

Σχολή Διοίκησης & Οικονομίας  
Τμήμα Τηλεπληροφορικής & Διοίκησης  
ΤΕΙ ΗΠΕΙΡΟΥ

School Of Management And Economics  
Department Of Communications,  
Informatics And Management



# **ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

**ΘΕΜΑ**

**UDT (UDP –BASED TRANSFER ROTOCOL)**

ΟΝΟΜΑ ΣΠΟΥΔΑΣΤΗ :ΛΑΜΠΡΟΠΟΥΛΟΣ ΓΕΡΑΣΙΜΟΣ  
ΕΙΣΗΓΗΤΗΣ :ΤΣΙΑΝΤΗΣ ΛΕΩΝΙΔΑΣ

**Τ.Ε.Ι. ΗΠΕΙΡΟΥ  
ΦΕΒΡΟΥΑΡΙΟΣ 2006**

## ΠΕΡΙΕΧΟΜΕΝΑ

ΕΙΣΑΓΩΓΗ – ΠΡΩΤΟΚΟΛΛΑ.....	3
ΚΑΘΥΣΤΕΡΗΣΗ ΜΕΤΑΦΟΡΑ.....	7
ΜΕΤΑΒΛΗΤΟΤΗΤΑ ΤΗΣ ΚΑΘΥΣΤΕΡΗΣΗΣ.....	8
ΙΣΟΧΡΟΝΙΣΜΟΣ.....	9
ΑΠΑΙΤΗΣΕΙΣ ΑΠΟ ΜΙΑ ΕΦΑΡΜΟΓΗ ΠΟΛΥΜΕΣΩΝ.....	10
ΠΕΡΙΛΗΨΗ.....	12
ΕΙΣΑΓΩΓΗ.....	13
ΑΙΤΙΟΛΟΓΙΑ ΣΧΕΔΙΑΣΗΣ.....	14
UDT ΠΡΩΤΟΚΟΛΛΟ.....	16
ΔΙΑΤΑΞΗ ΠΑΚΕΤΟΥ.....	18
ΑΛΓΟΡΙΘΜΟΣ ΑΠΟΣΤΟΛΗΣ ΚΑΙ ΛΗΨΗΣ.....	19
ΕΠΕΞΕΡΓΑΣΙΑ ΕΛΕΓΧΟΥ ΠΛΗΡΟΦΟΡΙΑΣ.....	22
ΕΛΕΓΧΟΣ ΣΥΜΦΟΡΗΣΗΣ.....	24
ΕΛΕΓΧΟΣ ΡΥΘΜΟΥ.....	25
ΑΝΑΛΥΣΗ ΚΑΙ ΑΠΟΤΙΜΗΣΗ.....	27
ΔΙΚΑΙΟΣΥΝΗ.....	29
ΠΕΡΙΛΗΨΗ.....	44
ΠΗΓΕΣ.....	82

## **Εισαγωγή – Γενικά**

### 1. Τα πρωτόκολλα μεταφοράς (Transport protocols)

Το πρωτόκολλο μεταφοράς είναι απαραίτητο για τον έλεγχο και την διόρθωση σφαλμάτων μετάδοσης μεταξύ των ακραίων χρηστών. Επιπλέον εκτελεί και άλλες χρήσιμες λειτουργίες όπως να μεταφέρει πληροφορίες για την εφαρμογή που κάνει χρήση της σύνδεσης ή να βοηθά τη ταυτόχρονη χρήση μιας σύνδεσης από πολλές εφαρμογές με μεθόδους πολύπλεξης.

Το πακέτο του επιπέδου μεταφοράς (συμπεριλαμβανομένων των δεδομένων και της επικεφαλίδας του) τοποθετείται μέσα σε ένα πακέτο IP για την μετάδοση ,δηλαδή αποτελεί το τμήμα δεδομένων του πακέτου IP.

Υπάρχουν δυο κύρια πρωτόκολλα μεταφοράς στο Internet ,το UDP και το TCP.

### 2. UDP (user datagram protocol)

Το UDP είναι εξαιρετικά απλό πρωτόκολλο που δεν εξασφαλίζει ασφαλή μετάδοση και δεν ενημερώνει τον αποστολέα για την επιτυχή ή ανεπιτυχή παράδοση των δεδομένων.

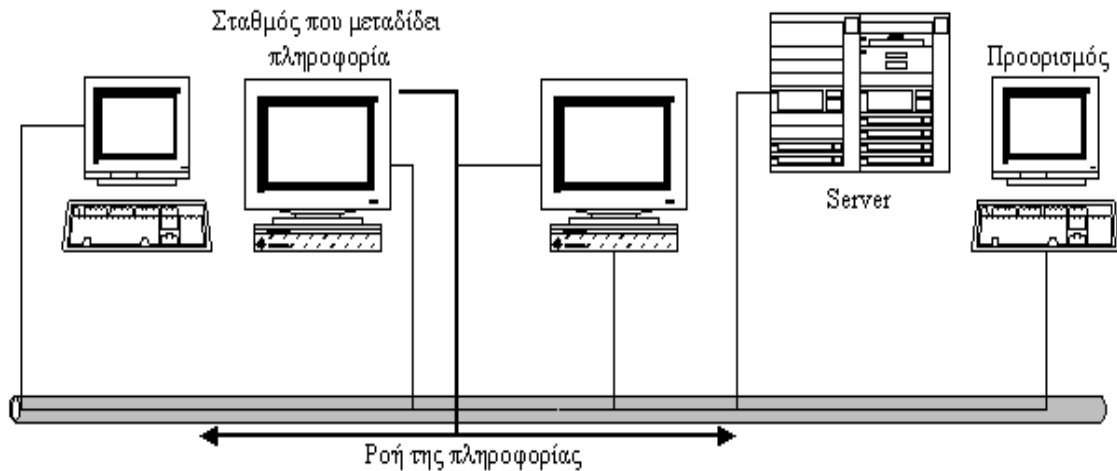
Το πακέτο UDP μεταφέρει δεδομένα , μεταφέρει πληροφορίες για την εφαρμογή που το χρησιμοποιεί και έχει ένα checksum ώστε ο δέκτης να αντιληφθεί εάν τα δεδομένα έχουν καταστραφεί από τυχόν σφάλματα μετάδοσης στο δίκτυο.

### 3. TCP-Transmission Control Protocol

Σε αντίθεση με το UDP το TCP έχει σχεδιασθεί να αντιμετωπίζει όλα τα προβλήματα που μπορεί να δημιουργήσει το δίκτυο στην μετάδοση εξασφαλίζοντας την σωστή μεταφορά τους. Εάν πακέτα χαθούν ή αλλοιωθούν το TCP θα φροντίσει για την επαναποστολή τους . Εάν κάποια πακέτα φθάσουν με λανθασμένη σειρά το TCP θα τα ανακατατάξει με τον σωστό τρόπο. Εάν υπάρχουν επανειλημμένες απώλειες πακέτων στην σύνδεση τότε το TCP υποθέτει ότι υπάρχει συμφόρηση στο δίκτυο και αναλόγως ρυθμίζει το ρυθμό μετάδοσης.

### 4. Τοπολογίες δικτύου

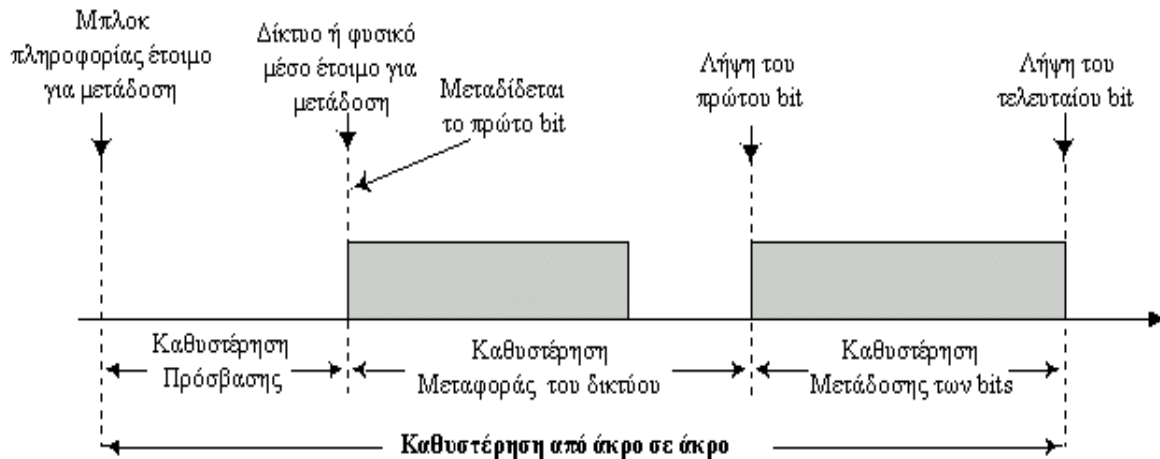
Δεδομένου ενός συνόλου υπολογιστών που θέλουμε να επικοινωνούν μεταξύ τους, υπάρχουν πολλοί τρόποι να οργανωθούν στα συστήματα μετάδοσης, όπως τα modems και τα καλώδια ή η μπάντα συχνοτήτων στην περίπτωση της εναέριας μετάδοσης. Η οργάνωση αυτή ονομάζεται τοπολογία του δικτύου.



Σχήμα 7-1. Η τοπολογία αρτηρίας

Η πιο απλή αλλά και υπερβολικά δαπανηρή μέθοδος, είναι η σύνδεση όλων με όλους, η οποία όμως δεν χρησιμοποιείται στην πράξη. Μια άλλη μέθοδος είναι η τοπολογία αστέρα (star topology). Υπάρχει ένα κόμβος που ονομάζεται διακόπτης (switch), ο οποίος συνδέεται απευθείας με όλους τους άλλους υπολογιστές. Έτσι, για να επικοινωνήσουν δύο υπολογιστές, πρέπει απαραίτητα να χρησιμοποιήσουν τον διακόπτη. Ο διακόπτης είναι επιφορτισμένος με τη λειτουργία της προώθησης των εισερχόμενων δεδομένων στο προορισμό τους. Αν συνενωθούν πολλές τοπολογίες αστέρα, προκύπτει ένα δένδρο, το οποίο περιέχει περισσότερους του ενός διακόπτες. Το επόμενο στάδιο πολυπλοκότητας είναι η κατανεμημένη τοπολογία (mesh topology), η οποία παίρνει ένα σύνολο τοπολογιών αστέρα και προσθέτει πλεονάζουσες συνδέσεις μεταξύ των διακοπών. Δηλαδή δημιουργούνται εναλλακτικοί δρόμοι επικοινωνίας μεταξύ των υπολογιστών. Τα τηλεφωνικά δίκτυα είναι μίξη της τοπολογίας αστέρα και της κατανεμημένης.

Δυο απλές και ευρέως χρησιμοποιούμενες τοπολογίες είναι η αρτηρία (bus) και ο δακτύλιος (ring). Η αρτηρία αποτελείται από ένα μόνο καλώδιο, πάνω στο οποίο είναι συνδεδεμένοι όλοι οι υπολογιστές. Όταν ένας υπολογιστής μεταδίδει, το σήμα διαδίδεται σε όλο το μήκος της αρτηρίας και προς τις δυο διευθύνσεις, έτσι ώστε όλοι οι άλλοι υπολογιστές να μπορούν να το λάβουν. Ο δακτύλιος είναι παρόμοιος με την αρτηρία, με τη διαφορά ότι το καλώδιο κλείνει δημιουργώντας ένα δακτύλιο, στον οποίο το σήμα διαδίδεται προς μια κατεύθυνση μόνο.



Σχήμα 7-2. Η τοπολογία δακτυλίου

### 5. Πρωτόκολλα και Πρότυπα Επικοινωνίας

Για να έχει νόημα κάθε είδους επικοινωνία, πρέπει να υπάρχουν κάποιοι κοινοί κανόνες και συμβάσεις μεταξύ των επικοινωνούντων μερών. Στις τηλεπικοινωνίες, ως πρωτόκολλο ορίζεται ένα σύνολο από κανόνες που διέπουν την επικοινωνία δύο συστημάτων. Όταν ένα πρωτόκολλο υποστηρίζεται από κάποιο διεθνή οργανισμό προτυποποίησης, ονομάζεται απλά πρότυπο. Υπάρχουν και πρωτόκολλα που ακολουθούνται από συγκεκριμένες εταιρείες. Όταν αυτά τα πρωτόκολλα χρησιμοποιούνται ευρέως, τότε μιλάμε για ένα de facto πρότυπο.

Συνήθως, τα πρότυπα είναι οργανωμένα σε επίπεδα (layers). Κάθε επίπεδο αντιστοιχεί σε κάποιο στάδιο της επικοινωνίας και χρησιμοποιεί όσα βρίσκονται κάτω από αυτό. Η επικοινωνία γίνεται μόνο μεταξύ επιπέδων ίδιου βάθους. Το πρότυπο OSI του ISO ορίζει μια δομή 7 επιπέδων: φυσικό, σύνδεσης δεδομένων, δικτύου, μεταφοράς, συνόδου, παρουσίασης και εφαρμογής.

### 6. Μεταγωγή Κυκλώματος και Με Πακέτα

Η μεταγωγή είναι η διαδικασία η οποία εξασφαλίζει ότι τα δεδομένα θα φτάσουν στον σωστό προορισμό τους. Υπάρχουν δυο τρόποι μεταγωγής. Ο πρώτος που χρησιμοποιήθηκε είναι η διαμεταγωγή κυκλώματος. Το χαρακτηριστικό αυτής της μεθόδου είναι η εξασφάλιση ενός πλήρους φυσικού κυκλώματος, για την επικοινωνία δύο οποιονδήποτε συστημάτων. Αυτό το κύκλωμα δεν μπορεί να χρησιμοποιηθεί από κανέναν άλλον, ακόμα και όταν δεν λαμβάνει χώρα ανταλλαγή μηνυμάτων. Σε πιο σύγχρονα συστήματα, αυτό που εξασφαλίζεται δεν είναι ένα φυσικό αλλά ένα νοητό κύκλωμα (virtual circuit). Αυτό σημαίνει ότι σε σταθερά χρονικά διαστήματα, το υπάρχον φυσικό κύκλωμα θα διατίθεται οπωσδήποτε για την επικοινωνία των δύο συστημάτων. Αντίθετα με το φυσικό, το νοητό κύκλωμα μοιράζεται

με άλλους, οπότε στην ουσία αυτό που εξασφαλίζεται είναι ένα σταθερό bit rate για την επικοινωνία. Αυτό είναι και το πλεονέκτημα αυτής της μεθόδου. Μειονεκτεί όμως στο βαθμό χρησιμοποίησης του συστήματος, γιατί το νοητό κύκλωμα παραμένει δεσμευμένο ακόμα και όταν τα συστήματα που επικοινωνούν δεν το εκμεταλλεύονται πλήρως.

Η μεταγωγή πακέτου στοχεύει στην πιο αποτελεσματική πολύπλεξη, ώστε να μεγιστοποιείται η χρησιμοποίηση του συστήματος. Η πληροφορία δεν μεταδίδεται συνεχώς αλλά σε πακέτα σταθερού μήκους. Πολλές πηγές μπορούν να στείλουν πακέτα στο δίκτυο, τα οποία θα δρομολογηθούν με τέτοιο τρόπο ώστε όταν μια πηγή δεν μεταδίδει να διατίθενται οι πόροι του συστήματος σε άλλη. Σε γενικές γραμμές, η συμπεριφορά ενός τέτοιου δικτύου είναι στατιστική. Δηλαδή οι καθυστερήσεις δεν μπορούν να προβλεφθούν ακριβώς, αλλά μόνο κατά μέσο όρο.

#### 7. Επικοινωνία με Σύνδεση ή Χωρίς Σύνδεση

Σε ένα δίκτυο μεταγωγής με πακέτα, η επικοινωνία μπορεί να έχει δύο μορφές: με σύνδεση (connection oriented) ή χωρίς σύνδεση (connectionless).

Όταν η επικοινωνία γίνεται με σύνδεση, πριν αρχίσει η ανταλλαγή δεδομένων το δίκτυο ενημερώνεται και εγκαθίσταται ένα κανάλι επικοινωνίας. Στη συνέχεια η ροή των δεδομένων μπορεί να είναι συνεχής και το δίκτυο φροντίζει για τη σωστή αποστολή των πακέτων και πιθανώς και για την ελάχιστη ταχύτητα.

Αντίθετα, όταν η επικοινωνία γίνεται χωρίς σύνδεση, η ανταλλαγή των μηνυμάτων γίνεται χωρίς έλεγχο από το δίκτυο. Το δίκτυο απλώς αποστέλλει ανεξάρτητα πακέτα, χωρίς να ξέρει ότι ποια αποτελούν μέρος της ίδιας ροής δεδομένων προς έναν κόμβο. Αυτός ο τρόπος είναι ταχύτερος, όταν δεν πρόκειται να αποσταλούν πολλά δεδομένα, αλλά πάσχει από ασφάλεια.

#### 8. Μέτρα Αξιολόγησης Ενός Δικτύου

Η αξιολόγηση μιας αρχιτεκτονικής δικτύου είναι πολύπλοκη υπόθεση και απαιτεί την εξέταση πολλών παραμέτρων. Όσον αφορά στην ικανότητα ενός δικτύου να υποστηρίξει εφαρμογές πολυμέσων, μπορούμε να διακρίνουμε έξι παράγοντες καθοριστικής σημασίας:

- Ρυθμός Εξυπηρέτησης (Throughput)
- Καθυστέρηση Μεταφοράς (Transit Delay)
- Μεταβλητότητα της Καθυστέρησης (Delay Variation)
- Ισοχρονισμός (Isochronism)
- Multicasting
- Ρυθμοί Λαθών (Error Rates)

### *Ρυθμός Εξυπηρέτησης (Throughput)*

Το δείκτη αυτό τον έχουμε ήδη χρησιμοποιήσει με τα ονόματα bit rate, ρυθμό μεταφοράς δεδομένων (transfer rate) ή εύρος ζώνης (bandwidth). Ο τελευταίος όρος τυπικά αναφέρεται στο εύρος συχνοτήτων ενός μέσου μετάδοσης, αλλά γενικεύεται κατά αναλογία και στην περίπτωση του δικτύου. Ο ρυθμός εξυπηρέτησης μπορεί να οριστεί ως εξής:

Ο ρυθμός μεταφοράς των δεδομένων μεταξύ δύο συστημάτων ορίζεται ως το πλήθος των δυαδικών ψηφίων (ή πακέτων) που μπορεί να δεχτεί και μεταδώσει το δίκτυο στη μονάδα του χρόνου.

Ο ορισμός αυτό έχει ένα κρυφό σημείο. Δεν καθορίζει ακριβώς τον τρόπο μέτρησης του ρυθμού εξυπηρέτησης. Έτσι μια τιμή μπορεί να αναφέρεται στο μέγιστο ρυθμό εξυπηρέτησης είτε στο ρυθμό εξυπηρέτησης που μπορεί να διατηρηθεί σταθερός από το δίκτυο.

Οι συνήθεις μονάδες μέτρησης είναι τα πολλαπλάσια του bps (bits per second): Kbps, Mbps, Gbps. Σε δίκτυα όπου η πληροφορία μεταδίδεται σε πακέτα, μπορούμε να μιλήσουμε για packets/sec.

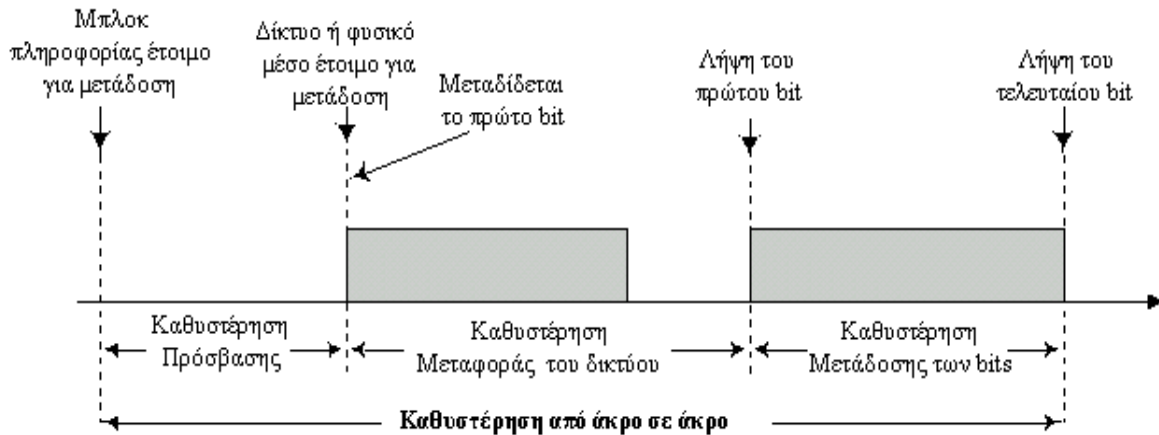
Στο ορισμό παρατηρούμε μια διαφοροποίηση μεταξύ του μέγιστου δυνατού ρυθμού αποδοχής των δεδομένων, που θα ονομάσουμε ρυθμό ή ταχύτητα πρόσβασης (access speed), και του ρυθμού μετάδοσης τους από το δίκτυο. Πράγματι, υπάρχουν δίκτυα, όπως τα περισσότερα από αυτά που χρησιμοποιούν διαμεταγωγή με πακέτα, που δέχονται δεδομένα τα οποία όμως, για διάφορους λόγους, δεν μπορούν να μεταδοθούν αμέσως και τοποθετούνται σε ουρές αναμονής. Αντίθετα, τα δίκτυα μεταγωγής κυκλώματος μπορούν να εξασφαλίσουν σταθερό bit rate παρόμοιο με αυτό του ρυθμού εισόδου πελατών.

### *Καθυστέρηση Μεταφοράς*

Ορίζουμε την καθυστέρηση μεταφοράς ως εξής:

Η καθυστέρηση μεταφοράς του δικτύου είναι το χρονικό διάστημα μεταξύ της αποστολής του πρώτου bit ενός κομματιού πληροφορίας και της λήψης του από το άλλο άκρο της επικοινωνίας.

Κανένα δίκτυο δεν μπορεί να αποφύγει την καθυστέρηση μεταφοράς λόγω της καθυστέρησης μετάδοσης του σήματος στο φυσικό μέσο. Υπάρχουν και περιπτώσεις δικτύου που αυτή η καθυστέρηση οφείλεται και σε άλλους παράγοντες όπως η δρομολόγηση και η αναγέννηση.



Σχήμα 7-3. Καθυστερήσεις κατά τη μετάδοση της πληροφορίας σε ένα δίκτυο

**Η καθυστέρηση μεταφοράς** αποτελεί ένα χαρακτηριστικό του δικτύου. Για τις περισσότερες εφαρμογές υπάρχει μια πιο σημαντική παράμετρος: η καθυστέρηση από άκρο σε άκρο, η οποία έχει τρεις συνιστώσες:

- Το χρόνο που απαιτείται για να ελευθερωθεί το μέσο, ώστε να επιτραπεί η αποστολή των δεδομένων από το δίκτυο. Αυτή η καθυστέρηση ονομάζεται καθυστέρηση πρόσβασης (access delay)
- Το χρόνο διάδοσης των δεδομένων πάνω στο φυσικό μέσο.
- Την καθυστέρηση μεταφοράς που ορίσαμε πριν.

Για τις interactive εφαρμογές ιδιαίτερη σημασία έχει και ο χρόνος απάντησης από τον λήπτη (round trip delay). Ο χρόνος αυτός δεν εξαρτάται πλήρως από το δίκτυο αλλά και από την ταχύτητα με την οποία απαντά ο λήπτης.

#### *Μεταβλητότητα της Καθυστέρησης*

Κανένα δίκτυο δεν μπορεί να εγγυηθεί σταθερή καθυστέρηση μεταφοράς ή από άκρο σε άκρο. Υπάρχουν δίκτυα με ελάχιστες καθυστερήσεις της τάξης του nanosecond στα οποία η μεταβλητότητα δεν παίζει καθοριστικό ρόλο. Όταν όμως αυξάνει η καθυστέρηση και η μεταβλητότητα είναι μεγάλη, όπως στα δίκτυα IP (Internet Protocol), τότε η παράμετρος αυτή είναι σημαντική. Η μεταβλητότητα μετράται με διάφορες στατιστικές μεθόδους.

Στη τεχνολογία μετάδοσης σημάτων ορίζεται η έννοια του jitter, ως η μεταβλητότητα της καθυστέρησης μετάδοσης που οφείλεται αποκλειστικά στις συσκευές μετάδοσης. Στα δίκτυα το jitter που οφείλεται στις ατέλειες των συσκευών μετάδοσης είναι αναπόφευκτο, αλλά συνήθως μικρό. Σε κυκλώματα μεγάλων αποστάσεων μπορεί να



φτάσει την τάξη των microsecond, ενώ συνήθως κυμαίνεται στην τάξη των nanosecond.

Εκτός από το jitter του υλικού, υπάρχει και μεταβλητότητα που οφείλεται στην αρχιτεκτονική του δικτύου. Για παράδειγμα, σε τοπικά δίκτυα αρτηρίας η μεταβλητότητα του χρόνου πρόσβασης ή σε δίκτυα IP της δρομολόγησης, προστίθενται σε αυτή του jitter.

### *Ισοχρονισμός*

Αυτό το χαρακτηριστικό έχει ιδιαίτερη σημασία, όσον αφορά στην καταλληλότητα ενός δικτύου για εφαρμογές πολυμέσων. Δεν αποτελεί εγγενές χαρακτηριστικό του δικτύου, αλλά έναν συνδυασμό ορισμένων βασικών χαρακτηριστικών.

Μια από άκρο σε άκρο επικοινωνία ονομάζεται ισόχρονη, εάν το bit rate της σύνδεσης είναι εξασφαλισμένο και αν η μεταβλητότητα της καθυστέρησης είναι επίσης εξασφαλισμένη και μικρή.

Αυτή η απαίτηση επιτρέπει την μετάδοση συνεχών ροών πληροφορίας, όπως για παράδειγμα video και ήχου πραγματικού χρόνου. Τέτοιου είδους μεταδόσεις απαιτούν ένα σταθερό ρυθμό μεταφοράς δεδομένων, ώστε η πληροφορία να διατηρεί τη χρονική της εξάρτηση στο άλλο άκρο αναλλοίωτη. Επίσης, σταθερή μεταβλητότητα, που βρίσκεται σε καθορισμένα όρια, μπορεί να αντιμετωπιστεί ή να περάσει απαρατήρητη.

### *Multicasting*

Ο ορισμός του multicasting είναι ο εξής:

Multicasting είναι η ιδιότητα ενός δικτύου να αντιγράφει, σε καθορισμένα σημεία του δικτύου, τα δεδομένα που εκπέμπει μια πηγή. Τα δεδομένα που αντιγράφονται προωθούνται στα συστήματα-παραλήπτες που αποτελούν μέλη ενός multicast group, με τέτοιο τρόπο ώστε να ελαχιστοποιηθούν τα τμήματα του δικτύου, στα οποία περνούν πολλά αντίγραφα της ίδιας πληροφορίας.

Η αντιγραφή μπορεί να γίνεται σε επίπεδο μεμονωμένων bits, μπλοκ πληροφορίας όπως τα πακέτα ή και σε επίπεδο αντικειμένων όπως έγγραφα, ηλεκτρονικά μηνύματα κ.λ.π.

### *Ρυθμοί Λαθών*

Το πιο προφανές ζητούμενο από ένα δίκτυο είναι η σωστή μετάδοση της πληροφορίας. Τα είδη των λαθών μπορούν να προκύψουν κατά τη μετάδοση της πληροφορίας μέσα από ένα δίκτυο είναι:

- Αλλοίωση των δεδομένων. Συνήθως εμφανίζεται με τη μορφή αντεστραμμένων bits.
- Χάσιμο δεδομένων. Αυτό μπορεί να οφείλεται στην αλλοίωση των δεδομένων. Ορισμένα δίκτυα ανιχνεύουν τα λάθη και απορρίπτουν τα μπλοκ πληροφορίας που έχουν επηρεαστεί. Στην

συνέχεια, είτε ενημερώνουν τον αποστολέα να επαναλάβει την αποστολή, είτε αφήνουν την εφαρμογή να αντιμετωπίσει την απώλεια. Σε σύγχρονα δίκτυα μεταγωγής με πακέτα, όπως τα IP, η απώλεια μπορεί να οφείλεται και στην υπερφόρτωση των κόμβων ή των γραμμών.

- Data Duplication. Ένα λάθος που συναντάται σπάνια, είναι η λήψη του ίδιου μπλοκ πληροφορίας περισσότερες από μια φορές.
- Λήψη σε λάθος σειρά. Σε δίκτυα που μεταφέρουν την πληροφορία σε κάποια μορφή πακέτα και προσφέρουν εναλλακτικούς δρόμους μετακίνησης των δεδομένων, είναι δυνατόν τα δεδομένα να φτάσουν στον προορισμό τους με λανθασμένη σειρά. Αυτό συμβαίνει συνήθως σε δίκτυα που εφαρμόζουν επικοινωνία χωρίς σύνδεση.

#### *Απαιτήσεις Από μια Εφαρμογή Πολυμέσων*

Ακόμα και αν λάβουμε υπ' όψη μας τις προόδους στην συμπίεση της πληροφορίας, οι απαιτήσεις που έχει η μετάδοση video και ήχου σε πραγματικό χρόνο, δυσχεραίνει σημαντικά την υλοποίηση πολλών εφαρμογών πολυμέσων. Παρ' όλο που υπάρχουν φυσικά μέσα, όπως οι οπτικές ίνες, που μπορούν να προσφέρουν μεγάλο εύρος ζώνης, και η τεχνολογία μεταγωγής (switching technology) και επεξεργασίας πρέπει να ακολουθήσει.

Οι εφαρμογές πολυμέσων έχουν το ιδιαίτερο χαρακτηριστικό ότι απαιτούν διάφορα είδη ροών πληροφορίας. Όταν μεταδίδεται video ή ήχος ή γενικότερα πληροφορία που εξαρτάται από το χρόνο, αυτό που παίζει σημαντικό ρόλο, είναι η ικανότητα του δικτύου να διατηρεί ένα σταθερό, και συνήθως αρκετά μεγάλο, ρυθμό μεταφοράς δεδομένων. Ο συγχρονισμός της πληροφορίας στο δέκτη μπορεί να γίνει μόνο αν το δίκτυο έχει μικρές καθυστερήσεις μεταφοράς με ανεκτή μεταβλητότητα. Αυτού του είδους η ροή ονομάζεται ισόχρονη (isochronous) και σχετίζεται με τα χαρακτηριστικά του δικτύου που περιγράψαμε στην προηγούμενη ενότητα.

Από την άλλη πλευρά, πολλές interactive εφαρμογές έχουν την τάση να μεταδίδουν πληροφορία με εκρήξεις (bursts). Αυτό σημαίνει ότι υπάρχουν μεγάλα διαστήματα με μικρή ή και καθόλου επικοινωνία, που διακόπτονται από μεταδόσεις μεγάλων όγκων πληροφορίας. Αυτού του είδους η ροή, που ονομάζεται αλλιώς και ασύγχρονη (asynchronous), συναντάται συχνότερα στην πράξη.

Η παρούσα τεχνολογία παρουσιάζει εναλλακτικές λύσεις για κάθε είδος ξεχωριστά: τα δίκτυα μεταγωγής κυκλώματος προσφέρονται για ισόχρονη μεταφορά πληροφορίας ενώ τα δίκτυα μεταγωγής με πακέτα για ασύγχρονη. Οι εφαρμογές πολυμέσων απαιτούν δίκτυα που να προσφέρουν και τα δύο χαρακτηριστικά ταυτόχρονα, πράγμα που είναι δύσκολο. Επιπλέον κανένας χρήστης δεν είναι διατεθειμένος να

πληρώνει για παραπάνω bandwidth από αυτό που χρησιμοποιεί κάθε φορά. Επειδή οι ανάγκες του κάθε χρήστη μεταβάλλονται συνεχώς, ανάλογα με την εφαρμογή που χρησιμοποιεί, απαιτείται τεχνολογία που να μπορεί να προσφέρει κάθε στιγμή το ζητούμενο εύρος ζώνης με την ανάλογη χρέωση.

## **ΧΡΗΣΙΜΟΠΟΙΩΝΤΑΣ ΤΟ UDP ΓΙΑ ΑΞΙΟΠΙΣΤΗ ΜΕΤΑΦΟΡΑ ΔΕΔΟΜΕΝΩΝ ΠΑΝΩ ΣΕ ΥΨΗΛΟΥ ΕΥΡΟΥΣ – ΚΑΘΥΣΤΕΡΗΣΗΣ ΠΑΡΑΓΩΓΗ ΔΙΚΤΥΩΝ.**

### Περίληψη

Όσο το δικτυακό εύρος και καθυστέρηση αυξάνουν, τόσο το TCP γίνεται αναξιόπιστο. Εφαρμογές εντατικών(πολλών-αποκλειστικών) δεδομένων πάνω σε υψηλής ταχύτητας δίκτυα όπως τα υπολογιστικά δίκτυα ηλεκτρικής ισχύος(GRID) χρειάζονται νέο πρωτόκολλο μεταφοράς για να τα υποστηρίξει.

Εδώ περιγράφουμε ένα πρωτόκολλο μεταφοράς δεδομένων, γενικού σκοπού, υψηλής απόδοσης, σαν μια λύση σε «application level». Το πρωτόκολλο ονομάζεται UDT, ή UDP – based data transfer protocol, δουλεύει επιπλέον του UDP με αξιοπιστία και μηχανισμούς ελέγχου της συμφόρησης.

