



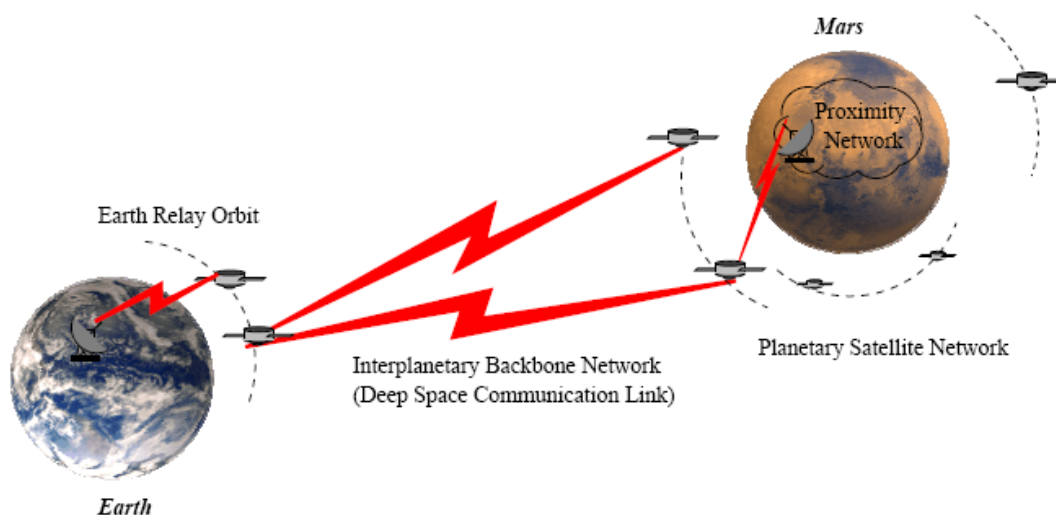
ΔΗΜΟΚΡΙΤΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΡΑΚΗΣ

Τμήμα Ηλεκτρολόγων Μηχανικών
και Μηχανικών Υπολογιστών
Πολυτεχνική Σχολή Ξάνθης

Τομέας Ηλεκτρονικής και Τεχνολογίας Συστημάτων Πληροφορικής
Εργαστήριο Ψηφιακών Συστημάτων

Διπλωματική Εργασία: Κόμνιος Ιωάννης

Αποτίμηση Απόδοσης Διαδικτυακών Πρωτοκόλλων σε Διαστημικό Περιβάλλον



Επιβλέπων: Επίκουρος Καθηγητής κ. Β. Τσαουσίδης

Ξάνθη 2007

Αφιερώνεται στους γονείς μου
και στην αδερφή μου, Άννα.

ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να ευχαριστήσω τον επίκουρο καθηγητή κ. Βασίλη Τσαουσίδα για την τεράστια συμβολή του στη διεκπεραίωση αυτής της εργασίας και για τη συνεχή του συμπαράσταση. Επίσης θα ήθελα να ευχαριστήσω ιδιαίτερα τον υποψήφιο διδάκτορα Ιωάννη Ψαρά, καθώς και όλη την εργαστηριακή ομάδα για την πολύτιμη βοήθεια τους.

Περιεχόμενα

Περίληψη.....	1
Κεφάλαιο 1: Transmission Control Protocol (TCP)	3
1.1. Εισαγωγή.....	3
1.2. Transmission Control Protocol.....	3
1.2.1. Επικεφαλίδα.....	4
1.2.2. Εγκαθίδρυση σύνδεσης (Connection Establishment).....	6
1.2.3. Τερματισμός σύνδεσης (Connection Termination).....	7
1.2.4. Αξιοπιστία και ταξινόμηση (Reliability, Ordering).....	8
1.2.5. Χρονικό διάστημα επαναποστολής (Retransmission Timeout).....	9
1.2.6. Καθυστερημένες επιβεβαιώσεις (Delayed Acknowledgments).....	10
1.3. Έλεγχος ροής και έλεγχος συμφόρησης.....	10
1.3.1. Έλεγχος ροής (Flow Control).....	10
1.3.2. Έλεγχος και αποφυγή συμφόρησης (Congestion avoidance and control).....	10
1.3.3. Κριτήρια κατάστασης ελέγχου συμφόρησης.....	12
1.3.4. Μηχανισμοί Αργό Ξεκίνημα (Slow Start) και Αποφυγή Συμφόρησης (Congestion Avoidance) του TCP.....	13
1.3.5. Μηχανισμοί Fast Retransmit και Fast Recovery.....	15
1.3.6. Το μοντέλο ανάλυσης για τον έλεγχο συμφόρησης.....	15
1.3.7. Προσθετική αύξησης / πολλαπλασιαστική μείωση (AIMD)...	17
1.4. Εκδόσεις του πρωτοκόλλου μεταφοράς TCP.....	18
1.4.1. TCP Tahoe.....	18
1.4.2. TCP Reno.....	19
1.4.3. TCP New Reno.....	19
1.4.4. TCP SACK.....	20
1.4.5. TCP Vegas.....	20
1.4.6. TCP Real.....	21
Κεφάλαιο 2: Διαπλανητικό Διαδίκτυο (InterPlanetary Internet)	23
2.1. Εισαγωγή.....	23
2.2. Αρχιτεκτονική δομή του Διαπλανητικού Διαδικτύου.....	23
2.3. Διαφορές δικτυακών χαρακτηριστικών ανάμεσα στο Διαδίκτυο και το Διαπλανητικό Διαδίκτυο.....	25
2.4. Απαιτήσεις σχεδιασμού πρωτοκόλλου για το Διαπλανητικό Διαδίκτυο.....	28
2.5. Εφαρμογές που χρησιμοποιούνται στις διαστημικές επικοινωνίες.....	29

2.6. Προβλήματα που παρουσιάζει το TCP στο Διάστημα.....	30
2.6.1. Αλγόριθμος Αργού Ξεκινήματος (Slow Start).....	31
2.6.2. Μηχανισμοί που βασίζονται στα παράθυρα.....	31
2.6.3. Ρυθμοί σφαλμάτων.....	31
2.6.4. Ασυμμετρία καναλιών.....	31
2.6.5. Χωρητικότητα της σύνδεσης.....	32
2.6.6. Διακοπές συνδεσιμότητας.....	32
2.6.7. Διακύμανση των μετρούμενων χρόνων RTT.....	33
2.6.8. Καθυστερημένη ανατροφοδότηση.....	33
2.6.9. Χωρητικότητα καταχωρητών.....	33
2.6.10. Εγκαθίδρυση σύνδεσης.....	34
2.6.11. Σειρά παράδοσης δεδομένων.....	34
2.6.12. Επικοινωνία από άκρο σε άκρο.....	34
2.7. Τρόποι αντιμετώπισης των προβλημάτων που παρουσιάζει το TCP στο Διάστημα.....	36
Κεφάλαιο 3: Εργαστηριακή μεθοδολογία.....	39
3.1. Τοπολογία των πειραμάτων και γενικές παράμετροι.....	39
3.2. Μονάδες μέτρησης της απόδοσης.....	40
3.2.1 Bandwidth Efficiency.....	40
3.2.2 Bandwidth X Delay Product.....	42
3.2.3 Fairness Index.....	43
3.3. Δείκτης κατανάλωσης ενέργειας.....	44
3.4. Επιλεκτικές Αρνητικές Επιβεβαιώσεις SNACK.....	44
3.5. Σενάρια προσομοίωσης συνδέσεων.....	45
Κεφάλαιο 4: Πειραματικά αποτελέσματα.....	47
4.1. Αποτελέσματα.....	47
4.1.1 Αύξηση της καθυστέρησης.....	47
4.1.2 Μέγεθος καταχωρητών.....	49
4.1.3 Ρυθμός σφαλμάτων ανά πακέτο.....	50
4.1.4 Χωρητικότητα της κύριας σύνδεσης.....	54
4.1.5 Ασύμμετρα κανάλια.....	56
4.1.6 Σφάλματα στο forward link και στο return link.....	58
4.1.7 Διακοπή συνδεσιμότητας.....	59
4.1.8 Bandwidth X Delay Product.....	62
4.1.9 Συνδυασμός προβλημάτων (PER, Asymmetry, Blackout).....	62
4.2. Συμπεράσματα και σχόλια.....	64
Κεφάλαιο 5: Σχετική έρευνα.....	65
5.1. TP Planet.....	65
5.2. SCPS-TP.....	69
5.3. RCP Planet.....	73

5.4. Saratoga.....	75
--------------------	----

Βιβλιογραφία.....	78
--------------------------	-----------

Παράρτημα

Συντομογραφίες

ACK	Acknowledgment
adwnd	Advertised Window
AIMD	Additive Increase / Multiplicative Decrease
ARQ	Automatic Repeat Request
AQM	Active Queue Management
BER	Bit Error Rate
Bps	Bytes per second
BW	Bandwidth
BWBB	Bandwidth Backbone
CBR	Constant Bit Rate
cwnd	Congestion Window
DACK	Duplicate Acknowledgment
DAR	Delayed Automatic Retransmission
DBB	Delay BackBone
DMC	Disaster Monitoring Constellation
DTN	Delay Tolerant Networking
ECN	Explicit Congestion Notification
FEC	Forward Error Correction
F.I.	Fairness Index
FIFO	First In First Out
IPN	InterPlanetary Internet
LEO	Low Earth Orbit
LBA	Last Byte Acked
LBS	Last Byte Sent

Mbps	Megabits per second
MSS	Maximum Sending Size
NAK	Negative Acknowledgment
NS	Network Simulator
PER	Packet Error Rate
RCS	Rate Control Scheme
RED	Random Early Detection
Ret.packet	Retransmitted Packets
RTO	Retransmission Timeout
RTT	Round Trip Time
SACK	Selective Acknowledgment
SCPS-TP	Space Communication Protocol Standards - Transport Protocol
SNACK	Selective Negative Acknowledgment
ssthresh	Slow Start Threshold
TCP	Transmission Control Protocol
TT	Total Time
UDP	User Datagram Protocol

Περίληψη

Οι αποστολές εξερεύνησης του Διαστήματος είναι ζωτικής σημασίας για την απόκτηση πληροφοριών σχετικά με το Διάστημα και το Σύμπαν. Οι αποστολές αυτές απαιτούν την ανάπτυξη εξελιγμένων τεχνολογιών επικοινωνίας και δικτύωσης. Είναι λογικό να σκεφτεί κανείς να χρησιμοποιήσει το διαδεδομένο και επιτυχημένο, όσον αφορά τις επίγειες επικοινωνίες, πρωτόκολλο TCP στις διαστημικές επικοινωνίες. Όμως, οι υπάρχουσες εκδόσεις του TCP έχουν σχεδιαστεί για να λειτουργούν αποδοτικά πάνω από τις παραδοσιακές ενσύρματες συνδέσεις, οι οποίες δεν παρουσιάζουν σφάλματα και έχουν σχετικά μικρές τιμές καθυστέρησης διάδοσης της τάξης των millisecond. Αντίθετα, το διαστημικό περιβάλλον θέτει ένα πλήθος από προκλήσεις για την παροχή αξιόπιστης επικοινωνίας από άκρο σε άκρο με ανεκτό επίπεδο υπηρεσιών. Μεγάλες καθυστερήσεις διάδοσης, περιορισμένο εύρος ζώνης, απώλειες λόγω σφαλμάτων, ασύμμετρες χωρητικότητες συνδέσεων και ενδιάμεση συνδεσιμότητα “συνωμοτούν” για τον ισχυρό περιορισμό της απόδοσης του TCP.

Η υπάρχουσα εργασία έχει ως σκοπό την αποτίμηση της απόδοσης των διαδεδομένων διαδικτυακών πρωτοκόλλων σε ένα εχθρικό περιβάλλον, όπως είναι το Διάστημα. Εξετάζεται ποιοι μηχανισμοί του TCP υστερούν και προτείνονται διάφορες εναλλακτικές μέθοδοι για την αντιμετώπιση των προβλημάτων. Πειραματικά εξετάζεται η επίδραση των χαρακτηριστικών του Διαστήματος στην απόδοση των πρωτοκόλλων μεταφοράς TCP. Αρχικά εκτελούνται πειράματα όπου μόνο μια παράμετρος επιδρά στη σύνδεση, π.χ. η ασυμμετρία των καναλιών δεδομένων-επιβεβαιώσεων, ενώ στο τέλος εκτελείται ένα πείραμα το οποίο προσομοιώνει τις πραγματικές συνθήκες μιας διαστημικής αποστολής. Στο πείραμα αυτό εμφανίζονται ταυτόχρονα σφάλματα, ασυμμετρία καναλιών και διακοπές λειτουργίας σε μία σύνδεση με μεγάλη τιμή καθυστέρησης.

Για την εκτέλεση των πειραμάτων χρησιμοποιήθηκε ο προσομοιωτής δικτύων Network Simulator 2 (NS-2) σε λειτουργικό σύστημα-πλατφόρμα Linux.

Συνοπτικά, καταλήγουμε στα παρακάτω συμπεράσματα όσον αφορά την εφαρμογή των διαφόρων εκδόσεων του διαδικτυακού πρωτοκόλλου TCP πάνω από διαστημικές συνδέσεις:

- Η απόδοση του TCP μειώνεται καθώς αυξάνεται η καθυστέρηση, ενώ για μεγάλες τιμές RTT η απόδοση σχεδόν μηδενίζεται. Το γεγονός αυτό καθιστά τα ισχύοντα πρωτόκολλα μεταφοράς κατάλληλα μόνο για μικρές αποστάσεις μετάδοσης.
- Σημαντικό ρόλο στην απόδοση των πρωτοκόλλων παίζει ο ρυθμός σφαλμάτων ανά πακέτο. Για μικρές τιμές PER η απόδοση δεν επηρεάζεται από σφάλματα. Όμως, για μεγαλύτερες τιμές παρατηρείται ραγδαία πτώση της απόδοσης, ανεξάρτητα από τη τιμή της καθυστέρησης.
- Η απώλεια επιβεβαιώσεων δεν επηρεάζει αισθητά την απόδοση του πρωτοκόλλου, σε αντίθεση με την απώλεια πακέτων δεδομένων οπότε η απόδοση μειώνεται σημαντικά.

- Οι διάφορες εκδόσεις του πρωτοκόλλου μεταφοράς TCP πετυχαίνουν καλή απόδοση μόνο για λόγους ασυμμετρίας μέχρι 50:1. Για μεγαλύτερες τιμές παρατηρείται κατακόρυφη πτώση της απόδοσης.
- Ο συνολικός χρόνος που διαρκούν οι διακοπές συνδεσιμότητας, λόγω παρεμβολής ενός αστεροειδή ή πλανήτη στη σύνδεση, είναι αντιστρόφως ανάλογος της απόδοσης του πρωτοκόλλου.

Με λίγα λόγια, αποδεικνύεται πως τα υπάρχοντα πρωτόκολλα μεταφοράς δεν είναι κατάλληλα για εφαρμογή σε διαστημικό περιβάλλον. Πρέπει να βρεθούν εναλλακτικοί μηχανισμοί οι οποίοι θα επιτυγχάνουν υψηλή απόδοση και αξιόπιστη μεταφορά δεδομένων σε συνδέσεις στο Απομακρυσμένο Διάστημα.

Η παρούσα εργασία περιλαμβάνει πέντε κεφάλαια. Το 1^ο κεφάλαιο αποτελεί μια περιγραφή του πρωτοκόλλου μεταφοράς TCP, των δομικών του στοιχείων και των βασικών λειτουργιών για έλεγχο ροής και έλεγχο συμφόρησης. Το 2^ο κεφάλαιο αναφέρεται στο Διαπλανητικό Διαδίκτυο (InterPlanetary Internet), την αρχιτεκτονική του δομή και τις διαφορές χαρακτηριστικών που παρουσιάζει με το Διαδίκτυο. Παρουσιάζονται οι εφαρμογές που χρησιμοποιούνται στις διαστημικές επικοινωνίες (Bulk Data Transfer, Telemetry, Telecommand) και αναλύονται τα προβλήματα που παρουσιάζει το TCP σε διαστημικό περιβάλλον, καθώς επίσης προτείνονται τρόποι αντιμετώπισης τους. Το 3^ο κεφάλαιο αναφέρεται στην εργαστηριακή μεθοδολογία, καταγράφοντας τις μονάδες μέτρησης απόδοσης, τις καθυστερήσεις, το δείκτη δικαιοσύνης καθώς, επίσης, και ένα δείκτη κατανάλωσης ενέργειας. Ακόμη, αναφέρονται όλες οι απαραίτητες παράμετροι ενός δικτύου και περιγράφονται τα σενάρια που εκτελούνται πειραματικά. Στο 4^ο κεφάλαιο δίνονται τα αποτελέσματα των πειραμάτων, ενώ στο τέλος του κεφαλαίου καταγράφονται τα συμπεράσματα που προκύπτουν από την ανάλυση των αποτελεσμάτων. Τέλος, το 5^ο κεφάλαιο περιλαμβάνει την ανάλυση τεσσάρων διαφορετικών πρωτοκόλλων μεταφοράς, τα οποία έχουν σχεδιαστεί για εφαρμογή σε διαστημικό περιβάλλον.

Κεφάλαιο 1

Transmission Control Protocol (TCP)

1.1. Εισαγωγή

Το σημαντικότερο πρωτόκολλο του Διαδικτύου (Internet) είναι το πρωτόκολλο ελέγχου μετάδοσης TCP (Transmission Control Protocol). Το TCP παρέχει αξιόπιστη μεταφορά δεδομένων και διαθέτει έλεγχο ροής (flow control) και έλεγχο συμφόρησης (congestion control), έχοντας μάλιστα μηχανισμούς που επιτρέπουν τη δίκαιη χρήση των δικτυακών πόρων από όλους του χρήστες. Το TCP χρησιμοποιείται από τη μεγάλη πλειονότητα των δικτυακών εφαρμογών και μεταφέρει περίπου το 95% των πακέτων που κυκλοφορούν στο Διαδίκτυο. Σχεδιάστηκε και αναπτύχθηκε σε μια εποχή όπου η υπάρχουσα υποδομή του δικτύου ήταν αποκλειστικά ενσύρματη. Για το λόγο αυτό τα θέματα που είχε να αντιμετωπίσει ήταν καθαρά θέματα σταθερότητας, ετερογένειας (με την έννοια του εύρους ζώνης του δικτύου, των buffers του παραλήπτη και των καθυστερήσεων), δικαιοσύνης σε εύρος ζώνης και ελέγχου συμφόρησης για την αποφυγή καταστάσεων καταστροφικής συμφόρησης.

Με την πάροδο του χρόνου και την εξέλιξη των τηλεπικοινωνιακών συστημάτων, πολλές συνδέσεις έγιναν πλέον ασύρματες καθιστώντας το TCP ανεπαρκές για τις νέες συνθήκες του Διαδικτύου. Το ασύρματο Διαδίκτυο προκάλεσε έναν αριθμό προβλημάτων, τα οποία απαιτούν το σχεδιασμό νέων πρωτοκόλλων και αρχιτεκτονικών και επαναξιολόγηση των υπάρχοντων προτύπων. Ακόμη μεγαλύτερα προβλήματα παρουσιάζει η εφαρμογή του TCP στο Διαπλανητικό Διαδίκτυο, λόγω των χαρακτηριστικών του δικτύου. Τα προβλήματα αυτά θα πρέπει να ληφθούν υπόψη και να επανεξεταστούν οι παράμετροι και οι μηχανισμοί του TCP, ώστε η εφαρμογή του TCP στο Διάστημα να είναι αποδοτική [1], [2], [3], [4].

1.2. Transmission Control Protocol (TCP)

Το TCP [5],[6] είναι ένα πρότυπο πρωτόκολλο μεταφοράς point-to-point, το οποίο ξεκινάει την επικοινωνία μεταξύ αποστολέα-δέκτη χωρίς μεταφορά δεδομένων (connection oriented) παρά μόνο με την ανταλλαγή μηνυμάτων ελέγχου. Στόχος του είναι να διαπιστώσει αν ο παραλήπτης υπάρχει και μπορεί να συνδεθεί, καθώς και να ανταλλάξει πληροφορίες σε σχέση με κάποιες λεπτομέρειες που αφορούν την επικοινωνία.

Οι τέσσερις βασικοί στόχοι που προσπαθεί να εκπληρώσει το TCP δίνονται παρακάτω:

- **Δικαιοσύνη (Fairness):** Κάθε flow πρέπει να είναι δίκαιο ως προς τα άλλα, ώστε να μπορούν να μοιράζονται το ίδιο κανάλι.

- **Απόδοση (Performance):** Η καλή απόδοση είναι στενά συνδεδεμένη με τη συνεχή ανακάλυψη διαθέσιμου εύρους ζώνης.
- **Αποφυγή Συμφόρησης (Congestion Avoidance):** Η υπερχειλίση του δικτύου και, συνεπώς, η απώλεια πακέτων δεδομένων λόγω συμφόρησης είναι μια κατάσταση που πρέπει να αποφεύγεται και σε αυτό πρέπει να συμβάλει αποτελεσματικά το εφαρμοζόμενο πρωτόκολλο.
- **Σταθερότητα (Stability):** Η σταθερότητα-ισορροπία κατά την ανταλλαγή δεδομένων αποτελεί πρωταρχική επιδίωξη για επιτυχή μετάδοση μέσα σε ένα περιβάλλον όπως το χαώδες Διαδίκτυο.

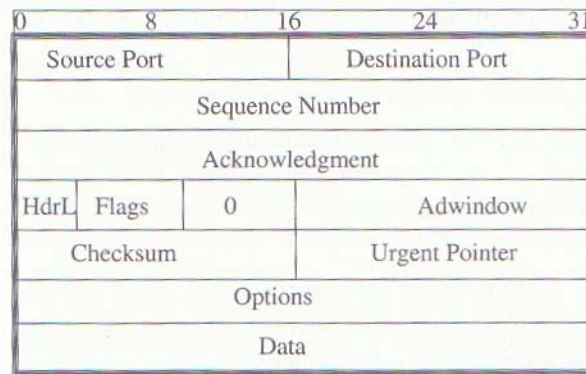
Οι κυριότερες λειτουργίες που διαθέτει το TCP είναι οι εξής:

- Εγκαθίδρυση και τερματισμός σύνδεσης (Connection establishment/termination)
- Ανίχνευση της διαθέσιμης χωρητικότητας (Available bandwidth)
- Αξιόπιστη μεταφορά δεδομένων με επαναμεταφορά (Retransmission)
- Επαναταξινόμηση εκτός-σειράς πακέτων
- Έλεγχος ροής (Flow control)
- Έλεγχος και αποφυγή συμφόρησης (Congestion avoidance and control).

Το Transmission Control Protocol αποτελεί ένα σύνδεσμο όπου από τη μία πλευρά είναι ο χρήστης ή η εφαρμογή και από την άλλη πλευρά ένα πρωτόκολλο επιπέδου δικτύου όπως το IP (Internet Protocol). Το TCP να λαμβάνει μία απλή αναξιόπιστη υπηρεσία πακέτων από το κατώτερο επίπεδο πρωτοκόλλων και είναι σχεδιασμένο να λειτουργεί αξιόπιστα πάνω από ένα ευρύ φάσμα μέσων μετάδοσης, χωρίς να επηρεάζεται από το ρυθμό μετάδοσης, τις καθυστερήσεις, τις παύσεις μετάδοσης (π.χ. handoff events), τη λανθασμένη σειρά με την οποία φτάνουν τα πακέτα κτλ.

1.2.1. Επικεφαλίδα

Το πρωτόκολλο TCP για κάθε πακέτο που μεταδίδει προσθέτει ένα κομμάτι πληροφορίας, το οποίο περιέχει χρήσιμες πληροφορίες για την επικοινωνία αποστολέα και δέκτη. Οι πληροφορίες αυτές εμπεριέχονται στην επικεφαλίδα και είναι γραμμένες με ένα προκαθορισμένο τρόπο που έχει επιβληθεί. Η δομή της επικεφαλίδας φαίνεται στο σχήμα 1.1.



Σχήμα 1.1. Επικεφαλίδα του TCP

Το κάθε πεδίο της επικεφαλίδας παρέχει συγκεκριμένες πληροφορίες για την επικοινωνία του αποστολέα και του δέκτη. Έτσι τα πεδία Source και Destination Port καθορίζουν που και από ποιόν θα παραδοθεί το πακέτο σε συνδυασμό με τα SrcIPAdress και DestIPAdress του IP.

Το πεδίο Sequence number χρησιμοποιείται για την ταξινόμηση των πακέτων και επίσης για να μπορούν να διακριθούν τα πακέτα που έχουν ξανασταλθεί, αν και έχουν φτάσει στον προορισμό τους (όταν για παράδειγμα χαθεί το πακέτο επιβεβαίωσης). Στο σημείο αυτό να αναφέρουμε ότι το TCP χρησιμοποιεί byte (byte-oriented) για να συμπληρώσει ένα πακέτο (που στο TCP ονομάζεται segment) και το πεδίο Sequence number περιλαμβάνει τον αριθμό-δείκτη του πρώτου byte στο πακέτο.

Το πεδίο Acknowledgment χρησιμοποιείται για την επιβεβαίωση λήψης ενός πακέτου. Το TCP επιτρέπει τη χρήση συνοπτικών επιβεβαιώσεων (cumulative acknowledgment), την επιβεβαίωση δηλαδή όχι μόνο για το συγκεκριμένο πακέτο που παραλαμβάνεται αλλά και για όλα τα προηγούμενα. Αυτό το πετυχαίνει με έναν παράδοξο τρόπο (που έχει όμως τη σκοπιμότητα του): κάθε φορά που λαμβάνεται ένα καινούργιο πακέτο x , το πεδίο acknowledgment παίρνει την τιμή $x+1$, προσδιορίζοντας το επόμενο πακέτο που αναμένεται. Αυτό όμως ισχύει μόνο για συνεχόμενα πακέτα. Αν π.χ. λείπει το $x+2$ όμως τα $x+1$ και $x+3$ έχουν παραληφθεί επιτυχώς, το TCP θα στείλει επιβεβαίωση λήψης για το $x+2$ και όχι για το $x+4$, δηλαδή το TCP χρησιμοποιεί duplicate acks (DACKs) όταν φτάσει ένα πακέτο εκτός σειράς. Αυτή η τακτική επιτρέπει στον αποστολέα να γνωρίζει ποιο πακέτο λείπει από τη σειρά και ότι κάποια πακέτα προωθούνται επιτυχώς.

Το πεδίο HdrL αναφέρεται στο μήκος του πακέτου σε μονάδες των 32-bits και είναι αναγκαίο γιατί η επικεφαλίδα, λόγω του πεδίου options, δεν έχει σταθερό μήκος. Το πρωτόκολλο επιτρέπει επίσης τη χρήση έξι διαφορετικών συμβολικών μονάδων (σημαιών - Flags):

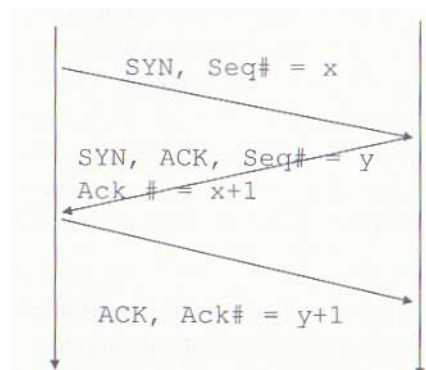
- **SYN, FIN:** Χρησιμοποιούνται στην εγκαθίδρυση και τον τερματισμό σύνδεσης (Connection establishment/ termination)
- **Reset:** Χρησιμοποιείται όταν ο παραλήπτης παίρνει πακέτα που δε μπορεί να κατατάξει.

- **Push:** Σηματοδοτεί άμεση παράδοση των πακέτων στην εφαρμογή
- **URGPNTNR:** Χρησιμοποιείται για να διαχωρίσει ένα επείγον τμήμα του πακέτου, σε συνδυασμό με το πεδίο URGPNTNR, το οποίο περιλαμβάνει ένα δείκτη που καθορίζει μέχρι ποιο σημείο είναι το μη επείγον τμήμα.
- **ACK:** Χρησιμοποιείται για να δηλώσει στον παραλήπτη ότι το πεδίο Acknowledgment χρειάζεται επεξεργασία. Αυτό συμβαίνει συνήθως όταν μαζί με το πακέτο στέλνεται και κάποια επιβεβαίωση.

