεβαιώσεις αυτές για να φτάσουν στον αποστολέα.

Πολλές πολυμεσικές εφαρμογές συχνά χρειάζονται ένα ελάχιστο ρυθμό αποστολής των δεδομένων επομένως δεν θέλουν να υπάρχει καθυστέρηση στην αποστολή των πακέτων αυτών και μπορούν να ανεχθούν ένα ποσοστό απώλειας των πακέτων αυτών. Το TCP δεν μπορεί να λειτουργήσει με αυτές τις προϋποθέσεις γι' αυτό και οι εφαρμογές αυτές τρέχουν με UDP.

Στο σημερινό διαδίκτυο το UDP χρησιμοποιείται με πολυμεσικές εφαρμογές όπως το διαδικτυακό τηλέφωνο, τηλεδιάσκεψη σε πραγματικό χρόνο και μετάδοση αποθηκευμένου ήχου και εικόνας [20]. Όλες αυτές οι εφαρμογές μπορούν να ανεχθούν ένα ποσοστό απώλειας πακέτων, επομένως η αξιόπιστη μεταφορά δεδομένων δεν είναι αναγκαία για την επιτυχία της εφαρμογής. Επίσης οι εφαρμογές πραγματικού χρόνου αντιδρούν πολύ απογοητευτικά στο μηχανισμό ελέγχου συμφόρησης του TCP. Γι' αυτούς τους λόγους οι εταιρείες που δημιουργούν πολυμεσικές εφαρμογές συχνά επιλέγουν να τρέχουν τις εφαρμογές τους σε UDP παρά σε TCP.

Σήμερα οι πολυμεσικές εφαρμογές χρησιμοποιούν το UDP για να στέλνουν τα πακέτα τους. Το UDP όπως προαναφέραμε δεν έχει μηχανισμό ελέγχου συμφόρησης. Ο μηχανισμός αυτός όμως είναι αναγκαίος ώστε να εμποδίσει το δίκτυο να εισέλθει σε μια κατάσταση όπου η χρήσιμη δουλειά που θα γίνεται θα είναι ελάχιστη. Εάν ο καθένας ξεκινούσε να στέλνει βίντεο με ψηλό ρυθμό χωρίς να χρησιμοποιεί μηχανισμό ελέγχου συμφόρησης, θα υπήρχε τόσο μεγάλη συμφόρηση στους δρομολογητές ώστε κανένας δεν θα μπορούσε να δει οτιδήποτε.

Επομένως αυτή η έλλειψη μηχανισμού στο UDP είναι ένα σοβαρό πρόβλημα. Πολλοί ερευνητές έχουν εισηγηθεί νέους μηχανισμούς ώστε να αναγκάζουν όλες τις ροές να χρησιμοποιούν κάποιου είδους μηχανισμό ελέγχου συμφόρησης. Επίσης έχουν υλοποιήσει νέα πρωτόκολλα όπως το TFRC τα οποία έχουν σκοπό την μεταφορά πολυμεσικών πακέτων στα οποία γίνεται και κάποιας μορφής έλεγχος συμφόρησης.

Η δομή του πακέτου του UDP αποτελείται από δύο κύρια μέρη. Από το μέρος των δεδομένων της εφαρμογής και από το μέρος των πεδίων της επικεφαλίδας. Στο μέρος των δεδομένων της εφαρμογής τοποθετούνται τα δεδομένα που η εφαρμογή στέλνει στο επίπεδο της μεταφοράς δεδομένων. Το μέρος της επικεφαλίδας αποτελείται από τέσσερα πεδία που το καθένα έχει μέγεθος δύο ψηφιολέξεις. Τα δύο πρώτα πεδία είναι ο αριθμός καναλιού της εφαρμογής από την οποία έρχονται τα δεδομένα και ο αριθμός καναλιού εφαρμογής για την οποία προορίζονται τα δεδομένα. Το τρίτο πεδίο είναι για να ορίζεται το μέγεθος του πακέτου UDP, μαζί με την επικεφαλίδα, σε ψηφιολέξεις. Το τέταρτο και τελευταίο πεδίο είναι για τον υπολογισμό, εάν υπάρχουν οποιαδήποτε λάθη στα δεδομένα που έχουν εισαχθεί στο πακέτο που έχει σταλεί στον παραλήπτη.

2.3 TCP Friendly Rate Control (TFRC)

Ο έλεγχος ροής TCP-Friendly (TFRC) είναι ένας μηχανισμός ελέγχου συμφόρησης για τις ροές προς μια κατεύθυνση (unicast) που λειτουργούν σε ένα περιβάλλον διαδικτύου καλύτερης προσπάθειας.. Το TFRC [3] προορίστηκε για τις εφαρμογές που χρησιμοποιούν ένα σταθερό μέγεθος πακέτων, και είχε ως σκοπό να είναι εύλογα δίκαιο, ως προς τον ανταγωνισμό για το εύρος ζώνης, με τις συνδέσεις TCP που

χρησιμοποιούν το ίδιο μέγεθος πακέτων. Η παραλλαγή VoIP TFRC σχεδιάζεται για τις εφαρμογές που στέλνουν μικρά πακέτα, όπου ο στόχος των σχεδιαστών είναι να επιτευχθεί το ίδιο εύρος ζώνης σε δυαδικά ψηφία ανά δευτερόλεπτο (bps) με μια ροή TCP που χρησιμοποιεί πακέτα δεδομένων 1500-ψηφιολέξεων (bytes). Η παραλλαγή VoIP TFRC επιβάλλει ένα ελάχιστο διάστημα 10 ms μεταξύ των πακέτων δεδομένων, για να αποτρέψει μια ροή από να στείλει μικρά πακέτα αυθαίρετα συχνά. Εισάγουμε επίσης το γρηγορότερο ξεκίνημα (Faster Restart), ένα προαιρετικό μηχανισμό που βελτιστοποιεί τη συμπεριφορά των ροών που χρησιμοποιούν TFRC. Το γρηγορότερο ξεκίνημα προτείνεται για χρήση και με την προεπιλογή TFRC και με την παραλλαγή VoIP TFRC.

Το TFRC έχει ως σκοπό να είναι εύλογα δίκαιο ως προς τον ανταγωνισμό για το εύρος ζώνης με τις ροές TCP, αλλά αποφεύγει τις απότομες αλλαγές στο ρυθμό αποστολής που είναι χαρακτηριστικό του μηχανισμού ελέγχου συμφόρησης του TCP. Το TFRC προορίζεται για εφαρμογές όπως οι εφαρμογές μέσων ροής (media streams) όπου ένα σχετικά ομαλό ποσοστό αποστολής είναι σημαντικό. Η παραλλαγή VoIP προορίζεται για τις ροές που πρέπει να στέλνουν συχνά μικρά πακέτα. Το συμβατικό TFRC μετρά τα ποσοστά απώλειας με τον υπολογισμό της αναλογίας γεγονότος απώλειας (loss event ratio) χωρίς να λαμβάνει υπόψη το μέγεθος των πακέτων [15]. Αυτό έχει συνέπειες ως προς το ποσοστό που μια ροή TFRC μπορεί να επιτύχει, κατά το συναγωνισμό με τις ροές TCP μεγάλων πακέτων που μοιράζονται μια γραμμή με συμφόρηση. Ειδικότερα, μια TFRC ροή με χαμηλό εύρος ζώνης και μικρά πακέτα που μοιράζεται μια γραμμή με συμφόρηση με μια ροή TCP που έχει υψηλό εύρος ζώνης και μεγάλα πακέτα μπορεί να αναγκαστεί να επιβραδύνει, ακόμα κι αν ο μέσος ρυθμός αποστολής της εφαρμογής σε ψηφιολέξεις ανά δευτερόλεπτο, είναι λιγότερος από το ποσοστό που επιτυγχάνεται από την ροή TCP. Διαισθητικά, αυτό θα ήταν "δίκαιο" μόνο εάν ο περιορισμός δικτύων ήταν στα πακέτα ανά δευτερόλεπτο παρά σε ψηφιολέξεις ανά δευτερόλεπτο. Πολλοί από τους περιορισμούς δικτύων στο σημερινό διαδίκτυο είναι σε ψηφιολέξεις ανά δευτερόλεπτο, αλλά οι περιορισμοί δικτύων του μέλλοντος μπορεί να κινηθούν προς τους περιορισμούς σε πακέτα ανά δευτερόλεπτο. Η παραλλαγή VoIP TFRC θα υποστηρίξει καλύτερα τις εφαρμογές που δεν θέλουν τα ποσοστά αποστολής τους σε ψηφιολέξεις ανά δευτερόλεπτο να πάσχουν από τη χρήση μικρών πακέτων. Αυτή η παραλλαγή είναι περιορισμένη στις εφαρμογές που στέλνουν πακέτα λιγότερο από μία

φορά κάθε 10 ms (που ορίζεται ως το ελάχιστο διάστημα αποστολής). Λαμβάνοντας υπόψη αυτόν τον περιορισμό, η παραλλαγή VoIP υπολογίζει το δίκαιο ρυθμό αποστολής του TFRC, υποθέτοντας ότι περιορισμοί ήταν σε ψηφιολέξεις ανά δευτερόλεπτο. Οι εφαρμογές που χρησιμοποιούν την παραλλαγή VoIP TFRC θα μπορούσαν να έχουν ένα σταθερό μέγεθος πακέτων, ή θα μπορούσαν να ποικίλουν το μέγεθος πακέτων τους σε σχέση με την συμφόρηση. Η παραλλαγή VoIP TFRC υποστηρίζει ότι είναι αποδεκτό για τις ροές VoIP να υποθέτουν ότι ο περιορισμός δικτύων είναι σε ψηφιολέξεις ανά δευτερόλεπτο (bps) παρά σε πακέτα ανά δευτερόλεπτο (pps). Ο ρυθμός αποστολής των πακέτων VoIP σε περίπτωση συμφόρησης θα είναι ο ίδιος με το ρυθμό αποστολής των πακέτων TCP με μέγεθος 1500 ψηφιολέξεις.

Το TFRC χρησιμοποιεί την εξίσωση ρυθμοαπόδοσης [5] (throughput equation) του TCP, η οποία δίνει το ποσοστό X της αποστολής σε ψηφιολέξεις ανά δευτερόλεπτο ως λειτουργία του ποσοστού γεγονότος απώλειας, του μεγέθους πακέτων, και του μετ' επιστροφής χρόνου (round trip time). Το μέγεθος s των πακέτων που χρησιμοποιείται στην εξίσωση ρυθμοαπόδοσης, είναι το μέγεθος των πακέτων που χρησιμοποιείται από την εφαρμογή, ή το κατ' εκτίμηση μέσο μέγεθος πακέτων, εάν υπάρχουν παραλλαγές στο μέγεθος των πακέτων ανάλογα με τα στοιχεία. Αυτό δίνει δικαιοσύνη με τις ροές TCP που χρησιμοποιούν το ίδιο μέγεθος πακέτων. Η παραλλαγή VoIP αλλάζει αυτήν την συμπεριφορά με τους ακόλουθους τρόπους.

- Το προκαθορισμένο μέγεθος s των πακέτων τίθεται στις 1460 ψηφιολέξεις. Αυτό παρέχει ένα στόχο δικαιοσύνης, από την άποψη του ρυθμού αποστολής σε ψηφιολέξεις ανά δευτερόλεπτο, μιας ροής TCP με 1460 ψηφιολέξεις δεδομένων εφαρμογής ανά πακέτο.
- Ο επιτρεπτός ρυθμός αποστολής X της μετάδοσης σε ψηφιολέξεις ανά δευτερόλεπτο, μειώνεται από έναν παράγοντα που αναλογεί στο μέγεθος της επικεφαλίδας των πακέτων. Αυτό δίνει στην εφαρμογή κάποιο κίνητρο, πέρα από το ελάχιστο διάστημα αποστολής, για να μην στέλνει απαραίτητα μικρά πακέτα. Ειδικότερα, εισάγουμε μια νέα παράμετρο H , η οποία αντιπροσωπεύει το αναμενόμενο μέγεθος σε ψηφιολέξεις των επικεφαλίδων δικτύων και μεταφορών

που θα χρησιμοποιηθούν για τα πακέτα της σύνδεσης TFRC. Αυτό χρησιμοποιείται για να μειώσει το ρυθμό αποστολής X ως εξής:

$$X = X * s_true / (s_true + X)$$

όπου s_true είναι το πραγματικό μέσο μέγεθος πακέτων για τη σύνδεση σε ψηφιολέξεις, εξαιρώντας τις επικεφαλίδες μεταφορών και δικτύων. Η παράμετρος H τίθεται σταθερή στις 40 ψηφιολέξεις. Κατά συνέπεια, εάν η εφαρμογή VoIP TFRC χρησιμοποίησε τμήματα στοιχείων 40 ψηφιολέξεων, ο ρυθμός μετάδοσης X θα μοιραζόταν από το γεγονός ότι το μισό από το ρυθμό αποστολής θα χρησιμοποιείτο από τις επικεφαλίδες. Εντούτοις, μια σύνδεση που χρησιμοποιεί το VoIP μπορεί να χρησιμοποιήσει αντί αυτού, μια ακριβέστερη εκτίμηση του X , βασισμένη στις πραγματικές επικεφαλίδες δικτύων και μεταφορών που χρησιμοποιούνται στα πακέτα της σύνδεσης. Παραδείγματος χάριν, μια σύνδεση DCCP πάνω σε IPv4, όπου τα πακέτα στοιχείων χρησιμοποιούν τον τύπο πακέτων DCCP, και δεν υπάρχει καμία από τις επιλογές IP ή DCCP, μπορεί να θέσει το H σε $20 + 12 = 32$ ψηφιολέξεις.

- Τέλος, η παραλλαγή VoIP TFRC επιβάλλει ένα ελάχιστο διάστημα αποστολής μεταξύ των πακέτων της τάξης των 10 ms. Μια ροή που επιθυμεί να υπερβεί αυτό το ελάχιστο διάστημα πρέπει να χρησιμοποιήσει τις συμβατικές εξισώσεις TFRC, παρά την παραλλαγή VoIP.

Το έγγραφο στο οποίο ορίστηκε το TFRC κάνει την αισιόδοξη υπόθεση ότι ο περιορισμός του δικτύου σε εύρος ζώνης είναι σε ψηφιολέξεις ανά δευτερόλεπτο (bps), και όχι σε κύκλους μηχανής (CPU cycles) ή σε πακέτα ανά δευτερόλεπτο (pps). Εντούτοις, κάποια προσοχή πρέπει να δοθεί στο φορτίο σε pps καθώς επίσης και στο φορτίο σε bps. Ακόμη και εκτός από το ελάχιστο διάστημα αποστολής, η παραλλαγή VoIP TFRC δίνει στην εφαρμογή κάποιο κίνητρο για να χρησιμοποιήσει λιγότερα αλλά μεγαλύτερα πακέτα, με τη συμπερίληψη του εύρους ζώνης που χρησιμοποιείται από την επικεφαλίδα των πακέτων στο ποσοστό αποστολής. Για παράδειγμα, ένας αποστολέας που χρησιμοποιεί πακέτα 120 ψηφιολέξεων χρειάζεται ένα TCP-friendly ρυθμό αποστολής 128 Kbps για να στείλει 96 Kbps στοιχείων εφαρμογής. Αυτό είναι επειδή ο

ρυθμός αποστολής TCP-friendly μειώνεται από έναν παράγοντα του τύπου $s_true / (s_true + H) = 120/160$, για να αποτελέσει την επίδραση των επικεφαλίδων των πακέτων. Εάν ο αποστολέας μεταπηδήσει ξαφνικά σε πακέτα δεδομένων 40 ψηφιολέξεων, το ποσοστό αποστολής θα μειωνόταν σε 64 Kbps στοιχείων εφαρμογής. Επίσης τα τμήματα στοιχείων με μία ψηφιολέξη θα μειώναν το ποσοστό αποστολής σε 3.12 Kbps στοιχείων εφαρμογής. (Στην πραγματικότητα, το ελάχιστο ποσοστό αποστολής θα απέτρεπε τους αποστολείς από την επίτευξη αυτών των ποσοστών, δεδομένου ότι οι εφαρμογές που χρησιμοποιούν την παραλλαγή VoIP δεν μπορούν να στείλουν περισσότερα από 100 πακέτα ανά δευτερόλεπτο).

Η παραλλαγή VoIP υποθέτει 40 ψηφιολέξεις για το μέγεθος των επικεφαλίδων, αν και η επικεφαλίδα θα μπορούσε να είναι μεγαλύτερη (λόγω των επιλογών IP, IPv6, σηράγγων IP) ή μικρότερη (λόγω της συμπίεσης επικεφαλίδων) στο καλώδιο, επειδή η χρησιμοποίηση του ακριβούς μεγέθους επικεφαλίδων σε ψηφιολέξεις θα είχε λίγο πρόσθετο όφελος. Η χρήση της παραλλαγής VoIP μιας υποτιθέμενης επικεφαλίδας 40 ψηφιολέξεων είναι επαρκής για να πάρει μια κατά προσέγγιση εκτίμηση της ρυθμοαπόδοσης, και για να δώσει στην εφαρμογή κάποιο κίνητρο για να μην χρησιμοποιήσει απαραίτητα πολλά μικρά πακέτα. Επειδή στοχεύουμε μόνο στην δικαιοσύνη μεταξύ των ροών, και σε ένα δυνατό κίνητρο για τις εφαρμογές, η χρήση μιας επικεφαλίδας 40 ψηφιολέξεων στους υπολογισμούς του εύρους ζώνης είναι ικανοποιητική.

Ο υπολογισμός μεγέθους της επικεφαλίδας παρέχει ένα κίνητρο για τις εφαρμογές για να χρησιμοποιηθούν λιγότερα, αλλά μεγαλύτερα, πακέτα. Ένα άλλο κίνητρο είναι ότι όταν ο περιορισμός μονοπατιού είναι σε rps, η εφαρμογή που χρησιμοποιεί περισσότερα μικρά πακέτα είναι πιθανόν να λάβει περισσότερες πτώσεις πακέτων, γεγονός που οδηγεί σε μείωση του ποσοστού αποστολής αναλόγως. Δηλαδή εάν η συμφόρηση είναι υπό την μορφή rps, η ροή που στέλνει περισσότερα rps θα λάβει περισσότερες ενδείξεις συμφόρησης, και θα πρέπει να ρυθμίσει το ρυθμό αποστολής της αναλόγως. Εντούτοις, η αυξανόμενη συμφόρηση που προκαλείται με την χρήση των μικρών πακέτων σε ένα περιβάλλον που περιορίζεται από rps βιώνεται όχι μόνο από τη ροή που χρησιμοποιεί τα μικρά πακέτα, αλλά από όλη την ανταγωνιστική κυκλοφορία ροών σε εκείνη την κορεσμένη σύνδεση. Αυτά τα κίνητρα είναι επομένως

ανεπαρκή να παρέχουν την ικανοποιητική προστασία για τους περιορισμούς δικτύων σε rps.

Η παραλλαγή VoIP για TFRC [16], περιλαμβάνει ένα ελάχιστο διάστημα αποστολής της τάξης των 10 ms. Αυτό παρέχει τους πρόσθετους περιορισμούς στη χρήση πολλών μικρών πακέτων. Μια αιτιολόγηση για το ελάχιστο διάστημα αποστολής είναι η λογική ότι οι εφαρμογές που θέλουν αυτήν την περίοδο να στείλουν μικρά πακέτα είναι οι εφαρμογές VoIP, που στέλνουν το πολύ ένα πακέτο κάθε 10ms, έτσι αυτός ο περιορισμός δεν έχει επιπτώσεις στην τρέχουσα κυκλοφορία. Μια δεύτερη αιτιολόγηση είναι ότι δεν υπάρχει καμία επείγουσα ανάγκη για την κυκλοφορία καλύτερης προσπάθειας στο τρέχον διαδίκτυο να σταλούν μικρά πακέτα συχνότερα από μία φορά κάθε 10ms (παρά τη λήψη της καθυστέρησης αυτής από τον αποστολέα, για τη συγχώνευση των δύο μικρών πακέτων σε ένα μεγαλύτερο). Αυτή η καθυστέρηση των 10ms για να συγχωνεύσουμε τα μικρά πακέτα είναι πιθανό να εξουσιαστεί από τη διάδοση των δικτύων, τη μετάδοση, και τις καθυστερήσεις αναμονής στους δρομολογητές στο τρέχον διαδίκτυο.

Κατά συνέπεια, το όφελος στο χρήστη της ύπαρξης λιγότερο από 10ms μεταξύ των πακέτων αντισταθμίζεται από το όφελος στο δίκτυο να αποφύγει πολλά μικρά πακέτα. Το ελάχιστο διάστημα αναγκάζει την παραλλαγή VoIP TFRC να μην υποστηρίζει τις εφαρμογές που στέλνουν μικρά πακέτα πολύ συχνά. Για παράδειγμα ας υποθέσουμε μια ροή TFRC με ένα σταθερό μέγεθος πακέτων 100 ψηφιολέξεων, με ένα μεταβλητό ρυθμό αποστολής και ένα μονοπάτι χωρίς συμφόρηση. Όταν αυτή η ροή έστελνε το πολύ 100 rps, θα ήταν σε θέση να χρησιμοποιήσει την παραλλαγή VoIP TFRC. Εάν η ροή επιθυμούσε να αυξήσει το ποσοστό αποστολής της σε περισσότερο από 100 rps, αλλά να κρατήσει το ίδιο μέγεθος πακέτων, δεν θα ήταν σε θέση πλέον να επιτύχει αυτό με τη παραλλαγή VoIP TFRC, και πρέπει να μεταπέσει στην προεπιλογή TFRC, που λαμβάνει μια δραματική, συνεχή μείωση στο ρυθμό αποστολής της. Αυτό φαίνεται όχι μόνο αποδεκτό, αλλά επιθυμητό για το σημερινό διαδίκτυο.

Δυστυχώς δεν υπάρχει τίποτα από το να αποτρέψει τις ροές να ανοίξουν πολλαπλές συνδέσεις για να αποφύγουν τον περιορισμό των 10ms, ακριβώς όπως δεν υπάρχει τίποτα αυτήν την περίοδο για να αποτρέψει τις ροές από τη χρησιμοποίηση του UDP, ή

από το άνοιγμα των πολλαπλών παράλληλων συνδέσεων TCP, ή από τη χρησιμοποίηση του μηχανισμού ελέγχου συμφόρησής τους. Φυσικά, οι εφαρμογές ή τα middleboxes είναι επίσης ελεύθερες να περιορίσουν τον αριθμό των παράλληλων συνδέσεων TFRC που ανοίγουν στον ίδιο προορισμό σε περιόδους συμφόρησης, εάν αυτό είναι επιθυμητό. Αλλά ακόμη και χωρίς έναν μηχανισμό για να αποτρέψει τις ροές από την υπονόμηση του ελάχιστου διαστήματος αποστολής με το άνοιγμα των πολλαπλάσιων παράλληλων συνδέσεων, φαίνεται χρήσιμο να περιληφθεί το ελάχιστο διάστημα αποστολής στην παραλλαγή VoIP TFRC.

Σε οποιοδήποτε RTT, μια ροή TFRC δεν μπορεί να στείλει περισσότερο από δύο φορές το X_{recv} , το ποσό που παραλήφθηκε στο προηγούμενο RTT. Το nofeedback χρονόμετρο του TFRC μειώνει αυτόν τον αριθμό στο μισό κατά τη διάρκεια που κάθε χρονόμετρο nofeedback λήξει (τουλάχιστον τέσσερα RTT, στα οποία καμία επιβεβαίωση αποστολής δεν παραλαμβάνεται). Η επίδραση αυτού, είναι ότι οι εφαρμογές πρέπει να επιβραδύνουν την έναρξη (slow start) μετά που θα παραμείνουν αδρανείς για οποιοδήποτε σημαντικό χρονικό διάστημα, ελλείψει των μηχανισμών όπως η γρήγορος έναρξη (Quick start). Αυτή η συμπεριφορά είναι ασφαλής για την κυκλοφορία καλύτερης προσπάθειας στο δίκτυο. Μια σιωπηλή εφαρμογή (silent application) σταματά την ανατροφοδότηση για τις τρέχοντες συνθήκες των δικτύων, και δεν είναι έτσι σε θέση να στείλει σε ένα αυθαίρετο ρυθμό. Αλλά αυτή η συμπεριφορά μπορεί να βλάψει την απόδοση των διαλογικών εφαρμογών όπως η φωνή.

Οι συνδέσεις για τις διαλογικές εφαρμογές τηλεφωνίας και διασκέψεων, παραδείγματος χάριν, θα έχουν συνήθως ένα μέρος ενεργό σε κάποιο χρόνο, με την εναλλαγή μεταξύ των ενεργών και συμβαλλόμενων μερών. Μερικές από τις στρατηγικές που προτάθηκαν για την αντιμετώπιση αυτού του προβλήματος, όπως η αποστολή δεδομένων κατά τη διάρκεια των μη ενεργών περιόδων έχουν χειρότερα αποτελέσματα στο δίκτυο από το να μεταστρέφονται επάνω στον επιθυμητό ρυθμό χωρίς την αργή έναρξη. Υπάρχει αιτιολόγηση για την επιτάχυνση της αργής διαδικασίας έναρξης μετά από τις μη απασχόλησης περιόδους (σε αντιδιαστολή με την αρχή μιας σύνδεσης). Μια σύνδεση που επιτυγχάνει ένα ποσοστό αποστολής X , έχει αποδειχθεί, τουλάχιστον, ότι κάποια πορεία μεταξύ των σημείων αρχής και τέλους μπορεί να υποστηρίξει εκείνο το ποσοστό. Η πορεία μπορεί να αλλάξει, λόγω στις ρυθμίσεις αναστοιχειοθέτησης

σημείου τέλους, ή δρομολόγησης, ή πολλές νέες συνδέσεις να ξεκινήσουν, μειώνοντας σημαντικά το δίκαιο ποσοστό της εφαρμογής. Εντούτοις, φαίνεται λογικό να επιτραπεί σε μια εφαρμογή να συμβάλει στην παροδική συμφόρηση σε περιόδους αλλαγής, σε αντάλλαγμα τη βελτίωση της ανταπόκρισης της εφαρμογής μετά από τις μη απασχολημένες περιόδους.

Μερικά πρωτόκολλα που χρησιμοποιούν TFRC ήδη διευκρινίζουν ότι το ποσοστό αποστολής δεν μειώνεται ποτέ κάτω από το ποσοστό των τεσσάρων πακέτων ανά RTT, κατά τη διάρκεια μιας μη απασχολημένης περιόδου. Το γρηγορότερο καινούργιο ξεκίνημα διευκρινίζει ότι το ποσοστό αποστολής δεν μειώνεται ποτέ κάτω από οκτώ πακέτα ανά RTT, για τα μικρά πακέτα. Επιπλέον, επειδή οι ροές έχουν ήδη κάποιες (ενδεχομένως παλαιές) πληροφορίες για την πορεία, το γρηγορότερο καινούργιο ξεκίνημα επιτρέπει στις ροές να τετραπλασιάσουν το ποσοστό αποστολής τους σε κάθε RTT που δεν υπάρχει συμφόρηση, αντί του διπλασιασμού, μέχρι το προηγουμένως πραγματοποιημένο ποσοστό. Οποιοδήποτε γεγονός συμφόρησης σταματά αυτό το γρηγορότερο καινούργιο ξεκίνημα και μεταστρέφει το TFRC στην αποφυγή συμφόρησης (congestion avoidance).

Το TCP έχει εξετάσει ιστορικά την αδράνεια είτε με την κράτηση του cwnd εξ ολοκλήρου ανοικτό ("άμεση έναρξη") είτε με την είσοδο στην αργή έναρξη. Η πρώτη επιλογή είναι πάρα πολύ φιλελεύθερη, η δεύτερη πάρα πολύ συντηρητική. Σαφώς μια μικρή αδρανής χρονική περίοδος δεν είναι μια νέα σύνδεση: τα πρόσφατα στοιχεία δείχνουν ότι η σύνδεση θα μπορούσε να στηρίζει κάποιο σημαντικό ρυθμό αποστολής. Εντούτοις, οι πιο μεγάλες αδρανείς περιόδους είναι πιο προβληματικές, και οι αδρανείς περιόδους ωρών θα φαίνονταν να απαιτούν την χρήση της αργής έναρξης. Στο TCP, το παράθυρο συμφόρησης μοιράζεται για κάθε RTT που ο αποστολέας έχει παραμείνει αδρανής, και το παράθυρο ανοίγει πάλι στην αργή έναρξη όταν τελειώνει η αδρανής περίοδος. Το γρηγορότερο καινούργιο ξεκίνημα πρέπει να είναι αποδεκτό για το TFRC εάν το σενάριο στη χειρότερη περίπτωση είναι αποδεκτό. Τα ρεαλιστικά σενάρια στη χειρότερη περίπτωση περιλαμβάνουν τα ακόλουθα σενάρια:

- Η πορεία δρομολόγησης των πακέτων αλλάζει και ο παλαιός ρυθμός αποστολής δεν είναι αποδεκτός στη νέα πορεία. Τα RTTs είναι επίσης μικρότερα στη νέα πορεία,

επομένως το γρηγορότερο καινούριο ξεκίνημα καταστρέφει άλλες συνδέσεις για πολλαπλά RTTs, όχι μόνο ένα

- δύο (ή περισσότερες) συνδέσεις μπαίνουν στο γρηγορότερο καινούριο ξεκίνημα ταυτόχρονα. Το ποσοστό πτώσης πακέτων μπορεί να είναι δύο φορές χειρότερο, για ένα RTT, από το εάν είχαν αρχίσει με αργό ξεκίνημα μετά από τις αδρανείς περιόδους τους.
- εκτός από τις συνδέσεις γρήγορου ξεκινήματος, υπάρχουν και σύντομες συνδέσεις TCP ή DCCP που αρχίζουν και που σταματούν όλη την ώρα, με τα αρχικά παράθυρα τριών ή τεσσάρων πακέτων. Υπάρχουν επίσης συνδέσεις TCP με μικρές ήρεμες χρονικές περιόδους. Οι ακουστικές και τηλεοπτικές συνδέσεις έχουν αδρανείς περιόδους. Το διαθέσιμο εύρος ζώνης μπορεί να ποικίλει μέσα στο χρόνο, λόγω του εύρους ζώνης που χρησιμοποιήθηκε από την κυκλοφορία υψηλότερης προτεραιότητας (κυκλοφορία δρομολόγησης, και diffserv). Όλο αυτό συμβαίνει αμέσως, έτσι το συνολικό ποσοστό άφιξης πακέτων αλλάζει από ένα RTT στο επόμενο.

Σημειώνουμε ότι το γρηγορότερο καινούριο ξεκίνημα σε VoIP TFRC είναι αρκετά πιο αυστηρό από το γρηγορότερο καινούριο ξεκίνημα στην προεπιλογή TFRC. Σε VoIP TFRC, ο αποστολέας είναι περιορισμένος στην αποστολή το πολύ ενός πακέτου κάθε ελάχιστο χρονικό διάστημα. Ομοίως, το γρηγορότερο καινούριο ξεκίνημα στην προεπιλογή TFRC είναι πιο αυστηρό από ότι το γρηγορότερο καινούριο ξεκίνημα εάν θα προστίθετο στο TCP. Το TFRC ελέγχεται από το ρυθμό αποστολής, ενώ το TCP ελέγχεται από ένα παράθυρο, που θα μπορούσε να στείλει σε ένα πολύ ταχύ ρυθμό, ελλείψει ενός μηχανισμού για ελεγχόμενο ρυθμό αποστολής.

Η προεπιλογή TFRC, χωρίς την παραλλαγή VoIP, σχεδιάστηκε με στόχο την δικαιοσύνη με το TCP, για ροές με το ίδιο μέγεθος πακέτων, και ίδιου ποσοστού πτώσης πακέτων. Όταν το ζήτημα της δικαιοσύνης μεταξύ των ροών με διαφορετικά μεγέθη πακέτων εμφανίζεται, πρέπει να ελέγξουμε εάν τα ποσοστά πτώσης πακέτων που βιώνονται από τις ροές συσχετίζονται με το μέγεθος πακέτων.

Στις προσομοιώσεις που έγιναν με ροές TCP που ανταγωνίζονται ροές VoIP TFRC με μικρότερα πακέτα, σε ένα σενάριο με έναν κορεσμένο δρομολογητή με σειρά αναμονής

Drop Tail, η ροή TCP VoIP λαμβάνει περισσότερο από το μέρος που της αναλογεί σε ψηφιολέξεις ανά δευτερόλεπτο. Αυτό συμβαίνει ακόμη και για ένα σενάριο όπου οι ροές TCP είναι πιο επιθετικές, SACK TCP, timestamps, και ECN. Όπως αναμένεται, η συμπεριφορά των πακέτων που απορρίφθηκαν μπορεί να ποικίλει αλλάζοντας τον ενεργό διοικητικό μηχανισμό σειράς αναμονής στο δρομολογητή (Active Queue Management mechanism). Όταν οι δρομολογητές χρησιμοποιούν το RED μηχανισμό στον τρόπο πακέτων, όπου κάθε πακέτο έχει την ίδια πιθανότητα να απορριφθεί, οι ροές TFRC και TCP λαμβάνουν κατά προσέγγιση το ίδιο ποσοστό πτώσης πακέτων. Αντίθετα, όταν οι δρομολογητές χρησιμοποιούν το RED σε ψηφιολέξεις, όπου κάθε ψηφιολέξη έχει την ίδια πιθανότητα να απορριφθεί, η ροή TFRC βιώνει ένα πολύ μικρότερο ποσοστό πτώσης πακέτων από τις ροές TCP, σε προσομοιώσεις με μέτρια επίπεδα συμφόρησης.

Ροές	TCP ρυθμός απόρριψης	Απόδοση TCP	TFRC ρυθμός απόρριψης	Απόδοση TFRC
10	0.010	0.71	0.010	0.14
25	0.067	0.61	0.063	0.32
50	0.180	0.37	0.162	0.57
75	0.257	0.18	0.261	0.75
100	0.295	0.15	0.407	0.82
150	0.370	0.13	0.571	0.83

Πίνακας 2.1 Αποτελέσματα προσομοίωσης με σειρά αναμονής DropTail.

Ο πίνακας 1 ανωτέρω παρουσιάζει τα αποτελέσματα των προσομοιώσεων με N ροές TCP, με απεριόριστα στοιχεία για να σταλούν και πακέτα 1460 ψηφιολέξεων, που ανταγωνίζονται σε N ροές TFRC VoIP με 100 πακέτα δεδομένων ανά δευτερόλεπτο και μέγεθος πακέτων 200 ψηφιολέξεων. Το N κυμαίνεται από 10 έως 150, με μια κορεσμένη σύνδεση 10 Mbps, και κάθε προσομοίωση διαρκεί 100 δευτερόλεπτα. Κάθε σειρά του πίνακα δίνει το αποτέλεσμα μιας προσομοίωσης, που δίνει το ποσοστό πτώσης πακέτων για τις ροές TCP και TFRC, και το ποσοστό του εύρους ζώνης συνδέσεων που χρησιμοποιείται από τις N ροές TCP και τις N ροές TFRC αντίστοιχα.

Οι προσομοιώσεις στον πίνακα επάνω χρησιμοποιούν σειρά αναμονής DropTail. Όταν το N είναι μικρό, η συμφόρηση είναι χαμηλή, και κάθε ροή VoIP TFRC περιορίζεται από το μέγιστο ρυθμό αποστολής στοιχείων των 160 Kbps από την εφαρμογή. Σε αυτές

τις περιπτώσεις, οι ροές TCP λαμβάνουν αρκετά περισσότερο εύρος ζώνης από τις ροές TFRC. Όταν το N είναι μεγάλο, οδηγώντας το ποσοστό πτώσης πακέτων μέχρι 25-50%, οι ροές TFRC λαμβάνουν σημαντικά περισσότερο εύρος ζώνης από τις ροές TCP. Σε κάθε μια από αυτές τις προσομοιώσεις, οι ροές TCP και TFRC έλαβαν κάπως συγκρίσιμα ποσοστά πτώσης πακέτων.

Οι συνδέσεις SACK TCP σε αυτές τις προσομοιώσεις χρησιμοποιούν τις προεπιλεγμένες παραμέτρους του προσομοιωτή NS, με περιορισμένη μετάβαση, και ένα ελάχιστο RTO της τάξης των 200 ms. Η προσθήκη timestamps στη σύνδεση TCP δεν άλλαξε τα αποτελέσματα.

Ροές	TCP ρυθμός απόρριψης	Απόδοση TCP	TFRC ρυθμός απόρριψης	Απόδοση TFRC
10	0.009	0.73	0.008	0.14
25	0.051	0.60	0.037	0.33
50	0.166	0.38	0.183	0.56
75	0.218	0.16	0.231	0.79
100	0.251	0.12	0.392	0.87
150	0.326	0.10	0.550	0.87

Πίνακας 2.2: Προσομοίωση με σειρές αναμονής RED με μέθοδο πακέτων.

Ο πίνακας 2 δείχνει ότι η προσομοίωση με σειρές αναμονής RED στον τρόπο πακέτων, όπου κάθε πακέτο απορρίπτεται με την ίδια πιθανότητα, είναι κατά προσέγγιση η ίδια με εκείνη με τις σειρές αναμονής DropTail. Οι πίνακες 1 και 2 προτείνουν ότι η λειτουργία απάντησης σε συμφόρηση του TCP που χρησιμοποιήθηκε στο TFRC θα μπορούσε να τροποποιηθεί για να δώσει πιο ρεαλιστικές τιμές για τη ρυθμοαπόδοση του TCP σε υψηλότερα ποσοστά πτώσης πακέτων. Οι προσομοιώσεις χρησιμοποιώντας ECN και TCP timestamps, έχουν μικρή αλλαγή στα γενικά αποτελέσματα.

Ροές	TCP ρυθμός απόρριψης	Απόδοση TCP	TFRC ρυθμός απόρριψης	Απόδοση TFRC
10	0.009	0.72	0.008	0.14
25	0.049	0.60	0.020	0.33
50	0.138	0.29	0.031	0.65
75	0.181	0.12	0.168	0.83
100	0.242	0.11	0.356	0.85
150	0.323	0.09	0.536	0.87

Πίνακας 2.3: Προσομοίωση με σειρές αναμονής RED με μέθοδο ψηφιολέξεων.

Ο πίνακας 3 παρουσιάζει τα αποτελέσματα προσομοίωσης με σειρές αναμονής RED με μέθοδο ψηφιολέξεων, όπου ο δρομολογητής λαμβάνει υπόψη το μέγεθος των πακέτων για να πάρει απόφαση εάν πρέπει να ριχτεί ένα πακέτο. Πάλι, για υψηλότερες τιμές του N , η ροή VoIP TFRC λαμβάνει περισσότερο από το μερίδιό της στο εύρος ζώνης συνδέσεων. Ο στόχος της παραλλαγής VoIP TFRC ήταν οι ροές TCP και οι ροές VoIP TFRC να έχουν δικαιοσύνη στο ρυθμό αποστολής σε bps, σε ένα σενάριο όπου κάθε πακέτο λαμβάνει κατά προσέγγιση την ίδια πιθανότητα να απορριφθεί. Σε ένα σενάριο όπου τα μεγάλα πακέτα είναι πιθανότερο να ριχτούν παρά τα μικρά πακέτα, ή όπου ροές με έντονο ρυθμό αποστολής είναι πιθανότερο να έχουν πακέτα που θα ριχτούν από ότι ροές με ένα ομαλό ρυθμό αποστολής, οι ροές που χρησιμοποιούν την παραλλαγή VoIP TFRC θα μπορούσαν να λάβουν περισσότερο εύρος ζώνης από τις ανταγωνιστικές ροές TCP.

Αν και η παραλλαγή VoIP TFRC δεν απαιτεί οι εφαρμογές να περιορίζονται από ένα μέγιστο ρυθμό αποστολής, στην πραγματικότητα οι ροές VoIP έχουν ένα τέτοιο περιορισμό. Όπως εμφανίζεται στις προσομοιώσεις του Tom Phelan [3], αυτό περιπλέκει το ζήτημα να ερευνήσουμε τη δικαιοσύνη μεταξύ των ροών TCP και VoIP TFRC. Επιπλέον, για τις ροές VoIP TFRC με ένα μέγιστο ρυθμό αποστολής 96 Kbps, ή με ένα μικρότερο μέγιστο ρυθμό αποστολής, το VoIP TFRC μειώνει το ρυθμό αποστολής αυτών των ροών όταν το ποσοστό πτώσης πακέτων είναι αρκετά υψηλό. Σε αυτό το καθεστώς, η απόδοση του TFRC καθορίζεται πάρα πολύ από την ακρίβεια της λειτουργίας απάντησης του TCP στην αντιπροσώπευση του πραγματικού ρυθμού αποστολής μιας σύνδεσης TCP. Σε αυτό το καθεστώς των υψηλών ποσοστών πτώσης πακέτων, η απόδοση της σύνδεσης TCP επηρεάζεται πάρα πολύ από τον αλγόριθμο TCP, από την χρήση timestamps ή/και ECN, από τον ελάχιστο RTO κ.τ.λ. Κατά συνέπεια, για τις προσομοιώσεις σε αυτό το καθεστώς, υπάρχουν πολλές παράμετροι που πρέπει να εξετάσουμε. Είναι καλό να ασφαλιστεί ότι οι προσομοιώσεις που ερευνούν τη δικαιοσύνη περιλαμβάνουν την εξερεύνηση της δικαιοσύνης με την επιθετικότερη προσαρμογή των μηχανισμών του TCP με τα τρέχοντα πρότυπα, δηλαδή το SACK TCP που χρησιμοποιεί timestamps, το ECN και ένα ελάχιστο RTO της τάξης των 100-200 ms.

2.4 Streaming Control Transport Protocol (SCTP)

Το SCTP (Stream Control Transmission Protocol) είναι ένα νέο πρωτόκολλο μεταφοράς δεδομένων το οποίο αναπτύχθηκε από ερευνητές της βιομηχανίας και από πανεπιστήμια, με στόχο την μεταφορά τηλεπικοινωνιακών σημάτων σε δίκτυα βασισμένα στο IP [2]. Είναι ένα πρωτόκολλο που λειτουργεί με την αρχή του παραθύρου, αξιόπιστο και με ένα μηχανισμό ελέγχου συμφόρησης παρόμοιο με αυτό του TCP. Έχει όμως πολλά νέα πλεονεκτήματα τα οποία το κάνουν πιο κατάλληλο για streaming media content.

Τα νέα αυτά πλεονεκτήματα που παρέχονται από το SCTP περιλαμβάνουν πολλαπλά streams ανάμεσα σε μια σύνδεση μεταξύ δύο άκρων, ομαδοποίηση δεδομένων και έλεγχο πληροφορίας μαζί μέσα στο ίδιο πακέτο και ευελιξία στην σειρά παράδοσης των πακέτων (τα πακέτα δεν είναι απαραίτητο να παραδίδονται με την σωστή σειρά). Το SCTP υποθέτει πως δεν υπάρχει η δυνατότητα QoS (quality of service) στους δρομολογητές, ούτε AQM (active queue management) και ECN (explicit congestion notification). Μπορεί όμως να επωφεληθεί από την ύπαρξη τέτοιων μηχανισμών.

Το κύριο νέο στοιχείο που υλοποιεί το SCTP [9] είναι η έννοια της συσχέτισης (association) η οποία είναι μια γενίκευση της σύνδεσης του TCP. Μια συσχέτιση ανάμεσα σε δύο SCTP άκρα είναι μια σύνδεση μεταξύ τους, όπου τα δεδομένα της εφαρμογής αλλά και τα δεδομένα ελέγχου, μπορούν να μεταδοθούν με ευελιξία και αξιοπιστία όσον αφορά την σειρά παράδοσης των πακέτων. Η έννοια της συσχέτισης στο SCTP δημιουργεί κύριες λειτουργίες οι οποίες κανονικά θα χρειάζονται από τις streaming εφαρμογές και θα παρέχει ένα API για χρήση από την εφαρμογή ώστε να έχει πρόσβαση στις λειτουργίες αυτές.

Τα κύρια χαρακτηριστικά του SCTP είναι ότι τα πακέτα χωρίζονται σε τεμάχια (chunks), και κάθε τεμάχιο μπορεί να ανήκει σε διαφορετικό λογικό stream [10]. Τα δεδομένα που περιέχει το κάθε τεμάχιο μπορεί να είναι διαφορετικών μεγεθών, φτάνει το ολικό μέγεθος του πακέτου να είναι μικρότερο ή ίσο με το μέγιστο ρυθμό μεταφοράς δεδομένων της σύνδεσης. Ορισμένα τεμάχια είναι προκαθορισμένα όπως τα τεμάχια

αρχικοποίησης της σύνδεσης, μεταφοράς δεδομένων και μεταφοράς επιβεβαιώσεων παραλαβής. Ο αποστολέας έχει την δυνατότητα να κωδικοποιήσει πολλά τεμάχια σε ένα πακέτο και η αντίστοιχη αποκωδικοποίηση γίνεται στον παραλήπτη. Σε αυτή την διαδικασία υπάρχουν δύο βασικές παράμετροι που πρέπει να λαμβάνονται υπόψη, η αξιοπιστία και η σειρά στην παράδοση των δεδομένων.

Η αξιοπιστία καθορίζεται ως ο αριθμός των επαναλαμβανόμενων αποστολών πακέτων για πακέτα που έχουν χαθεί, όπως καθορίζεται από τον αποστολέα μετά από μελέτη των επιβεβαιώσεων παραλαβής πακέτων που στέλνει ο παραλήπτης [11]. Μια αξιοπιστία βαθμού μηδέν σημαίνει πως δεν πρέπει να έχουμε επαναλαμβανόμενες αποστολές, ενώ μια αξιοπιστία βαθμού ένα σημαίνει πως πρέπει να επιχειρήσουμε το πολύ μία επαναλαμβανόμενη αποστολή κ.ο.κ.

Η σειρά στην παράδοση μπορεί να χρησιμοποιηθεί ώστε να αποφύγουμε το λεγόμενο head-of-line blocking το οποίο εμφανίζεται στο TCP. Εάν χρησιμοποιηθεί η σημαία για την παράδοση εκτός σειράς, το SCTP θα περάσει τα δεδομένα στην εφαρμογή με την σειρά με τα οποία τα έχει παραλάβει, αντί να περιμένει για επανάληψη της αποστολής πακέτων, εκεί που έχει διαπιστωθεί ένα κενό στην ακολουθία των αυξόντων αριθμών των πακέτων.

Ο μηχανισμός ελέγχου συμφόρησης του SCTP είναι σχεδόν ο ίδιος με αυτό του TCP New Reno. Κρατείται ένα παράθυρο συμφόρησης για κάθε συσχέτιση και ακολουθεί τον αλγόριθμο AIMD με τις φάσεις της αργής αρχής (slow start), της αποφυγής συμφόρησης (congestion avoidance), της γρήγορης αναμετάδοσης (fast retransmit) και της γρήγορης ανάκτησης (fast recovery). Αυτό οδηγεί σε καλή συμπεριφορά σε σχέση με το TCP και οι ροές ανάμεσα σε SCTP και TCP λαμβάνουν δίκαιο ποσοστό όταν συναγωνίζονται για απόκτηση εύρους ζώνης στην ίδια σύνδεση.

Το SCTP είναι ένα νέο IP πρωτόκολλο μεταφοράς δεδομένων που υπάρχει σε παράλληλο επίπεδο με το UDP (User Datagram Protocol) και το TCP (Transmission Control Protocol), το οποίο παρέχει πολλές λειτουργίες μεταφοράς δεδομένων σε πολλές εφαρμογές διαδικτύου.

Όπως και το TCP το SCTP παρέχει ένα αξιόπιστο τρόπο μεταφοράς δεδομένων και εγγυάται ότι τα δεδομένα μεταφέρονται στο δίκτυο χωρίς σφάλματα και στην σωστή σειρά. Όπως το TCP είναι ένας μηχανισμός που στηρίζεται σε τμήματα (session-oriented) πράγμα που σημαίνει ότι δημιουργείται μια σχέση ανάμεσα στα άκρα των τμημάτων του SCTP πριν αρχίσουν να στέλνονται τα πακέτα των δεδομένων. Αυτή η σχέση διατηρείται μέχρι όλα τα δεδομένα να σταλούν με επιτυχία.

Σε αντίθεση με το TCP το SCTP παρέχει ένα αριθμό λειτουργιών που είναι κρίσιμες για μεταφορά τηλεφωνικών σημάτων και ταυτόχρονα μπορεί να ωφελήσει άλλες εφαρμογές που χρειάζονται τρόπο μεταφοράς με επιπρόσθετη απόδοση και αξιοπιστία.

Τα κύρια χαρακτηριστικά του SCTP είναι ότι είναι ένα πρωτόκολλο μονής ροής (unicast) και υποστηρίζει την ανταλλαγή δεδομένων ανάμεσα σε ακριβώς δύο ακραία σημεία, παρόλο που αυτά μπορούν να αναπαρασταθούν με πολλαπλές IP διευθύνσεις. Το SCTP παρέχει αξιόπιστη μεταφορά δεδομένων, και αναγνωρίζει πότε τα δεδομένα έχουν απορριφθεί, αλλάζει σειρά, διπλασιαστεί ή φθαρεί. Επίσης είναι σε θέση να μεταδώσει ξανά δεδομένα τα οποία δεν έχουν φτάσει στο παραλήπτη όπως τα έστειλε ο αποστολέας. Η αποστολή του SCTP είναι full duplex [12]. Είναι επίσης προσαρμοσμένο με βάση τα μηνύματα (message oriented) σε αντίθεση με το TCP το οποίο είναι προσαρμοσμένο με βάση τα bytes (bytes oriented). Τέλος προσαρμόζει το ρυθμό αποστολής δεδομένων με παρόμοιο τρόπο με το TCP και μειώνει το ρυθμό αποστολής δεδομένων όσο αυξάνεται ο φόρτος του δικτύου. Είναι σχεδιασμένο να συμπεριφέρεται μη ανταγωνιστικά σε σχέση με το TCP προσπαθώντας να χρησιμοποιεί το ίδιο εύρος ζώνης (bandwidth).

Το όνομα του πρωτοκόλλου προέρχεται από την συνάρτηση multi-streaming που παρέχει το SCTP. Αυτή η συνάρτηση επιτρέπει στα δεδομένα να τεμαχιστούν σε πολλά μικρότερα κομμάτια και να τοποθετηθούν σε πολλαπλά streams που έχουν την ιδιότητα να παραδοθούν στον παραλήπτη σε ανεξάρτητη σειρά σε σχέση με την σειρά αποστολής. Επομένως εάν χαθεί κάποιο μήνυμα που βρίσκεται σε ένα stream θα επηρεάσει μόνο την παράδοση μέσα στο συγκεκριμένο stream.

Σε αντίθεση το TCP υποθέτει μόνο ένα stream δεδομένων και εγγυάται ότι η παράδοση σε αυτό το stream θα γίνει σε σειρά σε σχέση με την σειρά αποστολής των bytes μέσα στο stream. Καθώς αυτό είναι επιθυμητό για την παράδοση ενός αρχείου ή μιας σειράς δεδομένων, προκαλεί επιπρόσθετη καθυστέρηση όταν χάνονται μηνύματα ή δημιουργούνται σφάλματα στο δίκτυο. Εάν συμβεί αυτό το TCP πρέπει να καθυστερήσει την παράδοση των δεδομένων μέχρι να αποκατασταθεί η σωστή σειρά παράδοσης των δεδομένων, είτε με την παραλαβή πακέτου που είναι εκτός σειράς, είτε με την αναμετάδοση του συγκεκριμένου πακέτου που έχει χαθεί.

Για ορισμένες εφαρμογές το χαρακτηριστικό της παραλαβής πακέτων στην σωστή σειρά δεν είναι απόλυτα αναγκαίο. Στην μετάδοση τηλεφωνικών σημάτων είναι μόνο αναγκαίο να διατηρήσουμε την σειρά σε πακέτα που επηρεάζουν την ίδια πηγή, για παράδειγμα το ίδιο τηλεφώνημα ή το ίδιο κανάλι. Τα άλλα μηνύματα είναι ασθενώς συνδεδεμένα και μπορούν να παραδοθούν χωρίς να χρειάζεται να διατηρήσουμε την σειρά παράδοσης.

Ακόμη ένα παράδειγμα για την χρήση της συνάρτησης multi-streaming είναι η παράδοση πολυμεσικών κειμένων, όπως οι ιστοσελίδες, όταν αυτό γίνεται σε ένα μόνο session [13]. Αφού τα πολυμεσικά κείμενα αποτελούνται από αντικείμενα από διαφορετικά μεγέθη και διαφορετικούς τύπους, το multi-streaming επιτρέπει την μεταφορά αυτών των αντικειμένων να είναι με μερική σειρά αντί με ολική σειρά και αυτό μπορεί να έχει ως αποτέλεσμα την απόδοση στην μεταφορά των δεδομένων.

Την ίδια στιγμή η μεταφορά γίνεται σε μια και μόνο συσχέτιση (association) έτσι ώστε όλα τα streams να είναι υποκείμενα σε μία και μόνο ροή και σε ένα μηχανισμό ελέγχου συμφόρησης με αποτέλεσμα να μειώνεται το επιπρόσθετο κόστος που χρειαζόταν στο επίπεδο μεταφοράς δεδομένων.

Το SCTP καταφέρνει να εφαρμόσει το μηχανισμό multi-streaming με την ύπαρξη ανεξαρτησίας ανάμεσα στην μετάδοση των δεδομένων και την παραλαβή των δεδομένων. Συγκεκριμένα κάθε πακέτο στο πρωτόκολλο χρησιμοποιεί δύο σύνολα αριθμών. Το πρώτο είναι για την μετάδοση των δεδομένων που ελέγχει την μετάδοση των μηνυμάτων και την ανίχνευση χαμένων μηνυμάτων, και το δεύτερο είναι για το

ζεύγος stream id και stream sequence number το οποίο χρειάζεται για να καθορίσουμε την σειρά της παράδοσης των παραλαμβανόμενων δεδομένων.