Το UDT χρησιμοποιεί και τη θετική βεβαίωση λήψης και αρνητική βεβαίωση λήψης για να εξασφαλίσει αξιόπιστα δεδομένα. Αναμιγνύει τον rate based(βασισμένο στην κατανομή) και τον window based(βασισμένο στην θυρίδα-παράθυρο) μηχανισμούς ελέγχου συμφόρησης για να εξασφαλίσει αποδοτικότητα και δικαιοσύνη, μαζί με δύο ιδιαίτερα δίκαιους στόχους του TCP, τη φιλικότητα και την ανεξάρτητη καθυστέρηση. Και τα δύο, αποτέλεσμα προσομοίωσης και υλοποίησης έχουν δείξει ότι το UDT ικανοποιεί αυτά τα αντικείμενα πολύ καλά.

Εδώ περιγράφουμε τις λεπτομέρειες του UDT πρωτοκόλλου με αποτελέσματα προσομοίωσης και υλοποίησης ,και ανάλυση.

## Εισαγωγή

Η γρήγορη αύξηση του δικτυακού εύρους έχει καταστήσει ικανή τη μεταφορά δεδομένων σε ταχύτητα των gigabits per second. Υπολογιστικά και δίκτυα δεδομένων ηλεκτρικής ισχύος(GRID) είναι ανάμεσα στην πρώτη γενιά υποδομών για να αξιοποιήσουμε τους άφθονους δικτυακούς πόρους. Εν τω μεταξύ, νέες οπτικές τεχνολογίες αναμένονται να παρέχουν πιο γρήγορους δικτυακούς συνδέσμους στο κοντινό μέλλον.

Ωστόσο, άνθρωποι που δουλεύουν σε εφαρμογές εντατικών δεδομένων πάνω σε υψηλής ταχύτητας δίκτυα αναφέρουν ότι έχουν έλλειψη από ένα αποτελεσματικό – αποδοτικό πρωτόκολλο μεταφοράς. Το ευρέως χρησιμοποιούμενο TCP στο internet έχει εκθέσει τις αδυναμίες του όσο το δικτυακό εύρος και η καθυστέρηση παραγωγής αυξάνει(BDP).

Ο μηχανισμός Window based έλεγχου συμφόρησης του TCP υποφέρει από το «slow start»(αργή εκκίνηση). Κατά τη διάρκεια της φάσης αποφυγής συμφόρησης, αυτό αυξάνει το παράθυρο-θυρίδα αποστολής περίπου 1 byte per round trip time(RTT). Αυτό θα πάρει αρκετό χρόνο για μια ροή TCP ώστε να καταλάβει ένα υψηλό BDP σύνδεσμο.

Επιπλέον, αυτή η αργή αύξηση δεν εξασφαλίζει ότι η ροή TCP τελικά θα καταλάβει το μεγαλύτερο δυνατό εύρος. Τυχαία(απρογραμματίστη) απώλεια του φυσικού συνδέσμου, εμποδίζει το TCP, από αύξηση και κάνει το παράθυρο-θυρίδα αποστολής να πέφτει στο μισό. Θεωρητική ανάλυση έχει δείξει ότι για να φτάσει την υψηλή ταχύτητα πάνω σε υψηλά BDP δίκτυα, το TCP απαιτεί ένα εξαιρετικά χαμηλό loss error rate του συνδέσμου, το οποίο είναι ακατόρθωτο για τη τωρινή τεχνολογία. Το πρόβλημα είναι χειρότερο σε ασύρματα δίκτυα όπως δορυφορικός σύνδεσμος, το οποίο έχει ένα από τα υψηλότερα error rate συνδέσμου σε σχέση με τα ενσύρματα δίκτυα.

Από τότε που το TCP αυξάνει το παράθυρο-θυρίδα- per RTT, ροές με μικρότερο RTT θα αυξάνονται γρηγορότερα και προκαλεί αδικία-ανισότητα μεταξύ των ροών με διαφορετικά RTT. Αυτή η αδικία μπορεί να οδηγήσει σε σοβαρή κατάπτωση απόδοσης σε μια κατανομημένη υπολογιστική εφαρμογή όπου περιλαμβάνει – περικλείει αμφότερα και τοπικούς κόμβους και απομακρυσμένους κόμβους δια μέσου του ωκεανού. Ένα παράδειγμα τέτοιου είδους εφαρμογής είναι το «streaming join»(γίνομαι μέλος σε κατανομή φοιτητών σε μαθήματα), όπου η απόδοση περιορίζεται από την πιο αργή ροή δεδομένων.

Ερευνητές των δικτύων βρήκαν κάποιες λύσεις, μαζί με βελτίωση του TCP, νέα πρωτόκολλα ελέγχου δικτύου όπως το XCP και application level λύσεις. Τροποποίηση του TCP περιορίζεται από τις απαιτήσεις συμβατότητας των επικοινωνιών με το standard TCP. Απ' την άλλη μεριά, αν και νέα πρωτόκολλα αλλαγής θέσης(switching), δρομολόγησης(routing) και μεταφοράς(transporation) σίγουρα θα

εμπλουτίσουν ή θα αντικαταστήσουν τα τωρινά δικτυακά πρωτόκολλα στο μέλλον, αυτό δεν πρόκειται να γίνει σύντομα λόγω του ότι το κόστος μιας πιθανής αναβάθμισης σε σύστημα και υποδομή.

Αυτό το υπόβαθρο μας υποκινεί να σχεδιάσουμε και να αναπτύξουμε κάποιας μικρής σπουδαιότητας υψηλής απόδοσης πρωτόκολλο μεταφοράς δεδομένων σε application level. Χρησιμοποιούμε το UDP με μηχανισμούς ελέγχου συμφόρησης και βεβαίωσης λήψης για να εκπληρώσουμε αυτούς τους στόχους.

## Αιτιολογία σχεδίασης

Ο γενικός στόχος του νέου πρωτοκόλλου είναι στο να παρέχει μία γενικού σκοπού υπηρεσία μεταφοράς δεδομένων όπου μπορούμε να αξιοποιήσουμε το εύρος αποδοτικότερα και δικαιότερα.

Οι απαιτήσεις ανέρχονται-σηκώνονται από την κατάσταση των τεράστιων ποσοτήτων από δεδομένα μεταφοράς πάνω σε υψηλής ταχύτητας δίκτυα, π.χ. κατανομημένη μετάλλευση δεδομένων. Ωστόσο, το νέο πρωτόκολλο δεν είναι αφοσιωμένο να παραγεμίζει δεδομένα. Μπορεί να μεταφέρει δεδομένα από 1 single byte ως και 100(εκατοντάδες) από gigabits. Η σημασιολογία της διεπαφής(interface) πρέπει να είναι η ίδια, ή τουλάχιστον παρόμοια όπως η γενική υποδοχή(socket) API. Με τον όρο «γενικού σκοπού» επίσης σημαίνει ότι το πρωτόκολλο δεν πρέπει να τροποποιεί(αλλάζει) την τωρινή δικτυακή υποδομή έτσι ώστε να μπορεί να αναπτυχθεί με το μικρότερο κόστος, επίσης πρέπει να υπακούει στους end – to – end κανόνες.

Είναι φυσικό να χρησιμοποιούμε το UDP με loss detection/retransmission(απώλεια αναζήτησης-ανίχνευσης/επαναμετάδοσης) και congestion control(έλεγχος συμφόρησης) για το νέο πρωτόκολλο. Το UDP παρέχει ένα επίπεδο διεπαφής μετάδοσης του IP με ανίχνευση λάθους. Αυτή η ευκαμψία είναι ικανή-κατάλληλη για να χτίσουμε ένα νέο πρωτόκολλο στο application level πάνω στο UDP.

Ο στόχος της αποδοτικότητας απαιτεί ότι το πρωτόκολλο είναι απλό και «ελαφρύ». Η επικεφαλίδα του πρωτοκόλλου, τα υψηλά πακέτα ελέγχου, και ο υψηλός υπολογισμός πρέπει να κοστίζουν όσο το δυνατόν λιγότερους πόρους του συστήματος.

Περισσότερο σημαντικό, ο στόχος της αποδοτικότητας επίσης απαιτεί ότι το πρωτόκολλο πρέπει να προσαρμόζεται από μόνος του στις δικτυακές αλλαγές αυτόματα και γρήγορα-ταχύτατα. Ένας μηχανισμός ελέγχου του ρυθμού με ταχύ αντίδραση και μικρή ταλάντωση είναι επιθυμητός.

Εν τω μεταξύ, δικαιοσύνη είναι μία από τις βασικές ιδιότητες για ένα πρωτόκολλο μετάδοσης. Στο UDT μία ειδική απαίτηση δικαιοσύνης είναι η ανεξαρτησία του RTT(δικαιοσύνη μεταξύ πρωτοκόλλων με διαφορετικά RTT). Την ίδια στιγμή, χρειάζεται να μελετήσουμε την φιλικότητα του TCP, από τότε που το TCP είναι το de facto πρωτόκολλο μετάδοσης στο Internet. Ωστόσο, λαμβάνουμε υπόψιν ότι υπάρχει αντιλογία για τους σκοπούς της δικαιοσύνης, όσο το TCP δεν είναι RTT ανεξάρτητο και είναι αναποτελεσματικό σε υψηλά BDP περιβάλλοντα. Χρειάζεται να κάνουμε μία ανταλλαγή: σε χαμηλά BDP δίκτυα όπου το TCP μπορεί να δουλέψει κανονικά, το UDT πρέπει να εξασφαλίζει παρόμοιο εύρος όσο και η ροή TCP μπορεί, ενώ σε υψηλά BDP δίκτυα, το UDT μπορεί να εξασφαλίζει το εύρος όπου το TCP δεν μπορεί να αξιοποιήσει εφόσον αυτό αφήνει αρκετό διάστημα για το TCP για να αυξήσει την θυρίδα-παράθυρο.

Η σταθερότητα είναι ένας σημαντικός στόχος για να εγγυηθούμε την ιδιότητα της ασφάλειας του UDT. Το πρωτόκολλο πρέπει να είναι σταθερό σε δικτυακές αλλαγές και να συγκλίνει στην αποδοτικότητα και τη δικαιοσύνη σε ισορροπία σε κάθε κατάσταση. Ειδικά, η κατάσταση συμφόρησης πρέπει να αποφεύγεται.

## UDT πρωτόκολλο

### 1. Γενικά για το πρωτόκολλο

Το UDT είναι ένα end – to – end(άκρο με άκρο) unicast πρωτόκολλο μετάδοσης. Στην ουσία το UDT έχει δύο μέρη: τον αποστολέα και τον παραλήπτη. Ο αποστολέας είναι υπεύθυνος για τα πακέτα αποστολής δεδομένων και ο παραλήπτης είναι υπεύθυνος για τα πακέτα παραλαβής δεδομένων, για τα πακέτα ελέγχου αποστολής και παραλαβής, και για το χρονιστή λήξης – εκπνοής αναζήτησης. Όλα τα πακέτα δεδομένων και ελέγχου και στις δύο κατευθύνσεις μεταφέρονται ανάμεσα σε ένα ζευγάρι από UDP ports(θύρες). Ο τοπικός αποστολέας και παραλήπτης και η ισότιμη πλευρά του παραλήπτη συμφωνεί σε μια ροή δεδομένων σε μια κατεύθυνση. Αυτό είναι συμμετρικό και προς την άλλη αντίθετη κατεύθυνση.

Σημειώστε ότι κάθε UDT ύπαρξη έχει 1 αποστολέα και 1 παραλήπτη, τα οποία δεν είναι δύο UDT υπάρξεις όπου συμφωνούν σε μια σύνδεση μεταφοράς δεδομένων.

Θα χρησιμοποιήσουμε μία πλευρά αποστολής δεδομένων και μία πλευρά παραλαβής δεδομένων για να το παρουσιάσουμε(figure 1.)

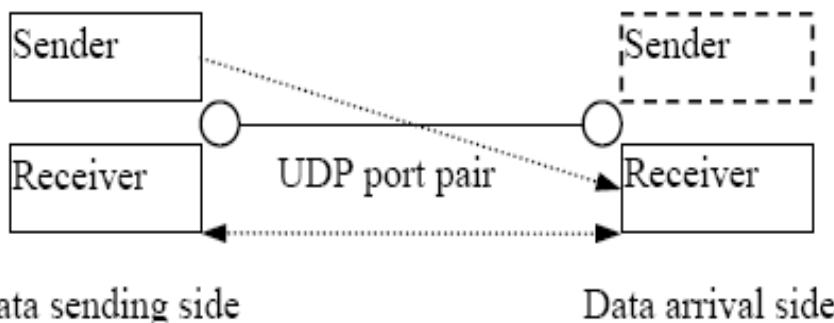


figure 1. Αρχιτεκτονική UDT πρωτοκόλλου (μονή διεύθυνση)

Ο αποστολέας βεβαιώνει ένα υψηλής ακρίβειας χρονιστή (ή χρησιμοποιεί τον χρονιστή του συστήματος αν είναι διαθέσιμος) για να προγραμματίσει χρονικά τα πακέτα. Ο εσωτερικός χρόνος του πακέτου αναβαθμίζεται από τον αλγόριθμο ελέγχου του ρυθμού. Το πακέτο αποστολής επίσης επηρεάζεται από τον έλεγχο ροής, όπου οριοθετεί το ανώτατο αριθμό από μη βεβαιωμένων πακέτων.

Ο παραλήπτης υποστηρίζει 4 αυτοχρονομετρούμενους χρονιστές, όπου εξετάζονται κάθε φορά βεβαιωμένης λήψης, loss report(αναφορά απώλειας), rate control(έλεγχος ρυθμού), και timer out detection(τερματισμό χρονιστή αναζήτησης), αντίστοιχα. ACK και SYN



χρονιστές ανανεώνονται κατά τη διάρκεια του UDT σύμφωνα προς το RTT και την τωρινή ταχύτητα μεταφοράς δεδομένων.

Το σταθερό και αμετάβλητο SYN διάστημα στο UDT είναι 0.01 seconds. Αυτό το νούμερο είναι μία εντυπωσιακή ευπρόσδεκτη ανταλλαγή(trade off) μεταξύ αποδοτικότητας και δικαιοσύνης (και τα δύο εσωτερικό πρωτόκολλο δικαιοσύνης και TCP-φιλικότητας), αντίθετα με τη θεωρητική τιμή. Το διάστημα ACK είναι ίσο με το διάστημα SYN. Το διάστημα NAK είναι μεγαλύτερο μεταξύ του RTT και του μέσου όρου του διαστήματος της λήψης πακέτων. Τελικά, το διάστημα EXP είναι  $(2*RTT+SYN)$ .

## 2.Διάταξη πακέτου

Υπάρχουν δύο βασικές κατηγορίες πακέτων στο UDT: τα πακέτα δεδομένων και τα πακέτα ελέγχου. Αυτά διακρίνονται από το πρώτο bit της επικεφαλίδας του πακέτου.

Η επικεφαλίδα του πακέτου δεδομένων είναι μία σημαία(flag) bit «0» ακολουθούμενη από 31 bit σειριακό αριθμό(sequence) αριθμό. Η τιμή του σειριακού αριθμού είναι μεταξύ του «0» και του  $(2^{30}-1)**$ . Το UDT δεν επιτρέπει στο μέγεθος του πακέτου να είναι μεγαλύτερο από το MTU(maximum transfer unit)(μέγιστη μονάδα μεταφοράς), έτσι το μεγαλύτερο μέγεθος εφαρμογής δεδομένων μπορεί να κουβαληθεί από ένα πακέτο σε  $(MTU-32)bytes$ , όπου 32 είναι το άθροισμα του UDP/IP μεγέθους επικεφαλίδας(28bytes) και UDT μεγέθους επικεφαλίδας(4bytes). Το UDT πάντα προσπαθεί να πακετάρει δεδομένα στο μέγιστο μέγεθος, και η μονάδα του αριθμού των πακέτων ανά δευτερόλεπτο χρησιμοποιείται για να μετρήσουμε την ταχύτητα μετάδοσης.

\*\* Χρησιμοποιώντας  $(2^{30}-1)$  αντί του  $(2^{31}-1)$  είναι λόγω της ικανότητας υλοποίησης σχετικά με τον έλεγχο υπερχείλισης.

Αν το πρώτο bit της επικεφαλίδας του UDT είναι 1, τότε το πακέτο αναλύεται σαν ένα πακέτο ελέγχου, το οποίο προσδιορίζεται στο figure 2.

Υπάρχουν 6 τύποι ελέγχου πακέτου στο UDT, οι οποίοι χαρακτηρίζονται από το πεδίο τύπου(type field) στην επικεφαλίδα. Οι πρώτοι δύο τύποι(0 και 1) χρησιμοποιούνται για να βεβαιώσουν την κατάσταση σύνδεσης, ενώ τα υπόλοιπα 4(2,3,4 και 6) σχετίζονται με την αξιοπιστία και τον έλεγχο συμφόρησης.

Bit 0: Flag = 1	Bit 1-3: Type	Bit 4-15: Reserved	Bit 16-31: ACK ID or Loss Length
Control Information Field			
Content depends on <i>type</i> field:			
type 000 (handshake): maximum flow window size, MTU			
type 001 (keep-alive): None			
type 010 (ACK): acknowledged sequence number, RTT, packet arrival speed, estimated bandwidth			
type 011 (NAK): loss information			
type 100 (delay increase warning): None			
type 110 (ACK <sup>2</sup> ): None			

Τύπος 2(010) και τύπος 6(110) είναι βεβαίωση λήψης(ACK) και βεβαίωση λήψης από τη βεβαίωση λήψης ( $ACK^2$ ). Έχουν το δικό τους 16 bit σειριακό αριθμό(ACK ID) τοποθετημένο στην επικεφαλίδα στα bit 16-31, το οποίο χρησιμοποιείται για ακριβή υπολογισμό του RTT. Τα πακέτα ACK κουβαλάνε τον σειριακό αριθμό από bit λέγοντας ότι ο παραλήπτης έχει λάβει όλα τα πακέτα προηγουμένως αυτού του αριθμού. Επίσης μεταφέρει το RTT, την ταχύτητα άφιξης του πακέτου, και το υπολογισμένο εύρος.

Τύπος 3(011) είναι αναφορά απώλειας ή αρνητική βεβαίωση λήψης(NAK). Ο συνολικός αριθμός των χαμένων πακέτων μεταφέρεται σε αυτό το πακέτο και είναι γραμμένο στα bit από 16-31. Η ακριβής πληροφορία απώλειας είναι στο πεδίο «έλεγχος πληροφορίας», το οποίο είναι συμπιεσμένο με τη περιεκτική διατύπωση που περιγράφεται παρακάτω.

«Η πληροφορία απώλειας μεταφέρεται στο πακέτο NAK σαν ένας πίνακας από 32-bitους ακέραιοις. Αν ένας ακέραιοις μέσα στον πίνακα είναι ένας κανονικός σειριακός αριθμός(το πρώτο bit είναι 0), αυτό σημαίνει ότι το πακέτο με τον ίδιον σειριακό αριθμό από bit είναι χαμένο. Αν το πρώτο bit είναι 1, αυτό σημαίνει ότι όλα τα πακέτα που αρχίζουν από αυτό τον αριθμό(μαζί με τον αριθμό αυτό) έως τον επόμενο αριθμό(εκτός αυτού) στον πίνακα(που το πρώτο του bit πρέπει να είναι 0) είναι χαμένα.

Τύπος 4(100) είναι ένα ειδικό μήνυμα με κενό έλεγχο πληροφορίας από την πλευρά λήψης των δεδομένων, ο παραλήπτης λέει ότι έχει αναγνωρίσει τάση αυξανόμενης καθυστέρησης από την διακύμανση του RTT.

### 3.Αλγόριθμος αποστολής και λήψης

Για να βοηθήσουμε στην εξήγηση του αλγόριθμου αποστολής λήψης, εισάγουμε μία δομή δεδομένων που ονομάζεται «loss list», η οποία χρησιμοποιείται για να καταγράψουμε τον σειριακό αριθμό των χαμένων πακέτων. Και ο αποστολέας και ο παραλήπτης έχουν το δικό τους «loss list».

Ο αποστολέας κυριολεκτικά αποστέλλει 1 πακέτο κάθε εσωτερικό χρόνο του πακέτου το οποίο αποφασίζεται από τον μηχανισμό ελέγχου ρυθμού. Επαναμεταδοτούμενο πακέτο έχει υψηλότερη προτεραιότητα από την πρώτη φορά του πακέτου. Όταν ένα πακέτο είναι resent packet τότε αυτό μετακινείται από του αποστολέα την loss list. Ο αποστολέας δεν θα στείλει κανένα νέο πακέτο δεδομένου αν ο αριθμός των μη βεβαιωμένης λήψης πακέτων υπερβαίνει ένα κατώφλι, ή το μέγεθος του παραθύρου ροής, το οποίο ανανεώνεται από το μηχανισμό ελέγχου ροής.

Αλγόριθμος1 αποστολής.

- 1) If there is no application data to send, sleep until it is waken up by the application.
- 2) Packet sending:
  - a) If the sender's lost list is not empty, remove the first lost sequence number from the list and send the proper packet out.
  - b) Otherwise, if the number of unacknowledged packets does not exceeded the flow window size, pack a new packet and send out.
- 3) Update the number of sent packets since last SYN time.
- 4) If the current packet and the next packet are sampled probing packet pair. Go to 1).
- 5) Wait to the next packet sending time; Wait for additional SYN time if the rate control has decided that data sending should be frozen. Go to 1).

1)Αν υπάρχουν δεδομένα εφαρμογής για να στείλεις, κοιμήσου μέχρι να σε ξυπνήσει η εφαρμογή.

2)Αποστολή πακέτου:

a) Αν η loss list του αποστολέα δεν είναι κενή, μετακίνησε τον πρώτο σειριακό αριθμό από την λίστα και στείλε το κατάλληλο πακέτο.

β) Αλλιώς, αν ο αριθμός των μη βεβαιωμένων λήψης πακέτα δεν υπερβαίνουν το μέγεθος του παραθύρου ροής, πακέταρε ένα νέο πακέτο και στείλε το.

3) Ανανέωσε τον αριθμό των σταλθέντων πακέτων από το τελευταίο SYN.

4) Αν το τωρινό πακέτο και το επόμενο πακέτο είναι (sampled probing packet pair) πήγαινε στο 1).

5) Περίμενε τον επόμενο χρόνο αποστολής πακέτου. Περίμενε για επιπρόσθετο SYN χρόνο αν ο έλεγχος ρυθμού έχει αποφασίσει ότι η αποστολή δεδομένων πρέπει να παγώσει. Πήγαινε στο 1).

Αλγόριθμος2 λήψης.

- 1) Query the timers
  - a) If ACK timer is expired and there are new packets to acknowledge, send back an ACK report;
  - b) If NAK timer is expired and the receiver's loss list is not empty, send back an NAK report;
  - c) If SYN timer is expired:
    - i) If the number of received packets since last SYN time is greater than 0, update the NAK interval.
    - ii) If the number of sent packets since last SYN time is greater than 0, update sending rate.
  - d) If EXP timer is expired, put all the sequence numbers of sent packets since the last acknowledged number into the sender's loss list.
  - e) Reset the expired timers.
- 2) Start time bounded UDP receiving. If nothing received before the UDP timer expires, go to 1).
- 3) If the received packet is a control packet, process it, and reset EXP timer; if it is ACK or NAK; go to 1);
- 4) Compare the sequence number of current data packet (A) and the largest sequence number ever received (B):
  - a) If  $A > B + 1$ , generate a loss report, put the sequence number between A and B into the receiver's loss list;
  - b) If  $A < B$ , remove A from the receiver's loss list;
  - c) Update B;
- 5) If the size of current packet is not equal to the fixed UDT packet size, record the current sequence number and the size difference (for buffer management use). Go to 1).

- 1) Εξετάζω τους χρονιστές
  - α) Αν ACK χρονιστής έχει λήξει και υπάρχουν νέα πακέτα για βεβαίωση λήψης, στείλε πίσω μία ACK αναφορά.
  - β) Αν NAK χρονιστής έχει λήξει και του παραλήπτη η loss list δεν είναι άδεια, στείλε πίσω μία NAK αναφορά.
  - γ) Αν SYN χρονιστής έχει λήξει
    - i) Αν ο αριθμός των ληφθέντων πακέτων από το τελευταίο SYN είναι μεγαλύτερος του 0, ανανέωσε το NAK διάστημα.
    - ii) Αν ο αριθμός των αποσταλμένων πακέτων από το τελευταίο SYN είναι μεγαλύτερος του 0, ανανέωσε το ρυθμό αποστολής.
  - δ) Αν ο EXP χρονιστής, τοποθετήσει όλους τους σειριακούς αριθμούς των αποσταλμένων πακέτων από τη τελευταία βεβαιωμένη λήψη αριθμού μέσα στη loss list του αποστολέα.
  - ε) Επανάφερε τους ληγμένους χρονιστές.
- 2) Άρχισε ο χρόνος UDP λήψης. Αν τίποτα δεν έχει ληφθεί πριν ο UDP χρονιστής λήξει, τότε πήγαινε στο 1).
- 3) Αν το λαμβανόμενο πακέτο είναι ένα πακέτο ελέγχου, επεξεργάσου το, και επανέφερε τον EXP χρονιστή. Αν αυτό είναι ACK ή NAK τότε πήγαινε στο 1).
- 4) Σύγκρινε τον σειριακό αριθμό από bit του τωρινού πακέτου δεδομένων(A) και του μεγαλύτερου σειριακού αριθμού που έχει ποτέ ληφθεί(B).
  - α) Αν  $A > B + 1$ , δημιούργησε μία loss αναφορά, τοποθέτησε τον σειριακό αριθμό μεταξύ του A και B μέσα στο παραλήπτη την loss list.
  - β) Αν  $A < B$ , μετακίνησε το A από του παραλήπτη την loss list.
  - γ) Ανανέωσε το B.
- 5) Αν το μέγεθος του τωρινού πακέτου δεν είναι ίσο με το αμετάβλητο UDT μέγεθος πακέτου, κατάγραψε το τωρινό σειριακό αριθμό και το διαφορετικό μέγεθος(για χρήση του manager του buffer) πήγαινε στο 1).

Ο παραλήπτης λαμβάνει και πακέτα δεδομένων και ελέγχου. Αναλύει την επικεφαλίδα του πακέτου και επεξεργάζομαι τα δεδομένα ανάλογα με τον τύπο του πακέτου. Για πακέτα ελέγχου, η επεξεργασία θα δοθεί στην επόμενη ενότητα. Για πακέτα δεδομένων, ο παραλήπτης ελέγχει τον σειριακό αριθμό και γράφει τα δεδομένα σε κατάλληλη θέση στο buffer(ενδιάμεση μνήμη) του πρωτοκόλλου. Το πακέτο μπορεί να απορριφθεί αν έχει βεβαίωση λήψης. Αν ο τωρινός σειριακός αριθμός είναι μεγαλύτερος από τον μεγαλύτερο σειριακό αριθμό που έχει ληφθεί +1, όλοι οι σειριακοί αριθμοί μεταξύ των δύο αριθμών θα εισέλθουν στην loss list του παραλήπτη. Μία NAK αναφορά θα δημιουργηθεί αμέσως και στέλνεται πίσω. Αν το πακέτο είναι resent packet(ο σειριακός αριθμός είναι μικρότερος από το μεγαλύτερο που έχει ληφθεί), μετακινείται από τη loss list.

Ο παραλήπτης επίσης διατηρεί όλους τους χρονιστές με αυτοχρονομέτρηση. Ελέγχει το τωρινό χρόνο κάθε φορά που γυρνά από ένα χρόνο διαβάσματος μιας πόρτας UDP. Όταν το SYN λήξει, ένας έλεγχος ρυθμού σκανδαλίζεται. Ο EXP χρονιστής λήξης θα εισάγει όλα τα μη βεβαιωμένα λήψης πακέτα μέσα στη λίστα του αποστολέα. ACK και NAK χρονιστές θα σκανδαλίσουν τις αναφορές βεβαιωμένης λήψης αν υπάρχει κάποια νέα ή χαμένα δεδομένα. Ο EXP χρονιστής επαναφέρεται κάθε φορά που μια βεβαίωση λήψης πακέτου λαμβάνεται (ACK ή NAK).

## 4. Επεξεργασία ελέγχου πληροφορίας

Εδώ θα εξηγήσουμε την πληροφορία ελέγχου ACK,  $ACK^2$  και NAK. Τα προειδοποιητικά πακέτα καθυστέρησης θα εξηγηθούν στον έλεγχο συμφόρησης. Τα άλλα δύο είδη πακέτων ελέγχου για επιβεβαίωση σύνδεσης δεν θα εξηγηθούν γιατί δεν σχετίζονται με μεταφορά δεδομένων.

### 4.1. ACK και $ACK^2$

Η αναφορά ACK σκανδαλίζεται όταν ένας χρονιστής ACK λήγει. Αν δεν υπάρχει απώλεια, ο σειριακός αριθμός βεβαιωμένης λήψης είναι ο μεγαλύτερος που έχει ληφθεί σειριακός αριθμός μείον 1. Αλλιώς είναι ο μικρότερος σειριακός αριθμός στην loss list του παραλήπτη. Το ACK στέλνεται μόνο αν ο σειριακός αριθμός βεβαιωμένης λήψης είναι μεγαλύτερος από το μεγαλύτερο βεβαιωμένης λήψης σειριακό αριθμό γνωστό στην πλευρά αποστολής δεδομένων (γνωστοποιημένο από το  $ACK^2$ ), και αν είναι ίσο με τον αριθμό στην τελευταία αποστολή ACK, το διάστημα μεταξύ αυτών των δύο ACK πακέτων πρέπει να είναι το λιγότερο  $2 * RTT$ . Σε κάθε ACK είναι προσδιορισμένο ένα μοναδικό αυξανόμενο ACK ID.

Μόλις ένα ACK ληφθεί, οι συνδεόμενες δομές δεδομένων του αποστολέα ανανεώνονται και ένα  $ACK^2$  με το ίδιο ACK ID στέλνεται πίσω αμέσως. Μόλις ένα  $ACK^2$  ληφθεί, ο παραλήπτης ψάχνει για το αποσταλμένο ACK με το ίδιο ACK ID στο παράθυρο ACK history. Αν βρεθεί, ο ποιο πρόσφατος βεβαιωμένης λήψης σειριακός αριθμός γνωστοποιημένος στην πλευρά αποστολής δεδομένων ανανεώνεται, και το RTT επαναυπολογίζεται σύμφωνα με το χρόνο όταν το πακέτο ACK στάλθηκε και όταν το  $ACK^2$  παραλήφθηκε.

Όσο υψηλότερος είναι ο ρυθμός αποστολής, τόσο λιγότερο τα ACK και  $ACK^2$  δημιουργούνται. Στην χειρότερη περίπτωση, όταν ο ρυθμός

αποστολής είναι πολύ χαμηλός(όχι παραπάνω από 1 πακέτο στέλνεται σε 1 SYN χρόνο), υπάρχει ισοδύναμος αριθμός από ACK και  $ACK^2$  πακέτα σε πακέτα δεδομένων. Αν η ταχύτητα μεταφοράς είναι 1 Gbps και το MTU είναι 1500 bytes, υπάρχει μόνο 1 ACK και 1  $ACK^2$  για κάθε 833 πακέτα δεδομένων.

Το ACK πακέτο επίσης μεταφέρει την RTT πληροφορία στην ισοδύναμη πλευρά, αφού το RTT είναι υπολογισμένο στην πλευρά άφιξης δεδομένων, αλλά η πλευρά αποστολής δεδομένων επίσης χρειάζεται να το ξέρει. Επίσης, η ταχύτητα άφιξης πακέτου και η εκτιμηταία πληροφορία εύρους μεταφέρεται σε ένα ACK πακέτο.

#### 4.2.NAK

Το NAK σκανδαλίζεται όταν ένας NAK χρονιστής λήγει ή όταν εντοπίζεται απώλεια κατά τη διάρκεια της λήψης δεδομένων. Στην προηγούμενη κατάσταση, η loss list εξετάζεται και εκείνα τα πακέτα του οποίου η τελευταία αναφορά NAK χρονομέτρησε είναι το λιγότερο  $k*RTT$  πριν διαλεχτεί και σταλεί πίσω, όπου το k αρχικοποιείται σε 2 και αυξάνεται κατά 1 κάθε φορά που το πακέτο αναφέρεται. Στην τελική κατάσταση, μόνο η απώλεια πακέτου ανιχνεύεται από το τωρινό πακέτο που στέλνεται πίσω.

Η πληροφορία απώλειας για να σταλεί πίσω συμπιέζεται σύμφωνα με τα προηγούμενα. Αν το μέγεθος είναι μεγαλύτερο από το 1 UDT πακέτο που μπορεί να μεταφέρει(MTU- μέγεθος επικεφαλίδας πακέτου), μόνο εκείνα που θα μπορούν να περιέχονται σε 1 πακέτο στέλνονται πίσω, και τα υπόλοιπα θα σταλούν σε μελλοντικό NAK πακέτο αν είναι απαραίτητο.