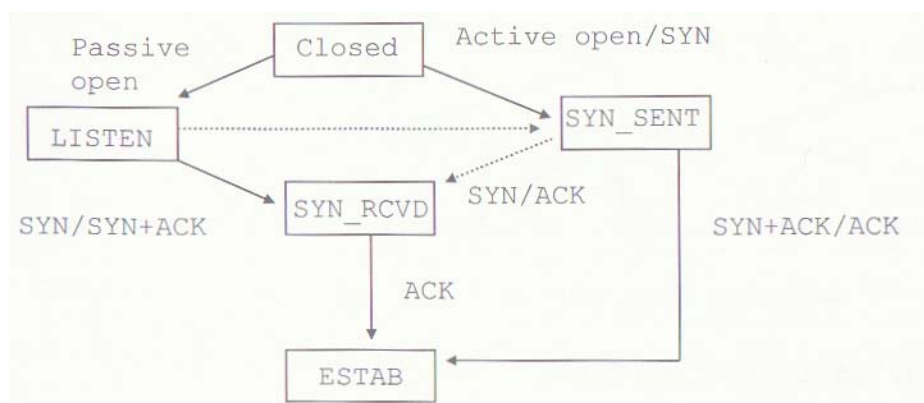
Το πεδίο AdWindow χρησιμοποιείται στον έλεγχο ροής (flow control) και το πεδίο Checksum στον έλεγχο λαθών. Το πεδίο Options είναι προαιρετικό.

1.2.2. Εγκαθίδρυση σύνδεσης (Connection Establishment)

Το TCP είναι πρωτόκολλο με σύνδεση (Connection-oriented), δηλαδή ξεκινάει την επικοινωνία με μια αρχική σύνδεση με τον παραλήπτη, χωρίς να μεταφέρει δεδομένα. Ο στόχος του είναι να διαπιστώσει αν ο παραλήπτης υπάρχει και μπορεί να συνδεθεί, καθώς και να ανταλλάξει πληροφορίες σε σχέση με κάποιες λεπτομέρειες που αφορούν την επικοινωνία (π.χ. σε τι μονάδες θα υπολογίζουμε το AdWindow, αν υπάρχουν κοινές επιλογές (options)) κ.λ.π.).



Σχήμα 1.2. Τριπλή «χειραψία»

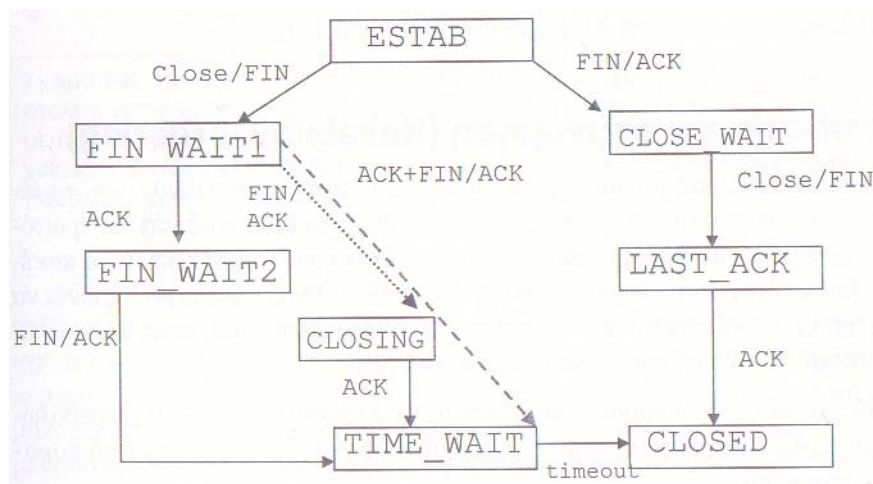


Σχήμα 1.3 Διάγραμμα μετάβασης καταστάσεων

Η αρχική επικοινωνία γίνεται με έναν αλγόριθμο που ονομάζεται 3-way handshake (τρίδρομη ή τριπλή χειραψία). Όπως φαίνεται από το σχήμα 1.2, ο αποστολέας ενημερώνει τον παραλήπτη για την πρόθεση του να συνδεθεί, στέλνοντας ένα SYN και τον αριθμό σειράς του byte (Sequence Number) από το οποίο θα ξεκινήσει τη μεταφορά των δεδομένων. Το νούμερο αυτό το διαλέγει στην τύχη, προκειμένου να αποφύγει επικαλύψεις με πρόσφατες συνδέσεις. Ο παραλήπτης επιβεβαιώνει τη λήψη (με ACK) και τη βούληση του να επικοινωνήσει (με SYN) και δηλώνει τον αριθμό σειράς με τον οποίο θα ξεκινήσει την επικοινωνία. Τέλος, ο αποστολέας επιβεβαιώνει τη λήψη του SYN από τον παραλήπτη και η σύνδεση είναι έτοιμη.

1.2.3. Τερματισμός σύνδεσης (Connection Termination)

Η διαδικασία τερματισμού της σύνδεσης είναι λίγο διαφορετική, κυρίως γιατί, όταν κάποιος από τους δυο υπολογιστές αποφασίζει να διακόψει τη σύνδεση, πιθανώς ο άλλος υπολογιστής να μην έχει ολοκληρώσει την αποστολή των δεδομένων του. Ο υπολογιστής που αποφασίζει να διακόψει στέλνει το πρώτο FIN και περνάει στην κατάσταση ACK_WAIT όπου αναμένει την επιβεβαίωση λήψης από τον παραλήπτη (Σχήμα 1.4). Όταν την παραλάβει, περνάει στην κατάσταση FIN_WAIT, όπου αναμένει τη σύμφωνη γνώμη του παραλήπτη για διακοπή της σύνδεσης. Όταν λάβει το FIN, επιβεβαιώνει με ACK και περνάει στην κατάσταση TIME_WAIT. Σε αυτή την κατάσταση παραμένει για χρόνο διπλάσιο του μέγιστου χρόνου που μπορεί ένα πακέτο να κυκλοφορήσει στο Διαδίκτυο. Ο λόγος είναι ο εξής: αν και ο πρώτος υπολογιστής (αποστολέας) έχει στείλει ACK στο FIN του παραλήπτη, δεν γνωρίζει αν ο παραλήπτης το έχει λάβει - αυτή η κατάσταση θα προκαλούσε την αποστολή καινούργιου FIN από τον παραλήπτη, το οποίο είναι δυνατόν να παραδοθεί σε μια καινούργια σύνδεση, προκαλώντας τον άμεσο τερματισμό της.



Σχήμα 1.4. Διάγραμμα μετάβασης καταστάσεων για τον τερματισμό σύνδεσης με το TCP

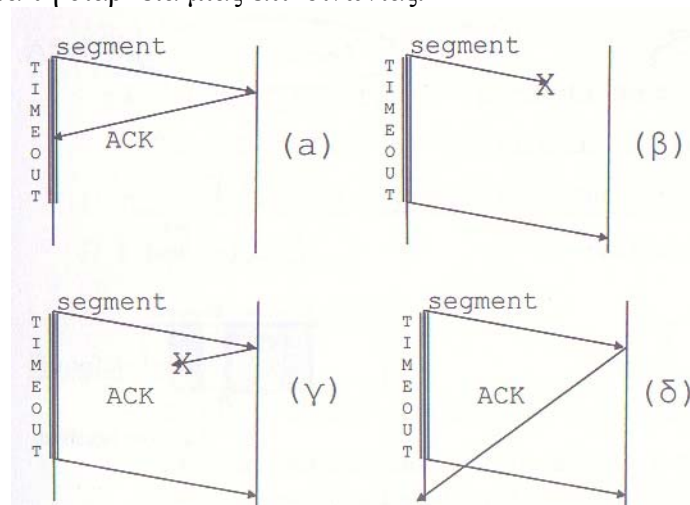
1.2.4. Αξιοπιστία και ταξινόμηση (Reliability, Ordering)

Το TCP μπορεί να ανακάμψει από δεδομένα που είναι αλλοιωμένα, έχουν χαθεί, έχουν φτάσει στον προορισμό τους δύο φορές ή έχουν φτάσει με λάθος σειρά. Αυτό επιτυγχάνεται με τον ορισμό ενός sequence number σε κάθε πακέτο που μεταφέρεται, και με την απαίτηση μίας θετικής επιβεβαίωσης - acknowledgment (ACK δηλαδή ότι το πακέτο έφτασε στον προορισμό του). Τα ACKs του TCP είναι συνοπτικά (cumulative) και μεταφέρουν τον sequence number της επόμενης οκτάδας δεδομένων που αναμένεται να φτάσει, κάνοντας γνωστό με αυτόν τον τρόπο τον αριθμό του τελευταίου πακέτου που έφτασε. Όταν φτάσει ένα πακέτο εκτός σειράς, επιστρέφεται ένα duplicate ACK (DACK). Αν δεν παραληφθεί το ACK μέσα σε ένα συγκεκριμένο χρονικό διάστημα (timeout interval), ο αποστολέας ξαναστέλνει το πακέτο δεδομένων. Στον παραλήπτη, τα sequence numbers χρησιμοποιούνται για να μουν σε σωστή σειρά τα πακέτα που πιθανώς έχουν φτάσει εκτός σειράς και για να απορριφθούν αυτά που έχουν φτάσει δύο φορές.

Υπάρχουν τέσσερα ενδεχόμενα κατά την επικοινωνία:

- α. Το πακέτο φθάνει στον παραλήπτη, ο οποίος στέλνει επιβεβαίωση. Η επιβεβαίωση παραλαμβάνεται από τον αποστολέα, ο οποίος γνωρίζει ότι το πακέτο παραδόθηκε κανονικά.
- β. Το πακέτο 'χάνεται' στον παραλήπτη. Ο αποστολέας θεωρεί ότι το πακέτο πρέπει να ξανασταλθεί.
- γ. Το πακέτο παραδίδεται, όμως η επιβεβαίωση χάνεται. Ο αποστολέας θεωρεί ότι το πακέτο πρέπει να ξανασταλθεί.
- δ. Το πακέτο φθάνει στον παραλήπτη και η επιβεβαίωση φθάνει στον αποστολέα - φθάνει όμως καθυστερημένα. Στο μεταξύ, ο αποστολέας έχει ξαναστείλει το πακέτο.

Στο σχήμα 1.5 φαίνονται και σε γραφική αναπαράσταση τα τέσσερα αυτά πιθανά ενδεχόμενα κατά τη διάρκεια μιας επικοινωνίας.



Σχήμα 1.5. Τέσσερα ενδεχόμενα κατά την επικοινωνία.

Για να δημιουργήσουμε ένα μηχανισμό αξιοπιστίας σε ένα πρωτόκολλα χρειαζόμαστε:

- α. Στην πρώτη περίπτωση μόνο επιβεβαίωση.
- β. Στη δεύτερη περίπτωση ένα μηχανισμό χρονικού περιθωρίου που θα σηματοδοτήσει την επαναποστολή.
- γ. Στην τρίτη περίπτωση ένα μηχανισμό χρονικού περιθωρίου που θα σηματοδοτήσει την επαναποστολή, καθώς και μια μέθοδο αρίθμησης (sequence number) για να διακρίνουμε το πακέτο που ξαναστέλνεται από ένα καινούριο πακέτο.
- δ. Η τέταρτη περίπτωση είναι ακριβώς ίδια με την τρίτη.

1.2.5. Χρονικό Διάστημα Επαναποστολής (Retransmission Timeout)

Το χρονικό διάστημα πέρα από το οποίο ένα πακέτο ξαναστέλνεται, αν δεν παραληφθεί επιβεβαίωση για αυτό, λέγεται Χρονικό Διάστημα Επαναποστολής (Retransmission Timeout) και δίνεται ως συνάρτηση του εκτιμηθέντος Round-Trip-Time (RTT) μεταξύ των δύο άκρων της σύνδεσης. Εξαιτίας της μεταβλητότητας του RTT μέσα στο Διαδίκτυο, το retransmission timeout καθορίζεται δυναμικά. Ο αποστολέας μετράει το χρόνο που χρειάζεται για να στείλει ένα πακέτο με συγκεκριμένο sequence number, και να παραλάβει το ACK που αντιστοιχεί σε αυτόν τον sequence number. Αυτός ο χρόνος είναι το Round - Trip - Time (RTT). Βασισμένο στα δείγματα RTT, τα οποία παρουσιάζουν μεγάλες διακυμάνσεις, το υπολογιζόμενο Retransmission Timeout (RTO) θα παρουσιάζει επίσης μεγάλη διακύμανση. Για το λόγο αυτό κρίνεται λογικό να προσδιορίσουμε μια αναμενόμενη μέση τιμή και να την αναπροσαρμόζουμε με βάση τις μετρήσεις που παίρνουμε κάθε φορά. Ο αρχικός αλγόριθμος που χρησιμοποιήθηκε για τον προσδιορισμό του της αναμενόμενης μέσης τιμής (EstimatedRTT) παρουσιάζεται παρακάτω.

Αρχικός Αλγόριθμος

- Μέτρησε το *SampleRTT* για κάθε ζευγάρι (*segment*, *ACK*)
- Υπολόγισε τον αναλογικό μέσο όρο (*weighted average*) των:

$$EstimatedRTT = a \times EstimatedRTT + b \times SampleRTT$$

όπου $a+b=1$ και a μεταξύ 0.8 και 0.9

Αρχικά το χρονικό περιθώριο υπολογιζόταν πολύ συντηρητικά ως το διπλάσιο του εκτιμώμενου RTT. Όμως, οι Jacobson/Karels παρατήρησαν ότι μερικές φορές η διακύμανση του RTT είναι μικρή και επομένως μια τιμή για το χρονικό περιθώριο κοντά στο εκτιμώμενο RTT θα ήταν καλύτερη προσέγγιση του προβλήματος. Από την άλλη μεριά, όταν πάλι η διακύμανση είναι μεγάλη, η τιμή της διακύμανσης θα έπρεπε να ληφθεί υπόψη για τον καθορισμό του χρονικού περιθωρίου. Για το λόγο αυτό ο υπολογισμός του timeout επομένως, πρέπει να περιλαμβάνει δύο όρους, RTT και διακύμανση, έτσι ώστε όταν η διακύμανση είναι αμελητέα, στον υπολογισμό του

timeout να κυριαρχεί το RTT, ενώ, αντίθετα, όταν η διακύμανση είναι μεγάλη να κυριαρχεί αυτή.

1.2.6. Καθυστερημένες επιβεβαιώσεις (Delayed Acknowledgments)

Ένας παραλήπτης TCP μπορεί να αυξήσει την αποδοτικότητα του Διαδικτύου αλλά και του τελικού χρήστη με το να μη στέλνει ACKs για κάθε εισερχόμενο πακέτο δεδομένων. Αυτό είναι γνωστό με τον όρο «καθυστερημένες επιβεβαιώσεις» (delayed ACK). Ο παραλήπτης πρέπει να στείλει ένα ACK για κάθε δεύτερο πακέτο δεδομένων που παραλαμβάνει. Αν αυτό το δεύτερο πακέτο δε φτάσει μέσα σε ένα συγκεκριμένο χρονικό διάστημα (όχι παραπάνω από 0.5 sec), τότε στέλνει ένα ACK. Αυτός ο μηχανισμός είναι ευρέως χρησιμοποιούμενος σε υπάρχουσες εκδόσεις του TCP.

1.3. Έλεγχος ροής και Έλεγχος Συμφόρησης

Το TCP είναι υπεύθυνο για τον έλεγχο ροής (flow control) και τον έλεγχο συμφόρησης (congestion control). Ο σκοπός του ελέγχου ροής είναι να αποτρέψει τους αποστολείς από το να υπερχειλίσουν τη χωρητικότητα των καταχωρητών των παραληπτών, ενώ ο σκοπός του ελέγχου συμφόρησης είναι να αποτρέψει την εισαγωγή μεγάλου ποσού δεδομένων στο δίκτυο και, έτσι, να υπερχειλίσει τους buffers των δρομολογητών στους κόμβους του δικτύου.

1.3.1. Έλεγχος ροής (Flow Control)

Η αξιοπιστία και η ταξινόμηση στο TCP γίνονται μέσω του μηχανισμού του κυλιόμενου παραθύρου (sliding window), περιορίζοντας το ποσό των σημαντικών δεδομένων το οποίο μπορεί να στείλει ένας TCP αποστολέας. Σε οποιαδήποτε χρονική στιγμή, ένας αποστολέας TCP απαγορεύεται να στείλει δεδομένα με sequence number μεγαλύτερο από το άθροισμα του μεγαλύτερου sequence number του οποίου η παραλαβή έχει επιβεβαιωθεί και του μεγέθους του παραθύρου αποστολής (sending window size). Ο αποστολέας TCP ρυθμίζει την ταχύτητα αποστολής δεδομένων αυξάνοντας ή μειώνοντας το παράθυρο αποστολής του. Σε περίπτωση που ο ρυθμός αποστολής δεδομένων ξεπερνά το ρυθμό με τον οποίο διαβάξει τα δεδομένα ο παραλήπτης από τον buffer του δημιουργείται υπερχειλίση του καταχωρητή και απώλεια δεδομένων. Για την αποφυγή αυτής της κατάστασης χρησιμοποιείται το πεδίο «Advertised Window» (ένα πεδίο 16-bit στην επικεφαλίδα του TCP) με κάθε ACK. Το Advertised Window δείχνει ένα εύρος δεκτών sequence number μετά το τελευταίο πακέτο που λήφθηκε επιτυχώς και, έτσι, ο παραλήπτης TCP επηρεάζει το ποσό των δεδομένων που στέλνονται από τον αποστολέα.

1.3.2. Έλεγχος και Αποφυγή Συμφόρησης (Congestion Control and Avoidance)

Η συμφόρηση στα δίκτυα υπολογιστών, ουσιαστικά προκαλείται από την ετερογένεια του δικτύου στις ταχύτητες των ζεύξεων (link speeds) [7]. Η πιθανότητα να συμβεί συμφόρηση είναι υψηλή όπου υπάρχει bottleneck, δηλαδή όπου το άθροισμα των χωρητικοτήτων των εισερχόμενων ζεύξεων ξεπερνάει τη χωρητικότητα της εξερχόμενης από το bottleneck ζεύξης. Όταν το ποσοστό των δεδομένων που

φτάνουν στον κόμβο ξεπερνάει το ποσοστό που μπορεί να εξυπηρετηθεί από την υπηρεσία, τότε αρχίζει να χτίζεται η ουρά μέσα στον buffer του κόμβου. Ο buffer μπορεί τελικά να υπερχειλίζει και να αρχίσει να 'ρίχνει' τα εισερχόμενα πακέτα. Αν οι αποστολές δεν μειώσουν την ταχύτητα με την οποία στέλνουν δεδομένα, τότε έχουμε μία κατάσταση η οποία καλείται καταστροφική συμφόρηση (congestive collapse). Όπως είναι φυσικό το goodput του συστήματος μειώνεται δραματικά. Πρέπει λοιπόν να γίνουν κάποιες μετρήσεις για να περιοριστεί το ποσοστό αποστολής και να αποφευχθεί η κατάσταση μόνιμης συμφόρησης.

Στις μετρήσεις αυτές δεν μπορεί να χρησιμοποιηθεί ο μηχανισμός ελέγχου ροής καθότι ακολουθεί τις αρχές του end-to-end argument, με αποτέλεσμα να μην έχει τη δυνατότητα ούτε και την υποχρέωση να εγγυηθεί ότι το δίκτυο μπορεί να προωθήσει τα δεδομένα που κατευθύνονται στον παραλήπτη. Έτσι, ο αποστολέας πρέπει με κάποιον τρόπο να ελέγξει που και αν υπάρχει bottleneck στο δίκτυο ή στον παραλήπτη. Από αυτήν την άποψη, είναι λογικό να θεωρήσουμε ότι πρέπει να συνδυαστεί ο έλεγχος ροής με τον έλεγχο συμφόρησης του δικτύου. Αυτός είναι ο λόγος ύπαρξης της μεταβλητής του παραθύρου συμφόρησης (congestion window).

Το congestion window έχει (ή τουλάχιστον προσπαθεί να έχει) τιμή ανάλογη με τη χωρητικότητα του δικτύου. Σε αυτή την προσπάθεια συμβάλουν οι μηχανισμοί του πρωτοκόλλου που θα αναφερθούν αναλυτικά παρακάτω. Αν θεωρήσουμε ότι αυτοί οι μηχανισμοί δουλεύουν ιδανικά, δηλαδή ότι υπολογίζουν το congestion window επιτυχώς, τότε είναι δεδομένο ότι ο αποστολέας θα στείλει προς μετάδοση τέτοια ποσότητα δεδομένων, ώστε να αποφύγει και την υπερφόρτωση του παραλήπτη και τη συμφόρηση του δικτύου. Αυτό σημαίνει ότι θα στείλει ποσότητα δεδομένων ίση με το ελάχιστο εκ των δύο windows (advertised - congestion) πλην της ποσότητας δεδομένων που έχει ήδη στείλει αλλά για τις οποίες δεν έχει λάβει επιβεβαιώσεις (data in flight). Δηλαδή αν,

Advertised window (adwnd) είναι η ποσότητα δεδομένων που επιτρέπει ο παραλήπτης,

Congestion window (cwnd) είναι η ποσότητα δεδομένων που επιτρέπει το δίκτυο και

Sending window είναι η ποσότητα δεδομένων που στέλνουμε τελικά,

τότε ισχύει:

$$\begin{aligned} \text{Sending window} &= \min(\text{adwnd}, \text{cwnd}) - \text{Data_in_flight} \\ \text{Data_in_flight} &= \text{LBS} - \text{LBA} \end{aligned}$$

όπου LBS, LBA είναι το τελευταίο Byte που στάλθηκε και το τελευταίο Byte το οποίο επιβεβαιώθηκε αντίστοιχα.

1.3.3. Κριτήρια κατάστασης ελέγχου συμφόρησης

Για την αξιολόγηση των καταστάσεων ελέγχου συμφόρησης έχουν προταθεί ορισμένα κριτήρια και κάποιες μονάδες μέτρησης της απόδοσης, τα οποία επεξηγούνται περιληπτικά παρακάτω.

Αποδοτικότητα - Efficiency: Οι πόροι του εύρους ζώνης πρέπει να χρησιμοποιούνται πλήρως. Εφόσον το συνολικό throughput του συστήματος περιορίζεται από τη χωρητικότητα του κόμβου της ζεύξης, η αποδοτικότητα μπορεί να μετρηθεί με το να δούμε πόσο κοντά είναι η τιμή του throughput του συστήματος σε αυτήν της χωρητικότητας της ζεύξης.

Δικαιοσύνη - Fairness: Οι χρήστες του ίδιου καναλιού πρέπει να μοιράζονται εξίσου το εύρος ζώνης του κόμβου (bottleneck). Είναι ανεπιθύμητη η κατάσταση κατά την οποία κάποια flows καταναλώνουν περισσότερο από το fair-share του εύρους ζώνης που τους αντιστοιχεί σε βάρος κάποιων άλλων flows, ακόμα και αν σε αυτήν την περίπτωση η αποδοτικότητα είναι μέγιστη. Παρόλα αυτά, επειδή σε πραγματικά συστήματα δεν είναι δυνατή η τέλεια κατανομή του εύρους ζώνης, προτάθηκε ο Δείκτης Απόδοσης - Fairness Index για να καταγράψει τη δικαιοσύνη σε ένα δίκτυο.

Σύγκλιση - Convergence: Η σύγκλιση είναι το μέτρο εκείνο που αξιολογεί την ταχύτητα με την οποία ένα σύστημα έρχεται σε ισορροπία από οποιαδήποτε κατάσταση, δεδομένου ότι τα flows εισέρχονται στο σύστημα δυναμικά.

Ανταποκριτικότητα - Responsiveness: Είναι μια μονάδα μέτρησης η οποία αξιολογεί την ταχύτητα με την οποία τα εναπομείναντα flows εκμεταλλεύονται το έξτρα εύρος ζώνης που γίνεται διαθέσιμο απότομα, μετά την εγκατάλειψη του συστήματος από μερικά flows.

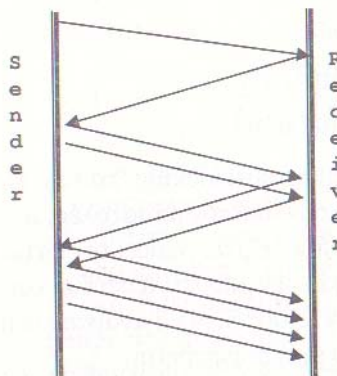
Ομαλότητα - Smoothness: Υπό κανονικές συνθήκες το σύστημα δε συγκλίνει σε μία απλή σταθερή κατάσταση. Αντί αυτού, το σύστημα φτάνει σε μία “δυναμική ισορροπία” (equilibrium) στην οποία ταλαντώνεται γύρω από μία βέλτιστη κατάσταση. Όσο μικρότερο είναι το μέγεθος αυτής της ταλάντωσης, τόσο πιο ομαλό είναι το ποσοστό αποστολής.

Καθυστέρηση Ουράς - Queuing Delay: Αν η ουρά (queue) μέσα στον buffer του κόμβου είναι συνεχώς άδεια, τότε δε γίνεται η καλύτερη δυνατή εκμετάλλευση της ζεύξης. Το bottleneck link είναι πλήρως εκμεταλλεύσιμο όταν η ουρά στον buffer του κόμβου δεν είναι άδεια. Από την άλλη, ένα μεγάλο μήκος ουράς σημαίνει υψηλή καθυστέρηση ουράς και συνεπώς υψηλή από άκρο - σε - άκρο καθυστέρηση, κάτι το οποίο μειώνει την πραγματική απόδοση διαλογικών εφαρμογών όπως το Telnet. Κανονικά το μέγεθος του buffer του bottleneck είναι αρκετά μεγάλο για να απορροφάει “εκρήξεις δεδομένων” (data bursts). Γενικά είναι επιθυμητό το μήκος της ουράς να είναι αρκετά μικρότερο από το μέγεθος του buffer. Το μέγεθος του μήκους της ουράς είναι βασικά δουλειά του Active Queue Management (AQM), το οποίο διαχειρίζεται το μήκος των πακέτων δεδομένων στην ουρά, με το να ρίχνει κάποια πακέτα όταν αυτό είναι απαραίτητο.

1.3.4. Μηχανισμοί Αργό Ξεκίνημα (Slow Start) και Αποφυγή Συμφόρησης (Congestion Avoidance) του TCP

Ένα από τα βασικά στοιχεία που διαθέτει το TCP, είναι ο μηχανισμός Self-clocking, σύμφωνα με τον οποίο ελέγχεται ο ρυθμός με τον οποίο στέλνει δεδομένα ο αποστολέας. Δηλαδή, καινούρια πακέτα μπορούν να σταλούν μόνο αν έχουν επιστρέψει επιβεβαιώσεις για τα πακέτα που είναι in flight. Με τον τρόπο αυτό, ακόμα και στην περίπτωση που δεν υπάρχει συμφόρηση στο δίκτυο, ο αποστολέας αποφεύγει να γεμίσει τον buffer του παραλήπτη. Αν υπάρχει συμφόρηση στο δίκτυο, δηλαδή το bottleneck δεν είναι στον παραλήπτη αλλά κάπου ενδιάμεσα, τότε και πάλι ο αποστολέας μειώνει το ρυθμό αποστολής δεδομένων.

Ένα άλλο σημαντικό σημείο στη συμπεριφορά του πρωτοκόλλου, είναι ότι προσπαθεί να αποφύγει τη δημιουργία συμφόρησης στην αρχή της επικοινωνίας, όταν ακόμα δεν γνωρίζει τη χωρητικότητα και τις δυνατότητες του δικτύου. Για να το πετύχει αυτό ο αποστολέας ξεκινάει στέλνοντας μικρή ποσότητα δεδομένων και σταδιακά προσπαθεί να την αυξήσει. Το TCP διαθέτει ένα μηχανισμό εκθετικής αύξησης της ποσότητας δεδομένων που στέλνονται, ο οποίος είναι γνωστός ως «Αργό Ξεκίνημα» (Slow Start). Η λειτουργία του Slow Start είναι η εξής: ο αποστολέας ξεκινάει τη μετάδοση δεδομένων με αρχικό cwnd ίσο με 1 και προσαυξάνει το cwnd κατά 1 πακέτο για κάθε επιβεβαίωση (ack) που παίρνει.



Σχήμα 1.6 Αργό Ξεκίνημα

Ουσιαστικά, διπλασιάζει το παράθυρο συμφόρησης κάθε φορά που παίρνει επιβεβαίωση για όλα τα πακέτα του cwnd. Ο αλγόριθμος που χρησιμοποιείται εδώ έχει ως στόχο την αποφυγή της συμφόρησης στην αρχή της σύνδεσης και την ανίχνευση της χωρητικότητας και των δυνατοτήτων του δικτύου με γρήγορο ρυθμό (γι' αυτό και αυξάνει τη ροή δεδομένων εκθετικά).