Αυτή η ανεξαρτησία των μηχανισμών επιτρέπει στον παραλήπτη να καθορίσει αμέσως πότε προκύπτει ένα κενό στην σειρά παραλαβής των δεδομένων, καθώς και κατά πόσο τα μηνύματα που έχουν παραληφθεί μετά την ανίχνευση του κενού, ανήκουν στο επηρεαζόμενο stream. Εάν το μήνυμα ανήκει στο επηρεαζόμενο stream θα υπάρχει ένα κενό στον αύξοντα αριθμό του stream, ενώ εάν το μήνυμα ανήκει σε άλλο stream το κενό αυτό δεν θα φαίνεται. Επομένως ο παραλήπτης μπορεί να συνεχίσει να παραλαμβάνει μηνύματα από τα streams που δεν επηρεάζονται από το χάσιμο του μηνύματος και να αποθηκεύει τα μηνύματα από το επηρεαζόμενο stream μέχρι να παραλάβει το πακέτο που θα συμπληρώσει το κενό.

Ακόμα ένα χαρακτηριστικό του SCTP είναι το multi-homing ή αλλιώς η δυνατότητα για ένα ακραίο κόμβο να υποστηρίζει πολλαπλές IP διευθύνσεις. Το κέρδος από το multi-homing είναι ότι υπάρχει η πιθανότητα μια ροή (session) να συνεχίσει να υποστηρίζει την μεταφορά δεδομένων σε περίπτωση βλάβης του δικτύου. Σε μια κανονική ροή (session) μια βλάβη σε ένα τοπικό σημείο του δικτύου μπορεί να ισοπεδώσει την μεταφορά δεδομένων, ενώ μια βλάβη στον πυρήνα του δικτύου μπορεί να δημιουργήσει προσωρινή ανικανότητα στην μεταφορά δεδομένων μέχρι τα πρωτόκολλα να βρουν ένα εναλλακτικό μονοπάτι. Η χρήση του multi-homed SCTP χρησιμοποιείται για να υποστηρίξει την αρχική σύνδεση και η χρήση του γίνεται μόνο για επαναληπτικούς σκοπούς δηλαδή όταν έχουμε αποστολή πακέτων που δεν έχουν φτάσει στον προορισμό τους.

Μια διεύθυνση ορίζεται ως η κύρια διεύθυνση (primary address) και χρησιμοποιείται ως η διεύθυνση στην οποία όλα τα πακέτα θα αποστέλλονται. Τα πακέτα τα οποία στέλλονται ξανά γιατί δεν έχουν φτάσει στον προορισμό τους, χρησιμοποιούν τις εναλλακτικές διευθύνσεις ώστε να βελτιώσουν τις πιθανότητες να φτάσουν στον τελικό προορισμό τους. Η ανικανότητα να σταλούν τα πακέτα από την κύρια διεύθυνση για κάποιο χρονικό διάστημα, έχει ως αποτέλεσμα όλα τα πακέτα να σταλούν από τις εναλλακτικές διευθύνσεις. Αυτό γίνεται μέχρι ο μηχανισμός heartbeat να μπορέσει να βρει και πάλι επαφή με την κύρια διεύθυνση.

Για να υποστηρίξουν το multi-homing τα άκρα του SCTP ανταλλάσσουν λίστες με τις διευθύνσεις κατά την αρχικοποίηση της σύνδεσης (session). Κάθε άκρο πρέπει να είναι σε θέση να λαμβάνει μηνύματα από οποιαδήποτε διεύθυνση που σχετίζεται με το άκρο αυτό. Για κάθε συγκεκριμένο session υπάρχει ένα port number που είναι μοναδικό ανάμεσα σε όλες τις διευθύνσεις του άκρου. Για να ενισχυθεί η ασφάλεια της σύνδεσης για κάθε νέο συμβάν που λαμβάνει χώρα, στέλλονται επιβεβαιώσεις στον παραλήπτη.

Η διαδικασία αρχικοποίησης βασίζεται σε μια σειρά ανταλλαγής τεσσάρων μηνυμάτων όπου τα δεδομένα αρχίζουν να ανταλλάσσονται κατά την αποστολή του τρίτου και τέταρτου μηνύματος, όταν δηλαδή η σύνδεση (association) έχει ήδη επικυρωθεί. Ένας μηχανισμός cookie έχει προστεθεί μέσα στην ακολουθία μηνυμάτων για να προστατεύσει από κάποιας μορφής επίθεση στην σύνδεση. Ο μηχανισμός αυτός προστατεύει κυρίως από ένα τυφλό αποστολέα (blind attacker) ο οποίος στέλλει πακέτα αρχικοποίησης INIT με σκοπό να σπαταλήσει την μνήμη του SCTP εξυπηρετητή. Ο εξυπηρετητής όμως αντί να σπαταλά μνήμη κάθε φορά για ένα μπλοκ ελέγχου μετάβασης, δημιουργεί ένα cookie με τις πληροφορίες του μπλοκ και με μια διάρκεια ζωής και μια υπογραφή αυθεντικότητας την οποία θα στείλει μόλις πάρει πίσω το μήνυμα INIT ACK από τον παραλήπτη. Ο παραλήπτης όμως αφού είναι τυφλός δεν θα στείλει ποτέ το μήνυμα αυτό και ο εξυπηρετητής δεν θα σπαταλήσει επιπλέον μνήμη για να δημιουργήσει ένα νέο μπλοκ. Ένας κανονικός παραλήπτης SCTP θα πάρει το cookie και θα επιστρέψει ένα πακέτο επιβεβαίωσης της παραλαβής το οποίο θα χρησιμοποιηθεί από τον εξυπηρετητή για να ξαναδημιουργήσει το μπλοκ.

Οι επαναληπτικές αποστολές πακέτων δεδομένων μπορούν να προκύψουν είτε από λήξη του χρονόμετρου επαναληπτικών αποστολών πακέτων είτε από την παραλαβή επιβεβαιώσεων SACK οι οποίες υποδεικνύουν ότι κάποια πακέτα δεδομένων δεν έχουν παραληφθεί κανονικά. Για να μειωθεί το ενδεχόμενο συμφόρησης ο ρυθμός των επαναληπτικών αποστολών είναι περιορισμένος. Το χρονόμετρο ρυθμίζεται με βάση τους υπολογισμούς για το μέσο χρόνο που χρειάζεται ένα πακέτο για να φτάσει στον προορισμό του.

Σε μια ενεργή σύνδεση με σχεδόν παρόμοιες αποστολές πακέτων για κάθε session οι επιβεβαιώσεις SACK είναι πιο πιθανή αιτία να προκαλέσουν επαναληπτικές αποστολές πακέτων από ότι η λήξη του χρονομέτρου. Για να μειώσουμε ακόμα περισσότερο την πιθανότητα να στείλουμε ξανά αλλά αχρείαστα ένα πακέτο, ένας κανόνας τεσσάρων SACK χρησιμοποιείται. Δηλαδή οι επαναληπτικές αποστολές πακέτων συμβαίνουν μόνο εφόσον ο αποστολέας έχει παραλάβει το τέταρτο SACK το οποίο υποδεικνύει ότι ένα πακέτο δεν έχει παραληφθεί. Ο κανόνας αυτός έχει σκοπό να αποκλείσει επαναληπτικές αποστολές οι οποίες θα συνέβαιναν επειδή κάποια πακέτα έχουν παραληφθεί εκτός σειράς.

Αρχικά διατηρούμε ένα αριθμό ο οποίος υποδεικνύει τον αριθμό των επαναληπτικών αποστολών πακέτων σε μια συγκεκριμένη διεύθυνση χωρίς να πάρουμε πίσω επιτυχώς κάποια επιβεβαίωση της αποστολής. Όταν αυτός ο αριθμός ξεπεράσει κάποιο αρχικά καθορισμένο μέγιστο αριθμό, η συγκεκριμένη διεύθυνση ορίζεται ως ανενεργή (inactive), γίνεται κάποια μορφής ειδοποίηση στην εφαρμογή και το SCTP ξεκινά να χρησιμοποιεί μια εναλλακτική διεύθυνση για να μπορεί να αποστείλει τα πακέτα δεδομένων.

Επίσης πακέτα κτύπου καρδιάς (heartbeat) στέλνονται περιοδικά σε όλες τις μη χρησιμοποιούμενες διευθύνσεις (δηλαδή σε όλες τις εναλλακτικές διευθύνσεις) , και ένας μετρητής διατηρείται στον αριθμό των πακέτων κτύπου καρδιάς που έχουν σταλεί στις διευθύνσεις αυτές χωρίς να έχουμε λάβει μια επιβεβαίωση κτύπου καρδιάς. Όταν αυτός ο μετρητής ξεπεράσει μια αρχικά καθορισμένη μέγιστη τιμή, οι διευθύνσεις αυτές ορίζονται ως ανενεργές.

Τα μηνύματα κτύπου καρδιάς συνεχίζονται να στέλνονται σε ανενεργές διευθύνσεις μέχρι να πάρουμε πίσω μια επιβεβαίωση. Εάν συμβεί κάτι τέτοιο η συγκεκριμένη διεύθυνση ορίζεται και πάλι ως ενεργή (active). Ο ρυθμός της αποστολής των πακέτων κτύπου καρδιάς είναι συνάρτηση του χρόνου αποστολής-παραλαβής πακέτων (round trip time RTO) και κάποιου επιπρόσθετου χρόνου για ταξινόμηση των δεδομένων.

Παρόμοια με την αποτυχία μονοπατιού υπάρχει και η αποτυχία κόμβου. Ένας μετρητής διατηρείται για όλες τις πιθανές διευθύνσεις και αυξάνεται όταν δεν έχουμε

επιβεβαίωση επαναληπτικής αποστολής πακέτου ή όταν ένα πακέτο κτύπου καρδιάς δεν πάρει επιβεβαίωση αποστολής. Όταν αυτός ο μετρητής ξεπεράσει μια αρχικά καθορισμένη μέγιστη τιμή το άκρο ορίζεται ως μη προσβάσιμο και η σύνδεση SCTP τερματίζεται.

Μετά από ανάλυση των ωφελημάτων που προκύπτουν, αποφασίστηκε η προσθήκη μιας μεθόδου μη αξιόπιστης μεταφοράς δεδομένων στο SCTP [14]. Η σχεδίαση που προέκυψε επιτρέπει την συνύπαρξη αξιόπιστων και μη αξιόπιστων ροών μέσα σε μια σύνδεση (association) SCTP. Πιο κάτω παρουσιάζουμε μερικά από τα πλεονεκτήματα της προσθήκης της μεθόδου μη αξιόπιστης μεταφοράς δεδομένων στο SCTP.

- Μερικές εφαρμογές μπορεί να επωφεληθούν από την μη αξιόπιστη μεταφορά δεδομένων με το να μπορούν χρησιμοποιώντας μόνο μια σύνδεση (association) να μεταφέρουν τόσο δεδομένα που χρειάζονται αξιόπιστη μεταφορά όπως σελίδες κειμένου και λογιστικά κείμενα, όσο και μη αξιόπιστα δεδομένα όπως πολυμεσικά αρχεία που δεν χρειάζονται αξιόπιστη μεταφορά.
- Οι ροές που μεταφέρουν μη αξιόπιστα δεδομένα μέσα σε μια σύνδεση SCTP μπορούν να τύχουν της ίδιας πρόβλεψης σε περίπτωση βλάβης του δικτύου, έχοντας τις ίδιες δυνατότητες με τις αξιόπιστες ροές, ώστε να καταφέρουν να επαναλειτουργήσουν την σύνδεση. Σε αυτές τις δυνατότητες περιλαμβάνεται και η γρήγορη αναγνώριση ενός προορισμού ο οποίος δεν μπορεί να λάβει πλέον δεδομένα και η εναλλαγή του δρομολογίου στον εναλλακτικό προορισμό. Επίσης υπάρχει η δυνατότητα ειδοποίησης εάν δεν μπορεί να υπάρξει επικοινωνία μεταξύ του αποστολέα και του παραλήπτη. Όλα αυτά τα εργαλεία επιτρέπουν σε κάποιο να κτίσει ένα σύστημα με μεγάλη ευρωστία σε εφαρμογές που μεταφέρουν μη αξιόπιστα δεδομένα.
- Με τις ροές μεταφοράς μη αξιόπιστων δεδομένων μια εφαρμογή μπορεί να ελέγξει τις δικές της επαναληπτικές αποστολές χαμένων πακέτων έτσι ώστε να στέλνει ξανά το ίδιο πακέτο μόνο ορισμένες φορές και όχι μέχρι αυτό να φτάσει στον τελικό του προορισμό.

- Εκτός από το ότι μπορεί να μεταφέρει τα μη αξιόπιστα δεδομένα εκτός σειράς όπως ακριβώς κάνει και το UDP, το SCTP μπορεί να παρέχει και παράδοση μη αξιόπιστων δεδομένων στην σειρά.
- Η μέθοδος μη αξιόπιστης μεταφοράς δεδομένων περιέχει τους ίδιους μηχανισμούς ελέγχου συμφόρησης και αποφυγής συμφόρησης σε μη αξιόπιστες ροές, όπως ακριβώς και στις αξιόπιστες ροές. Αυτό το γεγονός είναι πολύ επιθυμητό αφού είναι πιο φιλικό προς το δίκτυο από ότι το UDP.
- Λαμβάνοντας υπόψη το πλεονέκτημα του SCTP να στέλνει πακέτα από διαφορετικά streams στην ίδια σύνδεση (association), στέλνοντας πολλές ροές μη αξιόπιστων δεδομένων στην ίδια σύνδεση δημιουργείται ένας πολύ αποτελεσματικός και αποδοτικός τρόπος για πολυπλεξία (multiplexing) δεδομένων.

Με την επέκταση για μεταφορά μη αξιόπιστων δεδομένων ένας SCTP αποστολέας μπορεί να επιδείξει ένα υποσύνολο από τα εξερχόμενα streams ώστε να είναι μη αξιόπιστα. Τα πακέτα που στέλνονται από ένα μη αξιόπιστο stream θα τύχουν της ίδιας μεταχείρισης όσον αφορά τον έλεγχο συμφόρησης και τον έλεγχο για αποφυγή συμφόρησης και θα έχουν την ίδια προτεραιότητα αποστολής όπως τα πακέτα που έχουν σταλεί από ένα αξιόπιστο stream με την διαφορά όμως ότι δεν θα σταλούν ξανά (ή θα σταλούν μόνο ένα περιορισμένο αριθμό φορών) εάν θεωρηθούν χαμένα από τον παραλήπτη των πακέτων αυτών.

Κατά την αρχικοποίηση μιας σύνδεσης ο παραλήπτης περιλαμβάνει μια παράμετρο για να ενημερώσει τον αποστολέα ότι είναι σε θέση να υποστηρίξει μη αξιόπιστες ροές δεδομένων και ο αποστολέας υποδεικνύει ποιες ροές είναι αυτές.

Μερικές φορές είναι επιθυμητό να δημιουργήσουμε διαφορετικές πολυμεσικές ροές και να τις στείλουμε σε μια σύνδεση. Ως επί το πλείστον για να στείλουμε τις ροές αυτές χρησιμοποιούμε μη αξιόπιστες ροές μεταφοράς. Με την μέθοδο μη αξιόπιστης μεταφοράς δεδομένων του SCTP αυτό μπορεί να γίνει με το να αναθέσουμε κάθε διαφορετική πολυμεσική ροή σε μια διαφορετική ροή SCTP, και να αφήσουμε τον

μηχανισμό ένωσης (bundling) να εκτελέσει πολυπλεξία (multiplexing) στον αποστολέα και αποπολυπλεξία (demultiplexing) στον παραλήπτη.

Όταν ο παραλήπτης των δεδομένων είναι multi-homed, η μη αξιόπιστη μεταφορά δεδομένων που χρησιμοποιεί SCTP θα έχει τα ίδια οφέλη σε περίπτωση σφάλματος όπως αυτά που έχει η αξιόπιστη μεταφορά δεδομένων. Αυτό οφείλεται στο ότι ο αποστολέας εξακολουθεί να ακολουθεί τους ίδιους κανόνες σε περίπτωση σφάλματος και ακόμη μετρά το σύνολο των επαναλαμβανόμενων αποστολών πακέτων κατά μήκος όλης της σύνδεσης για τον συγκεκριμένο αρχικό προορισμό. Επομένως όταν προκύψει ένα σφάλμα στο δίκτυο, ο αποστολέας θα εντοπίσει το σφάλμα αυτό, και θα μεταφέρει την μεταφορά μη αξιόπιστων ροών σε μια εναλλακτική διεύθυνση προορισμού, ακολουθώντας τις ίδιες διεργασίες που γίνονται και για την μεταφορά αξιόπιστων δεδομένων.

Τέλος, η ανίχνευση χαμένων δεδομένων σε μια μη αξιόπιστη ροή είναι χρήσιμη για κάποιες εφαρμογές. Με την χρήση της μεθόδου για μεταφορά μη αξιόπιστων ροών του SCTP αυτό γίνεται δυνατό. Το πάνω στρώμα (upper layer) απλά χρειάζεται να εξετάσει τον αύξοντα αριθμό των πακέτων που έφτασαν στον προορισμό τους και ανήκουν στο ίδιο stream ώστε να εντοπίσει τα χαμένα πακέτα. Σημειώνεται ότι η ανίχνευση αυτή μπορεί να επιτευχθεί μόνο εάν έχουμε δηλώσει ότι τα πακέτα που ανήκουν στο συγκεκριμένο stream θα στέλνονται σε σειρά. (δηλαδή η σημαία σειράς – order flag - στην επικεφαλίδα θα πρέπει να είναι αρχικά δηλωμένη 0).

2.5 Real-time Transport Protocol (RTP)

Για την αποστολή πολυμεσικών πακέτων ο αποστολέας προσθέτει επικεφαλίδες στα πακέτα αυτά που περιλαμβάνουν αύξοντες αριθμούς (sequence numbers) και σφραγίδες χρόνου (timestamps). Επειδή οι περισσότερες πολυμεσικές εφαρμογές κάνουν χρήση των μηχανισμών αυτών, είναι απαραίτητη η ύπαρξη μιας προκαθορισμένης μορφής πακέτου που να περιλαμβάνει πεδία για πολυμεσικές εφαρμογές όπως αυτά που αναφέρθηκαν πιο πάνω καθώς και άλλα χρήσιμα πεδία. Το RTP [1] είναι ένα τέτοιο πρότυπο.

Το RTP μπορεί να χρησιμοποιηθεί για να μεταφέρει κοινές μορφές όπως το PCM, το GSM και το MP3 όσον αφορά τον ήχο, και MPEG, H.263 όσο αφορά το βίντεο. Σήμερα το RTP χρησιμοποιείται σε ευρεία κλίμακα σε εκατοντάδες προϊόντα και πρωτότυπα. Είναι επίσης συμβατό και με άλλα πολυμεσικά πρωτόκολλα όπως το SIP και το H.323.

Συνήθως το RTP τρέχει πάνω από το UDP [18]. Ο αποστολέας ενσωματώνει ένα τεμάχιο (chunk) μέσα σε ένα RTP πακέτο, μετά ενσωματώνει το πακέτο αυτό σε ένα UDP πακέτο και στέλνει το τελικό πακέτο στο IP. Ο παραλήπτης αφαιρεί το RTP πακέτο από το UDP πακέτο και το στέλνει στον πολυμεσικό αναπαραγωγέα για αποκωδικοποίηση. Για παράδειγμα υποθέτουμε ότι το RTP χρησιμοποιείται για να μεταφέρει ήχο με την PCM κωδικοποίηση στα 64Kbps και μαζεύουμε τα πακέτα κάθε 20 ms ώστε να περιέχουν 160 bytes δεδομένων.

Η πλευρά αποστολής τοποθετεί μια επικεφαλίδα RTP που περιλαμβάνει τον τύπο ακουστικής κωδικοποίησης, ένα αριθμό ακολουθίας, και μια σφραγίδα χρόνου. Η επικεφαλίδα RTP είναι κανονικά 12 ψηφιολέξεις. Το τεμάχιο ήχου μαζί με την επικεφαλίδα RTP διαμορφώνει το πακέτο RTP. Το πακέτο RTP στέλνεται έπειτα στη διεπαφή υποδοχών (socket) UDP. Στην πλευρά παραλήπτη, η εφαρμογή λαμβάνει το πακέτο RTP από τη διεπαφή υποδοχών της. Η εφαρμογή εξάγει το ακουστικό τεμάχιο από το πακέτο RTP και χρησιμοποιεί τις επικεφαλίδες του πακέτου RTP για να αποκωδικοποιήσει κατάλληλα και να αναπαράξει το ακουστικό τεμάχιο.

Εάν μια εφαρμογή ενσωματώσει RTP, θα επικοινωνήσει ευκολότερα με άλλες δικτυωμένες εφαρμογές πολυμέσων. Παραδείγματος χάριν, εάν δύο διαφορετικές επιχειρήσεις αναπτύξουν το τηλεφωνικό λογισμικό Διαδικτύου και οι δύο ενσωματώνουν RTP στο προϊόν τους, μπορεί να υπάρξει κάποια ελπίδα ότι ένας χρήστης που χρησιμοποιεί ένα από τα τηλεφωνικά προϊόντα Διαδικτύου θα είναι σε θέση να επικοινωνήσει με έναν χρήστη χρησιμοποιώντας το άλλο τηλεφωνικό προϊόν Διαδικτύου.

Πρέπει να σημειωθεί ότι το RTP από μόνο του δεν παρέχει οποιοδήποτε μηχανισμό για να εξασφαλίσει έγκαιρη παράδοση των στοιχείων ή να παρέχει άλλες εγγυήσεις

ποιότητας. Δεν εγγυάται ακόμη την παράδοση των πακέτων ή δεν αποτρέπει την παράδοση εκτός σειράς των πακέτων. Η ενθυσιαστική του RTP φαίνεται μόνο στα άκρα των συστημάτων. Οι δρομολογητές δεν ξεχωρίζουν μεταξύ των διαγραμμάτων δεδομένων IP που κουβαλούν τα πακέτα RTP και τα διαγράμματα δεδομένων IP που δεν κουβαλούν τέτοια πακέτα. Το RTP επιτρέπει σε κάθε πηγή (παραδείγματος χάριν, μια φωτογραφική μηχανή ή ένα μικρόφωνο) να ανατεθεί το δικό της RTP ρεύμα μεταφοράς δεδομένων. Παραδείγματος χάριν, για μια τηλεοπτική διάσκεψη με δύο συμμετέχοντες, τέσσερα ρεύματα RTP θα μπορούσαν να είναι ανοιχτά δηλαδή δύο ρεύματα για τον ήχο (ένα σε κάθε κατεύθυνση) και δύο ρεύματα για τη διαβίβαση του βίντεο (πάλι ένα σε κάθε κατεύθυνση). Εντούτοις, πολλές δημοφιλείς τεχνικές κωδικοποίησης – συμπεριλαμβανομένων του MPEG 1 και MPEG 2 – κωδικοποιούν το ακουστικό και τηλεοπτικό σήμα σε ένα ενιαίο ρεύμα [19]. Όταν το ακουστικό σήμα και το βίντεο συσσωρεύονται από τον κωδικοποιητή, κατόπιν μόνο ένα ρεύμα RTP παράγεται σε κάθε κατεύθυνση. Τα πακέτα RTP δεν περιορίζονται σε εφαρμογές unicast. Μπορούν επίσης να σταλούν και σε πολλαπλά δέντρα διανομής.

Οι τέσσερις κύριοι τομείς επιγραφών πακέτων RTP είναι ο τύπος ωφέλιμων φορτίων, ο αριθμός της ακολουθίας, οι χρονοσφραγίδες, και οι τομείς προσδιοριστικών πηγής. Ο τομέας ωφέλιμου φορτίου στο πακέτο RTP είναι επτά bit. Για ένα ακουστικό ρεύμα, ο τομέας ωφέλιμου φορτίου χρησιμοποιείται για να δείξει τον τύπο ακουστικής κωδικοποίησης (PCM, διαμόρφωση προσαρμοστικών δέλτα, γραμμική κωδικοποίηση) που χρησιμοποιείται. Εάν ένας αποστολέας αποφασίζει να αλλάξει την κωδικοποίηση στη μέση μιας συνόδου (session), ο αποστολέας μπορεί να ενημερώσει το δέκτη για την αλλαγή μέσω αυτού του πεδίου. Ο αποστολέας μπορεί να θελήσει να αλλάξει την κωδικοποίηση προκειμένου να αυξηθεί η ακουστική ποιότητα. Για ένα τηλεοπτικό ρεύμα, ο τύπος ωφέλιμων φορτίων χρησιμοποιείται για να δείξει τον τύπο της κωδικοποίησης (παραδείγματος χάριν, κίνηση JPEG, MPEG I, MPEG 2, H.261). Πάλι, ο αποστολέας μπορεί να αλλάξει την τηλεοπτική κωδικοποίηση κατά τη διάρκεια μιας συνόδου. Οι άλλοι σημαντικοί τομείς είναι οι ακόλουθοι :

Τομέας αριθμού ακολουθίας: Ο τομέας αριθμού ακολουθίας είναι 16 bit. Οι αυξήσεις αριθμού ακολουθίας αυξάνονται κατά ένα για κάθε πακέτο RTP που στέλνεται, και μπορούν να χρησιμοποιηθούν από το δέκτη για να ανιχνεύσουν την απώλεια πακέτων

και για να αποκαταστήσουν την ακολουθία πακέτων. Παραδείγματος χάριν, εάν η πλευρά δεκτών της εφαρμογής λαμβάνει ένα ρεύμα των πακέτων RTP με ένα χάσμα με αριθμούς ακολουθίας 86 και 89, ο δέκτης ξέρει ότι τα πακέτα 87 και 88 λείπουν. Ο δέκτης μπορεί έπειτα να προσπαθήσει να κρύψει τα χαμένα στοιχεία.

Τομέας χρονοσφραγίδας : Ο τομέας χρονοσφραγίδας είναι 32 bit. Απεικονίζει τη δειγματοληψία της πρώτης ψηφιολέξης στο πακέτο στοιχείων RTP. Ο δέκτης μπορεί να χρησιμοποιήσει χρονοσφραγίδες προκειμένου να αφαιρεθεί το πακέτο που εισάγεται στο δίκτυο αλλά φτάνει με καθυστέρηση ώστε να παρέχεται σύγχρονη αναπαραγωγή στο δέκτη.

Προσδιοριστικό πηγής συγχρονισμού (SSRC) : Ο τομέας SSRC είναι 32 bit. Αυτό προσδιορίζει τη πηγή του ρεύματος RTP. Χαρακτηριστικά, κάθε ρεύμα σε μια σύνοδο RTP έχει ένα ευδιάκριτο SSRC. Το SSRC δεν είναι η διεύθυνση IP του αποστολέα, αλλά είναι ένας αριθμός που η πηγή ορίζει τυχαία όποτε το νέο ρεύμα αρχίζει. Η πιθανότητα ότι δύο ρεύματα παίρνουν το ίδιο SSRC είναι πολύ μικρή. Εάν συμβεί αυτό, επιλέγεται μια νέα τιμή για το SSRC.

Το πρωτόκολλο ελέγχου RTP (RTCP)

Το RTCP, είναι ένα πρωτόκολλο που μια εφαρμογή δικτύωσης πολυμέσων μπορεί να χρησιμοποιήσει από κοινού με το RTP. Τα πακέτα RTCP διαβιβάζονται από κάθε συμμετέχοντα σε μια σύνοδο RTP σε όλους τους άλλους συμμετέχοντες στη σύνοδο χρησιμοποιώντας IP πολλαπλής διανομής. Για μια σύνοδο RTP, χαρακτηριστικά υπάρχει μια ενιαία διεύθυνση πολλαπλής διανομής και όλα τα πακέτα RTP και RTCP που ανήκουν στη σύνοδο χρησιμοποιούν την διεύθυνση αυτή. Τα πακέτα RTP και RTCP διακρίνονται το ένα από το άλλο μέσω της χρήσης διαφορετικών αριθμών υποδοχής (port numbers). (Ο αριθμός του RTCP τίθεται να είναι ίσος με τον αριθμό RTP συν ένα.) Τα πακέτα RTCP δεν περιέχουν τεμάχια ήχου ή εικόνας. Τα πακέτα RTCP στέλνονται περιοδικά και περιέχουν πληροφορίες και στατιστικές που μπορούν να είναι χρήσιμες στην εφαρμογή. Αυτές οι στατιστικές περιλαμβάνουν τον αριθμό των πακέτων που στέλνονται, τον αριθμό πακέτων που χάνονται, και του interarrival jitter. Η προδιαγραφή RTP [RFC 1889] δεν υπαγορεύει στην εφαρμογή τι πρέπει να κάνει με

αυτές τις πληροφορίες ανατροφοδότησης, αυτό εξαρτάται από τον υπεύθυνο για την ανάπτυξη της εφαρμογής. Οι αποστολείς μπορούν να χρησιμοποιήσουν τις πληροφορίες ανατροφοδότησης, παραδείγματος χάριν, για να τροποποιήσουν τα ποσοστά μετάδοσής τους. Οι πληροφορίες ανατροφοδότησης μπορούν επίσης να χρησιμοποιηθούν για διαγνωστικούς λόγους παραδείγματος χάριν, οι δέκτες μπορούν να καθορίσουν εάν τα προβλήματα είναι τοπικά, περιφερειακά, ή σφαιρικά.

Για κάθε ρεύμα RTP που ένας δέκτης λαμβάνει ως τμήμα μιας συνόδου, ο δέκτης παράγει μια έκθεση υποδοχής. Ο δέκτης αθροίζει τις εκθέσεις υποδοχής του σε ένα ενιαίο πακέτο RTCP. Το πακέτο στέλνεται έπειτα στο δέντρο που συνδέει όλους τους συμμετέχοντες της σύνδεσης. Η έκθεση υποδοχής περιλαμβάνει διάφορα πεδία, τα σημαντικότερα των οποίων είναι:

- Το SSRC του ρεύματος RTP για το οποίο παράγεται η έκθεση υποδοχής.
- Το ποσοστό των πακέτων που χάθηκε στο ρεύμα RTP. Κάθε δέκτης υπολογίζει τον αριθμό πακέτων RTP που χάνονται και διαιρούνται με τον αριθμό πακέτων RTP που στέλνονται ως τμήμα του ρεύματος. Εάν ένας αποστολέας λαμβάνει εκθέσεις υποδοχής που δείχνουν ότι οι παραλήπτες λαμβάνουν μόνο ένα μικρό μέρος των διαβιβασθέντων πακέτων του αποστολέα, μπορεί να μεταπηδήσει σε μια χαμηλότερη κωδικοποίηση με στόχο τη μείωση της συμφόρησης του δικτύου
- Ο τελευταίος αριθμός ακολουθίας που παραλαμβάνεται στο ρεύμα των πακέτων RTP.

Για κάθε ρεύμα RTP που ένας αποστολέας διαβιβάζει, ο αποστολέας δημιουργεί και στέλνει RTCP πακέτα εκθέσεων. Αυτά τα πακέτα περιλαμβάνουν τις πληροφορίες για το ρεύμα RTP, που περιλαμβάνει:

- Το SSRC του ρεύματος RTP
- Τη χρονοσφραγίδα και το χρόνο του πρόσφατα παραχθέντος πακέτου RTP
- Τον αριθμό των πακέτων που στέλνονται στο ρεύμα
- Τον αριθμό ψηφιολέξεων που στέλνονται στο ρεύμα

Οι εκθέσεις αποστολών μπορούν να χρησιμοποιηθούν για να συγχρονίσουν τα διαφορετικά ρεύματα μέσων, σε μια σύνοδο RTP. Ως παράδειγμα εξετάζουμε μια τηλεοπτική εφαρμογή σύσκεψης για την οποία κάθε αποστολέας παράγει δύο ανεξάρτητα ρεύματα RTP, ένα για το βίντεο και ένα για τον ήχο. Κάθε έκθεση αποστολών RTCP περιέχει, για το πρόσφατα παραχθέν πακέτο στο σχετικό ρεύμα RTP, μια χρονοσφραγίδα του πακέτου RTP όταν το πακέτο δημιουργήθηκε. Οι δέκτες μπορούν να χρησιμοποιήσουν αυτήν την σφραγίδα στις εκθέσεις αποστολών RTCP για να συγχρονίσουν την αναπαραγωγή της εικόνας και του ήχου.

Για κάθε ρεύμα RTP που ένας αποστολέας διαβιβάζει, ο αποστολέας δημιουργεί επίσης και διαβιβάζει τα πακέτα περιγραφής πηγής. Αυτά τα πακέτα περιέχουν τις πληροφορίες για την πηγή, όπως η διεύθυνση ηλεκτρονικού ταχυδρομείου του αποστολέα, του ονόματος του αποστολέα, και της εφαρμογής για την οποία παράγεται το ρεύμα RTP. Περιλαμβάνει επίσης το SSRC του σχετικού ρεύματος RTP. Αυτά τα πακέτα παρέχουν μια χαρτογράφηση μεταξύ του προσδιοριστικού πηγής (δηλαδή το SSRC) και του ονόματος χρηστών.

Τα πακέτα RTCP είναι ευέλικτα με την έννοια ότι οι εκθέσεις υποδοχής δεκτών, οι εκθέσεις αποστολών, και οι περιγραφές πηγής μπορούν να συνδεθούν σε ένα ενιαίο πακέτο. Το πακέτο που προκύπτει τοποθετείται σε ένα πακέτο UDP και μεταδίδεται στο δέντρο πολλαπλής διανομής.

Το RTCP έχει ένα πιθανό πρόβλημα στο εύρος ζώνης. Για παράδειγμα έχουμε, μια σύνοδο RTP που αποτελείται από έναν αποστολέα και έναν μεγάλο αριθμό δεκτών. Εάν κάθε ένας από τους δέκτες παράγει περιοδικά RTCP πακέτα, το συνολικό ποσοστό μετάδοσης πακέτων RTCP μπορεί πολύ να υπερβεί το ποσοστό πακέτων RTP που στέλνονται από τον αποστολέα. Το ποσοστό κυκλοφορίας RTP πακέτων που στέλνεται στο δέντρο πολλαπλής διανομής δεν αλλάζει καθώς ο αριθμός των δεκτών αυξάνεται, ενώ το ποσοστό κυκλοφορίας RTCP πακέτων αυξάνεται γραμμικά με τον αριθμό δεκτών.

Για να λύσει αυτό το πρόβλημα το RTCP τροποποιεί το ποσοστό στο οποίο ένας συμμετέχων στέλνει τα πακέτα RTCP στο δέντρο πολλαπλής διανομής ως λειτουργία

του αριθμού των συμμετεχόντων στη σύνοδο. Επίσης, δεδομένου ότι κάθε συμμετέχων στέλνει τα πακέτα ελέγχου σε όλους τους άλλους, κάθε συμμετέχων μπορεί να υπολογίσει το συνολικό αριθμό συμμετεχόντων στη σύνοδο [Friedman 1999]. Το RTCP προσπαθεί να περιορίσει την κυκλοφορία του σε 5 τοις εκατό του εύρους ζώνης της συνόδου.

ΚΕΦΑΛΑΙΟ 3

Περιγραφή Πειραμάτων και Παρουσίαση Αποτελεσμάτων

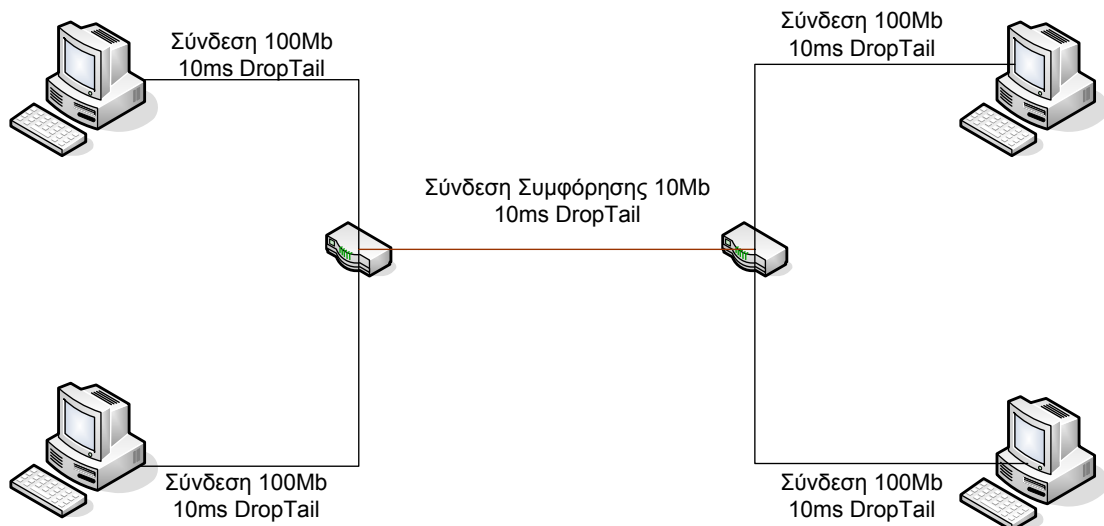
3.1 Ανάλυση Τοπολογίας – Μετρικές	39
3.2 Προσομοιώσεις με χρήση του TFRC	41
3.3 Προσομοιώσεις με χρήση του SCTP	59
3.4 Προσομοιώσεις με χρήση του RTP	76
3.5 Προσομοιώσεις με χρήση του UDP	93

3.1 Ανάλυση Τοπολογίας - Μετρικές

Για την δημιουργία και υλοποίηση του δικτύου μας, χρησιμοποιήσαμε τον προσομοιωτή δικτύων Network Simulator v.2.28 [6]. Η βασική τοπολογία παρουσιάζεται στο σχήμα 3.0.1. Αποτελείται από έξι κόμβους με αρίθμηση από το μηδέν μέχρι το πέντε. Η ιδέα είναι ότι θα χρησιμοποιήσουμε τους κόμβους 0, 2, 3 και 4 για την αποστολή πακέτων ενός πρωτοκόλλου και τους κόμβους 1, 2, 3 και 5 για την αποστολή πακέτων ενός άλλου πρωτοκόλλου.

Συνδέουμε τους κόμβους 0-2, 3-4, 1-2 και 3-5 μεταξύ τους με σύνδεση εύρους 100Mb και καθυστέρηση 10ms και τους κόμβους 2-3 με σύνδεση 10Mb και καθυστέρηση 10ms. Η σύνδεση αυτή είναι μικρότερου εύρους με σκοπό να παρατηρήσουμε την συμπεριφορά των πρωτοκόλλων σε επίπεδα ψηλής συμφόρησης. Στην θέση του ενός πρωτοκόλλου κάνουμε χρήση του TCP ώστε τα αποτελέσματα που θα προκύψουν να μπορούν να συγκριθούν μεταξύ τους.

Οι μετρικές που χρησιμοποιούνται στα πειράματα είναι η χρήση της σύνδεσης που συνδέει τον αποστολέα με τον παραλήπτη, τα χαμένα πακέτα που παρατηρούνται λόγω της συμφόρησης στο δίκτυο, η καθυστέρηση των πακέτων που παρατηρείται από την στιγμή που θα δημιουργηθούν μέχρι την ώρα που θα φτάσουν στον παραλήπτη και η καθυστέρηση των πακέτων που παρατηρείται στους δρομολογητές της σύνδεσης με υψηλή συμφόρηση.



Σχήμα 3.0.1 : Βασική Τοπολογία Προσομοιώσεων

Για την προσομοίωση περισσότερων συνδέσεων κάθε πρωτοκόλλου στο δίκτυο μας αντί να χρησιμοποιήσουμε περισσότερους κόμβους, χρησιμοποιούμε περισσότερες ροές που διασχίζουν την ίδια σύνδεση. Οι διάφορες προσομοιώσεις έγιναν με διαφορετικό αριθμό ροών σε κάθε περίπτωση. Επίσης έγιναν μετρήσεις για διαφορετικό τύπο ροών στο δίκτυο. Πιο κάτω αναλύουμε τις διάφορες μετρήσεις που έγιναν για καθένα από τα πρωτόκολλα που περιγράφηκαν στο κεφάλαιο 1.

Σε όλες τις προσομοιώσεις συγκρίνουμε το κάθε πρωτόκολλο με το TCP. Δημιουργούμε πράκτορες τύπου TCP/SACK1 και αλλάζουμε το μέγεθος του πακέτου το οποίο ορίζουμε σταθερό στις 1460 ψηφιολέξεις. Το TCP κάνει χρήση πακέτων από εφαρμογές FTP ώστε η παράδοση των δεδομένων να είναι αξιόπιστη.

3.2 Προσομοιώσεις με χρήση του TFRC

Όπως έχει αναφερθεί και προηγουμένως το πρωτόκολλο αυτό έγινε για να είναι δίκαιο όταν συναγωνίζεται ροές TCP με μέγεθος πακέτων 1460 ψηφιολέξεις. Στα πειράματά μας προσομοιώνουμε ισάριθμες ροές από τα δύο πρωτόκολλα.

Χρησιμοποιώντας την βασική τοπολογία που έχουμε περιγράψει στέλνουμε τις ροές TCP στο μονοπάτι που αποτελείται από τους κόμβους 0, 2, 3, 4 και τις ροές TFRC από το μονοπάτι που αποτελείται από τους κόμβους 1, 2, 3, 5. Οι μετρήσεις γίνονται με χρήση 10, 25, 50, 75, 100, 150, 200 και 250 ροών από το κάθε πρωτόκολλο. Οι προσομοιώσεις διαρκούν ένα σταθερό χρόνο που έχουμε ορίσει στα 100 δευτερόλεπτα.

Για να προσομοιώσουμε όσο δυνατό καλύτερο το θεωρητικό μοντέλο έχουμε τροποποιήσει μερικές από τις προκαθορισμένες τιμές των ροών του TFRC. Πιο συγκεκριμένα έχουμε μετατρέψει την παράμετρο SndrType από 1 σε 0 η οποία υποδεικνύει ότι το TFRC θα μεταφέρει πακέτα που περιέχουν πολυμεσικό περιεχόμενο. Έχουμε θέσει την τιμή VoIP από 0 σε 1 η οποία καθορίζει πως ο τύπος των πακέτων που θα στέλνονται θα είναι πολυμεσικά. Ο αριθμός των πακέτων που θα στέλνονται κάθε δευτερόλεπτο είναι αρχικά προκαθορισμένος στα 100 πακέτα.

Έπειτα δημιουργούμε πακέτα για την εφαρμογή που θα χρησιμοποιήσει το TFRC. Το μέγεθος των πακέτων αυτών έχει καθοριστεί στις 200 ψηφιολέξεις και ορίσαμε πως τα πακέτα αυτά θα στέλνονται κάθε ένα μεταβλητό αριθμό δευτερολέπτων.

Οι μετρήσεις μας έγιναν για 10ms και για 5ms. Επίσης έγιναν προσομοιώσεις με χρήση πραγματικής ταινίας. Αν για παράδειγμα κάνουμε χρήση πακέτων με μέγεθος 200 ψηφιολέξεις και χρόνο μεταξύ μετάδοσης πακέτων 10ms τότε ο ρυθμός μετάδοσης που αναμένουμε για κάθε ροή TFRC θα είναι : $200 * 8 \text{ bits} / 0,01\text{ms} = 160\text{Kbps}$. Τα αποτελέσματα από τις διάφορες μετρήσεις τα αποθηκεύουμε σε αρχεία τα οποία επεξεργαζόμαστε στην συνέχεια.

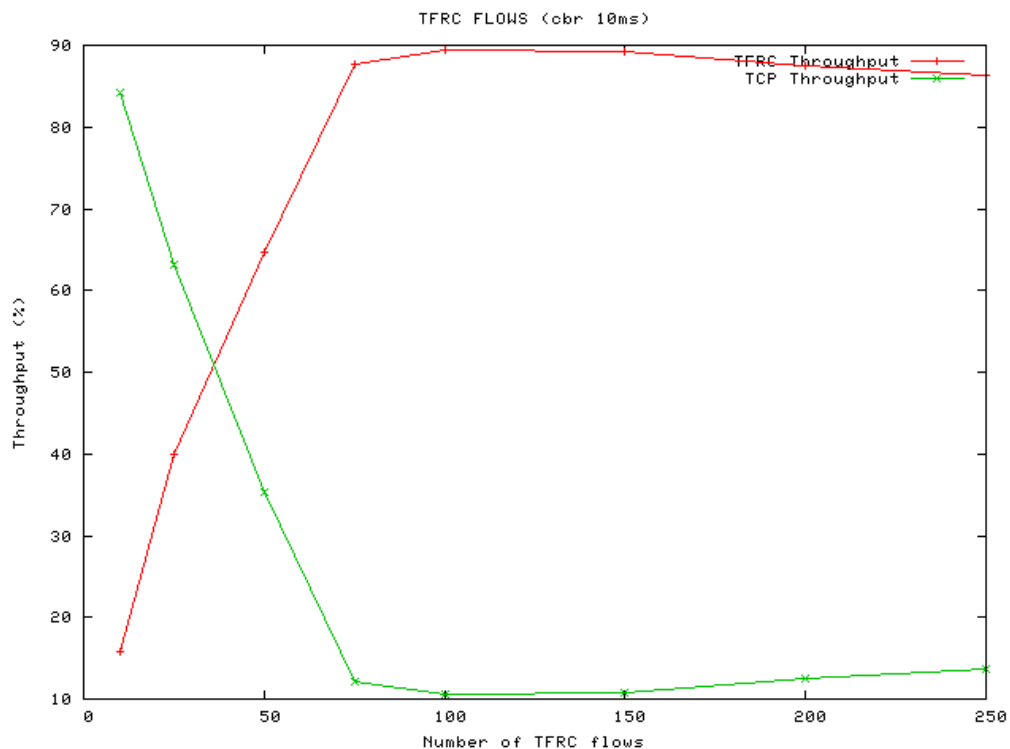
Η πρώτη προσομοίωση αφορά το ποσοστό χρήσης της σύνδεσης από τα δύο πρωτόκολλα στα άκρα. Οι μετρήσεις έγιναν για διαφορετικό αριθμό ροών και τα

αποτελέσματα καταγράφονται στον πίνακα 3.1 και παρουσιάζονται γραφικά στο σχήμα 3.1. Τα πακέτα που χρησιμοποιεί το TFRC είναι τύπου CBR με διάστημα μεταξύ αποστολής τα 10ms. Επίσης ο πίνακας 3.1 παρουσιάζει το μέσο εύρος ζώνης της σύνδεσης για την κάθε ροή από το κάθε πρωτόκολλο.

Αριθμός Ροών	TCP Throughput %	TCP Throughput Mb	TFRC Throughput %	TFRC Throughput Mb
10	84,1529	0,844	15,7207	0,159
25	63,0721	0,260	39,9279	0,147
50	35,2679	0,060	64,6602	0,130
75	12,1605	0,017	87,7751	0,110
100	10,539	0,015	89,4064	0,087
150	10,6813	0,005	89,2823	0,055
200	12,4551	0,013	87,5174	0,0428

Πίνακας 3.1 : Ποσοστό χρήσης σύνδεσης στα άκρα και μέσος όρος εύρους ζώνης

Παρατηρούμε πως με την αύξηση του αριθμού των ροών από το κάθε πρωτόκολλο το TFRC τείνει να μηδενίσει το ρυθμό αποστολής του TCP. Αρχικά όλες οι ροές του TFRC περιορίζουν τον ρυθμό αποστολής τους στα 160Kb που είναι ακριβώς όσα χρειάζονται από την εφαρμογή.



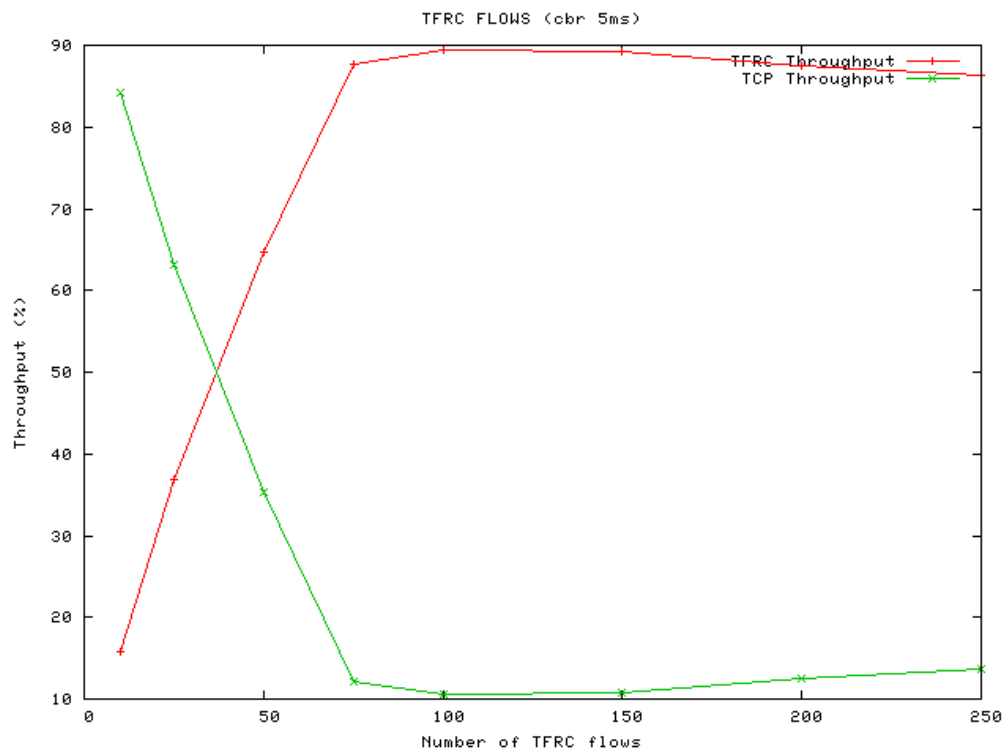
Σχήμα 3.1 : Ποσοστό χρήσης σύνδεσης συναρτήσει του αριθμού ροών TCP και TFRC

Αντίθετα το TCP τείνει να εκμεταλλευτεί όσο το δυνατό περισσότερο εύρος ζώνης γι' αυτό και οι ροές έχουν μεγαλύτερο ρυθμό. Με την αύξηση όμως των ροών και κατ' επέκταση την αύξηση της συμφόρησης, οι ροές TFRC δεν μπορούν να πάρουν τα επιθυμητά 160Kb και επομένως τείνουν να εξαλείφουν τις ροές TCP.

Επαναλαμβάνουμε την ίδια διαδικασία αλλά αυτή την φορά μειώνουμε τον χρόνο μεταξύ της κάθε αποστολής πακέτων της εφαρμογής στα 5ms. Οι μετρήσεις έγιναν για διαφορετικό αριθμό ροών και τα αποτελέσματα καταγράφονται στον πίνακα 3.2 και παρουσιάζονται γραφικά στο σχήμα 3.2. Επίσης ο πίνακας 3.2 παρουσιάζει το μέσο εύρος ζώνης της σύνδεσης για την κάθε ροή από το κάθε πρωτόκολλο.

Αριθμός Ροών	TCP Throughput %	TCP Throughput Mb	TFRC Throughput %	TFRC Throughput Mb
10	84,1529	0,843	15,7207	0,1579
25	63,0721	0,261	36,8469	0,1498
50	35,2679	0,049	64,6602	0,1344
75	12,1005	0,015	87,7761	0,1273
100	10,539	0,020	89,4064	0,1004
150	10,6813	0,006	89,2823	0,0516
200	12,4551	0,013	87,5174	0,0428

Πίνακας 3.2 : Ποσοστό χρήσης σύνδεσης στα άκρα και μέσος όρος εύρους ζώνης



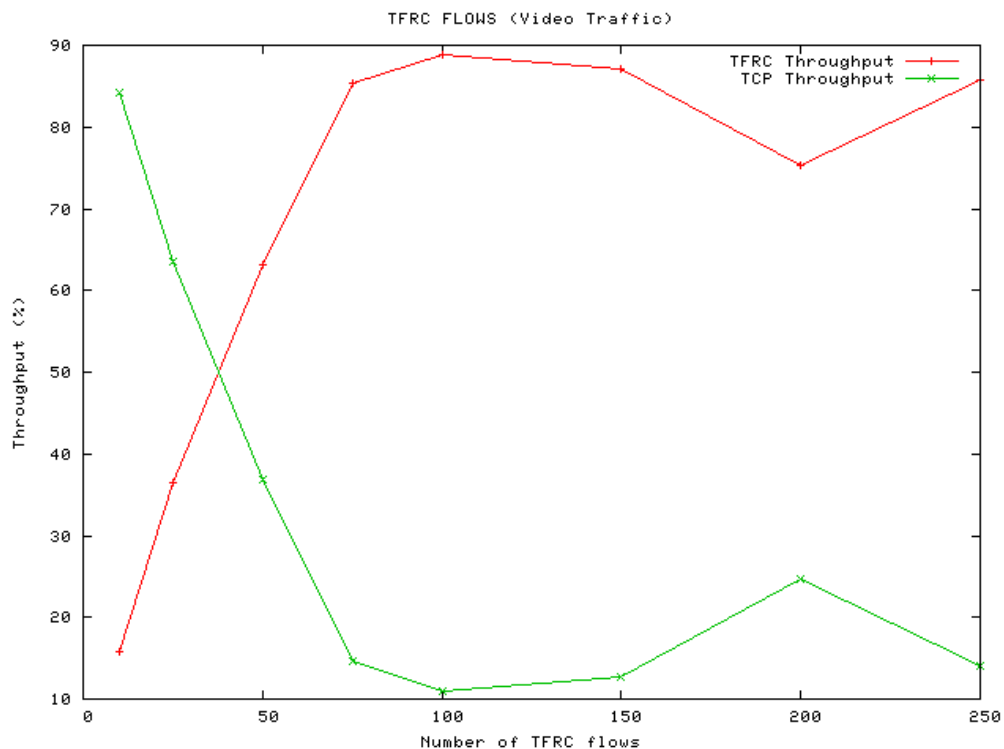
Σχήμα 3.2 : Ποσοστό χρήσης σύνδεσης συναρτήσει του αριθμού ροών TCP και TFRC

Παρατηρούμε όπως και στην προηγούμενη περίπτωση ότι οι ροές TFRC αρχικά καταλαμβάνουν μικρότερο εύρος ζώνης από αυτό που δικαιούνται, αλλά με την αύξηση των ροών και της συμφόρησης τείνουν να εξαλείψουν τις ροές TCP.