Το NAK δημιουργείται μόνο αν υπάρχει αναφορά απώλειας. Στην χειρότερη κατάσταση, ο αριθμός των άμεσων NAK είναι ο μισός των αποσταλμένων πακέτων όταν κάθε άλλο πακέτο έχει χαθεί. Η πιθανότητα είναι πολύ μικρή μιας και η απώλεια είναι συχνά συνεχόμενη. Το διάστημα NAK δεν είναι μεγαλύτερο από το διάστημα άφιξης πακέτου σύμφωνα με τον αλγόριθμο παραλήπτη. Εν τω μεταξύ, το διάστημα NAK για κάθε μονή απώλεια πακέτων αυξάνει κάθε φορά έτσι ώστε ο αποστολέας δεν θα είναι τόσο απασχολημένος με την επεξεργασία NAK για να στείλει έξω κάθε ένα πακέτο.

## 5. Έλεγχος συμφόρησης

Το UDT απασχολεί-χρησιμοποιεί τον έλεγχο συμφόρησης βασισμένο και στη θυρίδα-παράθυρο και στο ρυθμό.

Ο έλεγχος ρυθμού κυρίως αποφασίζει την απόδοση, ενώ ο έλεγχος ροής βασισμένος στη θυρίδα-παράθυρο είναι μια σημαντική υποστηρικτική μέθοδος για αποδοτικότητα και δικαιοσύνη.

### 5.1. Έλεγχος ροής

Το UDT χρησιμοποιεί έναν έλεγχο ροής παραθύρου-θυρίδας για να οριοθετήσει τον αριθμό των μη βεβαιωμένης λήψης πακέτων.

Ο παραλήπτης καταγράφει τα διαστήματα άφιξης των πακέτων. Όταν είναι η ώρα να ανατροφοδοτήσει με ACK, χρησιμοποιεί τα πιο πρόσφατα 16 διαστήματα άφιξης πακέτων για να υπολογίσει την ταχύτητα άφιξης πακέτων(AS) με τον ακόλουθο αλγόριθμο:

Ο παραλήπτης ψάχνει το μέσο M των 16 διαστημάτων, και μετακινεί αυτές τις τιμές οι οποίες είναι είτε μεγαλύτερες από 8M ή μικρότερες από M/8. Αν είναι παραπάνω από το μισό(8) παραμένουν, ο παραλήπτης υπολογίζει το μέσο διάστημα άφιξης πακέτων από τις υπόλοιπες τιμές και τότε παίρνει την ταχύτητα άφιξης πακέτων(αριθμός πακέτων ανά δευτερόλεπτο). Αλλιώς επιστρέφεται το 0.

Από την πλευρά του αποστολέα, όταν λαμβάνει ένα ACK και το AS είναι μεγαλύτερο από 0, το μέγεθος W της ροής παραθύρου-θυρίδας ανανεώνεται σύμφωνα με:

$$W = W * 0.875 + (RTT + SYN) * AS * 0.125$$

Ο έλεγχος ροής είναι σημαντικός για τη μείωση της απώλειας πακέτων. Κατά τη διάρκεια συμφόρησης, αναφορές απώλειας από την πλευρά άφιξης δεδομένων μπορεί να χάνονται ή να καθυστερούνται, έτσι η πλευρά αποστολής ίσως συνεχίσει να στέλνει νέα πακέτα και χειροτερεύει τη συμφόρηση. Το παράθυρο-θυρίδα ελέγχου ροής εμποδίζει αυτό να συμβεί.

Ελαττώνοντας την απώλεια πακέτων, ο έλεγχος ροής βοηθάει να βελτιωθεί η αποδοτικότητα και η δικαιοσύνη, αφού ο χαμηλός ρυθμός απώλειας μειώνει την συχνότητα του ρυθμού μείωσης αποστολής και κάνει το UDT λιγότερο επιθετικό.

Το μέγεθος παραθύρου ροής ξεκινάει από το 2 στο slow start, και ανανεώνεται στο μεγαλύτερο σειριακό αριθμό βεβαιωμένης λήψης όταν ο αποστολέας λαμβάνει ένα ACK. Το slow start τελειώνει όταν η πλευρά αποστολής δεδομένων λαμβάνει ένα NAK ή ένα προειδοποιητικό μήνυμα καθυστέρησης, μετά ο αλγόριθμος ελέγχου ροής που περιγράψαμε εφαρμόζεται.



## 5.2. Έλεγχος ρυθμού

Σε κάθε SYN χρόνο, ο αποστολέας υπολογίζει το υπερφορτωμένο εκθετικά μέσο όρο του ρυθμού απώλειας πακέτων (από τον αριθμό της απώλειας σε NAK πακέτα και τον συνολικό αριθμό των πακέτων που έχουν σταλεί έξω). Αν ο ρυθμός απώλειας είναι μικρότερος από ένα μικρό κατώφλι (1%), ο αριθμός των πακέτων που στάλθηκαν ανά SYN χρόνο αυξάνεται σύμφωνα με:

$$\max(10^{\lceil \log_{10} B \rceil} * SYN / 1000, 1 / MTU)$$

όπου το B είναι το υπολογιζόμενο εύρος. Το εσωτερικό διάστημα του πακέτου εν συνεχεία επαναυπολογίζεται.

Το υπολογιζόμενο εύρος εξετάζεται συστηματικά από δείγματα UDT πακέτων δεδομένων. Κάθε 16 πακέτα, ο αποστολέας δεν περιμένει το επόμενο χρόνο αποστολής πακέτου αλλά στέλνει έξω το επόμενο πακέτο αμέσως, έτσι ώστε τα δύο συνεχόμενα πακέτα από συστηματική έρευνα πακέτου να ζευγαρώνουνε-συνδυάζονται. Ο παραλήπτης στην πλευρά λήψης δεδομένων καταγράφει τον χρόνο άφιξης μεταξύ των ζευγαρωμένων πακέτων στο history window. Όταν ένα ACK στέλνεται, ο παραλήπτης υπολογίζει το μέσο των τελευταίων 16 διαστημάτων άφιξης των εξεταζόμενων συστηματικά ζευγαριών πακέτων MP, τότε το νέο υπολογιζόμενο εύρος NB (=1/MP), και στέλνει το NB σε κάθε ισότιμη-ισοδύναμη πλευρά εσωτερικά του ACK πακέτου. Το εύρος ανανεώνεται από μέσο όρο φορτωμένης εκθετικής συνάρτησης στην πλευρά δεδομένων αποστολής:

$$B = B * 0.875 + NB * 0.125$$

Μείωση του ρυθμού αποστολής μπορεί να σκανδαλιστεί από την απώλεια πακέτου ή την καθυστέρηση πακέτου, όπου και τα δύο χρησιμοποιούνται ως ενδείξεις της συμφόρησης στο UDT.

Το UDT χρησιμοποιεί την διακύμανση RTT για να μετρήσει την καθυστέρηση των πακέτων. Όταν ένα ACK<sup>2</sup> ληφθεί και νέο RTT είναι υπολογισμένο, το UDT χρησιμοποιεί τα τελευταία 16 νέα RTTs (όχι τη μέση τιμή) για να ελέγξουμε την καθυστέρηση με τη μέθοδο του pathload.

Αν μια αυξανόμενη τάση καθυστέρησης ανιχνευτεί, ένα προειδοποιητικό μήνυμα καθυστέρησης στέλνεται στην πλευρά της αποστολής δεδομένων αλλά όχι περισσότερα από 1 προειδοποιητικό πακέτο θα σταλεί σε 2\*RTT.

Όταν η πλευρά αποστολής δεδομένων λαμβάνει ένα προειδοποιητικό μήνυμα καθυστέρησης, ανανεώνει το εσωτερικό χρόνο πακέτου I σύμφωνα με:

$$I = I * 1.125$$

Εν τω μεταξύ, καταγράφει τον τωρινό μεγαλύτερο σειριακό αριθμό όπου έχει σταλεί (last\_dec\_seq) και επανεκκινεί τον μειώμενο μετρητή παράμετρο dc κατά 1, το κατώφλι παράμετρο e κατά 4.

Όταν η πλευρά αποστολής δεδομένων λαμβάνει μία αναφορά απώλειας, ελέγχεται αν το πρώτο χαμένο σειριακό αριθμό είναι μεγαλύτερος από το `last_dec_seq`. Αν η κατάσταση είναι ικανοποιητική, κάνει την ίδια ενέργεια όταν ληφθεί προειδοποίηση καθυστέρησης. Αλλιώς αυξάνει το `dc` κατά 1, και αναπτύσσει μόνο αν το  $dc = 2^e$ . Όταν γίνει, αυξάνεται το `e` κατά 1.

Κάθε φορά η μεταβλητή `last_dec_seq` ανανεώνεται, η αποστολή δεδομένων είναι παγωμένη για SYN χρόνο(κανένα πακέτο δεν θα σταλεί κατά τη διάρκεια του SYN χρόνου).

Ο αρχικός ρυθμός αποστολής είναι 0 κατά τη διάρκεια της φάσης `slow start` και επανεκκινεί στο διάστημα άφιξης πακέτου όταν το `slow start` τελειώνει. Μετά ο αλγόριθμος ελέγχου ροής που περιγράφηκε παραπάνω αρχίζει να δουλεύει.

## Ανάλυση και αποτίμηση

### 1.Αποδοτικότητα

Αφού το UDT χρησιμοποιεί την καθυστέρηση σαν ένα από τα κριτήρια συμφόρησης, είναι η ελεύθερη απώλεια υπό τον όρο ότι το μέγεθος της ουράς αναμονής(Drop Tail) είναι αρκετά μεγάλο για το UDT για να ανιχνεύσει την καθυστέρηση. Σε αυτή την κατάσταση, το εύρος εκμετάλλευσης μπορεί να φτάσει τουλάχιστον το 94,4% $(=(1+1/1.125)/2)$ , υποθέτωντας ότι ο ρυθμός αποστολής αλλάζει μεταξύ 1 και 1/1.125).

Αυτό είναι ικανοποιητικό αν:

A) ο ρυθμός απώλειας συμβάντος δεν είναι αρκετά μεγάλος για να προκαλέσει συνεχόμενη μείωση(ο αριθμός απώλειας συμβάντων προκαλείται από την αύξηση μικρότερη από  $2^4$  και

B) μία μονή μείωση μπορεί να εξουδετερώσει την αύξηση σε 1 RTT.

UDT εύρος εκμετάλλευσης με ουρά αναμονής DropTail

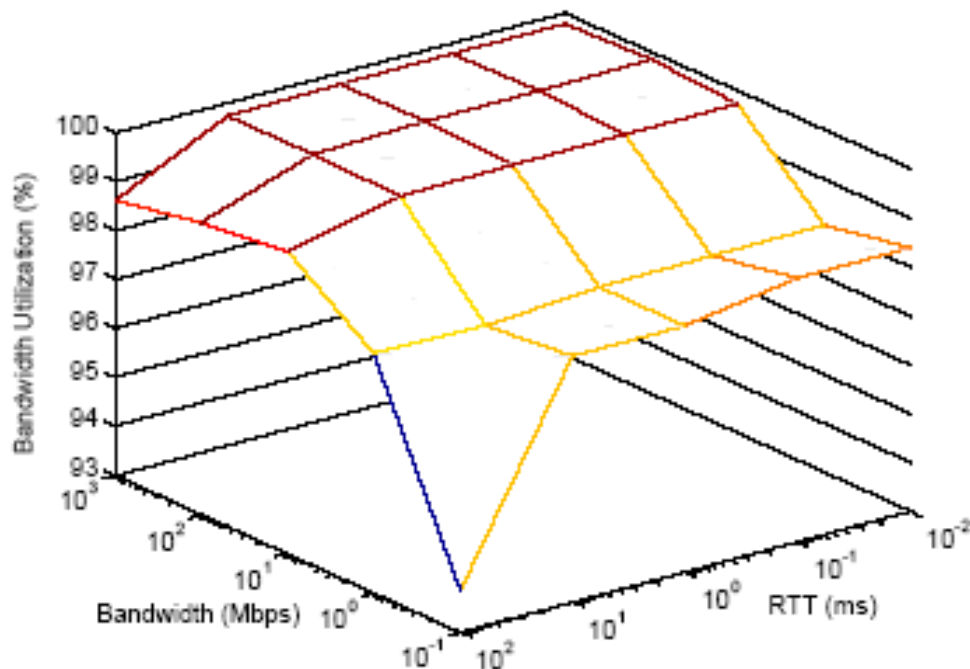
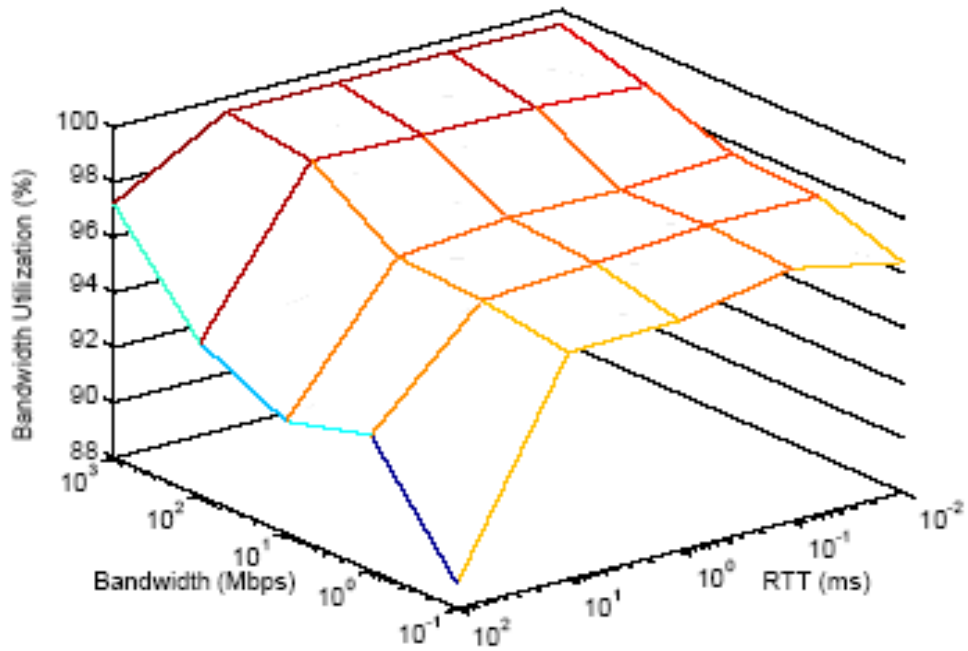
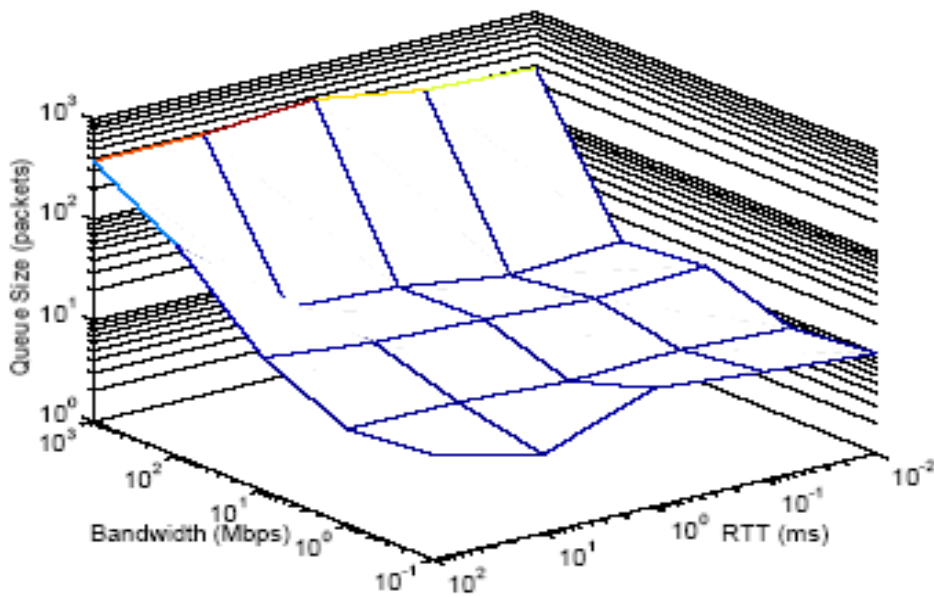


Figure 3: UDT bandwidth utilization with DropTail queue.



Εύρος εκμετάλλευσης και μέσο μέγεθος ουράς αναφοράς με RED ουρά αναφοράς



**Figure 4: Bandwidth utilization and average queue size with RED queue.**

Αυτές οι 2 καταστάσεις επηρεάζονται κυρίως από το RTT. Χρησιμοποιούμε προσομοίωση για να εξετάσουμε το εύρος εκμετάλλευσης κάτω από διαφορετικά RTT, χωρητικότητας συνδέσμων και σχέδια διαχείρισης ουρών αναμονής. Τα αποτελέσματα φαίνονται στα figure 4 και 5. Αυτή η προσομοίωση και όλα τα υπόλοιπα έγιναν στο NS-2.

Στο figure 3 δείχνει την απόδοση του UDT να αυξάνεται όσο το εύρος αυξάνεται και το RTT να μειώνεται. Το εύρος εκμετάλλευσης είναι γύρω στο 98% στα περισσότερα σενάρια, αναμενόμενο όταν το εύρος είναι μικρότερο από 1MB και RTT είναι μεγαλύτερο από 10ms.

Το figure 4 δείχνει παρόμοια τάση με τη RED διαχείριση ουράς αναμονής.

Το μέσο μέγεθος ουράς αναμονής επίσης δίνεται στο figure 5, το οποίο αυξάνει όσο το εύρος αυξάνεται. Με τη διαχείριση ουράς αναμονής RED, το UDT μπορεί να φτάσει παρόμοια αποδοτικότητα όπως το DropTail ουρά αναμονής, αλλά το μέσο μέγεθος ουράς αναμονής είναι πολύ χαμηλότερο.

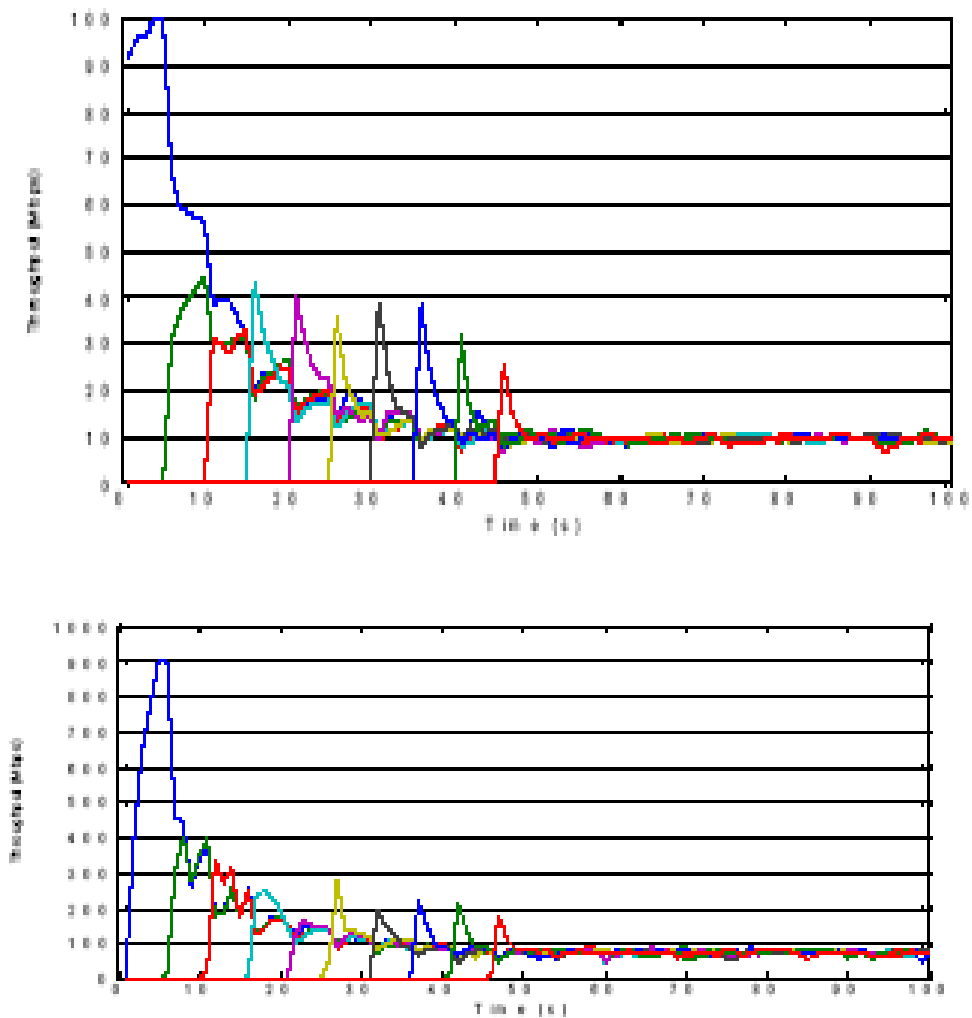
Η απόδοση του UDT επίσης οφείλει στην απευθείας αναμφίβολη-ρητή ανατροφοδότηση απώλειας, που μπορεί να ειδοποιήσει την απώλεια γρηγορότερα από 3-διπλά-ACK με χρονιστή μηχανισμό στο TCP. Επίσης επιτρέπει την πλευρά αποστολής δεδομένων να ξαναστείλει πίσω μόνο αυτά τα χαμένα πακέτα.

Τελικά, αφού το UDT χρησιμοποιεί μία παράμετρο αύξησης σχετική με τη χωρητικότητα συνδέσμου και ένα σταθερό διάστημα αύξησης, ο χρόνος σύγκλισης είναι σχεδόν ανεξάρτητος από το εύρος και την καθυστέρηση.

## 2. Δικαιοσύνη

Ο έλεγχος ρυθμού κυρίως αποφασίζει την κίνηση της ροής UDT αφού το μέγεθος παραθύρου-θυρίδας δεν είναι ποτέ μικρότερο από το γινόμενο του ρυθμού λήψης και του  $(RTT+SYN)$ . Στην πραγματικότητα ένα AIMD (additive increase multiplicative decrease) (προσθετική αύξηση πολλαπλάσια μείωση) αλγόριθμο, η παράμετρος αύξησης είναι σχετική με τη χωρητικότητα άκρο σε άκρο συνδέσμου και είναι σταθερή για όλες τις ροές που μοιράζονται το ίδιο στενό πέρασμα, και ο παράγων μείωσης είναι το  $1/9 (= 1 - 1/1.125)$ .

Για όλες τις UDT ροές μοιράζονται μόνο ένα στενό πέρασμα συνδέσμου, θα αναζητήσουν την ίδια χωρητικότητα συνδέσμου. Κάτω από αυτή τη κατάσταση, το UDT ικανοποιεί την δικαιοσύνη που το AIMD ικανοποιεί.



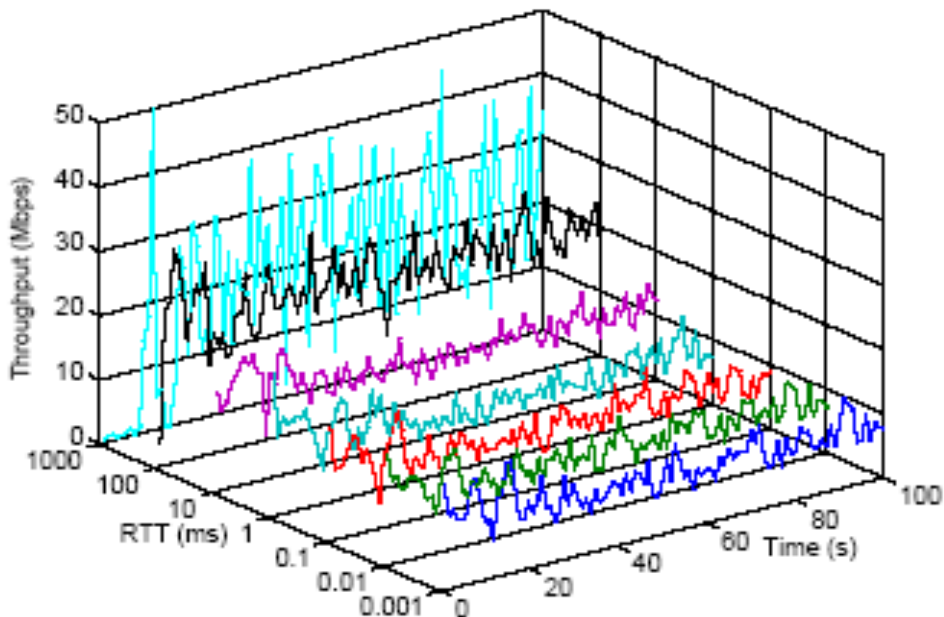
**Figure 5: 10 UDT flows share single bottleneck link, with 1 flow added every 5 seconds.**

figure 5. 10 ροές UDT μοιράζονται ένα μονό στενό πέρασμα συνδέσμου, με μία ροή να προστίθεται κάθε 5 δευτερόλεπτα.

Στο figure 5 αποδεικνύει τη δίκαιη μοιρασιά μεταξύ των UDT ροών. Στην προσομοίωση, μία UDT ροή προστίθεται κάθε 5 δευτερόλεπτα. Όλες οι ροές συγκλίνουν στη δίκαιη μοιρασιά γρήγορα και παραμένουν σε σταθερή κατάσταση. Οι ιδιότητες συνδέσμου είναι 100 Mbps με της RTT(επιπλέον) και 1Gbps με 100ms RTT(κάτω), αντίστοιχα. Η ουρά αναμονής DropTail χρησιμοποιείται σε αυτή την κατάσταση. Χρησιμοποιώντας σταθερό χρόνο συγχρονισμού αντί της βασισμένης στο διάστημα στο RTT, το UDT σχεδόν εξαλείφει το συστηματικό λάθος δικαιοσύνης από την καθυστέρηση δικτύου. Η επιρροή του RTT θα εξηγηθεί αργότερα.

Στο figure 6 αποδεικνύουμε την ανεξαρτησία του RTT χαρακτηριστικό του UDT χρησιμοποιώντας 7 παράλληλες UDT ροές μοιράζοντας 100 Mbps σε στενό πέρασμα συνδέσμου με RTT που κυμαίνεται από 1 microsecond σε 1 second, αλλά η κίνηση κυμαίνεται περίπου μεταξύ 10 και 20 Mbps. Η ουρά αναμονής DropTail χρησιμοποιείται σε αυτή την προσομοίωση.

Από τότε που το UDT χρησιμοποιεί τη καθυστέρηση για να υποδηλώσει την συμφόρηση ως επίσης και την απώλεια πακέτου, οπότε συνυπάρχει με το TCP (Toho, Reno, NewReno, SACK), το TCP θα εξασφαλίζει το περισσότερο από το εύρος εκτός σε υψηλά BDP περιβάλλοντα όπου το TCP είναι αναποτελεσματικό. Υπάρχει μία εξαίρεση: αν η χωρητικότητα του συνδέσμου είναι τόσο μικρή όπου το UDT δεν μπορεί να έχει αρκετά δοκιμαστικά πακέτα για να ανιχνεύσει την αύξηση καθυστέρησης πριν η ουρά αναμονής υπερχειλίσει, η σχέση μεταξύ UDT και TCP ροής εξαρτάται στο RTT και στο μέγεθος της ουράς αναμονής. Η αναλογία μεταξύ UDT και TCP εύρους μοιράζεται την αύξηση όσο το εύρος μειώνεται και/ή το RTT αυξάνει. Επιπλέον, ο έλεγχος ροής επίσης παίζει ένα σημαντικό ρόλο για τη φιλικότητα του TCP.



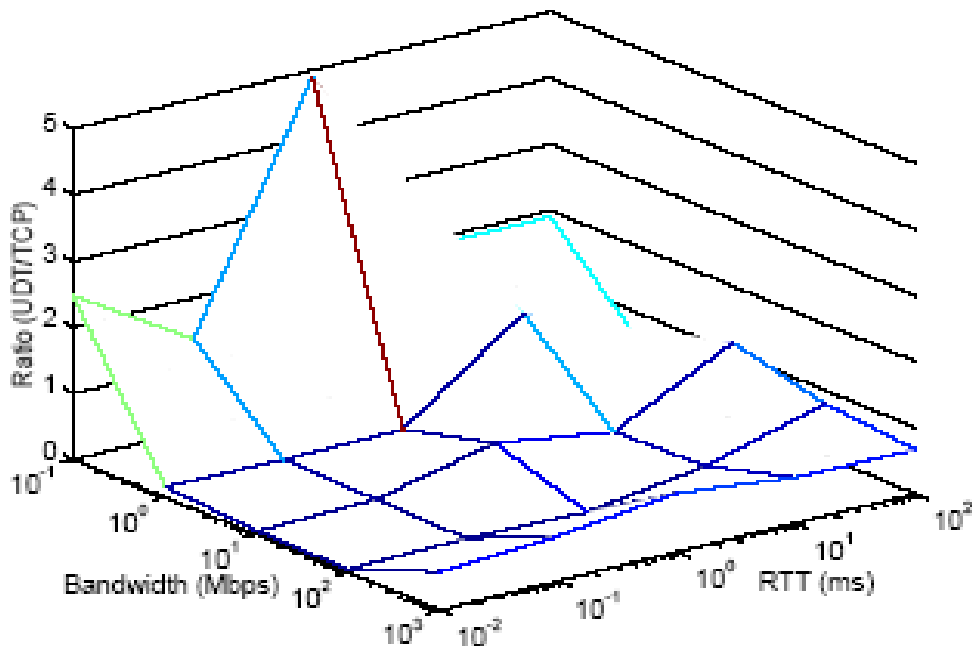
**Figure 6: RTT Independence.**

figure 6: ανεξαρτησία RTT

Η σχέση μεταξύ συνυπάρχοντος UDT και TCP είναι επίσης σχετική με το χρόνο συγχρονισμού. Αφού το TCP αλλάζει το μέγεθος του παραθύρου-θυρίδας σε κάθε RTT, ενώ το UDT χρησιμοποιεί σταθερό SYN χρόνο. Το TCP έχει πλεονεκτήματα έναντι του UDT μπορεί να βολιδοσκοπήσει το εύρος γρηγορότερα σε υψηλούς RTT συνδέσμους.

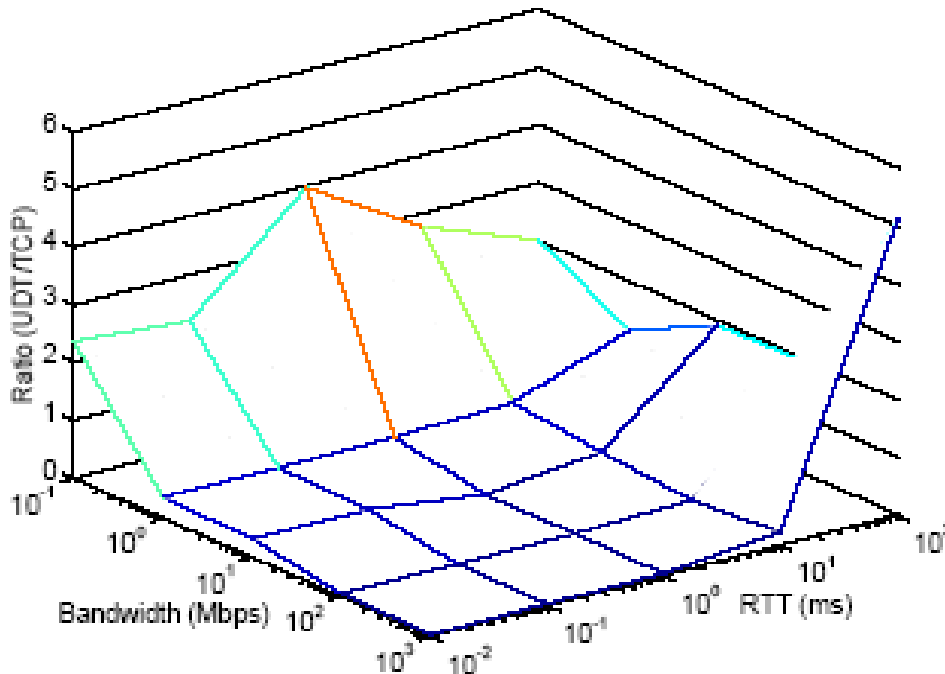
Το figure 7 και 8 δείχνουν την αναλογία μοιράσματος εύρους μεταξύ του UDT και του TCP κάτω από DropTail και RED διαχείριση ουράς αναμονής, αντίστοιχα. Η έκδοση του TCP είναι SACK με μέγεθος buffer(ενδιάμεσης μνήμης) που αρχίζει δραστικά σε κάθε περίπτωση BDP. Το μέγεθος της ουράς αναμονής στο DropTail είναι τοποθετημένο στο  $\max(2 * BDP, 10)$  ενώ στο RED είναι 20000.

Όταν το TCP και το UDT συνυπάρχουν, το γενικό εύρος εκμετάλλευσης είναι παρόμοιο όπως της μονής ροής εκμετάλλευσης UDT, δείχνοντας ότι το UDT μπορεί να εκμεταλλευτεί το εύρος εκεί που το TCP αποτυγχάνει να εξασφαλίσει, αλλά αφήνει αρκετό χώρο στο TCP να αυξήσει το παράθυρο-θυρίδα του.