Όπως γίνεται φανερό, ο αλγόριθμος αυτός δεν είναι και τόσο “αργός”, όπως αφήνεται να εννοηθεί εξ' αρχής από το όνομα του. Το όνομα του αλγορίθμου αναφέρεται στην επιδίωξη του να συμπεριφερθεί συντηρητικά στην αρχή και να μη δημιουργήσει συμφόρηση εφόσον δεν γνωρίζει την κατάσταση του δικτύου. Μία διαφορετική προσέγγιση θα ήταν να ξεκινήσει ο αποστολέας με μεγάλη ποσότητα δεδομένων και σταδιακά να τη μειώνει. Με αυτόν τον τρόπο το πρωτόκολλο θα οδηγούσε το δίκτυο σε σοβαρό επίπεδο συμφόρησης από πολύ νωρίς. Επομένως, αν και ο αλγόριθμος Slow Start είναι πιο αργός στην ανίχνευση της χωρητικότητας του δικτύου από κάποιον αλγόριθμο που χρησιμοποιεί προσθετική αύξηση του cwnd, είναι πιο αργός

από έναν αλγόριθμο που ξεκινά με ένα μεγάλο cwnd και κατόπιν αυξάνει ή μειώνει τη ροή δεδομένων ανάλογα με την κατάσταση του δικτύου.

Όπως είναι φυσικό, η εκθετική αύξηση δεν μπορεί να συνεχιστεί επ' άπειρον για όλη τη διάρκεια της σύνδεσης, καθώς ο διπλασιασμός του παραθύρου, θα οδηγήσει σε κάποια φάση σε μετάδοση μεγάλων ποσοτήτων δεδομένων κάτι που θα προκαλέσει σοβαρή συμφόρηση στο δίκτυο. Κρίθηκε λοιπόν απαραίτητη η εισαγωγή ενός κατωφλίου (ορίου), το οποίο θα σηματοδοτήσει μία αλλαγή τακτικής. Το κατώφλι αυτό είναι το γνωστό slow start threshold (ssthresh) και σημαίνει το τέλος της εκθετικής αύξησης και την αρχή μίας άλλης φάσης που ονομάζεται αποφυγή συμφόρησης (congestion avoidance) και βασίζεται στην προσθετική αύξηση του παραθύρου του αποστολέα. Στην προσθετική αύξηση δεν γίνεται αύξηση ενός πακέτου για κάθε επιβεβαίωση, αλλά αύξηση ενός πακέτου για κάθε παράθυρο συμφόρησης (cwnd).

Είναι λογικό να σκεφτούμε και πάλι ότι σε κάποια φάση θα επικρατήσει κατάσταση σοβαρής συμφόρησης στο δίκτυο, παρά τη σχετική μείωση του ρυθμού ροής. Εξάλλου τη στιγμή που μία εφαρμογή μπαίνει στη φάση της προσθετικής αύξησης, και συνεπώς ήδη μεταφέρει αρκετά μεγάλο όγκο δεδομένων, κάποια άλλη μπορεί να είναι ακόμα στη φάση της εκθετικής αύξησης με αποτέλεσμα το φορτίο δεδομένων που πρέπει να μεταφερθεί από το δίκτυο να αυξάνεται με μεγάλους ρυθμούς. Με άλλα λόγια, το σημείο συμφόρησης αλλάζει δυναμικά και επομένως ο αποστολέας θα πρέπει να το παρακολουθεί συνεχώς. Δε μπορεί η ανίχνευση του να καθορίσει τη στρατηγική των πρωτοκόλλων παρά μόνο για την επόμενη φορά που ο αποστολέας θα στείλει δεδομένα. Επίσης, η ανίχνευση του σημείου συμφόρησης δεν αποκρυπτογραφεί πραγματικά τη χωρητικότητα του δικτύου, καθότι όπως είπαμε παραπάνω υπάρχουν κι άλλες εφαρμογές οι οποίες τρέχουν και λειτουργούν με τον ίδιο τρόπο. Έτσι, δεν είναι ποτέ γνωστό αν η συμφόρηση προκλήθηκε από την προσθετική ή εκθετική αύξηση μιας εφαρμογής ή εάν προκλήθηκε από έναν αριθμό καινούριων εφαρμογών που ξεκίνησαν μία σύνδεση και άρχισαν να χρησιμοποιούν και αυτές το δίκτυο επικοινωνίας.

Όλες αυτές οι αμφιβολίες και οι αβεβαιότητες οδηγούν στην επιλογή μίας συντηρητικής στρατηγικής έτσι ώστε να ελεγχθεί η συμφόρηση και να αποφευχθεί μία κατάσταση καταστροφικής συμφόρησης (congestive collapse). Η αρχική έκδοση του TCP, το Old Tahoe, ακολουθούσε στρατηγική υποχώρησης στο αρχικό cwnd, με ταυτόχρονη μείωση του ssthresh στο μισό του προηγούμενου cwnd, υποχρεώνοντας έτσι τον αποστολέα να μειώσει δραστικά τη ροή δεδομένων. Οι νεότερες εκδόσεις του πρωτοκόλλου (Tahoe, Reno κτλ.) προσπαθούν να ελέγξουν τη συμφόρηση με τέτοιο τρόπο, ώστε και να αποφεύγεται η καταστροφική συμφόρηση και να αξιοποιούνται οι δικτυακοί πόροι. Όπως έχει ήδη αναφερθεί οι έξτρα μηχανισμοί που χρησιμοποιούν το TCP Tahoe και το TCP Reno είναι οι Fast Retransmit και Fast Recovery.

1.3.5. Μηχανισμοί Fast Retransmit και Fast Recovery

Ο μηχανισμός Fast Retransmit δίνει τη δυνατότητα στον αποστολέα να ξαναστείλει ένα πακέτο χωρίς να χρειάζεται να περιμένει τη λήξη του timeout. Η λογική του Fast Retransmit ξεκινάει από την εξής παρατήρηση: Όταν το δίκτυο δε βρίσκεται σε κατάσταση καταστροφικής συμφόρησης, επιτρέπει σε ορισμένα πακέτα να προωθηθούν στους παραλήπτες τους. Κάποια πακέτα πιθανώς να χαθούν στη διαδρομή, λόγω έλλειψης χώρου στους καταχωρητές. Επομένως, ο αποστολέας θα παραλάβει κάποιες επιβεβαιώσεις για τα πακέτα που παραλήφθηκαν επιτυχώς, με τη μορφή DACKs (Duplicate Acknowledgments) οι οποίες ουσιαστικά θα επισημαίνουν το επόμενο πακέτο που περιμένει ο παραλήπτης, δίνοντας τη δυνατότητα στον αποστολέα να το ξαναστείλει χωρίς να χρειάζεται να περιμένει τη λήξη του timeout. Ο αριθμός των DACKs που ενεργοποιεί την επαναποστολή των πακέτων πριν τη λήξη του timeout, τίθεται συνήθως στο 3, αφήνοντας έτσι ένα περιθώριο στα πακέτα που απλά καθυστέρησαν να φτάσουν, αφού η επαναποστολή πακέτων που έχουν ήδη φτάσει στον παραλήπτη δεν έχει νόημα και το μόνο που καταφέρνει είναι να αυξήσει το overhead.

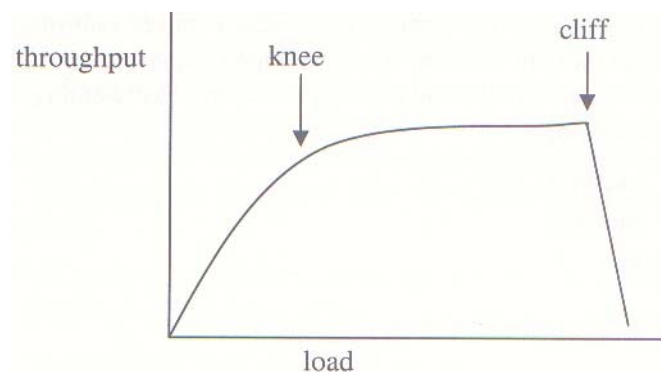
Σαν συνέχεια του μηχανισμού Fast Retransmit, έρχεται ο μηχανισμός Fast Recovery. Αυτός στηρίζεται στη λογική της “περιορισμένης συμφόρησης”, η οποία δικαιολογεί μία περισσότερο επιθετική στρατηγική. Η παραλαβή DACKs, δείχνει ότι οι συνθήκες του δικτύου έχουν βελτιωθεί, έστω και λίγο και επιτρέπουν τη μετάδοση κάποιων πακέτων στο δίκτυο. Με αυτό σαν δεδομένο, δεν είναι ανάγκη να ακολουθηθεί μία ακραία στρατηγική, όπως η ρύθμιση του παραθύρου στην αρχική του τιμή. Έτσι ο μηχανισμός Fast Recovery δεν ρυθμίζει το παράθυρο στην αρχική του τιμή, αλλά στο μισό της προηγούμενης τιμής του. Ομοίως υποχωρεί και το ssthresh, κάτι που αναγκάζει τον αποστολέα να περάσει αμέσως στη φάση της προσθετικής αύξησης, σε αντίθεση με τον αρχικό μηχανισμό που ρυθμίζει το ssthresh στη μισή της προηγούμενης τιμής του και το cwnd στην αρχική του τιμή, όπου ο αποστολέας μπαίνει από την αρχή σε φάση εκθετικής αύξησης.

Όπως αναφέρθηκε και παραπάνω, η έκδοση του TCP που διαθέτει μόνο τους μηχανισμούς Slow Start και Congestion Avoidance ονομάζεται Old Tahoe. Η έκδοση που διαθέτει επιπλέον το μηχανισμό Fast Retransmit ονομάζεται TCP Tahoe [8]. Η έκδοση που εκτός από το Fast Retransmit διαθέτει και το Fast Recovery ονομάζεται TCP Reno [9]. Όλες αυτές οι εκδόσεις συνυπάρχουν στο Διαδίκτυο και φυσικά μπορούν να επικοινωνούν μεταξύ τους. Αυτό γίνεται επειδή ο έλεγχος συμφόρησης είναι μία λειτουργία για την οποία υπεύθυνος είναι αποκλειστικά και μόνο ο αποστολέας και δε χρειάζεται την κωδικοποίηση στην επικεφαλίδα του παραλήπτη για να πάρει απόφαση σχετικά με τον όγκο των δεδομένων που θα μεταδώσει προς το δίκτυο.

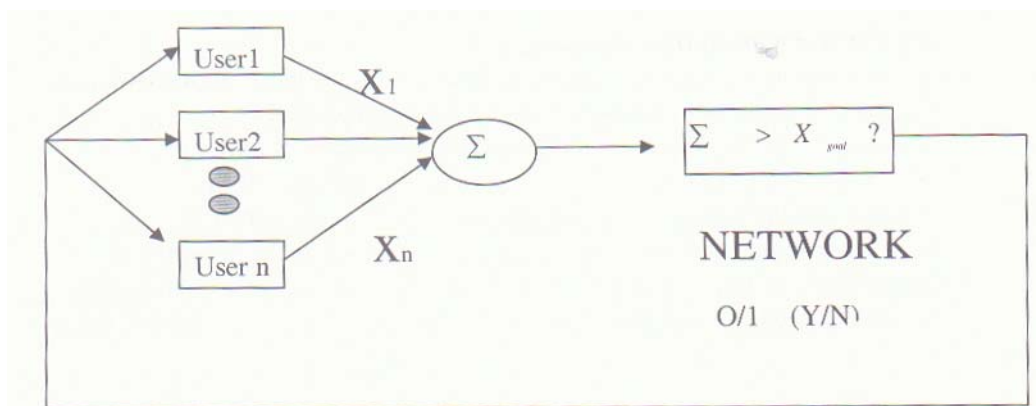
1.3.6. Το Μοντέλο Ανάλυσης για τον Έλεγχο Συμφόρησης

Στην ενότητα αυτή δίνεται μια περιγραφή της συμπεριφοράς του throughput ως συνάρτηση του φορτίου που υπάρχει στο δίκτυο, όπως φαίνεται στο Σχήμα 1.7. Όταν ο φόρτος στο δίκτυο είναι μικρός, το throughput του συστήματος αυξάνει αναλογικά προς το φόρτο. Αφού ο φόρτος φτάσει τη χωρητικότητα του κόμβου (bottleneck), το throughput σταματάει να αυξάνεται. Ο όρος knee χρησιμοποιείται για να χαρακτηρίσει αυτό το φόρτο. Όταν ο φόρτος ξεπεράσει το knee, η ουρά αρχίζει να

χτίζεται μέσα στον buffer του δρομολογητή. Αν ο φόρτος συνεχίσει να αυξάνεται πέρα από ένα σημείο το οποίο καλείται cliff, τα πακέτα θα αρχίσουν να ρίχνονται και το throughput θα μειωθεί απότομα (καταστροφική συμφόρηση). Σ' αυτό το σημείο λέμε ότι το δίκτυο έχει συμφόρηση. Όσον αφορά το χρόνο απόκρισης, στην αρχή αυξάνεται αργά μαζί με το φόρτο. Όταν ο φόρτος ξεπεράσει το σημείο knee και αρχίσει να δημιουργείται η ουρά, ο χρόνος απόκρισης αυξάνεται γραμμικά έως ότου, τελικά υπερχειλίσει η ουρά (το σύστημα φτάνει στο σημείο cliff), όπου ο χρόνος απόκρισης αυξηθεί δραματικά. Σημειώστε ότι το throughput του συστήματος μπορεί να είναι χαμηλό όταν ο φόρτος είναι κάτω από το σημείο knee (δεν γίνεται καλή εκμετάλλευση του διαθέσιμου εύρους ζώνης), πιθανώς λόγω πολύ συντηρητικής στρατηγικής. Το throughput του συστήματος μπορεί επίσης να είναι χαμηλό αν ο φόρτος παραμένει συνέχεια στο σημείο knee, κάτι που σημαίνει ότι η στρατηγική είναι πολύ επιθετική. Ένας αλγόριθμος ελέγχου συμφόρησης μειώνει το ποσοστό αποστολής στην περίπτωση συμφόρησης, κρατώντας το δίκτυο σε λειτουργία κάτω από το σημείο cliff (όπου εμφανίζεται η συμφόρηση), και πάνω από το σημείο knee, έτσι ώστε να γίνεται πλήρης εκμετάλλευση του εύρους ζώνης του κόμβου (bottleneck). Το σενάριο αποφυγής συμφόρησης, όπως αυτό μπορεί να καθοριστεί από το σενάριο ελέγχου συμφόρησης, προσπαθεί να κρατήσει το σύστημα σε λειτουργία πάνω από το σημείο knee, αλλά αρκετά κάτω από το σημείο cliff έτσι ώστε να μειώσει το χρόνο απόκρισης.



Σχήμα 1.7. Το throughput ως Συνάρτηση του Φορτίου του δικτύου.



Σχήμα 1.8. Το Μοντέλο του Συστήματος Ελέγχου

Στο Σχήμα 1.8 παρουσιάζεται ένα μοντέλο συστήματος ελέγχου η χρηστών οι οποίοι μοιράζονται το δίκτυο. Εδώ υποτίθεται μία λειτουργία διακριτού χρόνου με το χρόνο να διαιρείται σε μικρές μονάδες-θυρίδες χρόνου (time slots). Οι χρήστες θέτουν τα επίπεδα φόρτου τους σε ένα time slot σύμφωνα με την ανάδραση του δικτύου που παραλαμβάνεται από το προηγούμενο time slot. Πιο συγκεκριμένα αν το φορτίο του χρήστη i^{th} στο χρόνο t συμβολίζεται ως $x_i(t)$, τότε το συνολικό φορτίο του συστήματος στον κόμβο (bottleneck) θα είναι $\Sigma x_i(t)$. Ο κόμβος (bottleneck) του δικτύου συγκρίνει το φορτίο που παραλαμβάνει με ένα φορτίο στόχο X_{goal} , και στέλνει μία δυαδική ανάδραση $y(t)$ στους αποστολείς, επιδεικνύοντας αύξηση (if $\Sigma x_i(t) < X_{\text{goal}}$) ή μείωση (if $\Sigma x_i(t) > X_{\text{goal}}$) φορτίου. Υποθέτοντας ότι ο αποστολέας ρυθμίζει το ποσοστό αποστολής γραμμικά σύμφωνα με την ανάδραση, έχουμε:

$$\begin{aligned} x_i(t+1) &= a_i + b_i x_i(t) \text{ να } y(t) = 0 \text{ τότε έχουμε αύξηση} \\ x_i(t+1) &= a_D + b_D x_i(t) \text{ να } y(t) = 1 \text{ τότε έχουμε μείωση} \end{aligned}$$

όπου a_i, b_i, a_D, b_D είναι σταθερές.

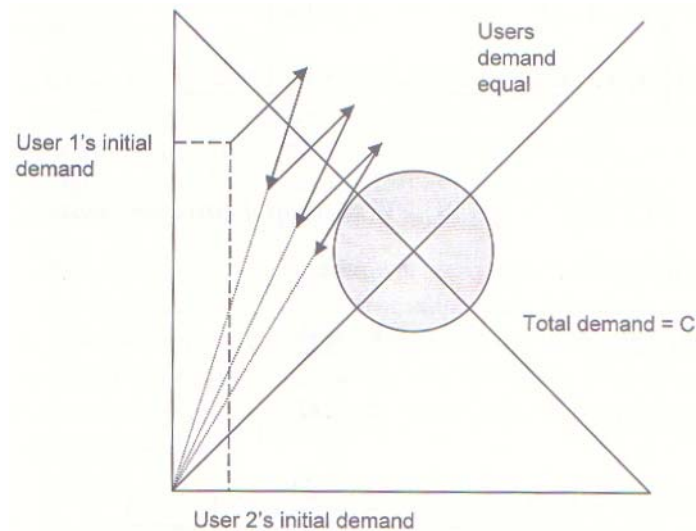
Ένα μειονέκτημα αυτού του μοντέλου είναι ότι υποθέτει ίσα RTT για όλα τα flows, κάτι το οποίο δεν αληθεύει σε πραγματικά συστήματα.

1.3.7. Προσθετική αύξηση / Πολλαπλασιαστική μείωση (AIMD - Additive Increase/Multiplicative Decrease)

Έχει αποδειχθεί πως για να επιτευχθεί σύγκλιση σε μια αποδοτική και δίκαιη κατάσταση, ανεξάρτητα από την αρχική κατάσταση του δικτύου, αρκεί ένας αλγόριθμος Προσθετικής Αύξησης / Πολλαπλασιαστικής Μείωσης. Σύμφωνα με τον αλγόριθμο, κάθε αποστολέας αυξάνει σταδιακά το φορτίο του με μία σταθερά a_i σε κάθε time slot (αθροιστική αύξηση), μέχρις ότου λάβει ειδοποίηση συμφόρησης. Μετά την ειδοποίηση, κάθε αποστολέας μειώνει απότομα το φορτίο του σύμφωνα με μία αναλογία b_D σε ένα time slot (πολλαπλασιαστική μείωση). Εξαιτίας της απότομης μείωσης του συνολικού φορτίου, η ανάδραση του δικτύου δεν στέλνει ειδοποίηση ανάδρασης στο επόμενο time slot, και όλοι οι αποστολείς μπαίνουν ξανά στο στάδιο αθροιστικής αύξησης. Καθορίζεται ως εποχή συμφόρησης - congestion epoch το χρονικό διάστημα μεταξύ δύο περιστατικών συμφόρησης. Διαισθητικά καταλαβαίνουμε ότι κατά τη διάρκεια κάθε εποχής συμφόρησης, το ποσό κατά το οποίο αυξάνεται το φορτίο κατά την αθροιστική αύξηση, είναι το ίδιο για όλους τους χρήστες. Εντούτοις, όσο μεγαλύτερο είναι το ποσό δεδομένων που στέλνει ένας χρήστης στο δίκτυο, τόσο μεγαλύτερη είναι η μείωση στο φορτίο όταν εφαρμόζεται πολλαπλασιαστική μείωση μετά από ειδοποίηση συμφόρησης.

Έστω ότι σε ένα δίκτυο έχουμε δύο χρήστες 1 και 2. Στο παρακάτω σχήμα 1.9 βλέπουμε τους άξονες x και y των χρηστών 1 και 2 αντίστοιχα. Οι χρήστες χρησιμοποιούν διαφορετικά παράθυρα, πράγμα που στην πράξη σημαίνει ότι ο πρώτος χρήστης με το μεγαλύτερο παράθυρο ξεκίνησε νωρίτερα την επικοινωνία μέσω του δικτύου και το παράθυρο του μεγάλωσε κατά τη διάρκεια της προσθετικής αύξησης. Ο δεύτερος χρήστης λοιπόν μόλις ξεκίνησε την αποστολή δεδομένων και το παράθυρο του είναι το αρχικό παράθυρο. Ο διαγώνιος άξονας του σχήματος είναι ο άξονας $x = y$ και ονομάζεται άξονας δικαιοσύνης γιατί πάνω στον άξονα αυτόν βρισκόμαστε μόνο όταν τα παράθυρα των δυο χρηστών είναι ίσα. Ο κάθετος προς

αυτόν άξονας ονομάζεται άξονας απόδοσης και δηλώνει ουσιαστικά τη χωρητικότητα του δικτύου. Όταν ξεπερνάμε αυτόν τον άξονα, τα πακέτα μας χάνονται και το δίκτυο σηματοδοτεί τη συμφόρηση. Πάνω σε αυτό τον άξονα το σύστημα αγγίζει τη μέγιστη απόδοση. Θέλουμε λοιπόν να πλησιάσουμε τη μέγιστη απόδοση, βρίσκόμενοι όμως πάνω στον άξονα της δικαιοσύνης. Θέλουμε δηλαδή να κινηθούμε περίπου στο εσωτερικό του σκιασμένου κύκλου του σχήματος 2.9.



Σχήμα 1.9. Γεωμετρική παράσταση της σύγκλισης με τον αλγόριθμο AIMD

Στο σημερινό Διαδίκτυο, ο έλεγχος συμφόρησης υλοποιείται στη μεριά του TCP αποστολέα. Ο έλεγχος συμφόρησης ολοκληρώνεται με τη ρύθμιση του μεγέθους του παραθύρου συμφόρησης (cwnd), το οποίο περιορίζει το ποσό των δεδομένων το οποίο ο αποστολέας μπορεί να στείλει στο δίκτυο πριν λάβει επιβεβαίωση (ACK) η οποία να καλύπτει τα ανεπιβεβαίωτα δεδομένα, ή ελέγχει τον αριθμό των πακέτων που μεταφέρθηκαν μέσα σε ένα RTT. Για τον έλεγχο ροής, το advertised window (rwnd) του παραλήπτη, είναι ένας περιορισμός στη μεριά του παραλήπτη για το ποσό των σημαντικών δεδομένων. Το παράθυρο αποστολής στη μεριά του αποστολέα, πρέπει να είναι το ελάχιστο των cwnd και adwnd. Οι ρυθμίσεις του παραθύρου συμφόρησης του πρότυπου TCP υιοθετούν τη στρατηγική του AIMD και ο αποστολέας λαμβάνει την απώλεια πακέτων ως ειδοποίηση συμφόρησης. Οι πραγματικές υλοποιήσεις του TCP είναι πιο περίπλοκες από αυτές του απλού μοντέλου του AIMD. Οι πιο γνωστές και ευρέως χρησιμοποιούμενες εκδόσεις του TCP είναι οι: Tahoe, Reno, SACK .

1.4. Εκδόσεις του πρωτοκόλλου μεταφοράς TCP

1.4.1. TCP Tahoe

Ο αλγόριθμος ελέγχου συμφόρησης που χρησιμοποιείται στο Tahoe [10], [11] περιλαμβάνει τους μηχανισμούς Slow Start, Congestion Avoidance και Fast Retransmit. Ο μηχανισμός slow start χρησιμοποιείται στην αρχή της μετάδοσης ή κατά τη διάρκεια ανάκαμψης μετά από απώλεια που ανιχνεύθηκε από λήξη του

timeout. Το παράθυρο συμφόρησης ξεκινάει με μέγεθος 2 φορές το μέγιστο μέγεθος πακέτου (maximum segment size - MSS) και διπλασιάζεται σε κάθε RTT μέχρι ένα κατώφλι (ssthresh) ή μέχρι να λήξει το timeout. Στην πρώτη περίπτωση, αναλαμβάνει ο μηχανισμός αθροιστικής αύξησης (congestion avoidance). Το παράθυρο συμφόρησης αυξάνεται κατά ένα πακέτο σε κάθε RTT ($\alpha = 1$). Κάθε φορά που συμβαίνει λήξη του timeout, το ssthresh τίθεται στο μισό του μεγέθους που έχει το παράθυρο εκείνη τη στιγμή και στη συνέχεια το παράθυρο συμφόρησης μειώνεται σε 2 φορές το μέγιστο μέγεθος πακέτου MSS. Με το μηχανισμό fast retransmit, τρία επιτυχημένα duplicate acknowledgments (Dacks) ενεργοποιούν επαναποστολή δεδομένων πριν εφαρμοστεί slow start, χωρίς να περιμένουν τη λήξη του timeout. Σημειώνεται, τέλος, ότι το TCP Tahoe δεν παρουσιάζει καλή απόδοση όταν έχουμε πολλαπλές απώλειες πακέτων στο ίδιο παράθυρο δεδομένων.

1.4.2. TCP Reno

Το TCP Reno [10], [12] έχει εισάγει σημαντικές βελτιώσεις σε σχέση με το Tahoe, αλλάζοντας τον τρόπο με τον οποίο αντιδρά στην ανίχνευση απώλειας μέσω των διπλών επιβεβαιώσεων. Σαν συνέχεια του fast retransmit, το TCP Reno χρησιμοποιεί επιπλέον το μηχανισμό fast recovery. Ο αλγόριθμος fast retransmit στέλνει το πακέτο που λείπει και στη συνέχεια ο αλγόριθμος fast recovery αναλαμβάνει τη μεταφορά δεδομένων μέχρις ότου φτάσει ένα non-duplicate ACK. Κατά τη διάρκεια του fast recovery, το παράθυρο συμφόρησης αυξάνεται κατά ένα πακέτο για κάθε duplicate ACK. Ο μηχανισμός fast recovery ενεργοποιείται από τα πακέτα δεδομένων που βρίσκονται στον buffer του παραλήπτη και φτάνουν εκτός σειράς. Στο τέλος του fast recovery, το παράθυρο συμφόρησης τίθεται στο μισό του μεγέθους που είχε όταν ξεκίνησε το στάδιο fast recovery ($\beta = 0.5$) και δεν ενεργοποιείται ο μηχανισμός slow start εφόσον το πακέτο που ξαναστάλθηκε επιβεβαιωθεί. Έτσι σε κατάσταση ισορροπίας, ο έλεγχος συμφόρησης του Reno ακολουθεί τον αλγόριθμο AIMD με $\alpha = 1$ και $\beta = 0.5$, εφόσον δεν υπάρχει μεγάλη απώλεια δεδομένων και μπορεί να ανιχνευθεί από 3 dacks πριν λήξει το timeout. Η επιλογή των α και β είναι εμπειρική.

1.4.3. TCP New Reno

Το TCP New Reno [9] είναι μια τροποποίηση του TCP Reno, η οποία χρησιμοποιείται σήμερα ευρέως στο χώρο του Διαδικτύου. Το TCP New Reno εντόπισε πως, αν το επιλεγόμενο Slow Start Threshold (ssthreshold) έχει μικρή τιμή, τότε η εκθετική αύξηση του παραθύρου συμφόρησης (κατά τη διάρκεια της φάσης του Slow Start) θα σταματήσει πολύ νωρίς για να δώσει τη θέση της στη σαφώς πιο αργή γραμμική αύξηση. Από την άλλη, αν το επιλεγόμενο Slow Start Threshold (ssthreshold) έχει μεγαλύτερη τιμή απ' ότι θα έπρεπε, τότε ο αποστολέας θα υπερφορτώσει τους buffers του δικτύου προκαλώντας συμφόρηση. Λαμβάνοντας υπόψη ότι η φάση του Slow Start και ο χρόνος που αυτή διαρκεί είναι καθοριστικός για τη συνολική απόδοση του πρωτοκόλλου, καταλαβαίνει εύκολα κανείς γιατί κρίθηκαν ως απαραίτητες οι επεμβάσεις του TCP New Reno. Το TCP New Reno έχει μια μέθοδο να υπολογίζει ένα Slow Start Threshold ιδανικού μεγέθους με τη βοήθεια του εκτιμώμενου bandwidth.