Τέλος δοκιμάζουμε την ίδια διαδικασία αλλά αντί για χρήση πακέτων εφαρμογής προσομοιώνουμε με την χρήση πραγματικού βίντεο. Το βίντεο που αρχικά είναι χωρισμένο σε διάφορα πλαίσια ταινίας μετατρέπεται σε πακέτα, με χρήση ενός μικρού προγράμματος το οποίο μετατρέπει τα πλαίσια αυτά σε πακέτα αποστολής σταθερού ρυθμού.

Αριθμός Ροών	TCP Throughput %	TCP Throughput Mb	TFRC Throughput %	TFRC Throughput Mb
10	84,1243	0,8436	15,7461	0,1573
25	63,5259	0,2614	36,3893	0,1373
50	36,8117	0,0654	63,1082	0,1243
75	14,6555	0,0267	85,2716	0,104
100	10,9951	0,0069	88,9361	0,0854
150	12,7951	0,0136	87,1477	0,0403
200	24,7261	0,0167	75,2745	0,05151

Πίνακας 3.3 : Ποσοστό χρήσης σύνδεσης στα άκρα και μέσος όρος εύρους ζώνης



Σχήμα 3.3 : Ποσοστό χρήσης σύνδεσης συναρτήσει του αριθμού ροών TCP και TFRC

Όπως αναφέρει ο προγραμματιστής του προγράμματος [8] αυτού η προσομοίωση θα είναι παρόμοια εάν είχαμε χρήση πακέτων με μέγεθος 200 ψηφιολέξεις που θα είχαν διαφορά χρόνου στην αποστολή τους 10ms. Τα αποτελέσματα της διαδικασίας παρουσιάζονται στον πίνακα 3.3 και η γραφική απεικόνιση στο σχήμα 3.3.

Όπως και στις προηγούμενες περιπτώσεις έτσι και εδώ παρατηρούμε πως η προσομοίωση με την χρήση πραγματικού βίντεο, δίνει περίπου τα ίδια αποτελέσματα με αυτά που παίρνουμε και στις άλλες περιπτώσεις. Συμπεραίνουμε τελικά πως στην ύπαρξη συμφόρησης οι ροές TFRC εκμεταλλεύονται το εύρος ζώνης της σύνδεσης σε βάρος του TCP.

Για την δεύτερη μας μέτρηση θα εξετάσουμε την καθυστέρηση που παρουσιάζουν τα πακέτα σε συνάρτηση με τον αριθμό των ροών του TFRC. Από τις προηγούμενες μετρήσεις παρατηρήσαμε πως οι ροές TFRC κερδίζουν σε εύρος ζώνης σε βάρος του TCP. Ένα λογικό σενάριο θα ήταν οι ροές TFRC να παρουσιάζουν λιγότερη καθυστέρηση στα άκρα ώστε να κερδίζουν σε εύρος ζώνης.

Πιο κάτω παρουσιάζουμε τα αποτελέσματα των μετρήσεων της καθυστέρησης σε συνάρτηση με τον αριθμό των ροών για τα δύο πρωτόκολλα. Αρχικά παρουσιάζουμε τα αποτελέσματα με την χρήση πακέτων εφαρμογής με μέγεθος 200 ψηφιολέξεις και χρόνο μεταξύ αποστολής τα 10ms.

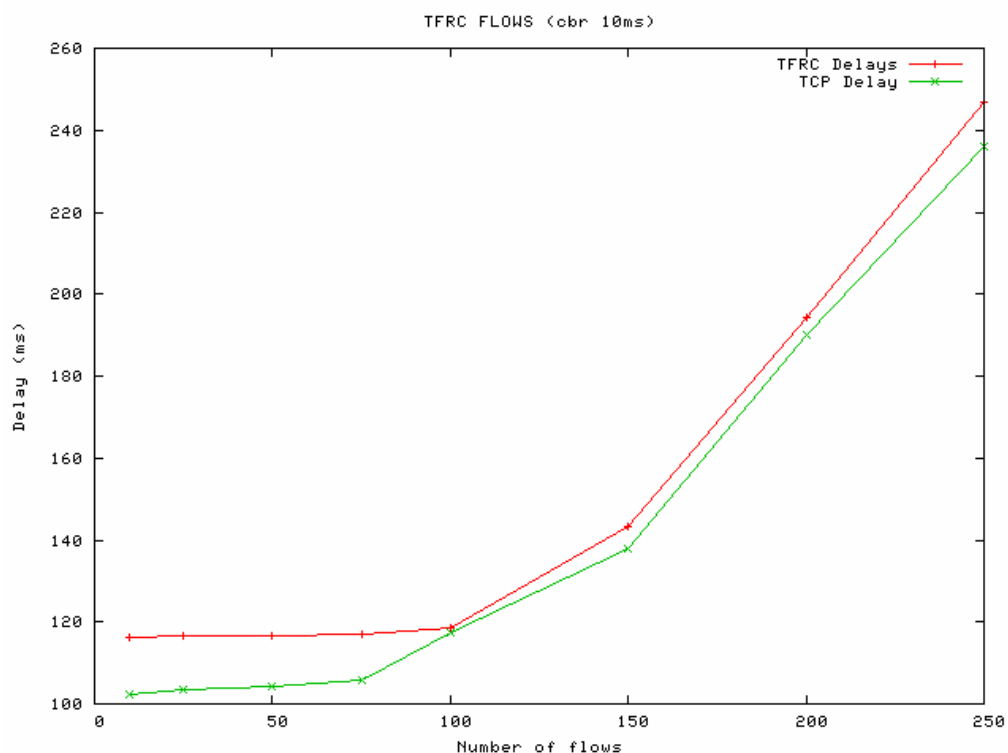
Αριθμός Ροών	TFRC Drop Rate %	Mean Delay TFRC	Mean Delay TCP	Jitter TFRC
10	0,25	115,123	113,589	9,68
25	3,575	116,593	118,648	23,11
50	16,43	116,547	118,651	14,42
75	19,436	117,084	120,654	15,46
100	26,988	118,428	121,264	15,43
150	37,01	143,271	142,485	17,75
200	38,134	194,395	190,268	16,40
250	42,23	246,75	242,248	21,56

Πίνακας 3.4 : Μέση καθυστέρηση πακέτων στα άκρα για το TFRC.

Τα αποτελέσματα παρουσιάζονται στον πίνακα 3.4 και γραφικά στο σχήμα 3.4. Επίσης ο πίνακας περιέχει το ποσοστό των πακέτων που χάνονται από την έξοδο της σύνδεσης

που παρουσιάζεται συμφόρηση μέχρι τον παραλήπτη, καθώς και τη διαφορά χρόνου στον οποίο κάποια πακέτα φτάνουν στον παραλήπτη μετά που ακολούθησαν διαφορετική διαδρομή (Jitter).

Παρατηρούμε πως με την αύξηση των ροών η καθυστέρηση που παρουσιάζουν οι ροές TFRC αυξάνονται δραματικά.. Οι ροές του TCP με την σειρά τους αυξάνουν την καθυστέρηση όσο αυξάνονται οι ροές επομένως και χάνουν εύρος ζώνης από το TFRC. Μέχρι ένα σημείο γύρω στις 75 ροές από το κάθε πρωτόκολλο, παρατηρούμε πως οι ροές TFRC έχουν σχεδόν τον ίδιο χρόνο καθυστέρησης των πακέτων.



Σχήμα 3.4 : Καθυστέρηση πακέτων σε συνάρτηση του αριθμού ροών TFRC.

Επαναλαμβάνουμε την ίδια διαδικασία αλλά αυτή την φορά κάνουμε χρήση πακέτων μεγέθους 200 ψηφιολέξεων με χρόνο μεταξύ αποστολής τα 5ms. Τα αποτελέσματα παρουσιάζονται στον πίνακα 3.5 και γραφικά στο σχήμα 3.5.

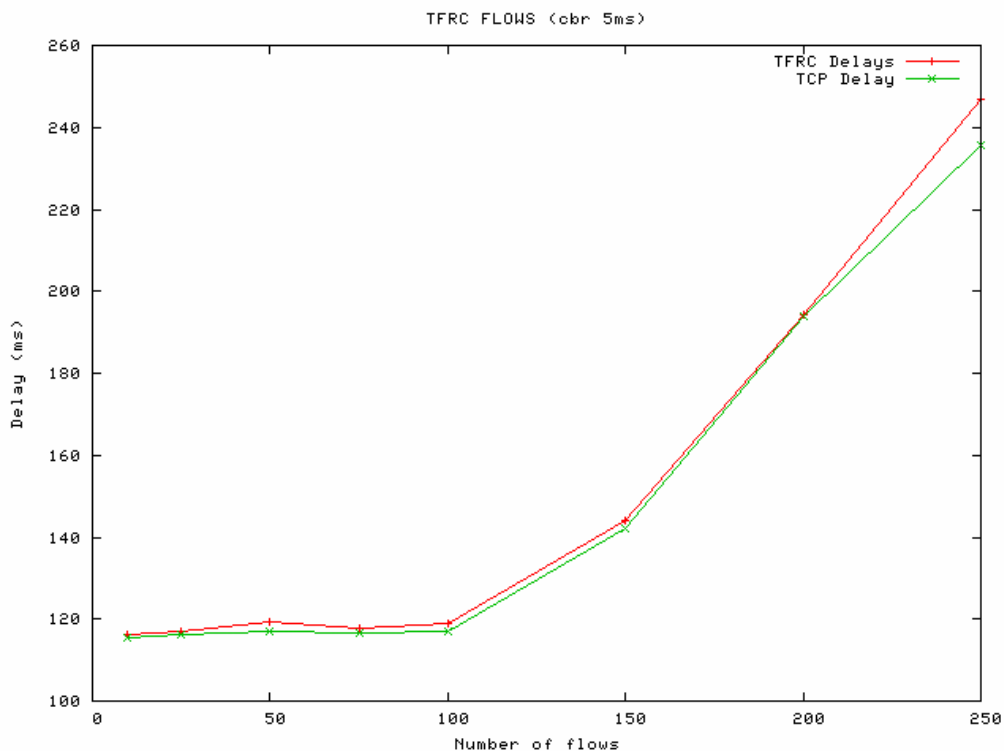
Επίσης ο πίνακας περιέχει το ποσοστό των πακέτων που χάνονται από την έξοδο της σύνδεσης που παρουσιάζεται συμφόρηση μέχρι τον παραλήπτη, καθώς και η διαφορά

χρόνου στον οποίο κάποια πακέτα φτάνουν στον παραλήπτη μετά που ακολούθησαν διαφορετική διαδρομή (Jitter).

Αριθμός Ροών	TFRC Drop Rate %	Mean Delay TFRC	Mean Delay TCP	Jitter TFRC
10	0,25	118,173	115,297	9,565
25	3,746	117,059	114,085	23,082
50	14,786	119,37	117,671	17,354
75	20,08	117,873	115,189	16,227
100	28,23	118,801	117,052	17,880
150	38,00	144,249	139,186	24,103
200	38,263	194,381	190,258	17,984
250	41,126	246,721	241,284	19,752

Πίνακας 3.5 : Μέση καθυστέρηση πακέτων στα άκρα για το TFRC.

Παρατηρούμε σε σύγκριση με τα προηγούμενα αποτελέσματα πως η καθυστέρηση κυμαίνεται στα ίδια περίπου επίπεδα και πως η αλλαγή στο χρόνο αποστολής μεταξύ των πακέτων δεν επιφέρει οποιαδήποτε ουσιαστική αλλαγή στα αποτελέσματα.

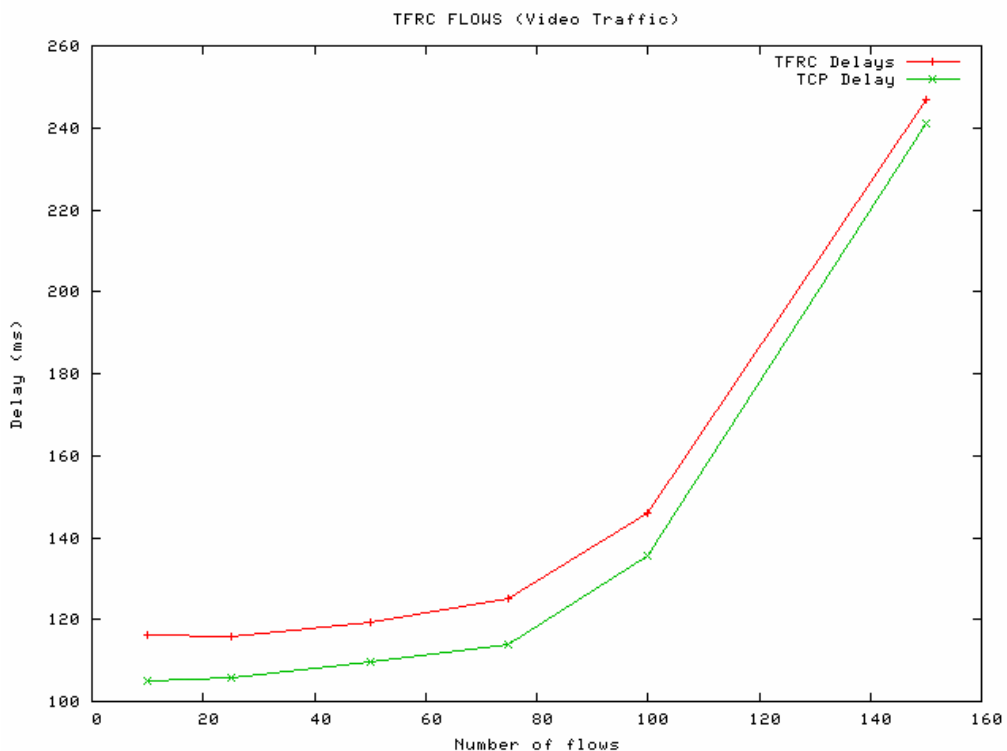


Σχήμα 3.5 : Καθυστέρηση πακέτων σε συνάρτηση του αριθμού ροών TFRC.

Τέλος προσομοιώνουμε με την χρήση αποστολής κανονικής ταινίας όπως ακριβώς και το προηγούμενο πείραμα. Τα αποτελέσματα παρουσιάζονται στον πίνακα 3.6 και γραφικά στο σχήμα 3.6. Επίσης ο πίνακας περιέχει το ποσοστό των πακέτων που χάνονται από την έξοδο της σύνδεσης που παρουσιάζεται συμφόρηση μέχρι τον παραλήπτη, καθώς και η διαφορά χρόνου στον οποίο κάποια πακέτα φτάνουν στον παραλήπτη μετά που ακολούθησαν διαφορετική διαδρομή (Jitter).

Αριθμός Ροών	TFRC Drop Rate %	Mean Delay TFRC	Mean Delay TCP	Jitter TFRC
10	0,25	116,095	114,085	8,082
25	19,156	115,665	115,486	21,080
50	3,116	119,492	114,286	17,151
75	19,02	125,279	123,096	18,618
100	28,436	146,001	143,132	14,404
150	35,816	247,009	425,128	19,726

Πίνακας 3.6 : Μέση καθυστέρηση πακέτων στα άκρα για το TFRC.



Σχήμα 3.6 : Καθυστέρηση πακέτων σε συνάρτηση του αριθμού ροών TFRC.

Συμπεραίνουμε πως η χρήση του βίντεο δεν προκαλεί μεγάλες αλλαγές στην καθυστέρηση που παρατηρείται στα πακέτα TFRC. Συγκρίνοντας τα αποτελέσματα των

μέχρι στιγμής πειραμάτων βλέπουμε πως επειδή η καθυστέρηση των πακέτων TFRC είναι μικρότερη από αυτή του TCP, οι ροές του TFRC παίρνουν μεγαλύτερο εύρος ζώνης.

Για τα επόμενα πειράματα χρησιμοποιούμε μετρικές τις οποίες λαμβάνουμε στην γραμμή από την οποία υπάρχει συμφόρηση. Το πρώτο τέτοιο πείραμα αφορούσε το ποσοστό της χρήσης της γραμμής στην οποία υπάρχει συμφόρηση σε συνάρτηση με τον αριθμό των ροών του κάθε πρωτοκόλλου.

Λόγω του ότι το εύρος ζώνης της σύνδεσης είναι 10 φορές μικρότερο από αυτό που ενώνει την έξοδο της σύνδεσης με τον παραλήπτη, δεν παρατηρείται χάσιμο πακέτων στην γραμμή αυτή. Επομένως τα αποτελέσματα αυτής της μέτρησης ταυτίζονται με τα αποτελέσματα του πειράματος που αφορά την χρήση της σύνδεσης στον παραλήπτη σε συνάρτηση με τον αριθμό των ροών. Σχηματικά τα αποτελέσματα είναι τα ίδια με αυτά των σχημάτων 3.1 μέχρι 3.3.

Το επόμενο πείραμα που αφορά την σύνδεση συμφόρησης αφορά το χάσιμο των πακέτων των διαφόρων ροών σε συνάρτηση με τον αριθμό των ροών. Θα περιμέναμε εδώ να δούμε πως οι ροές που έχουν το μεγαλύτερο εύρος ζώνης έχουν και το μεγαλύτερο ποσοστό χαμένων πακέτων.

Ένα πακέτο για να χαθεί θα πρέπει πρώτα να ριχτεί από τους δρομολογητές που βρίσκονται στο μονοπάτι που υπάρχει συμφόρηση. Συγκεκριμένα όταν φτάσουν πακέτα στους δρομολογητές αυτοί τα αποθηκεύουν σε ένα αποθηκευτικό χώρο και αφού εκτελέσουν κάποιες εργασίες πάνω τους, τα προωθούν στο επόμενο κομμάτι του μονοπατιού.

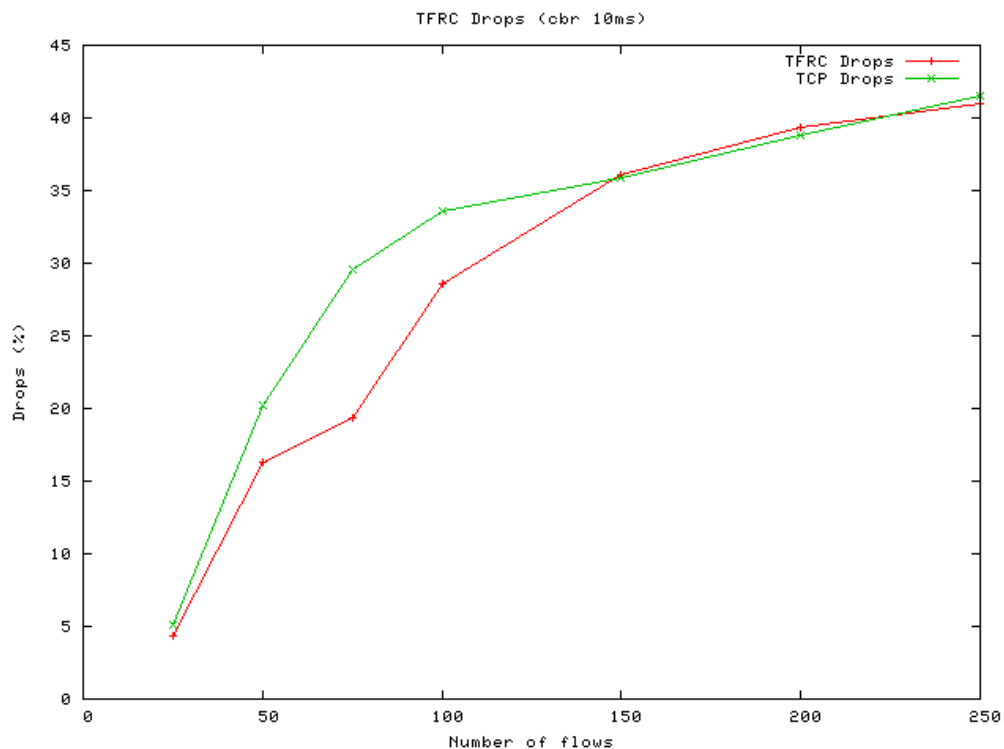
Εάν αυτός ο αποθηκευτικός χώρος γεμίσει τότε τα επόμενα πακέτα που θα φτάσουν στον συγκεκριμένο δρομολογητή, δεν θα έχουν χώρο να αποθηκευτούν και επομένως θα ριχτούν. Στο σημερινό διαδίκτυο όλα τα πακέτα έχουν την ίδια πιθανότητα να ριχτούν από τον δρομολογητή, ασχέτως από το μέγεθος τους.

Στο πρώτο πείραμα χρησιμοποιούμε πακέτα εφαρμογής μεγέθους 200 ψηφιολέξεων και χρόνο μεταξύ αποστολής των πακέτων 10ms. Τα αποτελέσματα παρουσιάζονται στον πίνακα 3.7 και γραφικά στο σχήμα 3.7.

Αριθμός Ροών	TCP Drop Rate %	TFRC Drop Rate %
25	5,064	4,325
50	20,2161	16,3095
75	29,558	19,3537
100	33,5809	28,6263
150	35,8794	36,0574
200	38,857	39,3621
250	41,4873	41,014

Πίνακας 3.7 : Ποσοστό χαμένων πακέτων στη σύνδεση συμφόρησης.

Παρατηρούμε πως στα αρχικά επίπεδα συμφόρησης οι ροές TFRC έχουν μικρότερο ποσοστό πακέτων που έχουν ριχτεί από αυτό του TCP. Όσο όμως αυξάνεται ο αριθμός των ροών το ποσοστό αυτό συγκλίνει. Έτσι παρατηρούμε πως σε ένα αριθμό ροών μεγαλύτερο του 150, το ποσοστό των πακέτων που έχει ριχτεί και για τα δύο πρωτόκολλα είναι σχεδόν ο ίδιος. Αυτό μπορεί να δικαιολογηθεί από το ότι τα δύο πρωτόκολλα έχουν σχεδιαστεί για να συμπεριφέρονται δίκαια μεταξύ τους.



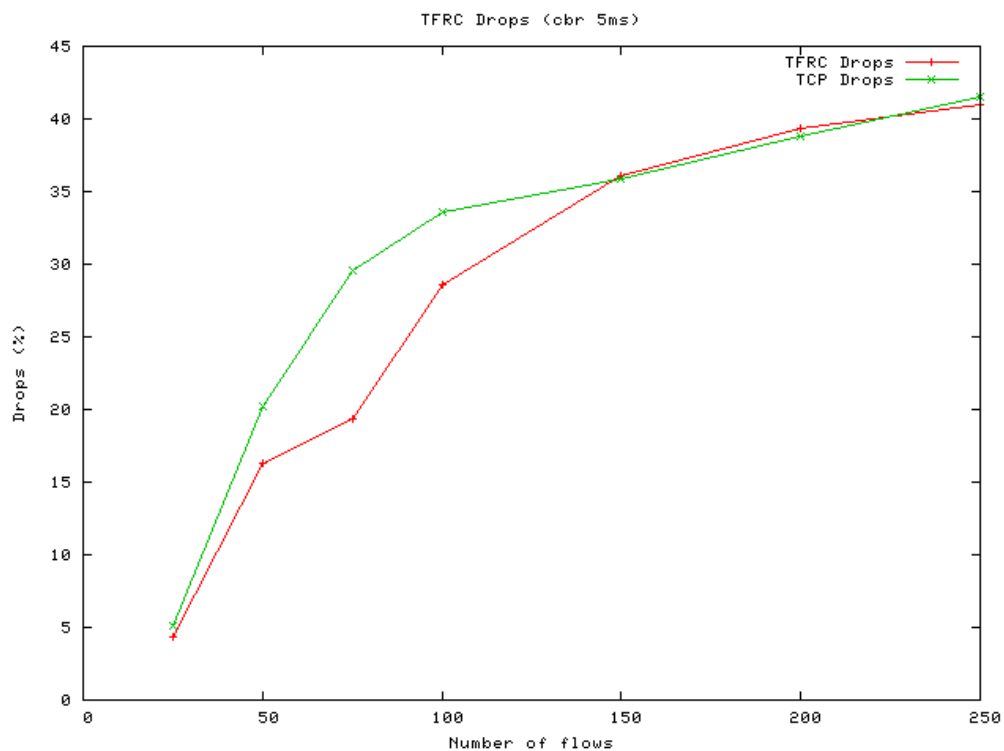
Σχήμα 3.7 : Ποσοστό χαμένων πακέτων στη σύνδεση συμφόρησης σε συνάρτηση ροών.

Επαναλαμβάνουμε το ίδιο πείραμα όπως και στην προηγούμενη περίπτωση αλλά αυτή την φορά χρησιμοποιώντας πακέτα μεγέθους 200 ψηφιολέξεων που έχουν χρόνο μεταξύ αποστολής τα 5ms. Τα αποτελέσματα παρουσιάζονται στον πίνακα 3.8 και γραφικά στο σχήμα 3.8.

Αριθμός Ροών	TCP Drop Rate %	TFRC Drop Rate %
10	-	-
25	5,0644	4,3258
50	20,2161	16,3095
75	29,558	19,3547
100	33,5809	28,6263
150	35,8794	36,0574
200	38,857	39,3621
250	41,4873	41,0147

Πίνακας 3.8 : Ποσοστό χαμένων πακέτων στη σύνδεση συμφόρησης.

Παρατηρούμε πως η μεταβολή στον χρόνο αποστολής των πακέτων δεν επηρεάζει την συμπεριφορά των ροών. Και πάλι παρατηρείται αρχικά το TFRC να έχει ένα μικρότερο ποσοστό χαμένων πακέτων αλλά αυτό αναιρείται με την αύξηση των ροών των δύο πρωτοκόλλων.



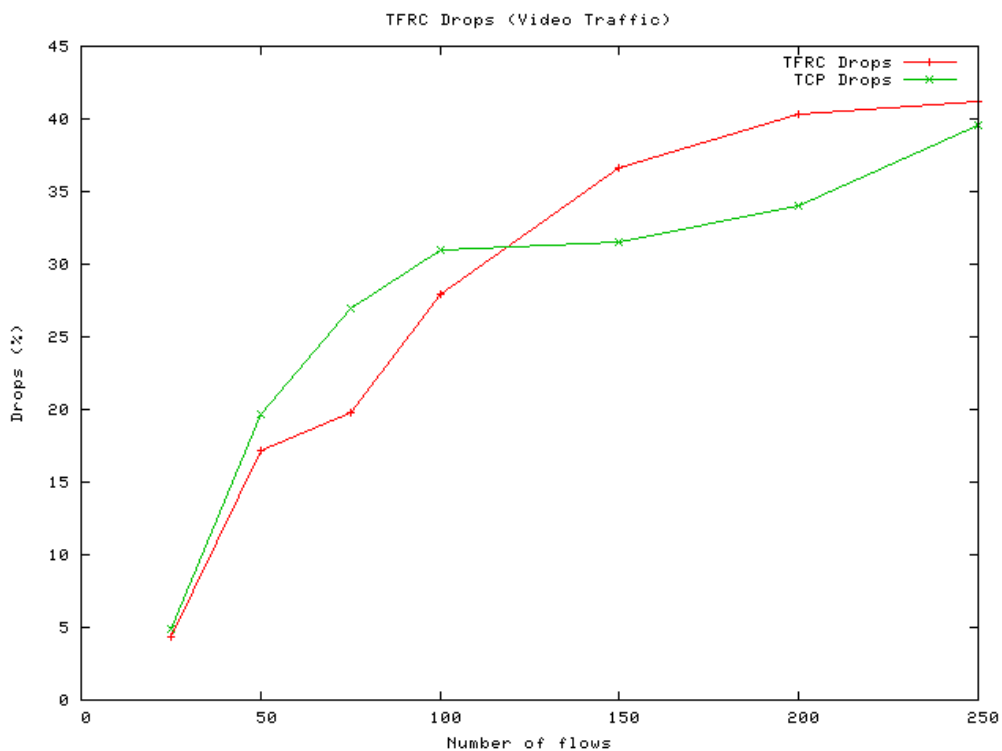
Σχήμα 3.8 : Ποσοστό χαμένων πακέτων στη σύνδεση συμφόρησης σε συνάρτηση ροών.

Τέλος όπως και στις προηγούμενες περιπτώσεις εφαρμόζουμε προσομοίωση με την χρήση πραγματικού βίντεο. Τα αποτελέσματα παρουσιάζονται στον πίνακα 3.9 και γραφικά στο σχήμα 3.9.

Αριθμός Ροών	TCP Drop Rate %	TFRC Drop Rate %
10	-	-
25	4,9254	4,3407
50	19,7236	17,1975
75	27,008	19,8234
100	30,9273	27,9419
150	31,5666	36,5944
200	34,053	40,3649
250	39,6029	41,244

Πίνακας 3.9 : Ποσοστό χαμένων πακέτων στη σύνδεση συμφόρησης.

Τα αποτελέσματα σε αυτή την περίπτωση διαφέρουν ελαφρά από τις άλλες περιπτώσεις. Σε χαμηλά ποσοστά συμφόρησης το TFRC παρουσιάζει μικρότερο ποσοστό χαμένων πακέτων, ενώ σε μεγάλα ποσοστά συμφόρησης το ποσοστό του TFRC ξεπερνά σε ένα μικρό ποσοστό εκείνο του TCP.

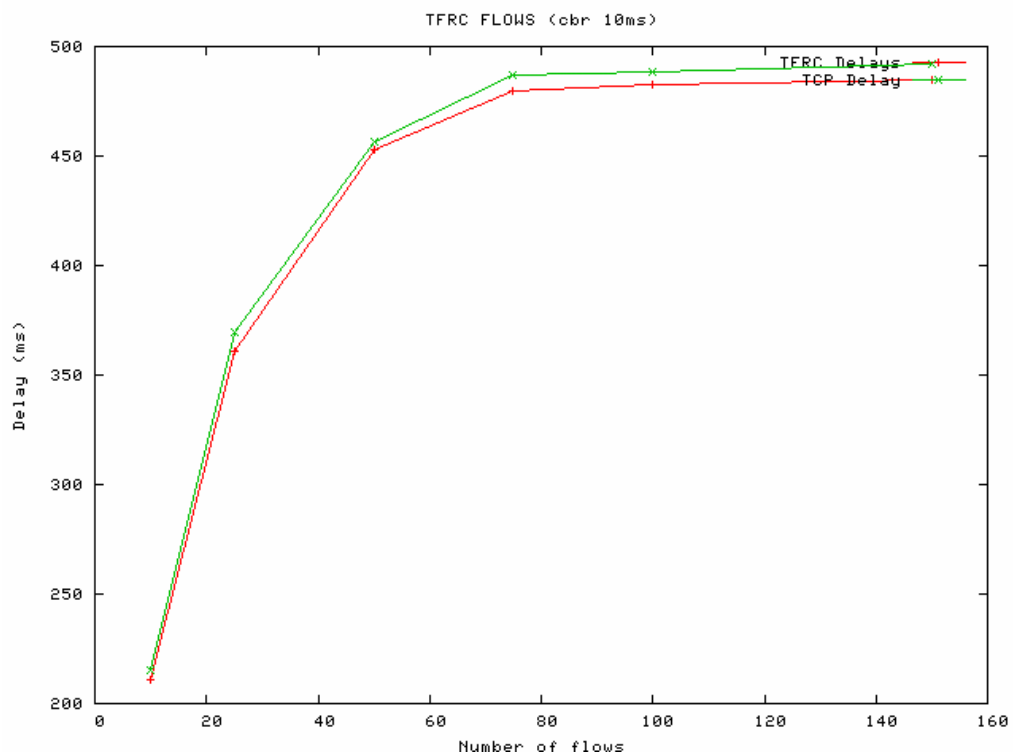


Σχήμα 3.9 : Ποσοστό χαμένων πακέτων στη σύνδεση συμφόρησης σε συνάρτηση ροών.

Το επόμενο πείραμα έχει σκοπό την εύρεση της καθυστέρησης των πακέτων και των δύο ροών που παρατηρείται στους δρομολογητές στη γραμμή όπου υπάρχει η συμφόρηση. Σκοπός του πειράματος αυτού είναι ο έλεγχος εάν τα πακέτα των δύο ροών τυγχάνουν της ίδιας προτεραιότητας στους δρομολογητές. Αρχικά ελέγχουμε τις μετρήσεις μας με χρήση πακέτων μεγέθους 200 ψηφιολέξεων που ο χρόνος αποστολής μεταξύ των πακέτων είναι 10ms. Τα αποτελέσματα παρουσιάζονται στον πίνακα 3.10 και γραφικά στο σχήμα 3.10.

Αριθμός Ροών	Mean Delay TFRC	Jitter TFRC	Mean Delay TCP	Jitter TCP
10	216,595	9,41864	215,083	12,3888
25	363,905	22,601	369,297	37,0816
50	414,01	16,2172	421,281	37,3135
75	490,0128	18,7673	493,301	62,789
100	488,7619	1,9488	489,43	63,1471
150	489,8115	25,1121	487,968	77,9676

Πίνακας 3.10 : Καθυστέρηση πακέτων στους δρομολογητές.



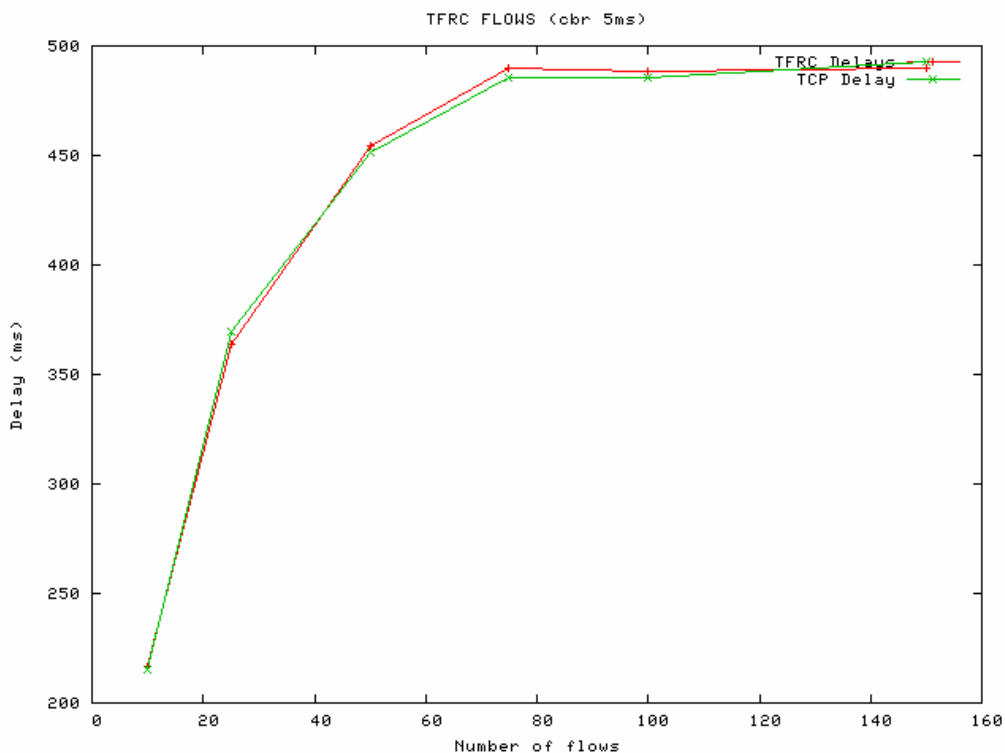
Σχήμα 3.10 : Καθυστέρηση πακέτων στους δρομολογητές.

Όπως παρατηρούμε και από το Σχήμα 3.10 στα αρχικά στάδια της συμφόρησης οι ροές και των δύο πρωτοκόλλων λαμβάνουν περίπου τον ίδιο χρόνο καθυστέρησης. Με την αύξηση όμως των ροών όπου η συμφόρηση γίνεται ακόμα πιο έντονη η καθυστέρηση των ροών TFRC αυξάνεται μέχρι ενός σημείου όπου παραμένει σταθερή.

Επαναλαμβάνουμε το ίδιο πείραμα αυτή την φορά κάνοντας χρήση πακέτων μεγέθους 200 ψηφιολέξεων που διαφέρουν στον χρόνο αποστολής τους κατά 5 ms. Τα αποτελέσματα παρουσιάζονται στον πίνακα 3.11 και γραφικά στο σχήμα 3.11.

Αριθμός Ροών	Mean Delay TFRC	Jitter TFRC	Mean Delay TCP	Jitter TCP
10	216,595	9,41864	215,083	12,3888
25	363,905	22,601	369,297	37,0816
50	414,01	16,2172	421,281	37,3135
75	490,0128	18,7673	493,301	62,789
100	488,7619	19,9488	487,43	63,1471
150	489,8115	25,1121	487,968	77,9676

Πίνακας 3.11 : Καθυστέρηση πακέτων στους δρομολογητές.

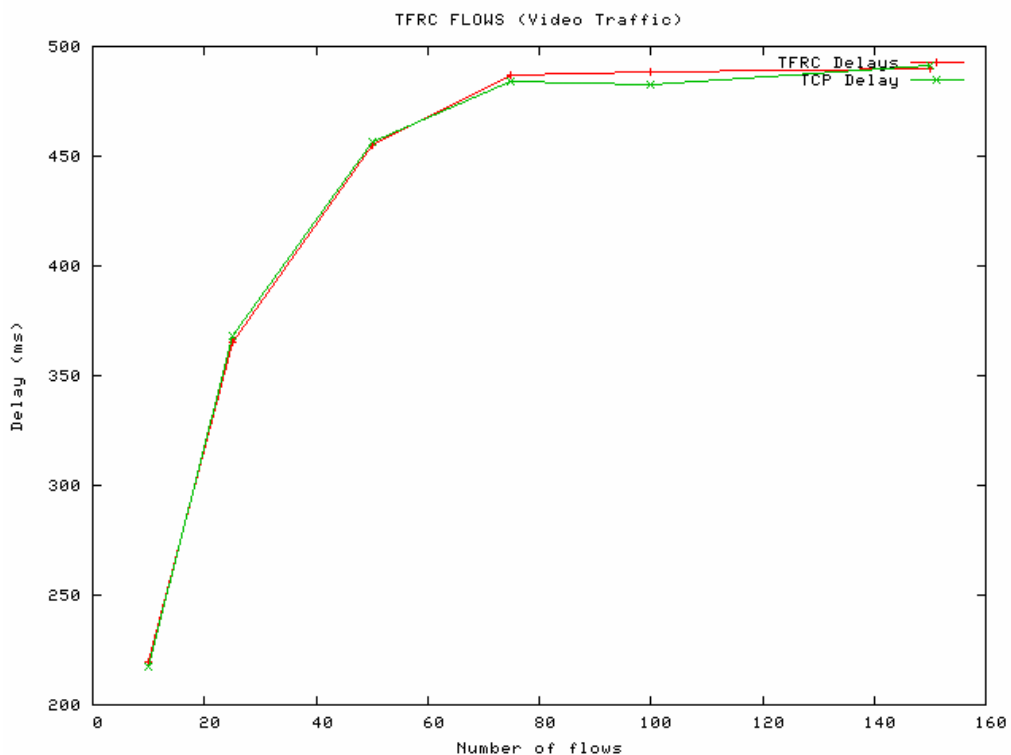


Σχήμα 3.11 : Καθυστέρηση πακέτων στους δρομολογητές.

Τα αποτελέσματα που παράγονται σε αυτή την περίπτωση δεν διαφέρουν κατά πολύ από την περίπτωση όπου ο χρόνος αποστολής μεταξύ των πακέτων ήταν σε πιο ψηλά επίπεδα. Και πάλι οι ροές TCP έχουν παρόμοια καθυστέρηση με τις ροές TFRC όσον αυξάνεται ο αριθμός των ροών. Δοκιμάζουμε για ακόμα μια φορά την προσομοίωση χρησιμοποιώντας πραγματική ταινία όπως και προηγουμένως. Τα αποτελέσματα παρουσιάζονται στον πίνακα 3.12 και γραφικά στο σχήμα 3.12.

Αριθμός Ροών	Mean Delay TFRC	Jitter TFRC	Mean Delay TCP	Jitter TCP
10	216,808	8,09178	215,182	12,5272
25	366,515	21,0344	373,536	41,6121
50	415,657	20,1741	429,161	50,5572
75	492,0061	24,5975	497,979	90,7429
100	489,1926	24,2077	482,172	85,5891
150	490,2404	31,8951	497,427	117,164

Πίνακας 3.12 : Καθυστέρηση πακέτων στους δρομολογητές.

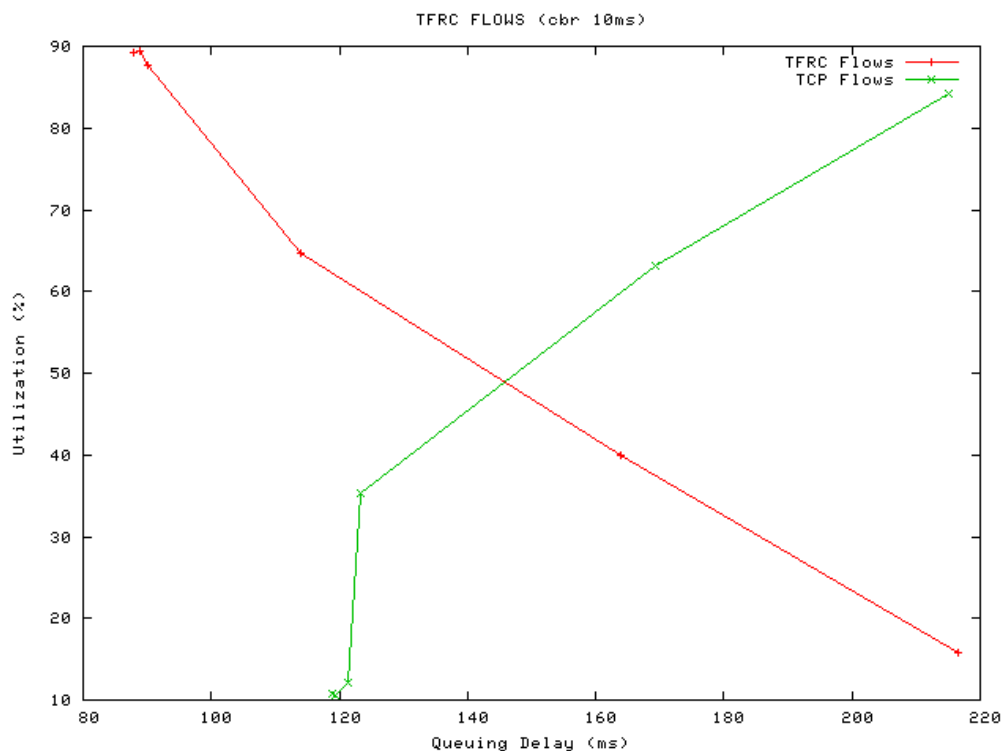


Σχήμα 3.12 : Καθυστέρηση πακέτων στους δρομολογητές.

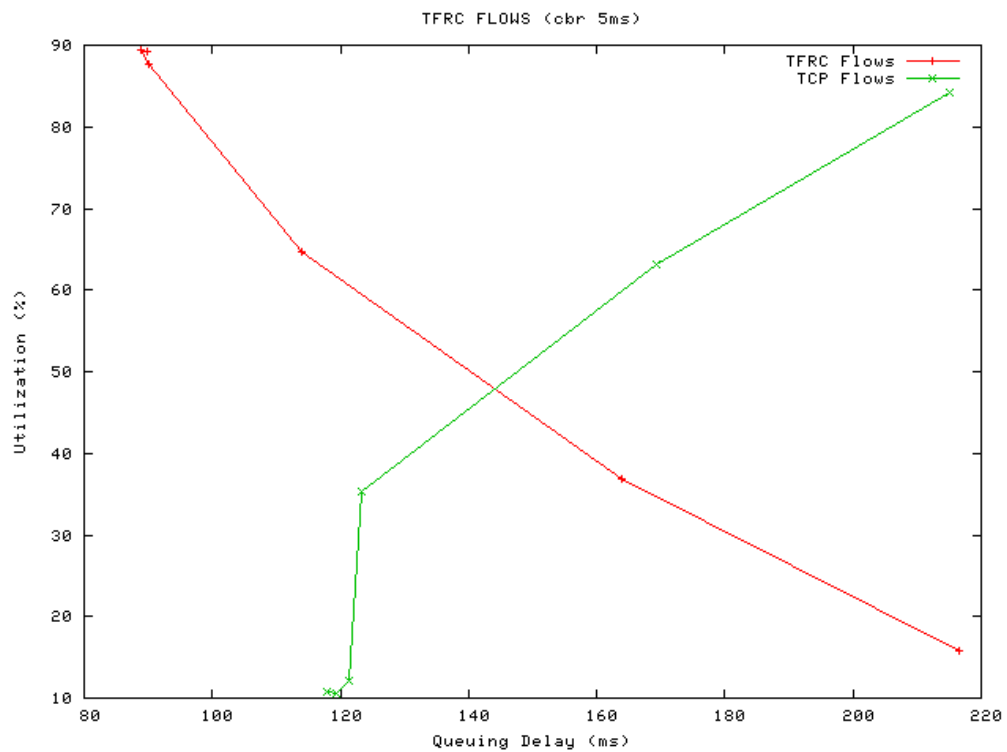
Παρατηρούμε πως σε αυτή την περίπτωση ισχύουν τα ίδια συμπεράσματα που έχουμε εξάξει και προηγουμένως ότι δηλαδή οι ροές των δύο πρωτοκόλλων παρουσιάζουν την

ίδια σχεδόν καθυστέρηση πακέτων η οποία μετά από ένα σημείο παραμένει σταθερή στα ίδια επίπεδα.

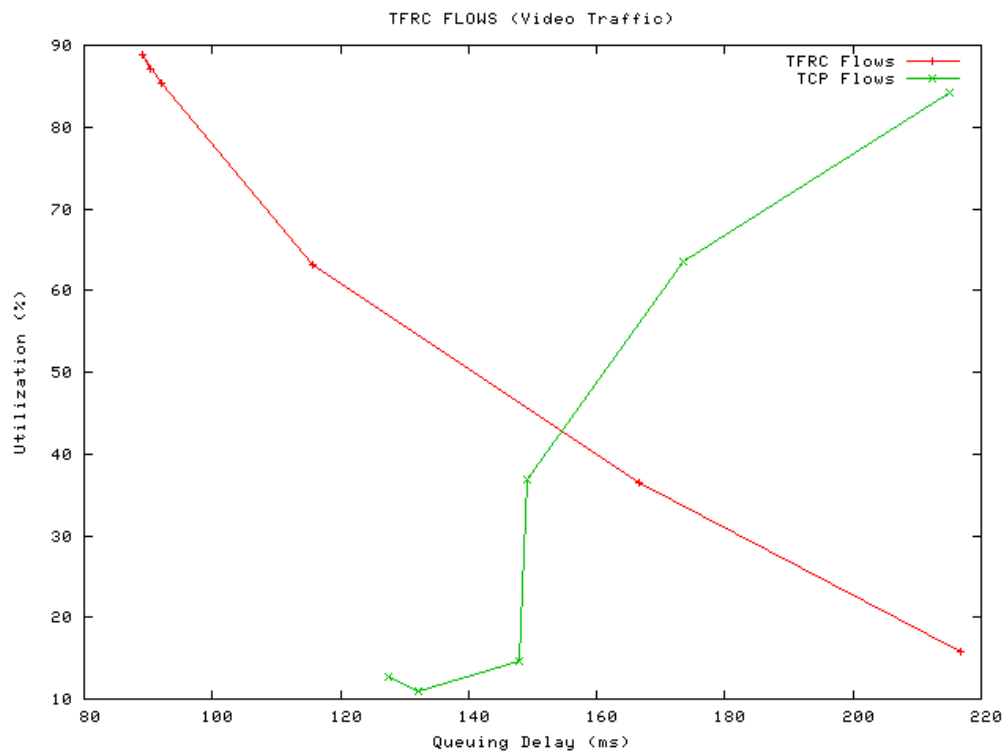
Πιο κάτω παρουσιάζουμε μια σύνοψη των πιο πάνω αποτελεσμάτων των μετρήσεων που έγιναν στην σύνδεση όπου παρατηρείται συμφόρηση. Στα σχήματα 3.13 μέχρι 3.15 παρουσιάζεται το ποσοστό χρήσης της σύνδεσης όπου παρατηρείται συμφόρηση σε συνάρτηση με την καθυστέρηση των πακέτων και των δύο ροών. Το σχήμα 3.13 είναι με χρήση πακέτων με χρόνο αποστολής μεταξύ τους 10ms, το σχήμα 3.14 με χρόνο 5ms και το σχήμα 3.15 με προσομοίωση βίντεο. Σημειώνουμε πως το άθροισμα των δύο ροών ως προς τον άξονα των Ψ δίνει 100% μιας και η χρήση της σύνδεσης είναι ολοκληρωτική.



Σχήμα 3.14 : Χρήση σύνδεσης συμφόρησης σε συνάρτηση του χρόνου καθυστέρησης των πακέτων για πακέτα με χρόνο αποστολής μεταξύ τους 10 ms.



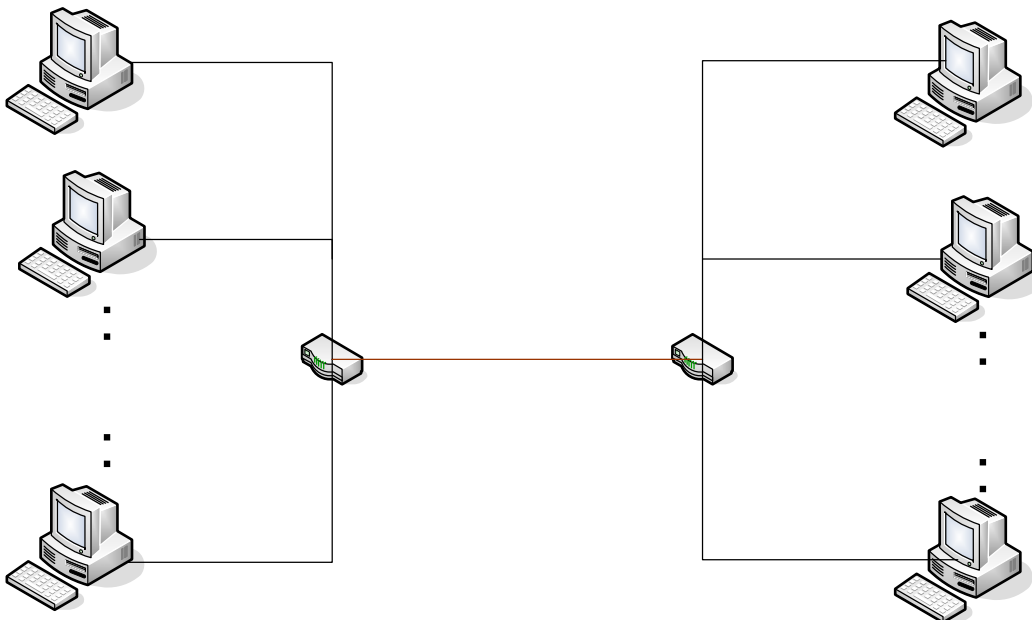
Σχήμα 3.15 : Χρήση σύνδεσης συμφόρησης σε συνάρτηση του χρόνου καθυστέρησης των πακέτων για πακέτα με χρόνο αποστολής μεταξύ τους 5 ms.



Σχήμα 3.16 : Χρήση σύνδεσης συμφόρησης σε συνάρτηση του χρόνου καθυστέρησης των πακέτων για πακέτα με προσομοίωση βίντεο.

Στο σημείο όπου διασταυρώνονται οι δύο γραφικές δηλώνει το χρόνο καθυστέρησης πακέτων των δύο ροών όπου το ποσοστό του εύρους ζώνης που καταλαμβάνουν στην σύνδεση είναι ίσο. Για χρόνο αποστολής μεταξύ πακέτων 10ms και 5ms ο χρόνος καθυστέρησης βρίσκεται γύρω στα 143ms ενώ στην προσομοίωση βίντεο ο χρόνος αυτός είναι κάπως μεγαλύτερος και βρίσκεται στα 155ms.

Για επιβεβαίωση των υπολογισμών μας υλοποιήσαμε μια διαφορετική τοπολογία που φαίνεται στο σχήμα 3.17. Κάθε κόμβος αντί να στέλνει πολλές ροές από το κάθε πρωτόκολλο, στέλνει μόνο μια ροή. Για να προσομοιώσουμε τον μεγάλο αριθμό ροών τοποθετήσαμε στην τοπολογία μας ένα μεγάλο αριθμό κόμβων τον οποίο ενώνουμε με τον δρομολογητή που βρίσκεται στην αρχή της σύνδεσης όπου θα παρατηρηθεί η συμφόρηση. Οι συνδέσεις αυτές έχουν και διαφορετικό βαθμό καθυστέρησης στην παράδοση των πακέτων. Με χρήση της τοπολογίας αυτής τρέξαμε όλα τα προηγούμενα σενάρια και είδαμε πως δίνει ακριβώς τα ίδια αποτελέσματα για όλα τα πρωτόκολλα που έχουν χρησιμοποιηθεί. Επομένως κρατήσαμε την βασική τοπολογία που περιγράψαμε στο σχήμα 3.0.1.