**Figure 7: UDT/TCP share with DropTail queue.**  
figure 7. UDT/TCP μοίρασμα με ουρά αναμονής DropTail





**Figure 8: UDT/TCP share with RED queue.**

figure 8. UDT/TCP μοίρασμα με ουρά αναμονής RED

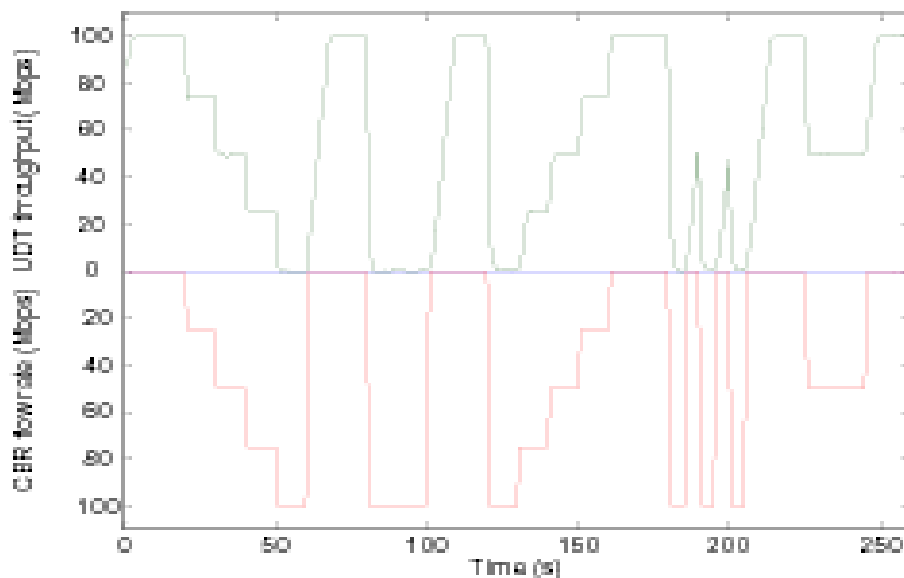
### 3. Σταθερότητα

Όπως αναφέρθηκε προηγουμένως, η ροή παραθύρου-θυρίδας όπου οριοθετεί τον αριθμό των μη βεβαιωμένης λήψης πακέτων αποφεύγει τον άνευ ορίων αριθμό των πακέτων να στέλνεται έξω πριν η πλευρά αποστολής λάβει μία πληροφορία ελέγχου ενδεικτικά της συμφόρησης. Εν τω μεταξύ, κατά τη διάρκεια υψηλής συμφόρησης ο αποστολέας ή θα μειώσει τον ρυθμό αποστολής κατά 1/9 τουλάχιστον μία φορά κάθε RTT ή θα σταματήσει να στέλνει νέα πακέτα μέχρι η συμφόρηση να εξαφανιστεί – η αποστολή παγώνει για SYN χρόνο για να καθαρίσει ο συμφορημένος σύνδεσμος.

Ακυρώνω-ανακαλώ εκείνο το διάστημα της αναφοράς απώλειας για το ίδιο πακέτο που αυξάνει κάθε φορά, ο αριθμός των resent πακέτων δεν θα προκαλέσουν συμφόρηση κατάπτωση. Από την άλλη, η πλευρά αποστολής δεδομένων δεν θα μπλοκαριστεί με τόσα πακέτα ελέγχου. Ο αριθμός των πακέτων ελέγχου είναι επίσης οριοθετημένος για να αποφευχθεί η συμφόρηση κατάπτωση από αυξημένη κίνηση ελέγχου. Εν τω μεταξύ, στο UDT δεν μειώνει για κανένα πακέτο το συμβάν απώλειας για να αποφύγει την άσκοπη πτώση στον ρυθμό αποστολής. Συνεχής μείωση του παραθύρου-θυρίδα αποστολής είναι ένα από τα

συνήθη προβλήματα για το TCP πάνω σε μεγάλη διαδρομή συνδέσμου, όπου μεγάλη ποσότητα πακέτων στέλνεται έξω πριν η πλευρά αποστολής δεδομένων μάθει για την ύπαρξη συμφόρησης.

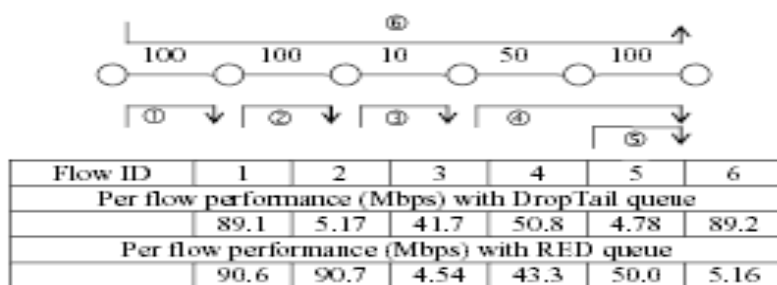
Το figure 9 δείχνει πως το δυνατό και το συγκλίνων UDT είναι ένα γρήγορης αλλαγής καταλλήλου εύρους συνδέσμου. Σε αυτή την κατάσταση, ένα σταθερός ρυθμός bit(CBR) ροής UDP θέτοντας σαν υπόβαθρο σε ένα 100Mbps σύνδεσμο με 10ms RTT και διαχείριση ουράς αναμονής DropTail. Μία μονή ροή UDT χρησιμοποιείται για να παρακολουθεί την σύγκλιση.



**Figure 9: UDT with rapid changing CBR background flow.**

figure 9. UDT με γρήγορη αλλαγή CBR υπόβαθρο ροής

Επίσης ελέγχουμε-δοκιμάζουμε το UDT σε μια πιο πολύπλοκη τοπολογία με πολλαπλά στενά περάσματα. Η τοπολογία και το αποτέλεσμα φαίνονται στο figure 10. Σε αυτό το figure, η τιμή κάτω από τη γραμμή είναι η χωρητικότητα αυτού του συνδέσμου (σε Mbps). Όλα τα RTT μεταξύ των δύο γειτονικών κόμβων είναι 10ms.



**Figure 10: UDT in multiple bottlenecks topology.**

figure 10. Το UDT σε πολλαπλή τοπολογία πολλών στενών περασμάτων

## 4. Επιρροή περί εκτίμησης λαθών

Σε πραγματικά περιβάλλοντα δικτύων, η άκρο σε άκρο χωρητικότητα συνδέσμου και η αύξηση καθυστέρησης μπορεί να εκτιμηθεί λανθασμένα.

Μία υποτίμηση ή υπερεκτίμηση χωρητικότητας συνδέσμου μπορεί να οδηγήσει σε πτώση απόδοσης. Από την άλλη, αν παράλληλη ροή παίρνει διαφορετική εκτιμητέα τιμή, η αδικία μπορεί να γίνει υπαρκτή, εξαρτάται από το πώς οι εκτιμητέες τιμές διαφέρουν από τις άλλες (μόνο δύο τιμές είναι διαφορετικές από την δέκατη δύναμη που μπορεί να οδηγήσει σε αδικία).

Ωστόσο, το UDT χρησιμοποιεί τη δικιά του κίνηση στην εξέταση πακέτων και η εξέταση είναι καταμήκος της ζωής της ροής, η μέθοδος του παραλήπτη βασιζόμενο σε ζεύγος πακέτων είναι υγιής και πρέπει να συγκλίνει σε σταθερή κατάσταση σε μεγάλη διάρκεια χρόνου. Η μόνη οριοθέτηση είναι ότι η μέθοδος δεν θα δουλέψει καλά σε πολλαπλό κανάλι συνδέσμου.

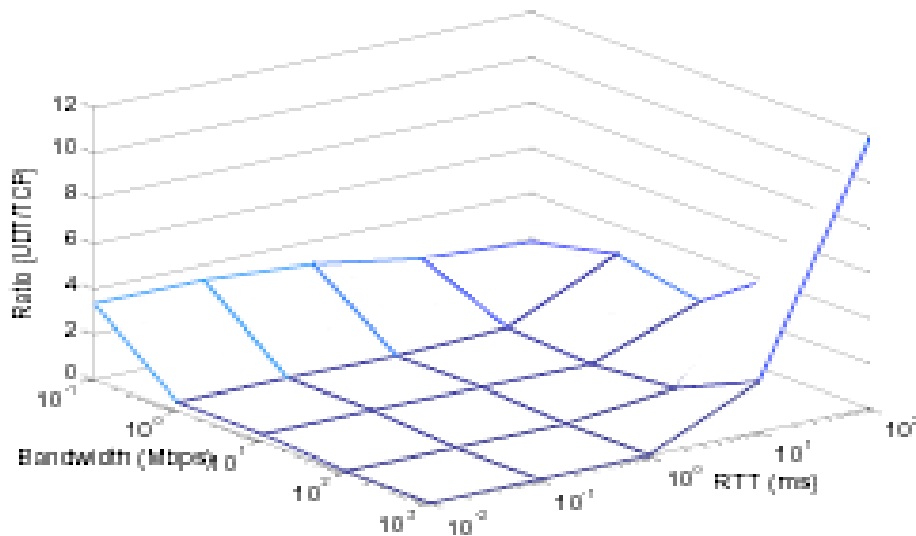
Λόγο του διακοπόμενου περιβάλλον συστήματος, το UDT μπορεί επίσης να αποτύχει να ανιχνεύσει την αύξηση καθυστέρησης (type 1 error), ή να δώσει μία λάθος προειδοποίηση αλλά η ουρά αναμονής δεν είναι πραγματικά γεμάτη (type 2 error). Οι τύποι 1 λαθών κάνουν το UDT λιγότερο TCP φιλικό, ενώ τύπου 2 λάθη μπορούν να οδηγήσουν σε πτώση απόδοσης.

Τύπου 2 λάθη μπορούν να γίνουν συμπτωματικά αφού ένα σταθερό σύστημα δεν θα προκαλεί συνέχεια αύξηση της RTT εκτίμησης σε μεγάλη διάρκεια χρόνου (αλλιώς η CPU πρέπει να χρησιμοποιηθεί και το σύστημα δεν είναι πλέον κατάλληλο για να τρέξει κάποιο πρόγραμμα). Επιπλέον, το UDT δεν επιτρέπει να στείλει και άλλα από μία προειδοποίηση καθυστέρησης σε  $2 * RTT$ .

Η συχνότητα του τύπου 1 λάθους μπορεί να είναι υψηλή αφού μικρή διαφορά σε RTT μπορεί να διαταραχθεί από τον θόρυβο συστήματος. Η πιθανότητα του τύπου 2 λάθους εξαρτάται στα συγκεκριμένα συστήματα.

Παρόλα αυτά, ακόμα και στη χειρότερη περίπτωση όπου το UDT αποτυγχάνει να ανιχνεύσει όλες τις τάσεις αύξησης καθυστέρησης, η δικαιοσύνη και η αποδοτικότητα δεν θα επηρεαστούν πολύ. Πρώτα απ'όλα, αφού το UDT μόνο μειώνεται  $1/9$  για 1 συμβάν απώλειας και δεν μπορεί κάθε συμβάν απώλειας να προκαλέσει μία μείωση, μπορεί ακόμα να φτάσει μία υψηλή απόδοση χωρίς προειδοποίηση καθυστέρησης. Δεύτερον, η δικαιοσύνη του εσωτερικού πρωτοκόλλου δεν θα επηρεαστεί. Τελικά, ο έλεγχος ροής βοηθάει να κάνουμε το UDT λιγότερο επιθετικό και φιλικό στο TCP. Το figure 11, δείχνει την προσομοίωση του UDT/TCP μοιράζοντας ρυθμό κλείνοντας τον

μηχανισμό προειδοποίησης καθυστέρησης. Η προσομοίωση χρησιμοποιεί την ουρά αναμονής RED.



**Figure 11: UDT/TCP share with RED queue when UDT delay warning is closed.**

figure 11. UDT/TCP μοιράζοντας ρυθμό κλείνοντας τον μηχανισμό προειδοποίησης καθυστέρησης

## 5. Επιρροή του RTT

Η επίδραση των μακριών RTT στην απόδοση του UDT είναι δύο πτυχές. Από τη μία, η αναφορά απώλειας ή η προειδοποίηση καθυστέρησης χρειάζεται μακρύτερο χρόνο για να ανατροφοδοτήσει στην πλευρά αποστολής δεδομένων σε μακρύτερο RTT σύνδεσμο. Κατά τη διάρκεια αυτή, η πλευρά αποστολής δεδομένων θα συνεχίσει να στέλνει πακέτα σε υψηλή ταχύτητα έτσι ώστε να είναι πιο επιθετικό. Από την άλλη, σαν συμπέρασμα αυτού του φαινομένου, μακρύ-RTT ροή στοχεύει να υποφέρει περισσότερη απώλεια περισσότερο συχνά από μικρό-RTT σύνδεσμο.

Το προηγούμενο χαρακτηριστικό επιδρά στην ισορροπία και κάνει το UDT να ευνοεί μακρύ-RTT λεπτοκαμωμένη ροή. Αυτό αποδεικνύεται στο figure 6.

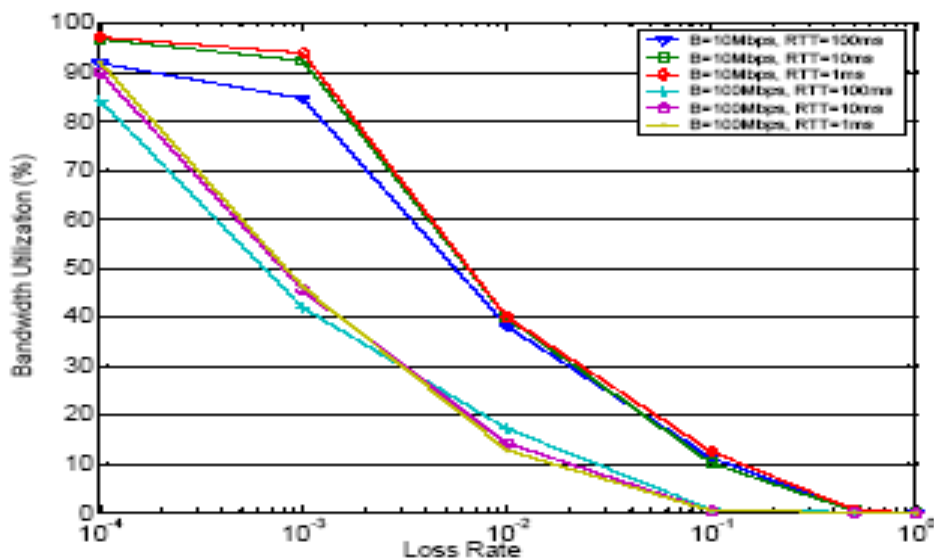
## 6. Επιρροή των χαλασμένων πακέτων

Χαλασμένα πακέτα αναγνωρίζεται σαν χαμένο στο UDT. Αν η πιθανότητα είναι υψηλή, μπορεί σημαντικά να επηρεάσει την απόδοση. Η απόκλιση της ανίχνευσης απώλειας στον αλγόριθμο παραλήπτη είναι για να περιμένει N πακέτα έτσι ώστε τα περισσότερα από τα χαλασμένα πακέτα να μπορούν να διορθωθούν. Η καθυστέρηση των N πακέτων της αναφοράς απώλειας φέρνει μικρή επίδραση στην απόδοση(αν και καλύτερο να ακούσει την συμφόρηση όσο το δυνατόν νωρίτερα).

## 7. Επιρροή του λάθους συνδέσμου

Το UDT ανιχνεύει όλες τις απώλειες σαν συμφόρηση, έτσι ώστε να μην είναι κατάλληλο για συνδέσμους με υψηλή πιθανότητα λάθους συνδέσμου όπως ασύρματα δίκτυα χωρίς καμία τροποποίηση. Αλλά ακόμα ελαφρύνει το πρόβλημα του TCP που προκαλείται από το λάθος συνδέσμου σε υψηλά BDP περιβάλλοντα, αφού το UDT δεν μειώνει τον ρυθμό αποστολής για κάθε αναφορά απώλειας, και αυξάνει αναλογικά με τη χωρητικότητα του συνδέσμου και ανεξάρτητα του RTT.

Το figure 12 δείχνει την αλλαγή εύρους εκμετάλλευσης μιας μονής UDT ροής όσο το λάθος συνδέσμου αυξάνει. Η χωρητικότητα συνδέσμου έχει σημαντική επίδραση προστίθοντας στην επιρροή του λάθους συνδέσμου, ενώ το RTT έχει μικρή επίδραση.

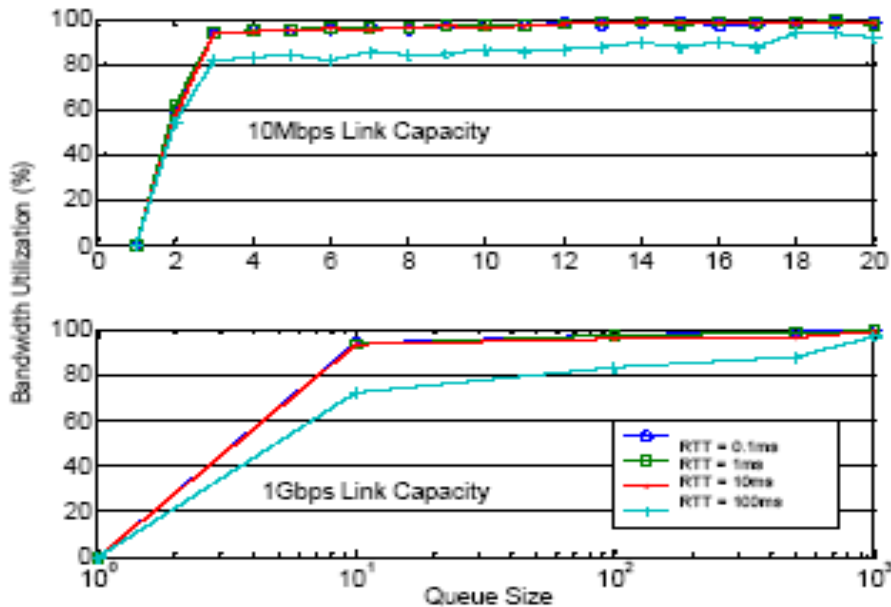


**Figure 12: Bandwidth utilization at different link error.**

figure 12. Εύρος εκμετάλλευσης με διαφορετικό λάθος συνδέσμου

## 8. Επιρροή του μεγέθους ουράς αναμονής διαχείρισης

Αφού το UDT στέλνει πακέτα σε κάθε εσωτερικό-πακέτου διάστημα, δεν χρειάζεται μεγάλο μέγεθος ουράς αναμονής για να υποστηρίξει υψηλή κίνηση. Οι κινήσεις είναι παρόμοιες στο DropTail και RED διαχειρίσεις ουρών αναμονής. Ωστόσο, χρησιμοποιώντας το RED να βοηθήσουμε να μειωθεί το μέσο μέγεθος ουράς αναμονής, με μικρό κόστος της πτώσης απόδοσης.



**Figure 13. Relationship between UDT performance and queue size (DropTail).**

figure 13. Σχέση μεταξύ UDT απόδοσης και μεγέθους ουράς αναμονής(DropTail).

Στο figure 13 δείχνει τη σχέση μεταξύ UDT απόδοσης και μεγέθους ουράς αναμονής. Οι χωρητικότητες συνδέσμου είναι 10Mbps(πάνω) και 1Gbps(κάτω), αντιστοίχως. Σε κάθε υπό-figure υπάρχουν 4 καμπύλες δείχνοντας τη μονή UDT απόδοση ροής με διαφορετικό RTT από 0.1ms σε 100ms. Και στα δύο σενάρια το UDT χρειάζεται πολύ μικρό μέγεθος ουράς αναμονής για να φτάσει υψηλό εύρος εκμετάλλευσης.

Εν τω μεταξύ, σχήματα διαχείρισης ουράς αναμονής έχουν μικρή επίδραση στο εσωτερικό πρωτοκόλλου δικαιοσύνη και σύγκληση.

## 9. Περιορισμοί

Συνεχόμενη μείωση του ρυθμού αποστολής μπορεί να συμβεί συχνά αν πολύ μεγάλες ποσότητες ροών μοιράζονται ένα υψηλή χωρητικότητας σύνδεσμο, όπου η αύξηση μπορεί να είναι πάρα πολύ υψηλή συγκρίνοντας με την ανά ροή κίνηση. Για παράδειγμα, όταν 10000 ροές μοιράζονται 1Gbps σύνδεσμο, η μέση κίνηση ανά ροή είναι περίπου 100Kbps, αλλά η αύξηση είναι 1 πακέτο(υποθέτοντας ότι το MTU είναι 1500bytes) ανά SYN χρόνο σύμφωνα με τον τύπο  $\max(10^{\lfloor \log_{10} B \rfloor} * SYN / 1000, 1 / MTU)$ . Αυτό μπορεί να προκαλέσει μεγάλες ταλαντώσεις, υψηλό ρυθμό απώλειας και λόγω του προηγούμενου πτώσεις στην απόδοση.

Αν και αυτή η αρνητική επίδραση μπορεί να ελαφρύνει από τον έλεγχο ροής, όπου παρουσιάζει ένα ανώτερο όριο αν ο ρυθμός αποστολής δεδομένων αυξάνει πάρα πολύ γρήγορα, αφού το μέγεθος παραθύρου-θυρίδας χρησιμοποιεί την ταχύτητα άφιξης πακέτου αποτέλεσμα από ένα προηγούμενο ρυθμό αποστολής. Είναι καλύτερο να χρησιμοποιήσουμε το διαθέσιμο εύρος αντί της χωρητικότητας συνδέσμου στον τύπο  $W = W * 0.875 + (RTT + SYN) * AS * 0.125$ , ο οποίος μπορεί να διαγράψει ολοκληρωτικά αυτόν τον περιορισμό. Ωστόσο, δεν έχουμε βρει ένα πρακτικά αποτελεσματική μέθοδο όπου μόνο μπορεί να χρησιμοποιήσει του UDT την κίνηση δεδομένων για να εκτιμήσει το διαθέσιμο εύρος.

Άλλος περιορισμός είναι του εσωτερικού-πρωτοκόλλου η δικαιοσύνη σε πολλαπλά στενά περάσματα συνδέσμου όπου μέγιστη-ελάχιστη δικαιοσύνη ίσως παραβιαστεί κάτω από μερικές περιστάσεις. Αυτό επίσης προκαλείται από την παράμετρο αύξησης, αφού διαφορετικές ροές σε πολλαπλά στενά περάσματα συνδέσμου ίσως έχουν διαφορετική άκρο με άκρο χωρητικότητα συνδέσμου. Αυτό το πρόβλημα μπορεί επίσης να λυθεί αν το διαθέσιμο εύρος σχέδιο εκτίμησης παρουσιαστεί.

Σημειώστε ότι το διαθέσιμο εύρος αλλάζει από στιγμή σε στιγμή, έτσι ο στόχος εδώ είναι για να βρούμε μία εκτιμητέα τιμή όπου είναι σύμφωνη με όλες τις παράλληλες ροές και περίπου εκφράζει το διαθέσιμο εύρος για την τωρινή στιγμή.

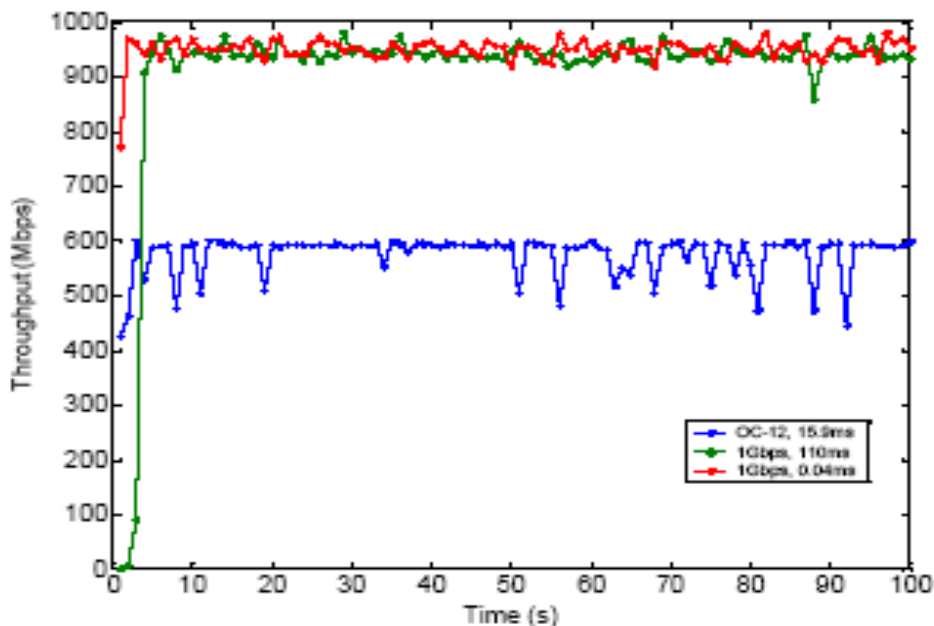
## **5. Υλοποίηση αποτελεσμάτων**

Το UDT πρωτόκολλο και ο προκάτοχός του το SABUL πρωτόκολλο έχουν υλοποιηθεί σε πολλές πλατφόρμες και χρησιμοποιούνται σε μερικές πραγματικές εφαρμογές. Επιδείξεις έχουν δειχθεί στα συνέδρια όπως το SuperComputing και το Igrid.

Σε αυτό το μέρος θα ανακαλύψουμε μερικά αποτελέσματα σε πραγματικά δίκτυα σαν ένα συμπλήρωμα στην προσομοίωση στο προηγούμενο μέρος, αφού το NS-2 περιβάλλον προσομοίωσης δεν μελετάει τις επιδράσεις των ικανοτήτων υπολογιστικών συστημάτων και εξωτερικών περιβαλλοντικών διακοπών. Ειδικά, θέλουμε να χρησιμοποιήσουμε αυτή την υλοποίηση αποτελεσμάτων για να εξετάσουμε την επίδραση του υπολογιστικού χρόνου στην επεξεργασία πακέτου και την επίδραση των εξωτερικών δυσπιστιών στον υπολογισμό της ταχύτητας άφιξης πακέτου και τη τάση αύξησης καθυστέρησης.

Τα πειράματα έγιναν πάνω σε 3 υψηλής ταχύτητας δοκιμαστικά δίκτυα: StarLight, SARA και Canarie.

Στο figure 14 φαίνεται η μονή UDT ροή που μπορεί να φτάσει γύρω στα 940Mbps πάνω σε 1Gbps σύνδεσμο με 40us μικρό RTT και 110ms μακρύ RTT. Μπορεί να φτάσει γύρω στα 580Mbps πάνω σε OC-12 σύνδεσμο με 15.9ms μεταξύ Canarie και StarLight.

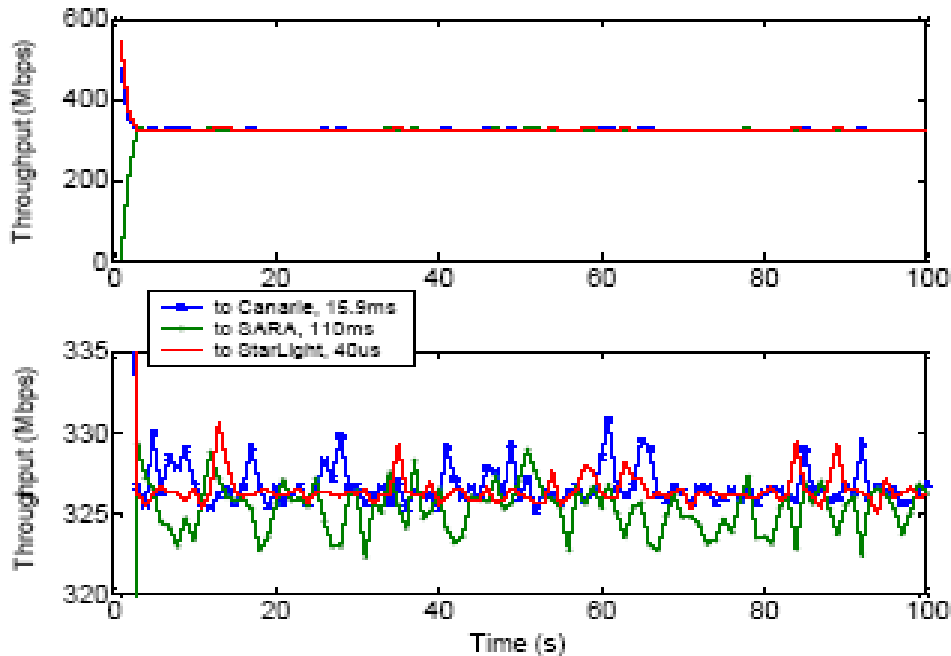


**Figure 14: UDT performance over real high speed testbeds.**

figure 14. Απόδοση του UDT πάνω σε υψηλής ταχύτητας δοκιμές



Ξεκινάμε τις 3 ροές στο figure 14 την ίδια στιγμή σε άλλο πείραμα για να εξετάσουμε τη δικαιοσύνη και την αποδοτικότητα των παράλληλων UDT ροών. Οι 3 ροές στέλνονται από το StarLight, μοιράζοντας 1Gbps στενό πέρασμα και πηγαίνουν στο Canarie, SARA και StartLight(local) αντιστοίχως. Το αποτέλεσμα φαίνεται στο figure 15.



**Figure 15: Three concurrent UDT flows across different testbeds sharing single 1Gbps bottleneck.**

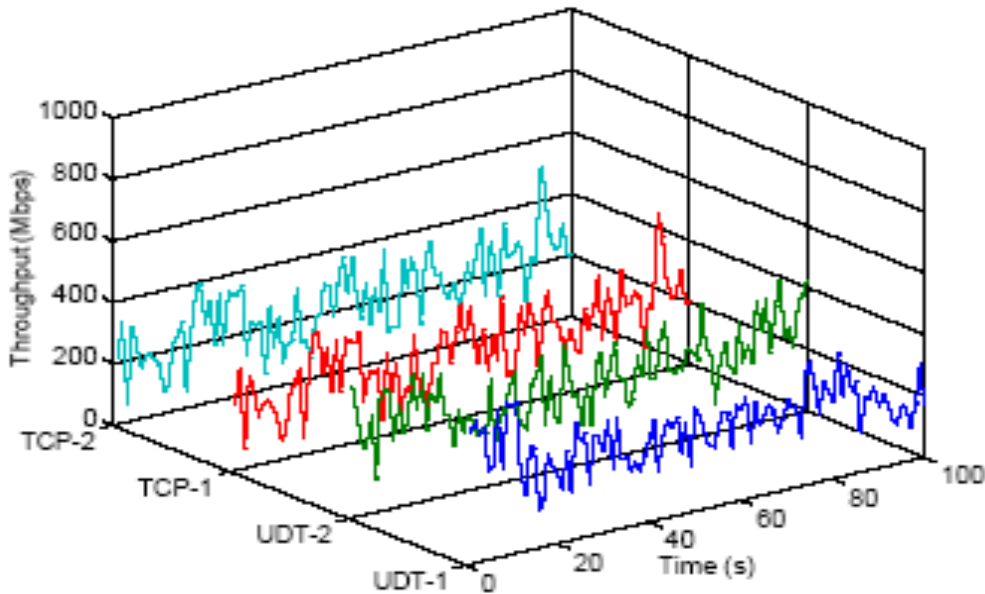
figure 15. 3 παράλληλες UDT ροές σε διαφορετικά δοκιμαστικά δίκτυα μοιράζοντας 1Gbps στενό πέρασμα

Όπως συζητήθηκε στο προηγούμενο μέρος, εκτιμητέα λάθη μπορούν να επηρεάσουν την ιδιότητα της φιλικότητας TCP. Στήνουμε μία ομάδα από πειράματα για να εξετάσουμε την φιλικότητα TCP σε πραγματικά δίκτυα. Η έκδοση του TCP στα πειράματα είναι το TCP SACK με ενδιάμεση μνήμη των λιγότερων BDP.

Στο figure 16 δείχνει το αποτέλεσμα των 2 TCP ροών και 2 UDT ροών παράλληλων στ StarLight τοπικό δίκτυο, με 1Gbps χωρητικότητα συνδέσμου και 40us RTT. Ροές TCP εκμεταλλεύονται λίγο υψηλότερο εύρος από τις UDT ροές.

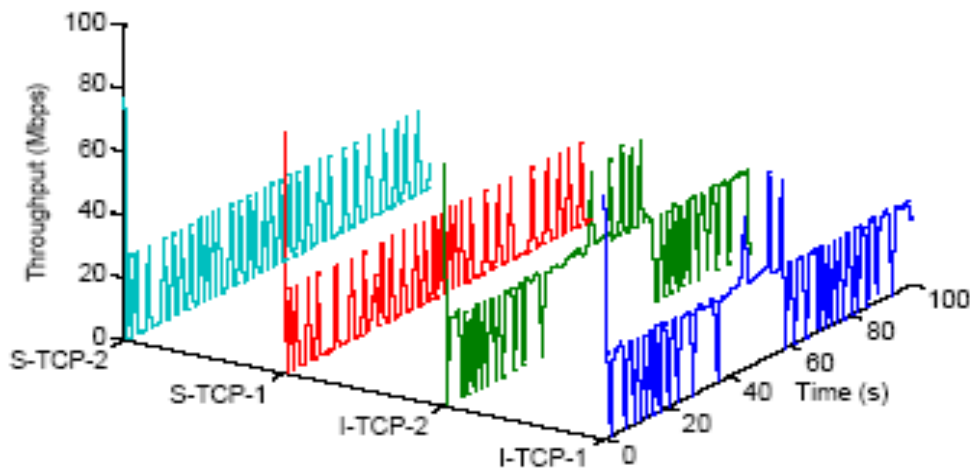
Ένα παρόμοιο πείραμα έγινε μεταξύ StarLight και SARA με 1Gbps εύρος και 110msRTT. Το TCP είναι αναποτελεσματικό σε αυτό το σύνδεσμο. Μία μονή ροή TCP μπορεί μόνο να εκμεταλλευτεί γύρω στα 20Mbps ακόμα και αν το μέγεθος ενδιάμεσης μνήμης είναι 12MB(περίπου ίσο του BDP). Το figure 17 δείχνει 2 παράλληλες αποδόσεις TCP ροών με

και χωρίς 2 συγκλίνουσες παράλληλες UDT ροές. Οι παράλληλες UDT ροές επιδρούν στην πτώση της TCP απόδοσης κατά περίπου 1/3. Σε αυτό το figure το I-TCP είναι TCP ροές χωρίς παράλληλες ροές UDT, ενώ S-TCP είναι με UDT ροές.