Όταν υπάρχει απώλεια πολλών πακέτων από ένα παράθυρο δεδομένων, η επιβεβαίωση που θα έρθει στο τέλος του fast recovery από το πακέτο που ξαναστάλθηκε θα επιβεβαιώσει κάποια, αλλά όχι όλα τα πακέτα που μεταφέρθηκαν πριν το fast retransmit. Αυτό καλείται partial acknowledgment. Εφόσον κάθε φορά που ενεργοποιείται ο μηχανισμός fast retransmit οδηγεί σε επαναποστολή ενός χαμένου πακέτου δεδομένων, κάποια duplicate ACKs θα ακολουθήσουν το partial acknowledgment, οι μηχανισμοί fast retransmit και fast recovery θα ενεργοποιηθούν ξανά. Καθότι ένα partial acknowledgment είναι ένδειξη πολλών χαμένων πακέτων, το TCP NewReno δε βγαίνει από το στάδιο fast recovery και συνεχίζει να ξαναστέλνει τα πολλαπλά χαμένα πακέτα μέχρι να σταματήσουν να εμφανίζονται partial acknowledgments. Παρόλα αυτά, όταν υπάρχει απώλεια πολλαπλών πακέτων από ένα παράθυρο του NewReno, το TCP μπορεί να ξαναστείλει το πολύ ένα χαμένο πακέτο δεδομένων σε κάθε RTT, επειδή το επόμενο πακέτο που θα ξανασταλθεί πρέπει να προκύψει από partial acknowledgment.

1.4.4. TCP SACK

Επειδή η φύση της επιβεβαίωσης του TCP είναι cumulative, όταν έρχεται ένα duplicate ACK, δεν έχουμε καμία πληροφορία για το ποιο πακέτο ενεργοποίησε αυτό το duplicate ACK. Στο στάδιο του fast recovery, ο αποστολέας μπορεί να ξαναστείλει πακέτα τα οποία έχουν ήδη φτάσει επιτυχώς στον παραλήπτη (Tahoe), ή να ξαναστείλει το πολύ ένα χαμένο πακέτο σε κάθε RTT (Reno, NewReno). Η επιλογή της επιλεκτικής επιβεβαίωσης (selective acknowledgment - SACK) [10], [11], [13] επιτρέπει στον παραλήπτη να ενημερώσει τον αποστολέα για τα πακέτα τα οποία έχουν φτάσει επιτυχώς, και συνεπώς να του υποδείξει έμμεσα ποια πακέτα έχουν πραγματικά χαθεί και πρέπει να ξανασταλθούν. Όπως το Reno, έτσι και το TCP SACK, κάνει πιο γρήγορη επαναποστολή πακέτων και εισέρχεται στο fast recovery μόλις λάβει duplicate acknowledgments. Κατά τη διάρκεια του fast recovery, αν ο αποστολέας μπορεί να στείλει ένα πακέτο, θα στείλει το επόμενο πακέτο που είναι στη λίστα από τα πακέτα που ο παραλήπτης έχει αναφέρει ως χαμένα. Αν δεν υπάρχουν τέτοια πακέτα, τότε ο αποστολέας στέλνει ένα νέο πακέτο. Ο αποστολέας εξέρχεται από τη φάση του fast recovery όταν φτάνει ένα ACK το οποίο επιβεβαιώνει όλα τα δεδομένα τα οποία ήταν σημαντικά στην αρχή του fast recovery. Αν προκύψει λήξη του timeout από ένα πακέτο που έχει ξανασταλθεί, ο αποστολέας στέλνει το πακέτο ξανά και μπαίνει σε slow start. Το TCP SACK μπορεί να ανταπεξέλθει καλύτερα σε απώλειες πολλαπλών πακέτων από ένα παράθυρο χωρίς να περιμένει τη λήξη του timeout.

1.4.5. TCP Vegas

Τα TCP Tahoe και Reno έχουν μια αντιδραστική στρατηγική ελέγχου συμφόρησης. Με αυτό εννοούμε ότι το TCP πρέπει να αντιληφθεί συμφόρηση για να ελέγξει το throughput. Αυτό μπορεί να έχει ως αποτέλεσμα αρνητικές επιπτώσεις στην απόδοση του δικτύου. Το TCP Vegas [14], [15] υιοθετεί μια εναλλακτική στρατηγική. Προσπαθεί να προβλέψει μέσω μετρήσεων πότε θα προκληθεί συμφόρηση και προσαρμόζει ανάλογα το παράθυρο.

Το TCP Vegas μετρά το ρυθμό αποστολής δεδομένων και συγκρίνει την απόδοση που μετρά κατά τακτά χρονικά διαστήματα με την αναμενόμενη απόδοση. Η αναμενόμενη απόδοση υπολογίζεται με βάση τις μετρήσεις RTT. Για την ακρίβεια, το TCP Vegas ορίζει τη μονάδα BaseRTT ως μικρότερο από όλα τα RTT, που αντιπροσωπεύει την ταχύτητα του δικτύου χωρίς την παρουσία συμφόρησης. Εδώ πρέπει να επισημάνουμε ότι δεν έχουμε καμία εγγύηση για το πότε, και αν πραγματικά, θα έχει την ευκαιρία ο αποστολέας να καταγράψει το πραγματικό BaseRTT. Το πρωτόκολλο, παρόλα αυτά, δοκιμάστηκε στην πράξη και φαίνεται να πετυχαίνει καλή απόδοση.

Ένα ζήτημα με το Vegas είναι η αποτελεσματικότητα του σε σχέση με το δείκτη δικαιοσύνης και την ταχύτητα σύγκλισης σε κατάσταση ισορροπίας. Στο σχεδιασμό πρωτοκόλλων που βασίζονται σε μετρήσεις πρέπει πάντα να λαμβάνει κανείς υπόψη την πιθανότητα απόκλισης των μετρήσεων από τις πραγματικές τιμές. Επιπλέον, η δυναμική των συστημάτων με πολλούς χρήστες είναι τέτοια, που δεν μας επιτρέπει στην πράξη να αποκλείσουμε το ενδεχόμενο λανθασμένων μετρήσεων. Για να θεμελιώσουμε τον ισχυρισμό μας, αρκεί να θεωρήσουμε ότι υπάρχει ενδεχόμενο κατά τη διάρκεια της μέτρησης να εισέλθουν καινούριοι χρήστες και να ανατρέψουν τα δεδομένα μας.

Σύμφωνα με μετρήσεις που έχουν γίνει, το Vegas επιτυγχάνει καλύτερη απόδοση κατά τη μεταφορά από το Reno και το Tahoe. Εντούτοις, παρατηρείται ότι το TCP Vegas δεν μπορεί να εγγυηθεί δικαιοσύνη. Το TCP Vegas δεν μπορεί να συμβάλλει στην ανίχνευση λάθους και την ανάκαμψη από αυτό σε ενσύρματα/ασύρματα δίκτυα.

1.4.6. TCP Real

Το TCP Real [16], [17] είναι ένα πειραματικό πρωτόκολλο το οποίο αποτελεί μία προσέγγιση που βασίζεται στον παραλήπτη (receiver - oriented). Το πρωτόκολλο διαθέτει μία στρατηγική μετάδοσης δεδομένων βασισμένη σε συνεχή μετρήσεις, η οποία εμπλουτίζει την «τυφλή» στρατηγική αύξησης / μείωσης του παραθύρου συμφόρησης του πρότυπου TCP. Σκοπός του παραλήπτη είναι να παρακολουθεί με ακρίβεια το επίπεδο της συμφόρησης και την πιθανή απώλεια δεδομένων. Ένα κύμα αποτελείται από έναν αριθμό από πακέτα δεδομένων σταθερού μεγέθους τα οποία στέλνονται από άκρο σε άκρο. Εξ' ορισμού, ένα TCP κύμα μεταφέρει το μέγεθος του παραθύρου συμφόρησης με μία πρόσθετη ικανότητα: γίνεται γνωστό και στις δύο μεριές. Διαθέτοντας ένα μηχανισμό ανταλλαγής δεδομένων ο οποίος κάνει γνωστό και στις δύο μεριές το μέγεθος του κύματος γίνεται πιο εύκολη η εκτίμηση του επιπέδου της συμφόρησης από τον παραλήπτη. Το κενό μεταξύ παραλαβής πακέτων δεδομένων και ο χρόνος παράδοσης του κύματος αποτελούν πληροφορίες που υποδηλώνουν την πιθανή εμφάνιση συμφόρησης και συνεπώς κάνουν δυνατή την ταξινόμηση του λάθους και την επιλογή της κατάλληλης στρατηγικής ανάκαμψης.

Πιο συγκεκριμένα, ο παραλήπτης παρακολουθεί το ρυθμό με τον οποίο κυκλοφορούν τα κύματα μέσα στο δίκτυο. Κατά τη διάρκεια συμφόρησης, ο ρυθμός αυτός θα παρουσιάζει μεγάλες διακυμάνσεις, αφού η κυκλοφορία ενός πακέτου ή κύματος εξαρτάται σε μεγάλο βαθμό από τη θέση που θα πάρει στην ουρά του buffer του router το συγκεκριμένο πακέτο-κύμα. Στην αντίθετη περίπτωση, όπου συμβαίνει ένα ασύρματο λάθος, δεν θα επηρεαστεί το ποσοστό παραλαβής δεδομένων. Αυτές οι

παρατηρήσεις που γίνονται με τη βοήθεια των κυμάτων βοηθούν στην υλοποίηση ενός μηχανισμού ο οποίος μπορεί να ξεχωρίσει τα λάθη που οφείλονται σε συμφόρηση και σε αυτά που οφείλονται σε ασύρματα-ετερογενή δίκτυα. Η στρατηγική η οποία ακολουθεί το πρωτόκολλο επαληθεύεται στο επόμενο RTT, όπου γίνεται σύγκριση της παρούσας εκτίμησης του ρυθμού ανταλλαγής δεδομένων με το ρυθμό που υπήρχε πριν.

Με τον τρόπο αυτό το TCP Real πραγματοποιεί ταξινόμηση του λάθους και εκτίμηση του επιπέδου της συμφόρησης και με αυτό τον τρόπο μπορεί να επιλέξει κάθε φορά μία στρατηγική ανάκαμψης η οποία βασίζεται στη φύση του λάθους. Ο παραλήπτης ξεχωρίζει τις απώλειες πακέτων δεδομένων από τη ρύθμιση του παραθύρου της συμφόρησης. Κάθε φορά που ο παραλήπτης παρατηρεί παραλαβή δεδομένων με μεγάλο ρυθμό, αλλά με μικρό ποσοστό αλλοίωσης, τότε αυτό είναι ένδειξη τυχαίου, ασύρματου λάθους κάτι που κάνει το πρωτόκολλο να αποφύγει μία συντηρητική στρατηγική ανάκαμψης. Στην αντίθετη περίπτωση, όπου καλλιεργείται συμφόρηση στο δίκτυο, ο παραλήπτης ενημερώνει τον αποστολέα να μειώσει το ρυθμό αποστολής δεδομένων.

Κεφάλαιο 2

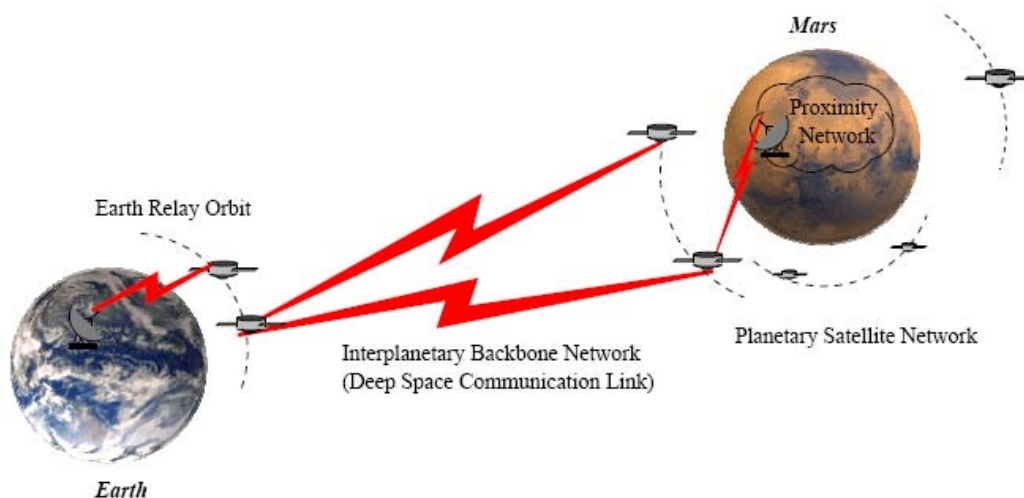
Διαπλανητικό Διαδίκτυο (InterPlanetary Internet)

2.1. Εισαγωγή

Οι εξελίξεις στις τεχνολογίες του διαστήματος την τελευταία δεκαετία έχουν καταστήσει ικανή την υλοποίηση επιστημονικών αποστολών στο Απομακρυσμένο Διάστημα (deep space), όπως είναι η εξερεύνηση του πλανήτη Άρη. Οι αποστολές αυτές παράγουν σημαντική ποσότητα επιστημονικών δεδομένων, τα οποία αποστέλλονται στη Γη. Για την επιτυχή μεταφορά των επιστημονικών δεδομένων και την αξιόπιστη επικοινωνία όσον αφορά θέματα πλοήγησης, οι διαστημικές αποστολές έχουν παρουσιάσει σημαντικές προκλήσεις σχετικά με την ανάπτυξη των αρχιτεκτονικών των διαστημικών δικτύων νέας γενιάς. Η νέα γενιά δικτύων που θα χρησιμοποιηθούν στο Απομακρυσμένο Διάστημα αναμένεται να παρέχει υπηρεσίες επικοινωνίας για παράδοση επιστημονικών δεδομένων και υπηρεσίες πλοήγησης για τα εξερευνητικά διαστημόπλοια και τα διάφορα αντικείμενα που βρίσκονται σε τροχιά γύρω από πλανήτες. Το επόμενο βήμα στο σχεδιασμό και την ανάπτυξη των δικτύων για το Απομακρυσμένο Διάστημα αναμένεται να είναι το Διαδίκτυο των πλανητικών δικτύων στο Απομακρυσμένο Διάστημα και ορίζεται ως Διαπλανητικό Διαδίκτυο (IPN – InterPlanetary Internet).

2.2. Αρχιτεκτονική δομή του Διαπλανητικού Διαδικτύου (InterPlanetary Internet)

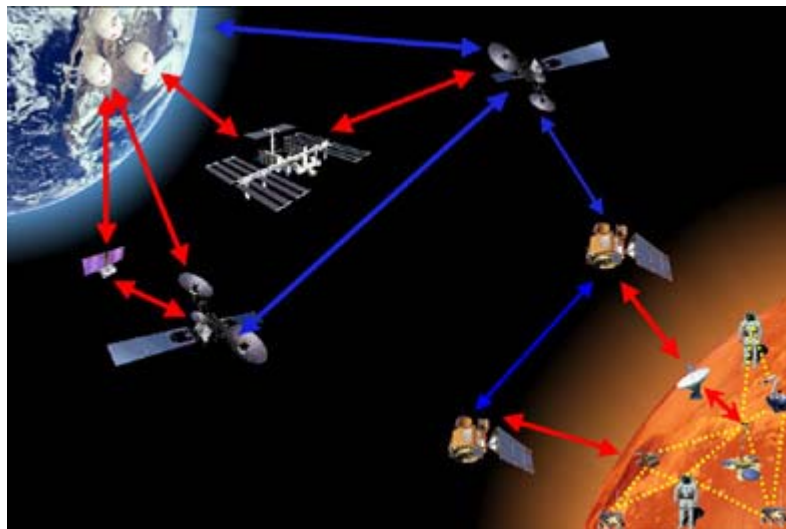
Μια τυπική αρχιτεκτονική δικτύου στο Απομακρυσμένο Διάστημα [18], [19] φαίνεται στην παρακάτω εικόνα και προτείνεται για αποστολές εξερεύνησης του πλανήτη Άρη.



Σχήμα 2.1 Αρχιτεκτονική δομή του Διαπλανητικού Διαδικτύου

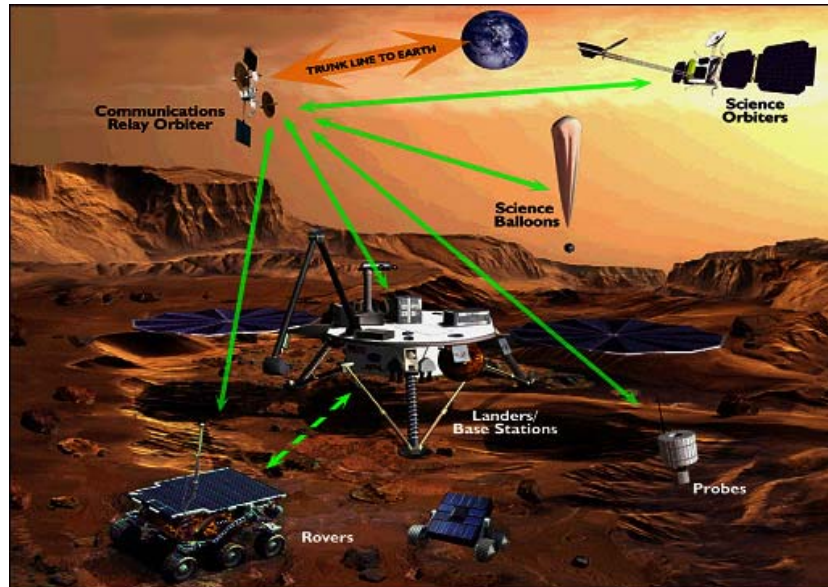
Τα αρχιτεκτονικά στοιχεία της προτεινόμενης υποδομής συνοψίζονται ως εξής:

- **Interplanetary Backbone Network:**
Περιλαμβάνει την απευθείας σύνδεση και τα μονοπάτια μέσω πολλαπλών κόμβων ανάμεσα στους πλανήτες του εξωτερικού διαστήματος και τη Γη, καθώς και τα στοιχεία της υποδομής που βρίσκονται στη Γη, όπως είναι οι επίγειοι σταθμοί για τα δίκτυα στο Απομακρυσμένο Διάστημα. Μπορεί επίσης να περιλαμβάνει σταθμούς αναμετάδοσης τοποθετημένους σε βαρυτικά σταθερά Lagrangian σημεία του εξωτερικού διαστήματος, όπως στο Δια ή τον Πλούτωνα.



Σχήμα 2.2. Η αρχιτεκτονική του Interplanetary Backbone Network.

- **Interplanetary Satellite Network:**
Αποτελείται από δορυφόρους που κινούνται σε τροχιά γύρω από τους πλανήτες για να παρέχουν υπηρεσίες επικοινωνίας και πλοήγησης στα στοιχεία της επιφάνειας.
- **Planetary Proximity Network:**
Παρέχει τις συνδέσεις επικοινωνίας ανάμεσα σε ποικίλου εύρους στοιχεία της επιφάνειας, τα οποία συχνά «απλώνονται» για να δημιουργήσουν ένα δίκτυο ad-hoc.

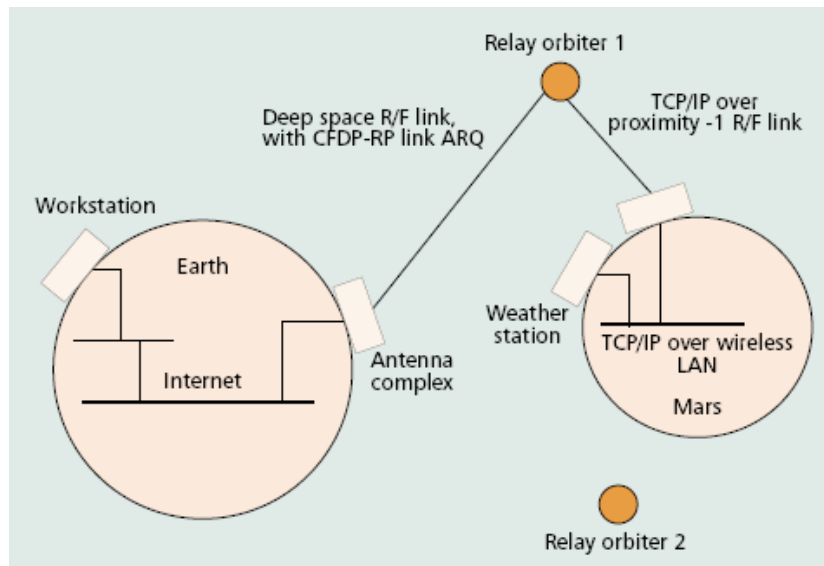


Σχήμα 2.3 Η αρχιτεκτονική του Planetary Proximity Network

Από τα παραπάνω αρχιτεκτονικά στοιχεία του Διαπλανητικού Διαδικτύου δίνουμε κυρίως βάση στο Interplanetary Backbone Network, όπου τα τελικά σημεία του αποστολέα και του δέκτη είναι ο επίγειος σταθμός και οι πλανητικές πύλες που συνδέονται στους δορυφόρους -σταθμούς αναμετάδοσης, οι οποίοι περιστρέφονται σε τροχιά γύρω από τους πλανήτες του εξωτερικού διαστήματος, όπως απεικονίζεται στην προηγούμενη εικόνα. Αυτό συμβαίνει επειδή το Interplanetary Backbone Network παίζει σημαντικό ρόλο στη λειτουργία ολόκληρου του συστήματος επικοινωνίας στο Απομακρυσμένο Διάστημα.

2.3. Διαφορές δικτυακών χαρακτηριστικών ανάμεσα στο Διαδίκτυο και το Διαπλανητικό Διαδίκτυο

Θεωρούμε έναν επιστήμονα που είναι υπεύθυνος για το χειρισμό ενός ρομποτικού μετεωρολογικού σταθμού που βρίσκεται στον πλανήτη Άρη [20].



Σχήμα 2.4. Ένα σενάριο για υψηλή τιμή καθυστέρησης

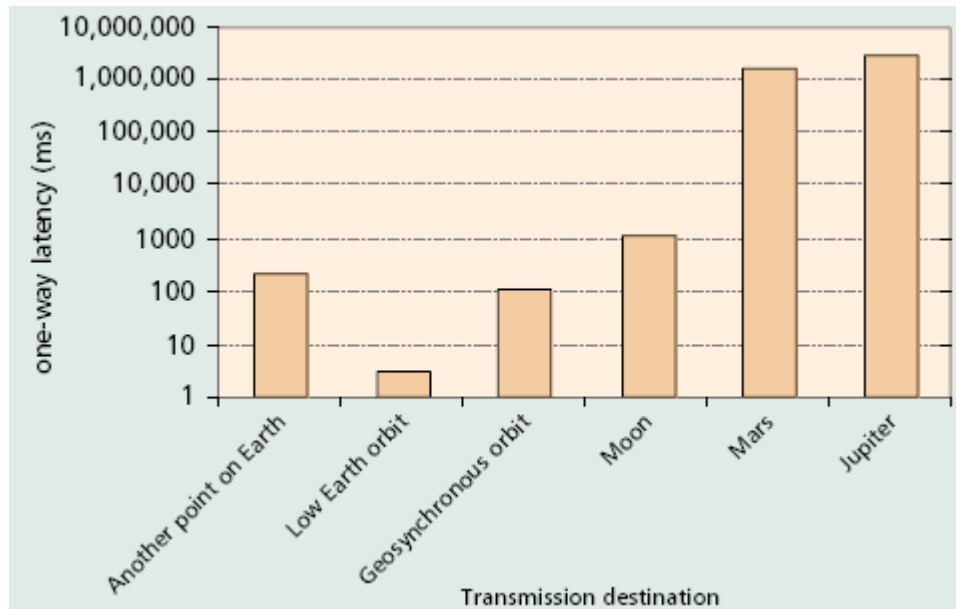
Ο μετεωρολογικός σταθμός είναι ένα από τις πολλές δεκάδες πλατφόρμες – εργαλεία που επικοινωνούν μεταξύ τους μέσω ενός ασύρματου τοπικού δικτύου το οποίο αναπτύσσεται στην επιφάνεια του Άρη. Ο επιστήμονας θέλει να αναβαθμίσει το λογισμικό στον υπολογιστή χειρισμού των δεδομένων του μετεωρολογικού σταθμού, εγκαθιστώντας και δυναμικά φορτώνοντας ένα νέο μεγάλο μοντέλο. Το μοντέλο πρέπει πρώτα να μεταδοθεί από το σταθμό εργασίας του επιστήμονα σε ένα σύμπλεγμα κεραιών που χρησιμοποιούνται για το Απομακρυσμένο Διάστημα, μετά από το σύμπλεγμα των κεραιών σε έναν αστερισμό από δορυφόρους αναμετάδοσης στη χαμηλή τροχιά του Άρη (κανένας από τους οποίους δεν είναι αρκετά ορατός από τη Γη, ώστε ένας μόνο δορυφόρος να μπορεί να παραλάβει ολόκληρο το μοντέλο) και τελικά από τους δορυφόρους αναμετάδοσης στο μετεωρολογικό σταθμό.

Το πρώτο κομμάτι του ταξιδιού αυτού τυπικά θα ολοκληρωνόταν χρησιμοποιώντας το σύνολο πρωτοκόλλων TCP/IP πάνω από το Διαδίκτυο, όπου η ηλεκτρονική επικοινωνία γενικά χαρακτηρίζεται από:

- Σχετικά μικρές τιμές καθυστέρησης (της τάξης των millisecond).
- Σχετικά υψηλούς ρυθμούς δεδομένων (μέχρι 40Gb/s).
- Αμφίδρομη επικοινωνία σε κάθε σύνδεση.
- Συνεχή συνδεσιμότητα από άκρο σε άκρο.
- Πρόσβαση στο δίκτυο μετά από αίτηση με υψηλή πιθανότητα συμφόρησης.

Παρόλα αυτά, για το δεύτερο κομμάτι της επικοινωνίας είναι απαραίτητη μια διαφορετική στοίβα πρωτοκόλλων. Η ηλεκτρονική επικοινωνία ανάμεσα σε ένα σταθμό παρακολούθησης και στο ρομποτικό διαστημόπλοιο στο Απομακρυσμένο Διάστημα γενικά χαρακτηρίζεται από:

- Πολύ μεγάλες τιμές καθυστέρησης διάδοσης. Οι συνδέσεις επικοινωνίας στο Απομακρυσμένο Διάστημα μπορεί να έχουν εξαιρετικά μεγάλες καθυστερήσεις διάδοσης. Για παράδειγμα, ο χρόνος μετ' επιστροφής από άκρο σε άκρο σε ένα δίκτυο επικοινωνίας μεταξύ της Γης και του Άρη ποικίλει από 8,5 λεπτά σε 40 λεπτά ανάλογα με την τροχιακή θέση των πλανητών. Στον παρακάτω πίνακα δίνονται κάποιες ενδεικτικές τιμές καθυστερήσεων.



Σχήμα 2.5. Μέγιστη τιμή καθυστέρησης (για μετάδοση από τη Γη)

- Σχετικά χαμηλούς ρυθμούς δεδομένων (τυπικά 8 – 256 kb/s).
- Υψηλούς ρυθμούς σφαλμάτων στις συνδέσεις. Οι ρυθμοί σφαλμάτων στις συνδέσεις στο Απομακρυσμένο Διάστημα είναι πάρα πολύ υψηλοί, συνήθως της τάξης του 10^{-1} .
- Πιθανές χρονικές περιόδους αποσύνδεσης της λήψης ή της μετάδοσης, εξαιτίας των τροχιών που κινούνται οι δορυφόροι ή/και της πολιτικής χειρισμού του διαστημοπλοίου.
- Διακοπτόμενη προγραμματισμένη συνδεσιμότητα.
- Κεντρικά διαχειριζόμενη πρόσβαση στο κανάλι επικοινωνίας με ουσιαστικά καμία πιθανότητα συμφόρησης.
- Ασυμμετρία εύρους ζώνης:
Η ασυμμετρία της χωρητικότητας του εύρους ζώνης των ευθέων και αντίστροφων καναλιών είναι τυπικά της τάξης του 1000:1 στις διαστημικές αποστολές.

Οι παραπάνω προκλήσεις πρέπει να εξεταστούν ώστε να επιτευχθούν οι επικοινωνιακές απαιτήσεις των αποστολών στο Απομακρυσμένο Διάστημα. Ο συνδυασμός των μεγάλων χρόνων διάδοσης των σημάτων και της ενδιάμεσης συνδεσιμότητας – που προκαλείται, για παράδειγμα, από την ενδιάμεση θέση ενός πλανητικού σώματος ανάμεσα στον αποστολέα και το δέκτη – μπορεί να οδηγήσει σε καθυστερήσεις επικοινωνίας μετ’ επιστροφής (RTT-Round Trip Time) που μετρώνται όχι σε millisecond, ούτε καν σε λεπτά, αλλά σε ώρες ή μέρες. Τα διαδικτυακά πρωτόκολλα δεν συμπεριφέρονται αποδοτικά κάτω από αυτές τις συνθήκες.