Σχήμα 3.17 : Εναλλακτική τοπολογία Προσομοίωσης

Επίσης μια δεύτερη μετατροπή η οποία έγινε ήταν να σταματούμε τις μισές ροές από κάθε πρωτόκολλο στο τεσσαρακοστό δευτερόλεπτο και να τις ξεκινήσουμε ξανά στο

εβδομηκοστό δευτερόλεπτο. Και πάλι τα αποτελέσματα που πήραμε δεν διαφέρουν σε μεγάλο βαθμό από αυτά που έδινε η βασική τοπολογία.

3.3 Προσομοιώσεις με χρήση του SCTP

Χρησιμοποιώντας την βασική τοπολογία που έχουμε περιγράψει στο σχήμα 3.0.1 στέλνουμε τις ροές TCP στο μονοπάτι που αποτελείται από τους κόμβους 0, 2, 3, 4 και τις ροές SCTP από το μονοπάτι που αποτελείται από τους κόμβους 1, 2, 3, 5. Οι μετρήσεις γίνονται με χρήση 10, 25, 50, 75, 100 και 150 ροών από το κάθε πρωτόκολλο. Οι προσομοιώσεις διαρκούν ένα σταθερό χρόνο που έχουμε ορίσει στα 100 δευτερόλεπτα.

Λόγω της ιδιομορφίας που παρουσιάζουν οι κόμβοι του SCTP, χρησιμοποιούμε τρεις κόμβους στον προσομοιωτή για να υλοποιήσουμε ένα κόμβο SCTP. Συγκεκριμένα ο ένας κόμβος είναι ο πυρήνας και οι άλλοι δύο είναι οι κόμβοι οι οποίοι θα προσομοιώνουν τα εναλλακτικά μονοπάτια αποστολής των δεδομένων. Οι τρεις αυτοί κόμβοι είναι ενωμένοι μεταξύ τους. Επίσης οι κόμβοι που προσομοιώνουν τα εναλλακτικά μονοπάτια είναι ενωμένοι με τους δρομολογητές που βρίσκονται στην σύνδεση που υπάρχει συμφόρηση.

Το πρωτόκολλο έχει δημιουργηθεί για να παρέχει αξιόπιστη μεταφορά δεδομένων. Όμως εμείς θα αλλάξουμε κάποιες παραμέτρους για να προσομοιώσουμε μη αξιόπιστες ροές. Πιο συγκεκριμένα έχουμε καθορίσει το μέγεθος των πακέτων να είναι 200 ψηφιολέξεις, έχουμε καθορίσει τον αριθμό των ρευμάτων στην σύνδεση να είναι 5 τα οποία θα είναι όλα μη αξιόπιστα αφού τα δεδομένα που θα στέλνουμε θα είναι πολυμεσικά και καθορίσαμε ότι δεν θα υπάρχουν επαναληπτικές αποστολές πακέτων σε περίπτωση που κάποιο από τα πακέτα που στέλνουμε θα έχει χαθεί.

Έπειτα δημιουργούμε πακέτα για την εφαρμογή που θα χρησιμοποιήσει το SCTP. Το μέγεθος των πακέτων αυτών έχει καθοριστεί στις 200 ψηφιολέξεις και ορίσαμε πως τα πακέτα αυτά θα στέλνονται κάθε ένα μεταβλητό αριθμό δευτερολέπτων.

Οι μετρήσεις μας έγιναν για 10ms και για 5ms. Επίσης έγιναν προσομοιώσεις με χρήση πραγματικής ταινίας. Αν για παράδειγμα κάνουμε χρήση πακέτων με μέγεθος 200 ψηφιολέξεις και χρόνο μεταξύ μετάδοσης πακέτων 10ms τότε ο ρυθμός μετάδοσης που αναμένουμε για κάθε ροή SCTP θα είναι : $200 * 8 \text{ bits} / 0,01\text{ms} = 160\text{Kbps}$. Τα αποτελέσματα από τις διάφορες μετρήσεις τα αποθηκεύουμε σε αρχεία τα οποία επεξεργαζόμαστε στην συνέχεια.

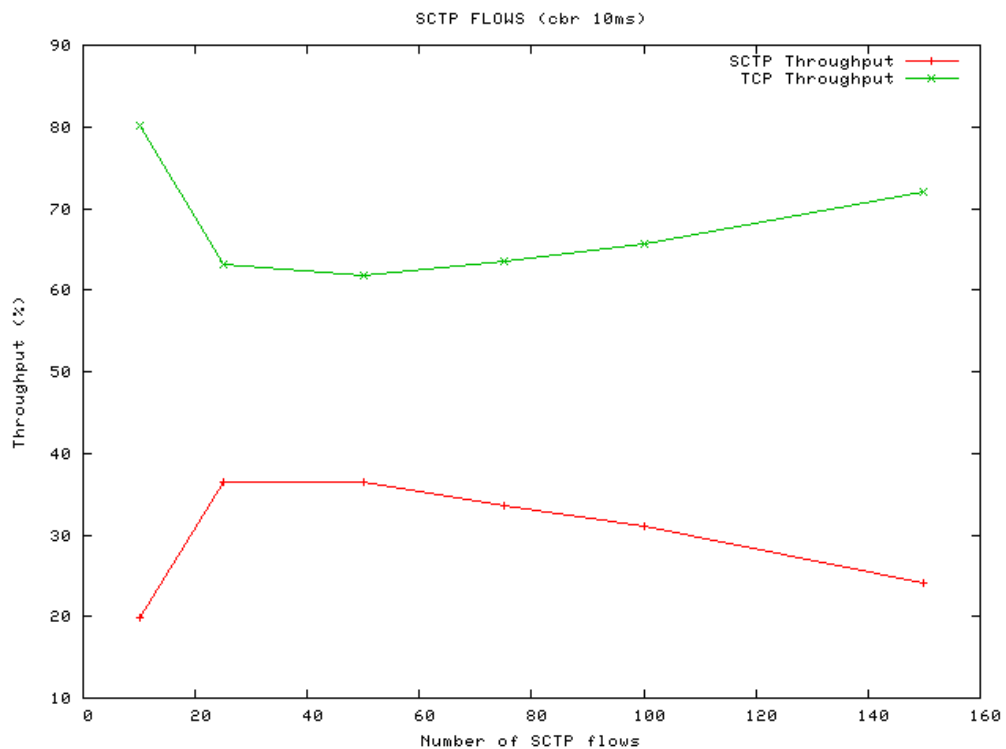
Η πρώτη προσομοίωση αφορά το ποσοστό χρήσης της σύνδεσης από τα δύο πρωτόκολλα στα άκρα. Οι μετρήσεις έγιναν για διαφορετικό αριθμό ροών και τα αποτελέσματα καταγράφονται στον πίνακα 3.13 και παρουσιάζονται γραφικά στο σχήμα 3.18. Τα πακέτα που χρησιμοποιεί το SCTP είναι τύπου CBR με διάστημα μεταξύ αποστολής τα 10ms. Επίσης ο πίνακας 3.13 παρουσιάζει το μέσο εύρος ζώνης της σύνδεσης για την κάθε ροή από το κάθε πρωτόκολλο.

Αριθμός Ροών	TCP Throughput %	TCP Throughput Mb	SCTP Throughput %	SCTP Throughput Mb
10	80,0836	0,8	19,7918	0,2
25	63,171	0,260	36,414	0,175
50	61,7666	0,101	36,4405	0,099
75	63,5406	0,065	33,5813	0,016
100	65,6014	0,068	30,9794	0,048
150	72,0016	0,055	24,0592	0,014

Πίνακας 3.13 : Ποσοστό χρήσης σύνδεσης στα άκρα και μέσος όρος εύρους ζώνης

Παρατηρούμε πως με την αύξηση του αριθμού των ροών από το κάθε πρωτόκολλο το SCTP παίρνει μεγαλύτερο ποσοστό από την σύνδεση σε σχέση με το TCP. Αρχικά όλες οι ροές του SCTP περιορίζουν τον ρυθμό αποστολής τους σε 200 Kbps που είναι μεγαλύτερο από το εύρος ζώνης που χρειάζεται η εφαρμογή αλλά πάλι δεν διεκδικούν όλο το διαθέσιμο εύρος ζώνης όπως κάνουν οι ροές TCP.

Αντίθετα το TCP τείνει να εκμεταλλευτεί όσο το δυνατό περισσότερο εύρος ζώνης γι' αυτό και οι ροές έχουν μεγαλύτερο ρυθμό. Με την αύξηση όμως των ροών και κατ' επέκταση την αύξηση της συμφόρησης, οι ροές SCTP δεν μπορούν να πάρουν τα επιθυμητά 160Kb αλλά ούτε και προσπαθούν να εκμεταλλευτούν το διαθέσιμο εύρος ζώνης εις βάρος του TCP.

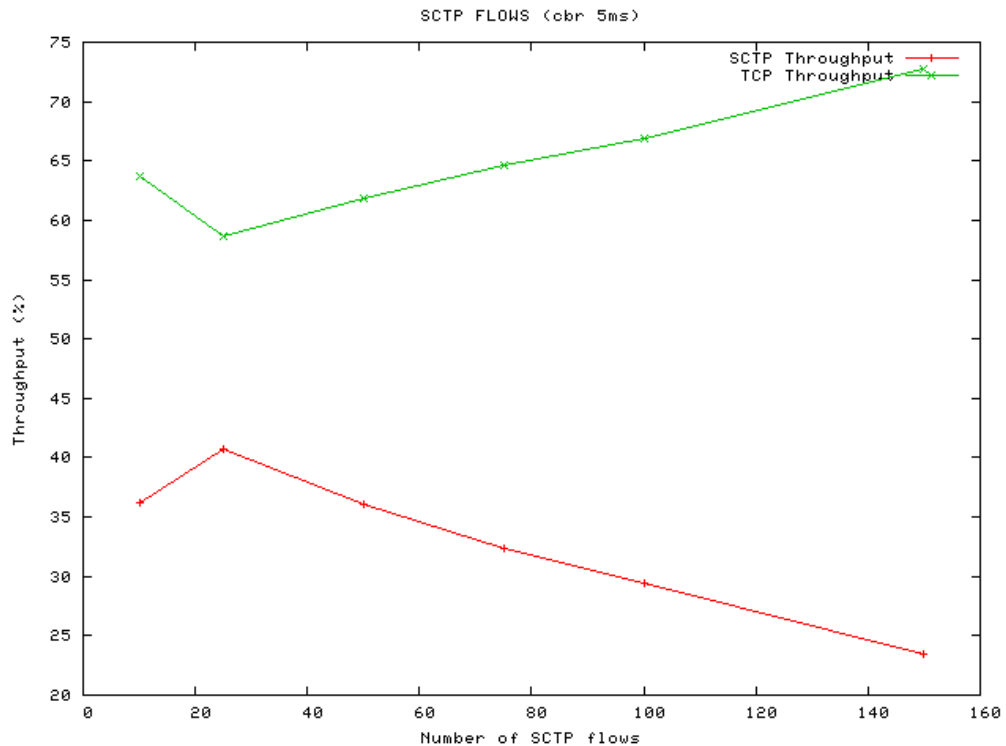


Σχήμα 3.18 : Ποσοστό χρήσης σύνδεσης συναρτήσει του αριθμού ροών TCP και SCTP

Επαναλαμβάνουμε την ίδια διαδικασία αλλά αυτή την φορά μειώνουμε τον χρόνο μεταξύ της κάθε αποστολής πακέτων της εφαρμογής στα 5ms. Οι μετρήσεις έγιναν για διαφορετικό αριθμό ροών και τα αποτελέσματα καταγράφονται στον πίνακα 3.14 και παρουσιάζονται γραφικά στο σχήμα 3.19. Επίσης ο πίνακας 3.14 παρουσιάζει το μέσο εύρος ζώνης της σύνδεσης για την κάθε ροή από το κάθε πρωτόκολλο.

Αριθμός Ροών	TCP Throughput %	TCP Throughput Mb	SCTP Throughput %	SCTP Throughput Mb
10	63,667	0,5800	36,2011	0,357
25	58,7051	0,3243	40,7721	0,1499
50	61,8787	0,1141	36,066	0,0813
75	64,6061	0,0758	32,303	0,0493
100	66,898	0,0801	29,497	0,0307
150	72,7119	0,0368	23,4473	0,01691

Πίνακας 3.14 : Ποσοστό χρήσης σύνδεσης στα άκρα και μέσος όρος εύρους ζώνης



Σχήμα 3.19 : Ποσοστό χρήσης σύνδεσης συναρτήσει του αριθμού ροών TCP και SCTP

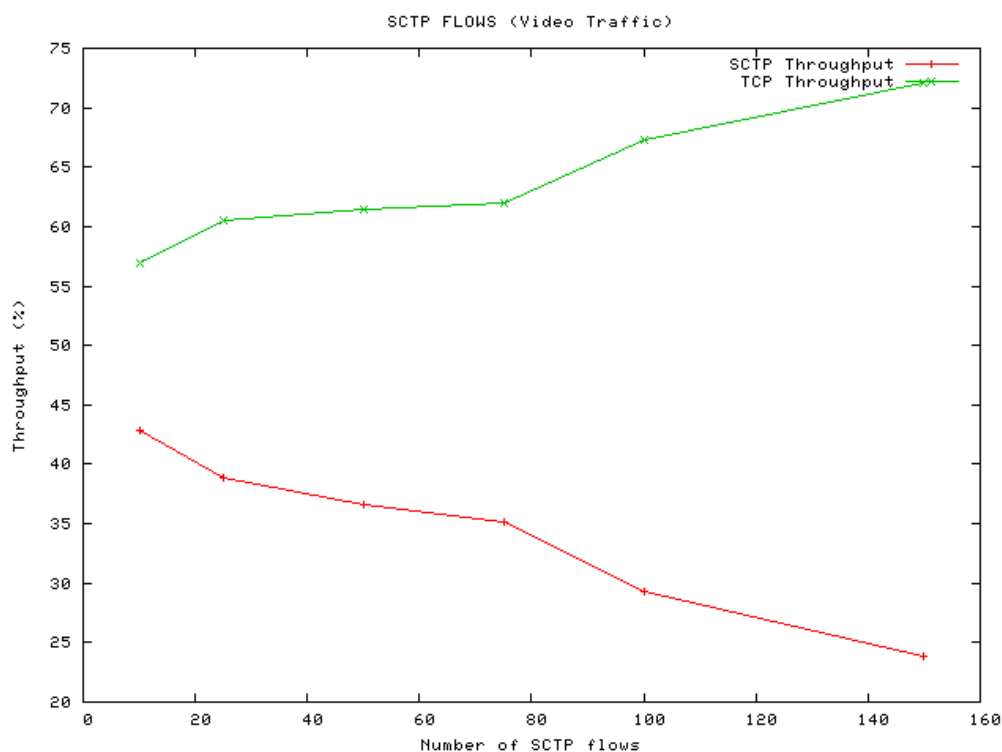
Παρατηρούμε όπως και στην προηγούμενη περίπτωση ότι οι ροές SCTP αρχικά καταλαμβάνουν μικρότερο εύρος ζώνης από αυτό που δικαιούνται, αλλά με την αύξηση των ροών και της συμφόρησης εκμεταλλεύονται το εύρος ζώνης εις βάρος του TCP.

Τέλος δοκιμάζουμε την ίδια διαδικασία αλλά αντί για χρήση πακέτων εφαρμογής προσομοιώνουμε με την χρήση πραγματικού βίντεο [4]. Το βίντεο που αρχικά είναι χωρισμένο σε διάφορα πλαίσια ταινίας μετατρέπεται σε πακέτα, με χρήση ενός μικρού προγράμματος το οποίο μετατρέπει τα πλαίσια αυτά σε πακέτα αποστολής σταθερού ρυθμού.

Όπως αναφέρει ο προγραμματιστής του προγράμματος αυτού η προσομοίωση θα είναι παρόμοια εάν είχαμε χρήση πακέτων με μέγεθος 200 ψηφιολέξεις που θα είχαν διαφορά χρόνου στην αποστολή τους 10ms. Τα αποτελέσματα της διαδικασίας παρουσιάζονται στον πίνακα 3.3 και η γραφική απεικόνιση στο σχήμα 3.3.

N	TCP Throughput %	TCP Throughput Mb	SCTP Throughput %	SCTP Throughput Mb
10	56,9909	0,6125	42,882	0,5734
25	60,5261	0,3380	38,897	0,1037
50	61,3861	0,1243	36,5863	0,0431
75	61,9931	0,0780	35,117	0,0469
100	67,2398	0,0768	29,3536	0,0177
150	72,0973	0,0447	23,8929	0,0098

Πίνακας 3.15 : Ποσοστό χρήσης σύνδεσης στα άκρα και μέσος όρος εύρους ζώνης



Σχήμα 3.20 : Ποσοστό χρήσης σύνδεσης συναρτήσει του αριθμού ροών TCP και SCTP

Όπως και στις προηγούμενες περιπτώσεις έτσι και εδώ παρατηρούμε πως η προσομοίωση με την χρήση πραγματικού βίντεο, δίνει περίπου τα ίδια αποτελέσματα με αυτά που παίρνουμε και στις άλλες περιπτώσεις. Συμπεραίνουμε τελικά πως στην ύπαρξη συμφόρησης οι ροές SCTP εκμεταλλεύονται το εύρος ζώνης της σύνδεσης εις βάρος του TCP.

Για την δεύτερη μας μέτρηση θα εξετάσουμε την καθυστέρηση που παρουσιάζουν τα πακέτα σε συνάρτηση με τον αριθμό των ροών του SCTP. Από τις προηγούμενες μετρήσεις παρατηρήσαμε πως οι ροές SCTP κερδίζουν σε εύρος ζώνης εις βάρος του

TCP. Ένα λογικό σενάριο θα ήταν οι ροές SCTP να παρουσιάζουν λιγότερη καθυστέρηση στα άκρα ώστε να κερδίζουν σε εύρος ζώνης.

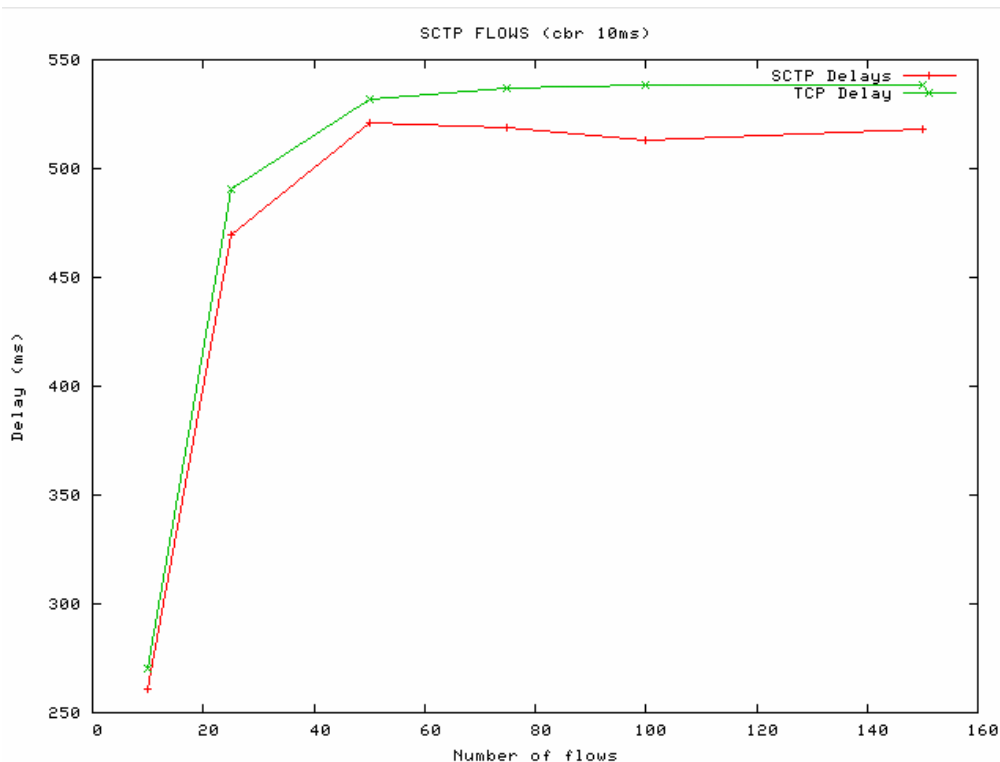
Πιο κάτω παρουσιάζουμε τα αποτελέσματα των μετρήσεων της καθυστέρησης σε συνάρτηση με τον αριθμό των ροών για τα δύο πρωτόκολλα. Αρχικά παρουσιάζουμε τα αποτελέσματα με την χρήση πακέτων εφαρμογής με μέγεθος 200 ψηφιολέξεις και χρόνο μεταξύ αποστολής τα 10ms.

Τα αποτελέσματα παρουσιάζονται στον πίνακα 3.16 και γραφικά στο σχήμα 3.21. Επίσης ο πίνακας περιέχει το ποσοστό των πακέτων που χάνονται από την έξοδο της σύνδεσης που παρουσιάζεται συμφόρηση μέχρι τον παραλήπτη, καθώς και η διαφορά χρόνου στον οποίο κάποια πακέτα φτάνουν στον παραλήπτη μετά που ακολούθησαν διαφορετική διαδρομή (Jitter).

N	SCTP Drop Rate %	Mean Delay SCTP	Mean Delay TCP	Jitter SCTP
10	0,26	260,827	258,425	9,300
25	6,6	469,789	467,128	75,629
50	9,61	520,941	519,128	50,1362
75	14,43	518,518	519,328	40,762
100	18,67	513,134	516,485	44,45
150	24,656	518,186	518,158	46,504

Πίνακας 3.16 : Μέση καθυστέρηση πακέτων στα άκρα για το SCTP.

Παρατηρούμε πως με την αύξηση των ροών η καθυστέρηση που παρουσιάζουν οι ροές SCTP αυξάνεται μέχρι ενός σημείου και από εκεί και πέρα η καθυστέρηση παραμένει σταθερή. Αυτός είναι και ο λόγος που κερδίζουν περισσότερο εύρος ζώνης εις βάρος του TCP. Οι ροές του TCP με την σειρά τους αυξάνουν και αυτές την καθυστέρηση όσο αυξάνονται οι ροές αλλά με μεγαλύτερη κλίμακα από αυτές του SCTP, επομένως και χάνουν εύρος ζώνης.



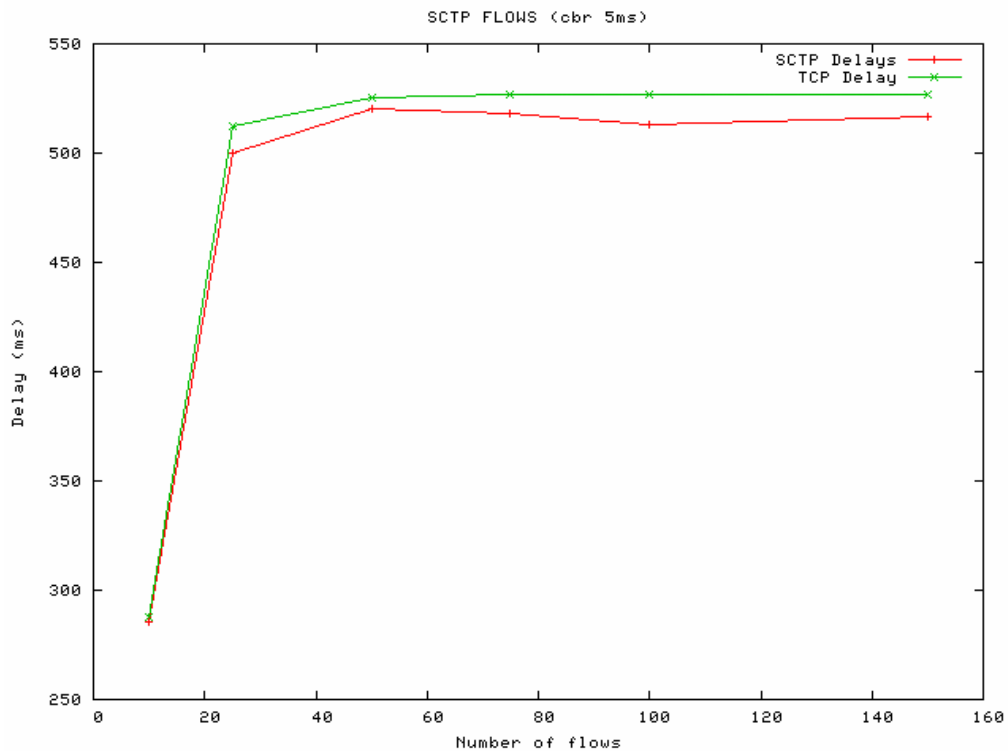
Σχήμα 3.21 : Καθυστέρηση πακέτων σε συνάρτηση του αριθμού ροών SCTP.

Επαναλαμβάνουμε την ίδια διαδικασία αλλά αυτή την φορά κάνουμε χρήση πακέτων μεγέθους 200 ψηφιολέξεων με χρόνο μεταξύ αποστολής τα 5ms. Τα αποτελέσματα παρουσιάζονται στον πίνακα 3.17 και γραφικά στο σχήμα 3.22.

Επίσης ο πίνακας περιέχει το ποσοστό των πακέτων που χάνονται από την έξοδο της σύνδεσης που παρουσιάζεται συμφόρηση μέχρι τον παραλήπτη, καθώς και η διαφορά χρόνου στον οποίο κάποια πακέτα φτάνουν στον παραλήπτη μετά που ακολούθησαν διαφορετική διαδρομή (Jitter).

Αριθμός Ροών	SCTP Drop Rate %	Mean Delay SCTP	Mean Delay TCP	Jitter SCTP
10	2,046	285,475	283,185	49,070
25	6,44	500,158	499,128	84,113
50	8,44	515,576	513,486	55,87
75	13,246	518,282	517,185	41,744
100	19,823	513,248	510,485	45,321
150	32,0166	516,603	512,449	51,652

Πίνακας 3.17 : Μέση καθυστέρηση πακέτων στα άκρα για το SCTP.

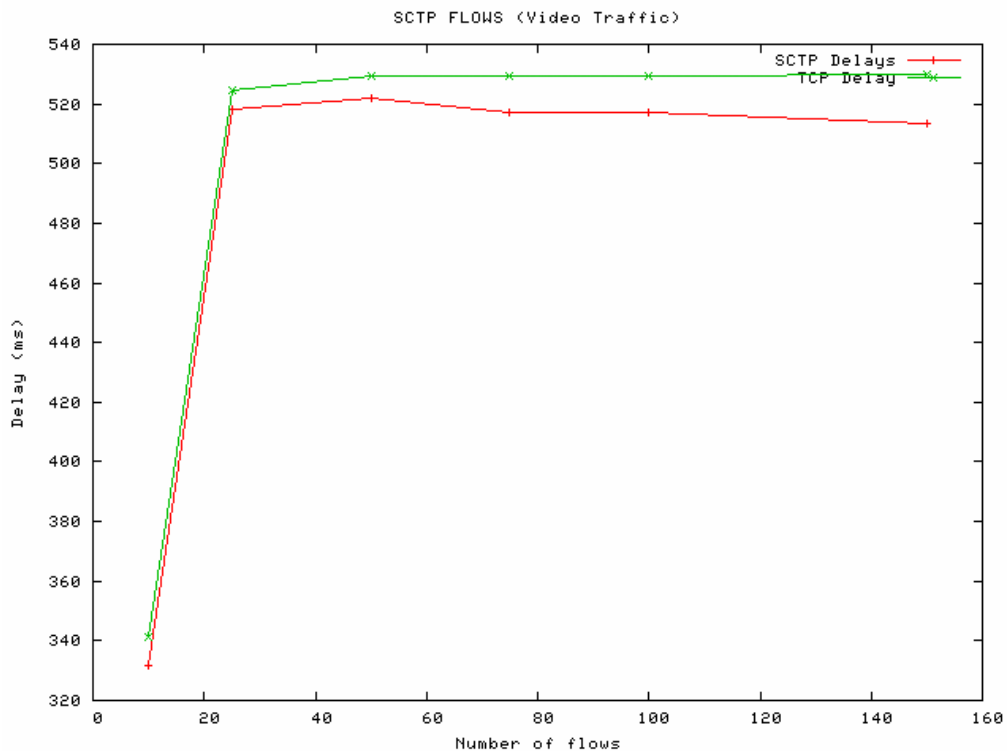


Σχήμα 3.22 : Καθυστέρηση πακέτων σε συνάρτηση του αριθμού ροών SCTP.

Παρατηρούμε σε σύγκριση με τα προηγούμενα αποτελέσματα πως η καθυστέρηση κυμαίνεται στα ίδια περίπου επίπεδα και πως η αλλαγή στο χρόνο αποστολής μεταξύ των πακέτων δεν επιφέρει οποιαδήποτε ουσιαστική αλλαγή στα αποτελέσματα. Τέλος προσομοιώνουμε με την χρήση αποστολής κανονικής ταινίας όπως ακριβώς και το προηγούμενο πείραμα. Τα αποτελέσματα παρουσιάζονται στον πίνακα 3.18 και γραφικά στο σχήμα 3.23. Επίσης ο πίνακας περιέχει το ποσοστό των πακέτων που χάνονται από την έξοδο της σύνδεσης που παρουσιάζεται συμφόρηση μέχρι τον παραλήπτη, καθώς και η διαφορά χρόνου στον οποίο κάποια πακέτα φτάνουν στον παραλήπτη μετά που ακολούθησαν διαφορετική διαδρομή (Jitter).

Αριθμός Ροών	SCTP Drop Rate %	Mean Delay SCTP	Mean Delay TCP	Jitter SCTP
10	2,343	331,646	328,128	510,84
25	4,793	517,954	516,843	64,111
50	10,24	521,898	519,125	48,129
75	13,856	517,314	516,182	38,632
100	17,63	516,981	515,154	43,920
150	24,61	513,536	512,185	46,776

Πίνακας 3.18 : Μέση καθυστέρηση πακέτων στα άκρα για το SCTP.



Σχήμα 3.23 : Καθυστέρηση πακέτων σε συνάρτηση του αριθμού ροών SCTP.

Συμπεραίνουμε πως η χρήση του βίντεο δεν προκαλεί μεγάλες αλλαγές στην καθυστέρηση που παρατηρείται στα πακέτα SCTP. Συγκρίνοντας τα αποτελέσματα των μέχρι στιγμής πειραμάτων βλέπουμε πως επειδή η καθυστέρηση των πακέτων SCTP είναι μικρότερη από αυτή του TCP, οι ροές του SCTP παίρνουν μεγαλύτερο εύρος ζώνης.

Για τα επόμενα πειράματα χρησιμοποιούμε μετρικές τις οποίες λαμβάνουμε στην γραμμή από την οποία υπάρχει συμφόρηση. Το πρώτο τέτοιο πείραμα αφορούσε το ποσοστό της χρήσης της γραμμής στην οποία υπάρχει συμφόρηση σε συνάρτηση με τον αριθμό των ροών του κάθε πρωτοκόλλου.

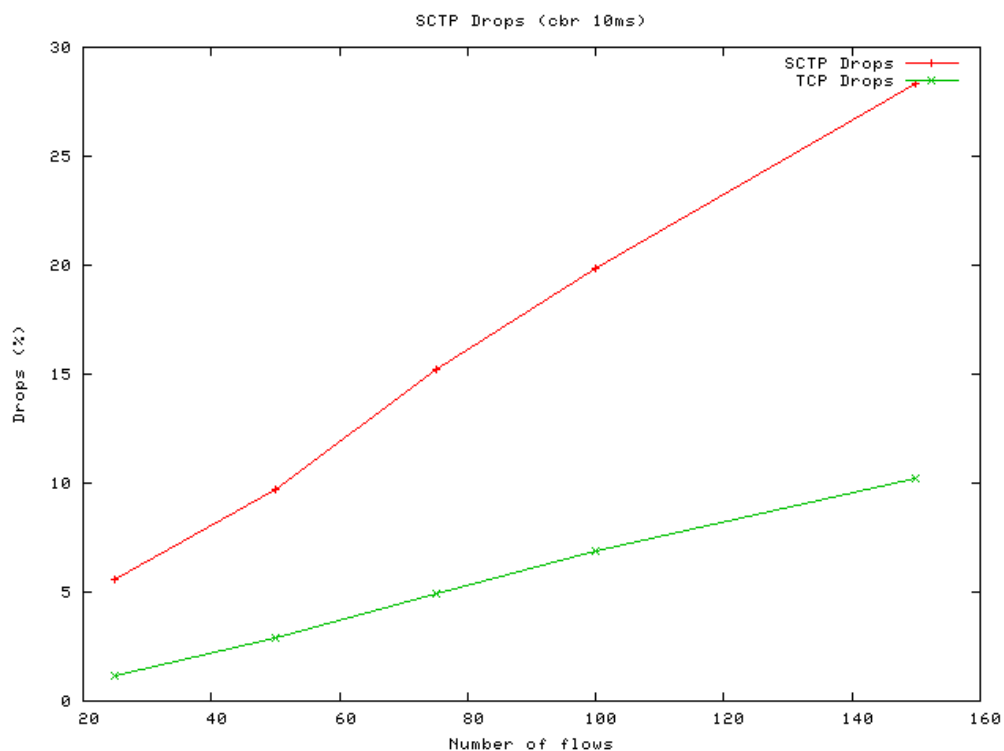
Λόγω του ότι το εύρος ζώνης της σύνδεσης είναι 10 φορές μικρότερο από αυτό που ενώνει την έξοδο της σύνδεσης με τον παραλήπτη, δεν παρατηρείται χάσιμο πακέτων στην γραμμή αυτή. Επομένως τα αποτελέσματα αυτής της μέτρησης ταυτίζονται με τα αποτελέσματα του πειράματος που αφορά την χρήση της σύνδεσης στον παραλήπτη σε συνάρτηση με τον αριθμό των ροών. Σχηματικά τα αποτελέσματα είναι τα ίδια με αυτά των σχημάτων 3.18 μέχρι 3.20.

Το επόμενο πείραμα που αφορά την σύνδεση συμφόρησης αφορά την απώλεια των πακέτων των διαφόρων ροών σε συνάρτηση με τον αριθμό των ροών. Θα περιμέναμε εδώ να δούμε πως οι ροές που έχουν το μεγαλύτερο εύρος ζώνης να έχουν και το μεγαλύτερο ποσοστό χαμένων πακέτων.

Στο πρώτο πείραμα χρησιμοποιούμε πακέτα εφαρμογής μεγέθους 200 ψηφιολέξεων και χρόνο μεταξύ αποστολής των πακέτων 10ms. Τα αποτελέσματα παρουσιάζονται στον πίνακα 3.19 και γραφικά στο σχήμα 3.24.

Αριθμός Ροών	TCP Drop Rate %	SCTP Drop Rate %
10	-	-
25	1,19502	5,58564
50	2,88839	9,73968
75	4,89209	15,2518
100	6,9074	19,8201
150	10,1843	28,3578

Πίνακας 3.19 : Ποσοστό χαμένων πακέτων στη σύνδεση συμφόρησης.

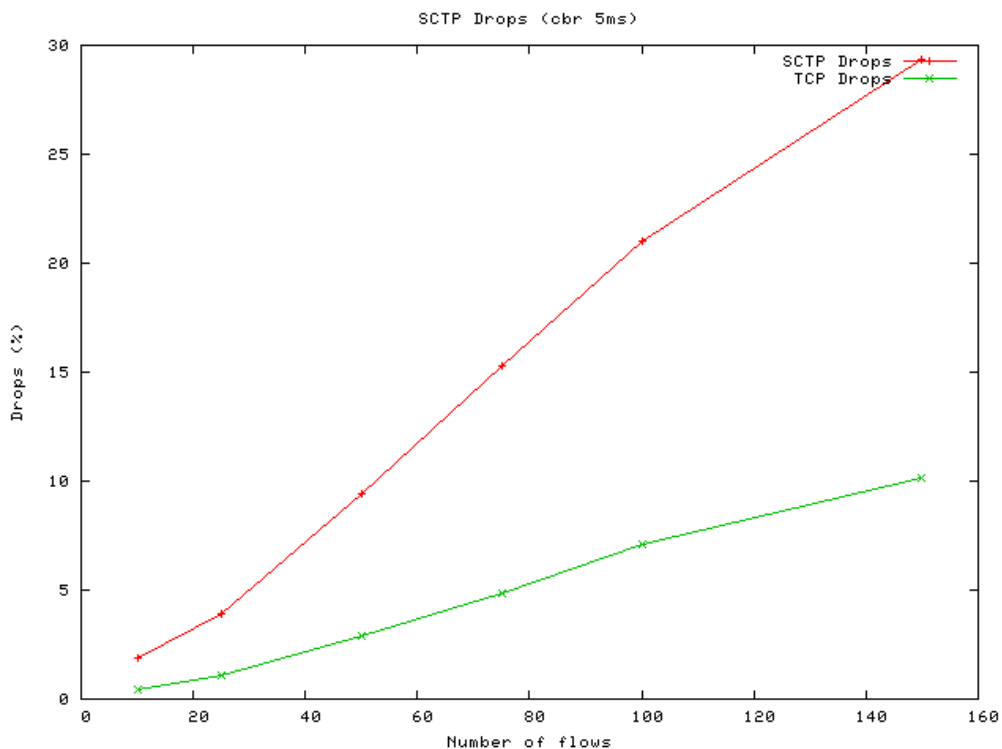


Σχήμα 3.24 : Ποσοστό χαμένων πακέτων στη σύνδεση συμφόρησης συναρτήσει ροών.

Παρατηρούμε πως οι ροές SCTP έχουν μεγαλύτερο ποσοστό χαμένων πακέτων σε όλη την διάρκεια των προσομοιώσεων. Οι γραφικές των πρωτοκόλλων είναι σχεδόν παράλληλες που δικαιολογεί το γεγονός ότι το ένα πρωτόκολλο δεν προσπαθεί να εκμεταλλευτεί το διαθέσιμο εύρος ζώνης εις βάρος του άλλου πρωτοκόλλου. Η συμπεριφορά του SCTP ως προς το TCP είναι πολύ καλύτερη από αυτή που παρουσιάζει το TFRC. Επαναλαμβάνουμε το ίδιο πείραμα όπως και στην προηγούμενη περίπτωση αλλά αυτή την φορά χρησιμοποιώντας πακέτα μεγέθους 200 ψηφιολέξεων που έχουν χρόνο μεταξύ αποστολής τα 5ms. Τα αποτελέσματα παρουσιάζονται στον πίνακα 3.20 και γραφικά στο σχήμα 3.25.

Αριθμός Ροών	TCP Drop Rate %	SCTP Drop Rate %
10	0,4670	1,897
25	1,0867	3,9291
50	2,8709	9,3878
75	4,8319	15,2763
100	7,0808	20,9942
150	10,1622	29,3224

Πίνακας 3.20 : Ποσοστό χαμένων πακέτων στη σύνδεση συμφόρησης.



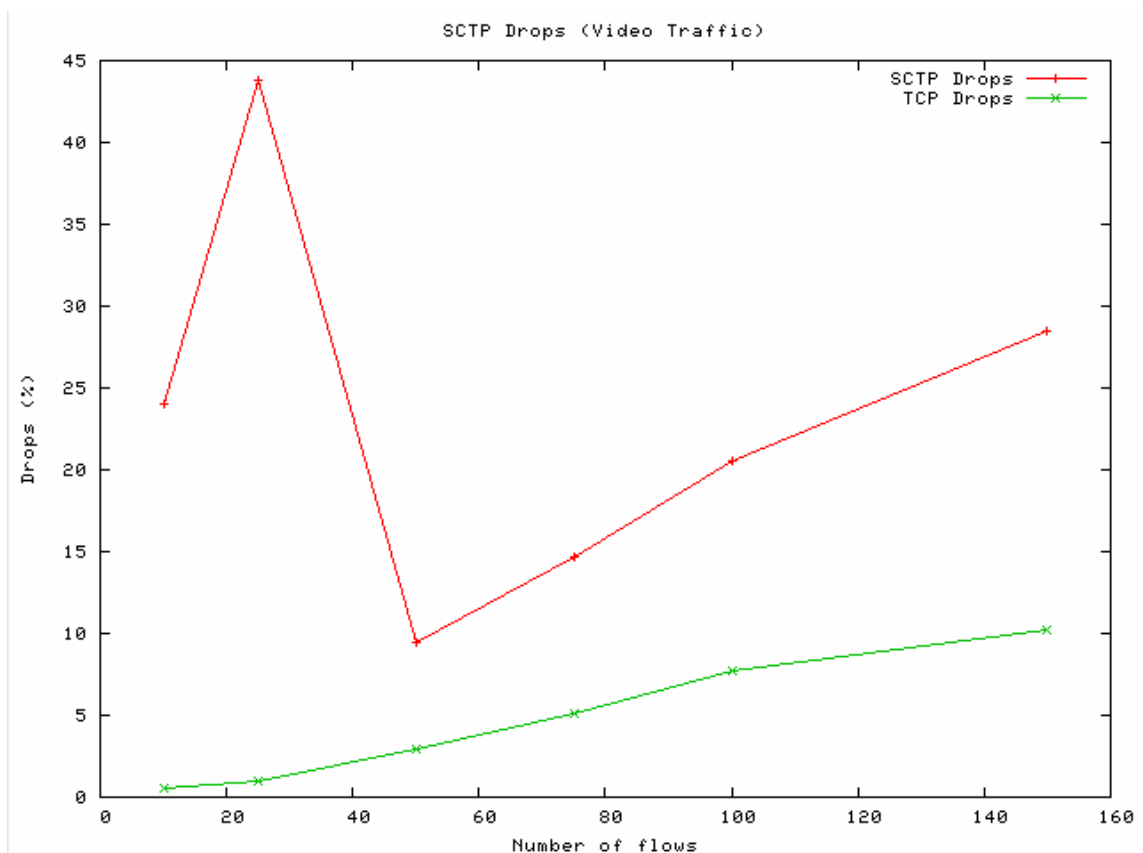
Σχήμα 3.25 : Ποσοστό χαμένων πακέτων στη σύνδεση συμφόρησης συναρτήσει ροών.

Παρατηρούμε πως η μεταβολή στον χρόνο αποστολής των πακέτων δεν επηρεάζει την συμπεριφορά των ροών. Και πάλι παρατηρείται το SCTP να έχει ένα μεγαλύτερο ποσοστό χαμένων πακέτων. Με την αύξηση των ροών παρατηρείται μια μεγαλύτερη αύξηση στο ποσοστό των χαμένων πακέτων του SCTP.

Τέλος όπως και στις προηγούμενες περιπτώσεις εφαρμόζουμε προσομοίωση με την χρήση πραγματικού βίντεο. Τα αποτελέσματα παρουσιάζονται στον πίνακα 3.21 και γραφικά στο σχήμα 3.26.

Αριθμός Ροών	TCP Drop Rate %	SCTP Drop Rate %
10	0,5882	24,0179
25	0,9837	43,7939
50	2,8824	9,5020
75	5,0958	14,6484
100	7,7106	20,5091
150	10,2687	28,51

Πίνακας 3.21 : Ποσοστό χαμένων πακέτων στη σύνδεση συμφόρησης.

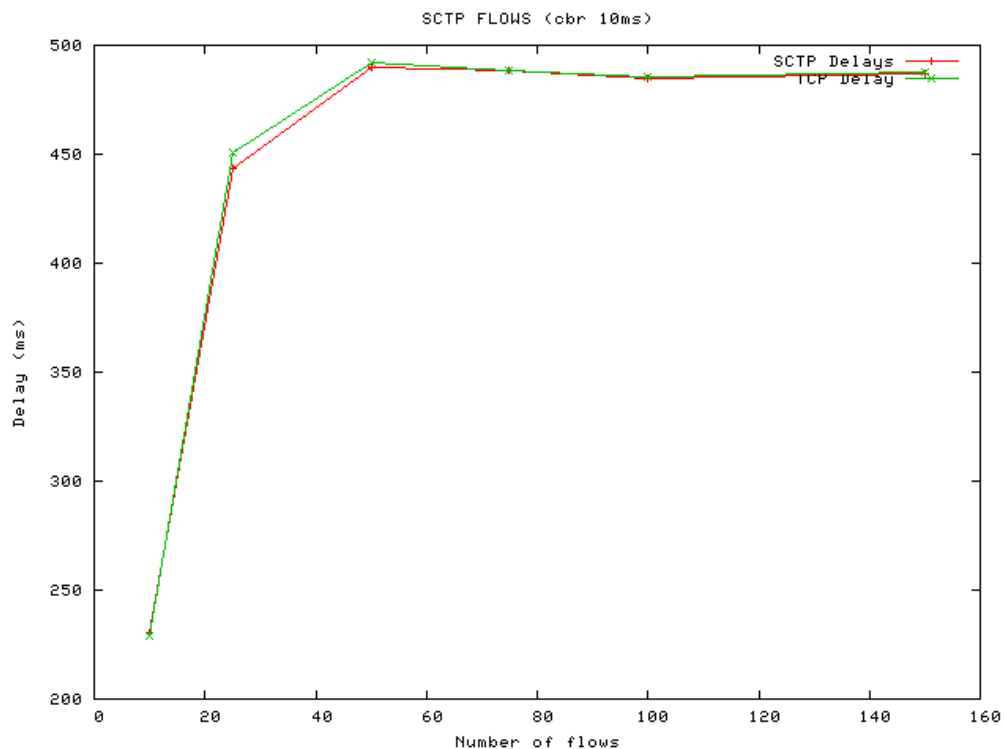


Σχήμα 3.26 : Ποσοστό χαμένων πακέτων στη σύνδεση συμφόρησης συναρτήσει ροών.

Τα αποτελέσματα σε αυτή την περίπτωση διαφέρουν ελαφρά από τις άλλες περιπτώσεις. Σε χαμηλά ποσοστά συμφόρησης το SCTP παρουσιάζει μεγαλύτερο ποσοστό χαμένων πακέτων, ενώ μετά τις 50 ροές μειώνεται απότομα και αυξάνεται γραμμικά. Σε όλες τις περιπτώσεις όμως το ποσοστό χαμένων πακέτων είναι μεγαλύτερο από αυτό του TCP.

Αριθμός Ροών	Mean Delay SCTP	Jitter SCTP	Mean Delay TCP	Jitter TCP
10	230,612	9,33994	229,062	11,9594
25	443,515	76,8557	450,509	74,4374
50	489,732	50,0025	492,3	43,6981
75	488,245	38,2562	488,707	35,7719
100	484,958	39,1647	485,831	37,0824
150	486,955	43,4855	487,733	37,5413

Πίνακας 3.22 : Καθυστέρηση πακέτων στους δρομολογητές.



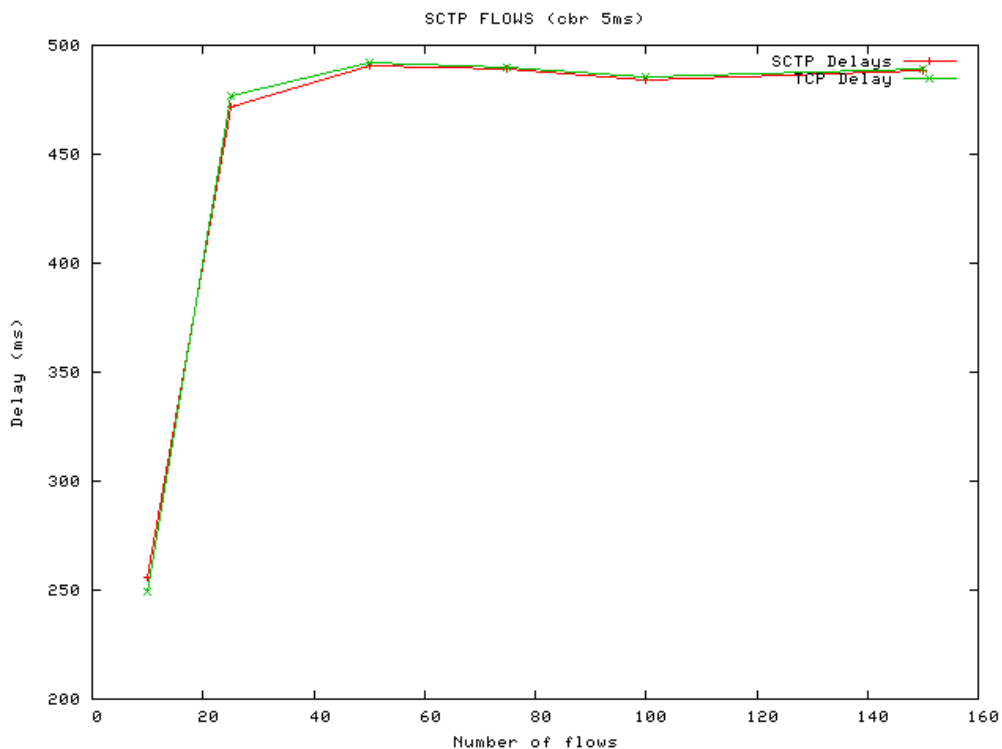
Σχήμα 3.27 : Καθυστέρηση πακέτων στους δρομολογητές.

Το επόμενο πείραμα έχει σκοπό την εύρεση της καθυστέρησης των πακέτων και των δύο ροών που παρατηρείται στους δρομολογητές στη γραμμή όπου υπάρχει η συμφόρηση. Σκοπός του πειράματος αυτού είναι ο έλεγχος εάν τα πακέτα των δύο ροών

τυγχάνουν της ίδιας προτεραιότητας στους δρομολογητές. Αρχικά ελέγχουμε τις μετρήσεις μας με χρήση πακέτων μεγέθους 200 ψηφιολέξεων που ο χρόνος αποστολής μεταξύ των πακέτων είναι 10ms. Τα αποτελέσματα παρουσιάζονται στον πίνακα 3.22 και γραφικά στο σχήμα 3.27. Όπως παρατηρούμε σε όλα τα στάδια της συμφόρησης οι ροές και των δύο πρωτοκόλλων λαμβάνουν περίπου τον ίδιο χρόνο καθυστέρησης. Επαναλαμβάνουμε το ίδιο πείραμα αυτή την φορά κάνοντας χρήση πακέτων μεγέθους 200 ψηφιολέξεων που διαφέρουν στον χρόνο αποστολής τους κατά 5 ms. Τα αποτελέσματα παρουσιάζονται στον πίνακα 3.23 και γραφικά στο σχήμα 3.28.

Αριθμός Ροών	Mean Delay Sctp	Jitter Sctp	Mean Delay TCP	Jitter TCP
10	255,508	49,7429	249,046	51,555
25	472,015	78,2192	477,103	73,0564
50	490,262	46,6659	491,968	44,2351
75	489,066	39,6245	489,782	37,2196
100	483,706	39,6242	485,725	36,9346
150	488,076	43,4882	489,078	36,9526

Πίνακας 3.23 : Καθυστέρηση πακέτων στους δρομολογητές.

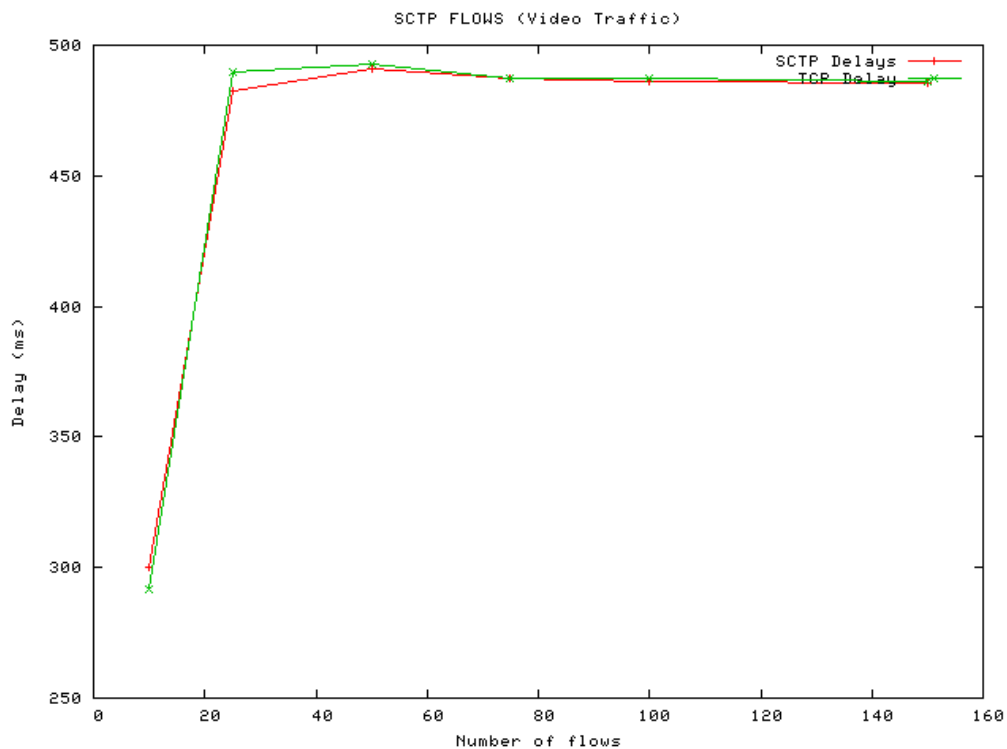


Σχήμα 3.28 : Καθυστέρηση πακέτων στους δρομολογητές.