**Figure 16: TCP and UDT coexist in local high speed network.**

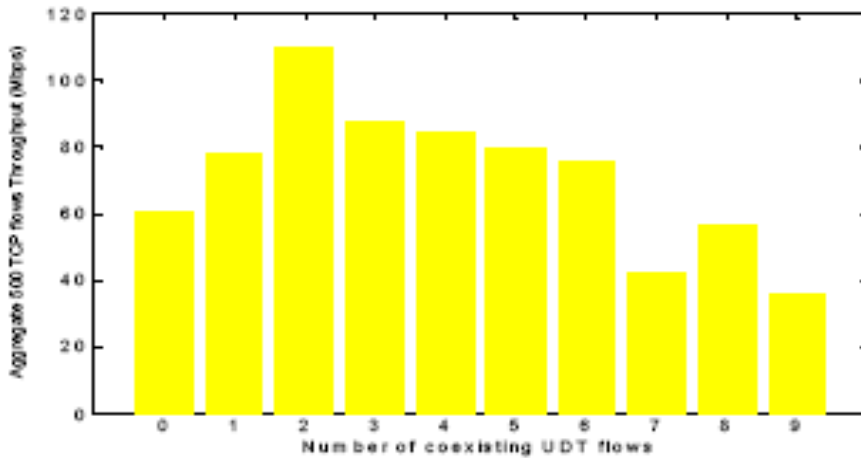
figure 16. TCP και UDT παράλληλα σε τοπικά υψηλής ταχύτητας δίκτυα



**Figure 17: TCP performance over high BDP network with and without concurrent UDT flows.**

figure 17. Απόδοση TCP πάνω σε υψηλά TCP BDP δίκτυα με και χωρίς παράλληλες ροές UDT.

Επίσης θέτουμε 500 1MB TCP ροές με διαφορετικό αριθμό από υπόβαθρες ροές UDT για να εξεταστούμε πως το UDT μπορεί να επιδράσει στην μικρή ζωή των TCP ροών σε τοπικές διαδρομές δικτύων(μεταξύ StarLight και SARA, 1Gbps εύρος, 110ms RTT). Το αποτέλεσμα φαίνεται στο figure 18.



**Figure 18: 500 TCP 1M streams with different number of background UDT flow.**

## **ΕΛΕΓΧΟΣ ΣΥΜΦΟΡΗΣΗΣ ΑΚΡΗ ΜΕ ΑΚΡΗ ΓΙΑ ΥΨΗΛΕΣ ΕΠΙΔΟΣΕΙΣ ΜΕΤΑΦΟΡΑΣ ΔΕΔΟΜΕΝΩΝ.**

### **Περίληψη**

Η μεγάλη ανάπτυξη στο χώρο των ηλεκτρονικών υπολογιστών έχει σαν αποτέλεσμα την αύξηση των απαιτήσεων σε διάφορους τομείς που σχετίζονται με αυτούς.

Ένας από τους σημαντικότερους τομείς ο οποίος βρίσκεται σε μεγάλη ανάπτυξη είναι αυτός της μεταφοράς δεδομένων μέσω υπολογιστικών δικτύων. Οι απαιτήσεις έχουν μεγαλώσει τα δεδομένα προς μεταφορά είναι πολύ μεγαλύτερα και ο χρόνος μεταφοράς τους είναι πολύτιμος. Η μεταφορά αυτή πρέπει να γίνει όσο το δυνατόν γρηγορότερα με τη μεγαλύτερη δυνατή απόδοση αξιόπιστα.

Ένα από τα κυριότερα προβλήματα στις μεγάλες επιδόσεις στον τομέα των ηλεκτρονικών υπολογιστών είναι η έλλειψη πρωτοκόλλου μεταφοράς για μεταφορά μεγάλου μεγέθους δεδομένων γρήγορα πάνω από υπολογιστικά δίκτυα. TCP, το de facto πρωτόκολλο μεταφοράς από το δια-δίκτυο αποτυγχάνει να χρησιμοποιήσει αποτελεσματικά το πλούσιο οπτικό εύρος ζώνης. Ακόμα έχει πρόβλημα 'αδικίας' σε διαφορετικά RTT που μπορεί να οδηγήσει σε πτώση της επίδοσης σε διάφορες εφαρμογές. Αυτή εδώ η εργασία αντιπροσωπεύει μια προσέγγιση από άκρη σε άκρη για να λυθούν αυτά τα προβλήματα της αποτελεσματικότητας και της "δικαιοσύνης", ειδικά σε υψηλού εύρους ζώνης καθυστέρησης δίκτυα. Αυτός ο αλγόριθμος ελέγχου συμφόρησης συνδυάζει μηχανισμούς ελέγχου βασισμένους στον ρυθμό (rate based) και στο "παράθυρο" (window based). Αναπτύσσει επιλεκτική αναγνώριση, σταθερό συγχρονισμό, μέτρηση του εύρους ζώνης, όπως και παρακολούθηση της καθυστέρησης των πακέτων που βοηθάει στον έλεγχο ρυθμού και παραθύρου. Τα αποτελέσματα της θεωρητικής ανάλυσης και της εξομοίωσης έδειξαν ότι ο αλγόριθμος επιτυγχάνει τους στόχους του την υψηλή απόδοση, την "δικαιοσύνη" του ενδιάμεσου πρωτοκόλλου, την φιλικότητα του TCP και την σταθερότητα.

## 1. Εισαγωγή

Η τεχνολογία των φωτονίων ανέβασε το εύρος ζώνης των δικτύων στα 10Gbps και κατέστησε δυνατή μία νέα γκάμα από καινούριες και ισχυρές εφαρμογές όπως υψηλής ανάλυσης ροή δεδομένων, πρόσβαση εξ' αποστάσεως σε συγκεκριμένες εφαρμογές και ειδίκευση στην εικονική πραγματικότητα σαν τηλέαπορρόφηση.

Ο τρόπος ροής των δεδομένων πάνω σε αυτά της υψηλής ταχύτητας ευρείας ζώνης δίκτυα, ή δίκτυα υψηλού εύρους ζώνης καθυστέρησης (BDP) είναι λίγο διαφορετικός από το γνωστό σε όλους μας INTERNET. Το επίσημο πρωτόκολλο μεταφοράς για το INTERNET το TCP χρησιμοποιεί το εύρος ζώνης με πολυπλεξία μεγάλων μεγεθών από παράλληλες ροές, από τις οποίες οι περισσότερες από αυτές είναι μικρής διάρκειας ζωής

Εν τούτοις στα υπολογιστικά δίκτυα είναι συχνή η περίπτωση που ένας μικρός αριθμός από μεγάλου μεγέθους πηγές μοιράζονται το άφθονο οπτικό εύρος ζώνης.

Ο αλγόριθμος ελέγχου του TCP που βασίζεται στο 'παράθυρο' AIMD (additive increase multiplicative decrease) υποφέρει περισσότερο από τυχαίες απώλειες καθώς το εύρος ζώνης καθυστέρησης (BDP) αυξάνεται. Ο μηχανισμός του για αποφυγή της συμφόρησης διαρκεί πάρα πολύ για να εξετάσει όλο το εύρος ζώνης. Η συνεχής απώλεια πακέτων είναι άλλη μία καταστροφή για το TCP στα δίκτυα υψηλού εύρους ζώνης καθυστέρησης (BDP), η οποία μπορεί να μειώσει το μέγεθος του παραθύρου σε πολύ μικρές τιμές και μετά να του πάρει ουσιαστικό χρόνο για να επανέλθει. Αυτά τα μειονεκτήματα έχουν αποδειχθεί σε πειράματα πάνω σε αληθινά δίκτυα.

Όταν οι ροές του TCP με διαφορετικό RTT μοιράζονται τον ίδιο σύνδεσμο, οι ροές με μικρότερο RTT θα καταλάβουν περισσότερο εύρος ζώνης συχνότητων από τις ροές με μεγαλύτερο RTT. Αυτό το πρόβλημα είναι ιδιαίτερα κρίσιμο σε πολλές εφαρμογές υψηλού επιπέδου. Παραδείγματος χάριν όταν ενώνονται οι ροές η απόδοση περιορίζεται από την πιο αργή ροή δεδομένων.

Οι ερευνητές δικτύων έχουν βρει πολλές λύσεις, συμπεριλαμβανομένης της αύξησης του TCP και των νέων πρωτοκόλλων στρώματος μεταφοράς. Οφειλόμενοι στο χρόνο και το οικονομικό κόστος της βελτίωσης και τυποποίησης του συστήματος, οι περισσότεροι από αυτούς δεν έχουν επεκταθεί ευρέως ακόμα και δεν είναι πιθανοί στο εγγύς μέλλον. Αν και μερικές εφαρμογές έχουν χρησιμοποιήσει το παράλληλο TCP για την υψηλής απόδοσης μεταφορά δεδομένων, δεν λύνει όλα τα προβλήματα του TCP και είναι πάρα πολύ επιθετικό παρά τις κανονικές ροές του TCP.

Μια νέα από άκρη σε άκρη προσπάθεια είναι απαραίτητη να εφευρεθεί για να υποστηρίξει εκείνες τις υψηλές εντατικές διανεμημένες εφαρμογές στοιχείων απόδοσης.

Ένας από άκρη σε άκρη αλγόριθμος ελέγχου συμμόρφωσης μπορεί να εφαρμοστεί στο στρώμα εφαρμογής(π.χ. χρησιμοποιώντας το UDP ως χαμηλότερη μεταφορά στρώματος) έτσι ώστε μπορεί να επεκταθεί χωρίς οποιαδήποτε τροποποίηση στο λειτουργικό σύστημα και στην υποδομή του υποσυστήματος. Επιπλέον η προσπάθεια μπορεί να χρησιμοποιηθεί στα δίκτυα άκρη σε άκρη ή επικάλυψης για να εξυπηρετήσουν υψηλής απόδοσης μεταφορά δεδομένων(στο στρώμα service).

Αυτή η άσκηση παρουσιάζει ένα αλγόριθμο ελέγχου συμμόρφωσης που έχει εφαρμοστεί σε πρωτοκόλλα υψηλής απόδοσης μεταφοράς που ονομάζονται UDT, ή UDP based data protocol, το οποίο είναι ένα αξιόπιστο πρωτόκολλο στρώματος εφαρμογής άνω των UDP. Στην υπόλοιπη εργασία αναφέρουμε τον αλγόριθμο ελέγχου συμμόρφωσης που περιγράφεται ως αλγόριθμος UDT.

Αυτός ο αλγόριθμος χρησιμοποιεί έλεγχο βασισμένο στο ρυθμό για το χρόνο των πακέτων και έναν βασισμένο στο παράθυρο για τον αριθμό των αναγνωρισμένων πακέτων. Ο αλγόριθμος ελέγχου ρυθμού είναι ο AIMD, όπου η παράμετρος αύξησης είναι σχετική, το κατ' εκτίμηση διαθέσιμο εύρος ζώνης συχνοτήτων και ο παράγοντας μείωσης είναι σταθερά. Το UDT μπορεί να χρησιμοποιήσει σχεδόν το πλήρες εύρος ζώνης συχνοτήτων ακόμη και σε συνδέσμους 1 Gbps και 100ms RTT. Εν το μεταξύ είναι ακόμα δίκαιο να δρουν παράλληλα και UDP και TCP ροές εάν υπάρχουν οπωσδήποτε.

Στο δεύτερο τμήμα θα εξηγήσουμε λεπτομερώς τις απαιτήσεις του αλγορίθμου και της λογικής σχεδίου. Ο αλγόριθμος περιγράφεται στην ενότητα 3 και μια θεωρητική ανάλυση με τα αποτελέσματα προσομοίωσης δίνεται στην ενότητα 4. Η ενότητα 5 αναλύει τους περιορισμούς του αλγορίθμου UDT και τις πιθανές βελτιώσεις στο μέλλον. Δίνουμε επίσης μερικά αποτελέσματα εφαρμογής στα πραγματικά υψηλά δίκτυα ταχύτητας στην ενότητα 6. Η ενότητα 7 εισάγει τη σχετική ερευνητική εργασία. Η εργασία ολοκληρώνεται στο τμήμα 8. Υπάρχει επίσης μία περιγραφή διαφόρων κρίσιμων ζητημάτων εφαρμογής στο παράρτημα.

## 2. Λογική σχεδίου

Η βασική απαίτηση έρχεται από την ανάγκη της γρήγορης μεταφοράς των μαζικών δεδομένων πέρα από τα υψηλά δίκτυα BDP. Έτσι η αποδοτικότητα είναι ένας από τους σημαντικότερους στόχους στον αλγόριθμο UDT. Πρέπει να εξετάσει το διαθέσιμο εύρος ζώνης συχνοτήτων γρήγορα και να ανακτήσει γρήγορα μετά από μια απώλεια. Και η ενιαία ροή UDT και οι πολυπλεγμένες παράλληλες ροές πρέπει να χρησιμοποιήσουν το εύρος ζώνης συχνοτήτων αποτελεσματικά. Ο στόχος αποδοτικότητας επίσης σημαίνει ότι δοθέντος του ρυθμού λαθών ανά bit των τρεχόντων συνδέσμων, το UDT μπορεί να χρησιμοποιήσει το περισσότερο από το εύρος ζώνης συχνοτήτων. Η δικαιοσύνη δια-πρωτοκόλλου είναι ένας άλλος κανόνας που πρέπει να υπακούσουν. Ιδιαίτερα, η δικαιοσύνη πρέπει να είναι ανεξάρτητη στην καθυστέρηση δικτύων ή RTT, δεδομένου ότι οποιοδήποτε είδος αδικίας μπορεί να προκαλέσει τις μειώσεις στο ρυθμό απόδοσης των εφαρμογών.

Πρέπει επίσης να είναι φιλικό, δεδομένου ότι σήμερα οι περισσότερες από τις ροές στο Δια-δίκτυο είναι TCP ροές. Εντούτοις, είναι δύσκολο να καθοριστεί ένας ποσοτικός TCP ως κανόνας φιλικής διάθεσης από τον οποίο το UDT και το TCP να μοιράζονται το εύρος ζώνης συχνοτήτων, επειδή το TCP μπορεί να είναι ανεπαρκές στην υψηλή σύνδεση BDP και δεν είναι ανεξάρτητο του RTT. Αν και έχει υπάρξει ένα μοντέλο για να ποσολογήσει στο TCP το ρυθμό απόδοσης, η χρησιμοποίηση αυτού του μοντέλου για να περιορίσει το ρυθμό απόδοσης του νέου πρωτοκόλλου είναι ανάρμοστη δεδομένου ότι έρχεται σε αντίθεση με τους στόχους αποδοτικότητας και δικαιοσύνης του δια-πρωτοκόλλου που συζητούνται ανωτέρω. Σύμφωνα με αυτό το υπόβαθρο, καθορίζουμε στο TCP ως κανόνα φιλικής διάθεσης το σχεδιασμό του αλγορίθμου UDT όπως κατωτέρω: Όταν ο ίδιος αριθμός ροών UDT και TCP μοιράζεται την ίδια συμφόρηση συνδέσμου, το UDT δεν μπορεί να καταλάβει περισσότερο εύρος ζώνης συχνοτήτων από το TCP, εκτός αν είναι επειδή το TCP είναι ανεπαρκές. Στην τελευταία περίπτωση, το UDT μπορεί να χρησιμοποιήσει το εύρος ζώνης συχνοτήτων που το TCP αποτυγχάνει να καταλάβει. Ο αλγόριθμος πρέπει να είναι σταθερός σε οποιαδήποτε σενάρια δικτύων και μπορεί να συγκλίνει γρήγορα στην ισορροπία αποδοτικότητας και δικαιοσύνης από οποιοδήποτε κατάσταση. Ιδιαίτερα, η κατάρρευση συμφόρησης πρέπει να αποφευχθεί. Επιπλέον, η μεταφορά δεδομένων απαιτείται να είναι αξιόπιστη.

Έτσι η αξιοπιστία είναι ακόμα ένας κανόνας για να υπακούσει.

Τελικά, το σχέδιο πρέπει να εξετάσει την αποδοτικότητα και τις δυσκολίες εφαρμογής. Η απασχόληση της CPU για τον υπολογισμό, της επικεφαλίδας πακέτων και των πακέτων ελέγχου πρέπει να είναι όσο το

δυνατόν λιγότερη δεδομένου ότι η υψηλής απόδοσης μεταφορά δεδομένων είναι επίσης μια εντατική εργασία για την ΚΜΕ, ο υπολογισμός αυτός μπορεί να περιορίσει τη ρυθμό απόδοσης ή να προκαλέσει την απώλεια πακέτων (επειδή η ΚΜΕ είναι επίσης απασχολημένη για να επεξεργαστεί τα εισερχόμενα πακέτα εγκαίρως). Βασισμένος στο ποσοστό συμφόρησης ο έλεγχος ρυθμού έχει θεωρηθεί ως καλύτερη μέθοδος από αυτόν του παραθύρου για υψηλή απόδοση. Εντούτοις, πρέπει να υπάρξει μια ευαισθησία για να περιορίσει τον αριθμό μη αναγνωρισμένων πακέτων για να μειώσει την απώλεια πακέτων πέρα από τα υψηλά δίκτυα BDP, όπου τα μεγάλα ποσά πακέτων έχουν σταλεί προτού να μάθει ο πομπός τη συμφόρηση. Χρησιμοποιούμε ένα δυναμικό παράθυρο σαν τιμή κατωφλίου. Το σχέδιο για αναβάθμιση του μεγέθους του παραθύρου καλείται έλεγχος ροής στο UDT.

Για να χρησιμοποιεί το εύρος ζώνης συχνοτήτων αποτελεσματικά, είναι απαραίτητο να είναι γνωστό το διαθέσιμο εύρος ζώνης, το οποίο έχει προταθεί για βελτιώσεις του TCP. Στο UDT χρησιμοποιούμε την εκτίμηση εύρους ζώνης συχνοτήτων για να αποφασίσουμε την παράμετρο αύξησης στον έλεγχο ρυθμού.

Για να φθάσει στην ανεξαρτησία RTT, είναι φυσικό να χρησιμοποιηθεί ένα σταθερό διάστημα ελέγχου αντί οποιωνδήποτε τιμών που αφορούν RTT, οι οποίες έχουν προταθεί. Επιπλέον, αυτό είναι επίσης μια αποτελεσματική μέθοδος για να φθάσει στη γρήγορη αποκατάσταση από απώλεια από μια μακριά σύνδεση RTT.

Δεδομένου ότι είναι δύσκολο να παρθεί το ίδιο εύρος ζώνης συχνοτήτων όπως παίρνει το TCP, επιλέγουμε να αφήσουμε το UDT να πάρει λιγότερο κατά τον ανταγωνισμό με τις συνυπάρχουσες ροές του TCP. Πάντα να προσαρμόζει το διαθέσιμο εύρος ζώνης συχνοτήτων, το UDT μπορεί να είναι πιο φιλικό από ότι η τυφλή αύξηση /μείωση του TCP.

Η χρησιμοποίηση της καθυστέρησης μπορεί να βοηθήσει να μειώσει την απώλεια και το επίμονο μέγεθος ουρών αναμονής που προκαλούνται από τον έλεγχο ρυθμού, έτσι αυξάνει την απόδοση και τη δικαιοσύνη. Εντούτοις, η καθυστέρηση μπορεί να εκτιμηθεί λάθος στα πραγματικά δίκτυα, έτσι είναι μόνο μια ενθαρρυντική μέθοδος στο UDT, δηλ., αυτό βοηθά να φθάσει στην υψηλότερη αποδοτικότητα και δικαιοσύνη, αλλά ακόμη και αν αποτυγχάνει, ο έλεγχος ρυθμού και ροής μπορεί ακόμα να εργαστεί καλά τουλάχιστον σε ένα αποδεκτό επίπεδο.

### **3 Αλγόριθμος**



## A. Η επισκόπηση

Ο βασικός μηχανισμός του αλγορίθμου είναι να ελέγξει το χρόνο διαπακέτων (έλεγχος ρυθμού) και τον αριθμό μη αναγνωρισμένων πακέτων (έλεγχος ροής). Ο έλεγχος ποσοστού προκαλείται από έναν χρονοδιακόπτη σε κάθε σταθερό διάστημα συγχρονισμού (SYN), ενώ ο έλεγχος ροής προκαλείται από κάθε θετική επιβεβαίωση λήψης σήματος (ask) που έρχεται.

Το σταθερό διάστημα SYN είναι 0,01 δευτερόλεπτα, το οποίο είναι μια ανταλλαγή μεταξύ της αποδοτικότητας, της δικαιοσύνης και της σταθερότητας. Η μικρότερη τιμή μπορεί να αυξήσει το ποσοστό αποστολής γρηγορότερα, αλλά είναι επιθετικότερη και μπορεί να είναι ασταθής στο πολύ μακρύ δίκτυο RTT. Δίνω μια πρακτική αξία RTT της πιο μακροχρόνιας σύνδεσης που μπορεί να έχουμε στο εγγύς μέλλον, που μπορεί να είναι διάφορα δευτερόλεπτα, το διάστημα 0.01s SYN είναι αποδεκτό για τη σταθερότητα.

Η επιβεβαίωση λήψης σήματος είναι εκλεκτική, η οποία στέλνεται ανά σταθερά διαστήματα όπως το χρόνο SYN αλλά ελέγχεται από έναν διαφορετικό χρονοδιακόπτη. Το UDT χρησιμοποιεί επίσης τη ρητή αρνητική ανατροφοδότηση, ή NAK, που στέλνεται μόλις ανιχνευθεί μια απώλεια πακέτων.

Το UDT προσπαθεί πάντα να συσκευάσει τα στοιχεία εφαρμογής (μέγιστη μονάδα μεταφοράς) στο μέγεθος MTU (συμπεριλαμβανομένης της επικεφαλίδας πακέτων) εάν είναι πιθανό, και δεν τα υπερβαίνει ποτέ. Σε αυτό το έγγραφο, χρησιμοποιούμε τα πακέτα και τον αριθμό πακέτων ανά το δευτερόλεπτο για να μετρήσουμε το μέγεθος παραθύρων και την ταχύτητα μεταφοράς, όπου ένα πακέτο είναι ίσο με το μέγεθος MTU, εκτός αν το στοιχείο ακολουθείται ρητά από μια μονάδα. Είναι εύκολο να μεταφραστεί ο αλγόριθμος UDT σε μια προσέγγιση βασισμένη στο byte. Εντούτοις, για την υψηλής απόδοσης μαζική μεταφορά στοιχείων, το σχέδιο βασισμένο στο πακέτο είναι αποδοτικότερο.

Για να βοηθήσουμε στην περιγραφή του αλγορίθμου, χρησιμοποιούμε τον πομπό και τον δέκτη για να αντιπροσωπεύσουμε τις οντότητες πρωτοκόλλου που στέλνουν τα στοιχεία και που λαμβάνουν τα στοιχεία, αντίστοιχα. Μόνο η μεταφορά στοιχείων στην ίδια κατεύθυνση συζητείται, η κατάσταση στην αντίθετη κατεύθυνση είναι συμμετρική.

## B. Ο έλεγχος ροής

Ο δέκτης υπολογίζει την ταχύτητα (AS) άφιξης πακέτων και την στέλνει πίσω μέσα στα πακέτα ask. Κατά λήψη του ask και η αξία του AS στο ask είναι μεγαλύτερο από 0, ο πομπός αναπροσαρμόζει το μέγεθος W παραθύρου του από:

$$W = W * a + AS * (RTT + SYN) * (1 - a). \quad (1)$$

Στο UDT το  $a$  είναι 0,875, το οποίο είναι η ανταλλαγή μεταξύ της επίδρασης ιστορίας και της αποδοτικότητας εφαρμογής. Η αξία χρησιμοποιείται σε όλες τις εξισώσεις EWMA (exponential weighted moving average) στο UDT. Η ταχύτητα άφιξης πακέτων υπολογίζεται με τον ακόλουθο αλγόριθμο: Ο δέκτης καταγράφει τα διαστήματα άφιξης των εισερχόμενων πακέτων δεδομένων. Μόλις ένα ask πρόκειται να σταλεί, βρίσκει το μεσαίο  $M$  των πιο πρόσφατων  $N0$  διαστημάτων ( $N0$  είναι 16 στην τρέχουσα προδιαγραφή και εφαρμογή), και αφαιρεί εκείνες τις τιμές που είναι είτε μεγαλύτερες από  $8M$  είτε λιγότερο από  $M/8$ . Εάν περισσότερες από τις μισές από τις τιμές έμειναν, ο δέκτης υπολογίζει τη μέση τιμή των διαστημάτων των υπόλοιπων πακέτων που έχουν φθάσει και μετά παίρνει την ταχύτητα άφιξης (σε αριθμό των πακέτων ανά δευτερόλεπτο). Διαφορετικά, επιστρέφεται 0.

### Γ. Ο έλεγχος ρυθμού

Κάθε χρόνο SYN, ο πομπός υπολογίζει τον κινούμενο μέσο όρο του ρυθμού απώλειας κατά τη διάρκεια του πιο πρόσφατου διαστήματος SYN από τον αριθμό απωλειών στα πακέτα NAK και το συνολικό αριθμό πακέτων που έχει στείλει. Εάν το ποσοστό απώλειας είναι λιγότερο από ένα μικρό κατώφλι (1%), ο αριθμός πακέτων που αυξάνονται ανά χρόνο SYN (INC) υπολογίζεται από τον ακόλουθο αλγόριθμο:

$$\text{if } (B \leq C) \\ \text{inc} = 1 / MTU \quad (2)$$

$$\text{else if } (C < D) \\ \text{inc} = \max(10^{\lceil \log_{10} B / 9 \times MTU \rceil} \times \beta / MTU, 1 / MTU) \quad (3)$$

$$\text{else} \\ \text{inc} = \max(10^{\lceil \log_{10} (B-C) \times MTU \rceil} \times \beta / MTU, 1 / MTU) \quad (4)$$

όπου το  $\beta$  είναι η υπολογισμένη από άκρη σε άκρη χωρητικότητα συνδέσμου, το  $C$  είναι ο τρέχον ρυθμός εκπομπής, και το  $D$  είναι ο ρυθμός αποστολής όταν τελευταία φορά εμφανίζεται μια μείωση.  $\beta$  είναι μια σταθερή εμπειρική αξία  $1,5 * 10^{-6}$ .

Ο χρόνος  $I$  Δία-πακέτων υπολογίζεται εκ νέου έπειτα σύμφωνα με:  
 $I = SYN / (SYN / I + INC)$ .

Ο ρυθμός αποστολής δεν θα αυξηθεί εάν το ποσοστό απώλειας είναι περισσότερο από το κατώφλι.

Αυτός ο αλγόριθμος προσπαθεί να υπολογίσει το διαθέσιμο εύρος ζώνης συχνοτήτων  $B$  (σε byte ανά δευτερόλεπτο) στο σύνδεσμο. Δεδομένου ότι το ποσοστό αποστολής μειώνεται από το  $1/9$  (αναφερθείτε στον αλγόριθμο μείωσης κατωτέρω), μετά από μια μείωση, το διαθέσιμο εύρος ζώνης συχνοτήτων υπολογίζεται ως  $B/9 * MTU$ . Διαφορετικά το  $B$  είναι  $(B - C) * MTU$ , θεωρώντας ότι όλο το υπόλοιπο εύρος ζώνης συχνοτήτων είναι διαθέσιμο. Ταξινομεί έπειτα το  $B$  σε διαφορετικά επίπεδα από δέκατης δύναμης  $P$ , και έπειτα το  $P$  χρησιμοποιείται για να αποφασίσει πόσα πακέτα μπορούν να αυξηθούν στον επόμενο χρόνο SYN. Εντούτοις, η ελάχιστη αύξηση είναι 1 byte, ή  $1/MTU$  πακέτα.

Αν και το  $B$  μπορεί να είναι μεγαλύτερο από το πραγματικό διαθέσιμο εύρος ζώνης συχνοτήτων, είναι σύμφωνο σε όλες τις παράλληλες ροές, το οποίο είναι πολύ σημαντικό για να φθάσεις στη δικαιοσύνη. Αυτό θα συζητηθεί περαιτέρω στο τμήμα 5.

Η από άκρη σε άκρη χωρητικότητα εξετάζεται από τα επιλεγμένα πακέτα δεδομένων του UDT. Κάθε  $N1$  πακέτα ( $N1$  είναι 16 στην τρέχουσα προδιαγραφή και εφαρμογή), ο πομπός δεν περιμένει στο χρόνο του επόμενου πακέτου αλλά στέλνει το επόμενο πακέτο αμέσως, έτσι ώστε τα δύο συνεχή πακέτα διαμορφώνουν ένα ζευγάρι πακέτων [ 1 ..8 ..9 ]. Ο δέκτης καταγράφει το χρόνο άφιξης μεταξύ αυτού του ζευγαριού πακέτων σε ένα window history. Μόλις ένα ask πρόκειται να σταλεί, υπολογίζει τη διάμεσο των διαστημάτων άφιξης των τελευταίων  $N2$  των ζευγαριών πακέτων  $MP$  ( $N2$  είναι 16 στην τρέχουσα προδιαγραφή και εφαρμογή στην αρχή μιας περιόδου UDT, όταν υπάρχουν λιγότερα από  $N2$  ζευγάρια πακέτων στο σύνολο, χρησιμοποιεί μόνο τις τρέχουσες διαθέσιμες τιμές), κατόπιν το νέο υπολογισμένο εύρος ζώνης συχνοτήτων  $NB$  ( $= 1/MP$ ), και το στέλνει στην όμοια πλευρά του μέσα στο πακέτο ask. Στην πλευρά πομπών, μετά από λήψη ενός ask, η υπολογισμένη χωρητικότητα συνδέσμου ενημερώνεται από:

$$B = B * a + NB * (1 - a). \quad (6)$$

Στο UDT, και η απώλεια πακέτων και η καθυστέρηση πακέτων χρησιμοποιούνται ως ενδείξεις της συμφόρησης. Το UDT μετρά την παραλλαγή RTT για να ελέγξει την τάση για αύξηση καθυστέρησης πακέτων. Ο πομπός στέλνει πίσω ACK2 για κάθε λαμβανόμενο ask, ως επιβεβαίωση λήψης σήματος στο ask. Σε κάθε ask ανατίθεται ένας μοναδικός αριθμός ακολουθίας ανεξάρτητος του αριθμού ακολουθίας δεδομένων, έτσι ώστε ο δέκτης να μπορεί να ταιριάξει το ask με το ACK2 ακριβώς. Όταν το ACK2 παραλαμβάνεται το νέο RTT υπολογίζεται από το χρόνο αφότου στέλνεται το ask μέχρι και όταν παραλαμβάνεται ACK2, το UDT χρησιμοποιεί τελευταίο  $N3$  ( $N3$  είναι 16 στην τρέχουσα προδιαγραφή και εφαρμογή), τιμές RTT καταγράφονται σε ένα

παράθυρο ιστορίας για να ελέγξουν την τάση για αύξηση καθυστέρησης με τη μέθοδο του PCT (pair wise comparison test) και του PDT (pair wise difference test) που περιγράφεται σε Path load. Εάν μια τάση αύξησης στις καθυστερήσεις πακέτων ανιχνευθεί, ένα μήνυμα προειδοποίησης στέλνεται στον πομπό αλλά λιγότερο από 1 πακέτο προειδοποίησης θα σταλεί σε 2 Rats.

Μόλις λάβει ο πομπός ένα μήνυμα προειδοποίησης καθυστέρησης, ενημερώνει το χρόνο δια-πακέτων από:

$$I = I * 1.125$$

Έν τω μεταξύ, καταγράφει τον μεγαλύτερο αριθμό ακολουθιών που έχει σταλεί (last\_deck\_esq.), και θέτει τον μετρητή ελάττωσης στο 1 και το κατώφλι  $e$  στο 4.

Όταν ο πομπός λαμβάνει ένα NAK, ελέγχει εάν ο τελευταίος χαμένος αριθμός ακολουθίας στο πακέτο είναι μεγαλύτερος από last\_deck\_esq. Εάν ο όρος είναι ικανοποιημένος, παίρνει την ίδια ενέργεια όπως όταν λαμβάνει την προειδοποίηση καθυστέρησης. Διαφορετικά αυξάνει το  $dc$  κατά 1, και παίρνει μόνο εάν  $dc == 2e$ . Μόλις ληφθεί, το  $e$  αυξάνεται κατά 1. Κάθε φορά η μεταβλητή του last\_deck\_esq. ενημερώνεται, η αποστολή δεδομένων είναι παγωμένη για το χρόνο SYN, δηλ., κανένα πακέτο δεν θα σταλεί κατά τη διάρκεια του επόμενου χρόνου SYN. Αυτό πρόκειται να βοηθήσει να καθαρίσει την κορεσμένη σειρά αναμονής.

Σημειώστε ότι ο τύπος (6) είναι ισοδύναμος για να μειώσει το ρυθμό αποστολής κατά  $1/9$ .