Ο παρακάτω πίνακας [21] συνοψίζει τις διαφορές που παρουσιάζουν οι γήινες και οι διαστημικές επικοινωνίες, όσον αφορά τα δικτυακά χαρακτηριστικά.

	ΓΗΙΝΕΣ ΕΠΙΚΟΙΝΩΝΙΕΣ	ΔΙΑΣΤΗΜΙΚΕΣ ΕΠΙΚΟΙΝΩΝΙΕΣ
Ρυθμός Σφαλμάτων Bit (BER)	Τυπικά $< 10^{-9}$	10^{-4} έως 10^{-12}
Καθυστέρηση μετ’ επιστροφής (RTT)	Από millisecond έως δευτερόλεπτα	Δευτερόλεπτα έως ώρες
Συνέχεια συνδεσιμότητας	Συνεχόμενη	Διακοπτόμενη
Ρυθμοί δεδομένων Πρόσω Καναλιού και Καναλιού Επιστροφής	Συμμετρικοί	10:1 έως 1000:1 λόγος ρυθμού δεδομένων Πρόσω Καναλιού προς Κανάλι Επιστροφής
Απαιτήσεις επεξεργαστή και μνήμης	Σχετικά μεγάλες	Σχετικά μικρές
Στόχοι Επικοινωνίας	Δίκαιη πρόσβαση με το χρόνο Υψηλή συνολική απόδοση Υψηλή αξιοπιστία	Μέγιστη απόδοση κατά τη διάρκεια της σύνδεσης Μέγιστη αξιοποίηση της σύνδεσης Επιλεγόμενο επίπεδο αξιοπιστίας
Πρωταρχική πηγή απώλειας	Συμφόρηση	Συμφόρηση Φθορά Διακοπή Σύνδεσης

2.4. Απαιτήσεις Σχεδιασμού πρωτοκόλλου για το Διαπλανητικό Διαδίκτυο

Ο σχεδιασμός πρωτοκόλλου που θα εφαρμοστεί στο Διαπλανητικό Διαδίκτυο δεν θα πρέπει να βασίζεται σε κάποιο από τα παρακάτω χαρακτηριστικά από άκρο σε άκρο:

- Συνεχής συνδεσιμότητα.
- Μικρή ή σταθερή καθυστέρηση μεταφοράς.
- Χαμηλός ρυθμός σφαλμάτων.
- Χαμηλή συμφόρηση.
- Υψηλοί ρυθμοί μεταφοράς.
- Συμμετρικοί ρυθμοί δεδομένων.
- Κοινό συντακτικό ή σημασιολογία, όσον αφορά εκφράσεις ονομάτων και διευθύνσεις.

- Άφιξη των δεδομένων με τη σειρά αποστολής τους.

Παρόλα αυτά, για βέλτιστη απόδοση από άκρο σε άκρο, θέλουμε να μπορούμε να εκμεταλλευόμαστε οποιαδήποτε από τις παραπάνω θεμιτές καταστάσεις όταν συμβαίνουν.

2.5. Εφαρμογές που χρησιμοποιούνται στις διαστημικές επικοινωνίες

Όπως είναι γνωστό από τις επίγειες επικοινωνίες, κάθε είδος εφαρμογής έχει διαφορετικές απαιτήσεις και προδιαγραφές. Όσον αφορά το Διαπλανητικό Διαδίκτυο, οι σημαντικότερες κατηγορίες εφαρμογών είναι η μεταφορά μεγάλου όγκου δεδομένων (Bulk Data Transfer), η Telecommand και η Telemetry.

Η μεταφορά μεγάλου όγκου δεδομένων [22] αναφέρεται στη μεταφορά μεγάλων μπλοκ δεδομένων από μια αποθηκευτική πηγή σε μια άλλη, χωρίς να μας ενδιαφέρει η σειρά παράδοσης. Μια τέτοια εφαρμογή έχει μεγάλες απαιτήσεις σε αποθηκευτικό χώρο, ειδικά σε διαστημικό περιβάλλον όπου οι τιμές της καθυστέρησης είναι πολύ μεγάλες. Οι καταχωρητές του αποστολέα πρέπει να έχουν ικανό μέγεθος ώστε να αποθηκεύουν τα δεδομένα μέχρι να επιστρέψουν οι επιβεβαιώσεις και να βεβαιωθεί ο αποστολέας πως τα δεδομένα έχουν αποσταλεί επιτυχώς. Επίσης, η αποστολή μεγάλου όγκου δεδομένων σε συνδέσεις με μεγάλες καθυστερήσεις απαιτούν μεγάλο χρονικό διάστημα για να ολοκληρωθούν. Για το λόγο αυτό κρίνεται σκόπιμο η χωρητικότητα της σύνδεσης πάνω από την οποία «τρέχει» η εφαρμογή, να είναι όσο το δυνατόν μεγαλύτερο.

Ο όρος Telecommand [23] αναφέρεται στη μεταφορά δεδομένων από τη Γη σε κάποιο διαστημόπλοιο. Οι πληροφορίες αυτές είναι συνήθως οδηγίες πλοήγησης και γενικότερα εντολές που αφορούν το χειρισμό του διαστημοπλοίου. Το γεγονός αυτό δημιουργεί την ανάγκη για παροχή 100% αξιοπιστίας. Δεν θα πρέπει σε καμία περίπτωση να έχουμε απώλεια δεδομένων, καθώς η απώλεια μιας εντολής μπορεί να οδηγήσει σε ανεπανόρθωτα προβλήματα. Για την επίτευξη των προδιαγραφών του Telecommand θα πρέπει η σειρά παράδοσης των δεδομένων να είναι η σωστή, θα πρέπει να υπάρχουν μηχανισμοί εντοπισμού και διόρθωσης σφαλμάτων και, φυσικά, μηχανισμός επαναμετάδοσης σε περίπτωση απώλειας ή φθοράς.

Αντίστοιχα με το Telecommand, ο όρος Telemetry [24] αναφέρεται στη μεταφορά δεδομένων από το διαστημόπλοιο στη Γη. Τα δεδομένα προς αποστολή, όμως, δεν είναι εντολές, αλλά συνήθως μετρήσεις και διάφορα άλλα επιστημονικά δεδομένα τα οποία συγκεντρώθηκαν. Στην περίπτωση αυτή, μπορεί κανείς να πει πως δεν απαιτείται 100% αξιοπιστία. Αν, για παράδειγμα, τα δεδομένα μας είναι μετρήσεις θερμοκρασίας που συμβαίνουν σε κάποιο πλανήτη και επαναλαμβάνονται κάθε 1 δευτερόλεπτο, σκέφτεται κανείς πως η απώλεια κάποιων τιμών δεν αποτελεί πρόβλημα. Όμως, αν τα δεδομένα προς μετάδοση ήταν κάποια μέτρηση η οποία δεν μπορεί να επαναληφθεί, τότε η απώλεια τους είναι ζωτικής σημασίας. Έτσι, στη περίπτωση του Telemetry, οι παράμετροι και οι απαιτήσεις της σύνδεσης εξαρτώνται από τη συγκεκριμένη εφαρμογή.

Μια άλλη κατηγορία εφαρμογών η οποία είναι ιδιαίτερα διαδεδομένη στο Διαδίκτυο είναι οι εφαρμογές πραγματικού χρόνου (Real Time). Οι εφαρμογές αυτές έχουν αυστηρές απαιτήσεις σχετικά με τη καθυστέρηση από άκρο σε άκρο, ενώ σημαντικό χαρακτηριστικό τους είναι η έλλειψη επαναμετάδοσης. Ο ρυθμός μετάδοσης τους ρυθμίζεται προσαρμόζοντας την ποιότητα της μεταδιδόμενης ροής, ανάλογα με τη διαθέσιμη χωρητικότητα. Το κόστος των εφαρμογών αυτών είναι συνήθως υψηλό, αφού έχουν μεγάλες απαιτήσεις σε διαθέσιμους πόρους. Το Διαπλανητικό Διαδίκτυο χαρακτηρίζεται, όπως είναι γνωστό, από υψηλό ρυθμό σφαλμάτων (που οδηγεί σε απώλειες πακέτων) και μεγάλες τιμές καθυστέρησης. Οι συνθήκες αυτές καθιστούν τη μεταφορά δεδομένων πραγματικού χρόνου πολύ δύσκολη. Αν η καθυστέρηση είναι πολύ μεγάλη (π.χ. 40 λεπτά για μετάδοση από τον πλανήτη Άρη στη Γη), η μετάδοση πραγματικού χρόνου δεν έχει νόημα, αφού στην ουσία δεν γίνεται σε πραγματικό χρόνο. Ένα πρωτόκολλο μεταφοράς που έχει προταθεί για εφαρμογές πραγματικού χρόνου είναι το RCP Planet, το οποίο περιγράφεται στο κεφάλαιο 5.

Αξίζει, τέλος, να σημειωθεί πως στο Διαπλανητικό Διαδίκτυο οι διαθέσιμοι πόροι είναι περιορισμένοι. Έτσι, σε κάθε είδος εφαρμογής θα πρέπει να πετυχαίνουμε βέλτιστη αξιοποίηση των πόρων αυτών.

2.6. Προβλήματα που παρουσιάζει το TCP στο Διάστημα

Μια τυποποιημένη συλλογή από πρότυπα για πρωτόκολλα επικοινωνίας δεδομένων στο Διάστημα θα ήταν χρήσιμη τόσο για τις στρατιωτικές, όσο για τις πολιτικές διαστημικές επικοινωνίες. Η χρήση κοινών πρωτοκόλλων για επικοινωνίες δεδομένων θα βελτιώσουν τη διαχείριση και θα μειώσουν το κόστος των διαστημικών συστημάτων, τα οποία συνήθως σχεδιάζονται και κατασκευάζονται ειδικά για κάθε διαστημική αποστολή ή για ένα σύνολο αποστολών. Στην ανάπτυξη μιας πρότυπης συλλογής πρωτοκόλλων για χρήση στις διαστημικές επικοινωνίες είναι φυσικό να χρησιμοποιούμε ως βάση τα επιτυχημένα και εδώ και πολύ καιρό χρησιμοποιούμενα πρωτόκολλα του Διαδικτύου, το TCP και το IP. Παρόλα αυτά, υπάρχουν σημαντικές διαφορές ανάμεσα στο διαστημικό περιβάλλον και αυτό των τυπικών γήινων επικοινωνιών. Αν και το TCP/IP έχει ήδη εφαρμοστεί με επιθυμητά αποτελέσματα σε μια πληθώρα διαφορετικών περιβαλλόντων, από δορυφορικές συνδέσεις χαμηλού ρυθμού μέχρι τις γραμμές οπτικών ινών, η επιτυχία στο πεδίο των διαστημικών αποστολών αναμένεται να απαιτεί συγκεκριμένες προεκτάσεις για να αντιμετωπίσει τους περιορισμούς που θέτονται από το εχθρικό διαστημικό περιβάλλον.

Οι υπάρχουσες παραλλαγές του TCP (Transmission Control Protocol) δείχνουν να πετυχαίνουν πολύ κακή απόδοση στα δίκτυα επικοινωνιών στο Απομακρυσμένο Διάστημα [18], [20], [25], [26]. Ο κυρίαρχος παράγοντας στη μείωση της απόδοσης είναι οι εξαιρετικά υψηλές τιμές της καθυστέρησης διάδοσης στις συνδέσεις στο Απομακρυσμένο Διάστημα. Αυτό οφείλεται κυρίως στο μηχανισμό που βασίζεται στα παράθυρα και χρησιμοποιείται από τις σημερινές εκδόσεις του πρωτοκόλλου TCP κατά τη διάρκεια του Αργού Ξεκινήματος (Slow Start) και των αλγορίθμων αποφυγής συμφόρησης (Congestion Avoidance Algorithms), καθώς επίσης και στο μηχανισμό Ack-clocking. Οι παράμετροι που οδηγούν στη κακή απόδοση του TCP στο Διάστημα επεξηγούνται αναλυτικά παρακάτω.

2.6.1. Αλγόριθμος Αργού Ξεκινήματος (Slow Start)

Στον αλγόριθμο του αργού ξεκινήματος, το μέγεθος του παραθύρου συμφόρησης (W) αυξάνεται κατά ένα πακέτο με κάθε λαμβανόμενη επιβεβαίωση (ACK) μέχρι να φτάσουμε το κατώφλι του αργού ξεκινήματος (W_{ss}), δηλαδή όσο $W < W_{ss}$. Όμως, η προσέγγιση αυτή αχρηστεύει τα μέσα της σύνδεσης μας για πολύ μεγάλη χρονική διάρκεια, η οποία είναι ανάλογη της καθυστέρησης διάδοσης. Για $W_{ss}=20$ και χρόνο μετ' επιστροφής $RTT=20$ λεπτά, ο αλγόριθμος αργού ξεκινήματος δεν μπορεί να αξιοποιήσει τα μέσα της σύνδεσης για περίπου 120 λεπτά.

2.6.2. Μηχανισμοί που βασίζονται στα παράθυρα

Η ανεπάρκεια αξιοποίησης της σύνδεσης εξαιτίας των μηχανισμών που βασίζονται στα παράθυρα υπάρχει ακόμη και κατά τη διάρκεια της φάσης αποφυγής συμφόρησης, δηλαδή όταν $W \geq W_{ss}$, όταν η πηγή TCP αυξάνει το μέγεθος του παραθύρου συμφόρησης κατά 1, κατά προσέγγιση, σε κάθε RTT . Έχει υπολογιστεί πως τα πρωτόκολλα TCP που βασίζονται σε παράθυρα πετυχαίνουν απόδοση περίπου 10b/s για χωρητικότητα σύνδεσης 1Mb/s, με πιθανότητα απώλειας πακέτων της τάξης του $p=10^{-3}$ και $RTT=40$ λεπτά. Με άλλα λόγια, ολόκληρη η σύνδεση στο Απομακρυσμένο Διάστημα παραμένει σχεδόν αχρησιμοποίητη καθ' όλη τη διάρκεια της σύνδεσης. Σημειώστε ότι ο χρόνος $RTT=40$ λεπτά ανήκει στο εύρος τιμών RTT για συνδέσεις επικοινωνίας ανάμεσα στον Άρη και τη Γη, δηλαδή από 8,5 έως 40 λεπτά, ανάλογα με την τροχιακή θέση.

2.6.3. Ρυθμοί σφαλμάτων

Τα σημερινά πρωτόκολλα TCP έχουν σχεδιαστεί για ενσύρματες συνδέσεις, οι οποίες όπως είναι λογικό, θεωρούμε πως έχουν αμελητέο ρυθμό σφαλμάτων (BER - Bit Error Rate). Στις ασύρματες επικοινωνίες γενικά, αλλά ιδιαίτερα στις διαστημικές επικοινωνίες, τα σφάλματα bits που δημιουργούνται από θόρυβο δεν είναι ασυνήθιστα. Σε κάποιο βαθμό, ο FEC (Forward Error Correction) [27], [28], [29] μπορεί να αποκαταστήσει τέτοια σφάλματα, ανταλλάσσοντας τη χωρητικότητα με αποτελεσματικό ρυθμό δεδομένων, όμως η σύνδεση στο Διάστημα, ακόμη, είναι σπάνια τόσο “καθαρή”, όσο αυτές των σύγχρονων επίγειων δικτύων. Το TCP είναι σχεδιασμένο για να χειρίζεται απώλειες πακέτων ταυτοποιώντας και αναμεταδίδοντας τα χαμένα τμήματα. Όμως, το TCP θεωρεί πως η πηγή της απώλειας όλων των πακέτων είναι η συμφόρηση του δικτύου. Κατά συνέπεια, επικαλείται τον έλεγχο συμφόρησης, μειώνει το παράθυρο συμφόρησης και με τη σειρά του το ρυθμό μετάδοσης δεδομένων ως αποτέλεσμα κάθε απώλειας πακέτων. Η απόκριση αυτή είναι ακατάλληλη όταν η απώλεια οφείλεται σε φθορά και όχι σε συμφόρηση, όπως συμβαίνει συχνά στο Διάστημα και σε άλλες συνδέσεις κινητής επικοινωνίας. Ο αλγόριθμος ελέγχου συμφόρησης του TCP λειτουργεί σωστά όταν η απώλεια προκαλείται από συμφόρηση, αλλά καταλήγει μόνο σε μείωση της απόδοσης σε θορυβώδεις συνδέσεις.

2.6.4. Ασυμμετρία καναλιών

Προβλήματα δημιουργεί ακόμη το γεγονός ότι τα κανάλια επικοινωνίας ανάμεσα στο διαστημόπλοιο και τη Γη είναι συχνά ασύμμετρα με την έννοια της χωρητικότητας των καναλιών και των χαρακτηριστικών των σφαλμάτων. Συχνά η χωρητικότητα της

πρόσω σύνδεσης (από το διαστημόπλοιο στη Γη) είναι σημαντικά μεγαλύτερη από τη χωρητικότητα της σύνδεσης επιστροφής, με λόγους της τάξης 1000:1 όχι ασυνήθιστες. Η ασυμμετρία αυτή είναι αποτέλεσμα διαφόρων μηχανικών ανταλλαγών (όπως είναι η ενέργεια, η ποσότητα και η ένταση), καθώς επίσης και του γεγονότος πως στις επιστημονικές αποστολές, τα περισσότερα δεδομένα προέρχονται από το δορυφόρο και μεταδίδονται στη Γη. Το κανάλι επιστροφής χρησιμοποιείται γενικά για το χειρισμό του διαστημοπλοίου, και όχι τη μεταφορά μεγάλης ποσότητας δεδομένων. Η ασυμμετρία εύρους ζώνης μπορεί να περιορίσει την απόδοση του TCP ακόμη και όταν τα δεδομένα μεταδίδονται στο μεγαλύτερο κανάλι και το μικρότερο μεταφέρει μόνο επιβεβαιώσεις. Ένα τελικό σημείο δέκτη TCP γενικά δέχεται επιβεβαιώσεις για κάθε δεύτερο κομμάτι που λαμβάνει, γεγονός που επιβάλλει χωρητικότητα του καναλιού των επιβεβαιώσεων ανάλογη της χωρητικότητας του καναλιού δεδομένων. Με μέγεθος πακέτων 1024 bytes, ένας λόγος εύρους ζώνης εμπρός σύνδεσης προς σύνδεσης επιστροφής 50:1 δεν επηρεάζει την απόδοση του TCP· για μεγαλύτερους λόγους όμως η απόδοση περιορίζεται από τη χωρητικότητα του καναλιού των επιβεβαιώσεων. Ένας επιπρόσθετος λόγος για τη μείωση του απαιτούμενου εύρους ζώνης του καναλιού που χρησιμοποιείται για τις επιβεβαιώσεις είναι ότι ταυτόχρονα μειώνονται οι εκπομπές ενέργειας μιας κινητής μονάδας, όταν αυτή μόνο λαμβάνει δεδομένα.

2.6.5. Χωρητικότητα της σύνδεσης

Ένα άλλο χαρακτηριστικό που επηρεάζει την απόδοση του TCP σε διαστημικό περιβάλλον είναι η περιορισμένη χωρητικότητα της σύνδεσης. Τα ασύρματα κανάλια επικοινωνίας συνηθίζουν να προσφέρουν μικρότερη διαθέσιμη χωρητικότητα από ότι τα ενσύρματα δίκτυα. Στο χώρο των κινητών και διαστημικών επικοινωνιών, το πρόβλημα συνδέεται με τον περιορισμό ότι η ενέργεια είναι περιορισμένη και η επάρκεια bit είναι σημαντική όσον αφορά το κόστος της μετάδοσης, καθώς επίσης και τη χωρητικότητα της σύνδεσης. Υπάρχει μια σημαντική επιβάρυνση που σχετίζεται με το πρωτόκολλο TCP, ειδικά όταν χρησιμοποιούμε μικρά πακέτα για να αυξήσουμε την πιθανότητα μιας επιτυχημένης μετάδοσης ενός πακέτου, χωρίς να υποστεί κάποιο σφάλμα bit. Η επιβάρυνση αυτή, τουλάχιστον 20 bytes από την επικεφαλίδα του TCP σε κάθε πακέτο, μπορεί να καταναλώσει ένα σημαντικό μερίδιο ενός καναλιού περιορισμένου εύρους.

2.6.6. Διακοπές συνδεσιμότητας

Πρέπει ακόμη να λάβουμε υπόψη μας ότι για δορυφόρους σε μη γεωστατικές τροχιές η συνδεσιμότητα για μια δεδομένη σύνδεση επικοινωνίας είναι συνήθως διακοπτόμενη. Η επαφή μπορεί να διακοπεί για ένα πλήθος από λόγους, όπως είναι η “παράδοση” σε επίγειους σταθμούς, οι αλλαγές στην τοπολογία του δικτύου, η επισκίαση της κεραίας, τα καιρικά φαινόμενα και η τροχιακή δυναμική. Όταν ένας δορυφόρος μεταφέρεται από την ορατότητα ενός επίγειου σταθμού σε έναν άλλον, η συμπεριφορά είναι παρόμοια με αυτήν μιας επίγειας ανταλλαγής ανάμεσα σε δύο βάσεις-σταθμούς, αν και ο χρόνος μεταφοράς μπορεί να είναι πολύ μεγαλύτερος.

Επιπρόσθετα, σύνθετες, δυναμικές τοπολογίες (π.χ. επικοινωνίες ανάμεσα σε δύο μεγάλους δορυφορικούς αστερισμούς) είναι πιθανό να εμφανίζονται σε μελλοντικά σενάρια διαστημικών επικοινωνιών, σε αντίθεση με το απλουστευμένο μοντέλο Διάστημα-Γη που επικρατεί σήμερα. Σε ένα δυναμικά εναλλασσόμενο δίκτυο, όταν

ένα ενδιάμεσο σύστημα, είτε ένας σταθμός-βάση, είτε ένας δορυφόρος μεταφέρει έναν κινητό χρήστη σε ένα άλλο ενδιάμεσο σύστημα, η συνδεσιμότητα μπορεί να διακοπεί προσωρινά.

Ακόμη και μικρής διάρκειας διακοπές λειτουργίας της σύνδεσης θέτουν ένα πρόβλημα για το TCP, καταλήγοντας σε χαμηλή απόδοση στην καλύτερη περίπτωση και σε τερματισμό της σύνδεσης στη χειρότερη. Με την έλλειψη μιας σταθερής ροής επιβεβαιώσεων, το TCP θα επικαλείται τον έλεγχο συμφόρησης και επαναλαμβανόμενα θα μεταδίδει και θα αποτυγχάνει. Αν η συνδεσιμότητα αποκατασταθεί πριν το TCP ξεπεράσει το μέγιστο χρονικό όριο αναμετάδοσης, θα συνεχίσει τη μετάδοση από εκεί που σταμάτησε. Όμως, μέχρι να επανέλθει η σύνδεση ο χρονομετρητής αναμετάδοσης μπορεί να έχει “υποχωρήσει” τόσες πολλές φορές που θα χρειαστούν λεπτά πριν το TCP το αναγνωρίσει. Αν το TCP ξεπεράσει το μέγιστο όριο αναμετάδοσης πριν επανέλθει η συνδεσιμότητα, τότε η σύνδεση τερματίζεται.

2.6.7. Διακύμανση των μετρούμενων χρόνων RTT

Ένα άλλο πιθανό αποτέλεσμα μιας εναλλασσόμενης τοπολογίας δικτύου είναι μια ευρεία διακύμανση των μετρούμενων χρόνων μετ’ επιστροφής RTT, εξαιτίας αυξήσεων ή μειώσεων των αποστάσεων διάδοσης και του πλήθους των ενδιάμεσων σταθμών που περιλαμβάνει η διαδρομή. Το TCP μπορεί να αναμεταδώσει είτε πολύ νωρίς, είτε πολύ αργά, αν η εκτίμηση του RTT είναι επαρκώς μεγαλύτερη από την τωρινή ακριβή τιμή. Πρόωρα χρονικά περιθώρια αναμετάδοσης καταλήγουν σε άχρηστες επικλήσεις του Αργού Ξεκινήματος (Slow Start), μειώνοντας την απόδοση.

2.6.8. Καθυστερημένη ανατροφοδότηση

Υπάρχουν πολλές ακόμη προκλήσεις που πρέπει να εξεταστούν από τα νέα πρωτόκολλα μεταφοράς στο Interplanetary Backbone Network, όπως είναι η καθυστερημένη ανατροφοδότηση. Το TCP αναμένεται να απαντά στην κατάσταση του δικτύου. Η προσδοκία αυτή δημιουργεί προβλήματα σε περιβάλλοντα υψηλής καθυστέρησης, από τη στιγμή που το TCP χρησιμοποιεί σηματοδότηση από άκρο σε άκρο για τον βρόχο ελέγχου του. Όσο υψηλότερο RTT έχουμε, τόσο παλαιότερες πληροφορίες λαμβάνονται στον αποστολέα σχετικά με τις συνθήκες της σύνδεσης. Έτσι, οι αποφάσεις του ελέγχου συμφόρησης που βασίζονται σε τέτοιες παλιές πληροφορίες είναι πολύ πιθανό να οδηγήσουν σε λανθασμένα μέτρα. Για αυτό, τα σχέδια ελέγχου συμφόρησης τα οποία αντιδρούν στις καταστάσεις άμεσης απώλειας πακέτων, δεν αποδίδουν κατάλληλη απόκριση στις συνδέσεις με υψηλές καθυστερήσεις διάδοσης.

2.6.9. Χωρητικότητα καταχωρητών

Ακόμη, για να εξασφαλίσουμε 100% αξιόπιστη μεταφορά, οι μηχανισμοί αναμετάδοσης είναι αναπόφευκτοι, γεγονός που προκαλεί σημαντικές απαιτήσεις μνήμης. Για παράδειγμα, η πηγή του πρωτόκολλου μεταφοράς θα πρέπει να έχει μέγεθος καταχωρητή 1,2GB για RTT=20 λεπτά και μέσο ρυθμό μετάδοσης δεδομένων 1MB/s.

2.6.10. Εγκαθίδρυση σύνδεσης

Σημαντικό πρόβλημα αποτελεί το γεγονός ότι η επικοινωνία με TCP απαιτεί ο αποστολέας και ο παραλήπτης να διαπραγματευτούν μια σύνδεση, η οποία θα οριοθετήσει τη ροή των δεδομένων. Η επίτευξη μιας σύνδεσης TCP τυπικά συνεπάγεται τουλάχιστον μια μετάδοση μετ' επιστροφής (μετάδοση και απόκριση) πριν μπορέσουν να μεταδοθούν οποιαδήποτε δεδομένα των εφαρμογών. Αν η λανθάνουσα κατάσταση μετάδοσης ξεπεράσει τη διάρκεια της δυνατής επικοινωνίας, δεν θα μεταδοθούν καθόλου δεδομένα.

2.6.11. Σειρά παράδοσης δεδομένων

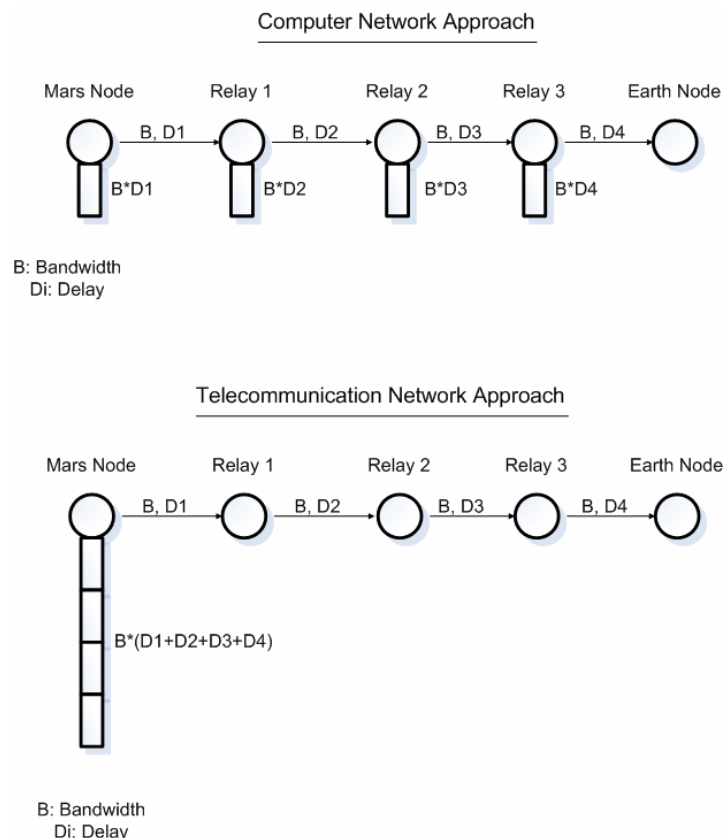
Πρόβλημα αποτελεί, ακόμη, το γεγονός ότι το TCP παραδίδει τα δεδομένα που λαμβάνει στις εφαρμογές του μόνο με τη σειρά μετάδοσης. Αυτό σημαίνει ότι μια απώλεια δεδομένων που απαιτεί επαναμεταφορά θα καθυστερήσει την παράδοση όλων των δεδομένων, που μεταδόθηκαν αμέσως μετά στην ίδια σύνδεση, μέχρι τα χαμένα δεδομένα να μεταδοθούν επιτυχώς – δηλαδή για χρόνο ίσο με τουλάχιστον ένα RTT. Για την αποφυγή αυτού του αποκλεισμού, η μόνη επιλογή της εφαρμογής αποστολής των δεδομένων είναι να «υποβληθεί» στο κόστος να ανοίξει επιπλέον παράλληλες συνδέσεις και να διανέμει τη μετάδοση μέσω των συνδέσεων αυτών.