Τα αποτελέσματα που παράγονται σε αυτή την περίπτωση δεν διαφέρουν κατά πολύ από την περίπτωση όπου ο χρόνος αποστολής μεταξύ των πακέτων ήταν σε πιο ψηλά επίπεδα. Και πάλι οι δύο ροές έχουν σχεδόν την ίδια καθυστέρηση όσο αυξάνεται ο αριθμός των ροών. Δοκιμάζουμε για ακόμα μια φορά την προσομοίωση χρησιμοποιώντας πραγματική ταινία όπως και προηγουμένως. Τα αποτελέσματα παρουσιάζονται στον πίνακα 3.24 και γραφικά στο σχήμα 3.29.

Αριθμός Ροών	Mean Delay Sctp	Jitter Sctp	Mean Delay TCP	Jitter TCP
10	300,389	51,3672	291,644	55,2617
25	482,326	75,1219	489,475	65,558
50	490,877	45,8723	492,75	44,3044
75	487,053	38,3665	487,23	38,0558
100	486,318	38,4089	487,536	37,1473
150	485,254	41,5776	485,816	37,1882

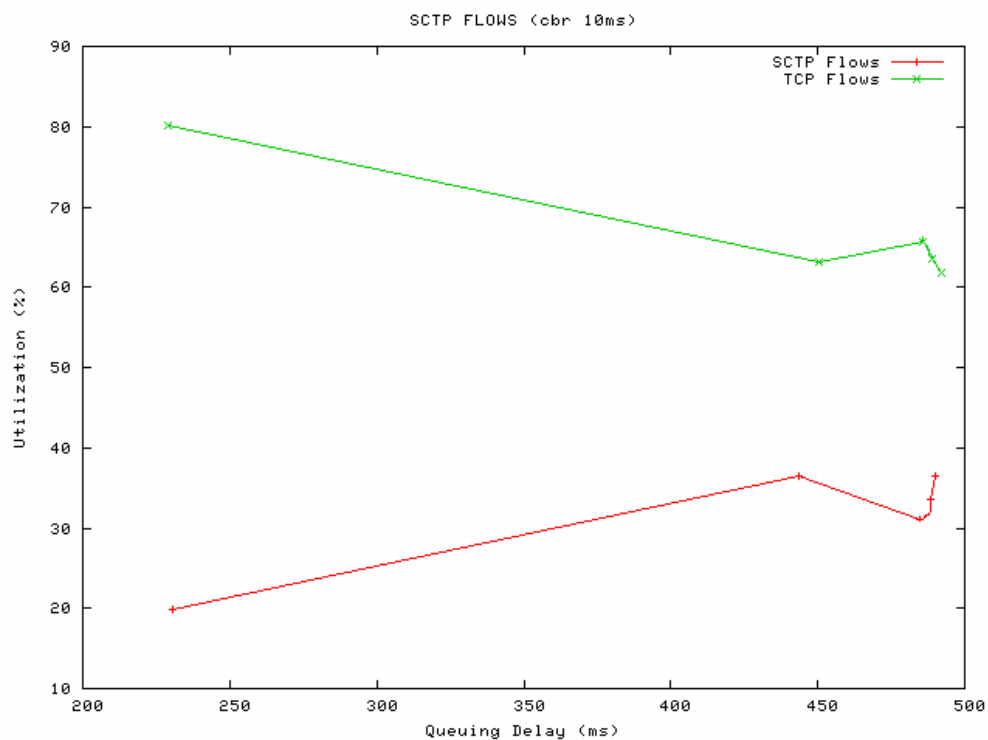
Πίνακας 3.24 : Καθυστέρηση πακέτων στους δρομολογητές.



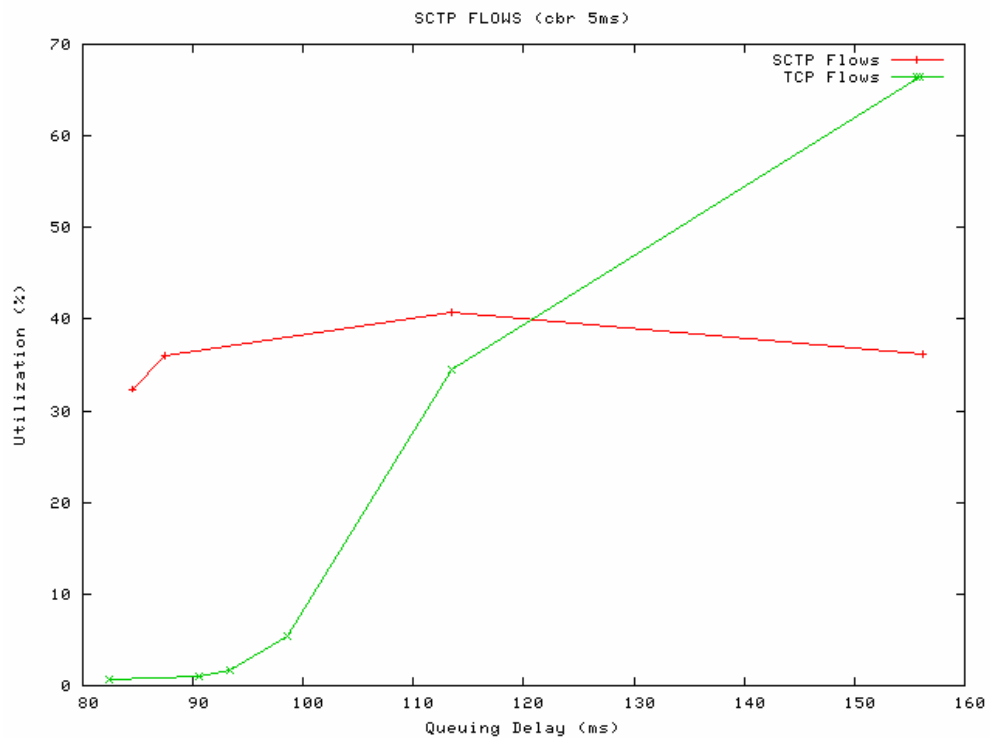
Σχήμα 3.29 : Καθυστέρηση πακέτων στους δρομολογητές.

Παρατηρούμε πως σε αυτή την περίπτωση ισχύουν τα ίδια συμπεράσματα που έχουμε εξάξει και προηγουμένως ότι δηλαδή τα δύο πρωτόκολλα έχουν παρόμοια καθυστέρηση όσο και αν αυξήσουμε τις ροές.

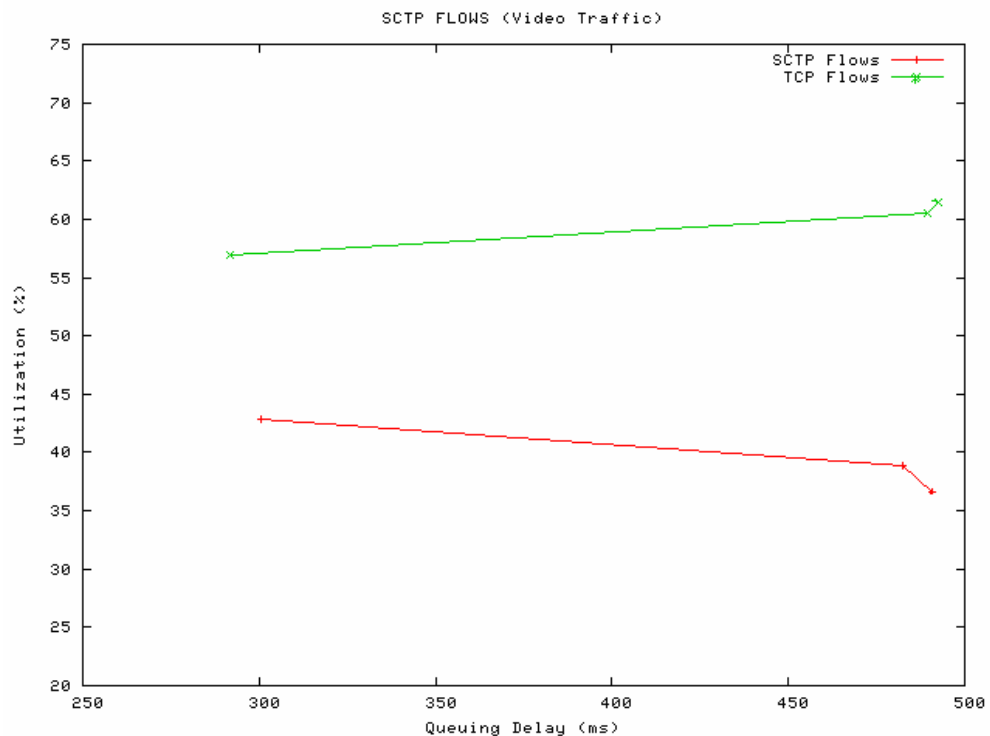
Πιο κάτω παρουσιάζουμε μια σύνοψη των πιο πάνω αποτελεσμάτων των μετρήσεων που έγιναν στην σύνδεση όπου παρατηρείται συμφόρηση. Στα σχήματα 3.30 μέχρι 3.32 παρουσιάζεται το ποσοστό χρήσης της σύνδεσης όπου παρατηρείται συμφόρηση σε συνάρτηση με την καθυστέρηση των πακέτων και των δύο ροών. Το σχήμα 3.30 είναι με χρήση πακέτων με χρόνο αποστολής μεταξύ τους 10ms, το σχήμα 3.31 με χρόνο 5ms και το σχήμα 3.32 με προσομοίωση βίντεο. Σημειώνουμε πως το άθροισμα των δύο ροών ως προς τον άξονα των Ψ δίνει 100% μιας και η χρήση της σύνδεσης είναι ολοκληρωτική.



Σχήμα 3.30 : Χρήση σύνδεσης συμφόρησης σε συνάρτηση του χρόνου καθυστέρησης των πακέτων για πακέτα με χρόνο αποστολής μεταξύ τους 10 ms.



Σχήμα 3.31 : Χρήση σύνδεσης συμφόρησης σε συνάρτηση του χρόνου καθυστέρησης των πακέτων για πακέτα με χρόνο αποστολής μεταξύ τους 5 ms.



Σχήμα 3.32 : Χρήση σύνδεσης συμφόρησης σε συνάρτηση του χρόνου καθυστέρησης των πακέτων για πακέτα με προσομοίωση βίντεο.

Το σημείο όπου διασταυρώνονται οι δύο γραφικές στο σχήμα 3.31 δηλώνει το χρόνο καθυστέρησης πακέτων των δύο ροών όπου το ποσοστό του εύρους ζώνης που καταλαμβάνουν στην σύνδεση είναι ίσο. Ο χρόνος αυτός για το συγκεκριμένο σχήμα είναι γύρω στα 120ms.

Για επιβεβαίωση των υπολογισμών μας τρέξαμε όλες τις προσομοιώσεις μας στην τοπολογία που περιγράψαμε στο σχήμα 3.17. Τα αποτελέσματα δεν διαφέρουν σε μεγάλο βαθμό από τα αποτελέσματα που παρουσιάζουμε. Επίσης μια δεύτερη μετατροπή η οποία έγινε ήταν να σταματούμε τις μισές ροές από κάθε πρωτόκολλο στο τεσσαρακοστό δευτερόλεπτο και να τις ξεκινήσουμε ξανά στο εβδομηκοστό δευτερόλεπτο. Και πάλι τα αποτελέσματα που πήραμε δεν διαφέρουν σε μεγάλο βαθμό από αυτά που έδινε η βασική τοπολογία.

3.4 Προσομοιώσεις με χρήση του RTP

Χρησιμοποιώντας την βασική τοπολογία που έχουμε περιγράψει στο σχήμα 3.0.1 στέλνουμε τις ροές TCP στο μονοπάτι που αποτελείται από τους κόμβους 0, 2, 3, 4 και τις ροές RTP από το μονοπάτι που αποτελείται από τους κόμβους 1, 2, 3, 5. Οι μετρήσεις γίνονται με χρήση 10, 25, 50, 75, 100 και 150 ροών από το κάθε πρωτόκολλο. Οι προσομοιώσεις διαρκούν ένα σταθερό χρόνο που έχουμε ορίσει στα 100 δευτερόλεπτα.

Για την προσομοίωση των ροών RTP χρησιμοποιούμε κόμβους του προσομοιωτή, στους οποίους έχουμε τοποθετήσει πράκτορες RTP. Στους τελικούς κόμβους όπου θα φτάνουν τα πακέτα του πρωτοκόλλου έχουμε τοποθετήσει πράκτορες RTCP οι οποίοι θα στέλνουν πληροφορίες πίσω στον αποστολέα ώστε αυτός με την σειρά του να υπολογίζει την συμφόρηση που υπάρχει στο δίκτυο και να αλλάζει ανάλογα τον ρυθμό αποστολής.

Το πρωτόκολλο έχει δημιουργηθεί για την μεταφορά πολυμεσικά αρχείων και δίνει την δυνατότητα στον αποστολέα να αλλάξει την κωδικοποίηση των πακέτων κατά την διάρκεια της σύνδεσης ώστε να προσαρμόζεται ανάλογα στα επίπεδα συμφόρησης που υπάρχουν. Έχουμε τροποποιήσει κάποιες μεταβλητές ώστε να προσαρμοστούν στις

προσομοιώσεις που χρησιμοποιήσαμε. Πιο συγκεκριμένα έχουμε καθορίσει το μέγεθος των πακέτων της εφαρμογής να είναι 200 ψηφιολέξεις και μεταβάλαμε τον χρόνο αποστολής μεταξύ των πακέτων.

Οι μετρήσεις μας έγιναν για 10ms και για 5ms. Επίσης έγιναν προσομοιώσεις με χρήση πραγματικής ταινίας. Αν για παράδειγμα κάνουμε χρήση πακέτων με μέγεθος 200 ψηφιολέξεις και χρόνο μεταξύ μετάδοσης πακέτων 10ms τότε ο ρυθμός μετάδοσης που αναμένουμε για κάθε ροή RTP θα είναι : $200 * 8 \text{ bits} / 0,01\text{ms} = 160\text{Kbps}$. Τα αποτελέσματα από τις διάφορες μετρήσεις τα αποθηκεύουμε σε αρχεία τα οποία επεξεργαζόμαστε στην συνέχεια.

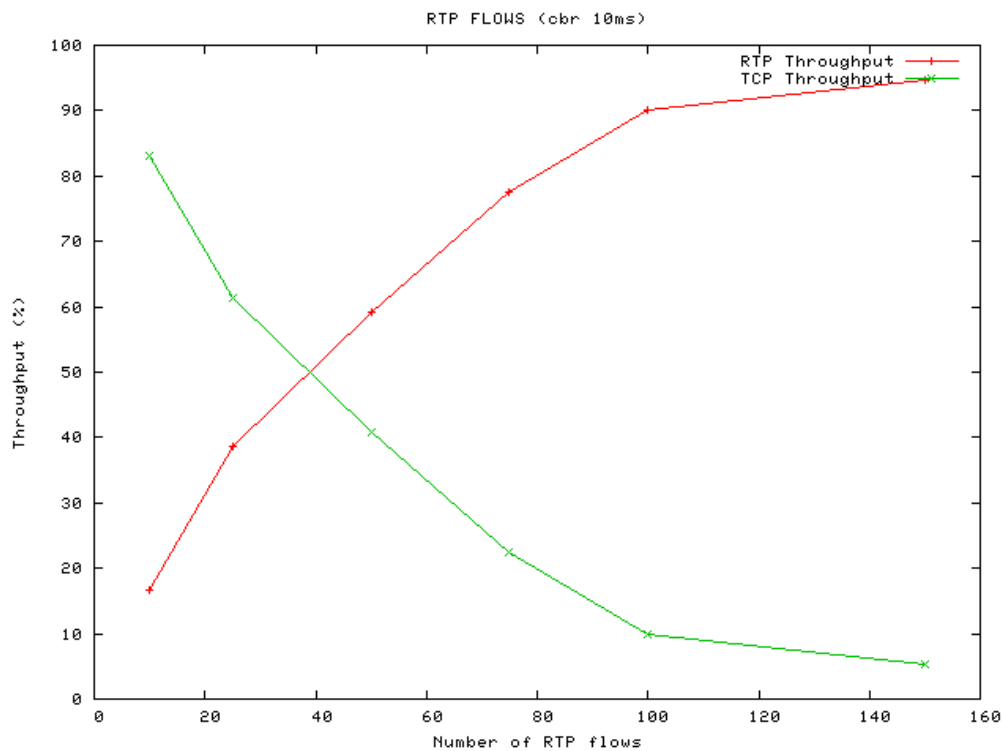
Η πρώτη προσομοίωση αφορά το ποσοστό χρήσης της σύνδεσης από τα δύο πρωτόκολλα στα άκρα. Οι μετρήσεις έγιναν για διαφορετικό αριθμό ροών και τα αποτελέσματα καταγράφονται στον πίνακα 3.25 και παρουσιάζονται γραφικά στο σχήμα 3.33. Τα πακέτα που χρησιμοποιεί το RTP είναι τύπου CBR με διάστημα μεταξύ αποστολής τα 10ms. Επίσης ο πίνακας 3.25 παρουσιάζει το μέσο εύρος ζώνης της σύνδεσης για την κάθε ροή από το κάθε πρωτόκολλο.

Αριθμός Ροών	TCP Throughput %	TCP Throughput Mb	RTP Throughput %	RTP Throughput Mb
10	83,0934	0,8323	16,7743	0,1677
25	61,4083	0,2267	38,528	0,1673
50	40,8498	0,1289	59,1136	0,167
75	22,547	0,070	77,4251	0,167
100	9,903	0,079	90,0732	0,167
150	5,394	0,0119	94,678	0,167

Πίνακας 3.25 : Ποσοστό χρήσης σύνδεσης στα άκρα και μέσος όρος εύρους ζώνης

Παρατηρούμε πως με την αύξηση του αριθμού των ροών από το κάθε πρωτόκολλο το RTP παίρνει μεγαλύτερο ποσοστό από την σύνδεση σε σχέση με το TCP. Αρχικά όλες οι ροές του RTP περιορίζουν τον ρυθμό αποστολής τους σε περίπου 160 Kbps που είναι ακριβώς το εύρος ζώνης που χρειάζεται η εφαρμογή αλλά δεν διεκδικούν όλο το διαθέσιμο εύρος ζώνης όπως κάνουν οι ροές TCP.

Αντίθετα το TCP τείνει να εκμεταλλευτεί όσο το δυνατό περισσότερο εύρος ζώνης γι' αυτό και οι ροές έχουν μεγαλύτερο ρυθμό. Με την αύξηση όμως των ροών και κατ' επέκταση την αύξηση της συμφόρησης, οι ροές RTP διατηρούν τα επιθυμητά 160Kb αλλά εκμεταλλεύονται το διαθέσιμο εύρος ζώνης εις βάρος των ροών TCP τις οποίες οδηγούν σε μηδενισμό του εύρους ζώνης.



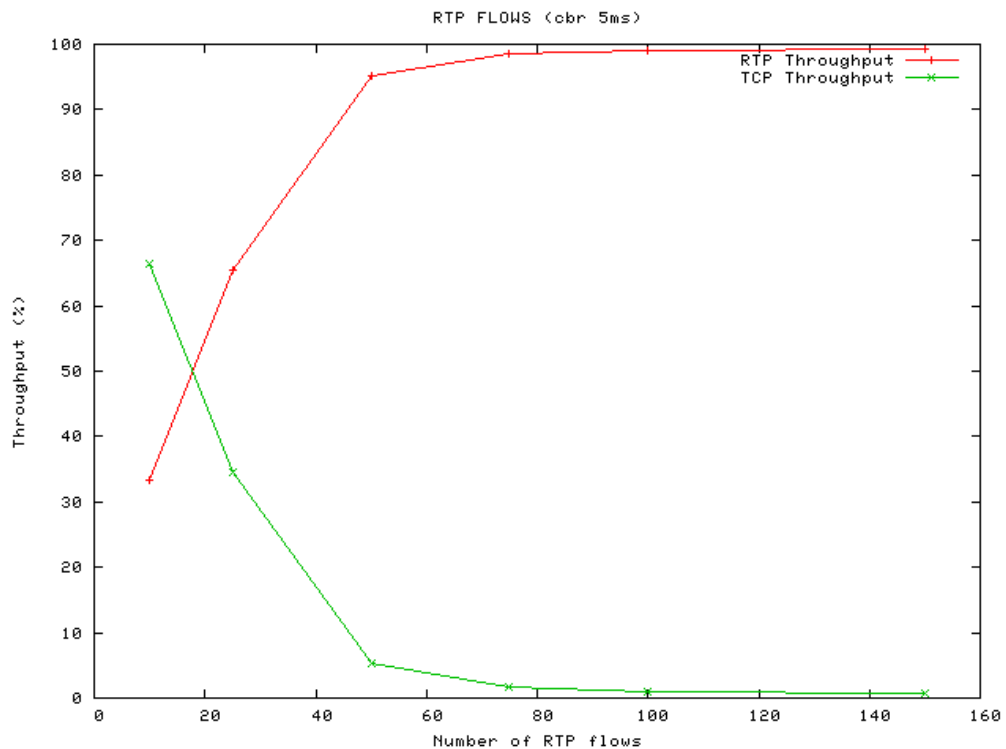
Σχήμα 3.33 : Ποσοστό χρήσης σύνδεσης συναρτήσει του αριθμού ροών TCP και RTP

Αριθμός Ροών	TCP Throughput %	TCP Throughput Mb	RTP Throughput %	RTP Throughput Mb
10	66,4613	0,6754	33,437	0,3352
25	34,531	0,1870	65,4308	0,3344
50	5,35	0,1455	95,1764	0,3351
75	1,6367	0,0436	98,55	0,3353
100	0,969	0,0013	99,002	0,3350
150	0,6767	0,094	99,3172	0,3349

Πίνακας 3.26 : Ποσοστό χρήσης σύνδεσης στα άκρα και μέσος όρος εύρους ζώνης

Επαναλαμβάνουμε την ίδια διαδικασία αλλά αυτή την φορά μειώνουμε τον χρόνο μεταξύ της κάθε αποστολής πακέτων της εφαρμογής στα 5ms. Οι μετρήσεις έγιναν για διαφορετικό αριθμό ροών και τα αποτελέσματα καταγράφονται στον πίνακα 3.26 και

παρουσιάζονται γραφικά στο σχήμα 3.34. Επίσης ο πίνακας 3.26 παρουσιάζει το μέσο εύρος ζώνης της σύνδεσης για την κάθε ροή από το κάθε πρωτόκολλο.



Σχήμα 3.34 : Ποσοστό χρήσης σύνδεσης συναρτήσει του αριθμού ροών TCP και RTP

Παρατηρούμε όπως και στην προηγούμενη περίπτωση ότι οι ροές RTP αρχικά καταλαμβάνουν το υπολογιζόμενο εύρος ζώνης που δικαιούνται, και με την αύξηση των ροών και της συμφόρησης εκμεταλλεύονται το εύρος ζώνης εις βάρος του TCP χωρίς οι ροές αυτές να παίρνουν λιγότερο από το υπολογιζόμενο εύρος ζώνης.

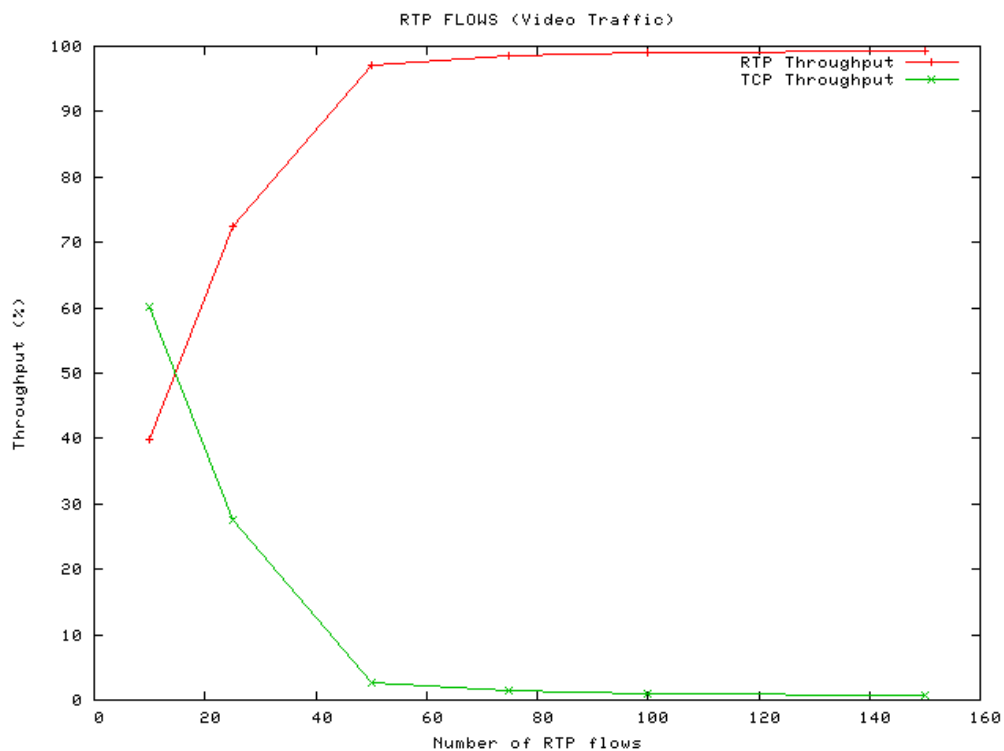
Τέλος δοκιμάζουμε την ίδια διαδικασία αλλά αντί για χρήση πακέτων εφαρμογής προσομοιώνουμε με την χρήση πραγματικού βίντεο. Το βίντεο που αρχικά είναι χωρισμένο σε διάφορα πλαίσια ταινίας μετατρέπεται σε πακέτα, με χρήση ενός μικρού προγράμματος το οποίο μετατρέπει τα πλαίσια αυτά σε πακέτα αποστολής σταθερού ρυθμού.

Όπως αναφέρει ο προγραμματιστής του προγράμματος αυτού η προσομοίωση θα είναι παρόμοια εάν είχαμε χρήση πακέτων με μέγεθος 200 ψηφιολέξεις που θα είχαν διαφορά

χρόνου στην αποστολή τους 10ms. Τα αποτελέσματα της διαδικασίας παρουσιάζονται στον πίνακα 3.27 και η γραφική απεικόνιση στο σχήμα 3.34.

Αριθμός Ροών	TCP Throughput %	TCP Throughput Mb	RTP Throughput %	RTP Throughput Mb
10	60,0269	0,6393	39,8932	0,399
25	27,4569	0,1274	72,5164	0,2691
50	2,7510	0,0107	97,2212	0,2129
75	1,382	0,1982	98,5876	0,16217
100	0,898	0,0031	99,0844	0,0898
150	0,6965	0,0006	99,274	0,05761

Πίνακας 3.27 : Ποσοστό χρήσης σύνδεσης στα άκρα και μέσος όρος εύρους ζώνης



Σχήμα 3.34 : Ποσοστό χρήσης σύνδεσης συναρτήσει του αριθμού ροών TCP και RTP

Όπως και στις προηγούμενες περιπτώσεις έτσι και εδώ παρατηρούμε πως η προσομοίωση με την χρήση πραγματικού βίντεο, δίνει περίπου τα ίδια αποτελέσματα με αυτά που παίρνουμε και στις άλλες περιπτώσεις. Συμπεραίνουμε τελικά πως στην ύπαρξη συμφόρησης οι ροές RTP εκμεταλλεύονται το εύρος ζώνης της σύνδεσης σε βάρος του TCP.

Για την δεύτερη μας μέτρηση θα εξετάσουμε την καθυστέρηση που παρουσιάζουν τα πακέτα σε συνάρτηση με τον αριθμό των ροών του RTP. Από τις προηγούμενες μετρήσεις παρατηρήσαμε πως οι ροές RTP κερδίζουν σε εύρος ζώνης εις βάρος του TCP. Ένα λογικό σενάριο θα ήταν οι ροές RTP να παρουσιάζουν λιγότερη καθυστέρηση στα άκρα ώστε να κερδίζουν σε εύρος ζώνης.

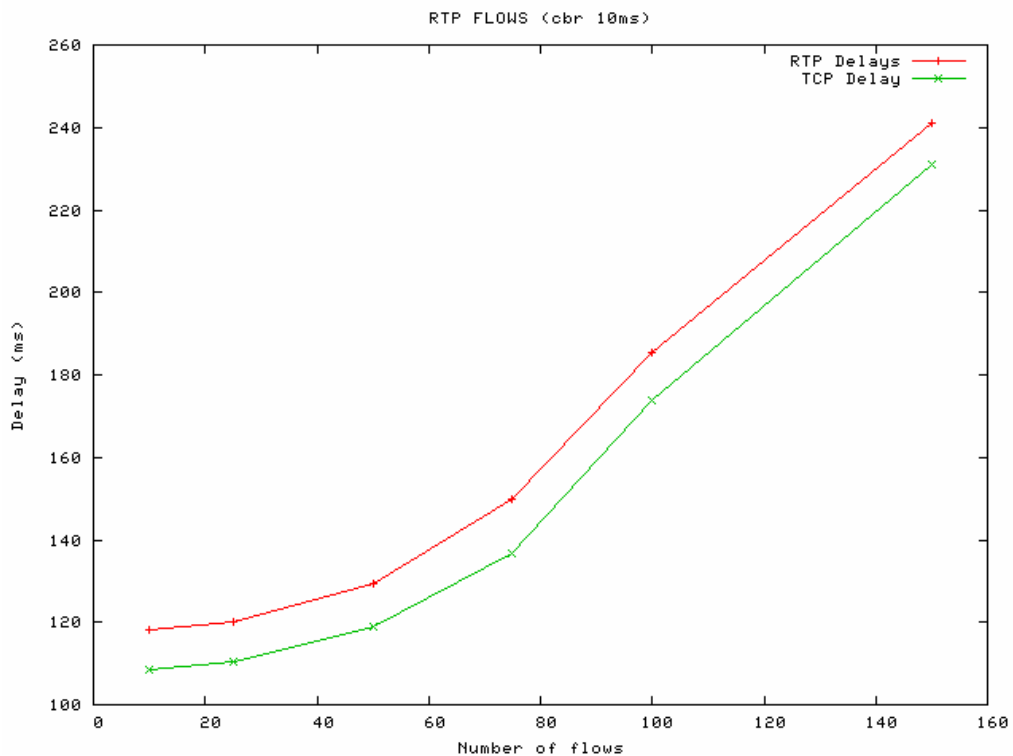
Πιο κάτω παρουσιάζουμε τα αποτελέσματα των μετρήσεων της καθυστέρησης σε συνάρτηση με τον αριθμό των ροών για τα δύο πρωτόκολλα. Αρχικά παρουσιάζουμε τα αποτελέσματα με την χρήση πακέτων εφαρμογής με μέγεθος 200 ψηφιολέξεις και χρόνο μεταξύ αποστολής τα 10ms.

Τα αποτελέσματα παρουσιάζονται στον πίνακα 3.28 και γραφικά στο σχήμα 3.35. Επίσης ο πίνακας περιέχει το ποσοστό των πακέτων που χάνονται από την έξοδο της σύνδεσης που παρουσιάζεται συμφόρηση μέχρι τον παραλήπτη, καθώς και η διαφορά χρόνου στον οποίο κάποια πακέτα φτάνουν στον παραλήπτη μετά που ακολούθησαν διαφορετική διαδρομή (Jitter).

Αριθμός Ροών	RTP Drop Rate %	Mean Delay RTP	Mean Delay TCP	Jitter RTP
10	0,16	118,231	116,285	6,760
25	18,6	120,114	116,248	13,687
50	29,62	129,346	120,458	10,5312
75	51,75	149,888	145,058	11,447
100	61,286	161,879	158,158	12,265
150	61,68	185,51	181,185	7,781

Πίνακας 3.28 : Μέση καθυστέρηση πακέτων στα άκρα για το RTP.

Παρατηρούμε πως με την αύξηση των ροών η καθυστέρηση που παρουσιάζουν οι ροές RTP αυξάνεται γραμμικά μέχρι ενός σημείου και από εκεί και πέρα η καθυστέρηση αυξάνεται εκθετικά. Αυτός είναι και ο λόγος που κερδίζουν περισσότερο εύρος ζώνης εις βάρος του TCP. Οι ροές του TCP με την σειρά τους αυξάνουν την καθυστέρηση όσο αυξάνονται οι ροές, επομένως και χάνουν εύρος ζώνης.



Σχήμα 3.35 : Καθυστέρηση πακέτων σε συνάρτηση του αριθμού ροών RTP.

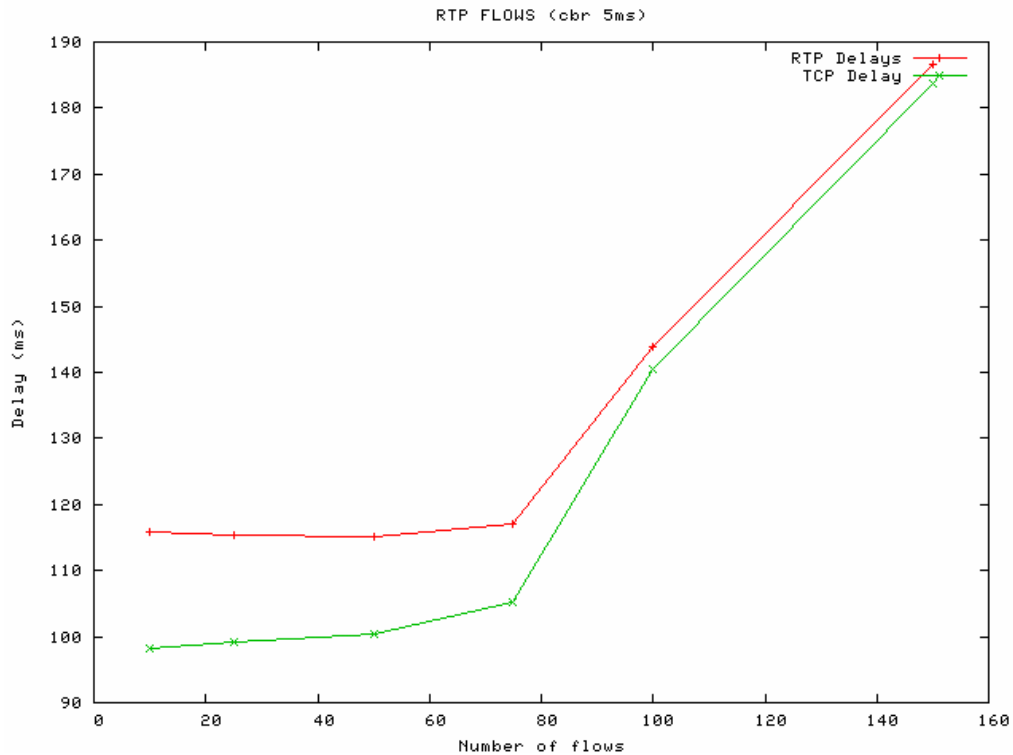
Επαναλαμβάνουμε την ίδια διαδικασία αλλά αυτή την φορά κάνουμε χρήση πακέτων μεγέθους 200 ψηφιολέξεων με χρόνο μεταξύ αποστολής τα 5ms. Τα αποτελέσματα παρουσιάζονται στον πίνακα 3.29 και γραφικά στο σχήμα 3.36.

Επίσης ο πίνακας περιέχει το ποσοστό των πακέτων που χάνονται από την έξοδο της σύνδεσης που παρουσιάζεται συμφόρηση μέχρι τον παραλήπτη, καθώς και η διαφορά χρόνου στον οποίο κάποια πακέτα φτάνουν στον παραλήπτη μετά που ακολούθησαν διαφορετική διαδρομή (Jitter).

Αριθμός Ροών	RTP Drop Rate %	Mean Delay RTP	Mean Delay TCP	Jitter RTP
10	0,476	115,807	113,458	18,265
25	40,933	115,455	113,584	7,574
50	62,84	115,047	113,524	7,869
75	61,30	117,097	115,284	3,999
100	61,34	143,747	141,587	3,781
150	90,5	186,58	183,365	3,397

Πίνακας 3.29 : Μέση καθυστέρηση πακέτων στα άκρα για το RTP.

Παρατηρούμε σε σύγκριση με τα προηγούμενα αποτελέσματα πως η καθυστέρηση κυμαίνεται στα ίδια περίπου επίπεδα και πως η αλλαγή στο χρόνο αποστολής μεταξύ των πακέτων δεν επιφέρει οποιαδήποτε ουσιαστική αλλαγή στα αποτελέσματα.



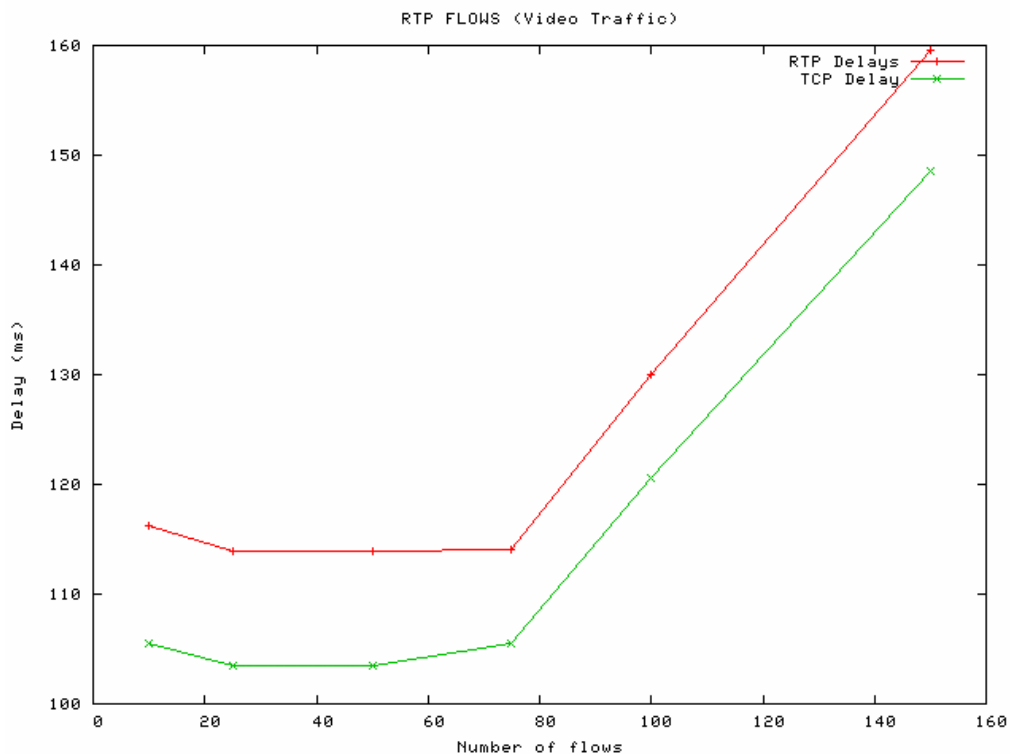
Σχήμα 3.36 : Καθυστέρηση πακέτων σε συνάρτηση του αριθμού ροών RTP.

Αριθμός Ροών	RTP Drop Rate %	Mean Delay RTP	Mean Delay TCP	Jitter RTP
10	2,636	113,883	112,458	15,035
25	36,85	113,881	111,284	8,303
50	61,22	114,006	112,548	4,379
75	71,286	116,278	114,128	3,525
100	70,243	130,006	128,268	2,578
150	87,73	159,61	147,158	3,759

Πίνακας 3.30 : Μέση καθυστέρηση πακέτων στα άκρα για το RTP.

Τέλος προσομοιώνουμε με την χρήση αποστολής κανονικής ταινίας όπως ακριβώς και το προηγούμενο πείραμα. Τα αποτελέσματα παρουσιάζονται στον πίνακα 3.30 και γραφικά στο σχήμα 3.37. Επίσης ο πίνακας περιέχει το ποσοστό των πακέτων που χάνονται από την έξοδο της σύνδεσης που παρουσιάζεται συμφόρηση μέχρι τον

παραλήπτη, καθώς και η διαφορά χρόνου στον οποίο κάποια πακέτα φτάνουν στον παραλήπτη μετά που ακολούθησαν διαφορετική διαδρομή (Jitter).



Σχήμα 3.37 : Καθυστέρηση πακέτων σε συνάρτηση του αριθμού ροών RTP.

Συμπεραίνουμε πως η χρήση του βίντεο δεν προκαλεί μεγάλες αλλαγές στην καθυστέρηση που παρατηρείται στα πακέτα RTP. Συγκρίνοντας τα αποτελέσματα των μέχρι στιγμής πειραμάτων βλέπουμε πως επειδή η καθυστέρηση των πακέτων RTP είναι μικρότερη από αυτή του TCP, οι ροές του RTP παίρνουν μεγαλύτερο εύρος ζώνης.

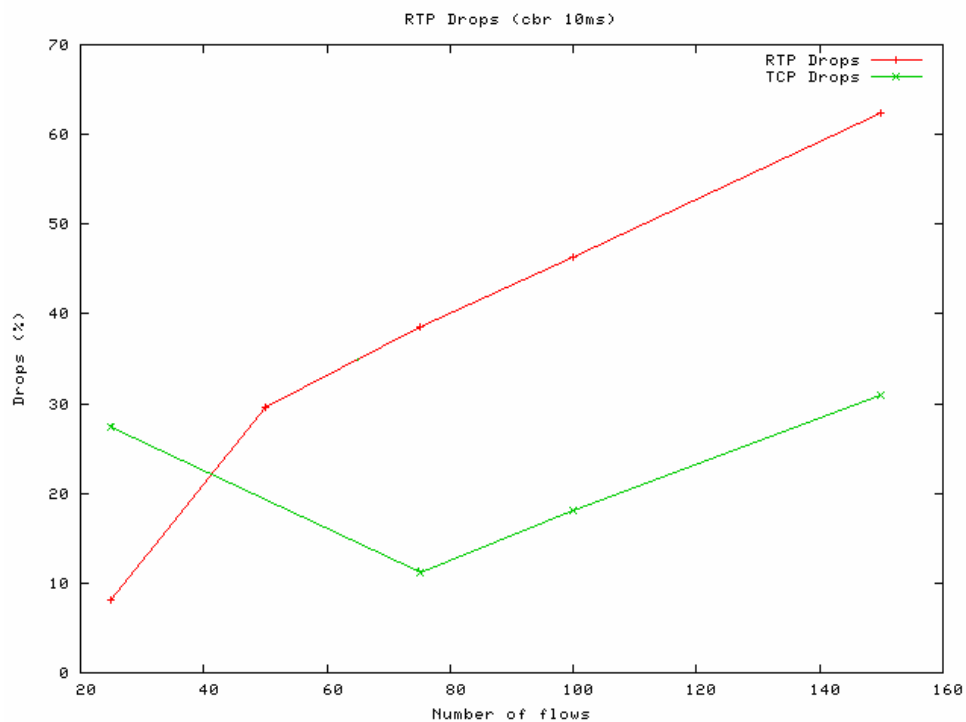
Για τα επόμενα πειράματα χρησιμοποιούμε μετρικές τις οποίες λαμβάνουμε στην γραμμή από την οποία υπάρχει συμφόρηση. Το πρώτο τέτοιο πείραμα αφορούσε το ποσοστό της χρήσης της γραμμής στην οποία υπάρχει συμφόρηση σε συνάρτηση με τον αριθμό των ροών του κάθε πρωτοκόλλου.

Λόγω του ότι το εύρος ζώνης της σύνδεσης είναι 10 φορές μικρότερο από αυτό που ενώνει την έξοδο της σύνδεσης με τον παραλήπτη, δεν παρατηρείται χάσιμο πακέτων στην γραμμή αυτή. Επομένως τα αποτελέσματα αυτής της μέτρησης ταυτίζονται με τα

αποτελέσματα του πειράματος που αφορά την χρήση της σύνδεσης στον παραλήπτη σε συνάρτηση με τον αριθμό των ροών. Σχηματικά τα αποτελέσματα είναι τα ίδια με αυτά των σχημάτων 3.35 μέχρι 3.37. Το επόμενο πείραμα που αφορά την σύνδεση συμφόρησης αφορά το χάσιμο των πακέτων των διαφόρων ροών σε συνάρτηση με τον αριθμό των ροών. Θα περιμέναμε εδώ να δούμε πως οι ροές που έχουν το μεγαλύτερο εύρος ζώνης να έχουν και το μεγαλύτερο ποσοστό χαμένων πακέτων. Στο πρώτο πείραμα χρησιμοποιούμε πακέτα εφαρμογής μεγέθους 200 ψηφιολέξεων και χρόνο μεταξύ αποστολής των πακέτων 10ms. Τα αποτελέσματα παρουσιάζονται στον πίνακα 3.31 και γραφικά στο σχήμα 3.38.

Αριθμός Ροών	TCP Drop Rate %	RTP Drop Rate %
10	-	-
25	27,333	8,099
50	69,9742	29,5278
75	11,1827	38,4725
100	18,0409	46,3235
150	30,9813	62,3873

Πίνακας 3.31 : Ποσοστό χαμένων πακέτων στη σύνδεση συμφόρησης.

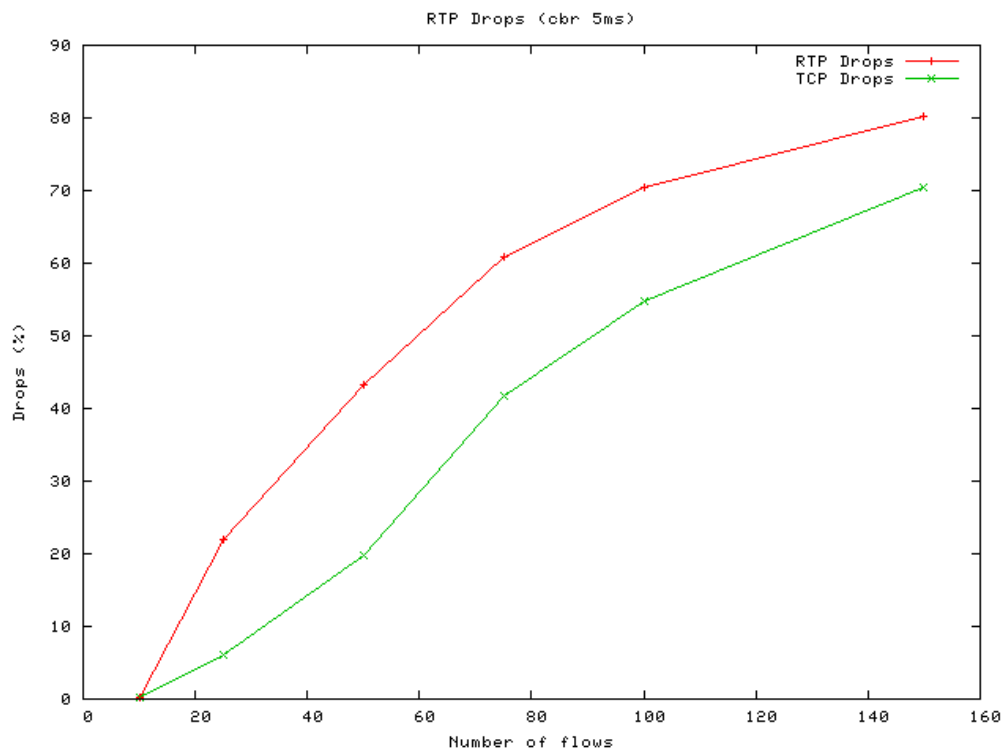


Σχήμα 3.38 : Ποσοστό χαμένων πακέτων στη σύνδεση συμφόρησης συναρτήσει ροών.

Παρατηρούμε πως οι ροές RTP έχουν μεγαλύτερο ποσοστό χαμένων πακέτων κατά την διάρκεια των προσομοιώσεων. Οι γραφικές των πρωτοκόλλων είναι σχεδόν παράλληλες μετά από 50 ροές από κάθε πρωτόκολλο αλλά η διαφορά κλίμακας είναι πολύ μεγαλύτερη. Περίπου τα 2/3 των πακέτων RTP χάνονται σε αντίθεση με το 1/3 του TCP. Αυτός είναι και ο λόγος που οι ροές RTP διατηρούν το αρχικό τους εύρος ζώνης. Επαναλαμβάνουμε το ίδιο πείραμα όπως και στην προηγούμενη περίπτωση αλλά αυτή την φορά χρησιμοποιώντας πακέτα μεγέθους 200 ψηφιολέξεων που έχουν χρόνο μεταξύ αποστολής τα 5ms. Τα αποτελέσματα παρουσιάζονται στον πίνακα 3.32 και γραφικά στο σχήμα 3.39.

Αριθμός Ροών	TCP Drop Rate %	RTP Drop Rate %
10	0,1679	0,3040
25	6,1429	21,9966
50	19,8833	43,2823
75	41,6614	60,8492
100	54,7	70,5023
150	70,4615	80,2721

Πίνακας 3.32 : Ποσοστό χαμένων πακέτων στη σύνδεση συμφόρησης.



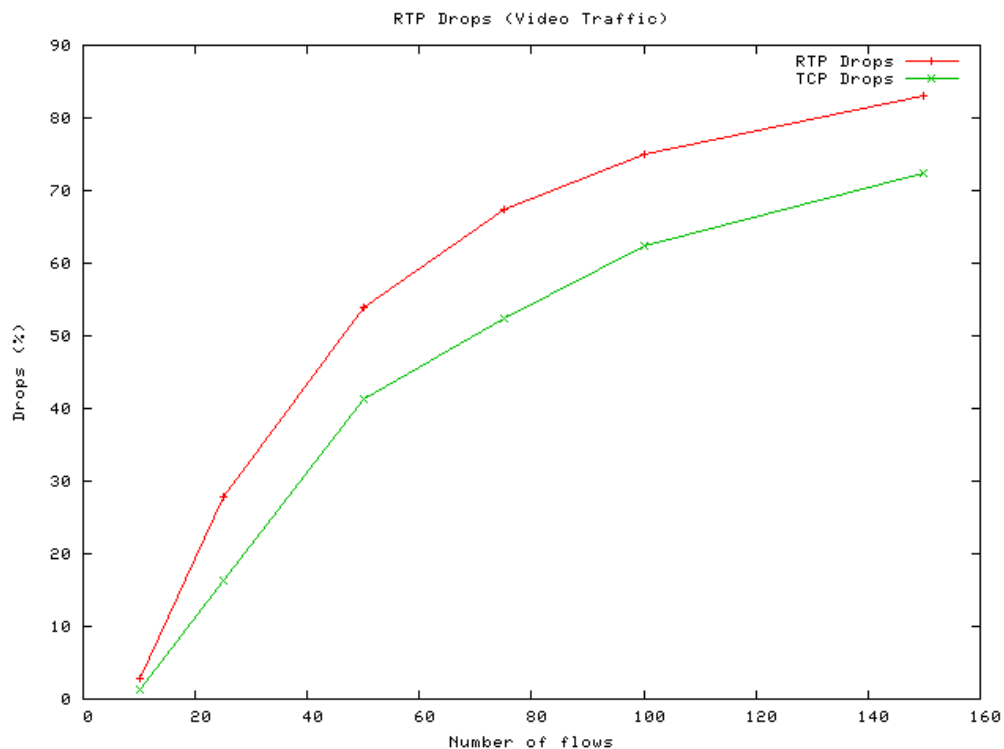
Σχήμα 3.39 : Ποσοστό χαμένων πακέτων στη σύνδεση συμφόρησης συναρτήσει ροών.

Παρατηρούμε πως η μεταβολή στον χρόνο αποστολής των πακέτων επηρεάζει την συμπεριφορά των ροών. Και πάλι παρατηρείται το RTP να έχει ένα μεγαλύτερο ποσοστό χαμένων πακέτων. Όμως το ποσοστό αυτό είναι αρκετά μικρότερο από την προηγούμενη περίπτωση και τείνει να είναι 20% μεγαλύτερο από όλες τις μετρήσεις συγκριτικά με το TCP.

Τέλος όπως και στις προηγούμενες περιπτώσεις εφαρμόζουμε προσομοίωση με την χρήση πραγματικού βίντεο. Τα αποτελέσματα παρουσιάζονται στον πίνακα 3.33 και γραφικά στο σχήμα 3.40.

Αριθμός Ροών	TCP Drop Rate %	RTP Drop Rate %
10	1,2982	2,89821
25	16,2031	27,8318
50	41,2974	53,9823
75	52,4477	67,3396
100	62,4532	75,103
150	72,4185	83,1082

Πίνακας 3.33 : Ποσοστό χαμένων πακέτων στη σύνδεση συμφόρησης.

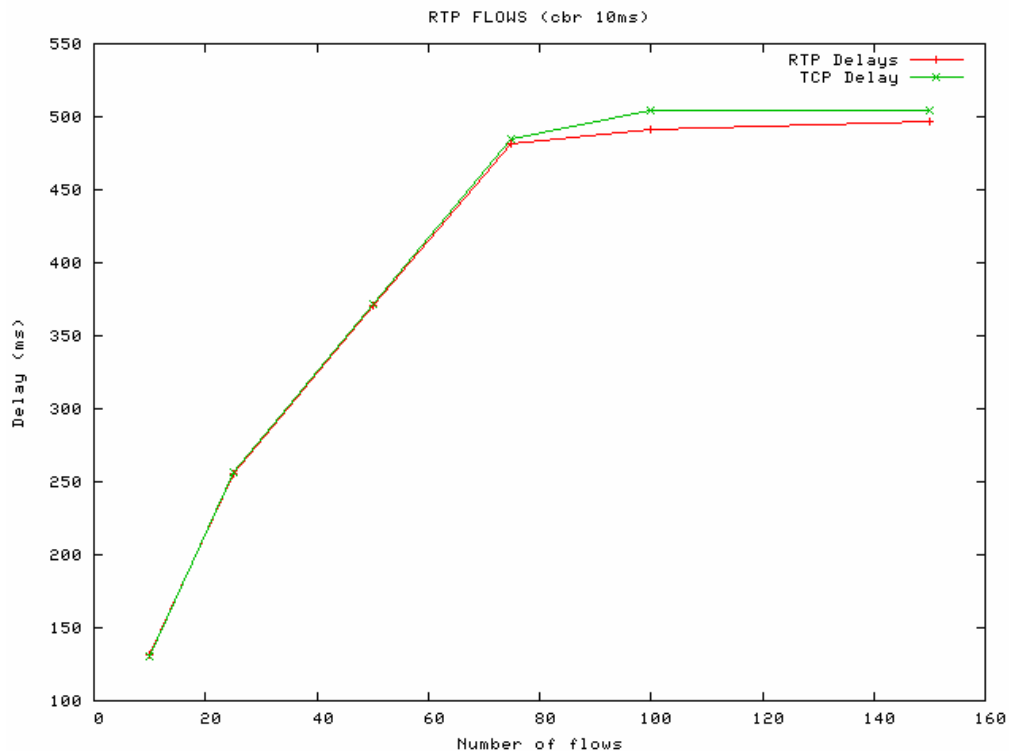


Σχήμα 3.40 : Ποσοστό χαμένων πακέτων στη σύνδεση συμφόρησης συναρτήσει ροών.