#### *Δ. Η αργή έναρξη*

Το μέγεθος παραθύρων αρχικοποιείται σε 2 και ενημερώνεται στον αριθμό αναγνωρισμένων πακέτων για κάθε πακέτο ack έως ότου λαμβάνει ο πομπός μια έκθεση απώλειας ή μια προειδοποίηση καθυστέρησης, όταν τελειώνει η αργή έναρξη. Ο χρόνος διά-πακέτων είναι 0 κατά τη διάρκεια της αργής φάσης έναρξης και ενημερωμένος στο διάστημα άφιξης πακέτων στο τέλος της φάσης. Στο UDT, η αργή έναρξη εμφανίζεται μόνο στην αρχή μιας περιόδου επικοινωνίας.

#### *Ε. Διαστήματα ανατροφοδότησης*

Υπάρχουν 3 είδη ανατροφοδοτήσεων που απαιτούνται στο UDT: ACK/ACK2, NAK, και μήνυμα προειδοποίησης καθυστέρησης. Το ack προκαλείται κάθε χρόνο SYN, αλλά μόνο όταν υπάρχουν νέα λαμβανόμενα πακέτα να αναγνωριστούν. Ο πομπός στέλνει ένα ACK2 πακέτο ως επιβεβαίωση λήψης σήματος στο ack στον δέκτη. Ο δέκτης μπορεί να μάθει ποιος είναι ο μεγαλύτερος αριθμός ακολουθίας που προηγείται σε αυτά που ο πομπός έχει αναγνωρίσει, έτσι μπορεί να αποφύγει να στείλει περιττά ACKs. Αφ' ετέρου, εάν ένα ack φέρνει τον ίδιο αριθμό ακολουθίας με αυτόν του προηγούμενου, το διάστημα μεταξύ αυτών των δύο ACKs πρέπει να είναι τουλάχιστον 2 RTTs.

Όσο υψηλότερος είναι ο ρυθμός αποστολής, τόσο λιγότερη είναι η αναλογία του κόστους εύρους ζώνης συχνοτήτων από το ACK/ACK2. Στη χειρότερη περίπτωση, όταν το ποσοστό αποστολής είναι πολύ χαμηλό (λιγότερο από 1 πακέτο στέλνεται σε 1 χρόνο SYN), υπάρχει ίσος αριθμός από ack και ACK2 πακέτα στον αριθμό πακέτων δεδομένων. Εάν η ταχύτητα μεταφοράς είναι 1Gbps και το MTU είναι 1500 Bytes, υπάρχει μόνο 1 ack και 1 ACK2 για κάθε 833 πακέτα δεδομένων.

NAK προκαλείται όταν λήγεται ένας χρονοδιακόπτης NAK ή όταν ανιχνεύεται η απώλεια κατά τη διάρκεια της λήψης στοιχείων. Στην προηγούμενη περίπτωση, η δομή στοιχείων που καταγράφει την απώλεια (herefrom την καλούμε τον κατάλογο απώλειας) ελέγχεται και εκείνα τα πακέτα οι των οποίων τελευταίοι χρόνοι εκθέσεων NAK είναι τουλάχιστον K RTTs πριν επιλέγονται και στέλνονται, όπου το K αρχικοποιείται σε 2 και αυξάνεται κατά 1 κάθε φορά που αναφέρεται το πακέτο. Στην τελευταία περίπτωση, μόνο η απώλεια που ανιχνεύεται από το τρέχον πακέτο στέλνεται. Οι πληροφορίες απώλειας συμπιέζονται σε NAK, και μόνο εκείνοι που μπορούν να περιληφθούν σε 1 πακέτο NAK στέλνονται, ενώ οι 2 άλλοι μπορούν να σταλούν σε ένα μελλοντικό πακέτο εάν είναι απαραίτητο.

NAK παράγεται μόνο εάν υπάρχει απώλεια να αναφέρει. Στη χειρότερη περίπτωση, υπάρχει ως μισός αριθμός του άμεσου NAKs ως πακέτων δεδομένων όταν κάθε άλλο πακέτο πήγε χαμένο κατά τη διάρκεια της συμφόρησης. Η δυνατότητα αυτής της κατάστασης είναι πολύ μικρή δεδομένου ότι η απώλεια είναι συχνά συνεχής. Το διάστημα NAK είναι το μεγαλύτερο μεταξύ RTT και του μέσου διαστήματος άφιξης δεδομένων από τον τελευταίο χρόνο SYN έτσι ώστε δεν παράγονται άλλα πακέτα ελέγχου από τα πακέτα δεδομένων. Εν τω μεταξύ, το διάστημα NAK για κάθε χαμένο πακέτο αυξάνει κάθε φορά έτσι ώστε ο πομπός δεν θα είναι πάρα πολύ απασχολημένος με την επεξεργασία των NAK για να στείλει οποιαδήποτε πακέτα.

Ένα μήνυμα προειδοποίησης στέλνεται μόλις ανιχνεύσει ο δέκτης μια αυξανόμενη τάση στο RTT, το οποίο ελέγχεται αφότου παραλαμβάνεται κάθε ACK2. Λιγότερη από μία προειδοποίηση καθυστέρησης

ανατροφοδοτείται σε 2 RTTs, δίνοντας αρκετό χρόνο για τον πομπό να καθαρίσει τη συμφόρηση.

## 4 Θεωρητική ανάλυση

### A. αποδοτικότητα

Στο UDT, ο ρυθμός απόδοσης αποφασίζεται κυρίως από τον έλεγχο ρυθμού στη σταθερή κατάσταση, ενώ ο έλεγχος ροής μπορεί να βοηθήσει να μειώσει την απώλεια κατά τη διάρκεια της συμφόρησης. Εάν η συμφόρηση μπορεί να καθαριστεί από μια ενιαία απελευθέρωση του ρυθμού αποστολής, τότε η χρησιμοποίηση εύρους ζώνης συχνοτήτων είναι τουλάχιστον  $94,4$  ( $= 2/(1 + 8/9)$ ) κατά τη διάρκεια της φάσης μετα-αργής-έναρξης. Αυτό είναι ικανοποιημένο εάν

1. Ο αριθμός γεγονότος απώλειας πακέτων δεν είναι αρκετά μεγάλος να προκαλέσει τις πολλαπλάσιες απελευθερώσεις ( $< 24$ ).
2. Το BDP δεν είναι τόσο μεγάλο ώστε μια ενιαία απελευθέρωση να μπορεί να εξουδετερώσει την επίδραση των αυξήσεων από την τελευταία απελευθέρωση.

Αυτοί οι όροι επηρεάζονται από το RTT, τη διαχείριση σειρών αναμονής και τον αριθμό ταυτόχρονων ροών.

Μεγάλο RTT είναι ο σημαντικός λόγος που προκαλεί τη συνεχή μείωση ποσοστού αποστολής, δεδομένου ότι χρειάζεται το μεγάλο χρόνο να ανατροφοδοτηθούν οι πληροφορίες απώλειας από τον δέκτη.

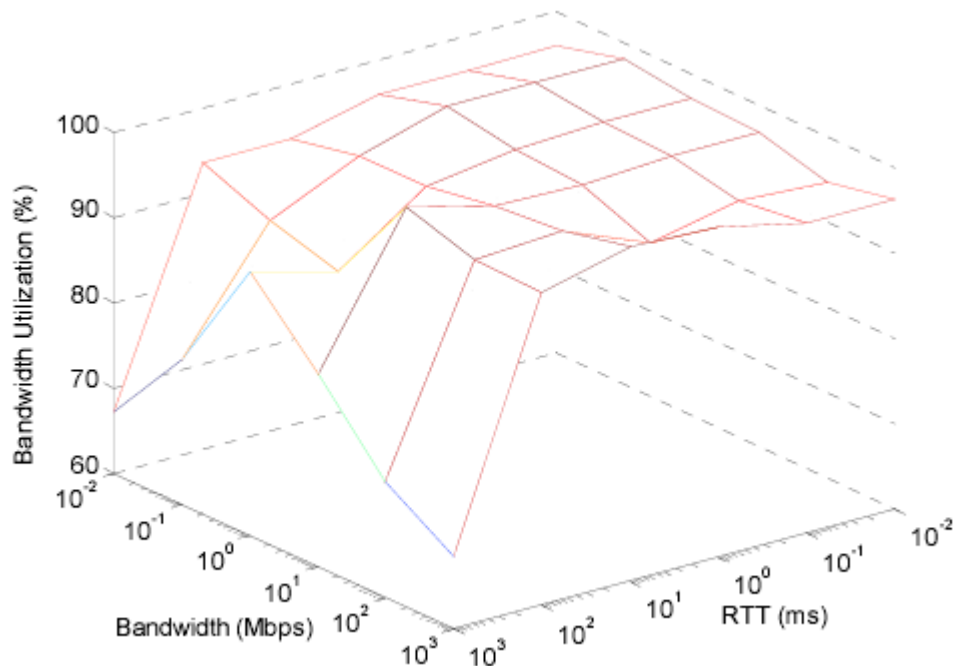
Η ενεργός διαχείριση σειρών αναμονής μπορεί να ρίξει ένα πακέτο πριν από τις υπερχειλίσεις σειρών αναμονής. Εντούτοις, αυτό μπορεί να μην έχει επιπτώσεις στο UDT δεδομένου ότι η ομαλή ροή UDT έχει τον καλό στατιστικό τρόπο για τις διαχειρίσεις σειρών αναμονής. Θα συζητήσουμε τον αντίκτυπο των σχεδίων μεγέθους και διαχείρισης σειρών αναμονής αργότερα σε αυτό το τμήμα.

Επιπλέον, καθώς ο αριθμός ταυτόχρονων ροών αυξάνεται, μειώνεται ο ρυθμός απόδοσης μιας ενιαίας ροής, αλλά η αύξηση ανά τη ροή κρατά αμετάβλητο έτσι όπως μια ενιαία ροή, δεδομένου ότι η διαθέσιμη εκτίμηση εύρους ζώνης συχνοτήτων γίνεται από το ρυθμό μιας ροής και τη χωρητικότητα του συνδέσμου.

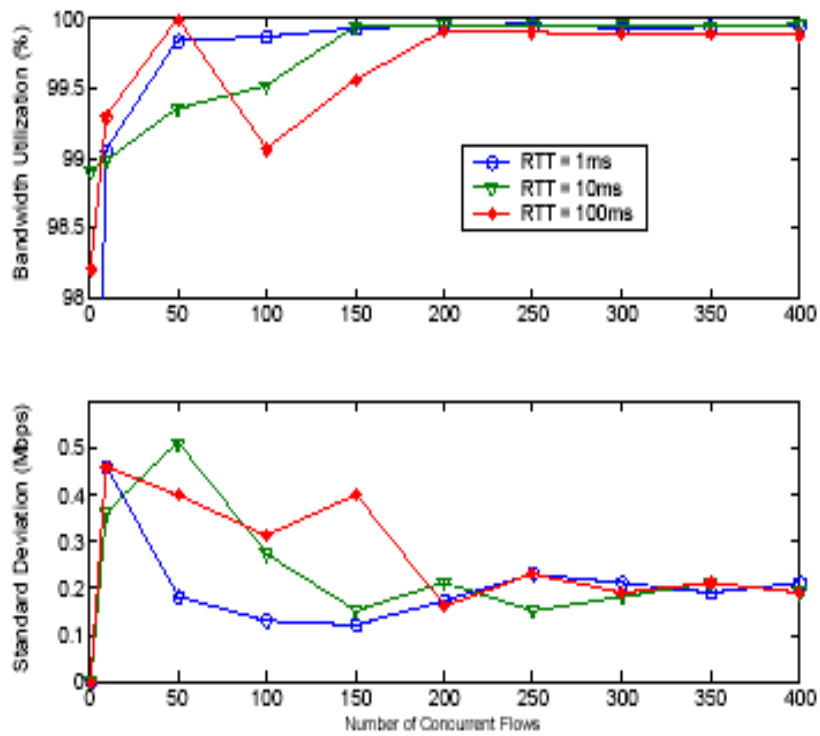
Η ανάλυση παραλείπει ανωτέρω την επίδραση της ανίχνευσης αυξανόμενης τάσης στις καθυστερήσεις πακέτων. Με τη χρησιμοποίηση της καθυστέρησης ως μια από τις ενδείξεις συμφόρησης, η απώλεια

μπορεί να μειωθεί περαιτέρω καθώς επίσης και μειώσεις του ρυθμού αποστολής.

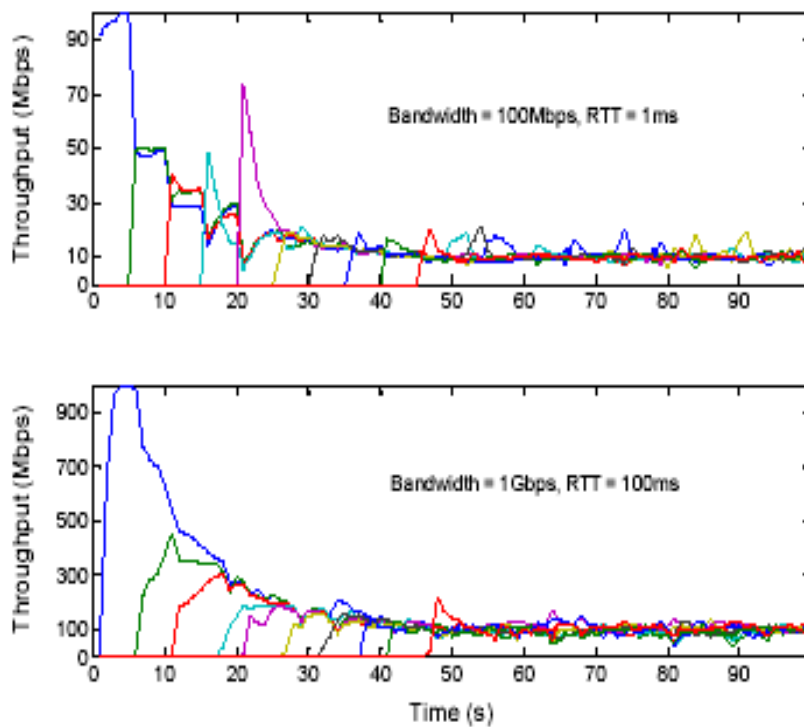
Η χρησιμοποίηση του εύρους ζώνης συχνοτήτων για την ενιαία ροή UDT με τα διαφορετικές άκρη με άκρη χωρητικότητες συνδέσμων και RTTs εμφανίζονται στη σημείωση Fig.1. Σημειώστε ότι η αργή απόδοση φάσης μετριέται στο γενικό σύνολο και είναι ο βασικός λόγος που προκαλεί τη χαμηλή χρησιμοποίηση εύρους ζώνης συχνοτήτων στα υψηλά σενάρια RTT, αφού προσομοίωση τρέχει μόνο 50 δευτερόλεπτα (που οφείλεται στη χρονική πολυπλοκότητα της προσομοίωσης). Όλες οι προσομοιώσεις σε αυτό το έγγραφο γίνονται NS-2 .



Οι πολυπλεγμένες παράλληλες ροές UDT πρέπει να παραγάγουν παρόμοια απόδοση σαν μία ενιαία ροή. Το σχέδιο 2 εμφανίζει την απόδοση πέρα από μια σύνδεση 100Mbps καθώς ο αριθμός ταυτόχρονων ροών αυξάνεται μέχρι 400. Ο αντίκτυπος του αριθμού παραλληλισμού θα συζητηθεί περαιτέρω αργότερα στο τμήμα 5.



Σχήμα 2. Σχέση μεταξύ UDT απόδοσης και αριθμού παραλληλισμού



Σχήμα 3. Δικαιοσύνη των UDT ροών



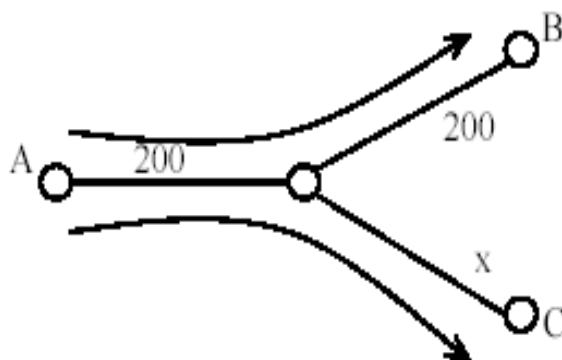
## B. Η δικαιοσύνη

Στον αλγόριθμο AIMD του ελέγχου ρυθμού, που εξουσιάζει την απόδοση του UDT, ο παράγοντας μείωσης καθορίζεται ως  $1/9$ . Η παράμετρος αύξησης, εντούτοις, είναι σχετική με το υπολογισμένο διαθέσιμο εύρος ζώνης συχνότητων. Για τους τύπους (2) και (3), είναι ίδιοι για όλες τις ροές. Για τον τύπο (4), οι ροές με τα μεγαλύτερα ποσοστά αποστολής είναι πιθανές να έχουν τις μικρότερες παραμέτρους αύξησης, αλλά δεν έχουν ποτέ μεγαλύτερες. Έτσι μπορούμε να ολοκληρώσουμε ότι μια ροή με το μικρότερο ρυθμό αποστολής έχει πάντα μια παράμετρο αύξησης περισσότερη από αυτή μιας ταυτόχρονης ροής με το μεγαλύτερο ρυθμό αποστολής, υποθέτοντας ότι υπάρχει μόνο μια δυσχέρεια στο δίκτυο και ότι όλες τις ροές υποφέρουν τη συμφόρηση συγχρόνως. Η ανάλυση εμφανίζει ότι το UDT μπορεί να συγκλίνει στην ισορροπία δικαιοσύνης στο ενιαίο δίκτυο δυσχερειών.

Το σχέδιο 3 εμφανίζει την δικαιοσύνη του δια-πρωτοκόλλου του UDT στα ενιαία δίκτυα δυσχερειών. Μπορεί να συγκλίνει στη δικαιοσύνη γρήγορα και να μείνει σταθερά στην κατάσταση ισορροπίας.

Στο σενάριο πολυπλεξιών, το UDT μπορεί να αποτύχει να ικανοποιήσει την ανώτατος-ελάχιστη δικαιοσύνη, δεδομένου ότι κάθε ροή μπορεί να έχει διαφορετική από άκρη σε άκρη χωρητικότητα συνδέσμου.

Συζητάμε τώρα το πρόβλημα χρησιμοποιώντας ένα απλό σενάριο 2-δυσχερειών/2-ροών (όπως φαίνεται στο σχέδιο 4). Στην τοπολογία 4-κόμβων του σχεδίου 4, αλλάζουμε το  $X$  μεταξύ 0 και 200 Mbps για να δούμε τη δικαιοσύνη μεταξύ των ροών AB και AC. Τα αποτελέσματα στον πίνακα 1 εμφανίζουν ότι η αδικία μπορεί να προκύψει όταν είναι το  $X$  μεταξύ 60 και 100 Mbps, πέφτοντας στην παρακείμενη χαμηλότερη κλάση 103. Για ένα πολύ μικρότερο  $X$  που πέφτει σε 102 ή σε χαμηλότερη κλάση, οι δύο ροές είναι δίκαιες.

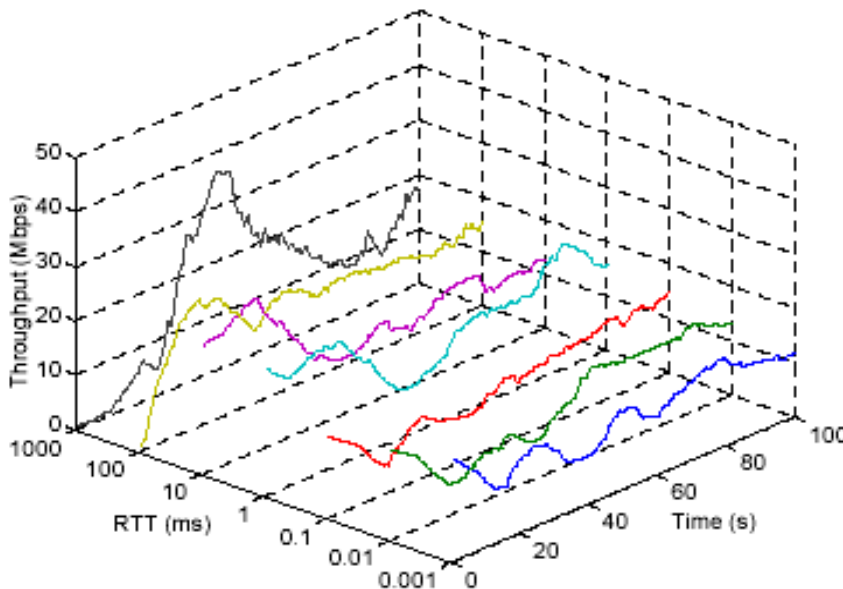


Σχήμα 4. Απόδοση του UDT σε δίκτυα τοπολογίας στενών περασμάτων.

TABLE 1  
UDT PERFORMANCE (IN MBPS) IN THE SCENARIO OF FIG. 4.

X	0.1	1	10	20	40	60
AB	198.5	189.2	180.1	170.9	152.6	139.7
AC	0.098	0.979	9.955	19.88	39.24	52.56
X	80	100	120	140	160	180
AB	137.7	105.4	100.8	101.3	100.5	100.3
AC	53.22	91.62	98.47	98.20	98.85	99.01

θα συζητήσουμε περαιτέρω το πρόβλημα δικαιοσύνης πολυπλεξιών στο τμήμα 5 και θα δώσουμε μια λεπτομερή ανάλυση στην προσθήκη A.



Σχήμα 5. Ανεξαρτησία RTT

Με τη χρησιμοποίηση του σταθερού χρόνου συγχρονισμού, το UDT είναι σχεδόν ανεξάρτητο του RTT. Η επίδραση μεγάλου RTT στην απόδοση του UDT έχει δύο πτυχές. Από τη μία, η αναφορά μίας απώλειας ή μίας καθυστέρησης χρειάζεται περισσότερο χρόνο να ανατροφοδοτηθεί πίσω στον εκπομπό σε μακρύτερο σύνδεσμο RTT. Κατά τη διάρκεια αυτής της περιόδου, ο πομπός θα συνεχίσει να στέλνει τα πακέτα με υψηλή ταχύτητα έτσι θα είναι πιο επιθετικότερο. Από την άλλη, συνεπεία αυτού του φαινομένου, μακρής ροή RTT τείνει να έχει την περισσότερη απώλεια ή καθυστέρηση και θα ρίξει το ρυθμό αποστολής πιο συχνά από τη σύνδεση μικρού RTT.

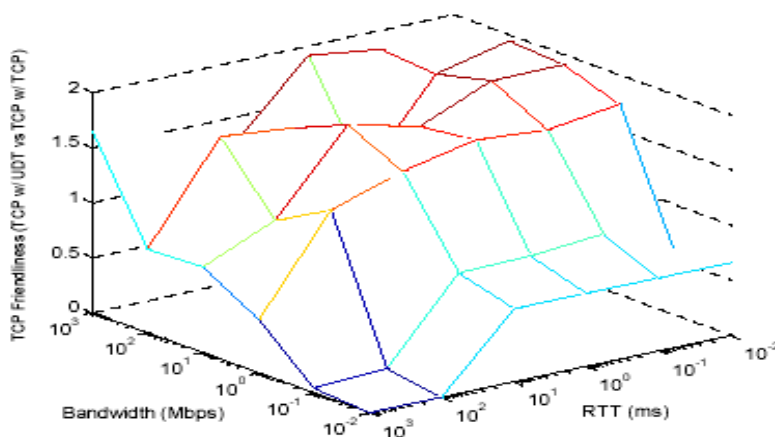
Ο έλεγχος ρυθμού και ροής στο UDT είναι αρκετά φιλικός σε άλλα πρωτόκολλα. Ο ρυθμός αυξάνει πιο αργά καθώς γίνεται υψηλότερος, έτσι επιτρέπει σε άλλες ροές με το χαμηλότερο ρυθμό να αυξηθούν. Σύμφωνα με του TCP τον αλγόριθμο αποφυγής συμφόρησης, το παράθυρο του μπορεί να αυξηθεί σχετικά γρηγορότερα. Ο έλεγχος ροής συμβάλλει επίσης στη φιλική διάθεση διά-πρωτοκόλλου, η οποία περιορίζει τον αριθμό πακέτων που στέλνονται, ακόμα κι αν ο ρυθμός έχει αυξηθεί υψηλότερα.

Ο σταθερός χρόνος συγχρονισμού διαδραματίζει έναν σημαντικό ρόλο εδώ. Οι συνδέσεις με RTTs μικρότερα από SYN μπορούν να ευνοήσουν το TCP, ενώ μακροχρόνια RTTs ευνοούν το UDT (η ανάκληση που σε μακρύ RTT συνδέεται με την τυχαία απώλεια, στο TCP είναι ανεπαρκής).

Επιπλέον, η χρησιμοποίηση της καθυστέρησης στο UDT αυξάνει επίσης τη φιλική διάθεσή του TCP, το οποίο υποχωρεί μόνο όταν εμφανίζεται απώλεια πακέτων.

Το σχέδιο 6 εμφανίζει τη σχέση μεταξύ ταυτόχρονων TCP και UDT ροών. Επιβεβαιώνει τον καθορισμό στο TCP της φιλικής διάθεσης της λογικής σχεδίου του UDT. Σημειώστε ότι η διαχείριση ουρών αναμονής και το μέγεθος των ουρών αναμονής μπορούν να έχουν επιπτώσεις στο μερίδιο εύρους ζώνης συχνότητων μεταξύ TCP και UDT, τα οποία θα συζητηθούν περαιτέρω αργότερα σε αυτό το τμήμα. Η διαμόρφωση μεγέθους ουρών αναμονής DropTail και ουρών αναμονής σε αυτήν την προσομοίωση είναι λογική για τα πρακτικά δίκτυα.

Από το σχέδιο 6 μπορούμε να βρούμε ότι το TCP μπορεί να υπερβεί το UDT περισσότερες φορές εκτός από εκείνες τις οποίες η χωρητικότητα συνδέσμου είναι πολύ μικρή (< 100kbps), επειδή πολλά σχέδια στον αλγόριθμο UDT θα χρειαστούν τον πολύ μακρύ χρόνο να αντιδράσουν δεδομένου ότι είναι βασισμένα στον αριθμό πακέτων.



Σχήμα 6. Κατανομή εύρους ζώνης του TCP με παράλληλες UDT και TCP ροές.

### *Γ. Η σταθερότητα*

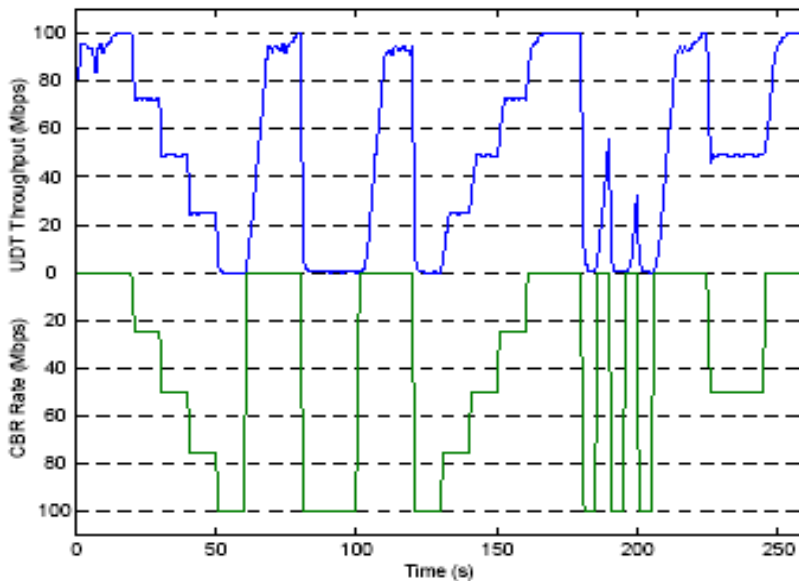
Έχει εμφανιστεί επάνω στο UDT έτσι ώστε να μπορεί να συγκλίνει στις ισορροπίες αποδοτικότητας και δικαιοσύνης στην κανονική κατάσταση. Συζητάμε τώρα την απόδοσή της στις ακραίες καταστάσεις της υψηλής συμφόρησης, του γρήγορου μεταβαλλόμενου διαθέσιμου εύρους ζώνης συχνότητων, και των σύνθετων τοπολογιών δικτύων.

Κατά τη διάρκεια της υψηλής συμφόρησης, το παράθυρο ροής αποφεύγει τον απεριόριστο αριθμό πακέτων που στέλνονται προτού να λάβει ο πομπός οποιοσδήποτε πληροφορίες ελέγχου που δείχνουν τη συμφόρηση. Ο πομπός θα σταματήσει σύμφωνα με τον έλεγχο ροής και θα μειώσει το ρυθμό αποστολής κατά  $1/9$  τουλάχιστον μιά φορά ανά RTT. Η αποστολή είναι παγωμένη για το χρόνο SYN για να βοηθήσει να καθαρίσει την κορεσμένη σύνδεση.

Επειδή το διάστημα της αναφοράς απώλειας για το ίδιο πακέτο αυξάνει κάθε φορά, τα πακέτα δεν θα προκαλέσουν την κατάρρευση συμφόρησης αφού η πολλαπλή αναμετάδοση χρειάζεται να περιμένει όλο και περισσότερο χρόνο. Επιπλέον, ο πομπός δεν θα εμποδιστεί με πάρα πολλά πακέτα ελέγχου. Ο αριθμός πακέτων ελέγχου είναι επίσης περιορισμένος για να αποφύγει την κατάρρευση από την συμφόρηση λόγω της αυξανόμενης κυκλοφορίας ελέγχου.

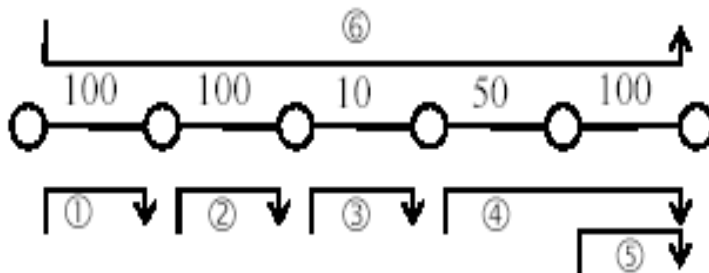
Εν τω μεταξύ, το UDT δεν μειώνεται για οποιοδήποτε γεγονός απώλειας πακέτων για να αποφύγει την περιττή απελευθέρωση. Οι συνεχείς μειώσεις παραθύρων αποστολής (ρυθμός) είναι ένα από τα κοινά προβλήματα για το TCP πέρα από τη μεγάλη απόσταση σύνδεση, όπου το μεγάλο ποσό πακέτων έχει σταλεί προτού να ξέρει ο πομπός την ύπαρξη της συμφόρησης.

Το σχέδιο 7 εμφανίζει πώς το γερό και συγκλίνον UDT είναι σε μία γρήγορη μεταβαλλόμενη χωρητικότητα σύνδεσης. Σε αυτήν την προσομοίωση, μία σταθερού ρυθμού ροή BIT(CBR) UDP ιδρύεται ως ροή υποβάθρου σε μια σύνδεση 100 Mbps με 10ms RTT. Μια ενιαία ροή UDT χρησιμοποιείται για να παρατηρήσει την σύγκλιση.



Σχήμα 7. Ισχυρότητα του UDT και σύγκλιση με γρήγορες αλλαγές στις ροές του υποβάθρου CBR.

Εξετάζουμε επίσης την απόδοση UDT σε μια πιο σύνθετη τοπολογία με τις πολλαπλές δυσχέρειες (το σχέδιο 8). Το UDT λειτουργεί αποδοτικά και δίκαια στο σύστημα(πίνακας 2).



Σχήμα 8. Απόδοση του UDT σε περίπλοκης τοπολογίας δίκτυα

UDT εκτέλεση σε (MBPS) του σχήματος 8

Τρεχων Id	1	2	3	4	5	6
	89.3	90.0	5.18	41.7	50.8	4.78

#### *Δ. Επιρροή των υπολογιζόμενων λαθών*

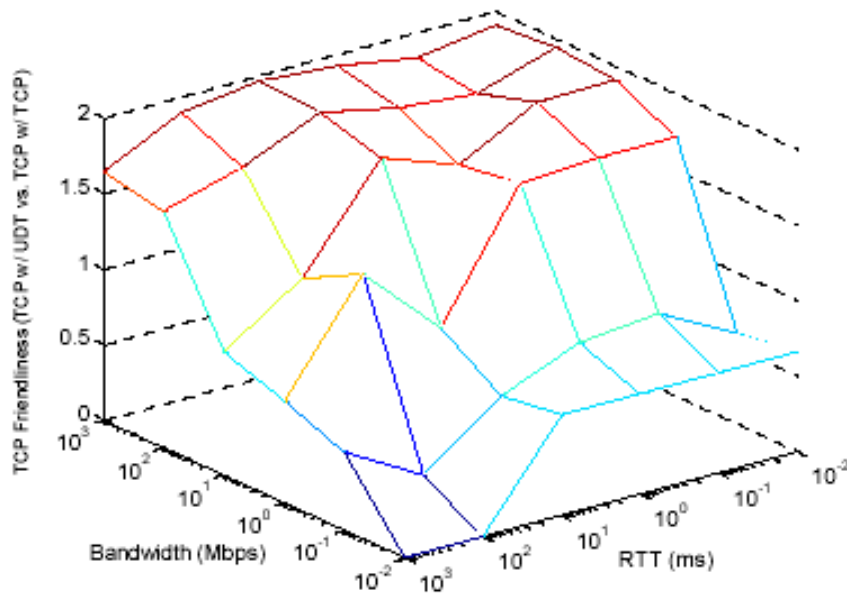
Ένας υποτιμημένος ή υπερτιμημένος σύνδεσμος χωρητικότητας ίσως οδηγηθεί σε πτώση απόδοσης . Από την άλλη πλευρά, εάν συνυπάρχουν ροές παίρνουν διαφορετικές υπολογιζόμενες τιμές, εμφανίζεται αδικία, εξαρτώμενη από το πώς οι υπολογιζόμενες τιμές διαφέρουν από κάθε άλλη (θυμήσου ότι στην διατύπωση (3) μόνο δυο τιμές είναι σε διαφορετικές κλάσεις της δέκατης δύναμης και ίσως οδηγηθούν σε αδικία εσωτερικού πρωτοκόλλου).