2.6.12. Επικοινωνία από άκρο σε άκρο

Η χρήση του TCP από άκρη σε άκρη πάνω από ένα μονοπάτι που περιλαμβάνει πολλαπλές συνδέσεις, κάποιες από τις οποίες μπορεί να είναι μακρινές ή ενδιάμεσες, παρουσιάζει επιπρόσθετες δυσκολίες. Η επαναμεταφορά του TCP είναι από άκρο σε άκρο και η επαναμεταφορά από άκρο σε άκρο καθυστερεί την απελευθέρωση του χώρου του καταχωρητή επαναμεταφοράς.

Για παράδειγμα, υποθέστε μια διαδρομή τριών κόμβων, ΑΒΓΔ, όπου η καθυστέρηση του σήματος μεταφοράς από το ένα άκρο στο άλλο είναι 500ms στον κόμβο ΑΒ, 8 λεπτά (480000ms) στον κόμβο ΒΓ και 100ms στον κόμβο ΓΔ. Φανταστείτε ότι Α είναι ο σταθμός μετάδοσης στον Άρη, Β είναι ο δορυφόρος που κινείται σε τροχιά γύρω από τον Άρη, Γ είναι ένας σταθμός παρακολούθησης στη Γη και Δ είναι ο σταθμός εργασίας ενός επιστήμονα κάπου στο Διαδίκτυο. Η επαναμεταφορά είναι πιθανή μόνο όταν ο αρχικός αποστολέας ενός μηνύματος διατηρεί ένα αντίγραφο μέχρι να είναι σίγουρος ότι η επαναμεταφορά του δεν είναι αναγκαία (π.χ. ο παραλήπτης ειδοποιεί ότι το μήνυμα έχει παραδοθεί). Αν η μετάδοση είναι από κόμβο σε κόμβο – δηλαδή πραγματοποιείται ανάμεσα στα τελικά σημεία της κάθε σύνδεσης μεμονωμένα – ο Α μπορεί να ελευθερώσει το χώρο από τον καταχωρητή επαναμεταφοράς μετά από περίπου $1000\text{ms} + 500\text{ms}$ για τη μεταφορά από τον Α στον Β και 500ms για την επιβεβαίωση από τον Β στον Α. Αν η μεταφορά είναι από άκρο σε άκρο, ο Α πρέπει να διατηρεί το χώρο στον καταχωρητή του για 961200ms (ΑΒ, ΒΓ, ΓΔ, ΔΓ, ΓΒ, ΒΑ). Η απώλεια δεδομένων θα καθυστερούσε ακόμη περισσότερο την ελευθέρωση του χώρου του καταχωρητή επαναμεταφοράς κατά τουλάχιστον ένα RTT, καταναλισκόμενο από την αίτηση επαναμεταφοράς και την επαναμεταφορά του μηνύματος · η απώλεια είτε της αίτησης, είτε της απόκρισης, φυσικά, θα χειροτέρευε την κατάσταση.

Αυτό με τη σειρά του σημαίνει πως ο χώρος που απαιτείται για τον καταχωρητή επαναφοράς στα τελικά σημεία της διαδρομής αυξάνεται με τη μεταφορά από άκρο σε άκρο. Για βέλτιστη χρήση της σύνδεσης, οι συνδέσεις θα πρέπει ανά πάσα στιγμή να μεταφέρουν όσο περισσότερα δεδομένα μπορούν. Ας υποθέσουμε πως ο μέγιστος ρυθμός δεδομένων σε κάθε σύνδεση είναι 1Mb/s. Στη χειρότερη περίπτωση, όπου ο A είναι η πηγή όλων των δεδομένων, οι συνδέσεις υπο-χρησιμοποιούνται αν ο A δεν στέλνει δεδομένα με ρυθμό 1Mb/s. Ας υποθέσουμε ότι ο A αποστέλλει μια εικόνα μεγέθους 256kb από μια κάμερα χαμηλής ανάλυσης κάθε 250ms. Στο ρυθμό αυτό, υποθέτοντας πως δεν έχουμε καθόλου απώλεια δεδομένων, οι καταχωρητές του A για συνολική επαναμεταφορά θα καταναλώνουν μέχρι 1Mb μνήμης αν η επαναμεταφορά γίνεται από κόμβο σε κόμβο: 1000ms μετά την έναρξη της λειτουργίας, οι επιβεβαιώσεις από τον B θα προκαλούν την απελευθέρωση χώρου με τον ίδιο ρυθμό που τον καταναλώνουν κατά την αρχική μεταφορά από τον A (256kb κάθε 0,25s καθώς οι εικόνες αποστέλλονται, παραλαμβάνονται και επιστρέφουν οι επιβεβαιώσεις). Αλλά αν η επαναμεταφορά γίνεται από άκρο σε άκρο, οι επιβεβαιώσεις από τον Δ δεν θα αρχίσουν να απελευθερώνουν τους καταχωρητές επαναμεταφοράς του A μέχρι 961200ms μετά την έναρξη της αποστολής · οι καταχωρητές επαναμεταφοράς του A θα καταναλώνουν πάνω από 961MB μνήμης. Η σχηματική απεικόνιση ενός παραδείγματος όμοιου με αυτό που περιγράφηκε προηγουμένως δίνεται στην παρακάτω εικόνα. Στο παράδειγμα αυτό το ένα άκρο βρίσκεται στη Γη, το άλλο στον πλανήτη Άρη, ενώ έχουμε τρεις ενδιάμεσους δορυφόρους αναμετάδοσης.



Σχήμα 2.6. Οι απαιτήσεις σε καταχωρητές ενός δικτύου σύμφωνα με δύο προσεγγίσεις. Πρώτα προσέγγιση ενός δικτύου υπολογιστών και στη συνέχεια προσέγγιση ενός δικτύου επικοινωνιών.

2.7. Τρόποι αντιμετώπισης των προβλημάτων που παρουσιάζει το TCP στο Διάστημα

Υπάρχουν διάφορες προτάσεις-προσεγγίσεις [30], [31], [32] για την αντιμετώπιση των προβλημάτων που αντιμετωπίζει το TCP σε διαστημικό περιβάλλον και τα οποία αναφέρθηκαν στη προηγούμενη ενότητα.

Αρχικά, πρέπει να εκμεταλλευτούμε το χρόνο που απαιτείται για τη δημιουργία μιας νέας σύνδεσης, κατά τον οποίο η σύνδεση μένει αδρανής. Μια πιθανή εναλλακτική λύση στη «Τριπλή χειραψία» είναι η διαπραγμάτευση της δημιουργίας σύνδεσης ταυτόχρονα με την αποστολή δεδομένων. Δηλαδή, αποστέλλονται τα κατάλληλα πακέτα που απαιτούνται για τη δημιουργία μιας σύνδεσης, ακολουθούμενα από πακέτα δεδομένων, πριν ακόμη επιβεβαιωθεί η σύνδεση. Αν η σύνδεση επιβεβαιωθεί, τότε η αποστολή δεδομένων συνεχίζεται κανονικά. Αν η σύνδεση δεν επιβεβαιωθεί ή δεν μπορεί να εντοπιστεί ο παραλήπτης, τότε η σύνδεση τερματίζεται.

Ένας ακόμη μηχανισμός ο οποίος κρίνεται ακατάλληλος για συνδέσεις στο Διάστημα είναι ο αλγόριθμος Slow Start. Το Αργό Ξεκίνημα είναι πάρα πολύ αργό για συνδέσεις με μεγάλες τιμές καθυστέρησης, με αποτέλεσμα να μην εκμεταλλευόμαστε τη διαθέσιμη χωρητικότητα για μεγάλο χρονικό διάστημα. Μια λύση στο πρόβλημα αυτό είναι η χρήση διαφορετικών αλγορίθμων οι οποίοι παρουσιάζουν ταχύτερη αύξηση του παραθύρου συμφόρησης, έτσι ώστε να φτάνουμε στον ακριβή ρυθμό δεδομένων γρήγορα.

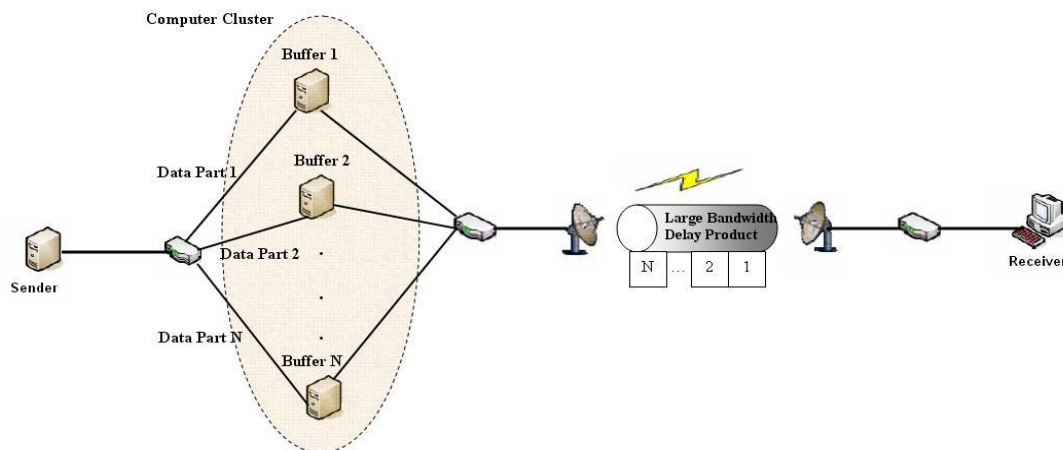
Επίσης οι μηχανισμοί Fast Recovery και Fast Retransmit κρίνονται ανεπαρκείς για το Διαπλανητικό Διαδίκτυο, αφού οι μεγάλες τιμές καθυστέρησης οδηγούν σε καθυστερημένη (και συνεπώς λανθασμένη) ανατροφοδότηση.

Πρωταρχική πηγή απώλειας πακέτων στα επίγεια δίκτυα είναι η συμφόρηση. Έτσι, το TCP επικαλείται τον έλεγχο συμφόρησης μόλις παρατηρήσει απώλειες πακέτων. Αντίθετα, στις διαστημικές συνδέσεις πρωταρχική πηγή απώλειας πακέτων αποτελούν τα φυσικά στοιχεία του δικτύου και όχι το πλήθος των χρηστών. Έτσι, στη περίπτωση αυτή, η χρήση του ελέγχου συμφόρησης οδηγεί μονάχα σε άσκοπη μείωση της απόδοσης. Ιδανική λύση στο πρόβλημα αυτό θα αποτελούσε η δυνατότητα εντοπισμού και διαχωρισμού των απωλειών πακέτων σε απώλειες που οφείλονται σε συμφόρηση και απώλειες που οφείλονται σε σφάλματα. Πρακτικά, ένας τρόπος ανίχνευσης συμφόρησης είναι η αποστολή πλαστών πακέτων χαμηλής προτεραιότητας, τα οποία απορρίπτονται πρώτα σε περίπτωση συμφόρησης. Αυτή τη μέθοδο χρησιμοποιεί και το πρωτόκολλο μετάδοσης TP Planet, το οποίο περιγράφεται αναλυτικά στο πέμπτο κεφάλαιο.

Ο υψηλός ρυθμός σφαλμάτων αποτελεί ένα από τα σημαντικότερα προβλήματα σε συνδέσεις στο Απομακρυσμένο Διάστημα. Ο FEC (Forward Error Correction) μπορεί να ελαχιστοποιήσει την ανάγκη για επαναμετάδοση, αν και δεν μπορεί μόνος του να παρέχει 100% αξιόπιστη παράδοση δεδομένων. Ο FEC εισάγει, όμως, επιπλέον επιβάρυνση bits για να αντιμετωπίσει τη φθορά πακέτων. Τα πακέτα που εμφανίζουν σφάλματα διορθώνονται απευθείας, χωρίς επαναμετάδοση, γεγονός που είναι πολύ σημαντικό ιδιαίτερα για περιβάλλοντα με μεγάλες τιμές καθυστέρησης. Όμως, ο FEC καταλήγει σε άσκοπη δέσμευση εύρους ζώνης σε περίπτωση έλλειψης σφαλμάτων, ενώ έχει επιπρόσθετες απαιτήσεις σε χρόνο επεξεργασίας, μνήμη και

κατανάλωση ενέργειας. Για το λόγο αυτό είναι προτιμότερος ο συνδυασμός του FEC με την Αυτόματη Καθυστερημένη Επαναμετάδοση (Delayed Automatic Retransmission), ένα νέο μηχανισμό μετάδοσης ανοιχτού βρόχου. Χρήσιμη είναι ακόμη η άμεση ενημέρωση του αποστολέα από τον παραλήπτη, σε περίπτωση που παρατηρήσει σφάλματα στα δεδομένα.

Πρόβλημα αποτελεί ακόμη η ασυμμετρία των δύο καναλιών, αφού το κανάλι των επιβεβαιώσεων είναι κατά πολύ μικρότερο από το κανάλι που χρησιμοποιείται για τα δεδομένα. Το γεγονός αυτό οδήγησε στη χρήση διαφορετικών μηχανισμών επιβεβαιώσεων από αυτές που χρησιμοποιεί το πρότυπο TCP. Εναλλακτικές προτάσεις αποτελούν το TCP SACK (Selective Acknowledgments-Επιλεκτικές Επιβεβαιώσεις) και η χρήση SNACK (Selective Negative Acknowledgments-Επιλεκτικές Αρνητικές Επιβεβαιώσεις). Η χρήση των μεθόδων αυτών εμφανίζει τον περιορισμό της χρήσης μεγάλου μεγέθους καταχωρητή, ο οποίος είναι ικανός να αποθηκεύει τα πακέτα που στάλθηκαν μέχρι να επιστρέψει στον αποστολέα η επιβεβαίωση παραλαβής τους. Μια λύση στο πρόβλημα αυτό είναι η χρήση πολλών ομάδων από καταχωρητές, καθένας από τους οποίους αποθηκεύει ένα μέρος των δεδομένων τα οποία μεταδόθηκαν. Φυσικά σε ένα τέτοιο σύστημα απαιτείται ένας κατάλληλος αλγόριθμος χρονοπρογραμματισμού, ώστε τα πακέτα δεδομένων να φτάνουν στην πύλη με τη σωστή σειρά. Η λύση αυτή απεικονίζεται στο παρακάτω σχήμα.



Σχήμα 2.7. Χρήση ομάδων από καταχωρητές για την αντιμετώπιση του προβλήματος του μεγάλου μεγέθους των καταχωρητών.

Η διακοπτόμενη συνδεσιμότητα λόγω αστεροειδών, διαστημοπλοίων ή πλανητών που παρεμβάλλονται στη γραμμή επικοινωνίας είναι μια τυπική κατάσταση σε συνδέσεις στο Απομακρυσμένο Διάστημα. Η κατάσταση αυτή οδηγεί σε ραγδαία μείωση της απόδοσης και μπορεί να οδηγήσει ακόμη και στο τερματισμό της σύνδεσης. Μια λύση αποτελεί η συχνή χρήση πλαστών πακέτων, τα οποία αρχίζουν να αποστέλλονται μόλις παρατηρηθεί διακοπή της σύνδεσης και ως σκοπό έχουν την άμεση ενημέρωση του αποστολέα μόλις αποκατασταθεί η σύνδεση.

Τέλος, αξίζει να σημειωθεί πως χρήσιμο θα ήταν να μειωθεί το μέγεθος της επικεφαλίδας όσο το δυνατόν περισσότερο, ώστε να έχουμε μικρότερη επιβάρυνση. Αυτό μπορεί να επιτευχθεί με την αφαίρεση πεδίων τα οποία δεν είναι απαραίτητα (όπως είναι κάποιες επιπρόσθετες επιλογές) ή με τη συμπίεση της επικεφαλίδας, αν αυτό είναι εφικτό.

Κεφάλαιο 3

Εργαστηριακή Μεθοδολογία

Σκοπός αυτού του κεφαλαίου είναι η περιγραφή του τρόπου διεξαγωγής των εργαστηριακών πειραμάτων, που σκοπό έχουν στο σύνολο τους την αξιολόγηση των διαδικτυακών πρωτοκόλλων μεταφοράς δεδομένων σε διαστημικό περιβάλλον. Η αξιολόγηση αυτή αφορά τομείς απόδοσης που σχετίζονται άμεσα μεταξύ τους, όπως η ποσότητα δεδομένων που αποστέλλονται κατά τη διάρκεια μιας επικοινωνίας (χρήσιμη και μη χρήσιμη πληροφορία), οι χρονικές καθυστερήσεις που επιβάλλονται από τη φύση του δικτύου, καθώς επίσης η ενεργειακή συμπεριφορά των πρωτοκόλλων. Για τη μαθηματική αποτύπωση όλων των παραπάνω παραμέτρων απόδοσης επιβάλλεται η χρήση μονάδων μέτρησης απόδοσης, που θα μας δίνουν την ακριβή απόδοση στον κάθε τομέα. Για να καθορίσουμε ένα πείραμα-σενάριο με όλες τις απαραίτητες λεπτομέρειες που θα το χαρακτηρίσουν πλήρως, πρέπει να παραμετροποιήσουμε την τοπολογία του δικτύου και να καθορίσουμε όλες τις λεπτομέρειες που θα οδηγήσουν στα κατάλληλα συμπεράσματα από τη διεξαγωγή των πειραμάτων.

3.1 Τοπολογία των πειραμάτων και γενικές παράμετροι

Τα πειράματα εκτελούνται στον προσομοιωτή δικτύων Network Simulator 2 [33], [34]. Έχουν ως βάση την τοπολογία Simple [35], που απεικονίζεται στο παρακάτω σχήμα.



Σχήμα 3.1 Simple Network Topology

Η τοπολογία αυτή είναι πολύ απλή και αποτελείται από μια κύρια σύνδεση (backbone link) η οποία έχει χωρητικότητα `bw_backbone` και καθυστέρηση `delay_backbone`. Η σύνδεση χρησιμοποιείται από ένα μόνο αποστολέα (source) για την αποστολή δεδομένων προς ένα δέκτη (sink).

Κατά την εκτέλεση των πειραμάτων πρέπει να προσδιοριστούν τα παρακάτω μεγέθη:

Application Type → Τύπος της εφαρμογής που «τρέχουν» οι χρήστες

TCP Version → Έκδοση TCP

Queue Type → Τύπος ουράς

Metric Type → Τύπος μετρικής

Flows Entering Time → Χρόνος έναρξης προσομοίωσης

Flows Life Time → Χρόνος λήξης προσομοίωσης

Start Method → Μέθοδος εκκίνησης

Error Rate → Ρυθμός σφαλμάτων

Handoff → Διακοπή συνδεσιμότητας

Buffer Size → Μέγεθος καταχωρητή

Number of CBR/UDP Flows → Πλήθος ροών CBR/UDP

Maximum cwnd → Μέγιστο παράθυρο συμφόρησης

TCP Timeout → Χρονικό διάστημα επαναποστολής TCP

Mimimum RTO → Ελάχιστο χρονικό διάστημα επαναποστολής

Maximum RTO → Μέγιστο χρονικό διάστημα επαναποστολής

SNACK RTO → Χρονικό διάστημα επαναποστολής SNACK

Οι τιμές των παραπάνω παραμέτρων προσδιορίζονται ανάλογα με το εκάστοτε πείραμα που εκτελείται.

3.2 Μονάδες Μέτρησης της Απόδοσης

3.2.1. Bandwidth Efficiency

Το κύριο χαρακτηριστικό που παρακολουθείται σε ένα δίκτυο ή σε μια απλή ζεύξη κατά τη διάρκεια της επικοινωνίας είναι η κυκλοφορία των πακέτων στο δίκτυο και ελέγχεται κατά πόσο το δίκτυο πλησιάζει στη μέγιστη απόδοση του. Αυτό το χαρακτηριστικό είναι το εύρος ζώνης (bandwidth) και ορίζεται ως η θεωρητική «ταχύτητα» ροής δεδομένων ή διαφορετικά η «χωρητικότητα» μιας τομής του καναλιού και μετράται σε bits per second (bps). Το bandwidth είναι ένα προκαθορισμένο στοιχείο μιας σύνδεσης (link) και βάσει αυτού μετράται η απόδοση του πρωτοκόλλου. Η απόδοση του δικτύου μετριέται τις περισσότερες φορές με τις ακόλουθες τρεις μονάδες μέτρησης: την απόδοση (throughput), την πραγματική απόδοση (goodput) και την επιβάρυνση (overhead), οι οποίες ορίζονται ως εξής:

Απόδοση (Throughput) (Bps): Η μετρήσιμη ταχύτητα ροής, η μέγιστη ροή, όταν πάντα υπάρχουν δεδομένα προς αποστολή κατά το διάστημα της μέτρησης. Περιλαμβάνει και την επιβάρυνση (overhead) η οποία μετράται σε bytes.

$$\text{Throughput} = \text{Data Sent in Bytes} / \text{Time}$$

$$\text{Application Throughput} = \text{Data Received in Bytes} / \text{Time}$$

Πραγματική Απόδοση (Goodput) (Bps): Το Throughput χωρίς τις επικεφαλίδες και την επιβάρυνση λόγω επαναμεταφοράς δεδομένων. Τα «καθαρά» δεδομένα που παραλαμβάνει η εφαρμογή.

$$\text{Goodput} = \text{Throughput} - (\text{Overhead} / \text{Time})$$

ή αλλιώς

$$\text{Goodput} = \text{Original Data} / \text{Connection Time}$$

όπου Original Data είναι ο αριθμός των bytes που φτάνουν στην εφαρμογή (χωρίς τα πακέτα που ξαναστάλθηκαν και τις επικεφαλίδες) και Connection Time είναι ο χρόνος που χρειάστηκε για τη μεταφορά των δεδομένων. Έτσι ορίζουμε το System Goodput ως το άθροισμα των Goodput όλων των flows:

$$\text{System Goodput} = \sum_{1 \leq i \leq n} g_i$$

όπου g_i είναι το goodput για το flow i και n ο συνολικός αριθμός των flows.

Επιβάρυνση (Overhead) (Bytes): Οι επιπλέον διαχειριστικές πληροφορίες καθώς και τα δεδομένα που ξαναστέλνονται από τους αποστολείς.

$$\text{Overhead} = \text{Headers} + \text{Retransmission Overhead}$$

ή αλλιώς

$$\text{Overhead} = (\text{Bytes Sent} - \text{Original Bytes}) / \text{Bytes Sent}$$

όπου Bytes Sent είναι τα συνολικά bytes που στάλθηκαν από τους TCP αποστολείς και Original Bytes είναι ο αριθμός των bytes που έφτασαν στην εφαρμογή, αφαιρώντας τα πακέτα που ξαναστάλθηκαν και τις TCP επικεφαλίδες. Αυτή η μονάδα μέτρησης αναφέρεται στο ποσοστό του εύρους ζώνης που “καταναλώθηκε”, ή αλλιώς στο ποσοστό της ενέργειας που καταναλώθηκε για τη μεταφορά δεδομένων (μέχρι την εφαρμογή), κάτι που παίζει πολύ σημαντικό ρόλο στις ασύρματες επικοινωνίες.

Κρίσιμο ερώτημα αποτελεί το ποια μονάδα μέτρησης μπορεί να αντικατοπτρίσει καλύτερα την απόδοση του πρωτοκόλλου. Όπως γίνεται φανερό από τους ορισμούς που δόθηκαν παραπάνω, το Bandwidth αποτελεί μόνο τη θεωρητική ταχύτητα ροής δεδομένων μέσω του δικτύου, κάτι που σημαίνει ότι το πρωτόκολλο δεν είναι σίγουρο ότι θα καταλάβει πλήρως ή ότι δεν θα προσπαθήσει να υπερκαλύψει (χάνοντας πακέτα), λαμβάνοντας υπόψη τη δυναμική συμπεριφορά του πρωτοκόλλου κατά τη διάρκεια της επικοινωνίας.

Γενικά το Throughput είναι μία αρκετά παρεξηγημένη έννοια. Μπορεί κανείς να το μετρήσει για μία συγκεκριμένη εφαρμογή (application throughput) ή μπορεί κανείς

να το μετρήσει σε διαφορετικά σημεία του Διαδικτύου. Υποθέστε για παράδειγμα ότι ένα δίκτυο συνδέεται με κάποιο άλλο μέσω ενός δρομολογητή. Σε κάποια φάση της λειτουργίας του ο δρομολογητής γεμίζει τον buffer του και αναγκάζεται να «ρίξει» μερικά πακέτα. Αν μετρήσουμε την κυκλοφορία σε ένα σημείο του Διαδικτύου που βρίσκεται μετά το δρομολογητή, θα διαπιστώσουμε μικρότερη κυκλοφορία από ότι αν τη μετρήσουμε στον αποστολέα. Επομένως, μπορούμε να πούμε ότι το Throughput καθρεπτίζει την προσπάθεια αποστολής δεδομένων μιας εφαρμογής, χωρίς να καθορίζει αυτό αν τα δεδομένα αυτά πραγματικά μεταφέρονται από το δίκτυο και τελικά καταλήγουν στην εφαρμογή του δέκτη.

Πρέπει ακόμη να λάβουμε υπόψη μας το γεγονός ότι πολλά πακέτα χάνονται και ξαναστέλνονται από τους αποστολείς. Αυτό σημαίνει ότι ένα μέρος μόνο από τα πακέτα που κυκλοφορούν στο Διαδίκτυο είναι «χρήσιμα» πακέτα για τη δικτυακή εφαρμογή. Τελικά, όμως, αυτό που μας ενδιαφέρει ως χρήστες είναι το πόσο καλά «τρέχει» η εφαρμογή μας, δηλαδή τι ροή χρήσιμων δεδομένων υπάρχει. Όπως γίνεται φανερό, λοιπόν, το throughput δεν μπορεί να αποτελέσει τη μοναδική και απόλυτη μονάδα μέτρησης απόδοσης, και αυτός είναι ο λόγος που εισήχθη ως μονάδα το Goodput, το οποίο μετράει ακριβώς τη ροή δεδομένων τα οποία είναι χρήσιμα, δηλαδή φτάνουν στην εφαρμογή προς επεξεργασία.

3.2.2. Delay x Bandwidth Product

Ένας άλλος εξίσου σημαντικός παράγοντας που επηρεάζει την απόδοση του δικτύου, και κατ' επέκταση του πρωτοκόλλου, είναι η καθυστέρηση (Delay). Η καθυστέρηση απαρτίζεται από επιμέρους μονάδες:

Καθυστέρηση διάδοσης (Propagation Delay) (ms) είναι η καθυστέρηση του σήματος μέχρι να φτάσει στον παραλήπτη και έχει να κάνει με την απόσταση και την ταχύτητα του φωτός πάνω στο συγκεκριμένο μέσο μεταφοράς.

Καθυστέρηση μετάδοσης / μεταβίβασης (Transmission Delay) (ms) είναι η καθυστέρηση που οφείλεται στην ταχύτητα μετάδοσης δεδομένων πάνω από το κανάλι και έχει να κάνει με την ποσότητα των δεδομένων και την ταχύτητα μετάδοσης του καναλιού.

$$\text{Transmission Delay} = \text{Transmitted Data} / \text{Bandwidth}$$

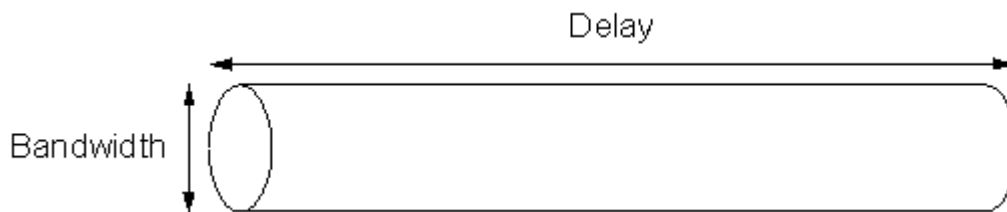
Καθυστέρηση αναμονής (Queuing Delay) (ms) είναι η καθυστέρηση που οφείλεται σε αναμονή στους καταχωρητές (buffers) των δρομολογητών λόγω της πολύπλεξης.