Τα αποτελέσματα σε αυτή την περίπτωση δεν διαφέρουν από την προηγούμενη περίπτωση. Παρατηρείται το RTP να έχει ένα μεγαλύτερο ποσοστό χαμένων πακέτων και το ποσοστό αυτό να αυξάνεται περίπου με την ίδια ένταση που αυξάνεται και ο ρυθμός χαμένων πακέτων του TCP.

Αριθμός Ροών	Mean Delay RTP	Jitter RTP	Mean Delay TCP	Jitter TCP
10	131,652	6,779	130,943	5,8311
25	255,57	13,7407	256,552	15,1443
50	320,516	11,3561	321,394	14,1982
75	481,139	13,7265	484,693	22,3055
100	490,9588	16,3304	494,852	28,8311
150	486,7078	11,7288	494,296	24,5534

Πίνακας 3.34 : Καθυστέρηση πακέτων στους δρομολογητές.



Σχήμα 3.41 : Καθυστέρηση πακέτων στους δρομολογητές.

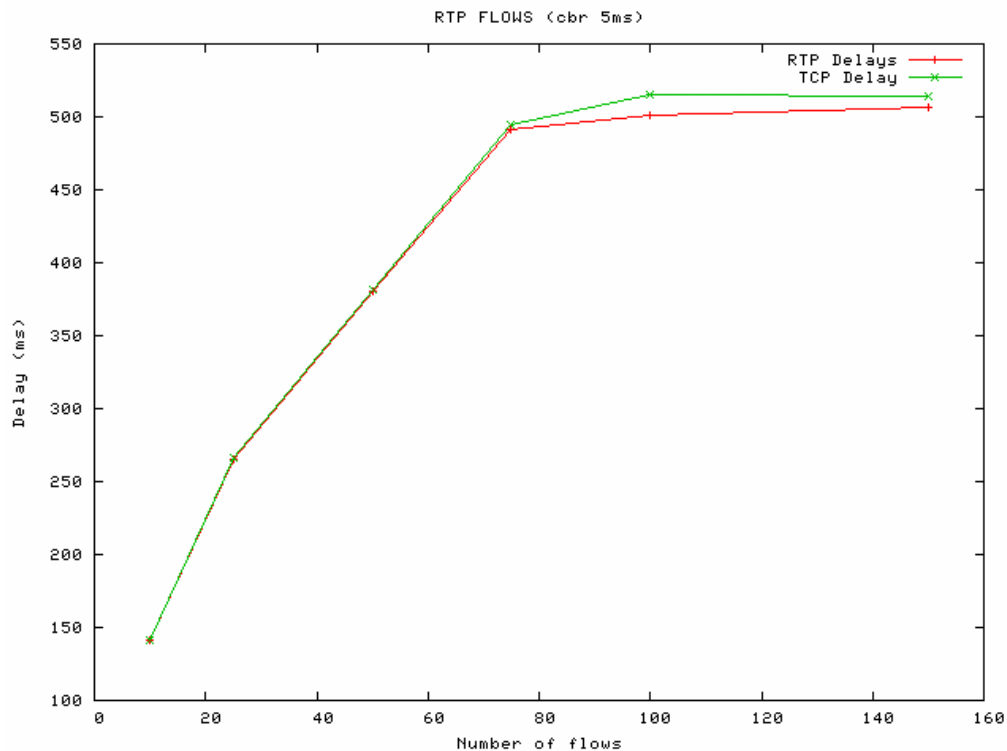
Το επόμενο πείραμα έχει σκοπό την εύρεση της καθυστέρησης των πακέτων και των δύο ροών που παρατηρείται στους δρομολογητές στη γραμμή όπου υπάρχει η συμφόρηση. Σκοπός του πειράματος αυτού είναι ο έλεγχος εάν τα πακέτα των δύο ροών τυγχάνουν της ίδιας προτεραιότητας στους δρομολογητές. Αρχικά ελέγχουμε τις

μετρήσεις μας με χρήση πακέτων μεγέθους 200 ψηφιολέξεων που ο χρόνος αποστολής μεταξύ των πακέτων είναι 10ms. Τα αποτελέσματα παρουσιάζονται στον πίνακα 3.34 και γραφικά στο σχήμα 3.41.

Όπως παρατηρούμε σε όλα τα στάδια της συμφόρησης οι ροές και των δύο πρωτοκόλλων λαμβάνουν περίπου τον ίδιο χρόνο καθυστέρησης. Επαναλαμβάνουμε το ίδιο πείραμα αυτή την φορά κάνοντας χρήση πακέτων μεγέθους 200 ψηφιολέξεων που διαφέρουν στον χρόνο αποστολής τους κατά 5 ms. Τα αποτελέσματα παρουσιάζονται στον πίνακα 3.35 και γραφικά στο σχήμα 3.42.

Αριθμός Ροών	Mean Delay RTP	Jitter RTP	Mean Delay TCP	Jitter TCP
10	156,211	18,2711	155,912	18,0743
25	213,444	7,62799	213,463	8,847
50	387,4105	9,44194	398,5782	21,4469
75	484,5404	6,59403	493,3162	22,8658
100	484,5009	6,56399	490,5627	28,5986
150	484,8135	6,69618	482,3851	35,8129

Πίνακας 3.35 : Καθυστέρηση πακέτων στους δρομολογητές.

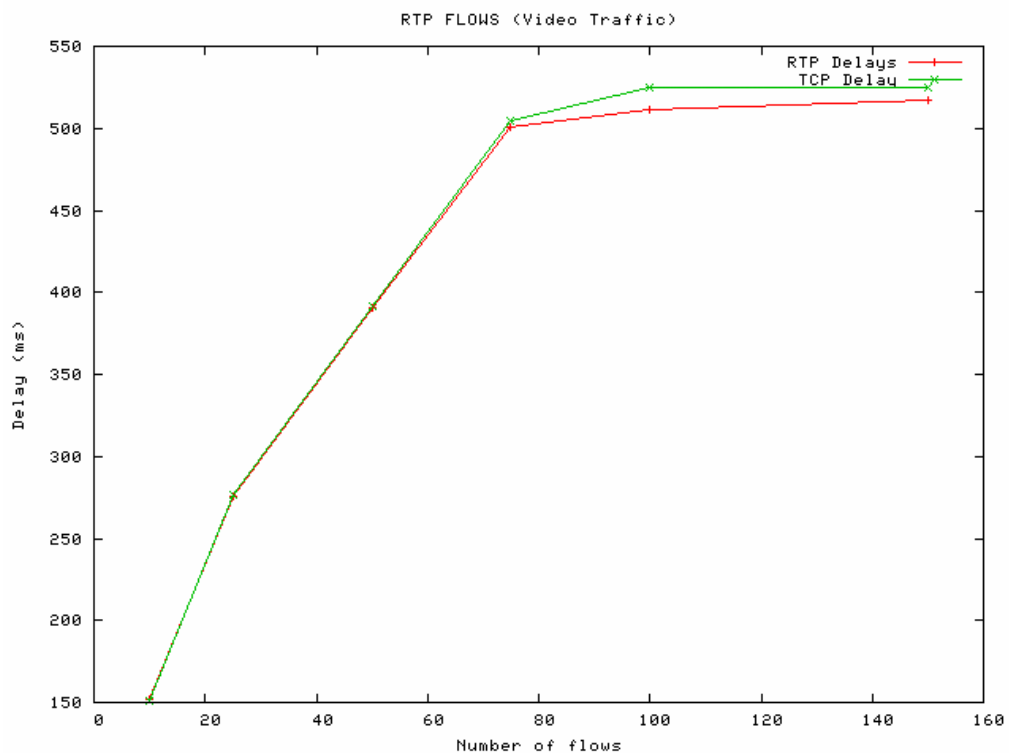


Σχήμα 3.42 : Καθυστέρηση πακέτων στους δρομολογητές.

Τα αποτελέσματα που παράγονται σε αυτή την περίπτωση δεν διαφέρουν κατά πολύ από την περίπτωση όπου ο χρόνος αποστολής μεταξύ των πακέτων ήταν σε πιο ψηλά επίπεδα. Και πάλι οι δύο ροές έχουν σχεδόν την ίδια καθυστέρηση όσο αυξάνεται ο αριθμός των ροών. Δοκιμάζουμε για ακόμα μια φορά την προσομοίωση χρησιμοποιώντας πραγματική ταινία όπως και προηγουμένως. Τα αποτελέσματα παρουσιάζονται στον πίνακα 3.36 και γραφικά στο σχήμα 3.43.

Αριθμός Ροών	Mean Delay RTP	Jitter RTP	Mean Delay TCP	Jitter TCP
10	130,509	15,6931	131,287	15,6192
25	97,009	9,25495	95,7411	8,86424
50	84,1057	4,78727	83,338	11,4624
75	83,9106	5,13148	82,712	18,9821
100	84,3416	4,98758	80,5018	25,1463
150	84,6463	5,43886	75,289	31,2815

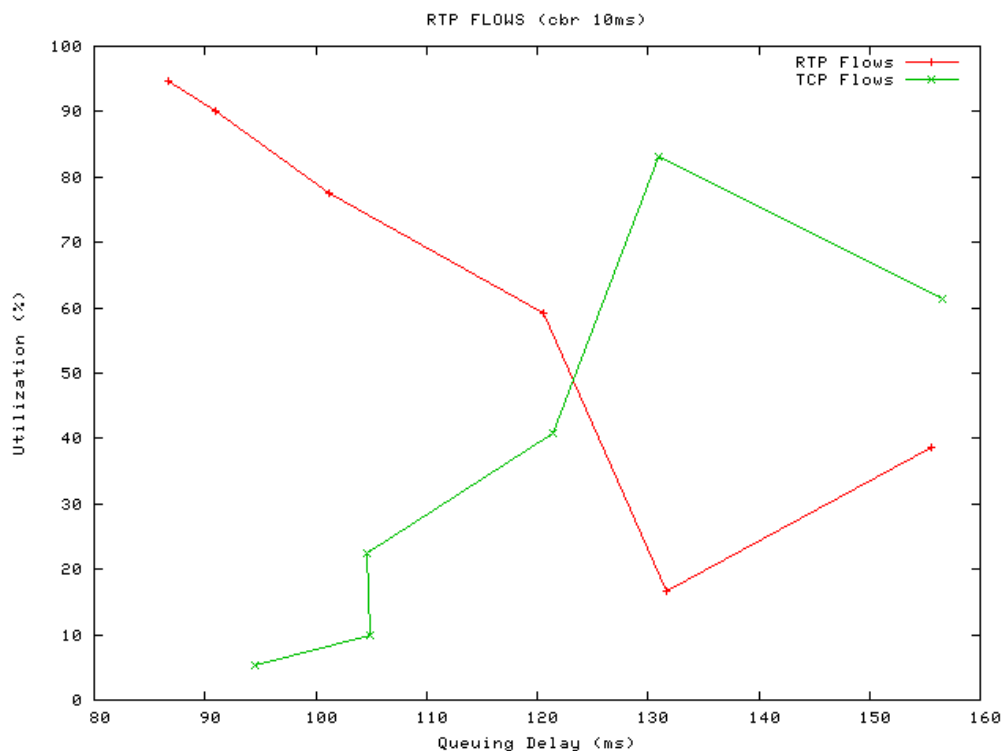
Πίνακας 3.36 : Καθυστέρηση πακέτων στους δρομολογητές.



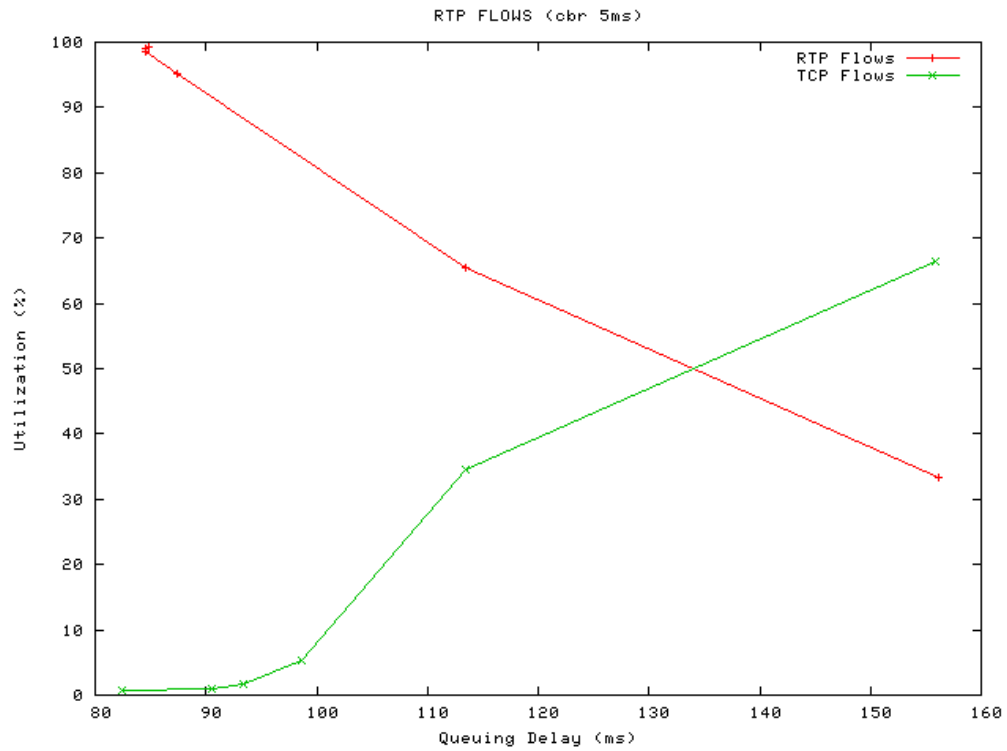
Σχήμα 3.43 : Καθυστέρηση πακέτων στους δρομολογητές.

Παρατηρούμε πως σε αυτή την περίπτωση ισχύουν τα ίδια συμπεράσματα που έχουμε εξάξει και προηγουμένως ότι δηλαδή τα δύο πρωτόκολλα έχουν παρόμοια καθυστέρηση όσο και αν αυξήσουμε τις ροές.

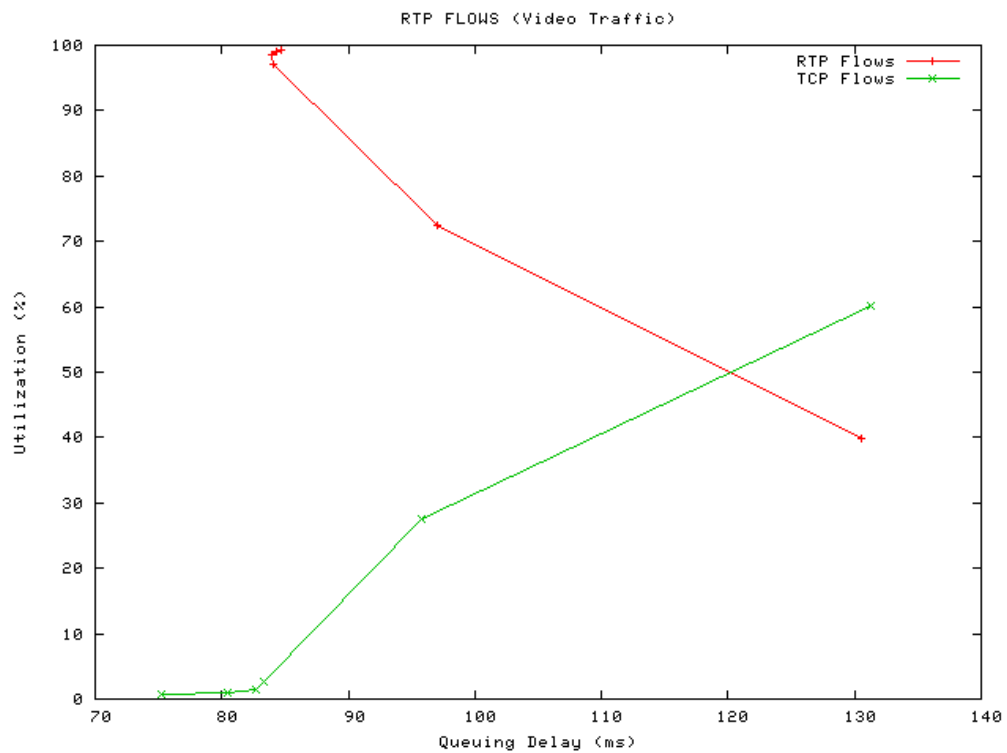
Πιο κάτω παρουσιάζουμε μια σύνοψη των πιο πάνω αποτελεσμάτων των μετρήσεων που έγιναν στην σύνδεση όπου παρατηρείται συμφόρηση. Στα σχήματα 3.44 μέχρι 3.46 παρουσιάζεται το ποσοστό χρήσης της σύνδεσης όπου παρατηρείται συμφόρηση σε συνάρτηση με την καθυστέρηση των πακέτων και των δύο ροών. Το σχήμα 3.44 είναι με χρήση πακέτων με χρόνο αποστολής μεταξύ τους 10ms, το σχήμα 3.45 με χρόνο 5ms και το σχήμα 3.46 με προσομοίωση βίντεο. Σημειώνουμε πως το άθροισμα των δύο ροών ως προς τον άξονα των Ψ δίνει 100% μιας και η χρήση της σύνδεσης είναι ολοκληρωτική.



Σχήμα 3.44 : Χρήση σύνδεσης συμφόρησης σε συνάρτηση του χρόνου καθυστέρησης των πακέτων για πακέτα με χρόνο αποστολής μεταξύ τους 10 ms.



Σχήμα 3.45 : Χρήση σύνδεσης συμφόρησης σε συνάρτηση του χρόνου καθυστέρησης των πακέτων για πακέτα με χρόνο αποστολής μεταξύ τους 5 ms.



Σχήμα 3.46 : Χρήση σύνδεσης συμφόρησης σε συνάρτηση του χρόνου καθυστέρησης των πακέτων για πακέτα με προσομοίωση βίντεο.

Το σημείο όπου διασταυρώνονται οι δύο γραφικές στο σχήμα 3.44 δηλώνει το χρόνο καθυστέρησης πακέτων των δύο ροών όπου το ποσοστό του εύρους ζώνης που καταλαμβάνουν στην σύνδεση είναι ίσο. Ο χρόνος αυτός για το συγκεκριμένο σχήμα είναι γύρω στα 123ms. Για το σχήμα 3.45 ο χρόνος αυτός είναι 132ms ενώ για το σχήμα 3.46 ο χρόνος είναι 119ms.

Για επιβεβαίωση των υπολογισμών μας τρέξαμε όλες τις προσομοιώσεις μας στην τοπολογία που περιγράψαμε στο σχήμα 3.17. Τα αποτελέσματα δεν διαφέρουν σε μεγάλο βαθμό από τα αποτελέσματα που παρουσιάζουμε. Επίσης μια δεύτερη μετατροπή η οποία έγινε ήταν να σταματούμε τις μισές ροές από κάθε πρωτόκολλο στο τεσσαρακοστό δευτερόλεπτο και να τις ξεκινήσουμε ξανά στο εβδομηκοστό δευτερόλεπτο. Και πάλι τα αποτελέσματα που πήραμε δεν διαφέρουν σε μεγάλο βαθμό από αυτά που έδινε η βασική τοπολογία.

3.5 Προσομοιώσεις με χρήση του UDP

Χρησιμοποιώντας την βασική τοπολογία που έχουμε περιγράψει στο σχήμα 3.0.1 στέλνουμε τις ροές TCP στο μονοπάτι που αποτελείται από τους κόμβους 0, 2, 3, 4 και τις ροές UDP από το μονοπάτι που αποτελείται από τους κόμβους 1, 2, 3, 5. Οι μετρήσεις γίνονται με χρήση 10, 25, 50, 75, 100 και 150 ροών από το κάθε πρωτόκολλο. Οι προσομοιώσεις διαρκούν ένα σταθερό χρόνο που έχουμε ορίσει στα 100 δευτερόλεπτα.

Για την προσομοίωση των ροών UDP χρησιμοποιούμε κόμβους του προσομοιωτή, στους οποίους έχουμε τοποθετήσει πράκτορες UDP. Λόγω της έλλειψης οποιουδήποτε μηχανισμού ελέγχου συμφόρησης για το πρωτόκολλο αυτό δεν στέλνονται πίσω στον αποστολέα οποιαδήποτε μηνύματα.

Το πρωτόκολλο έχει δημιουργηθεί για την μεταφορά αρχείων όπου μπορούν να ανεχθούν ένα ποσοστό απώλειας πακέτων. Έχουμε τροποποιήσει κάποιες μεταβλητές ώστε να προσαρμοστούν στις προσομοιώσεις που χρησιμοποιήσαμε. Πιο συγκεκριμένα έχουμε καθορίσει το μέγεθος των πακέτων της εφαρμογής να είναι 200 ψηφιολέξεις και μεταβάλαμε τον χρόνο αποστολής μεταξύ των πακέτων.

Οι μετρήσεις μας έγιναν για 10ms και για 5ms. Επίσης έγιναν προσομοιώσεις με χρήση πραγματικής ταινίας. Αν για παράδειγμα κάνουμε χρήση πακέτων με μέγεθος 200 ψηφιολέξεις και χρόνο μεταξύ μετάδοσης πακέτων 10ms τότε ο ρυθμός μετάδοσης που αναμένουμε για κάθε ροή UDP θα είναι : $200 * 8 \text{ bits} / 0,01\text{ms} = 160\text{Kbps}$. Τα αποτελέσματα από τις διάφορες μετρήσεις τα αποθηκεύουμε σε αρχεία τα οποία επεξεργαζόμαστε στην συνέχεια.

Η πρώτη προσομοίωση αφορά το ποσοστό χρήσης της σύνδεσης από τα δύο πρωτόκολλα στα άκρα. Οι μετρήσεις έγιναν για διαφορετικό αριθμό ροών και τα αποτελέσματα καταγράφονται στον πίνακα 3.37 και παρουσιάζονται γραφικά στο σχήμα 3.46. Τα πακέτα που χρησιμοποιεί το UDP είναι τύπου CBR με διάστημα μεταξύ αποστολής τα 10ms. Επίσης ο πίνακας 3.37 παρουσιάζει το μέσο εύρος ζώνης της σύνδεσης για την κάθε ροή από το κάθε πρωτόκολλο.

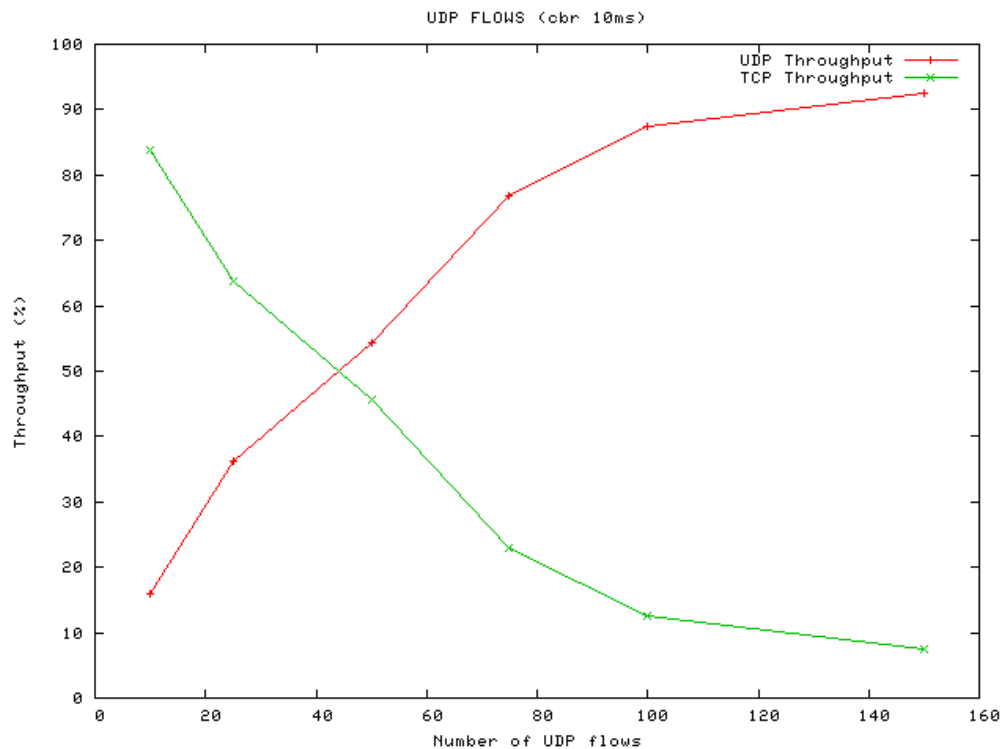
Αριθμός Ροών	TCP Throughput %	TCP Throughput Mb	UDP Throughput %	UDP Throughput Mb
10	83,8908	0,840	15,9746	0,159
25	63,7996	0,228	36,1355	0,159
50	45,712	0,130	54,248	0,159
75	23,0423	0,141	76,9323	0,159
100	12,5448	0,094	87,4268	0,159
150	7,5172	0,215	92,4532	0,159

Πίνακας 3.37 : Ποσοστό χρήσης σύνδεσης στα άκρα και μέσος όρος εύρους ζώνης

Παρατηρούμε πως με την αύξηση του αριθμού των ροών από το κάθε πρωτόκολλο το UDP παίρνει μεγαλύτερο ποσοστό από την σύνδεση σε σχέση με το TCP. Αρχικά όλες οι ροές του UDP περιορίζουν τον ρυθμό αποστολής τους σε περίπου 160 Kbps που είναι ακριβώς το εύρος ζώνης που χρειάζεται η εφαρμογή αλλά δεν διεκδικούν όλο το διαθέσιμο εύρος ζώνης όπως κάνουν οι ροές TCP.

Αντίθετα το TCP τείνει να εκμεταλλευτεί όσο το δυνατό περισσότερο εύρος ζώνης γι' αυτό και οι ροές έχουν μεγαλύτερο ρυθμό. Με την αύξηση όμως των ροών και κατ' επέκταση την αύξηση της συμφόρησης, οι ροές UDP διατηρούν τα επιθυμητά 160Kb

αλλά εκμεταλλεύονται το διαθέσιμο εύρος ζώνης εις βάρος των ροών TCP τις οποίες οδηγούν σε ελαχιστοποίηση του εύρους ζώνης.

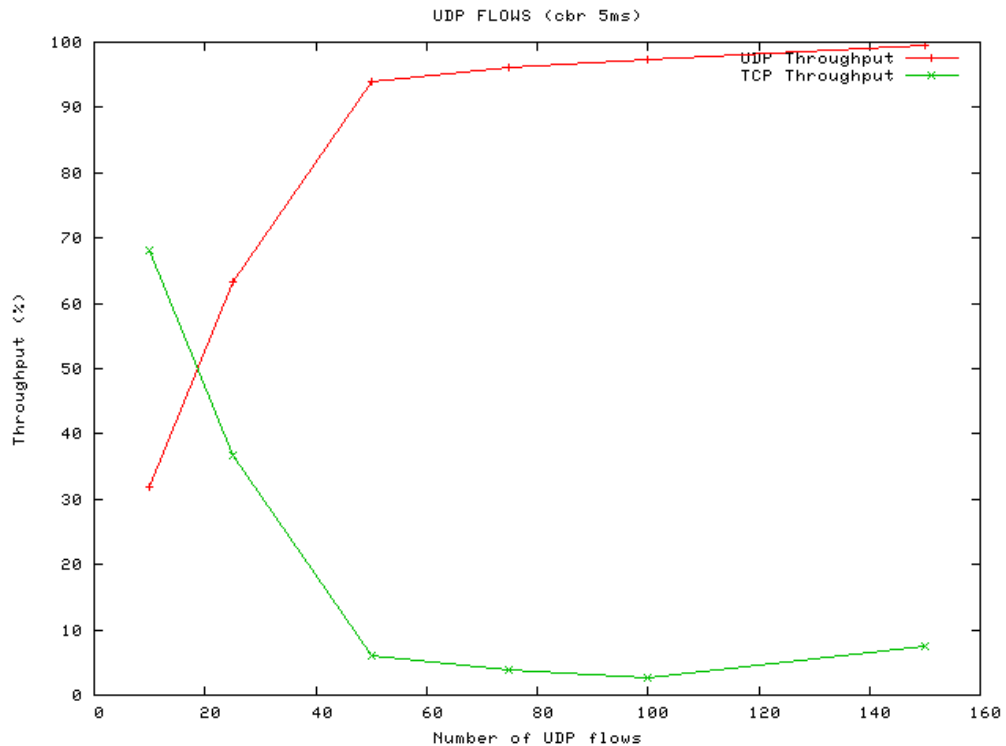


Σχήμα 3.46 : Ποσοστό χρήσης σύνδεσης συναρτήσει του αριθμού ροών TCP και UDP

Επαναλαμβάνουμε την ίδια διαδικασία αλλά αυτή την φορά μειώνουμε τον χρόνο μεταξύ της κάθε αποστολής πακέτων της εφαρμογής στα 5ms. Οι μετρήσεις έγιναν για διαφορετικό αριθμό ροών και τα αποτελέσματα καταγράφονται στον πίνακα 3.38 και παρουσιάζονται γραφικά στο σχήμα 3.47. Επίσης ο πίνακας 3.38 παρουσιάζει το μέσο εύρος ζώνης της σύνδεσης για την κάθε ροή από το κάθε πρωτόκολλο.

Αριθμός Ροών	TCP Throughput %	TCP Throughput Mb	UDP Throughput %	UDP Throughput Mb
10	68,0428	0,6818	31,8522	0,3193
25	36,6072	0,1729	63,354	0,3194
50	6,008	0,0757	93,9676	0,3195
75	3,8017	0,1806	96,1724	0,3196
100	2,7466	0,1062	97,226	0,3196
150	7,3778	0,1097	99,4996	0,3195

Πίνακας 3.38 : Ποσοστό χρήσης σύνδεσης στα άκρα και μέσος όρος εύρους ζώνης



Σχήμα 3.47 : Ποσοστό χρήσης σύνδεσης συναρτήσει του αριθμού ροών TCP και UDP

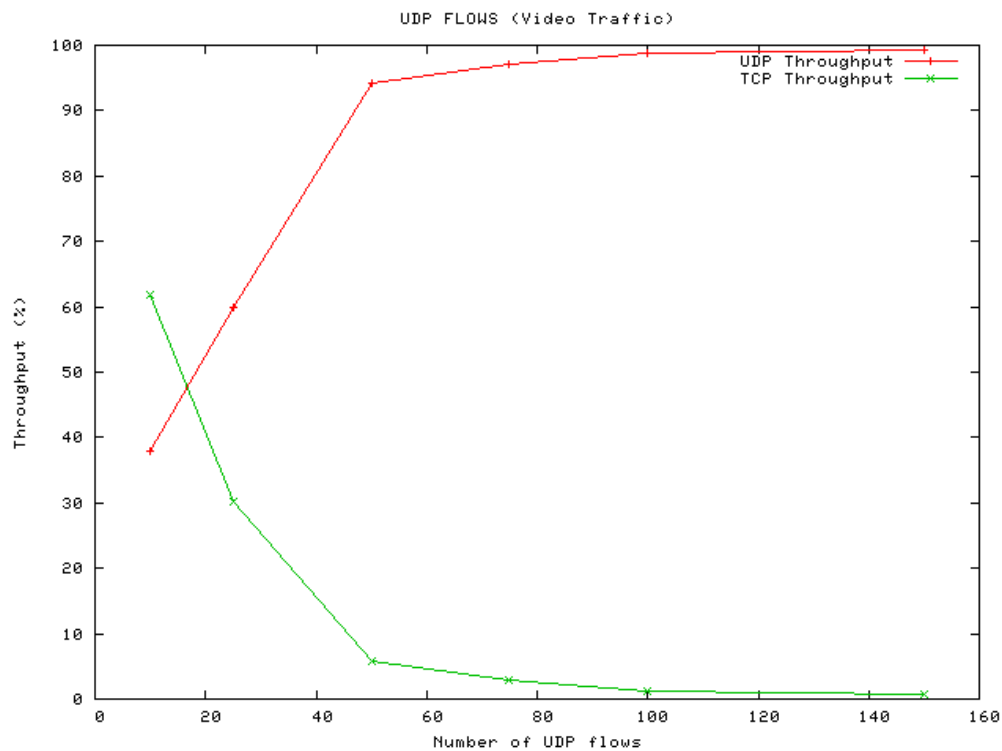
Παρατηρούμε όπως και στην προηγούμενη περίπτωση ότι οι ροές UDP αρχικά καταλαμβάνουν το υπολογιζόμενο εύρος ζώνης που δικαιούνται, και με την αύξηση των ροών και της συμφόρησης εκμεταλλεύονται το εύρος ζώνης εις βάρος του TCP χωρίς οι ροές αυτές να παίρνουν λιγότερο από το υπολογιζόμενο εύρος ζώνης.

Τέλος δοκιμάζουμε την ίδια διαδικασία αλλά αντί για χρήση πακέτων εφαρμογής προσομοιώνουμε με την χρήση πραγματικού βίντεο. Το βίντεο που αρχικά είναι χωρισμένο σε διάφορα πλαίσια ταινίας μετατρέπεται σε πακέτα, με χρήση ενός μικρού προγράμματος το οποίο μετατρέπει τα πλαίσια αυτά σε πακέτα αποστολής σταθερού ρυθμού.

Όπως αναφέρει ο προγραμματιστής του προγράμματος αυτού η προσομοίωση θα είναι παρόμοια εάν είχαμε χρήση πακέτων με μέγεθος 200 ψηφιολέξεις που θα είχαν διαφορά χρόνου στην αποστολή τους 10ms. Τα αποτελέσματα της διαδικασίας παρουσιάζονται στον πίνακα 3.39 και η γραφική απεικόνιση στο σχήμα 3.48.

Αριθμός Ροών	TCP Throughput %	TCP Throughput Mb	UDP Throughput %	UDP Throughput Mb
10	61,8924	0,6348	38,0272	0,5075
25	30,0861	0,1407	69,8852	0,2973
50	5,8635	0,0094	94,1092	0,3056
75	2,8054	0,0123	97,166	0,1218
100	1,1590	0,0009	98,8204	0,1083
150	0,7142	0,0008	99,278	0,0474

Πίνακας 3.39 : Ποσοστό χρήσης σύνδεσης στα άκρα και μέσος όρος εύρους ζώνης



Σχήμα 3.48 : Ποσοστό χρήσης σύνδεσης συναρτήσει του αριθμού ροών TCP και UDP

Όπως και στις προηγούμενες περιπτώσεις έτσι και εδώ παρατηρούμε πως η προσομοίωση με την χρήση πραγματικού βίντεο, δίνει περίπου τα ίδια αποτελέσματα με αυτά που παίρνουμε και στις άλλες περιπτώσεις. Συμπεραίνουμε τελικά πως στην ύπαρξη συμφόρησης οι ροές UDP εκμεταλλεύονται το εύρος ζώνης της σύνδεσης σε βάρος του TCP.

Για την δεύτερη μας μέτρηση θα εξετάσουμε την καθυστέρηση που παρουσιάζουν τα πακέτα σε συνάρτηση με τον αριθμό των ροών του UDP. Από τις προηγούμενες μετρήσεις παρατηρήσαμε πως οι ροές UDP κερδίζουν σε εύρος ζώνης εις βάρος του

TCP. Ένα λογικό σενάριο θα ήταν οι ροές UDP να παρουσιάζουν λιγότερη καθυστέρηση στα άκρα ώστε να κερδίζουν σε εύρος ζώνης.

Πιο κάτω παρουσιάζουμε τα αποτελέσματα των μετρήσεων της καθυστέρησης σε συνάρτηση με τον αριθμό των ροών για τα δύο πρωτόκολλα. Αρχικά παρουσιάζουμε τα αποτελέσματα με την χρήση πακέτων εφαρμογής με μέγεθος 200 ψηφιολέξεις και χρόνο μεταξύ αποστολής τα 10ms.

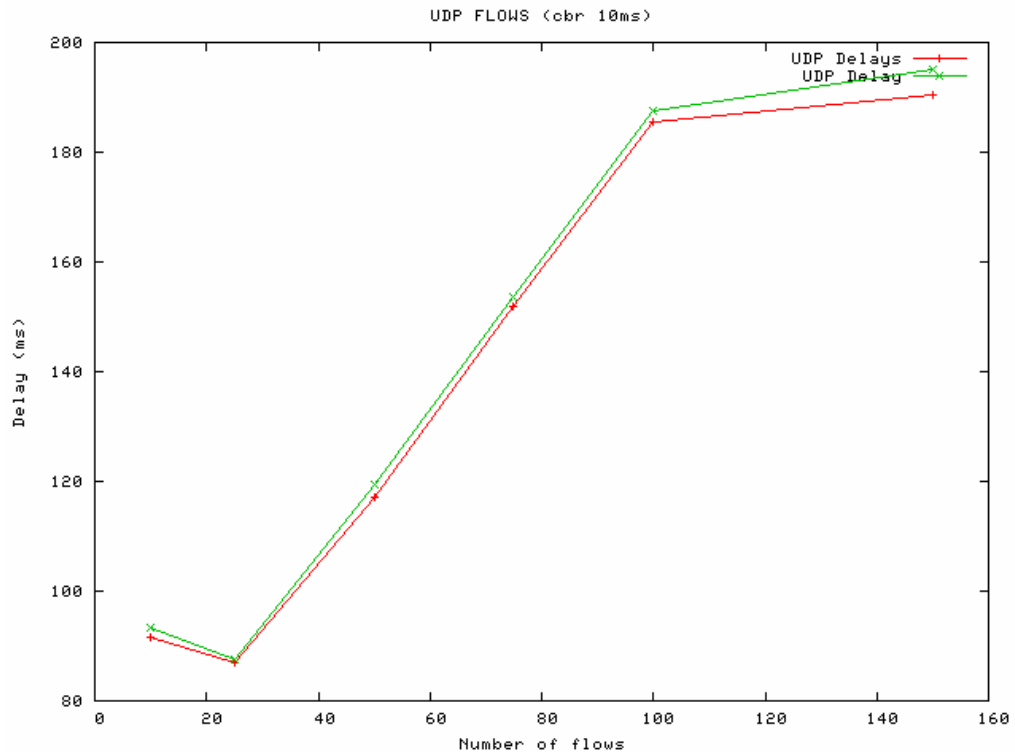
Τα αποτελέσματα παρουσιάζονται στον πίνακα 3.40 και γραφικά στο σχήμα 3.49. Επίσης ο πίνακας περιέχει το ποσοστό των πακέτων που χάνονται από την έξοδο της σύνδεσης που παρουσιάζεται συμφόρηση μέχρι τον παραλήπτη, καθώς και η διαφορά χρόνου στον οποίο κάποια πακέτα φτάνουν στον παραλήπτη μετά που ακολούθησαν διαφορετική διαδρομή (Jitter).

Αριθμός Ροών	UDP Drop Rate %	Mean Delay UDP	Mean Delay TCP	Jitter UDP
10	0,16	91,574	90,554	6,672
25	16,77	86,908	85,128	14,16
50	32,19	117,198	116,185	12,7031
75	53,31	151,848	150,284	28,302
100	67,83	160,472	159,185	24,236
150	69,07	185,630	184,122	15,378

Πίνακας 3.40 : Μέση καθυστέρηση πακέτων στα άκρα για το UDP.

Παρατηρούμε πως με την αύξηση των ροών η καθυστέρηση που παρουσιάζουν οι ροές UDP αυξάνεται γραμμικά μέχρι ενός σημείου και από εκεί και πέρα η καθυστέρηση παραμένει σταθερή. Αυτός είναι και ο λόγος που κερδίζουν περισσότερο εύρος ζώνης σε βάρος του TCP. Οι ροές του TCP με την σειρά τους αυξάνουν την καθυστέρηση όσο αυξάνονται οι ροές, επομένως και χάνουν εύρος ζώνης.

Επαναλαμβάνουμε την ίδια διαδικασία αλλά αυτή την φορά κάνουμε χρήση πακέτων μεγέθους 200 ψηφιολέξεων με χρόνο μεταξύ αποστολής τα 5ms. Τα αποτελέσματα παρουσιάζονται στον πίνακα 3.41 και γραφικά στο σχήμα 3.50.



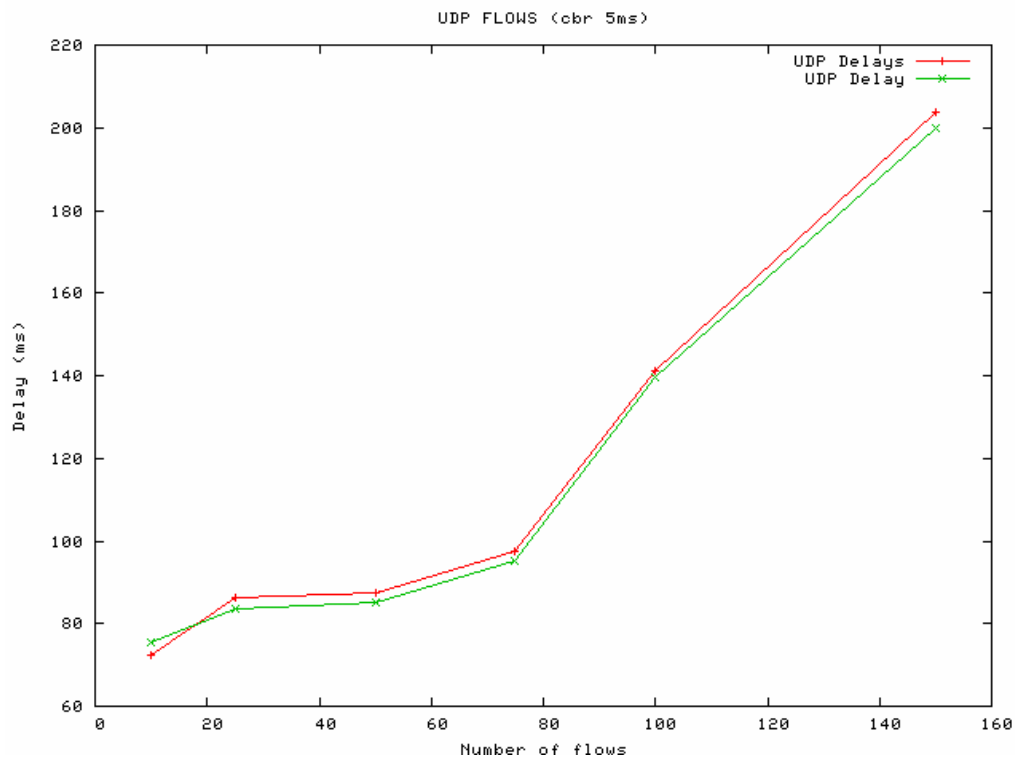
Σχήμα 3.49 : Καθυστέρηση πακέτων σε συνάρτηση του αριθμού ροών UDP.

Επίσης ο πίνακας περιέχει το ποσοστό των πακέτων που χάνονται από την έξοδο της σύνδεσης που παρουσιάζεται συμφόρηση μέχρι τον παραλήπτη, καθώς και η διαφορά χρόνου στον οποίο κάποια πακέτα φτάνουν στον παραλήπτη μετά που ακολούθησαν διαφορετική διαδρομή (Jitter).

Αριθμός Ροών	UDP Drop Rate %	Mean Delay UDP	Mean Delay TCP	Jitter UDP
10	0,2	72,449	71,218	9,104
25	39,273	86,384	84,158	7,421
50	69,15	87,411	85,125	35,196
75	68,976	97,604	97,002	15,274
100	69,06	141,058	139,254	14,440
150	99,97	203,882	202,148	17,426

Πίνακας 3.41 : Μέση καθυστέρηση πακέτων στα άκρα για το UDP.

Παρατηρούμε σε σύγκριση με τα προηγούμενα αποτελέσματα πως η καθυστέρηση κυμαίνεται στα ίδια περίπου επίπεδα και πως η αλλαγή στο χρόνο αποστολής μεταξύ των πακέτων δεν επιφέρει οποιαδήποτε ουσιαστική αλλαγή στα αποτελέσματα.

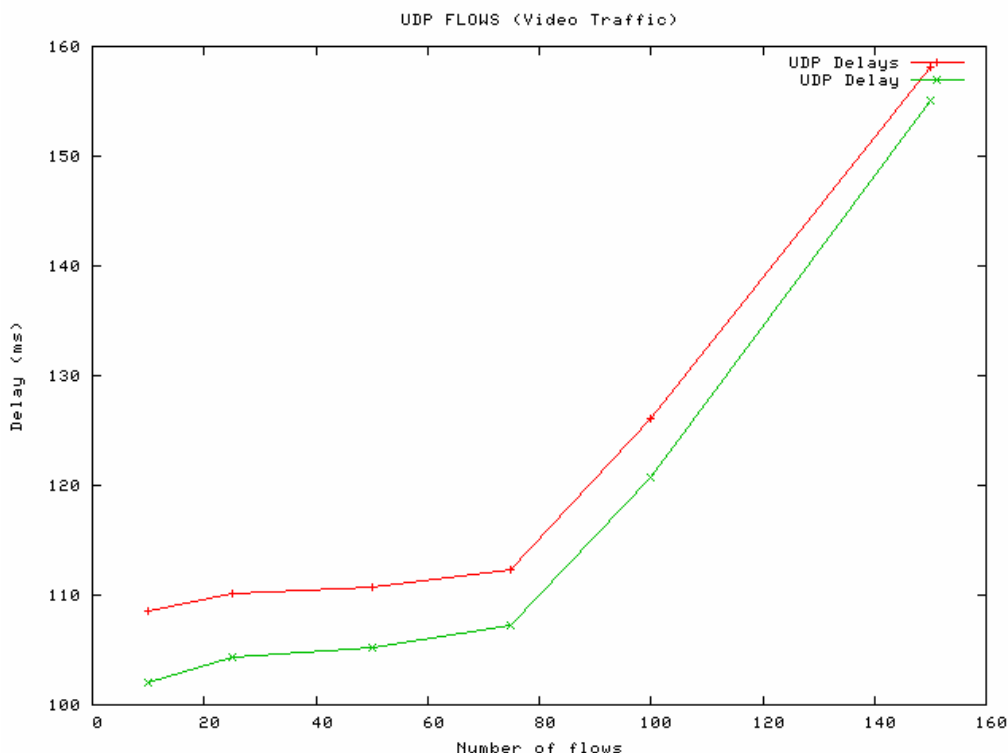


Σχήμα 3.50 : Καθυστέρηση πακέτων σε συνάρτηση του αριθμού ροών UDP.

Τέλος προσομοιώνουμε με την χρήση αποστολής κανονικής ταινίας όπως ακριβώς και το προηγούμενο πείραμα. Τα αποτελέσματα παρουσιάζονται στον πίνακα 3.42 και γραφικά στο σχήμα 3.50. Επίσης ο πίνακας περιέχει το ποσοστό των πακέτων που χάνονται από την έξοδο της σύνδεσης που παρουσιάζεται συμφόρηση μέχρι τον παραλήπτη, καθώς και η διαφορά χρόνου στον οποίο κάποια πακέτα φτάνουν στον παραλήπτη μετά που ακολούθησαν διαφορετική διαδρομή (Jitter).

Αριθμός Ροών	UDP Drop Rate %	Mean Delay UDP	Mean Delay TCP	Jitter UDP
10	2,086	110,098	108,458	14,866
25	32,86	110,762	108,675	9,080
50	56,43	111,576	108,996	4,142
75	69,843	112,279	110,258	3,626
100	74,266	126,027	125,486	2,807
150	83,72	158,158	157,158	3,491

Πίνακας 3.42 : Μέση καθυστέρηση πακέτων στα άκρα για το UDP.



Σχήμα 3.50 : Καθυστέρηση πακέτων σε συνάρτηση του αριθμού ροών UDP.

Συμπεραίνουμε πως η χρήση του βίντεο δεν προκαλεί μεγάλες αλλαγές στην καθυστέρηση που παρατηρείται στα πακέτα UDP. Ο μέσος χρόνος καθυστέρησης σταθεροποιείται στα 110ms σε σύγκριση με 80ms που είναι στις προηγούμενες περιπτώσεις. Συγκρίνοντας τα αποτελέσματα των μέχρι στιγμής πειραμάτων βλέπουμε πως επειδή η καθυστέρηση των πακέτων UDP είναι μικρότερη από αυτή του TCP, οι ροές του UDP παίρνουν μεγαλύτερο εύρος ζώνης.

Για τα επόμενα πειράματα χρησιμοποιούμε μετρικές τις οποίες λαμβάνουμε στην γραμμή από την οποία υπάρχει συμφόρηση. Το πρώτο τέτοιο πείραμα αφορούσε το ποσοστό της χρήσης της γραμμής στην οποία υπάρχει συμφόρηση σε συνάρτηση με τον αριθμό των ροών του κάθε πρωτοκόλλου.

Λόγω του ότι το εύρος ζώνης της σύνδεσης είναι 10 φορές μικρότερο από αυτό που ενώνει την έξοδο της σύνδεσης με τον παραλήπτη, δεν παρατηρείται χάσιμο πακέτων στην γραμμή αυτή. Επομένως τα αποτελέσματα αυτής της μέτρησης ταυτίζονται με τα αποτελέσματα του πειράματος που αφορά την χρήση της σύνδεσης στον παραλήπτη σε

συνάρτηση με τον αριθμό των ροών. Σχηματικά τα αποτελέσματα είναι τα ίδια με αυτά των σχημάτων 3.46 μέχρι 3.48.

Το επόμενο πείραμα που αφορά την σύνδεση συμφόρησης αφορά την απώλεια των πακέτων των διαφόρων ροών σε συνάρτηση με τον αριθμό των ροών. Θα περιμέναμε εδώ να δούμε πως οι ροές που έχουν το μεγαλύτερο εύρος ζώνης να έχουν και το μεγαλύτερο ποσοστό χαμένων πακέτων.

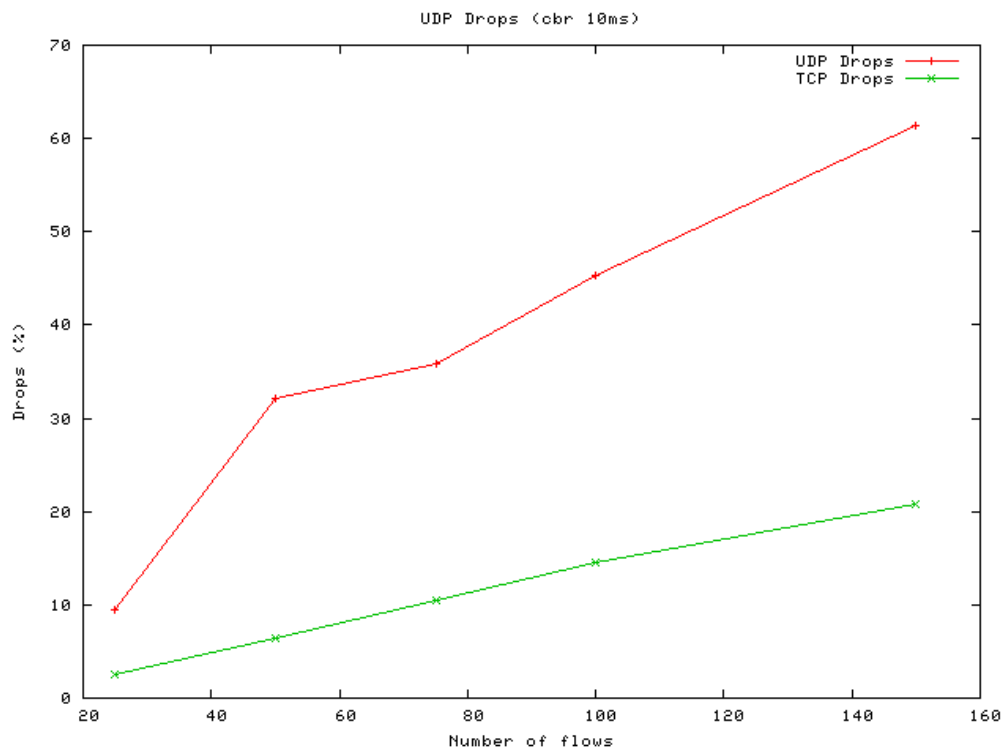
Στο πρώτο πείραμα χρησιμοποιούμε πακέτα εφαρμογής μεγέθους 200 ψηφιολέξεων και χρόνο μεταξύ αποστολής των πακέτων 10ms. Τα αποτελέσματα παρουσιάζονται στον πίνακα 3.43 και γραφικά στο σχήμα 3.51.

Αριθμός Ροών	TCP Drop Rate %	UDP Drop Rate %
10	-	-
25	2,57258	9,53954
50	6,37424	32,087
75	10,5442	35,8097
100	14,6236	45,2956
150	20,7756	61,4359

Πίνακας 3.43 : Ποσοστό χαμένων πακέτων στη σύνδεση συμφόρησης.