Ωστόσο,ο αποδέκτης βασισμένος στο γραφικό περίγραμμα των ζευγαριών πακέτων (RBPP) έχει δείξει να είναι ακλόνητος σε έναν απλό σύνδεσμο καναλιού. Σε γενικές γραμμές από την χρήση UDT η δική του κίνηση σαν probing πακέτα και το probing του χρόνου ζωής της ροής, η μέθοδος είναι ισχυρή και πρέπει να προκαλέσει σύγκληση σε μια σταθερή κατάσταση με μεγάλη διάρκεια χρόνου.

Σύμφωνα με τα τελευταία το περιβάλλον του συστήματος αλλάζει θέση, το UDT μπορεί επίσης να αποτύχει να αναζητήσει την καθυστέρηση αυξάνοντας την γενική κατεύθυνση (τύπος 1 λάθος),η να δώσει μια λάθος προειδοποίηση αλλά η σειρά είναι συνήθως μη αναπτυσσόμενη (τύπος 2 λάθος).Αμφότερα τα είδη των λαθών μπορούν να οδηγήσουν σε μείωση της λειτουργίας ,αν και τα λάθη του τύπου 1 μπορούν επίσης να φτιάξουν UDT λιγότερο φιλικό στο TCP .

Τα λάθη του τύπου 2 μπορούν να συμβούν μόνο περιστασιακά δεδομένου ότι ένα τέλος συστήματος δε θα προκαλεί συνεχώς αύξηση της τιμής RTT σε μεγάλη διάρκεια χρόνου(Στην χειρότερη περίπτωση η εκμετάλλευση της CPU θα αυξάνεται στο 100% αλλά ακόμη θα σταματήσει τελικά εκεί).Η συχνότητα του λάθους του τύπου 1 ίσως είναι υψηλότερη δεδομένου ότι μια μικρή διαφορά μπορεί να διακοπεί από τον θόρυβο του συστήματος.

Το ακριβές χάσιμο ρυθμού ίσως εξαρτάται από τα διαφορετικά δίκτυα,τις συνυπάρχοντες ροές στο σύστημα και την τελική κατάσταση του συστήματος. Εμείς εγκαθιστούμε μια UDT ροή ανάμεσα σε ένα ζεύγος από κόμβους με 1 Gbps εύρος ζώνης και 110ms RTT και αφήνουμε αυτό να τρέχει για 1000 δευτερόλεπτα. Ο ρυθμός λάθους του τύπου 1 είναι περίπου 1% αν και το λάθος του τύπου 2 ποτέ δεν συνέβη.

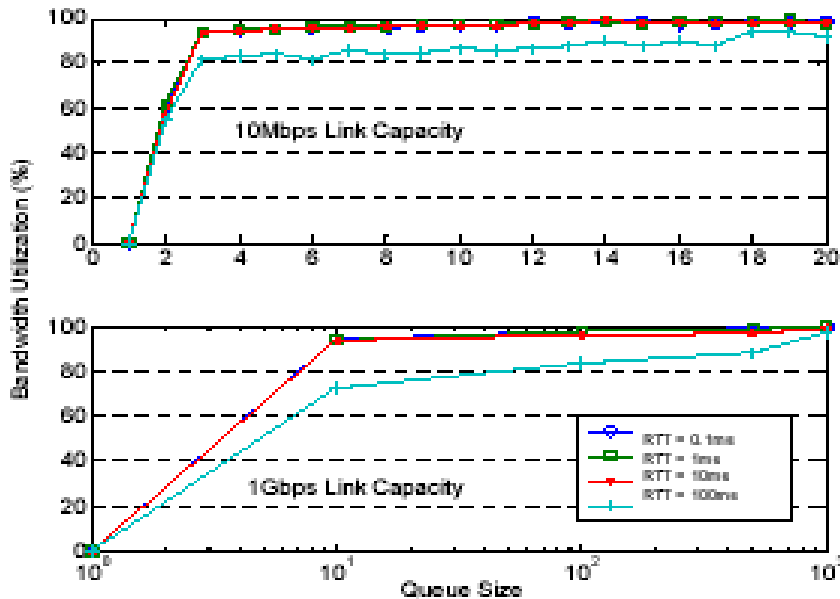


Σχήμα 9. Το TCP εύρος ζώνης κατανέμεται με ταυτόχρονη δράση UDT (με την αυξανόμενη καθυστέρηση ανίχνευση εκτός λειτουργίας) ροή πέρα από αυτήν με μια άλλη ροή TCP. Η έκδοση TCP είναι SACK και το μέγιστο παράθυρο είναι τουλάχιστον BDP. Διαχείριση ουρών αναμονής DropTail, χρησιμοποιείται. Το μέγεθος ουρών αναμονής γίνεται  $\max(\text{BDP}, 10)$ . Η παραγωγή TCP είναι υπολογισμένη κατά μέση τιμή στην προσομοίωση 2-TCP (δεδομένου ότι δεν είναι ισοδύναμοι).

Εντούτοις, ακόμη και στη χειρότερη περίπτωση εκείνο το UDT αποτυγχάνει να ανιχνεύσει όλες τις αυξανόμενες τάσεις καθυστέρησης, η δικαιοσύνη και η αποδοτικότητα επηρεάζεται πολύ. Καταρχήν, από UDT μόνο μείωση  $1/9$  για ένα γεγονός απώλειας και όχι κάθε γεγονός απώλειας μπορούν να προκαλέσουν μια πτώση, μπορεί ακόμα να φτάσει υψηλή απόδοση χωρίς προειδοποίηση καθυστέρησης. Δεύτερον, η δικαιοσύνη δια-πρωτοκόλλου δεν θα επηρεαστεί. Τέλος, το ποσοστό και ο έλεγχος ροής είναι αρκετά φιλικά στις τυφλές αυξήσεις του TCP και ο σταθερός χρόνος συγχρονισμού κάνει τις διευκολύνσεις RTT TCP περισσότερο από UDT. Το σχέδιο 9 παρουσιάζει την προσομοίωση του TCP χαρακτηριστικό γνώρισμα φιλικής διάθεσης μετά από απενεργοποίηση του μηχανισμού προειδοποίησης καθυστέρησης του UDT.

### E. Αντίκτυπος του μεγέθους και της διαχείρισης ουρών αναμονής

Δεδομένου ότι UDT στέλνει τα πακέτα σε κάθε διάστημα διά-πακέτων δεν χρειάζεται μεγάλο μέγεθος ουρών αναμονής για να υποστηρίξει υψηλή απόδοση. Η σχέση μεταξύ της απόδοσης UDT και του μεγέθους σειρών αναμονής φαίνεται στο σχέδιο 10.

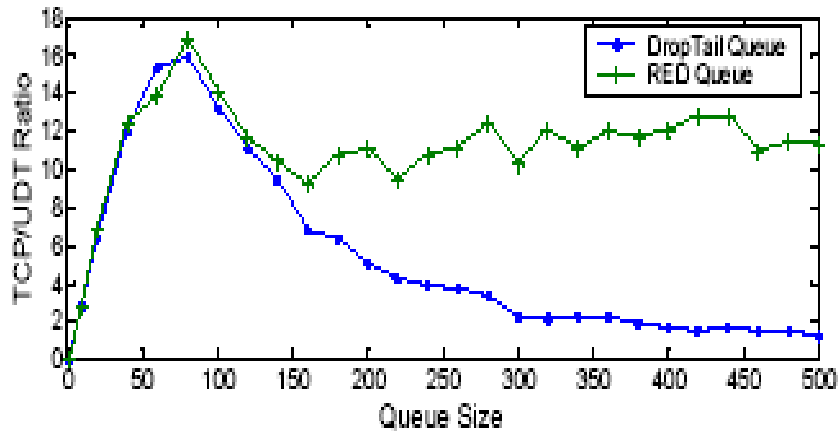


Σχήμα 10. Σχέση, μεταξύ της απόδοσης UDT και του μεγέθους ουρών αναμονής (DropTail). Οι δυο εξομοιωτές χρησιμοποιούν σύνδεσμο χωρητικότητας, 10Mbps (ανωτέρω) και 1Gbps (κατωτέρω) αντίστοιχα, το RTT ποικίλλει μεταξύ 0.1ms, 1ms, 10ms, και 100ms σε κάθε μια εξομοίωση.

Εντούτοις, το μέγεθος ουρών αναμονής έχει πολλές επιπτώσεις στην απόδοση του TCP και ως εκ τούτου έχει επιπτώσεις στην TCP φιλική διάθεση του UDT. Εάν το μέγεθος ουρών αναμονής είναι πάρα πολύ μικρό για να αποθηκεύσει τη ροή της έκρηξης δεδομένων του TCP, θα μειώσει την απόδοση του TCP, επομένως, η συνύπαρξη με UDT μπορεί να φτάσει σε υψηλότερη απόδοση.

Εν τω μεταξύ, η RED σειρά αναμονής ευνοεί το TCP δεδομένου ότι δεν έχει κλίση στην έκρηξη ροών όπως κάνει το DropTail. Η κλίση γίνεται μεγαλύτερη καθώς αυξάνεται το μέγεθος της ουράς αναμονής, δεδομένου ότι ο αριθμός των συνεχόμενων πτώσεων πακέτων γίνεται μεγαλύτερη. Με DropTail σειρά αναμονής, η αναλογία μεριδίου εύρους ζώνης μεταξύ του TCP και UDT μειώνεται καθώς αυξάνεται το μέγεθος των ουρών αναμονής, ενώ το μέγεθος ουρών αναμονής δεν έχει σχεδόν καμία επίδραση στην RED ουρά αναμονής.



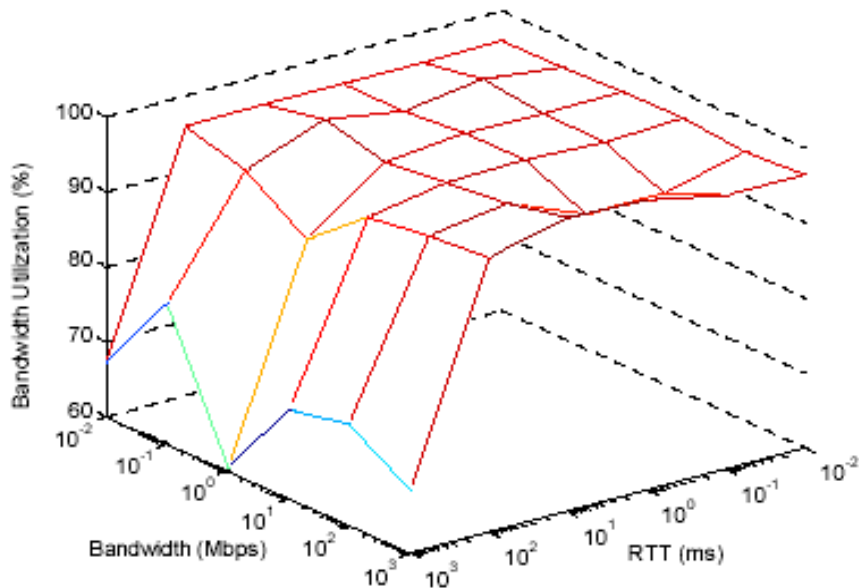


Σχήμα 11. Σχέση μεταξύ της φιλικής διάθεσης TCP και του μεγέθους ουρών αναμονής κάτω από διαφορετική διαχείριση σχεδίων ουρών αναμονής. Η προσομοίωση χρησιμοποιεί 100Mbps σύνδεση εύρους ζώνης, με 10 ms RTT. Το μέγεθος πακέτων είναι 1500 byte και το BDP είναι περίπου 83 πακέτα. Οι RED παράμετροι τίθενται εξαρχής από NS2. Το μέγεθος ουρών αναμονής είναι στα πακέτα.

Η συζήτηση ανωτέρω αποδεικνύει την προσομοίωση του σχεδίου 11. Οι ροές TCP λαμβάνουν την υψηλότερη απόδοση στο μέγεθος των ουρών αναμονής του BDP. Μετά από αυτό το σημείο η ροή με DropTail μειώνεται καθώς αυξάνεται το μέγεθος των ουρών αναμονής αλλά η ροή με RED κρατιέται αμετάβλητη μετά από έναν μικρή πτώση.

Χάρη στον έλεγχο ρυθμού και στην χαμηλή απαίτηση μεγέθους των ουρών αναμονής, το σχέδιο διαχείρισης των ουρών δεν επηρεάζει αρκετά το UDT. Οι αποδόσεις είναι παρόμοιες και κάτω απ τις DropTail και RED ουρές διαχείρισης. Εντούτοις, η χρησιμοποίηση του RED μπορεί να βοηθήσει, για να μειωθεί το μέσο μέγεθος ουρών αναμονής, με ασήμαντο κόστος της πώσης απόδοσης. Η απόδοση UDT με την RED διαχείριση σειρών αναμονής είναι παρουσιασμένες στο σχέδιο 12.

Δεδομένου ότι το RED δεν διαχειρίζεται ανά δυναμική ροή, η δικαιοσύνη διανομής του RTT πρέπει να είναι παρόμοια όπως και με την διαχείριση DropTail. Συζήτηση για UDT πάνω από άλλη διαχείριση ουρών αναμονής είναι πέρα από το πεδίο αυτού του εγγράφου.



Σχήμα 12. Χρησιμοποίηση εύρους ζώνης με RED διαχείριση ουρών αναμονής. Το μέγεθος ουρών αναμονής είναι  $\max(\text{BDP}, 10)$  και οι RED παράμετροι τίθενται από τον προεπιλεγμένο αλγόριθμο NS2. Η προσομοίωση τρέχει για 50 δεύτερα περιλαμβάνοντας την φάση έναρξης.

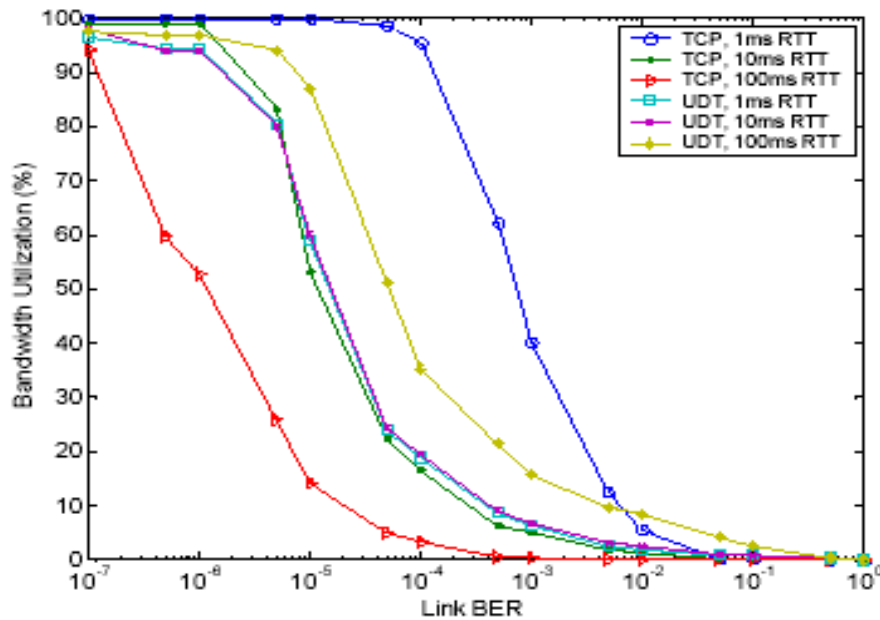
### ΣΤ. Αντίκτυπος του λάθους συνδέσεων

Το UDT θεωρεί όλη την απώλεια ως συμφόρηση έτσι δεν είναι κατάλληλο για συνδέσεις με υψηλή πιθανότητα λάθους συνδέσεων όπως τα ασύρματα δίκτυα χωρίς οποιαδήποτε τροποποίηση. Αλλά ανακουφίζει ακόμα το TCP πρόβλημα που προκαλείται από τη λάθος σύνδεση στο υψηλό περιβάλλον BDP δεδομένου ότι το UDT δεν μειώνει τον απεσταλμένο ρυθμό του για οποιαδήποτε αναφορά απώλειας, και αυξάνεται σε αναλογία προς τον σύνδεσμο χωρητικότητας και την ανεξαρτησία του RTT.

Το σχέδιο 13 παρουσιάζει αλλαγές στην χρησιμοποίηση εύρους ζώνης στην ενιαία ροή UDT ως αυξήσεις συνδέσμου λάθους. Είναι ένα καλό χαρακτηριστικό γνώρισμα ότι το RTT έχει μικρή επίδραση στο ρυθμό λάθους bit σύνδεσης, το οποίο επιτρέπει στο UDT να δουλεύει για τα μεγάλης απόστασης δίκτυα. Το σχήμα επίσης μας δείχνει τη διπλή απόδοση TCP, η οποία εκτελείται πολύ ανεπαρκώς στα υψηλής κατάσταση RTT.

## 5 Περιορισμοί και μελλοντικές βελτιώσεις.

Η αδικία στις συνδέσεις πολύ-δυσχερειών που συζητήθηκαν προηγουμένως είναι ένα από τα σημαντικότερα μειονεκτήματα του UDT. Αυτό γίνεται επειδή το διαθέσιμο σχέδιο εκτίμησης του εύρους ζώνης δεν δουλεύει καλά για τα δίκτυα πολύ-δυσχερειών. Στην πραγματικότητα, (B-C) είναι καλύτερα από B/9 ως διαθέσιμη εκτίμηση εύρους ζώνης σε σενάρια πολύ-δυσχερειών. Εάν ο τύπος αφαιρείται, μπορούμε να επιτύχουμε καλύτερη δικαιοσύνη για την τοπολογία στο σχέδιο 4. Η νέα απόδοση παρουσιάζεται στον πίνακα 3.



Σχήμα 13. αντίκτυπος της λάθους σύνδεσης. Αυτό το σχήμα παρουσιάζει την ενιαία UDT απόδοση κάτω από διαφορετικό σύνδεσμο bit σπάνιου λάθους.(BER). Έξι προσομοιώσεις γίνονται με διαφορετικό εύρος ζώνης και RTTs. Όλοι οι κόμβοι χρησιμοποιούν DropTail ουρά αναμονής.

### ΣΧΗΜΑ 3

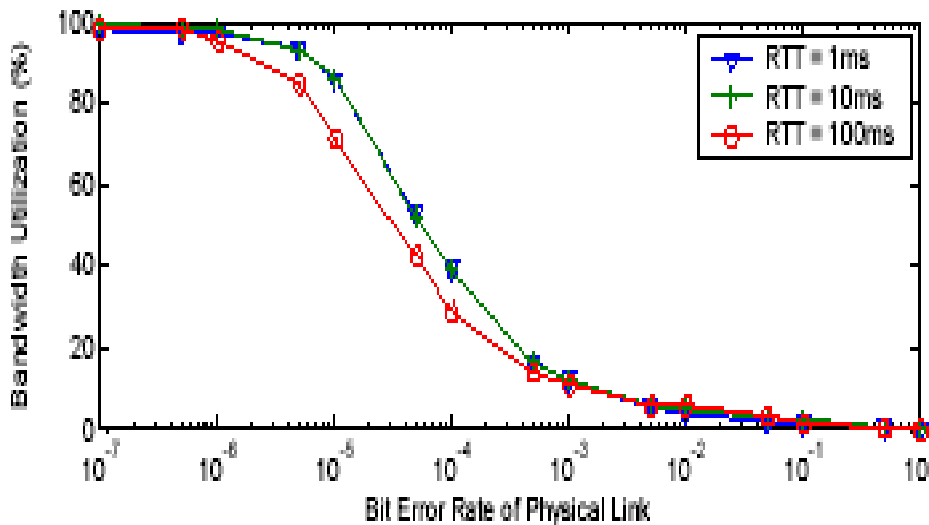
Απόδοση του σχήματος 4 με τροποποιημένο έλεγχο ρυθμού

X	0.1	1	10	20	40	60
AB	198.8	189.2	180.1	170.9	152.5	137.6
AC	0.098	0.979	9.955	19.88	39.46	57.70
X	80	100	120	140	160	180
AB	108.4	104.6	100.8	101.3	100.7	100.3
AC	73.49	92.42	98.47	98.04	98.65	99.00

Ένας άλλος περιορισμός είναι ότι η απόδοση του UDT με τυχαία απώλεια δεν είναι ακόμα ικανοποιητική αν ο ρυθμός λάθους είναι υψηλός(ασύρματο δίκτυο που ο ρυθμός λάθους bit συνδέσεων μπορεί να είναι μέχρι  $10^{-3}$ ).Αφαιρώντας τον τύπο (3) αυτό βοηθά την αύξηση απόδοσης με

τυχαία απώλεια. . Ένα πρόσθετο σχέδιο για να εξεταστεί η φυσική απώλεια συνδέσεων δεν είναι

να μειώσουμε τον ρυθμό αποστολής για την πρώτη ενιαία απώλεια για τον τύπο (6) .Το μειονέκτημα αυτής της τροποποίησης είναι ελαφρώς περισσότερη καθυστέρηση στην ανταπόκριση της απώλειας. Η νέα απόδοση μετά από τις βελτιώσεις ανωτέρω παρουσιάζεται στο σχέδιο 14. Σημείωση, αυτό το σχέδιο ελαφρύνει επίσης το αρνητικό αποτέλεσμα της εκτός σειράς πακέτων.



Σχήμα 14. UDT απόδοση σαν αλλαγές του συνδέσμου ρυθμού απωλειών, μετά από αφαίρεση του τύπου (3) και τροποποιώντας τον τύπο (4). Εντούτοις, ο τύπος (3) είναι απαραίτητος σε UDT για τη σταθερότητα όταν παρουσιάζονται μεγάλες ταυτόχρονες ροές ποσών. Η μεγάλη ταλάντωση μπορεί να συμβαίνει συχνά σε πολλά σενάρια επειδή η αύξηση είναι πάρα πολύ μεγάλη σε σύγκριση με τον ανά ρυθμό απόδοσης ροής.

(3) και (4), B/9 είναι μια καλύτερη εκτίμηση από (B-C) ,σε αυτήν την περίπτωση στις μειωμένες ταλαντώσεις , αν και μπορεί να είναι επίσης μεγάλη καθώς ο αριθμός ροών αυξάνει.

Από το σχήμα 2 μπορούμε να δούμε τον αντίκτυπο του αριθμού παραλληλισμού στην απόδοση και την σταθερότητα του UDT.Η συνολική χρησιμοποίηση εύρους ζώνης αυξάνεται ως αριθμός ταυτόχρονων αυξήσεων ροών. Εν 'τούτοις η στάνταρ απόκλιση ανά

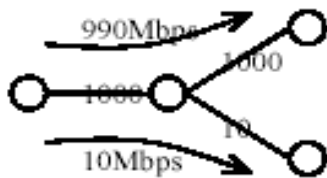
απόδοση ροής κρατιέται αμετάβλητη δεδομένου ότι ο παραλληλισμός συνεχίζει να αυξάνεται ,παρουσιάζοντας ότι η ταλάντωση αυξάνεται.

Αυτή η αρνητική επίδραση μπορεί να ανακουφιστεί από τον έλεγχο ροής ο οποίος παρουσιάζει ένα ανώτερο όριο εάν ο ρυθμός αποστολής αυξάνεται πολύ γρήγορα δεδομένου ότι το παράθυρο ροής ενημερώνεται από τον ληφθέν ρυθμό που είναι αποτέλεσμα από έναν προηγούμενο αποστελλόμενο ρυθμό.

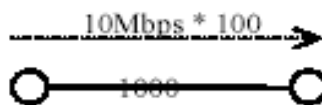
Το ακριβέστερο διαθέσιμο σχέδιο εκτίμησης εύρους ζώνης συχνοτήτων μπορεί να βοηθήσει να λυθεί το πρόβλημα που περιγράφεται ανωτέρω ,αλλά μπορεί να φέρει άλλα προβλήματα.

Εδώ καθορίζουμε το διαθέσιμο εύρος ζώνης συχνοτήτων συχνοτήτων ως εύρος ζώνης συχνοτήτων υπολοίπου όπου μια ροή μπορεί περαιτέρω να το χρησιμοποιήσει κατά μήκος ενός μονοπατιού σε μία δεδομένη στιγμή. Δεδομένου ότι το διαθέσιμο εύρος ζώνης συχνοτήτων αλλάζει κατά διαστήματα (δηλ., είναι μια λειτουργία του χρόνου), δεν είναι κατάλληλο να χρησιμοποιηθεί άμεσα ,ειδώλλως η αδικία μπορεί να προκύψει εάν οι ταυτόχρονες ροές ανιχνεύσουν το εύρος ζώνης συχνοτήτων στο διαφορετικό χρόνο. Υποθέστε ότι έχουμε ήδη μια τέτοια μέθοδο για να ανιχνεύσουμε μια αξία περίπου ίση με το διαθέσιμο εύρος ζώνης συχνοτήτων και το σημαντικότερο , είναι σύμφωνη μεταξύ όλων των ροών.

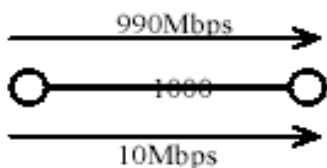
Τώρα θεωρήστε τις 3 καταστάσεις στο σχέδιο 15.Είναι προφανές ότι το διαθέσιμο εύρος ζώνης είναι 0 και στις 3 καταστάσεις .Στις 2 πρώτες καταστάσεις η αύξηση μπορεί να αποφασιστεί από το διαθέσιμο εύρος ζώνης (εάν είναι 0 μπορούμε να καθορίσουμε μια ελάχιστη τιμή για να το αυξήσουμε ,για παράδειγμα 1 byte).Εν τούτοις στην κατάσταση (c) η παράμετρος υπολογισμού αύξησης , σύμφωνα με το διαθέσιμο εύρος ζώνης δεν είναι κατάλληλη και μπορεί να προκαλέσει την ροή 10 Mbps ουσιαστικός χρόνος για να επιτευχθεί η ισορροπία δικαιοσύνης.



(a)



(b)



(c)

Σχήμα 15.3 κλάσεις του εύρους ζώνης συχνοτήτων που μοιράζεται μεταξύ των ταυτόχρονων ροών που μπορούν να επηρεάσουν την απόφαση της αύξησης των παραμέτρων.

Η πρακτική δυσκολία στην ανίχνευση της συνεπούς προσέγγισης του διαθέσιμου εύρους ζώνης συχνοτήτων και των μειονεκτημάτων που περιγράφονται ανωτέρω καθιστά τον περαιτέρω καθαρισμό της διαθέσιμης εκτίμησης εύρους ζώνης συχνοτήτων λιγότερο ελκυστικό στα άκρα με άκρα πρωτόκολλα μεταφοράς.

Σαν γενικό πρωτόκολλο μεταφοράς, ο τύπος (3) μπορεί να βοηθήσει να διατηρήσει χαμηλή ταλάντωση και υψηλή απόδοση στα μεγάλα ποσά αν έχουμε ταυτόχρονες ροές, ενώ στα υπολογιστικά πλέγματα όπου υπάρχει μόνο μικρός αριθμός μαζικών πηγών, μπορεί να αφαιρεθεί.

Οι τετριμμένοι περιορισμοί υπάρχουν. Υπάρχει ένα μειονέκτημα για την βασισμένη εκτίμηση χωρητικότητας συνδέσεων του αποδέκτη όταν ο πολυαγωγός χρησιμοποιείται στην σύνδεση. Βελτίωση ίσως θεωρηθεί η προσαρμογή της PBM μεθόδου στο (9). Έν τούτοις στον πραγματικό κόσμο, σύνδεση πολυαγωγών σπάνια υπάρχει στα υψηλής ταχύτητας δίκτυα.

Τελικά, η χρησιμοποίηση RTT για να ανιχνεύσει την αύξουσα τάση καθυστέρησης μπορεί να αποτύχει στις ασύμμετρες συνδέσεις, δεδομένου ότι το RTT μπορεί να επηρεαστεί από την κυκλοφορία στην αντίστροφη σύνδεση. Ένας εναλλακτικός τρόπος είναι να χρησιμοποιηθεί η μονόδρομη καθυστέρηση (OWD) από τον πομπό στον δέκτη. Το μειονέκτημα αυτού του σχεδίου είναι ότι απαιτούνται το πρόσθετο timestamp στο ACK<sup>2</sup> πακέτο και η χρονική ασύμμετρη διόρθωση.

Στην πραγματικότητα, καθώς η επέκταση ECN γίνεται δημοφιλής, μπορούμε να χρησιμοποιήσουμε το ECN για να αντικαταστήσουμε την καθυστέρηση της τάσης αύξησης ανίχνευσης σε μελλοντική εφαρμογή.

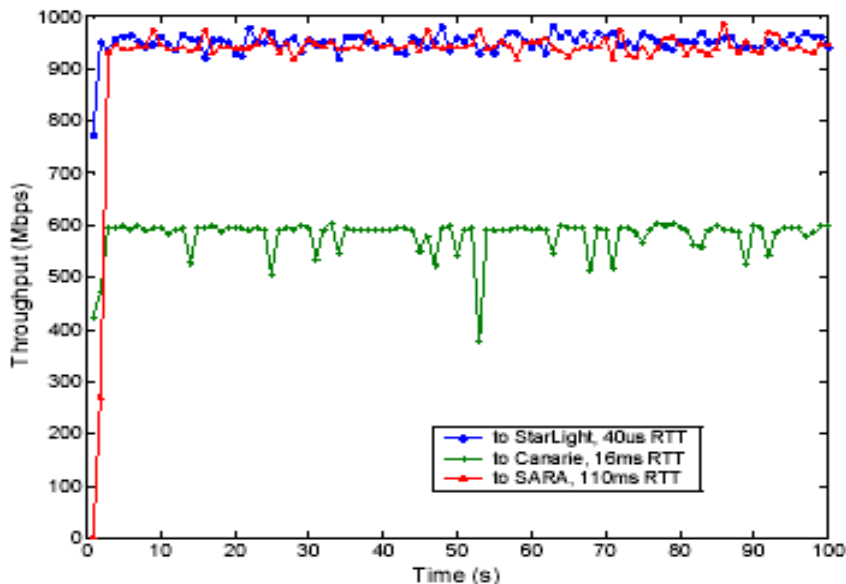
## 6. ΠΕΙΡΑΜΑΤΙΚΑ ΑΠΟΤΕΛΕΣΜΑΤΑ

Σε αυτό το τμήμα θα εξερευνήσουμε μερικά αποτελέσματα στα πραγματικά δίκτυα ως συμπλήρωμα στην προσομοίωση στα προηγούμενα τμήματα, δεδομένου ότι το NS-2 περιβάλλον προσομοίωσης δεν λαμβάνει υπ' όψιν τα αποτελέσματα των υπολογιστικών δυνατοτήτων στο τελικό σύστημα και τις εξωτερικές διαταραχές του περιβάλλοντος όπως οι διακόπτες πλαισίου μεταξύ εφαρμογών στο σύστημα. Ιδιαίτερα θέλουμε να χρησιμοποιήσουμε αυτά τα αποτελέσματα εφαρμογής για να εξετάσουμε τον αντίκτυπο του χρόνου υπολογισμού της επεξεργασίας πακέτων και τον αντίκτυπο των εξωτερικών διαταραχών στον υπολογισμό της ταχύτητας άφιξης πακέτων και της αυξημένης τάσης καθυστέρησης.

Τα πειράματα γίνονται πάνω από 3 παραδείγματα δοκιμής υψηλής ταχύτητας δικτύων : StarLight ,SARA and Canarie.Οι τελευταίοι κεντρικοί υπολογιστές δικτύων

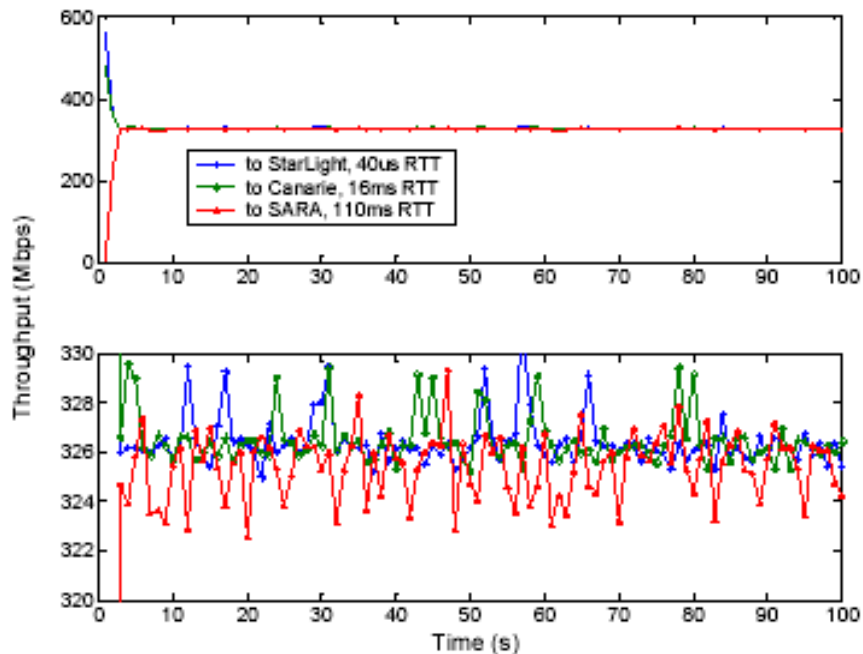
Χρησιμοποιούν Linux στην αρχιτεκτονική της Intel.(Linux πυρήνας 2,4 σε διπλό Intel Xeon 2,4G).