Καθυστέρηση επεξεργασίας (Processing Delay) (ms) είναι η καθυστέρηση που οφείλεται στην επικόλληση των επικεφαλίδων, στην υλοποίηση του δικτύου και στην ενδοεπικοινωνία διαφορετικών πρωτοκόλλων μέχρι να διοχετεύσουν τα δεδομένα στο δίκτυο.

Τελικά η **συνολική καθυστέρηση** είναι το άθροισμα όλων των προηγούμενων:

$$\text{Delay} = \text{Propagation Delay} + \text{Transmission Delay} + \text{Queuing Delay} + \text{Processing Delay}$$

Στο σημείο αυτό είναι εύλογη η απορία σχετικά με το ποια μονάδα εκφράζει καλύτερα τη δυναμικότητα ενός δικτύου (ποια έχει μεγαλύτερη σημαντικότητα), η ταχύτητα (Bandwidth) ή η καθυστέρηση (Delay); Για να απαντήσουμε στο ερώτημα αυτό πρέπει να επισημάνουμε πως η καθυστέρηση διάδοσης δεν σχετίζεται με την ποσότητα δεδομένων και η καθυστέρηση μεταβίβασης δεν σχετίζεται με την απόσταση. Επομένως, η σημαντικότητα της καθυστέρησης μεταβίβασης αυξάνεται όσο αυξάνεται η ποσότητα δεδομένων ή όσο μειώνεται η ταχύτητα μετάδοσης. Για το λόγο αυτό στην πράξη χρησιμοποιείται ένας συνδυασμός του εύρους ζώνης και της καθυστέρησης, το λεγόμενο **Delay X Bandwidth Product**. Αυτό εκφράζει τη χωρητικότητα του καναλιού, δηλαδή το πόσα πακέτα μπορούμε να στείλουμε συνεχόμενα μέχρι να φτάσουν τα δεδομένα στον παραλήπτη, ή μέχρι να έχουμε επιβεβαίωση στον αποστολέα. Συνήθως, αντί για Delay χρησιμοποιείται μία άλλη μονάδα, το RTT (Round Trip Time).



Σχήμα 3.2 Bandwidth X Delay Product

3.2.3. Fairness Index

Βασική επιδίωξη των διαδικτυακών πρωτοκόλλων είναι η εξασφάλιση της δίκαιης κατανομής των διαδικτυακών πόρων. Φυσικά η δίκαιη κατανομή έχει νόημα μόνο όταν έχουμε στο δίκτυο πολλές εφαρμογές και λιγότερους διαθέσιμους πόρους, παρέχοντας ίσες ευκαιρίες στους χρήστες. Παρόλα αυτά, σε ένα πραγματικό περιβάλλον δεν είναι δυνατόν να κατανεμηθούν ισάξια οι πόροι του συστήματος σε όλους τους χρήστες. Για το λόγο αυτό καθορίστηκε ο Δείκτης Δικαιοσύνης (Fairness Index) έτσι ώστε να ποσοτικοποιήσουν τη δικαιοσύνη σε ένα σύστημα. Ο γενικός τύπος που χρησιμοποιείται για τη μέτρηση της δικαιοσύνης είναι ο Throughput Fairness Index και ο εξής:

$$\text{Fairness Index} = (\sum g_i)^2 / [n(\sum g_i^2)]$$

όπου g_i είναι το Throughput για το flow i και n ο συνολικός αριθμός των flows.

Ο δείκτης δικαιοσύνης είναι πάντοτε ένας αριθμός μεταξύ του 0 και του 1, όπου το 1 παριστάνει τη μέγιστη δικαιοσύνη, δηλαδή όλα τα flows καταναλώνουν ακριβώς το fair-share που τους αντιστοιχεί.

Σε όλες τις μετρήσεις της παρούσας εργασίας ο Fairness Index είναι πάντοτε μονάδα, αφού έχουμε μόνο ένα flow.

3.3. Δείκτης Κατανάλωσης Ενέργειας

Η ενέργεια που καταναλώνει ένα πρωτόκολλο κατά την αποστολή των δεδομένων σε μια επικοινωνία αποτελεί έναν πολύ σημαντικό παράγοντα. Η σημαντικότητα της ενέργειας προέρχεται από το γεγονός ότι η επικοινωνία μπορεί να γίνεται μέσω τερματικών συσκευών, οι οποίες να χρησιμοποιούν ενέργεια από μπαταρία. Για το λόγο αυτό θα πρέπει να καταναλώνεται όσο το δυνατόν λιγότερη ενέργεια, αποφεύγοντας την άσκοπη κατανάλωση κατά τη διάρκεια της επικοινωνίας.

Retransmitted Packets

Μια μονάδα μέτρησης που μπορεί να χαρακτηρίσει την επιπλέον ενέργεια που καταναλώνει ένα πρωτόκολλο κατά την προσπάθεια μετάδοσης δεδομένων, είναι τα «πακέτα που ξαναστέλνονται» (Retransmitted Packets). Τα πακέτα αυτά προέρχονται από απώλειες προηγούμενων προσπαθειών μετάδοσης. Τα Retransmitted Packets προκαλούν επιπλέον κατανάλωση ενέργειας και έτσι, γνωρίζοντας το πλήθος των πακέτων που ξαναστέλνονται, έχουμε κατά αναλογία την επιπλέον ενέργεια που καταναλώνεται στην προσπάθεια μετάδοσης.

3.4. Επιλεκτικές Αρνητικές Επιβεβαιώσεις SNACK (Selective Negative Acknowledgments)

Επιλεκτικές αρνητικές επιβεβαιώσεις SNACK [36], [37], όπως υπονοεί και το όνομα τους, ονομάζεται η διαδικασία επιλεκτικής αποστολής αρνητικών επιβεβαιώσεων. Ο δέκτης των δεδομένων ενημερώνει τον αποστολέα σχετικά με τα πακέτα που δεν παρέλαβε. Οι συνηθισμένες επιβεβαιώσεις μπορούν να μεταφέρουν πληροφορίες σχετικά με ένα μόνο κενό στο παράθυρο· αν το TCP εμφανίζει πολλαπλά κενά σε ένα παράθυρο, τότε θα πρέπει να στείλουμε πολλαπλές επιβεβαιώσεις για να ενημερώσουμε τον αποστολέα για όλα τα κενά στη πλευρά του δέκτη. Αυτό είναι ασύμφορο ιδιαίτερα σε ασύμμετρα κανάλια, όπου το κανάλι των επιβεβαιώσεων είναι περιορισμένο και μία σύνδεση μπορεί να έχει λόγο καναλιού δεδομένων προς καναλιού επιβεβαιώσεων, ο οποίος κυμαίνεται από 10:1 έως 1000:1.

Οι επιλεκτικές αρνητικές επιβεβαιώσεις (SNACK) είναι συνδυασμός των επιλεκτικών επιβεβαιώσεων (SACK – Selective Acknowledgement) και των αρνητικών επιβεβαιώσεων (NAK – Negative Acknowledgment). Οι επιβεβαιώσεις SNACKs εμφανίζονται όταν παρατηρείται λανθασμένη διάταξη των πακέτων στο δίκτυο και, τότε, είναι χρήσιμο να καθυστερήσουμε την αποστολή τους για ένα χρονικό διάστημα, ώστε να δώσουμε αρκετό χρόνο στα πακέτα που δεν ήταν στη σειρά να φτάσουν. Ο δέκτης δεν θα πρέπει να στέλνει SNACKs αν δεν είναι σίγουρος ότι έχουν χαθεί πακέτα στο δίκτυο. Άσκοπη χρήση των SNACKs μπορεί να οδηγήσει σε άχρηστη επαναποστολή, γεγονός που μπορεί να μειώσει την απόδοση του συστήματος, παρά να τη βελτιώσει.

3.5. Σενάρια προσομοίωσης συνδέσεων

Στα πειράματα που εκτελέστηκαν έγινε η χρήση των παρακάτω σεναρίων, καθένα από τα οποία εξετάζει κάποια διαφορετική παράμετρο της βασικής τοπολογίας.

Επίδραση της καθυστέρησης

Στο σενάρια αυτό εξετάζεται η επίδραση της τιμής της καθυστέρησης RTT στην τελική απόδοση των διαδικτυακών πρωτοκόλλων. Είναι ήδη γνωστό πως τα διαδικτυακά πρωτόκολλα είναι σχεδιασμένα να λειτουργούν αποδοτικά σε περιβάλλοντα με μικρούς χρόνους καθυστέρησης. Μέσω αυτού του σεναρίου αποδεικνύεται πειραματικά η σταδιακή πτώση της απόδοσης των πρωτοκόλλων για αυξανόμενους χρόνους καθυστέρησης.

Μέγεθος καταχωρητών

Ο ρόλος των καταχωρητών στο σχεδιασμό ενός αποδοτικού πρωτοκόλλου είναι ζωτικής σημασίας. Ο αποστολέας θα πρέπει να έχει αρκετό ελεύθερο χώρο στον καταχωρητή, ώστε διατηρεί αντίγραφα των δεδομένων που αποστέλλει μέχρι να βεβαιωθεί πως έχουν παραδοθεί επιτυχώς. Στις διαστημικές συνδέσεις, εξαιτίας των μεγάλων χρόνων καθυστέρησης, ο όγκος των δεδομένων αυτών είναι πολύ μεγάλος. Αυτό έρχεται σε σύγκρουση με τους περιορισμούς όγκου και ενέργειας που θέτει ένα τέτοιο περιβάλλον. Στο σενάριο αυτό θα γίνουν τα κατάλληλα πειράματα, ώστε να υπολογιστεί ένα ικανοποιητικό μέγεθος που θα πρέπει να έχουν οι καταχωρητές που χρησιμοποιούνται σε διαστημικές συνδέσεις.

Ρυθμός σφαλμάτων ανά πακέτο

Τα σημερινά πρωτόκολλα TCP έχουν σχεδιαστεί για ενσύρματες συνδέσεις, οι οποίες έχουν αμελητέο ρυθμό σφαλμάτων. Στις διαστημικές επικοινωνίες, τα σφάλματα bits που δημιουργούνται από θόρυβο είναι πολύ συνηθισμένα και αποτελούν ένα σημαντικό πρόβλημα. Στο σενάριο αυτό θα εξετάσουμε κατά πόσο επηρεάζουν την απόδοση των πρωτοκόλλων, σφάλματα ανά πακέτο (PER – Packet Error Rate) που κυμαίνονται σε ρυθμούς από 10^{-6} έως 10^{-1} . Θα εξετάσουμε, ακόμη, ξεχωριστά την περίπτωση τα σφάλματα να παρουσιάζονται μόνο στο forward link (δηλαδή να χάνονται πακέτα) και την περίπτωση τα σφάλματα να παρουσιάζονται στο return link (δηλαδή να χάνονται επιβεβαιώσεις).

Χωρητικότητα της κύριας σύνδεσης

Πρώτιστης σημασίας σε κάθε είδους δίκτυο είναι η χωρητικότητα της σύνδεσης. Στο σενάριο αυτό θα εξετάσουμε κατά πόσο επηρεάζεται η απόδοση των πρωτοκόλλων από τη χωρητικότητα της κύριας σύνδεσης. Για να γίνει αισθητή η σημασία της χωρητικότητας, το ίδιο πείραμα θα επαναληφθεί για διάφορες τιμές καθυστερήσεων.

Ασύμμετρα κανάλια

Τα κανάλια επικοινωνίας ανάμεσα σε ένα διαστημόπλοιο και τη Γη είναι συχνά ασύμμετρα με την έννοια της χωρητικότητας των καναλιών. Συχνά το εύρος ζώνης της πρόσω σύνδεσης (από το διαστημόπλοιο στη Γη) είναι σημαντικά μεγαλύτερο από το εύρος ζώνης της σύνδεσης επιστροφής. Ο λόγος των δύο καναλιών επηρεάζει την τελική απόδοση του πρωτοκόλλου. Το γεγονός αυτό θα αποδειχθεί πειραματικά μέσω ενός σεναρίου όπου ο λόγος των δύο καναλιών κυμαίνεται από 2:1 έως 1000:1.

Σφάλματα στο forward link και στο return link

Στο σενάριο αυτό θα εξετάσουμε την απόδοση των πρωτοκόλλων για δύο διαφορετικές περιπτώσεις: πρώτον όταν το forward link παρουσιάζει σφάλματα με αποτέλεσμα να χάνονται πακέτα δεδομένων και, δεύτερον, όταν το return link παρουσιάζει σφάλματα με συνέπεια την απώλεια επιβεβαιώσεων. Θα έχει μεγάλο ενδιαφέρον να παρατηρηθεί η αντίδραση των πρωτοκόλλων υπό αυτές τις συνθήκες.

Διακοπή συνδεσιμότητας

Στις διαστημικές συνδέσεις η συνδεσιμότητα για μια δεδομένη σύνδεση επικοινωνίας είναι συνήθως διακοπτόμενη. Η επαφή μπορεί να διακοπεί για ένα πλήθος από λόγους όπως είναι η παρεμβολή αστεροειδών, διαστημοπλοίων ή πλανητών στη γραμμή επικοινωνίας. Στο σενάριο αυτό θα εξετάσουμε την επίδραση διακοπών συνδεσιμότητας διαφορετικής διάρκειας στη τελική απόδοση του πρωτοκόλλου.

Bandwidth X Delay Product

Στο σενάριο αυτό προσομοιώνονται διαφορετικές περιπτώσεις, οι οποίες έχουν το ίδιο Delay X Bandwidth Product, αλλά διαφορετική καθυστέρηση και χωρητικότητα κύριας σύνδεσης. Μέσω του πειράματος αυτού θα μπορέσουμε να διακρίνουμε αν σημαντικότερο ρόλο στην απόδοση ενός πρωτοκόλλου «παίζει» η τιμή της καθυστέρησης ή η τιμή της χωρητικότητας.

Συνδυασμός προβλημάτων (PER, Asymmetry, Blackout)

Σε όλα τα σενάρια που παρουσιάστηκαν ως τώρα εξετάζονταν μεμονωμένα οι παράμετροι που οδηγούν σε μείωση της απόδοσης ενός πρωτοκόλλου TCP σε διαστημικό περιβάλλον. Στο σενάριο αυτό παρατηρούμε την απόδοση ενός πρωτοκόλλου για πραγματικές συνθήκες, έχοντας ταυτόχρονα σφάλματα, ασυμμετρία καναλιών και διακοπές συνδεσιμότητας.

Κεφάλαιο 4

Πειραματικά Αποτελέσματα

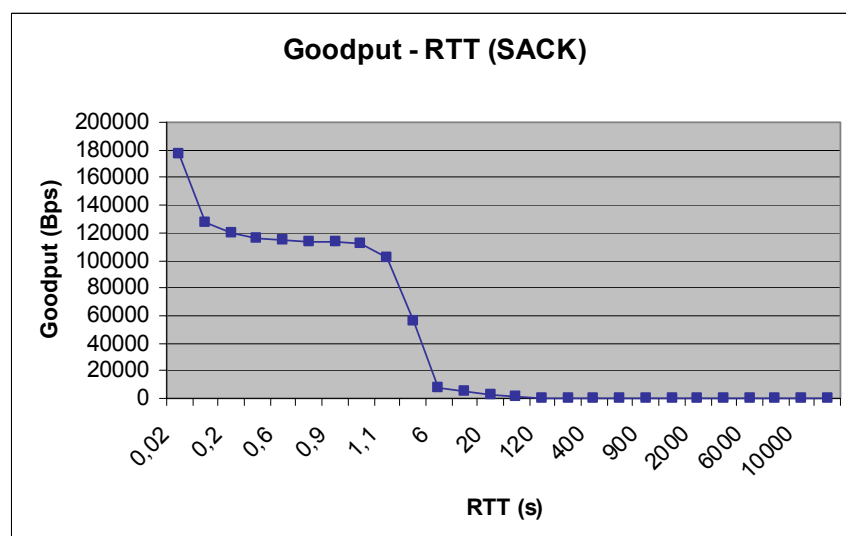
Στο κεφάλαιο αυτό παρουσιάζονται τα αποτελέσματα από τα πειράματα που περιγράφηκαν στο προηγούμενο κεφάλαιο. Για κάθε πείραμα γίνεται μία σύντομη περιγραφή των παραμέτρων που χρησιμοποιήθηκαν, ακολουθούμενη από γραφικές αναπαραστάσεις των αποτελεσμάτων για την ευκολότερη κατανόηση τους. Τέλος, γίνεται ένας σχολιασμός των αποτελεσμάτων και των παραμέτρων που οδήγησαν στο αποτέλεσμα. Στο παράρτημα που συμπεριλαμβάνεται στο τέλος, δίνονται συνοπτικοί πίνακες των παραμέτρων του κάθε πειράματος, καθώς επίσης και οι ακριβείς τιμές των αποτελεσμάτων.

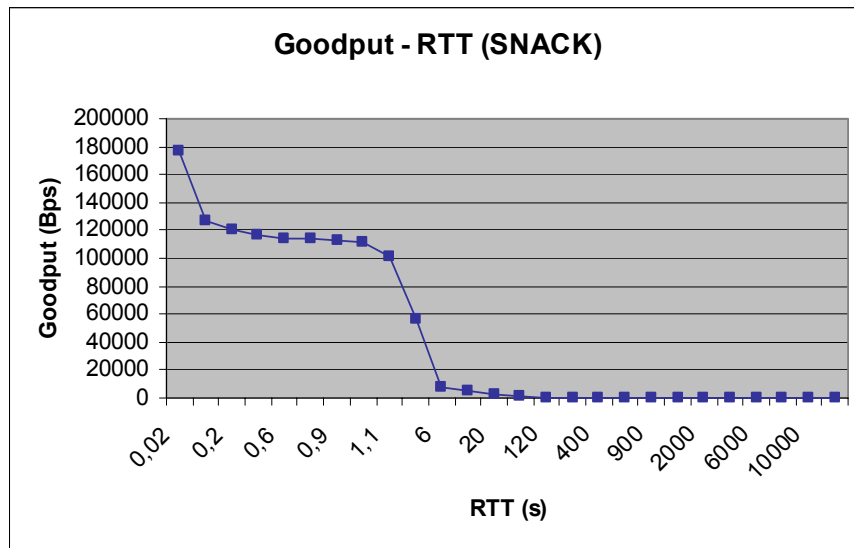
4.1. Αποτελέσματα

4.1.1. Αύξηση της καθυστέρησης

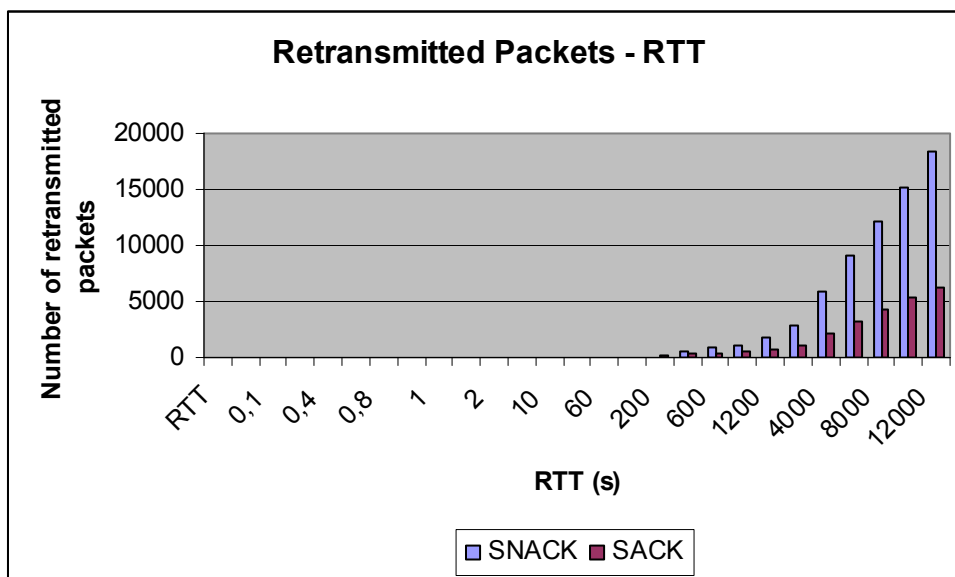
Στο σενάριο αυτό προσομοιώνεται μια τοπολογία τύπου simple, όπως αυτή που απεικονίζεται στο σχήμα 3.1, όπου η χωρητικότητα της κύριας σύνδεσης είναι 1Mbps. Ο τύπος της εφαρμογής που “τρέχουν” οι χρήστες του δικτύου είναι μεταφορά μεγάλου όγκου δεδομένων (Bulk Data Transfer) και η διαχείριση της ουράς δεδομένων των καταχωρητών του δικτύου γίνεται βάσει της μεθόδου Drop Tail. Υποθέτουμε ότι το σενάριο μας δεν παρουσιάζει καθόλου σφάλματα και χρησιμοποιούμε τις εκδόσεις του πρωτοκόλλου μεταφοράς TCP SACK και TCP NewReno με SNACK. Χρησιμοποιούμε διάφορες τιμές καθυστέρησης (από 10 msec μέχρι 100 λεπτά) και ταυτόχρονα σε κάθε περίπτωση θέτουμε το μέγεθος των καταχωρητών ίσο με το Bandwidth_X_Delay_Product και το χρόνο της προσομοίωσης 200 φορές μεγαλύτερο από την τιμή της καθυστέρησης.

Παρακάτω δίνονται οι γραφικές παραστάσεις του Goodput συναρτήσει της καθυστέρησης για τις δύο διαφορετικές εκδόσεις των πρωτοκόλλων μεταφοράς.





Παρατηρούμε ότι το Goodput μειώνεται δραματικά, καθώς αυξάνεται ο χρόνος καθυστέρησης, ενώ για τιμές RTT μεγαλύτερες από 200 δευτερόλεπτα, η πραγματική απόδοση σχεδόν μηδενίζεται.



Το παραπάνω διάγραμμα απεικονίζει τον αριθμό των πακέτων που επαναμεταδίδονται σε κάθε πρωτόκολλο για διάφορες τιμές καθυστέρησης. Παρατηρούμε ότι όσο αυξάνεται η καθυστέρηση, τόσο αυξάνεται και το πλήθος των πακέτων που επαναμεταδίδονται, δηλαδή έχουμε μεγαλύτερη άσκοπη κατανάλωση ενέργειας. Συγκρίνοντας τα retransmitted packets των δύο πρωτοκόλλων (TCP SACK και TCP NewReno με SNACK) παρατηρούμε ότι για μεγάλες τιμές RTT, το πρώτο επαναμεταδίδει σημαντικά μικρότερο πλήθος πακέτων από το δεύτερο.

4.1.2. Μέγεθος καταχωρητών

Στο σενάριο αυτό οι παράμετροι παραμένουν ίδιοι σε σχέση με το προηγούμενο, με μοναδική αλλαγή το μέγεθος των καταχωρητών. Δοκιμάζουμε τιμές μεγέθους καταχωρητή:

2_X_Bandwidth_X_Delay_Product

0.5_X_Bandwidth_X_Delay_Product

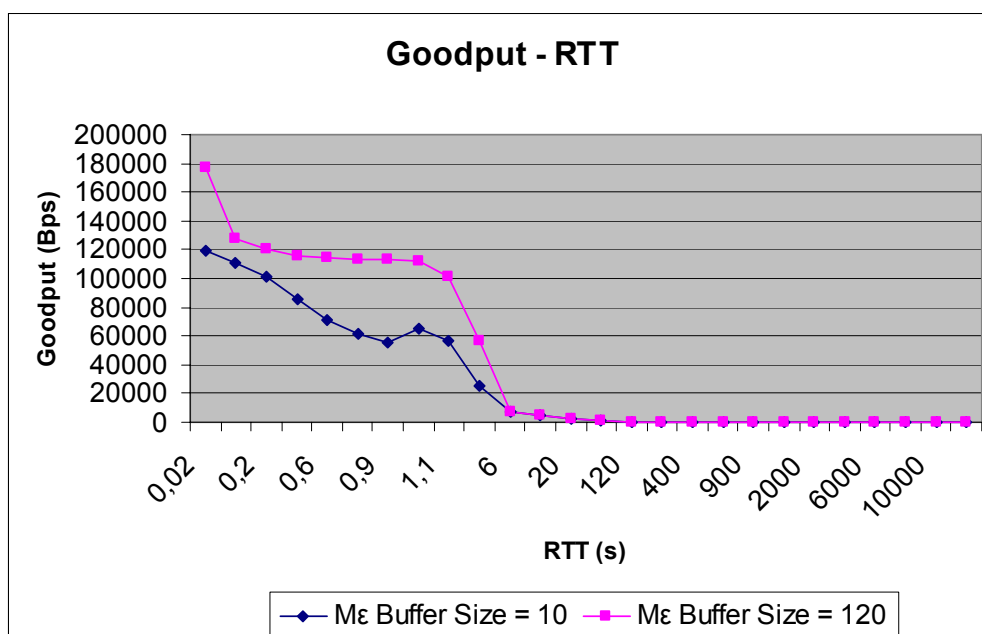
0.2_X_Bandwidth_X_Delay_Product

0.1_X_Bandwidth_X_Delay_Product

και παρατηρούμε πως τα αποτελέσματα παραμένουν ίδια για όλες τις τιμές.

Χρησιμοποιώντας ακόμη μικρότερες τιμές, παρατηρούμε πως για μέγεθος καταχωρητή ίσο με 120 πακέτα παίρνουμε τη μέγιστη δυνατή τιμή του Goodput. Δηλαδή, όσο και να αυξήσουμε το μέγεθος του καταχωρητή πάνω από 120 πακέτα, δεν θα σημειωθεί αλλαγή στα αποτελέσματα μας. Αντίθετα, αν το μειώσουμε, τότε η τιμή της απόδοσης θα μειωθεί.