Παρατηρούμε πως οι ροές UDP έχουν μεγαλύτερο ποσοστό χαμένων πακέτων κατά την διάρκεια των προσομοιώσεων. Οι γραφικές των πρωτοκόλλων είναι σχεδόν παράλληλες με τη διαφορά κλίμακας να είναι πολύ μεγαλύτερη. Περίπου τα 2/3 των πακέτων UDP χάνονται σε αντίθεση με το 1/3 του TCP. Αυτός είναι και ο λόγος που οι ροές UDP διατηρούν το αρχικό τους εύρος ζώνης.

Επαναλαμβάνουμε το ίδιο πείραμα όπως και στην προηγούμενη περίπτωση αλλά αυτή την φορά χρησιμοποιώντας πακέτα μεγέθους 200 ψηφιολέξεων που έχουν χρόνο μεταξύ αποστολής τα 5ms. Τα αποτελέσματα παρουσιάζονται στον πίνακα 3.44 και γραφικά στο σχήμα 3.52.

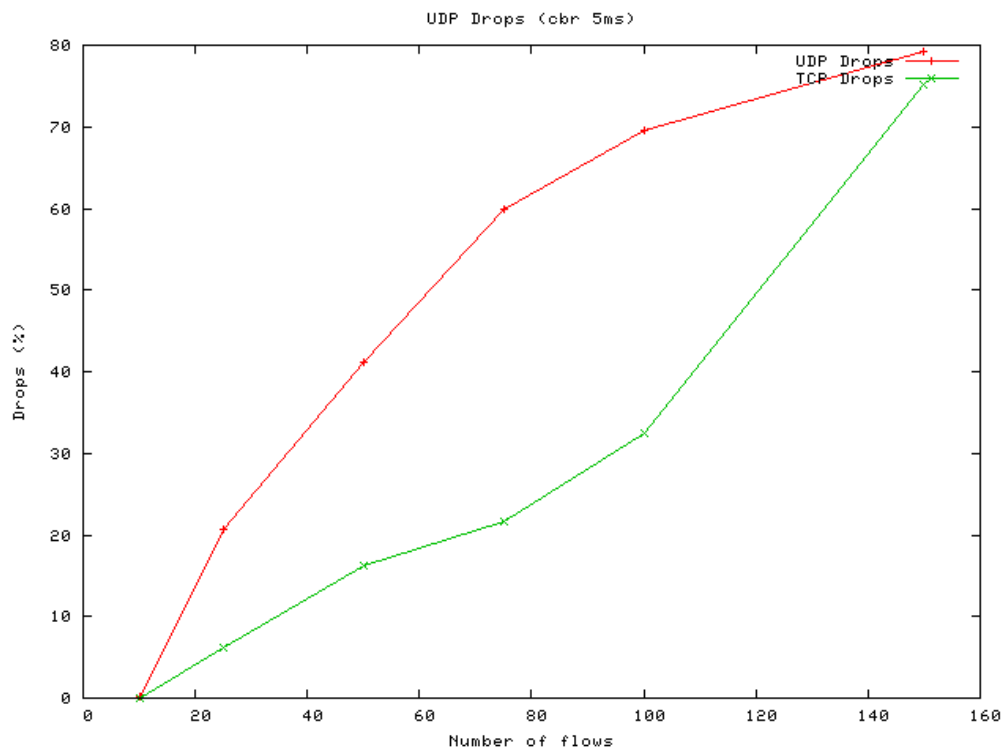


Σχήμα 3.51 : Ποσοστό χαμένων πακέτων στη σύνδεση συμφόρησης συναρτήσει ροών.

Αριθμός Ροών	TCP Drop Rate %	UDP Drop Rate %
10	-	0,26209
25	6,1416	20,6989
50	16,2984	41,2051
75	21,553	59,8853
100	32,4246	69,5839
150	75,0851	79,2486

Πίνακας 3.44 : Ποσοστό χαμένων πακέτων στη σύνδεση συμφόρησης.

Παρατηρούμε πως η μεταβολή στον χρόνο αποστολής των πακέτων επηρεάζει την συμπεριφορά των ροών. Και πάλι παρατηρείται το UDP να έχει ένα μεγαλύτερο ποσοστό χαμένων πακέτων. Όμως με την αύξηση των ροών τα δύο πρωτόκολλα παρουσιάζουν σχεδόν το ίδιο ποσοστό χαμένων πακέτων.



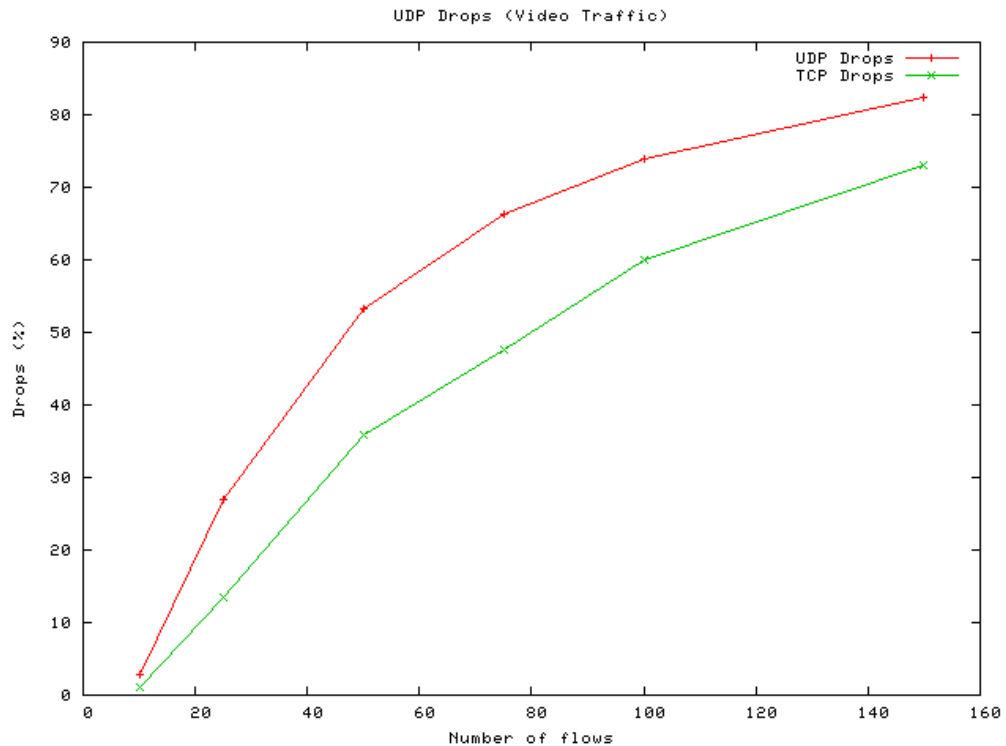
Σχήμα 3.52 : Ποσοστό χαμένων πακέτων στη σύνδεση συμφόρησης συναρτήσει ροών.

Τέλος όπως και στις προηγούμενες περιπτώσεις εφαρμόζουμε προσομοίωση με την χρήση πραγματικού βίντεο. Τα αποτελέσματα παρουσιάζονται στον πίνακα 3.45 και γραφικά στο σχήμα 3.53.

Αριθμός Ροών	TCP Drop Rate %	UDP Drop Rate %
10	1,0215	2,7895
25	13,4659	26,9668
50	35,8379	53,2291
75	47,7053	66,2026
100	60,0052	73,9288
150	73,1506	82,4048

Πίνακας 3.45 : Ποσοστό χαμένων πακέτων στη σύνδεση συμφόρησης.

Τα αποτελέσματα σε αυτή την περίπτωση δεν διαφέρουν από την προηγούμενη περίπτωση. Παρατηρείται το UDP να έχει ένα μεγαλύτερο ποσοστό χαμένων πακέτων αλλά και τα δύο πρωτόκολλα έχουν παρόμοια συμπεριφορά με την αύξηση των ροών.

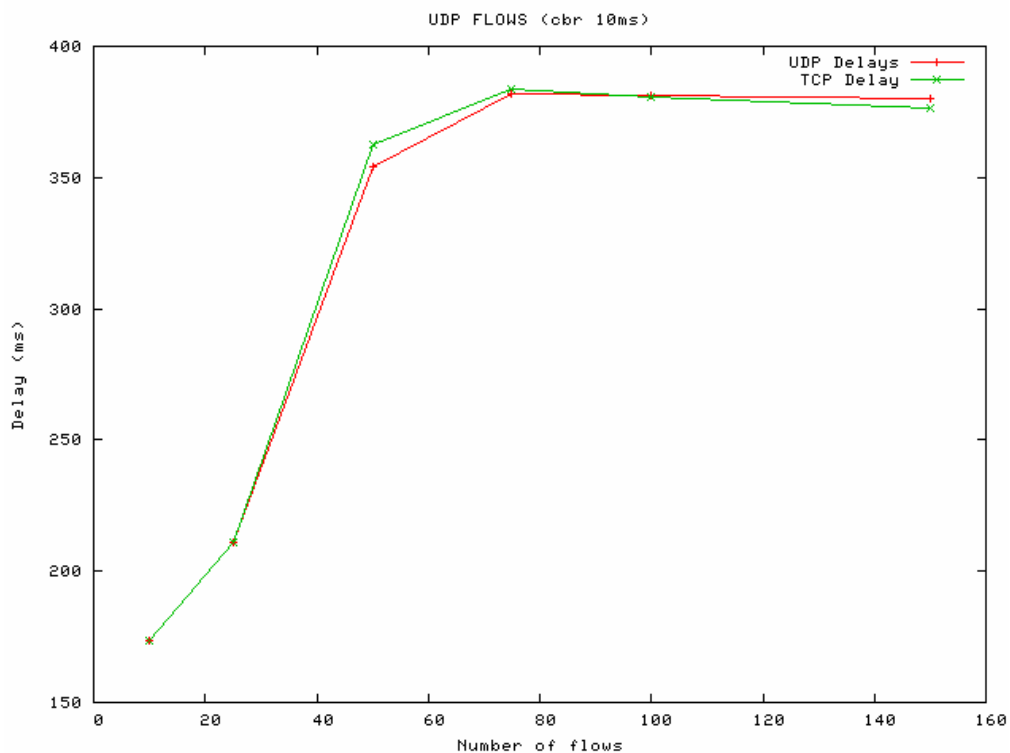


Σχήμα 3.53 : Ποσοστό χαμένων πακέτων στη σύνδεση συμφόρησης συναρτήσεως ροών.

Το επόμενο πείραμα έχει σκοπό την εύρεση της καθυστέρησης των πακέτων και των δύο ροών που παρατηρείται στους δρομολογητές στη γραμμή όπου υπάρχει η συμφόρηση. Σκοπός του πειράματος αυτού είναι ο έλεγχος εάν τα πακέτα των δύο ροών τυγχάνουν της ίδιας προτεραιότητας στους δρομολογητές. Αρχικά ελέγχουμε τις μετρήσεις μας με χρήση πακέτων μεγέθους 200 ψηφιολέξεων που ο χρόνος αποστολής μεταξύ των πακέτων είναι 10ms. Τα αποτελέσματα παρουσιάζονται στον πίνακα 3.46 και γραφικά στο σχήμα 3.54.

Αριθμός Ροών	Mean Delay UDP	Jitter UDP	Mean Delay TCP	Jitter TCP
10	130,141	6,70503	129,122	5,6807
25	256,225	14,2792	257,387	15,6479
50	324,094	12,7134	324,319	15,2364
75	367,5945	14,6128	372,294	23,1855
100	488,5632	14,0255	396,4934	25,7751
150	483,2587	10,7044	387,0607	20,9369

Πίνακας 3.46 : Καθυστέρηση πακέτων στους δρομολογητές.

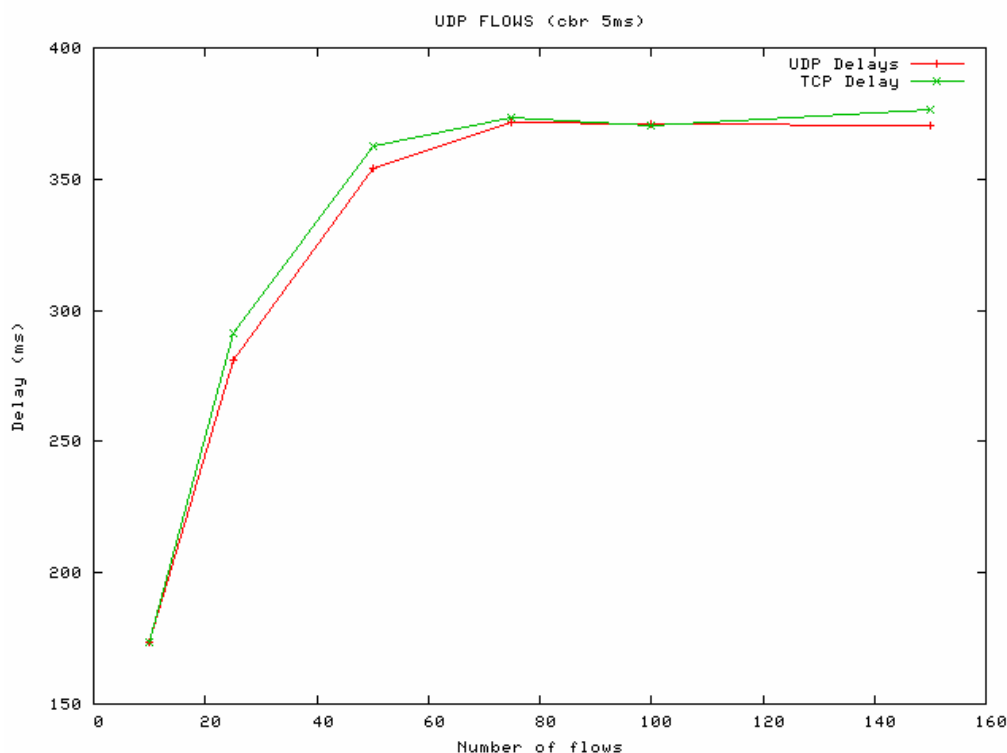


Σχήμα 3.54 : Καθυστέρηση πακέτων στους δρομολογητές.

Όπως παρατηρούμε σε όλα τα στάδια της συμφόρησης οι ροές και των δύο πρωτοκόλλων λαμβάνουν περίπου τον ίδιο χρόνο καθυστέρησης. Επαναλαμβάνουμε το ίδιο πείραμα αυτή την φορά κάνοντας χρήση πακέτων μεγέθους 200 ψηφιολέξεων που διαφέρουν στον χρόνο αποστολής τους κατά 5 ms. Τα αποτελέσματα παρουσιάζονται στον πίνακα 3.47 και γραφικά στο σχήμα 3.55.

Αριθμός Ροών	Mean Delay UDP	Jitter UDP	Mean Delay TCP	Jitter TCP
10	173,535	9,12566	173,334	8,79222
25	211,081	7,96066	211,238	9,79273
50	383,9222	9,37283	392,4868	21,6509
75	381,7052	6,43534	383,641	14,5003
100	381,1346	5,70171	380,6199	15,8103
150	380,3157	6,12817	376,4219	38,2866

Πίνακας 3.47 : Καθυστέρηση πακέτων στους δρομολογητές.

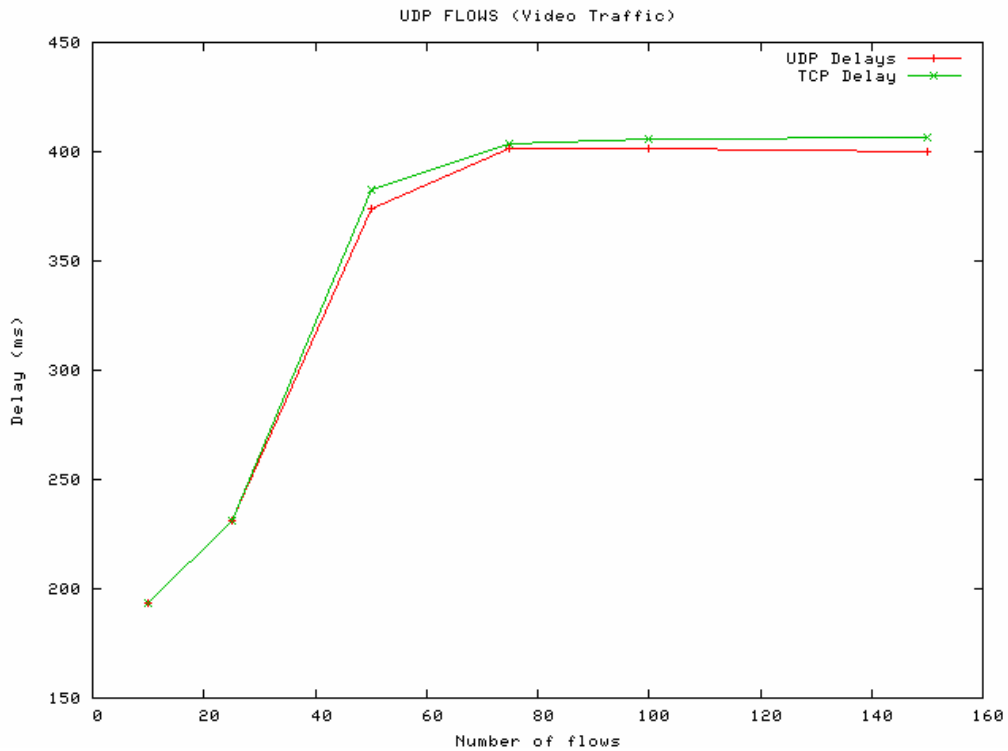


Σχήμα 3.55 : Καθυστέρηση πακέτων στους δρομολογητές.

Τα αποτελέσματα που παράγονται σε αυτή την περίπτωση δεν διαφέρουν κατά πολύ από την περίπτωση όπου ο χρόνος αποστολής μεταξύ των πακέτων ήταν σε πιο ψηλά επίπεδα. Και πάλι οι δύο ροές έχουν σχεδόν την ίδια καθυστέρηση όσο αυξάνεται ο αριθμός των ροών. Δοκιμάζουμε για ακόμα μια φορά την προσομοίωση χρησιμοποιώντας πραγματική ταινία όπως και προηγουμένως. Τα αποτελέσματα παρουσιάζονται στον πίνακα 3.48 και γραφικά στο σχήμα 3.56.

Αριθμός Ροών	Mean Delay UDP	Jitter UDP	Mean Delay TCP	Jitter TCP
10	129,896	14,9929	130,028	15,1438
25	295,0406	9,4268	294,9836	9,49374
50	381,9716	4,79223	381,9622	4,9369
75	380,6833	5,07181	380,6744	5,34351
100	380,5575	5,01402	380,5355	5,441
150	380,3602	4,82714	380,2706	5,66759

Πίνακας 3.48 : Καθυστέρηση πακέτων στους δρομολογητές.

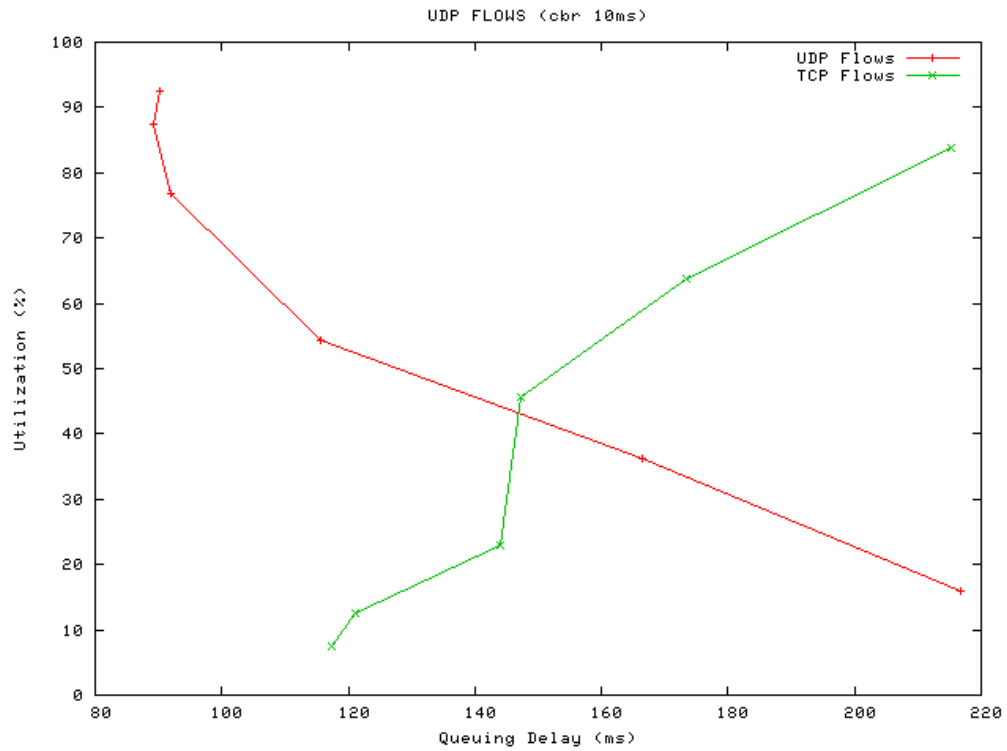


Σχήμα 3.56 : Καθυστέρηση πακέτων στους δρομολογητές.

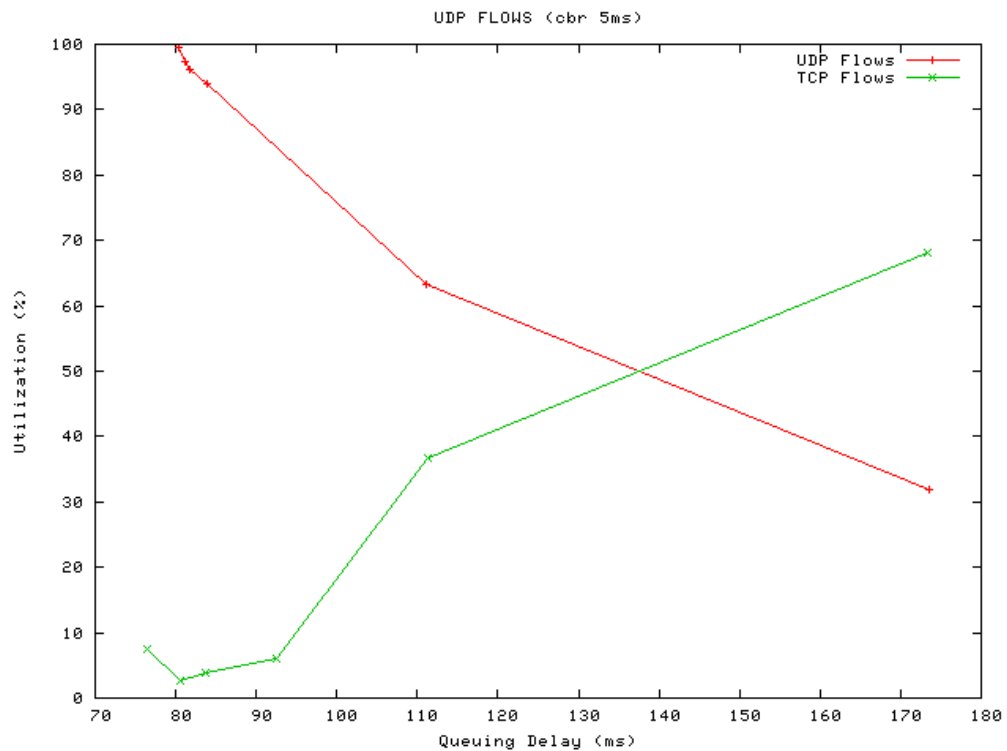
Παρατηρούμε πως σε αυτή την περίπτωση ισχύουν τα ίδια συμπεράσματα που έχουμε εξάξει και προηγουμένως ότι δηλαδή τα δύο πρωτόκολλα έχουν παρόμοια καθυστέρηση όσο και αν αυξήσουμε τις ροές.

Πιο κάτω παρουσιάζουμε μια σύνοψη των πιο πάνω αποτελεσμάτων των μετρήσεων που έγιναν στην σύνδεση όπου παρατηρείται συμφόρηση. Στα σχήματα 3.57 μέχρι 3.59 παρουσιάζεται το ποσοστό χρήσης της σύνδεσης όπου παρατηρείται συμφόρηση σε συνάρτηση με την καθυστέρηση των πακέτων και των δύο ροών.

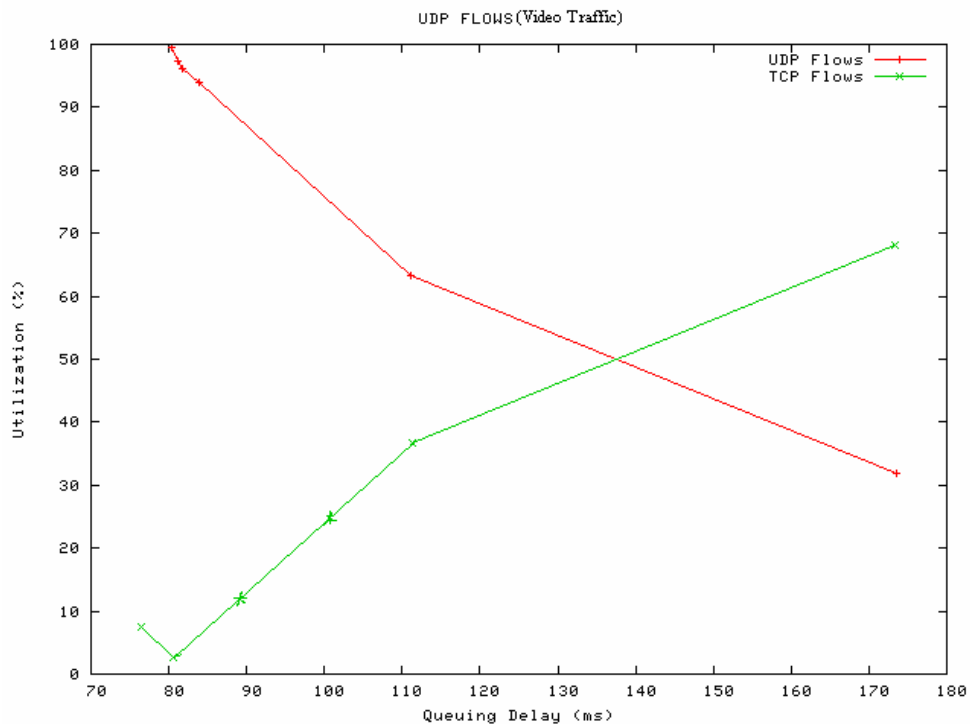
Το σχήμα 3.57 είναι με χρήση πακέτων με χρόνο αποστολής μεταξύ τους 10ms, το σχήμα 3.58 με χρόνο 5ms και το σχήμα 3.59 με προσομοίωση βίντεο. Σημειώνουμε πως το άθροισμα των δύο ροών ως προς τον άξονα των Ψ δίνει 100% μιας και η χρήση της σύνδεσης είναι ολοκληρωτική.



Σχήμα 3.57 : Χρήση σύνδεσης συμφόρησης σε συνάρτηση του χρόνου καθυστέρησης των πακέτων για πακέτα με χρόνο αποστολής μεταξύ τους 10 ms.



Σχήμα 3.58 : Χρήση σύνδεσης συμφόρησης σε συνάρτηση του χρόνου καθυστέρησης των πακέτων για πακέτα με χρόνο αποστολής μεταξύ τους 5 ms.



Σχήμα 3.59 : Χρήση σύνδεσης συμφόρησης σε συνάρτηση του χρόνου καθυστέρησης των πακέτων για πακέτα με προσομοίωση βίντεο.

Το σημείο όπου διασταυρώνονται οι δύο γραφικές στο σχήμα 3.57 δηλώνει το χρόνο καθυστέρησης πακέτων των δύο ροών όπου το ποσοστό του εύρους ζώνης που καταλαμβάνουν στην σύνδεση είναι ίσο. Ο χρόνος αυτός για το συγκεκριμένο σχήμα είναι γύρω στα 150ms. Για το σχήμα 3.58 ο χρόνος αυτός είναι 135ms ενώ για το σχήμα 3.59 ο χρόνος είναι 136ms.

Για επιβεβαίωση των υπολογισμών μας τρέξαμε όλες τις προσομοιώσεις μας στην τοπολογία που περιγράψαμε στο σχήμα 3.17. Τα αποτελέσματα δεν διαφέρουν σε μεγάλο βαθμό από τα αποτελέσματα που παρουσιάζουμε. Επίσης μια δεύτερη μετατροπή η οποία έγινε ήταν να σταματούμε τις μισές ροές από κάθε πρωτόκολλο στο τεσσαρακοστό δευτερόλεπτο και να τις ξεκινήσουμε ξανά στο εβδομηκοστό δευτερόλεπτο. Και πάλι τα αποτελέσματα που πήραμε δεν διαφέρουν σε μεγάλο βαθμό από αυτά που έδινε η βασική τοπολογία.

ΚΕΦΑΛΑΙΟ 4

Ανάλυση Αποτελεσμάτων

4.1 Ανάλυση Αποτελεσμάτων Πρωτοκόλλου UDP	111
4.2 Ανάλυση Αποτελεσμάτων Πρωτοκόλλου RTP	115
4.3 Ανάλυση Αποτελεσμάτων Πρωτοκόλλου TFRC	116
4.4 Ανάλυση Αποτελεσμάτων Πρωτοκόλλου SCTP	117

4.1 Ανάλυση Αποτελεσμάτων Πρωτοκόλλου UDP

Για να κατανοήσουμε καλύτερα τα αποτελέσματα τα οποία έχουμε λάβει από τις προσομοιώσεις θα πρέπει να ελέγξουμε και να κατανοήσουμε τον τρόπο λειτουργίας των μηχανισμών ελέγχου συμφόρησης των διαφόρων πρωτοκόλλων που έχουμε χρησιμοποιήσει. Το Πρωτόκολλο UDP, όπως αναφέραμε και στα προηγούμενα κεφάλαια, είναι το πρωτόκολλο που χρησιμοποιείται κυρίως στο σημερινό διαδίκτυο για την μεταφορά πολυμεσικών αρχείων και το οποίο παρέχει ελάχιστο έως καθόλου μηχανισμό για τον έλεγχο των δεδομένων τα οποία στέλνει.

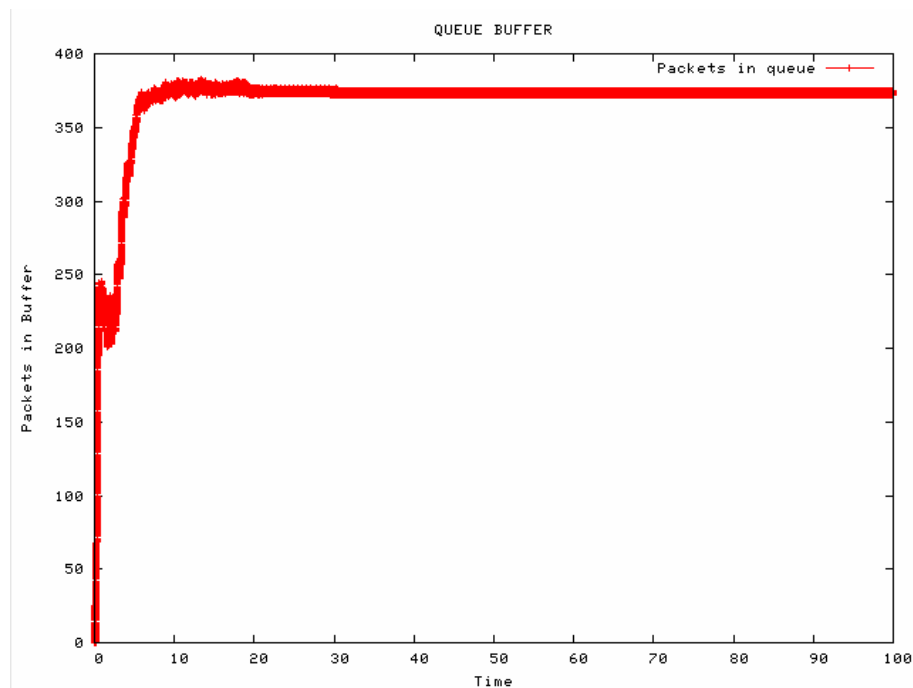
Στην περίπτωση όπου ελέγχουμε το ποσοστό χρήσης της σύνδεσης στην οποία παρατηρείται συμφόρηση (Σχήματα 3.46-3.48) και κατ' επέκταση του ποσοστού χρήσης της σύνδεσης στα άκρα, βλέπουμε ότι το UDP χρησιμοποιεί την σύνδεση σε ένα ποσοστό μεγαλύτερο του 90% και πως οι ροές του, εξασφαλίζουν το επιθυμητό εύρος ζώνης σε βάρος του TCP. Αυτή η παρατήρηση πρέπει να αναλυθεί σε συνδυασμό με το ποσοστό των χαμένων πακέτων στους δρομολογητές που βρίσκονται στην άκρη της σύνδεσης όπου υπάρχει συμφόρηση (Σχήματα 3.51–3.53).

Παρατηρούμε πως τα ποσοστά των χαμένων πακέτων βρίσκονται σε ένα μεγάλο ποσοστό που στην περίπτωση της προσομοίωσης με πραγματικό βίντεο βρίσκονται σε ένα ποσοστό μεγαλύτερο του 80%. Αυτό σημαίνει πως λιγότερο από το 20% των πακέτων είναι πιθανό να φτάσουν στον προορισμό τους, πράγμα που θα προκαλούσε αδυναμία στον παραλήπτη να κατανοήσει το περιεχόμενο της ταινίας. Παρατηρούμε δηλαδή πως η έλλειψη μηχανισμού ελέγχου συμφόρησης οδηγεί μεν τις ροές να παίρνουν το επιθυμητό εύρος ζώνης αλλά τα πακέτα τα οποία στέλνονται ρίχνονται από τους δρομολογητές και τους ενδιάμεσους κόμβους της σύνδεσης. Επομένως το εύρος ζώνης που χρησιμοποιούν είναι σαν να πηγαίνει χαμένο και σε βάρος των άλλων ροών οι οποίες έχουν κάποιο μηχανισμό ελέγχου ροής δεδομένων.

Εάν δούμε και την καθυστέρηση η οποία παρατηρείται στους δρομολογητές και για τους δύο τύπους πακέτων βλέπουμε πως είναι παρόμοια για το TCP και για το UDP επειδή έχουν και τα δύο την ίδια προτεραιότητα. Η καθυστέρηση αυτή εξηγείται από το γεγονός ότι ο ελεύθερος αποθηκευτικός χώρος των δρομολογητών από κάποιο σημείο και μετά γεμίζει με αποτέλεσμα όλα τα πακέτα να λαμβάνουν τον ίδιο χρόνο καθυστέρησης.

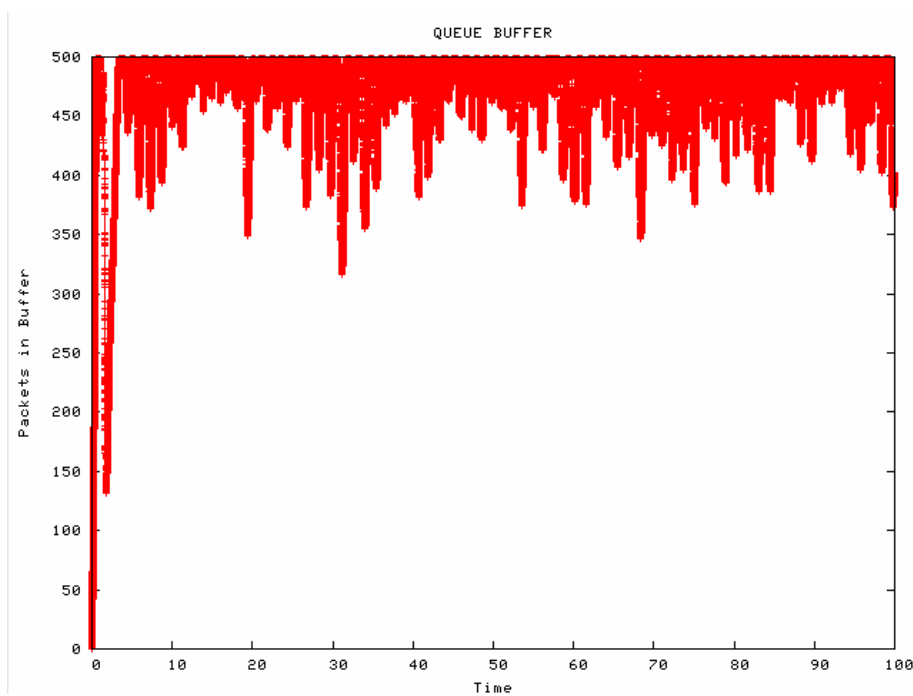
Ο χρόνος αυτός αρχίζει από την ώρα που ένα πακέτο φτάνει στο τέλος της ουράς του δρομολογητή και τελειώνει την ώρα που το συγκεκριμένο πακέτο φεύγει από τον δρομολογητή. Αφού η ουρά έχει πάντα σταθερό μέγεθος και είναι πάντα γεμάτη, σημαίνει πως η καθυστέρηση στους δρομολογητές από το σημείο που γεμίζει πλήρως ο αποθηκευτικός χώρος και μετά θα είναι σταθερή.

Για επαλήθευση της συγκεκριμένης ανάλυσης παρουσιάζουμε στα σχήμα 4.1 μέχρι 4.3 την συμπεριφορά του ελεύθερου αποθηκευτικού χώρου (buffer) των δρομολογητών σε συνάρτηση με τον χρόνο της προσομοίωσης. Η συμπεριφορά του αποθηκευτικού χώρου είναι ανάλογος για όλα τα πρωτόκολλα τα οποία εξετάζουμε, επομένως παρουσιάζουμε μόνο μερικά αποτελέσματα. Ο αποθηκευτικός χώρος του δρομολογητή είναι ορισμένος στα 500 πακέτα.



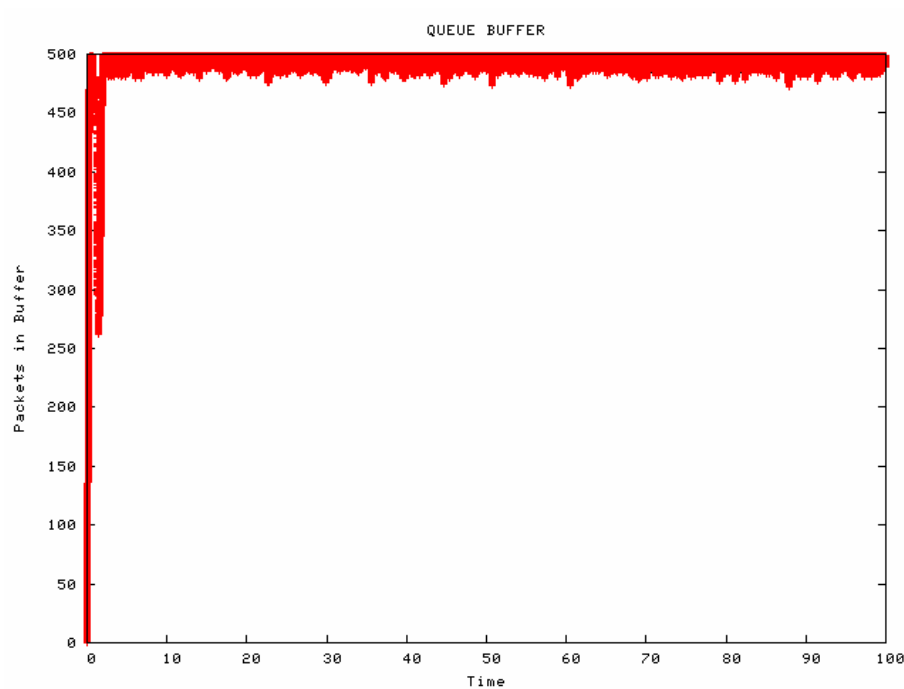
Σχήμα 4.1 : Μέγεθος Αποθηκευτικού χώρου για 10 ροές

Παρατηρούμε πως σε αριθμό ροών που δεν παρατηρείται συμφόρηση (Σχήμα 4.1) ο αποθηκευτικός χώρος του δρομολογητή δεν γεμίζει ποτέ με αποτέλεσμα η καθυστέρηση που παρατηρείται να βρίσκεται σε ένα σχετικά χαμηλό χρόνο.



Σχήμα 4.2 : Μέγεθος Αποθηκευτικού χώρου για 25 ροές

Με την αύξηση του αριθμού των ροών (Σχήμα 4.2) παρατηρούμε πως ο αποθηκευτικός χώρος του δρομολογητή φτάνει στα όρια του με αποτέλεσμα να αρχίσουν να χάνονται πακέτα και να αυξάνεται ο χρόνος καθυστέρησης. Βλέπουμε πως υπάρχουν χρονικές στιγμές στις οποίες ο χώρος αρχίζει να αδειάζει μέχρι τα 350 πακέτα αλλά και πάλι αυξάνεται απότομα φτάνοντας στο όριο των 500 πακέτων.



Σχήμα 4.2 : Μέγεθος Αποθηκευτικού χώρου για 50 ροές

Τέλος με την εισαγωγή ακόμη περισσότερων ροών (Σχήμα 4.3) παρατηρούμε πως ο ελεύθερος αποθηκευτικός χώρος γεμίζει πιο απότομα και εξακολουθεί να παραμένει γεμάτος σε όλη την διάρκεια της προσομοίωσης. Η συμπεριφορά του δρομολογητή είναι η ίδια από το σημείο αυτό και μετά άσχετα με των αριθμό των ροών που υπάρχουν στην προσομοίωση. Αυτός είναι και ο λόγος που από ένα σημείο και μετά η καθυστέρηση στους δρομολογητές παραμένει σταθερή.

Με την αλλαγή του ρυθμού αποστολής που επιτυγχάνεται με την μείωση του χρόνου μεταξύ αποστολής των πακέτων, βλέπουμε και πάλι πως το UDP στέλνει τα πακέτα του με τον ρυθμό αποστολής τον οποίο υπολογίζει πως χρειάζεται, και συνεχίζει να στέλνει με τον ρυθμό αυτό άσχετα εάν τα πακέτα χάνονται κατά την διάρκεια της αποστολής.

Συμπεραίνουμε τελικά πως ένα πρωτόκολλο όπως το UDP το οποίο δεν παρέχει κάποιο μηχανισμό ελέγχου συμφόρησης δεν είναι κατάλληλο για την αποστολή πολυμεσικών δεδομένων γιατί μπορεί να μην εκμεταλλεύεται το διαθέσιμο εύρος ζώνης όπως το TCP αλλά σε περίπτωση συμφόρησης οι ροές του δεν αφήνουν το εύρος ζώνης που είχαν πάρει αδιαφορώντας εάν τα πακέτα φτάνουν ή όχι στον προορισμό τους και οδηγώντας τις ροές άλλων πρωτοκόλλων σε αφανισμό.

4.2 Ανάλυση Αποτελεσμάτων Πρωτοκόλλου RTP

Το RTP είναι ένα πρωτόκολλο το οποίο τρέχει από πάνω από το UDP στη στοίβα δικτύου. Οι πληροφορίες που στέλνει το πρωτόκολλο ελέγχου RTCP όπως το ποσοστό χαμένων πακέτων και ο χρόνος γύρου αποστολής πρέπει να εκμεταλλευτούν από τον σχεδιαστή της εφαρμογής ο οποίος χρησιμοποιεί αυτό το πρωτόκολλο. Δυστυχώς στον προσομοιωτή μας οι πληροφορίες αυτές δεν λαμβάνονται υπόψη με αποτέλεσμα το πρωτόκολλο να συμπεριφέρεται περίπου όπως το UDP.

Παρατηρούμε στην πρώτη προσομοίωση ότι το ποσοστό εύρους ζώνης της σύνδεσης είναι κατειλημμένο από το RTP επειδή δεν λαμβάνεται υπόψη το ποσοστό χαμένων πακέτων. Το πρωτόκολλο στέλνει όλες τις ροές του με το επιθυμητό εύρος ζώνης που υπολογίζει και διατηρεί το εύρος αυτό σε όλες τις περιπτώσεις σε βάρος του TCP. Βλέπουμε πως το υπολογιζόμενο εύρος είναι μεγαλύτερο από αυτό που χρειάζεται το UDP λόγω των μεγαλύτερων και περισσότερων πεδίων που υπάρχουν στην επικεφαλίδα του RTP.

Το ποσοστό των χαμένων πακέτων αυξάνεται με τον ίδιο τρόπο που αυξάνεται και το TCP με την διαφορά ότι είναι σε κάπως μεγαλύτερα ποσοστά τα οποία αγγίζουν μέχρι και το 80%. Είναι κατανοητό πως με τέτοιο ποσοστό χαμένων πακέτων δεν μπορεί να επιβιώσει μια πολυμεσική εφαρμογή. Όσον αφορά την καθυστέρηση των πακέτων στους δρομολογητές και οι δύο τύποι πακέτων λαμβάνουν σχεδόν την ίδια καθυστέρηση.

Καταλήγουμε τελικά στο συμπέρασμα πως ένα πρωτόκολλο όπως το RTP δεν μπορεί να χρησιμοποιηθεί για την αποστολή πολυμεσικών δεδομένων έτσι όπως παρουσιάζεται στον προσομοιωτή. Υπάρχουν μεν οι επιθυμητές πληροφορίες ανατροφοδότησης αλλά αυτές δεν χρησιμοποιούνται οδηγώντας σε μη ικανοποιητική απόδοση το πρωτόκολλο μας.

4.3 Ανάλυση Αποτελεσμάτων Πρωτοκόλλου TFRC

Το πρωτόκολλο αυτό όπως είδαμε και στο κεφάλαιο 2 έχει σχεδιαστεί για να υποστηρίζει την αποστολή πολυμεσικών εφαρμογών. Ο μηχανισμός ελέγχου συμφόρησης όσον αφορά την αποστολή των δεδομένων είναι μια παραλλαγή του πρωτοκόλλου TCP. Ο μηχανισμός αυτός υπολογίζεται σε συνάρτηση του μεγέθους των πακέτων, του ποσοστού χαμένων πακέτων και του χρόνου γύρου αποστολής των πακέτων. Θεωρητικά το TFRC είναι δίκαιο με ροές TCP που χρησιμοποιούν το ίδιο μέγεθος πακέτων.

Σε γενικές γραμμές ο μηχανισμός ελέγχου συμφόρησης του TFRC δουλεύει με τον εξής τρόπο : Ο παραλήπτης μετρά το ποσοστό χαμένων και στέλνει το αποτέλεσμα πίσω στον αποστολέα. Ο αποστολέας χρησιμοποιεί αυτή την πληροφορία για να υπολογίζει το χρόνο γύρου αποστολής δεδομένων. Το ποσοστό χαμένων πακέτων και ο χρόνος γύρου πακέτων τροφοδοτούνται στην εξίσωση ρυθμού αποστολής ώστε να υπολογιστεί ο νέος ρυθμός αποστολής. Ο αποστολέας τέλος ρυθμίζει τον ρυθμό αποστολής να είναι ίσος με τον υπολογιζόμενο ρυθμό αποστολής.

Η παραλλαγή VoIP που χρησιμοποιήσαμε έχει την εξής ιδιομορφία. Λόγω του ότι είναι σχεδιασμένη για πολυμεσικές εφαρμογές πρέπει να διατηρεί συνεχώς ένα ελάχιστο ρυθμό αποστολής πακέτων. Ο ρυθμός αυτός για την παραλλαγή είναι καθορισμένος στα οκτώ πακέτα ανά χρόνο γύρου αποστολής πακέτων σε περίπτωση που δεν λαμβάνει μηνύματα ανατροφοδότησης από τον παραλήπτη. Σε περίπτωση μεγάλης συμφόρησης όπως στις προσομοιώσεις μας ο ρυθμός αυτός παραμένει σταθερός στα οκτώ πακέτα λόγω του ότι τα μηνύματα ανατροφοδότησης δεν φτάνουν στον αποστολέα με αποτέλεσμα οι ροές TFRC να εκμεταλλεύονται το εύρος ζώνης εις βάρος του TCP.

Έχοντας αυτό εις γνώση αναλύουμε τα αποτελέσματα τα οποία έχουν δώσει οι προσομοιώσεις μας. Στην παρουσίαση του ποσοστού εύρους ζώνης σε συνάρτηση των ροών βλέπουμε πως το TFRC σε μεγάλα ποσοστά συμφόρησης έχει ένα τεράστιο ποσοστό του εύρους ζώνης σε σχέση με τις ροές TCP. Ο λόγος είναι ότι ο μηχανισμός ελέγχου συμφόρησης είναι σχεδιασμένος να μην αντιδρά αυστηρά στην ύπαρξη συμφόρησης και να μειώνει αργά το ρυθμό αποστολής που διατηρεί.

Όσον αφορά το ποσοστό των χαμένων πακέτων και οι δύο ροές έχουν την ίδια συμπεριφορά και λαμβάνουν το ίδιο ποσοστό χαμένων πακέτων. Ο ελεύθερος χώρος των δρομολογητών γεμίζει με αποτέλεσμα όποια πακέτα φτάνουν και δεν έχουν χώρο να αποθηκευτούν, ρίχνονται ακριβώς με την ίδια πιθανότητα άσχετα με το μέγεθος των πακέτων.

Η καθυστέρηση που λαμβάνουν τα πακέτα του TFRC στην ύπαρξη μεγάλης συμφόρησης είναι παρόμοια με αυτή που λαμβάνουν τα πακέτα του TCP. Αυτό σημαίνει πως και οι δύο τύποι πακέτων λαμβάνουν την ίδια προτεραιότητα. Ένα πακέτο έχει την ίδια πιθανότητα να ριχτεί από τον δρομολογητή άσχετα από το μέγεθος του. Αυτός ο κανόνας ισχύει στο σημερινό διαδίκτυο.

Συμπεραίνουμε λοιπόν ότι παρότι το TFRC παρέχει ένα καλό μηχανισμό ελέγχου συμφόρησης σε περίπτωση όπου δεν υπάρχει συμφόρηση, εντούτοις σε μεγάλα ποσοστά συμφόρησης ο μηχανισμός αυτός δεν συμπεριφέρεται καθόλου καλά και επιμένει να διατηρεί ένα εύρος ζώνης για τις ροές TFRC το οποίο όμως δεν είναι χρήσιμο για τις πολυμεσικές εφαρμογές.

4.4 Ανάλυση Αποτελεσμάτων Πρωτοκόλλου SCTP

Θυμίζουμε από την περιγραφή του πρωτοκόλλου στο κεφάλαιο 2 ότι το SCTP έχει δημιουργηθεί για να συναγωνίζεται δίκαια ροές TCP που έχουν μέγεθος πακέτων 1500 ψηφιολέξεων συμπεριλαμβανομένων των πεδίων της επικεφαλίδας. Ο μηχανισμός ελέγχου συμφόρησης του SCTP έχει σχεδιαστεί για να ισχύει σε κάθε σύνδεση (session) και όχι σε κάθε υποροή (stream). Ο μηχανισμός αυτός είναι παρόμοιος με του TCP με δύο κύριες διαφορές.

Η πρώτη διαφορά είναι ότι το SCTP λόγω της ικανότητας σύνδεσης με περισσότερες της μίας διευθύνσεις προορισμού, διατηρεί ένα μηχανισμό ελέγχου συμφόρησης για κάθε διεύθυνση. Σε περίπτωση δηλαδή που ένα μονοπάτι έχει συμφόρηση, ο μηχανισμός ελέγχου συμφόρησης στο μονοπάτι αυτό, θα αναγκάσει τον ρυθμό αποστολής δεδομένων να μειωθεί. Με την αλλαγή του μονοπατιού θα υπάρχει ένας νέος μηχανισμός ελέγχου συμφόρησης ο οποίος θα είναι αρχικά στην φάση της αργής εκκίνησης με αποτέλεσμα να αυξάνει τον ρυθμό αποστολής δεδομένων εκθετικά μέχρι να ανιχνεύσει συμφόρηση και σε αυτή την σύνδεση.

Η δεύτερη κύρια διαφορά είναι ότι το SCTP μπορεί να παραδώσει τα πακέτα στην εφαρμογή άσχετα εάν έχει ανιχνευτεί ένα κενό σε μια υποροή. Δηλαδή για παράδειγμα εάν σε μια σύνδεση υπάρχουν πέντε υποροές, και έχουμε ανιχνεύσει κενό στην δεύτερη υποροή, τότε μπορούμε να παραδώσουμε τα πακέτα των υπόλοιπων υποροών στην εφαρμογή χωρίς να περιμένουμε να παραλάβουμε το πακέτο της δεύτερης υποροής που έχει χαθεί. Αυτή η ικανότητα του πρωτοκόλλου συγκριτικά με το TCP δίνει την δυνατότητα να υπάρχουν λιγότερες επαναληπτικές αποστολές πακέτων αφού τα πακέτα άλλων υποροών δεν επηρεάζονται από την ανίχνευση χαμένων πακέτων σε άλλες υποροές.

Όπως και το TCP, το SCTP χρησιμοποιεί τρεις μεταβλητές ελέγχου για να ρυθμίζει το ρυθμό αποστολής των δεδομένων. Η πρώτη μεταβλητή είναι το μέγεθος του ελεύθερου χώρου στον παραλήπτη, τον οποίο ο παραλήπτης στέλνει στον αποστολέα για να δείξει πόσα δεδομένα μπορεί να παραλάβει χωρίς να γεμίσει ο ελεύθερος χώρος. Η μεταβλητή αυτή υπάρχει για κάθε σύνδεση. Η δεύτερη μεταβλητή είναι το μέγεθος του παραθύρου αποστολής δεδομένων το οποίο διατηρεί ο αποστολέας και αλλάζει ανάλογα με την συμφόρηση που υπάρχει στο δίκτυο. Η μεταβλητή αυτή υπάρχει μια φορά για κάθε διεύθυνση προορισμού. Η τρίτη μεταβλητή είναι αυτή του κατωφλίου αργής εκκίνησης την οποία διατηρεί ο αποστολέας για να ξεχωρίσει τις φάσεις της αργής εκκίνησης από την φάση της αποφυγής συμφόρησης. Η μεταβλητή αυτή υπάρχει μία φορά για κάθε διεύθυνση προορισμού.