Στο σχήμα 16 φαίνεται ότι η απλή ροή UDT μπορεί να φτάσει περίπου τα 940Mbps μέσω 1Gbps συνδέσμου μαζί με 40μs μικρής διάρκειας RTT και 110 ms μεγάλης διάρκειας RTT.Αυτό μπορεί να φτάσει σχεδόν τα 580Mbps μέσω ενός OC-12 συνδέσμου με 16ms RTT ανάμεσα στο Canarie και στο StarLight.



Σχήμα 16.Απλή UDT απόδοση ροής σε αληθινά δίκτυα. Τα 3 πειράματα έγιναν ανεξάρτητα σε διαφορετικά περιβάλλοντα δικτύων.

Το σχήμα 7 παρουσιάζει την δικαιοσύνη των 3 ροών από StarLight σε StarLight (τοπικό), Canarie και SARA ,αντιστοίχως.

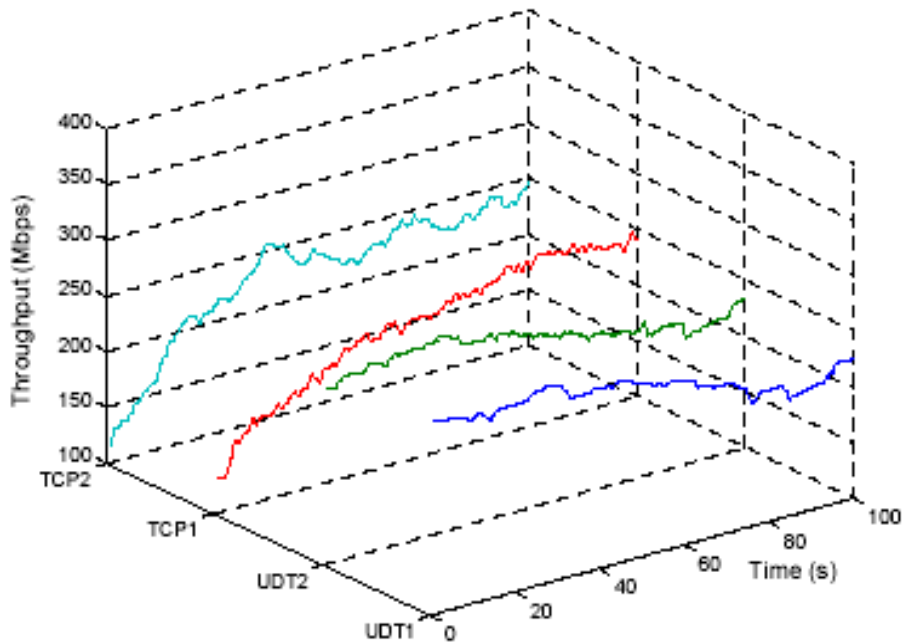


Σχήμα 17.UDT δικαιοσύνη σε πραγματικά δίκτυα. Οι 3 UDT ροές αρχίζουν ταυτόχρονα από StarLight σε Canarie , SARA , και μια άλλη μηχανή σε τοπικό StarLight. .Ο υπο--αριθμός κατωτέρω είναι μια τοπική επέκταση του υπο--αριθμού ανωτέρω.

Οι εφαρμογές στο πραγματικό σύστημα μπορούν να έχουν επιπτώσεις στην φιλικότητα του TCP.Επιπλέον τα σφάλματα εκτίμησης μπορούν επίσης να προκαλέσουν στο UDT λιγότερη φιλικότητα στο TCP όπως συζητούνται στο τμήμα 4. Συνιστάμε μια ομάδα πειραμάτων για να εξετάσουμε την φιλικότητα του TCP σε αληθινά δίκτυα. Η TCP έκδοση στα πειράματα είναι TCP SACK στα Linux μηχανήματα με τον προσωρινό χώρο αποθήκευσης που τίθεται τουλάχιστον σε BDP και η επιλογή window scaling είναι ενεργοποιημένη. Η ταχύτητα της CPU και της μνήμης δεν είναι μπουτιλιαρισμένες σε κανένα από τα πειράματα.

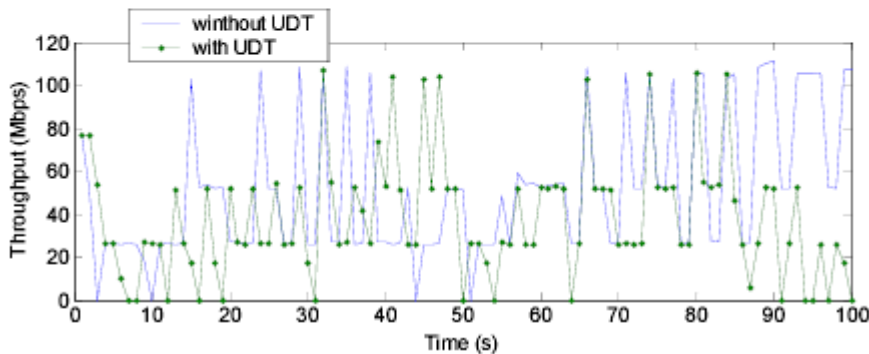
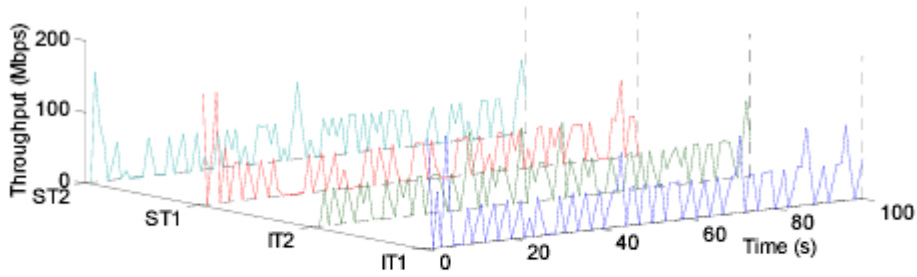
Στο σχήμα 18 εμφανίζεται το αποτέλεσμα των 2 TCP ροών και των 2 UDT ροών συνυπάρχοντας στο StarLiaght τοπικό δίκτυο, με 1 Gbps σύνδεσμο χωρητικότητας και 40μs RTT.Οι TCP ροές χρησιμοποιούν ασήμαντα υψηλότερο εύρος ζώνης από τις ροές UDT.Δεδομένου ότι στα πραγματικά συστήματα το μέγεθος του χώρου αποθήκευσης ενός διακόπτη μπορεί να τεθεί σε μια σταθερή αξία που χρησιμοποιείται στα μεγάλης απόστασης περιβάλλοντα, έτσι μπορεί να είναι πολύ μεγαλύτερο από το BDP μιας τοπικής σύνδεσης. Η προσομοίωση του σχεδίου 18 συμμορφώνεται με την προσομοίωση στο σχέδιο 11.





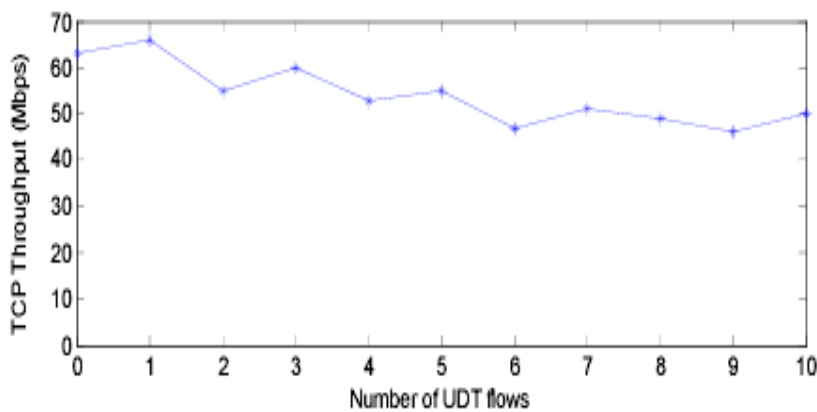
Σχήμα 18. TCP φιλική διάθεση στο LAN. Το σχήμα δείχνει 2 TCP ροές και 2 UDT ροές να συνυπάρχουν στα StarLight τοπικά δίκτυα με 1 Gbps σύνδεσμο χωρητικότητας και 40  $\mu$ s RTT.

Ένα παρόμοιο πείραμα πραγματοποιείται μεταξύ StarLight και Sara, με 1 Gbps εύρος ζώνης συχνότητας και 110ms RTT. (Υπάρχουν 6 δρομολογητές μεταξύ των 2 τελευταίων τερματικών). Το TCP είναι ανεπαρκές σε αυτή την σύνδεση. Μια απλή ροή TCP μπορεί μόνο να χρησιμοποιήσει γύρω στα 30 Mbps ακόμα και όταν το μέγεθος του προσωρινού χώρου αποθήκευσης τίθεται στα 12MB (περίπου ίσος σε BDP). Το σχήμα 19 παρουσιάζει την απόδοση 2 παράλληλων TCP ροών με και χωρίς 2 ταυτόχρονες UDT ροές. Οι συνυπάρχοντες ροές UDT προκαλούν την απελευθέρωση απόδοσης TCP γύρω στο 75% (από 63,3Mbps σε 49,0 Mbps).



Σχήμα 19. Η TCP φιλικότητα στα υψηλά BDP δίκτυα. Το σχήμα παρουσιάζει την απόδοση 2 ταυτόχρονων TCP ροών με και χωρίς 2 ταυτόχρονες ροές υποβάθρου UDT στο δίκτυο μεταξύ StarLight και SARA. IT1 και IT2 είναι απόδοση χωρίς UDT ροές, ενώ ST1 και ST2 είναι με UDT ροές. Ο υποαριθμός κατωτέρω εμφανίζει την συνολική απόδοση των 2 TCP ροών.

Εγκαθιστούμε επίσης 500 TCP ρεύματα του 1MB με διαφορετικό αριθμό ροών υποβάθρου UDT για να εξετάσουμε πως το UDT μπορεί να έχει επιπτώσεις στις μικρές ροές ζωής TCP στα μεγάλης απόστασης δίκτυα (στο ίδιο δίκτυο με αυτό του σχεδίου 19). Το αποτέλεσμα φαίνεται στο σχέδιο 20.



Σχήμα 20. Συνολική απόδοση 500 μικρών TCP ροών με διαφορετικό αριθμό ροών υποβάθρου UDT από 0 σε 10.

## 7. ΣΧΕΤΙΚΗ ΕΡΓΑΣΙΑ

Ο έλεγχος ρυθμού έχει θεωρηθεί σαν αποτελεσματική μέθοδο για την υψηλή απόδοση για μεγάλο χρονικό διάστημα. NETBLT [24] είναι το πιο πρόσφατο παράδειγμα για υψηλή απόδοση μεταφοράς μαζικών δεδομένων χρησιμοποιώντας τον έλεγχο ρυθμού. Μεταφέρει τα δεδομένα , ομάδα ανά ομάδα. Το δεδομένο αποστολής ρυθμού είναι συντονισμένο ατότου στέλνεται κάθε ομάδα δεδομένων. Το μέγεθος της ομάδας δεδομένων μπορεί να θεωρηθεί σαν στατικό παράθυρο ελέγχου ροής.

Προτάθηκε στο [1] ότι είναι απαραίτητο να χρησιμοποιηθεί βασικός έλεγχος παραθύρου για να υποστηρίξει τον έλεγχο ρυθμού για να φτάσει σε έναν καλύτερο αλγόριθμο ελέγχου συμφόρησης . Το ΧΤΡ [25] είναι ένα τέτοιο πρωτόκολλο που συνδυάζει δυο είδη σχεδίων ελέγχου. Ωστόσο το ΧΤΡ περιλαμβάνεται στην σχεδίαση των πυλών και δεν είναι μια άκρη με άκρη προσπέλαση.

Στην πραγματικότητα , το leaky bucket και ο token bucket που χρησιμοποιούνται στο δίκτυο του ATM είναι μια ομοιότητα ως συνδυασμός του ελέγχου ρυθμού.( leaky bucket) και του ελέγχου παραθύρων (token bucket).

Ωστόσο , τα πιο πρόσφατα σχέδια βασισμένα στο ρυθμό, είναι παρακινήματα από εφαρμογές πολυμέσων ροής δια-δικτύου, του οποίου ο σημαντικός στόχος δεν είναι η υψηλή απόδοση αλλά η ομαλή ροή δεδομένων. Ένα τυπικό παράδειγμα είναι το TFRC [13] , ή ο TCP φιλικός έλεγχος ρυθμού. Συντονίζει το ποσοστό αποστολής σύμφωνα με το TCP μοντέλο απόδοσης [21].

Εν τω μεταξύ , έχει προταθεί στο TCP να παγώσει το χρόνο δια-πακέτων για κάθε αποστολή πακέτου. Εντούτοις έχει βρεθεί ότι ο ρυθμός δεν λύνει το πρόβλημα αποδοτικότητας , αλλά μερικές φορές το κάνει χειρότερο.

Η εκτίμηση εύρους ζώνης συχνότητων έχει προσδιοριστεί επίσης ως μια από τις σημαντικές οδηγίες για να βελτιώσει την απόδοση των πρωτοκόλλων μεταφοράς [ 2 ]. Οι περισσότερες από τις προτάσεις επιλέγουν την άκρη με άκρη εκτίμηση εύρους ζώνης συχνότητων συνδέσεων, δεδομένου ότι είναι πολύ πιο πρακτικές να ανιχνευτούν σε πραγματικό χρόνο, πιο σταθερές και ευκολότερες να εφαρμοστούν.

Η διαθέσιμη μέτρηση εύρους ζώνης συχνότητων είναι πολύ πιο δύσκολο να ανιχνευτεί και έχει τα προβλήματα που συζητήσαμε προηγουμένως σε αυτό το έγγραφο για τα πρωτόκολλα μεταφοράς. Ο αυξανόμενος αλγόριθμος ανίχνευσης τάσης καθυστέρησης του UDT δανείζεται από μια διαθέσιμη μέτρηση εύρους ζώνης συχνότητων ένα εργαλείο που ονομάζεται Pathload [7]. Ο αυθεντικός αλγόριθμος σε Pathload χρησιμοποιείται για αυξανόμενη ανίχνευση OWD. Εντούτοις , το σχέδιο εκτίμησης εύρους ζώνης συχνότητων σε Pathload χρειάζεται

ειδικό ρυθμό που ελέγχει ρεύματα δειγμάτων και δεν μπορεί να ανιχνεύσει το εύρος ζώνης συχνοτήτων σε πραγματικό χρόνο, έτσι δεν είναι κατάλληλο να χρησιμοποιηθεί άμεσα στα πρωτόκολλα μεταφοράς. Υπάρχουν πρωτόκολλα που χρησιμοποιούν τις τεχνικές εκτίμησης εύρους ζώνης αλλά δεν υπολογίζουν ούτε τον σύνδεσμο χωρητικότητας ούτε το διαθέσιμο εύρος ζώνης συχνοτήτων. Το TCP WESTWOOD [4] είναι ένα τέτοιο παράδειγμα. Το οποίο υπολογίζει το δίκαιο μερίδιο (στα πρότυπα TCP) του εύρους ζώνης συχνοτήτων και συντονίζει το TCP παράθυρο και μέγεθος κατωφλίου κατάλληλα για να βελτιωθεί η απόδοση. Στην πραγματικότητα, τα περισσότερα απ τα βασισμένα στην εξίσωση σχέδια (π.χ TFRC) ανήκουν σε αυτή την κατηγορία.

Το βασισμένο στην καθυστέρηση σχέδιο είναι επίσης μια προσέγγιση, για να βελτιωθεί η απόδοση η οποία πρωτοαναφέρεται στο [12]. Το TCP VEGAS [5] χρησιμοποιεί καθυστέρηση ως ένδειξη της συμφόρησης για να βελτιωθεί η απόδοση.

Μια άλλη σχετική τεχνική είναι η εκλεκτική επιβεβαίωση λήψη σήματος, η οποία έχει χρησιμοποιηθεί ευρέως σε πολλά πρωτόκολλα όπως το NETBLT [24], XTP [25] και TCP SACK [17]. Η διαφορά του UDT από τα άλλα πρωτόκολλα που χρησιμοποιούν την εκλεκτική επιβεβαίωση λήψης σήματος είναι ότι το UDT χρησιμοποιεί έναν χρονιστή για να αποφασίσει πότε να το αναγνωρίσει.

Ο ορισμός της TCP φιλικότητας είναι επίσης αμφισβητήσιμος σήμερα.

Πρόσφατα οι ερευνητές σκοπεύουν να κάνουν την ανταλλαγή μεταξύ της αποδοτικότητας και της δικαιοσύνης για τα υψηλής ταχύτητας πρωτόκολλα όπως το HighSpeed TCP [27], το οποίο αναγκάζει το καθένα να επιδρά το ίδιο σαν TCP σε χαμηλά BDP περιβάλλοντα δεδομένου ότι επιτρέπει στο ίδιο να υπερβεί το TCP στα υψηλά BDP περιβάλλοντα.

Εντωμεταξύ το πρόβλημα χρησιμοποίησης εύρους ζώνης συχνοτήτων άνω των υψηλών BDP μπορεί να λυθεί καλύτερα με τον ανοιχτό έλεγχο βρόχων, όπως XCP [6], παρά το κόστος της πιθανής βελτίωσης των υποδομών δικτύων. Το XCP χρησιμοποιεί τις ρητές πληροφορίες συμφόρησης στους δρομολογητές για τον έλεγχο συμφόρησης. Οι δρομολογητές παρατηρούν το μέγεθος των σειρών αναμονής και το διαθέσιμο εύρος ζώνης συχνοτήτων και λένε στις πηγές πώς να αντιδράσουν από την τροποποίηση της επικεφαλίδας πακέτων XCP.

## 8. ΣΥΜΠΕΡΑΣΜΑ

Παρακινήμενοι από την απαίτηση της γρήγορης μεταφοράς των μαζικών δεδομένων στην υψηλή περιοχή υπολογισμού απόδοσης, σχεδιάσαμε και αναπτύξαμε ένα άκρη με άκρη σχέδιο ελέγχου συμφόρησης για να υπερνικήσουμε τις ανεπάρκειες του TCP και την πόλωση δικαιοσύνης πέρα από το υψηλό δίκτυο BDP με την τυχαία απώλεια.

Ο αλγόριθμος UDT χρησιμοποιεί ρυθμό βασισμένο στον έλεγχο συμφόρησης για να φτάσει στην υψηλή απόδοση, η οποία χρησιμοποιεί την τεχνική εκτίμησης εύρους ζώνης συχνοτήτων για να τεθεί η κατάλληλη παράμετρος αύξησης ποσοστού του αλγορίθμου AIMD. Ένας βασισμένος έλεγχος ροής παραθύρου χρησιμοποιείται ως ενθαρρυντική μέθοδος για την αποδοτικότητα και τη δικαιοσύνη. Ένα σταθερό διάστημα ελέγχου ρυθμού χρησιμοποιείται για να αφαιρέσει την πόλωση RTT.

Τα αποτελέσματα προσομοίωσης και εφαρμογής έχουν εμφανίσει καλή χρησιμοποίηση εύρους ζώνης συχνοτήτων στα πολύ υψηλά δίκτυα BDP. Εν τω μεταξύ, είναι ακόμα δίκαιο σε άλλες UDT ροές με διαφορετικούς χρόνους έναρξης και διαφορετικά RTTs. Ιδιαίτερα, το UDT είναι φιλικό προς το TCP, αν και μερικές φορές το TCP μπορεί να υπερβεί τις συνυπάρχοντες ροές UDT. Επιπλέον το UDT βοηθάει να μειωθούν οι εκρήξεις ροών δεδομένων στο δίκτυο και απαιτούν μικρότερες ουρές αναμονής στους δρομολογητές.

Έχουμε συζητήσει τους σημαντικούς περιορισμούς του UDT, συμπεριλαμβανομένης της αδικίας στα σενάρια και τις ταλαντώσεις πολυ-δυσχερειών στον πολύ υψηλό συναγωνισμό. Οι κατάλληλες βελτιώσεις έχουν προταθεί για να λύσουν αυτά τα προβλήματα στο μέλλον.

Σαν άκρη με άκρη προσπέλαση, το πρωτόκολλο είναι πρακτικά πολύ χρήσιμο για υψηλή απόδοση μαζικής μεταφοράς δεδομένων στα υψηλά BDP δίκτυα, όπως τα υπολογιστικά πλέγματα. Αυτό έχει χρησιμοποιηθεί σε μια δέσμη από εφαρμογές σε υψηλής απόδοσης υπολογιστική περιοχή, περιλαμβάνοντας DSTP [30] και ένωση ροής [29].

Ο στόχος του αλγορίθμου UDT δεν είναι να αντικαταστήσει το TCP αλλά να παρέχει έναν εναλλακτικό τρόπο που μπορεί να υπερνικήσει τα μειονεκτήματα του TCP σε υψηλά BDP περιβάλλοντα, ενώ το γενικό σχέδιο επιτρέπει εφαρμογές χρησιμοποιώντας το UDT για να τρέξει στο παραδοσιακό χαμηλό BDP δίκτυο εξίσου καλά.

Αν και πολλές από τις ιδέες που χρησιμοποιήθηκαν στο UDT προτάθηκαν πριν, τις αναπτύξαμε περαιτέρω λεπτομερώς και τις ενσωματώσαμε στα σχεδόν αποτελεσματικά σχέδια με την εργασία εφαρμογής για να λύσουμε ένα πρόβλημα, ότι πολλές εντατικές εφαρμογές στοιχείων υποφέρουν σήμερα. Ιδιαίτερα, η θεωρητική και

πειραματική εργασία μας έχει προσδιορίσει τη σημασία του συνδυασμού ελέγχου ροής και ελέγχου παραθύρου και τη σημασία της εκτίμησης εύρους ζώνης συχνότητων στα πρωτόκολλα μεταφοράς. Οι τελευταίες λύσεις στο μέλλον θα μπορούσαν να περιλάβουν τις πληροφορίες και τις αποφάσεις από τις πύλες (δηλ. ανοικτός-βρόχος έλεγχος συμφόρησης), όπως ECN και XCP. Η εργασία μας βοηθά επίσης να προσδιορίσουμε ποιες πληροφορίες πρέπει να παρασχεθούν από την πύλη, και τι δεν είναι απαραίτητο, για να ελαχιστοποιηθεί η πολυπλοκότητα των πυλών και να αυξηθεί η ευελιξία και η εξελισιμότητα του δια-δικτύου.

## ΠΡΟΣΘΗΚΗ Α:ΔΙΚΑΙΟΣΥΝΗ ΣΤΟ ΣΕΝΑΡΙΟ ΣΤΕΝΩΝ ΠΕΡΑΣΜΑΤΩΝ

Σε αυτήν την προσθήκη συζητάμε το σενάριο 2-bottleneck/2-flow λεπτομερέστερα. Το σενάριο είναι παρόμοιο όπως στο σχέδιο 4.

Υποθέστε ότι οι δύο δυσχέρειες έχουν το δυναμικό μονάδας συνδέσεων  $B_1$  και  $B_2$  ( $B_1 < B_2$ ), αντίστοιχα. Στην ισορροπία ο ρυθμός αποστολής εάν οι δύο ροές είναι  $r_1$  και  $r_2$ , αντίστοιχα.

Κατά μια πλευρά σύμφωνα με το (4), το  $r_1$  θα συνεχίσει να αυξάνεται μέχρι

$$B_1 - r_1 < 10^y \leq B_2 - r_2 \quad (8)$$

όπου  $y$  είναι ένας ακέραιος.

Αφετέρου, δεδομένου ότι  $B_1 < B_2$ , έχουμε

$$r_1 \leq B_2 / 2 \quad (9)$$

Στην φάση ισορροπίας, το ΟΥΚ μπορεί περίπου να χρησιμοποιήσει όλο το εύρος ζώνης συχνοτήτων σύμφωνα με την ανάλυση αποδοτικότητας, επομένως:

$$r_1 + r_2 \approx B_2 \quad (10)$$

Από τις (8),(9) και (10) μπορούμε να φτάσουμε:

$$B_1 / 2 < r_1 < B_2 / 2 \quad (11)$$

Σύμφωνα με την ανώτατη-ελάχιστη αρχή δικαιοσύνης, το δίκαιο μερίδιο του  $r_1$  πρέπει να είναι είτε  $B_1$  (εάν  $B_1 < B_2/2$ ) είτε  $B_2/2$  (ειδιάλλως). Έτσι στη χειρότερη περίπτωση το  $r_1$  μπορεί ακόμα να φθάσει στο μισό από το δίκαιο μερίδιο. Λεπτομερέστερα, υποθέστε:

$$10^a \leq B_1 < 10^{a+1} \quad (12)$$

όπου ο  $a$  είναι ακέραιος.

Σύμφωνα με τις (8),(11) και (12),

$$r_1 \approx \min(\max(10^a, B_1 - 10^a), B_2 / 2) \quad (13)$$

Έτσι η χειρότερη περίπτωση συμβαίνει όταν  $B_1 < B_2/2$  and  $B_1 = 2 \cdot 10^a$ .

Η ανάλυση ανωτέρω υποθέτει ότι τα αποτελέσματα του τύπου (3) και οι μικρότερες αυξήσεις μετά από αυτήν του διαθέσιμου εύρους ζώνης συχνοτήτων  $r_1$  ικανοποιούν (8). Στην πραγματικότητα ο τύπος (3) μπορεί να φέρει μερικά αρνητικά αποτελέσματα στη δικαιοσύνη όπως εμφανίζεται στον πίνακα 1 και 3. Εν τω μεταξύ, οι μικρότερες αυξήσεις μπορούν να αυξήσουν το  $r_1$  μέχρι 11%.

## ΠΡΟΣΘΗΚΗ Β:ΖΗΤΗΜΑΤΑ ΕΦΑΡΜΟΓΗΣ

Η εφαρμογή είναι κρίσιμη για τα υψηλής απόδοσης πρωτόκολλα μεταφοράς δεδομένων. Μερικοί αλγόριθμοι εργάζονται καλά στη θεωρία και την προσομοίωση αλλά ίσως έχουν φτωχή απόδοση εάν η εφαρμογή δεν σχεδιάζεται καλά. Σε αυτήν την προσθήκη περιγράφουμε διάφορα κρίσιμα μέρη του αλγορίθμου UDT στα οποία πρέπει να δοθεί περισσότερη προσοχή στις εφαρμογές.

Ένα από τα πιο κοινά προβλήματα είναι ότι η παραγωγή της πληροφορίας ελέγχου και ανάγνωσης εφαρμογής δεδομένων στην πλευρά δεκτών, μπορεί να πάρει ουσιαστικό χρόνο συγκρίνοντας με το υψηλό ρυθμό άφιξης πακέτων. Μια φτωχή εφαρμογή μπορεί να προκαλέσει τη συχνή απελευθέρωση πακέτων και ακόμα χιονοστιβάδα απώλειας πακέτων (απώλειες επεξεργασίας προκαλούν περισσότερες απώλειες) , ο δέκτης διατηρεί έναν κατάλογο απώλειας για να καταγράψει τις πληροφορίες απώλειας. Η πρόσβαση στον κατάλογο απώλειας μπορεί να χρειαστεί ένα μεγάλο διάστημα ώστε τα πακέτα που φτάνουν να υπερχειλίσουν τον προσωρινό χώρο αποθήκευσης του πρωτοκόλλου.

Για να χειριστεί αυτό το είδος προβλήματος, είναι απαραίτητο να διανεμηθεί η επεξεργασία ομοιόμορφα σε μικρά κομμάτια ακόμα κι αν αυτό οδηγεί στον υψηλότερο συνολικό χρόνο επεξεργασίας, εάν η επεξεργασία δεν μπορεί να γίνει αρκετά μικρή για να αποφύγει το πρόβλημα.

Παραδείγματος χάριν, χρησιμοποιούμε μια ειδική δομή δεδομένων για τις χαμένες πληροφορίες έτσι ώστε όλες οι διαδικασίες ,εισαγωγής ,διαγραφής και ανανέωσης παίρνουν όμοια μικρό χρόνο, ενώ χρησιμοποιώντας έναν απλό πίνακα ή μια συνδεδεμένη δομή καταλόγων οι λειτουργίες εισαγωγής και διαγραφής θα χρειαστεί να ανιχνεύσουν και /ή να μετακινήσουν δεδομένα τα οποία παίρνουν τον ουσιαστικό χρόνο κατά τη διάρκεια της υψηλής συμφόρησης οφειλόμενης στα μεγάλα ποσά χαμένων πακέτων.

Το αντίγραφο μνήμης είναι ένα άλλο πρόβλημα που μπορεί να προκαλέσει παρόμοιο πρόβλημα. Εντούτοις, αυτό το είδος προβλήματος έχει εξεταστεί λεπτομερώς σε μεγάλο μέρος της εργασίας όπως [23].

Τέλος, η ειδική ειδοποίηση μπορεί να προκύψει στον υψηλό χρονοδιακόπτη ακρίβειας που απαιτείται για τον έλεγχο ρυθμού. Αυτήν την περίοδο ένας τέτοιος χρονοδιακόπτης δεν είναι διαθέσιμος σε πολλά γενικής χρήσης λειτουργούντα συστήματα. Για τη μεταφορά στοιχείων σε gigabits ανά δευτερόλεπτο, η ακρίβεια συγχρονισμού πρέπει να είναι τουλάχιστον επίπεδο μικροδευτερολέπτου, και είναι καλύτερη στο επίπεδο συχνότητας της CPU. Μια απλή εφαρμογή μπορεί να απασχολήσει κύκλους ρολογιού της CPU αυξάνοντάς της την ουρά αναμονής. Υπάρχει επίσης υποστήριξη υλικού να χρησιμοποιεί



διακόπτες σε μερικές πλατφόρμες όπως το APIC [32] INTEL αρχιτεκτονικής.

Πολλές εφαρμογές που βασίζονται στον ρυθμό πρωτοκόλλου, χρησιμοποιούν πρόσθετη μεταβλητή μεγέθους έκρηξης, για να ελέγξει τον αριθμό των πακέτων που μπορούν να σταλούν συνεχώς χωρίς inter-packet καθυστέρηση και απενεργοποίηση για ένα μακρύτερο χρονικό διάστημα. Αυτή η μέθοδος χρησιμοποιείται για να αποφευχθεί το υψηλό πρόβλημα συγχρονισμού ακρίβειας εφ' όσον ο συνολικός χρόνος απενεργοποίησης είναι αρκετά μεγάλος. Εντούτοις στα υψηλής ταχύτητας δίκτυα ο αριθμός πακέτων που μπορεί να σταλεί στο ελάχιστο διάστημα απενεργοποίησης του συστήματος, μπορεί να είναι πολύ μεγάλος και καθιστά τον έλεγχο ποσοστού χωρίς νόημα.

Οι άλλοι χρονοδιακόπτες για να προκαλέσουν το διάστημα ανατροφοδότησης, τη ρύθμιση ποσοστού, και το γεγονός λήξης δεν χρειάζονται υψηλή ακρίβεια, έτσι μπορούν να εφαρμοστούν με την αυτό-χρονομέτρηση, δηλ., ρωτώντας το χρόνο του συστήματος μετά από κάθε αποστολή ή τον οριακό χρόνο του λαμβάνοντος γεγονότος.

Η υλοποίηση πηγαίου κώδικα και η προσομοίωση είναι απευθείας προσβάσιμα.

## **ΠΗΓΕΣ**

### **ΒΙΒΛΙΟΓΡΑΦΙΑ**

tcp / ip ΕΚΔΟΣΕΙΣ Μ.ΓΚΙΟΥΡΔΑΣ

ΔΙΚΤΥΑ ΥΠΟΛΟΓΙΣΤΩΝ ΤΑΝΕΝΒΑΟΥΜ

---

### **ΔΙΚΤΥΑΚΟΙ ΧΩΡΟΙ**

[www-unix.mcs.anl.gov/~kettimut/pfldnet04.ppt](http://www-unix.mcs.anl.gov/~kettimut/pfldnet04.ppt)

<http://udt.sourceforge.net/>

<http://dsd.lbl.gov/DIDC/PFLDnet2004/talks/Grossman-slides.pdf>

[http://www.igrid2005.org/media/press\\_09.28.05\\_ncdm.html](http://www.igrid2005.org/media/press_09.28.05_ncdm.html)

<http://www.sciencedaily.com/release>

<http://whitepapers.silicon.com/0,39024759,60105701p-39000676q,00.htm>

[dsd.lbl.gov/DIDC/PFLDnet2004/papers/Grossman.pdf](http://dsd.lbl.gov/DIDC/PFLDnet2004/papers/Grossman.pdf)

<https://mail.internet2.edu/wws/arc/i2-news/2003-10/msg00006.html>

<http://www.codeproject.com/internet/udt.asp>

---