Τέλος μια τελευταία διαφορά με τον μηχανισμό του TCP είναι ότι υπάρχει η δυνατότητα αποστολής πολλών χαμένων δεδομένων σε ένα και μόνο πακέτο. Η τριπλή

αποστολή επιβεβαίωσης του ιδίου πακέτου σημαίνει πως υπάρχει ένα κενό στα δεδομένα μας. Το κομμάτι αυτό των δεδομένων στέλνεται ξανά, μαζί με άλλα κομμάτια δεδομένων τα οποία έχουν σημειωθεί προηγουμένως ως χαμένα. Με την μέθοδο αυτή μειώνεται η πιθανότητα να πρέπει να στείλουμε ξανά πακέτα τα οποία έχουμε ήδη ξαναστείλει και χαθεί λόγω μεγάλης συμφόρησης στην σύνδεση.

Βάση του πιο πάνω μηχανισμού αναλύουμε τα αποτελέσματα που έχουμε παρουσιάσει στο προηγούμενο κεφάλαιο. Αρχικά ελέγχουμε το ποσοστό χρήσης της σύνδεσης όπου υπάρχει συμφόρηση για τα δύο πρωτόκολλα (Σχήματα 3.18-3.20). Βλέπουμε πως οι δύο μηχανισμοί συναγωνίζονται σχεδόν δίκαια μεταξύ τους και δεν προσπαθεί να εκμεταλλευτεί ο ένας το εύρος ζώνης του άλλου. Ο λόγος που το SCTP έχει μεγαλύτερο εύρος ζώνης είναι γιατί έχουμε θέσει να μην υπάρχουν επαναληπτικές αποστολές έστω κι αν έχουμε απώλεια πακέτων. Επίσης ένας άλλος σημαντικός παράγοντας είναι ότι έχουμε πέντε υποροές για το SCTP οι οποίες λαμβάνουν ένα ελάχιστο εύρος ζώνης στην περίπτωση συμφόρησης που αθροιστικά είναι μεγαλύτερο από το ελάχιστο εύρος ζώνης που λαμβάνει το TCP.

Όσον αφορά την απώλεια των πακέτων έχουμε μια αναλογία με το ποσοστό του TCP αλλά σε μεγαλύτερη κλίμακα. Αυτό οφείλεται στο ότι κάθε υποροή του πρωτοκόλλου χάνει πακέτα τα οποία αθροιστικά δίνουν ένα ποσοστό μεγαλύτερο από αυτό του TCP. Γενικά όμως το ποσοστό χαμένων πακέτων είναι πολύ μεγάλο για να χρησιμοποιηθεί για πολυμεσικές εφαρμογές.

Ο χρόνος καθυστέρησης τόσο στα άκρα όσο και στους δρομολογητές στην σύνδεση όπου υπάρχει συμφόρηση αυξάνει με την αύξηση των ροών και κατ' επέκταση της συμφόρησης, σχεδόν το ίδιο και για τα δύο πρωτόκολλα. Ο λόγος που γίνεται αυτό μπορεί να ερμηνευτεί πολύ εύκολα. Όσο αυξάνεται η συμφόρηση ο ελεύθερος αποθηκευτικός χώρος στους δρομολογητές μειώνεται σημαντικά και από κάποιο σημείο και μετά, τα πακέτα αρχίζουν να ρίχνονται γιατί δεν υπάρχει χώρος για να αποθηκευτούν. Τα πακέτα που φτάνουν και αποθηκεύονται τελικά στους δρομολογητές έχουν μπροστά τους πολλά άλλα πακέτα που και αυτά περιμένουν να έρθει η σειρά τους για να συνεχίσουν την διαδρομή τους. Επομένως από ένα σημείο και μετά όπου ο

ελεύθερος αποθηκευτικός χώρος των δρομολογητών είναι γεμάτος, ο μέσος χρόνος καθυστέρησης των πακέτων παραμένει σταθερός.

Από την ανάλυση του μηχανισμού ελέγχου συμφόρησης για το SCTP φτάνουμε στο συμπέρασμα ότι δεν είναι κατάλληλο για πολυμεσικές εφαρμογές. Μπορεί να παρέχει ένα καλό μηχανισμό ελέγχου συμφόρησης αλλά ο μηχανισμός αυτός δεν ανταποκρίνεται στις προσδοκίες μας με την χρήση μη αξιόπιστων ροών.

ΚΕΦΑΛΑΙΟ 5

Νέες Τεχνολογίες Πρωτοκόλλων

5.1 Η προσθήκη του ECN στο IP	121
5.2 Εύρωστο ECN με αποστολή μηνυμάτων	122

5.1 Η προσθήκη του ECN στο IP

Το ECN [7] είναι ένας μηχανισμός δύο bits που ορίζεται στην επικεφαλίδα του IP, και έχει σκοπό να αναγνωρίσει συμφόρηση στο δίκτυο χωρίς να χρειαστεί να χάσουμε κάποια πακέτα λόγω συμφόρησης. Η μεταβολή στον ρυθμό μετάδοσης δεδομένων μετά από την παραλαβή κάποιου πακέτου με ενεργοποιημένο τον μηχανισμό ECN, πρέπει να είναι η ίδια με αυτή στην περίπτωση που θα είχαμε απώλεια πακέτου.

Σε περίπτωση που αρχίζει να εμφανίζεται συμφόρηση στο δίκτυο, υπάρχει η δυνατότητα να μαρκάρουμε κάποια πακέτα αντί να τα ρίχνουμε. Για να γίνει αυτό κάνουμε χρήση του πεδίου ECN στην επικεφαλίδα του πρωτοκόλλου μεταφοράς δεδομένων που περιλαμβάνει δύο bits με τα οποία σχηματίζουμε τους κωδικούς από 00 μέχρι 11. Οι κωδικοί 10 και 01 ορίζονται από τον αποστολέα των δεδομένων για να δείξουν ότι τα άκρα της σύνδεσης υποστηρίζουν το ECN. Οι δρομολογητές χειρίζονται τους κωδικούς 01 και 10 ως ισοδύναμους και οι αποστολείς έχουν την δυνατότητα να επιλέξουν όποιο από τους δυο κωδικούς επιθυμούν. Η χρήση του κωδικού 00 προσδιορίζει ένα πακέτο που δεν χρησιμοποιεί ECN ενώ ο κωδικός 11 ορίζεται από τους δρομολογητές για να δείξουν την ύπαρξη συμφόρησης στους ακραίους κόμβους.

Πιο κάτω παρουσιάζουμε τους κωδικούς για τα bits έξι και επτά που αντιστοιχούν στο πεδίο ECN του IP.

Binary	Keyword
00	Not – ECN
01	ECT(1) (ECN-Capable Transport(1))
10	ECT(0) (ECN-Capable Transport(0))
11	CE (Congestion Experienced)

Πίνακας 5.1 – Κωδικοί στο πεδίο ECN της επικεφαλίδας του IP

Με την παραλαβή ενός πακέτου όπου στο πεδίο ECN υπάρχει η ένδειξη συμφόρησης (11), ο αποστολέας ανταποκρίνεται με τον ίδιο τρόπο όπως θα έκανε στην περίπτωση που το πακέτο θα είχε χαθεί. Δηλαδή εάν είχαμε χρήση του TCP πρωτοκόλλου ο αποστολέας θα μείωνε τον ρυθμό αποστολής δεδομένων στο μισό.

Ένας δρομολογητής ορίζει τον κωδικό 11 σε ένα πακέτο για να ειδοποιήσει για ύπαρξη συμφόρησης, μόνο εάν το πακέτο κάνει χρήση του ECN και εάν το πακέτο θα ριχνόταν από τον δρομολογητή υπό άλλες συνθήκες. Σημειώνουμε πως το buffer του δρομολογητή σε αυτή την περίπτωση δεν είναι γεμάτο. Εάν ένας δρομολογητής πάρει ένα πακέτο με το κωδικό 11 το χειρίζεται κανονικά όπως κάθε άλλο πακέτο. Όταν το buffer του δρομολογητή γεμίσει τότε αναγκαστικά θα ρίξει κάποια πακέτα ασχέτως με την ένδειξη στο πεδίο ECN.

Ο παραλήπτης από την στιγμή που θα πάρει ένα πακέτο με την ένδειξη CE τότε στις επιβεβαιώσεις πακέτων που ακολουθούν ενεργοποιεί την σημαία ECE (ECN-echo) και εξακολουθεί να στέλνει την ένδειξη αυτή σε διαδοχικές επιβεβαιώσεις μέχρι να πάρει από τον αποστολέα την ένδειξη CWR που σημαίνει πως ο αποστολέας έχει ανταποκριθεί στην σήμανση για συμφόρηση και μείωσε τον ρυθμό αποστολής των δεδομένων.

5.2 ECN με αποστολή μηνυμάτων

Το ECN-nonce είναι μια επιπρόσθετη επιλογή του ECN που προστατεύει από τυχαία ή σκόπιμη παραπλάνηση, τα μαρκαρισμένα πακέτα που στέλνονται από τους TCP αποστολείς. Βελτιώνει την δύναμη του ελέγχου συμφόρησης με το να εμποδίζει τους παραλήπτες να εκμεταλλευτούν το ECN για να πάρουν ένα άδικο μερίδιο του εύρους

της σύνδεσης. Το ECN-nonce χρησιμοποιεί τους δύο κωδικούς ενεργοποίησης όπως το ECN στην επικεφαλίδα του IP και χρειάζεται μια σημαία στην επικεφαλίδα του TCP. Είναι υπολογιστικά αποτελεσματικό τόσο για τους δρομολογητές όσο και για τους host.

Ο σκοπός του ECN-nonce είναι να επιβεβαιώνει την σωστή συμπεριφορά του ECN παραλήπτη και να προστατεύει από λάθη στην υλοποίηση και λανθασμένη χρήση του ECN από τους παραλήπτες. Πιο συγκεκριμένα το ECN-nonce μπορεί να ανακαλύψει ένα παραλήπτη που δυσλειτουργεί με μεγάλη πιθανότητα, δεν αλλοιώνει την κανονική λειτουργία του ECN, ούτε μειώνει τα ωφέληματα από το ECN για τους παραλήπτες που συμπεριφέρονται κανονικά. Είναι φτηνό όσο αφορά το υπολογιστικό κόστος και τις προσθήκες στις επικεφαλίδες των πακέτων, (μια σημαία στην επικεφαλίδα του TCP). Τέλος είναι απλό και μέχρι σήμερα δεν δέχεται επιθέσεις.

Το ECN-nonce στηρίζεται στους ήδη υπάρχοντες μηχανισμούς του ECN-Echo και του Congestion Window Reduced (CWR). Πιο κάτω περιγράφουμε τον τρόπο λειτουργίας του ECN-nonce σε μια κατεύθυνση αν και τρέχει παράλληλα και προς τις δύο κατευθύνσεις. Το ECN πρωτόκολλο παραμένει αναλλοίωτο εκτός από την προσθήκη ενός νέου πεδίου στην επικεφαλίδα του TCP.

Αρχικά ένας από τους κωδικούς ECT(0) και ECT(1) τοποθετείται στο πεδίο ECN της επικεφαλίδας του IP, στο πακέτο που θα σταλεί. Οι δρομολογητές που έχουν συμφόρηση αλλάζουν το πεδίο του πακέτου σε CE. Όταν ο παραλήπτης πάρει πακέτο με την ένδειξη CE η σημαία ECE (ECN-echo) στέλνεται σε διαδοχικές επιβεβαιώσεις πακέτων μέχρι να πάρει πακέτο με την σημαία CWR. Η σημαία αυτή στέλνεται με ένα νέο πακέτο αφού πρώτα ο αποστολέας ανταποκριθεί στην ένδειξη για συμφόρηση με μείωση του παραθύρου αποστολής δεδομένων.

Το ECN-nonce προσθέτει σε αυτό το πρωτόκολλο και επιτρέπει στον παραλήπτη να δείξει πίσω στον αποστολέα ότι τα πακέτα που παραλαμβάνει δεν ήταν μαρκαρισμένα. Μία τυχαία τιμή ενός bit (nonce) κωδικοποιείται σε κάθε ECT κωδικό. Το άθροισμα αυτών των nonces επιστρέφεται στην επικεφαλίδα του TCP στο πεδίο που προστέθηκε για το σκοπό αυτό και είναι το NS (nonce sum) bit. Με την ύπαρξη συμφόρησης ο κωδικός στο ECT διαγράφεται για να γραφτεί στην θέση του ο κωδικός CE. Έτσι τώρα

ο παραλήπτης θα πρέπει να μαντέψει την σωστή αρχική τιμή του bit ώστε να στείλει πίσω στον αποστολέα το σωστό άθροισμα. Αφού το άθροισμα είναι μόνο ένα bit ο αποστολέας έχει πιθανότητα 50% να αναγνωρίσει ένα παραλήπτη ο οποίος προσπαθεί να τον ξεγελάσει. Επειδή η διαδικασία αυτή γίνεται για κάθε πακέτο, ένας παραλήπτης που προσπαθεί να ξεγελάσει τον αποστολέα θα ανακαλυφθεί πολύ γρήγορα. Μετά από αναγνώριση συμφόρησης ο αποστολέας μειώνει τον ρυθμό αποστολής δεδομένων και συγχρονίζει το άθροισμα του, κάνοντας το, το ίδιο με αυτό του παραλήπτη.

Αποστολέας	Παραλήπτης (Αρχικό Άθροισμα 1)
(1:4) ECT(0)	$NS = 1 + 0(1:4) = 1$
ACK 4, NS=1	
(4:8) ECT(1)	$NS = 1 + 1(4:8) = 0$
ACK 8, NS=0	
(8:12) ECT(1)	$NS = 0 + 1(8:12) = 1$
ACK 12, NS=1	
(12:16) ECT(1)	$NS = 1 + 1(12:16) = 0$
ACK 16, NS=0	

Πίνακας 5.2 : Παράδειγμα υπολογισμού του nonce sum στο παραλήπτη

Πιο κάτω δείχνουμε τον υπολογισμό του nonce sum στον παραλήπτη όταν το πακέτο (4:8) είναι μαρκκαρισμένο. Ο παραλήπτης μπορεί να υπολογίσει το λανθασμένο άθροισμα όταν το αρχικό nonce χάνεται μετά το μαρκάρισμα του πακέτου.

Αποστολέας	Παραλήπτης (Αρχικό Άθροισμα 1)
(1:4) ECT(0)	$NS = 1 + 0(1:4) = 1$
ACK 4, NS=1	
(4:8) ECT(1) - CE	$NS = 1 + ?(4:8) = 1$
ACK 8, ECE, NS=1	
(8:12) ECT(1), CWR	$NS = 1 + 1(8:12) = 0$
ACK 12, NS=0	
(12:16) ECT(1)	$NS = 0 + 1(12:16) = 1$
ACK 16, NS=1	

Πίνακας 5.3 : Υπολογισμός nonce sum στο παραλήπτη σε μαρκκαρισμένο πακέτο

Συμπεριφορά Αποστολέα κατά την αποστολή

Οι αποστολές χειρίζονται τα CWR και τα ECN-echo όπως προηγουμένως. Επιπρόσθετα τοποθετούν nonces στα πακέτα καθώς μεταδίδονται και ελέγχουν την εγκυρότητα των αθροισμάτων των nonces στις επιβεβαιώσεις του παραλήπτη.

Για την τοποθέτηση του ενός bit nonce σε κάθε ECN-capable IP πακέτο, ο αποστολέας χρησιμοποιεί τους δύο ECT κωδικούς: το ECT(0) αντιπροσωπεύει τον κωδικό nonce 0 και το ECT(1) αντιπροσωπεύει τον κωδικό nonce 1. Όπως και στο ECN τα πακέτα που ξαναστέλλονται δεν είναι ECN επομένως δεν χρησιμοποιούν nonces.

Ο αποστολέας διατηρεί ένα πίνακα με το επιμέρους άθροισμα του nonce για κάθε πακέτο που στέλνει ώστε να μπορεί να το συγκρίνει με το άθροισμα που παίρνει από τις επιβεβαιώσεις των πακέτων.

Συμπεριφορά των δρομολογητών

Οι δρομολογητές μαρκάρουν τα πακέτα με την ένδειξη CE σε περίπτωση συμφόρησης. Με αυτή την ενέργεια η αρχική τιμή του nonce στο ECT(0) ή στο ECT(1) διαγράφεται. Ούτε ο παραλήπτης ούτε κανένα άλλο μέρος μπορούν να ξεμαρκάρουν το πακέτο χωρίς να μαντέψουν την αρχική τιμή του nonce.

Συμπεριφορά παραλήπτη κατά την λήψη και την αποστολή

Οι παραλήπτες διατηρούν το επιμέρους άθροισμα των nonces καθώς τα πακέτα φτάνουν με την σωστή σειρά και επιστρέφουν αυτό το άθροισμα στις επιβεβαιώσεις των πακέτων. Η επιστροφή του αθροίσματος πίσω στον αποστολέα είναι προαιρετικό, αλλά αυτό δίνει το δικαίωμα στον αποστολέα να σταματήσει να στέλνει ECN πακέτα.

Καθώς τα πακέτα αφαιρούνται από την ουρά των εκτός σειράς πακέτων, παίρνουν το nonce από την επικεφαλίδα του IP. Το nonce προστίθεται στο επιμέρους άθροισμα των nonces και στέλνεται πίσω στον αποστολέα μέσα στην επιβεβαίωση της παραλαβής του πακέτου.

Σε περίπτωση ενός μαρκαρισμένου πακέτου με την ένδειξη CE ο παραλήπτης δεν γνωρίζει την αρχική τιμή του nonce και υποθέτει μια δική του τιμή, συνήθως την τιμή 0. Η αρχική τιμή του αθροίσματος είναι 1 και καθορίζεται την ώρα της δημιουργίας της σύνδεσης.

Το ECN-nonce είναι μια απλή παραλλαγή του ECN που βελτιώνει την αξιοπιστία του ECN με το να εμποδίζει τους παραλήπτες να αποκρύπτουν μαρκαρισμένα ή χαμένα πακέτα. Είναι πολύ εύκολη η υλοποίηση του στο δίκτυο. Χρησιμοποιεί τους κωδικούς ECT(0) και ECT(1) και μια σημαία στην επικεφαλίδα του TCP, όπως και το CWR και το ECN-echo και έχει απλούς κανόνες λειτουργίας.

ΚΕΦΑΛΑΙΟ 6

Συμπεράσματα

6.1 Συμπέρασμα Ατομικής Διπλωματικής Εργασίας	127
6.2 Εισηγήσεις για μελλοντική Εργασία	130

6.1 Συμπέρασμα Ατομικής Διπλωματικής Εργασίας

Μέχρι σήμερα δεν έχουν γίνει αρκετές προσομοιώσεις χρησιμοποιώντας τα πρωτόκολλα που αναλύσαμε στα προηγούμενα κεφάλαια (TCP, UDP, RTP, TFRC, SCTP), τα οποία να μεταφέρουν πολυμεσικά δεδομένα. Οι περισσότερες προσομοιώσεις είχαν γίνει δίνοντας ως φορτίο αρχεία δεδομένων τα οποία δεν είναι κατάλληλα για χρήση με τα πρωτόκολλα αυτά και σε συνθήκες όπου δεν υπήρχε συμφόρηση, επομένως τα συμπεράσματα τα οποία είχαν εξαχθεί δημιουργούσαν μια λάθος εντύπωση ως προς την δυνατότητα χρήσης των πρωτοκόλλων αυτών.

Οι πολυμεσικές εφαρμογές όπως είχαμε αναφέρει, χρειάζονται ένα ελάχιστο ρυθμό αποστολής δεδομένων και μπορούν να ανεχθούν ένα μικρό ποσοστό χαμένων πακέτων χωρίς να επηρεάζεται η εφαρμογή μας. Οι προσομοιώσεις μας, έδειξαν πως τα πρωτόκολλα αυτά έχουν καλή συμπεριφορά σε περίπτωση όπου δεν υπάρχει συμφόρηση στο δίκτυο, ενώ σε περιπτώσεις με μεγάλη συμφόρηση, τα πρωτόκολλα αυτά λειτουργούν σε βάρος άλλων ροών χωρίς τα ίδια να πετυχαίνουν τον σκοπό τους.

Η προσπάθεια των πρωτοκόλλων να διατηρήσουν ένα ελάχιστο ρυθμό αποστολής σε περίπτωση συμφόρησης έρχεται σε σύγκρουση με το μεγάλο ποσοστό χαμένων πακέτων το οποίο έχουν οι ροές αυτές. Αποτέλεσμα είναι το εύρος ζώνης στην σύνδεση συμφόρησης να είναι πλήρως χρησιμοποιημένο αλλά καμιά από τις ροές να μην

καταφέρνει να παραδώσει τα πακέτα με τις ελάχιστες συνθήκες τις οποίες πρέπει να τηρούν οι πολυμεσικές εφαρμογές. Αναλυτικότερα εντοπίζουμε τα πιο κάτω προβλήματα στα πρωτόκολλα όσο αφορά τους δύο κύριους κανόνες που πρέπει να τηρούν οι πολυμεσικές εφαρμογές, δηλαδή την διατήρηση ενός ελάχιστου ρυθμού αποστολής και την ανοχή σε ένα μικρό ποσοστό χαμένων πακέτων τόσο σε περίπτωση ύπαρξης συμφόρησης όσο και σε περιπτώσεις μη ύπαρξης συμφόρησης.

Το UDP δεν παρέχει κάποιο μηχανισμό ελέγχου συμφόρησης. Αποτέλεσμα αυτού είναι σε περίπτωση μη ύπαρξης συμφόρησης να διατηρεί ένα ρυθμό αποστολής που είναι όσος χρειάζεται για την μεταφορά των δεδομένων. Επίσης η μη ύπαρξη συμφόρησης δεν παρέχει μεγάλα ποσοστά χαμένων πακέτων επομένως το πρωτόκολλο λειτουργεί ικανοποιητικά σε περίπτωση μη ύπαρξης συμφόρησης. Σε περίπτωση όμως που υπάρχει συμφόρηση το πρωτόκολλο επιμένει να διατηρεί τον ελάχιστο ρυθμό αποστολής άσχετα εάν ο ρυθμός που καταφέρνει να αποκτήσει είναι μικρότερος από τον επιθυμητό. Τότε υπάρχει ένα τεράστιο ποσοστό χαμένων πακέτων και έχουμε κράτηση του εύρους ζώνης το οποίο αποβαίνει μοιραίο για άλλες ροές άλλων πρωτοκόλλων τις οποίες τείνει να εξαφανίσει. Επομένως σε περίπτωση συμφόρησης το UDP κρατά άσκοπα εύρος ζώνης και καμιά ροή δεν μπορεί να λειτουργήσει ικανοποιητικά πράγμα το οποίο δεν είναι επιθυμητό στο σημερινό διαδίκτυο.

Το TFRC παρέχει ένα μηχανισμό ο οποίος σε περίπτωση συμφόρησης τείνει να μειώνει τον ρυθμό αποστολής αλλά όχι απότομα ανάλογα με της πληροφορίες που λαμβάνει από τον παραλήπτη. Σε περίπτωση που δεν υπάρχει συμφόρηση ο ρυθμός αυτός είναι μικρότερος από το μέσο εύρος ζώνης που θα δικαιούταν κανονικά και δεν υπάρχουν χαμένα πακέτα πράγμα που υποδηλώνει την σωστή λειτουργία του πρωτοκόλλου. Σε περίπτωση όμως που υπάρχει συμφόρηση το πρωτόκολλο προσπαθεί να διατηρήσει τον ελάχιστο ρυθμό αποστολής έστω και αν δεν τα καταφέρνει. Ο μηχανισμός ελέγχου συμφόρησης λέει πως σε περίπτωση όπου δεν έχουμε ανατροφοδοτήσεις από τον αποστολέα, που συμβαίνει σε περιπτώσεις μεγάλης συμφόρησης, το ελάχιστο που μπορούμε να στέλνουμε είναι οκτώ πακέτα ανά χρόνο αποστολής. Ο ρυθμός αυτός έχει ως αποτέλεσμα οι εφαρμογές να μην μπορούν να λάβουν το ελάχιστο εύρος ζώνης και να λειτουργούν εις βάρος άλλων πρωτοκόλλων. Το ποσοστό χαμένων πακέτων στην περίπτωση αυτή εξακολουθεί να είναι πάρα πολύ μεγάλο.

Το RTP έχει παρόμοια συμπεριφορά όπως αυτή του UDP. Ο μηχανισμός ελέγχου συμφόρησης υπάρχει απλά για να στέλνει πληροφορίες πίσω στον παραλήπτη ο οποίος είναι υπεύθυνος να ερμηνεύσει σωστά τα αποτελέσματα ώστε να προσαρμόσει σωστά τον νέο ρυθμό αποστολής. Σε περίπτωση μη ύπαρξης συμφόρησης το πρωτόκολλο υπολογίζει τον ρυθμό αποστολής δεδομένων και κρατά τον ρυθμό αυτό άσχετα από την συμφόρηση που υπάρχει στο δίκτυο πράγμα που εξασφαλίζει σωστή λειτουργία σε περίπτωση μη συμφόρησης. Σε περίπτωση όμως που υπάρχει συμφόρηση το πρωτόκολλο επιμένει να διατηρεί τον ελάχιστο ρυθμό αποστολής άσχετα εάν δεν το καταφέρνει με αποτέλεσμα να οδηγεί σε μηδενισμό το εύρος ζώνης των άλλων πρωτοκόλλων. Το ποσοστό χαμένων πακέτων παραμένει αρκετά μεγάλο με αποτέλεσμα το πρωτόκολλο να μην μπορεί να χρησιμοποιηθεί για την μεταφορά των πολυμεσικών εφαρμογών.

Τέλος το SCTP παρέχει ένα μηχανισμό ο οποίος έχει δημιουργηθεί για να είναι δίκαιος όσο αφορά τις ροές TCP. Οι προσομοιώσεις μας έδειξαν σε αντίθεση με τα άλλα πρωτόκολλα ότι δεν είναι τόσο εχθρικό απέναντι του. Όμως το εύρος ζώνης το οποίο εξασφαλίζει στην ύπαρξη συμφόρησης δεν είναι ικανοποιητικό για να εξασφαλίσει το ελάχιστο εύρος ζώνης που χρειάζονται οι πολυμεσικές εφαρμογές. Επίσης το ποσοστό των χαμένων πακέτων έστω και αν είναι το μικρότερο από όλα τα άλλα πρωτόκολλα εντούτοις είναι μεγαλύτερο από αυτό που απαιτούν οι πολυμεσικές εφαρμογές ως ελάχιστο.

Παρατηρούμε τελικά πως κανένα από τα πρωτόκολλα τα οποία χρησιμοποιήσαμε δεν είναι ικανό να ικανοποιήσει όλες τις απαιτήσεις των πολυμεσικών εφαρμογών και παράλληλα να είναι δίκαιο σε σχέση με άλλα πρωτόκολλα τα οποία μεταφέρουν αρχεία δεδομένων.

Το πρόβλημα είναι φανερό πως βρίσκεται στους μηχανισμούς ελέγχου συμφόρησης των πρωτοκόλλων. Οι μηχανισμοί αυτοί ίσως θα πρέπει να προβλέπουν σε περίπτωση συμφόρησης, την μην επιτυχή αποστολή των πολυμεσικών πακέτων και να μην αναγκάζουν τις ροές να διατηρούν ένα ελάχιστο ρυθμό αποστολής.

Σε συζητήσεις σε διαδικτυακά κανάλια άρχισε να γίνεται συζήτηση για την ανάγκη δημιουργίας ενός νέου πρωτοκόλλου το οποίο θα μπορεί να ξεπερνά τα προβλήματα αυτά. Σίγουρα το μέλλον κρύβει πολλή έρευνα για την σχεδίαση και υλοποίηση νέων πρωτοκόλλων τα οποία θα λειτουργούν ικανοποιητικά στην μεταφορά της νέας ανάγκης που δημιουργεί το σημερινό διαδίκτυο, αυτή της μεταφοράς των πολυμεσικών δεδομένων.

6.2 Εισηγήσεις για μελλοντική Εργασία

Η έρευνα για την ανεύρεση ενός νέου πρωτοκόλλου το οποία θα μπορεί να μεταφέρει πολυμεσικά δεδομένα είναι ένας από τους κύριους στόχους της διαδικτυακής ομάδας σχεδίασης πρωτοκόλλων. Επίσης πολλές από τις δυνατότητες των ήδη υπαρχόντων πρωτοκόλλων δεν έχουν ακόμη εξερευνηθεί.

Οι συνεχείς προσομοιώσεις των πρωτοκόλλων αυτών μπορούν να φέρουν στο φως παρατηρήσεις οι οποίες δεν έχουν ανιχνευτεί μέχρι σήμερα και να οδηγήσουν στην εξέλιξη των πρωτοκόλλων ώστε να βελτιωθεί η συμπεριφορά τους κατά την μεταφορά πολυμεσικών εφαρμογών.

Μεγάλη έμφαση θα πρέπει να δοθεί και στην κατανόηση της λειτουργίας του μηχανισμού ελέγχου συμφόρησης του κάθε πρωτοκόλλου ώστε να μπορεί να τροποποιηθεί κατάλληλα για να χρησιμοποιηθεί βάση των απαιτήσεων των εφαρμογών. Ο μηχανισμός είναι το κλειδί στην βελτιστοποίηση των πρωτοκόλλων και κατ' επέκταση στην καλύτερη λειτουργία του σημερινού διαδικτύου.

Βιβλιογραφία

- [1] Computer Networking , A Top-Down Approach Featuring the Internet, Second Edition , James F. Kurose, Keith W. Ross Chapter 3 p.196-199, p.228-246, Chapter 6 p.531 – 555.
- [2] SCTP : State of the Art in Research Products, and Technical Challenges, Shaojian Fu and Mohammed Atiquzzaman.
- [3] TFRC with Self Limiting Sources, Tom Phelan Oct 2004.
- [4] Investigating of MPEG-4 Video Streaming over SCTP, Alex Balk, Marc Sigler, Mario Gerla, M.Y.Sanadidi.
- [5] Ιστοσελίδα για πληροφορίες TFRC : www.icir.org/tfrc.
- [6] Ιστοσελίδα για πληροφορίες του Network Simulator 2 : www.isi.edu/nsnam/ns.
- [7] Ιστοσελίδα για πληροφορίες του ECN : www.icir.org/floyd/ecn.html.
- [8] Ιστοσελίδα για πληροφορίες προσομοίωσης πραγματικού βίντεο στον NS2 www.research.att.com/~breslau/vint/trace.html.
- [9] R. Stewart et al., “Stream Control Transmission Protocol”, IETF RFC 2960, Oct. 2000.
- [10] R. Stewart and C. Metz, “SCTP : New Transport Protocol for TCP/IP”, IEEE Internet Comp., vol. 5, no.6, Nov./Dec. 2001 pp. 64-69.
- [11] A. L. Caro et al., “SCTP : A Proposed Standard for Robust Internet Data Transport”, IEEE Comp. vol.36, no.11, Nov. 2003, pp. 56-63.
- [12] A. Jungmaier, “Performance Evaluation of the Stream Control Transmission Protocol”, Proc. IEEE Conf. on High Perf. Switching and Routing, Heidelberg, Germany, June 2000, pp. 141-148.
- [13] R. Alamgir, M. Atiquzzaman, and W. Ivanoncic, “Effect of Congestion Control on the Performance of TCP and SCTP”, NASA Earth Sci. Tech. Conf., Pasadena CA, June 2002.
- [14] A. L. Caro et al. “Improving Multimedia Performance over Lossy Networks via SCTP,” ATIRP 2001, College Park, MD, Mar.2001.
- [15] M. Handley, S. Floyd, J. Padhye, and J. Widmer, TCP Friendly Rate Control (TFRC) : Protocol Specification, RFC 3448, Proposed Standard, January 2003 .

- [16] S. Floyd, E. Kohler, TCP Friendly Rate Control (TFRC) for Voice : VoIP Variant and Faster Restart, November 2004.
- [17] M. Allman, V. Paxson, and W. Stevens. TCP Congestion Control. RFC 2581.
- [18] S. Casner, V. Jacobson, RTP: A Transport Protocol for Real-Time Applications, January 1996.
- [19] G. Fernando, V. Goyal, RTP Payload Format for MPEG1/MPEG2 Video, RFC 2038 October 1996.
- [20] J. Postel , User Datagram Protocol, RFC 768 , August 1980.

ΠΑΡΑΡΤΗΜΑ Α

Network Simulator

Η ανταλλαγή πληροφοριών στις μέρες μας στηρίζεται κυρίως στην χρήση των δικτύων. Τα δίκτυα είναι συστήματα που υλοποιούν τόσο το υλικό όσο και το λογισμικό ώστε να παραδώσουν με επιτυχία την πληροφορία σε όλες τις μορφές της. Γενικά το υλικό κομμάτι του δικτύου σχηματίζει τις γέφυρες μέσα από τις οποίες θα ταξιδέψει η πληροφορία χρησιμοποιώντας το λογισμικό. Κάθε άτομο που θέλει να σχεδιάσει ένα δίκτυο πρέπει να επιλέξει προσεκτικά τόσο το υλικό όσο και το λογισμικό, ώστε να εξασφαλίσει την ευρωστία και την απόδοση των δεδομένων που διακινούνται στο δίκτυο. Καθώς οι επικοινωνίες εξελίσσονται μέρα με την μέρα, τα δίκτυα τείνουν να γίνονται όλο και πιο πολύπλοκα, καθιστώντας την φάση της σχεδίασης των δικτύων όλο και πιο δύσκολη. Στην σχεδίαση πιο πολύπλοκων δικτύων, η πιθανότητα λάθος εκτίμησης στην φάση της σχεδίασης είναι αυξημένη με αποτέλεσμα τη σχεδίαση μη ισορροπημένων και ανεπαρκών δικτύων. Αυτός είναι ο λόγος που χρειάζεται κάποια μορφής προσομοίωση ανάμεσα στις φάσεις της σχεδίασης και της φάσης της υλοποίησης των δικτύων.

Για να προσομοιώσουμε τα σχέδια των δικτύων, μπορούν να χρησιμοποιηθούν κάποια CASE (Computer Aided Software Engineering) εργαλεία, που βοηθούν τους σχεδιαστές να ελέγξουν τα σχέδια των δικτύων τους. Ένα τέτοιο εργαλείο προσομοίωσης είναι και ο Network Simulator.

Ανάμεσα στις δυνατότητες του εργαλείου αυτού είναι η δημιουργία τοπολογιών με την δημιουργία κόμβων και συνδέσεων. Επίσης έχουμε την δυνατότητα δημιουργίας agents που υλοποιούν διαφορετικές συνδέσεις μεταξύ των κόμβων με την χρήση πρωτοκόλλων όπως TCP και UDP. Έχουμε την δυνατότητα να προσθέτουμε δεδομένα στο δίκτυο και να παρακολουθήσουμε την ροή των δεδομένων αυτών ώστε να εξάγουμε χρήσιμα αποτελέσματα. Για καλύτερη αναπαράσταση υπάρχει δυνατότητα για γραφική παρουσίαση των αποτελεσμάτων.

ΠΑΡΑΡΤΗΜΑ Β

Ενδεικτικός Κώδικας Προσομοιώσεων για UDP

```
proc init {BuffSize} {  
  
    global ns SimTime packetSize window queueType trace_file trace_file1 trace_file2  
  
    set packetSize 1000  
    set SimTime 100.0  
  
    set old_data 0  
  
    set ns [new Simulator]  
  
    # ecn capability  
    Agent/TCP set ecn_ 1  
  
    Agent/TCP/Newreno set packetSize_ $packetSize  
  
    Queue set limit_ $BuffSize  
  
    $ns trace-all [open all_traces.tr w]  
}
```

```
proc topo {} {  
  
    global ns node_ queueType  
  
    # Basic topology  
    #  
    # [node_(0)]---+    +---[node_(4)]  
    #             |    |  
    # [node_(2)]-----[node_(3)]  
    #             |    |  
    # [node_(1)]---+    +---[node_(5)]  
    #             |---udp flows----|  
  
    # Create nodes  
    set node_(0) [$ns node]  
    set node_(1) [$ns node]  
    set node_(2) [$ns node]  
    set node_(3) [$ns node]  
    set node_(4) [$ns node]
```

```

set node_(5) [$ns node]

$ns duplex-link $node_(0) $node_(2) 100Mb 10ms DropTail
$ns duplex-link $node_(3) $node_(4) 100Mb 10ms DropTail

# The chokepoint link (n2<-->n3)
$ns duplex-link $node_(2) $node_(3) 10Mb 10ms DropTail

# UDP access links
$ns duplex-link $node_(1) $node_(2) 100Mb 10ms DropTail
$ns duplex-link $node_(3) $node_(5) 100Mb 10ms DropTail

}

proc make_traffic {} {

global ns node_ trace_file trace_file1 trace_file2

# Get simulator instance
set ns [Simulator instance]

for { set i 1 } { $i <= 10 } { incr i } {

# Make UDP agent and attach to source
set udp_($i) [new Agent/UDP]
$udp_($i) set fid_ $i
$ns attach-agent $node_(1) $udp_($i)

# Create LossMonitor and attach to sink
set loss_($i) [new Agent/LossMonitor]
$ns attach-agent $node_(5) $loss_($i)

# Make CBR application
set cbr_($i) [new Application/Traffic/CBR]
$cbr_($i) set packetSize_ 200
$cbr_($i) set interval_ 0.010

# Connect source and sink
$ns connect $udp_($i) $loss_($i)

# Attach to UDP agent
$cbr_($i) attach-agent $udp_($i)

# Start it up
$ns at 0.0 "$cbr_($i) start"
$ns at 100.0 "$cbr_($i) stop"

```

```

# Create TCP flow
set tcp_($i) [new Agent/TCP/Sack1]
$tcp_($i) set fid_ $i
$tcp_($i) set class_ $i

set snk_($i) [new Agent/TCPSink/Sack1]
$ns attach-agent $node_(0) $tcp_($i)
$ns attach-agent $node_(4) $snk_($i)
$ns connect $tcp_($i) $snk_($i)

set ftp_($i) [new Application/FTP]
$ftp_($i) attach-agent $tcp_($i)
$ns at 0.0 "$ftp_($i) start"
$ns at 100.0 "$ftp_($i) stop"
}

set f [open $trace_file w]
set q [open $trace_file1 w]
set r [open $trace_file2 w]

$ns at 0.0 "$ns trace-queue $node_(3) $node_(5) $f"
$ns at 0.0 "$ns trace-queue $node_(3) $node_(4) $q"
$ns at 0.0 "$ns trace-queue $node_(2) $node_(3) $r"

}

```

ΠΑΡΑΡΤΗΜΑ Γ

Ενδεικτικός Κώδικας Προσομοιώσεων για RTP

```
proc init {BuffSize} {

    global ns SimTime packetSize window queueType trace_file trace_file1 trace_file2

    set packetSize 1000
    set SimTime 100.0

    set old_data 0

    set ns [new Simulator]

    Agent/TCP set useHeaders_ false

    # ecn capability
    Agent/TCP set ecn_ 1

    Agent/TCP/Newreno set packetSize_ $packetSize

    Session/RTP set debug_ FALSE

    Queue set limit_ $BuffSize
    $ns trace-all [open all_traces.tr w]
}

proc topo {} {

    global ns node_ queueType

    # Basic topology
    #
    # [node_(0)]---+      +---[node_(4)]
    #              |      |
    # [node_(2)]-----[node_(3)]
    #              |      |
    # [node_(1)]---+      +---[node_(5)]
    #              |---rtp flows----|

    # Create nodes
    set node_(0) [$ns node]
    set node_(1) [$ns node]
    set node_(2) [$ns node]
```

```

set node_(3) [$ns node]
set node_(4) [$ns node]
set node_(5) [$ns node]

$ns duplex-link $node_(0) $node_(2) 100Mb 10ms DropTail
$ns duplex-link $node_(3) $node_(4) 100Mb 10ms DropTail

# The chokepoint link (n2<-->n3)
$ns duplex-link $node_(2) $node_(3) 10Mb 10ms DropTail

# RTP access links
$ns duplex-link $node_(1) $node_(2) 100Mb 10ms DropTail
$ns duplex-link $node_(3) $node_(5) 100Mb 10ms DropTail

}

proc make_traffic {} {

    global ns node_ trace_file trace_file1 trace_file2

    # Get simulator instance
    set ns [Simulator instance]

    for { set i 1 } { $i <= 10 } { incr i } {

        # Make RTP agent and attach to source
        set rtp_($i) [new Agent/RTP]
        set rtpr_($i) [new Agent/RTCP]
        set ses_($i) [new Session/RTP]

        $ns attach-agent $node_(1) $rtp_($i)
        $ns attach-agent $node_(5) $rtpr_($i)

        $ns connect $rtp_($i) $rtpr_($i)

        $rtp_($i) session $ses_($i)
        $rtpr_($i) session $ses_($i)

        $rtp_($i) set fid_ $i

        # Make CBR application
        set cbr_($i) [new Application/Traffic/CBR]
        $cbr_($i) set packetSize_ 200
        $cbr_($i) set interval_ 0.010

        # Attach to RTP agent
        $cbr_($i) attach-agent $rtp_($i)

```

```

$rtpr_($i) set interval_ 900ms

# Start it up
$ns at 0.0 "$cbr_($i) start"
$ns at 100.0 "$cbr_($i) stop"

$ns at 0.0 "$rtpr_($i) start"
$ns at 100.0 "$rtpr_($i) stop"

# Create TCP flow
set tcp_($i) [new Agent/TCP/Sack1]
$tcp_($i) set fid_ $i
$tcp_($i) set class_ $i

set snk_($i) [new Agent/TCPSink/Sack1]
$ns attach-agent $node_(0) $tcp_($i)
$ns attach-agent $node_(4) $snk_($i)
$ns connect $tcp_($i) $snk_($i)

set ftp_($i) [new Application/FTP]
$ftp_($i) attach-agent $tcp_($i)
$ns at 0.0 "$ftp_($i) start"
$ns at 100.0 "$ftp_($i) stop"

}

set f [open $trace_file w]
set q [open $trace_file1 w]
set r [open $trace_file2 w]

$ns at 0.0 "$ns trace-queue $node_(3) $node_(5) $f"
$ns at 0.0 "$ns trace-queue $node_(3) $node_(4) $q"
$ns at 0.0 "$ns trace-queue $node_(2) $node_(3) $r"

}

```

ΠΑΡΑΡΤΗΜΑ Δ

Ενδεικτικός Κώδικας Προσομοιώσεων για TFRC

```
proc init {BuffSize} {  
  
    global ns SimTime packetSize window queueType trace_file trace_file1 trace_file2  
  
    set packetSize 1000  
    set SimTime 100.0  
  
    set old_data 0  
    set all_data 0  
  
    set ns [new Simulator]  
  
    Agent/TCP set useHeaders_ false  
  
    # ecn capability  
    Agent/TCP set ecn_ 1  
  
    Agent/TCP/Sack1 set packetSize_ $packetSize  
  
    Queue set limit_ $BuffSize  
    $ns trace-all [open all_traces.tr w]  
  
}  
  
proc topo {} {  
  
    global ns node_ queueType  
  
    # Basic topology  
    #  
    # [node_(0)]---+    +---[node_(4)]  
    #             |    |  
    # [node_(2)]-----[node_(3)]  
    #             |    |  
    # [node_(1)]---+    +---[node_(5)]  
    #             |---tfrc flows----|  
  
    # Create nodes  
    set node_(0) [$ns node]  
    set node_(1) [$ns node]  
    set node_(2) [$ns node]  
    set node_(3) [$ns node]  
    set node_(4) [$ns node]
```

```

set node_(5) [$ns node]

$ns duplex-link $node_(0) $node_(2) 100Mb 10ms DropTail
$ns duplex-link $node_(3) $node_(4) 100Mb 10ms DropTail

# The chokepoint link (n2<-->n3)
$ns duplex-link $node_(2) $node_(3) 10Mb 10ms DropTail

# TFRC access links
$ns duplex-link $node_(1) $node_(2) 100Mb 10ms DropTail
$ns duplex-link $node_(3) $node_(5) 100Mb 10ms DropTail

}

proc make_traffic {} {

global ns node_ trace_file trace_file1 trace_file2

# Get simulator instance
set ns [Simulator instance]

for { set i 1 } { $i <= 10 } { incr i } {

#Create TFRC flow
set tfrc_($i) [new Agent/TFRC]

$tfrc_($i) set SndrType_ 0
$tfrc_($i) set packetSize_ 200
$tfrc_($i) set maxqueue_ 2
$tfrc_($i) set fid_ $i
$tfrc_($i) set maxXmtRate_ 1000000
$tfrc_($i) set voip_ 1
$tfrc_($i) set voip_max_pkt_rate_ 100
$ns attach-agent $node_(1) $tfrc_($i)

set snk_($i) [new Agent/TFRCSink]

$ns attach-agent $node_(5) $snk_($i)
$ns connect $tfrc_($i) $snk_($i)

# Make CBR application
set cbr_($i) [new Application/Traffic/CBR]
$cbr_($i) set packetSize_ 200
$cbr_($i) set interval_ 0.010

# Attach to TFRC agent
$cbr_($i) attach-agent $tfrc_($i)

```

```

# Start it up
$ns at 0.0 "$cbr_($i) start"
$ns at 100.0 "$cbr_($i) stop"

$ns at 0.0 "$tfrc_($i) start"
$ns at 100.0 "$tfrc_($i) stop"

# Create TCP flow
set tcp_($i) [new Agent/TCP/Sack1]
$tcp_($i) set packetSize_ 1460
$tcp_($i) set fid_ $i
$tcp_($i) set class_ $i

set snk_($i) [new Agent/TCPSink/Sack1]
$ns attach-agent $node_(0) $tcp_($i)
$ns attach-agent $node_(4) $snk_($i)
$ns connect $tcp_($i) $snk_($i)

set ftp_($i) [new Application/FTP]
$ftp_($i) attach-agent $tcp_($i)
$ns at 0.0 "$ftp_($i) start"
$ns at 100.0 "$ftp_($i) stop"

}

set f [open $trace_file w]
set q [open $trace_file1 w]
set r [open $trace_file2 w]

$ns at 0.0 "$ns trace-queue $node_(1) $node_(2) $f"
$ns at 0.0 "$ns trace-queue $node_(3) $node_(4) $q"
$ns at 0.0 "$ns trace-queue $node_(2) $node_(3) $r"

}

proc trace_queue {} {
    global ns node_ tchan_ stats_file
    set q [[$ns link $node_(2) $node_(3)] queue]
    set tchan_ [open $stats_file w]
    $q trace curq_

    $q attach $tchan_
}

```

ΠΑΡΑΡΤΗΜΑ Ε

Ενδεικτικός Κώδικας Προσομοιώσεων για SCTP

```
proc init {BuffSize} {

    global ns SimTime packetSize window queueType trace_file1 trace_file2 trace_file3

    set packetSize 1000
    set SimTime 100.0

    set old_data 0
    set all_data 0
    set ns [new Simulator]

    Agent/TCP set useHeaders_ false

    # ecn capability
    Agent/TCP set ecn_ 1

    Agent/TCP/Newreno set packetSize_ $packetSize

    Queue set limit_ $BuffSize
    $ns trace-all [open all_traces.tr w]
}

proc topo {} {

    global ns node_ queueType

    # Basic topology
    #
    #          |---sctp flows---|
    # [node_( )]---+          +---[node_( )]
    #          |          |
    #   [router_(2)]-----[router_(3)]
    #          |          |
    # [node_(1)]---+          +---[node_(5)]
    #          |---tcp flows ---|

    # Create nodes
    set node_(1) [$ns node]
    set node_(router2) [$ns node]
    set node_(router3) [$ns node]
    set node_(5) [$ns node]
```

```

# The chokepoint link (n2<-->n3)
$ns duplex-link $node_(router2) $node_(router3) 10Mb 10ms DropTail

# TFRC access links
$ns duplex-link $node_(1) $node_(router2) 100Mb 10ms DropTail
$ns duplex-link $node_(router3) $node_(5) 100Mb 10ms DropTail

set node_(host0_core) [$ns node]
set node_(host0_if0) [$ns node]
set node_(host0_if1) [$ns node]

$ns multihome-add-interface $node_(host0_core) $node_(host0_if0)
$ns multihome-add-interface $node_(host0_core) $node_(host0_if1)

$ns duplex-link $node_(host0_if0) $node_(router2) 100Mb 10ms DropTail
$ns duplex-link $node_(host0_if1) $node_(router2) 100Mb 10ms DropTail

set node_(host1_core) [$ns node]
set node_(host1_if0) [$ns node]
set node_(host1_if1) [$ns node]

$ns multihome-add-interface $node_(host1_core) $node_(host1_if0)
$ns multihome-add-interface $node_(host1_core) $node_(host1_if1)

$ns duplex-link $node_(host1_if0) $node_(router3) 100Mb 10ms DropTail
$ns duplex-link $node_(host1_if1) $node_(router3) 100Mb 10ms DropTail

}

proc make_traffic {} {

    global ns node_ trace_file1 trace_file2 trace_file3

    # Get simulator instance
    set ns [Simulator instance]

    for { set i 1 } { $i <= 10 } { incr i } {

        # Create TCP flow
        set tcp_($i) [new Agent/TCP/Sack1]
        $tcp_($i) set packetSize_ 1460
        $tcp_($i) set fid_ $i
        $tcp_($i) set class_ $i

        set snk_($i) [new Agent/TCPSink/Sack1]
        $ns attach-agent $node_(1) $tcp_($i)
        $ns attach-agent $node_(5) $snk_($i)
        $ns connect $tcp_($i) $snk_($i)
    }
}

```

```

set ftp_($i) [new Application/FTP]
$ftp_($i) attach-agent $tcp_($i)
$ns at 0.0 "$ftp_($i) start"
$ns at 100.0 "$ftp_($i) stop"

set sctp0_($i) [new Agent/SCTP]
$ns multihome-attach-agent $node_(host0_core) $sctp0_($i)
$sctp0_($i) set fid_ $i
$sctp0_($i) set debugMask_ -1
$sctp0_($i) set debugFileIndex_ 0
$sctp0_($i) set mtu_ 1500
$sctp0_($i) set dataChunkSize_ 200
$sctp0_($i) set numOutStreams_ 5
$sctp0_($i) set oneHeartbeatTimer_ 60 # each dest has its own heartbeat timer
$sctp0_($i) set pathMaxRetrans_ 0
$sctp0_($i) set maxInitRetrans_ 0
$sctp0_($i) set numUnrelStreams_ 5

set trace_ch [open trace.sctp w]
$sctp0_($i) set trace_all_ 1 # trace them all on oneline
$sctp0_($i) trace cwnd_
$sctp0_($i) trace rto_
$sctp0_($i) trace errorCount_
$sctp0_($i) attach $trace_ch

set sctp1_($i) [new Agent/SCTP]
$ns multihome-attach-agent $node_(host1_core) $sctp1_($i)
$sctp1_($i) set debugMask_ -1
$sctp1_($i) set debugFileIndex_ 1
$sctp1_($i) set mtu_ 1500
$sctp1_($i) set initialRwnd_ 131072
$sctp1_($i) set useDelayedSacks_ 0

$ns connect $sctp0_($i) $sctp1_($i)
$sctp0_($i) set-primary-destination $node_(host1_if0)

# Make CBR application
set cbr0_($i) [new Application/Traffic/CBR]
$cbr0_($i) set packetSize_ 200
$cbr0_($i) set interval_ 0.010

# Attach to TFRC agent
$cbr0_($i) attach-agent $sctp0_($i)

# Start it up
$ns at 0.0 "$cbr0_($i) start"
$ns at 100.0 "$cbr0_($i) stop"

}

```

```
set f [open $trace_file1 w]
set q [open $trace_file2 w]
set r [open $trace_file3 w]

$ns at 0.0 "$ns trace-queue $node_(router3) $node_(5) $f"
$ns at 0.0 "$ns trace-queue $node_(router3) $node_(host1_if0) $q"
$ns at 0.0 "$ns trace-queue $node_(router2) $node_(router3) $r"

}
```

ΠΑΡΑΡΤΗΜΑ ΣΤ

Κώδικας λήψης εύρους ζώνης και χαμένων πακέτων

```
set old_data 0

#####TOTAL THROUGHPUT#####

exec awk {
    {
        if ( $1=="r" && $5=="tcp" ) {
            old_data = old_data + $6
            print $2, old_data*8.0/$2/1000000
        }
    }
} out-tcp.tr > tcp-throughput.data

exec awk {
    {
        if ($1=="r" && $5=="tcpFriend" ) {
            old_data = old_data + $6
            print $2, old_data*8.0/$2/1000000
        }
    }
} out-tfrc.tr > tfrc-throughput.data

set old_data 0
set all_data 0

#####TOTAL DROPS#####

exec awk {
    {

        if ( $1 == "+" && $5 == "tcp")
        {
            all_data = all_data + 1
        }

        if ( $1=="d" && $5=="tcp") {
            old_data = old_data + 1
            print $2, (old_data *100.0)/all_data
        }
    }
} bottleneck.tr > tcp-drops.data
```

```
exec awk {  
    {  
        if ( $1 == "+" && $5 == "tcpFriend")  
        {  
            all_data = all_data + 1  
        }  
  
        if ($1=="d" && $5=="tcpFriend") {  
            old_data = old_data + 1  
  
            print $2, (old_data *100.0)/all_data  
        }  
    }  
} bottleneck.tr > tfrc-drops.data
```