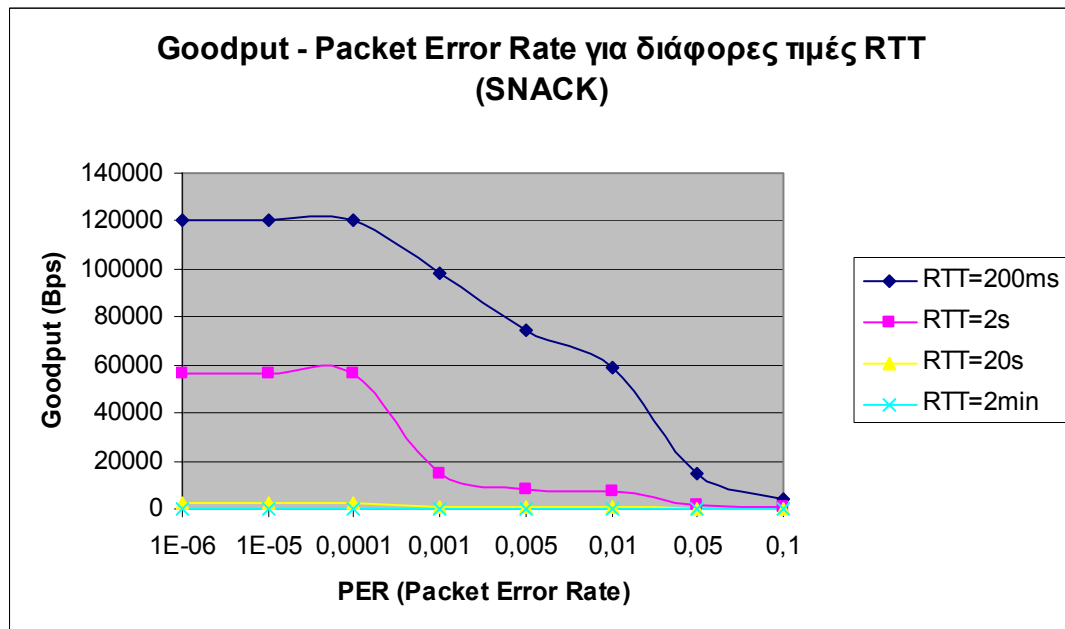
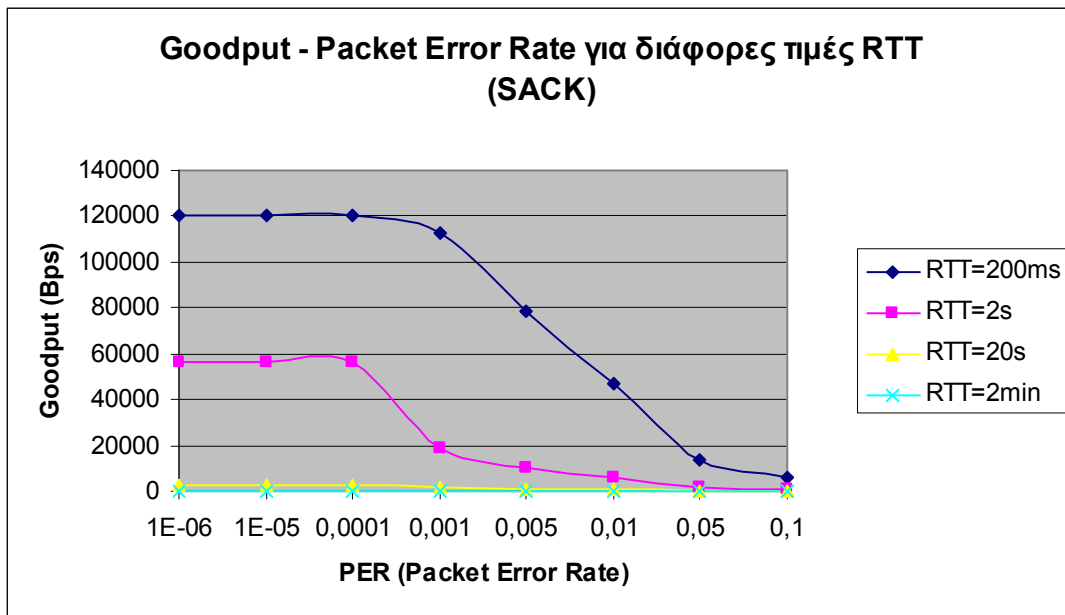
Παρακάτω δίνεται η γραφική παράσταση του Goodput συναρτήσει της καθυστέρησης, για το πρωτόκολλο μεταφοράς TCP SACK, για μεγέθη καταχωρητή ίσα με 10 και 120 πακέτα. Παρατηρούμε πως για μικρές τιμές καθυστερήσεων, το μέγεθος του καταχωρητή παίζει σημαντικό ρόλο, καθώς η απόδοση με Buffer Size ίσο με 120 είναι πολύ καλύτερη από την αντίστοιχη για Buffer Size ίσο με 10 πακέτα. Αντίθετα, για μεγάλες τιμές καθυστερήσεων η απόδοση είναι ανεξάρτητη από το μέγεθος των καταχωρητών.



4.1.3. Ρυθμός σφαλμάτων ανά πακέτο (PER-Packet Error Rate)

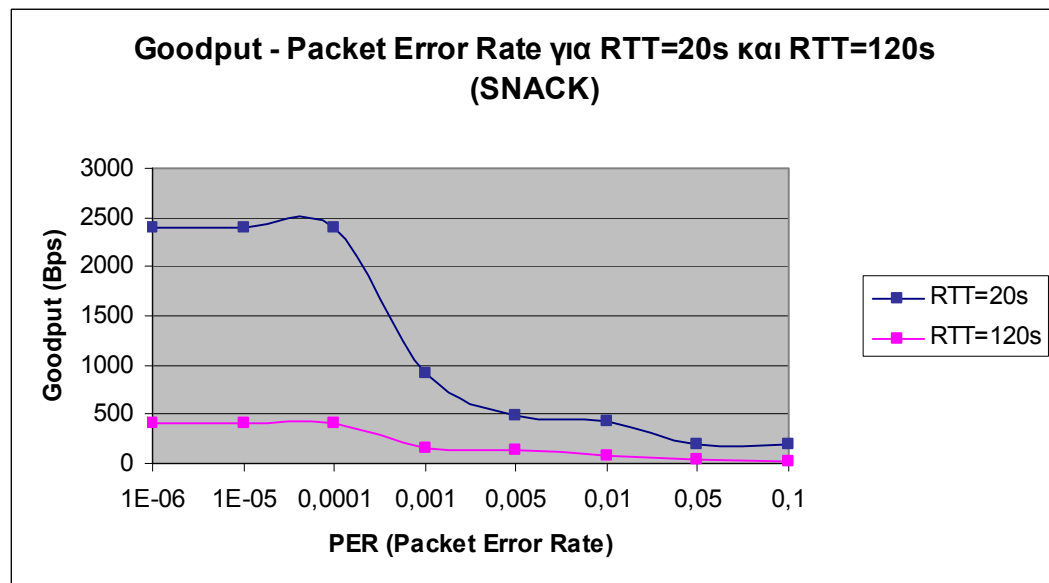
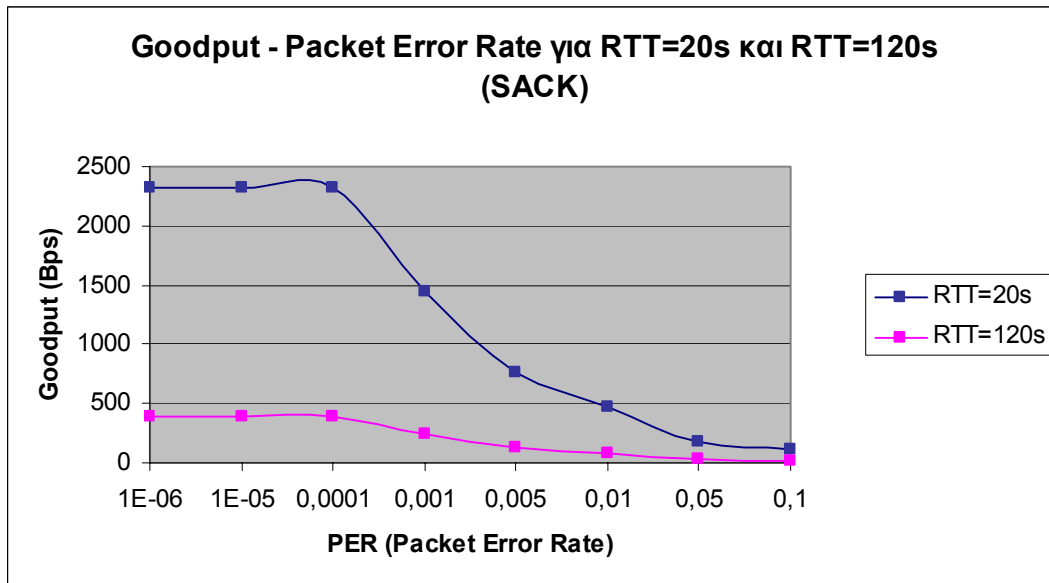
Στο σενάριο αυτό προσομοιώνεται μια τοπολογία ίδια με αυτή του πρώτου σεναρίου, με τη διαφορά ότι πλέον η σύνδεση παρουσιάζει σφάλματα, με PER που κυμαίνεται από 10^{-6} έως 0,1.

Ακολουθούν οι γραφικές παραστάσεις του Goodput συναρτήσει του PER, για διάφορες τιμές καθυστέρησης και για τα δύο πρωτόκολλα μεταφοράς.

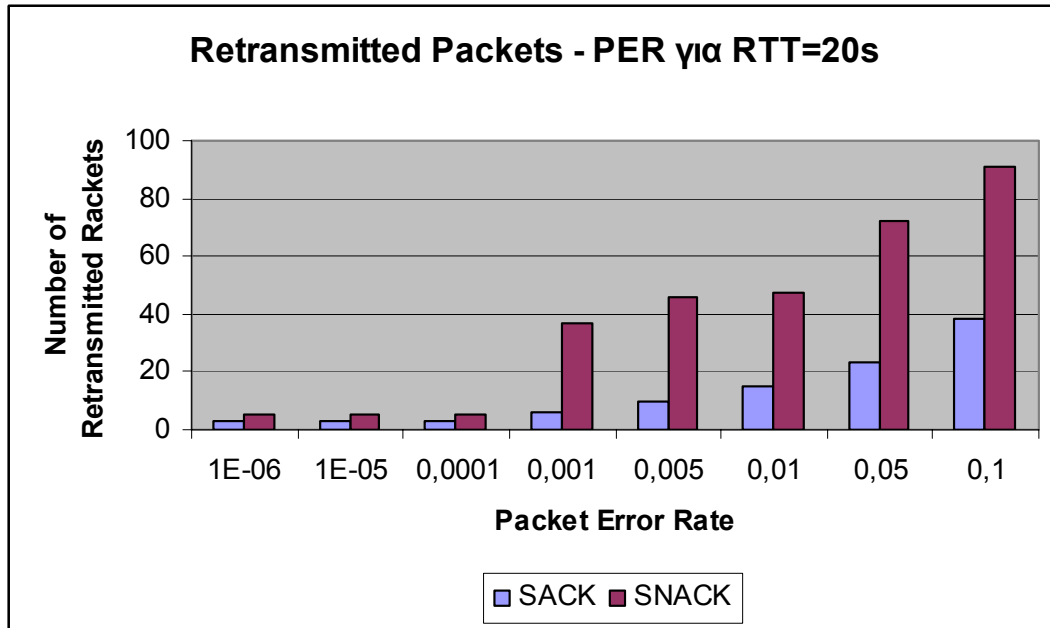


Παρατηρούμε ότι όσον αφορά την απόδοση και τα δύο πρωτόκολλα παρουσιάζουν σχεδόν την ίδια συμπεριφορά. Η πτώση της απόδοσης όσο αυξάνεται ο ρυθμός σφαλμάτων ανά πακέτο είναι σημαντική, ειδικά για μεγάλες τιμές ρυθμού σφαλμάτων. Αξίζει να σημειωθεί ότι για τιμές PER μέχρι της τάξης του 10^{-4} η απόδοση παραμένει ανεπηρέαστη από τα σφάλματα.

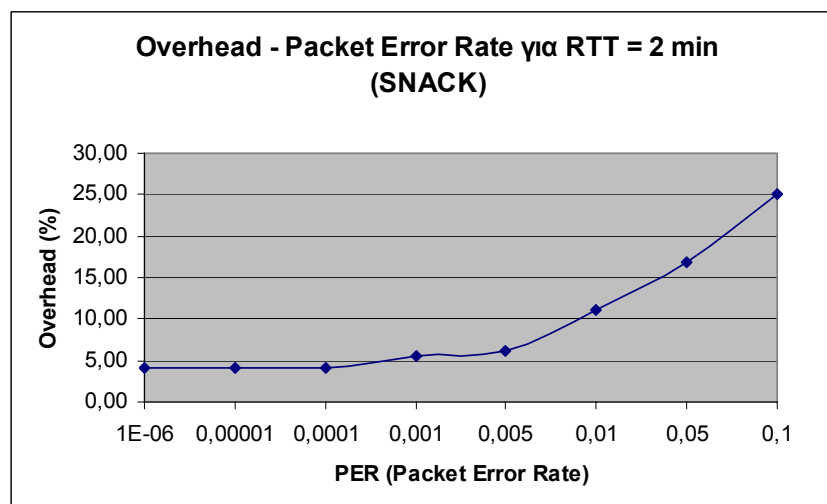
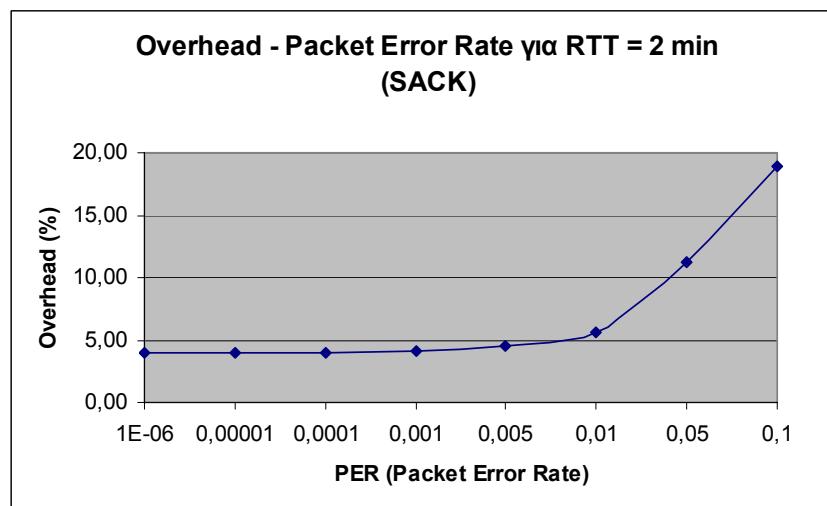
Οι παρακάτω κυματομορφές είναι διευκρινιστικές των προηγούμενων γραφημάτων για RTT ίσο με 20s και 120s, αφού σε αυτά απεικονίζονται ως μηδενικές.



Παρακάτω δίνεται γραφικά το πλήθος των Retransmitted Packets και για τα δύο πρωτόκολλα για αυξανόμενο ρυθμό σφαλμάτων ανά πακέτο. Είναι προφανές πως στο TCP NewReno με SNACK το πλήθος των retransmitted packets είναι αρκετά μεγαλύτερο.

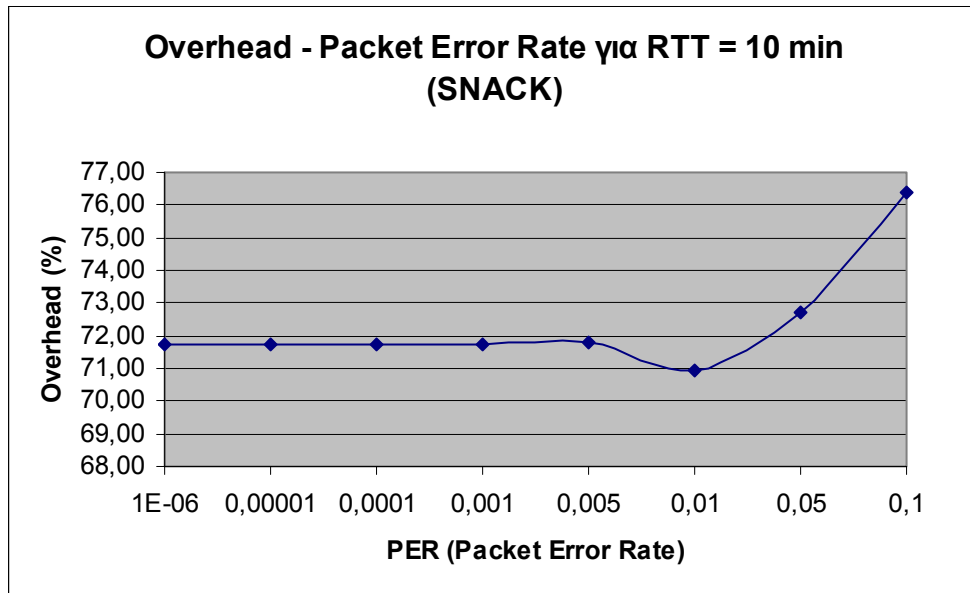


Στις παρακάτω γραφικές παραστάσεις απεικονίζεται η ποσοστιαία αύξηση της επιβάρυνσης για αυξανόμενο ρυθμό σφαλμάτων.



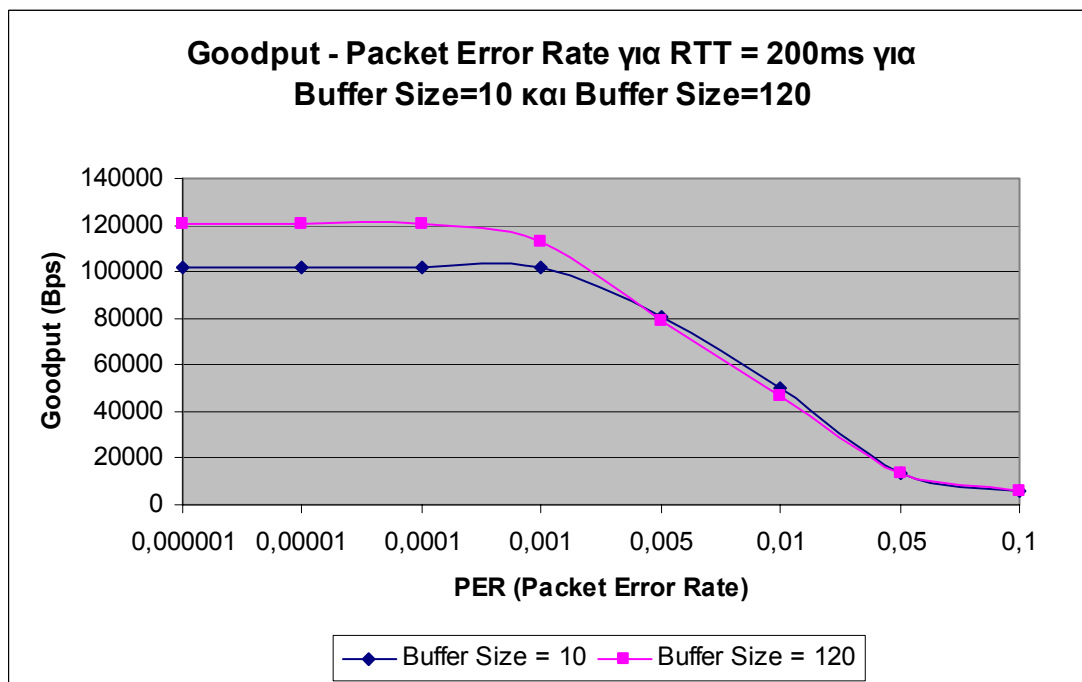
Παρατηρούμε πως για αυξανόμενο ρυθμό σφαλμάτων, η επιβάρυνση αυξάνεται σημαντικά. Σημειώνεται ότι το πρωτόκολλο TCP New Reno με SNACK παρουσιάζει μεγαλύτερο ποσοστό επιβάρυνσης από το TCP SACK για μεγάλες τιμές PER.

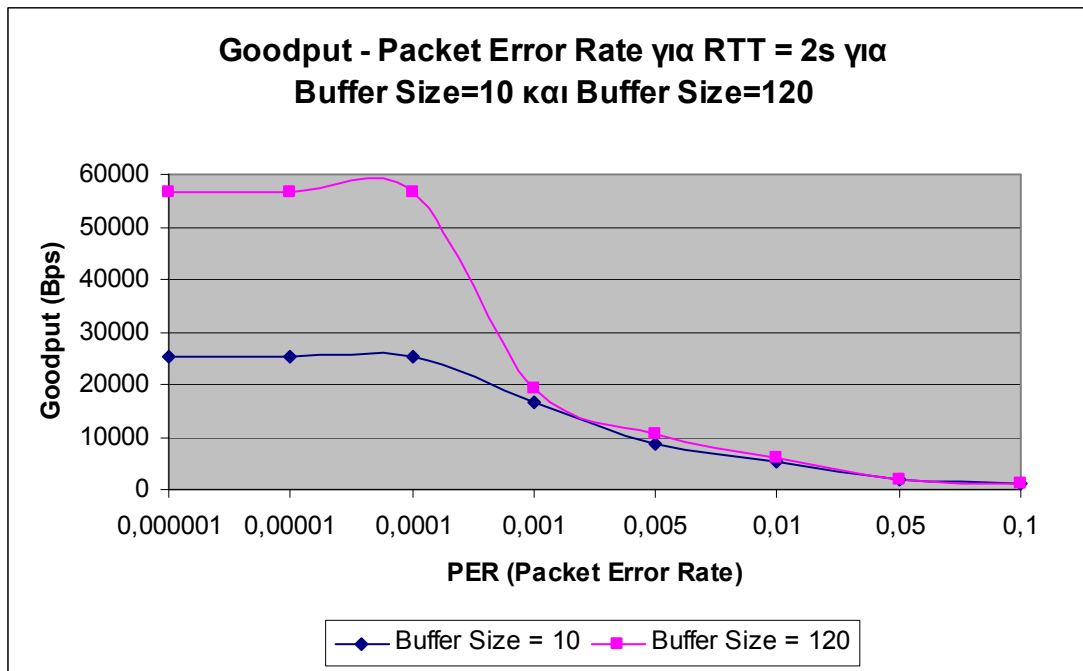
Παρακάτω δίνεται η γραφική παράσταση του ποσοστού επιβάρυνσης συναρτήσει του ρυθμού σφαλμάτων για RTT=10 λεπτά.



Είναι προφανές πως για μεγάλες τιμές καθυστέρησης, το ποσοστό της επιβάρυνσης είναι τεράστιο, γεγονός που καθιστά τη χρήση του πρωτοκόλλου ασύμφορη.

Για το πρωτόκολλο TCP SACK εκτελέστηκε το τρίτο πείραμα για μέγεθος καταχωρητή ίσο με 10 και 120 πακέτα και τα αποτελέσματα δίνονται παρακάτω.





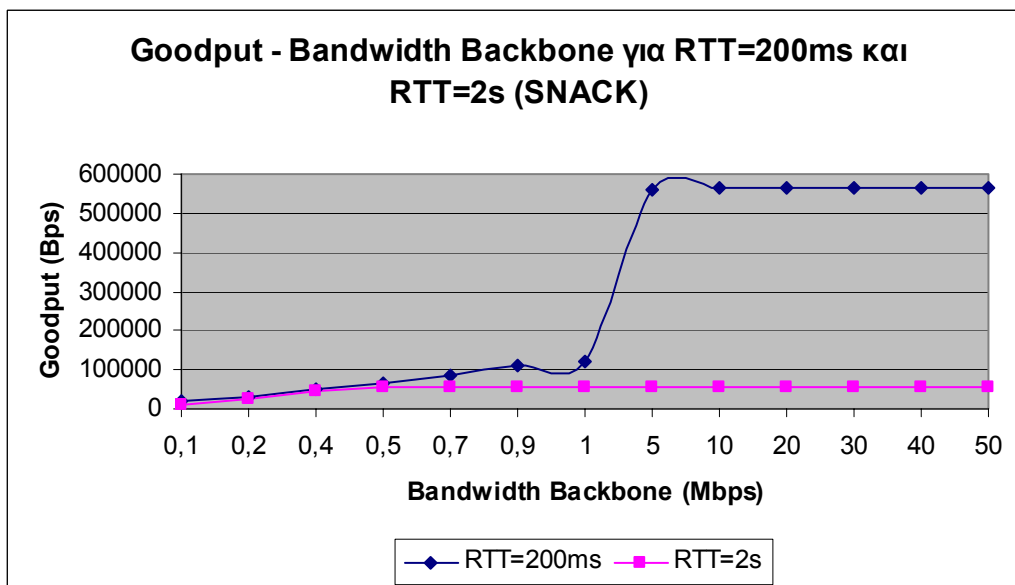
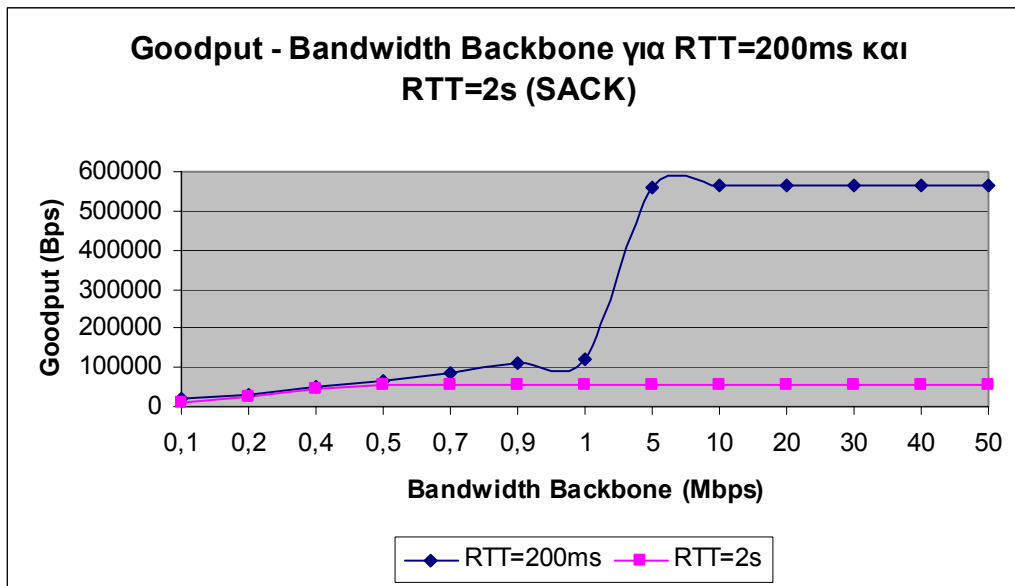
Στην περίπτωση αυτή παρατηρούμε πως για μικρές τιμές ρυθμού σφαλμάτων, το μέγεθος του καταχωρητή παίζει σημαντικό ρόλο στην απόδοση του πρωτοκόλλου. Αντίθετα, για τιμές PER μεγαλύτερες από 0,001 η επίδραση του μεγέθους του καταχωρητή στην απόδοση είναι σχεδόν μηδενική.

4.1.4. Χωρητικότητα της κύριας σύνδεσης (Bandwidth Backbone)

Στο σενάριο αυτό προσομοιώνεται μια τοπολογία τύπου simple, για διάφορες τιμές καθυστέρησης, αλλάζοντας σε κάθε περίπτωση την τιμή της χωρητικότητας της κύριας σύνδεσης (με τιμές από 0,1Mbps έως 50Mbps). Ο τύπος της εφαρμογής που “τρέχουν” οι χρήστες του δικτύου είναι πάλι μεταφορά μεγάλου όγκου δεδομένων και η διαχείριση της ουράς δεδομένων του καταχωρητή του δικτύου γίνεται βάσει της μεθόδου DropTail. Υποθέτουμε πως το σενάριο μας δεν παρουσιάζει καθόλου σφάλματα και οι δύο εκδόσεις του πρωτοκόλλου μεταφοράς που χρησιμοποιούμε είναι και πάλι η TCP SACK και η TCP NewReno με SNACK.

Από τα αποτελέσματα των μετρήσεων παρατηρούμε πως για τιμές καθυστερήσεων μεγαλύτερες από 10 δευτερόλεπτα, η τιμή της απόδοσης παραμένει σταθερή, ανεξάρτητα από τη χωρητικότητα της κύριας σύνδεσης. Δηλαδή, η χωρητικότητα επηρεάζει την τιμή της απόδοσης του πρωτοκόλλου μόνο για μικρές καθυστερήσεις.

Συγκριτικά τα δύο πρωτόκολλα δεν παρουσιάζουν σχεδόν καμία διαφορά στην απόδοση. Ενδεικτικά δίνονται παρακάτω οι γραφικές παραστάσεις του Goodput συναρτήσει της χωρητικότητας της σύνδεσης για RTT ίσο με 200ms και 2s και για τα δύο πρωτόκολλα.

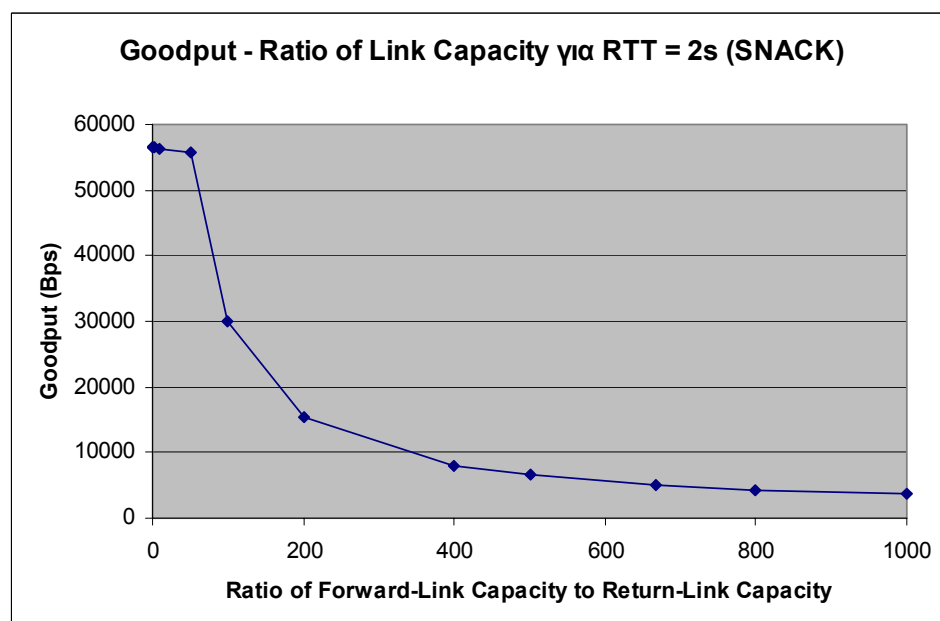
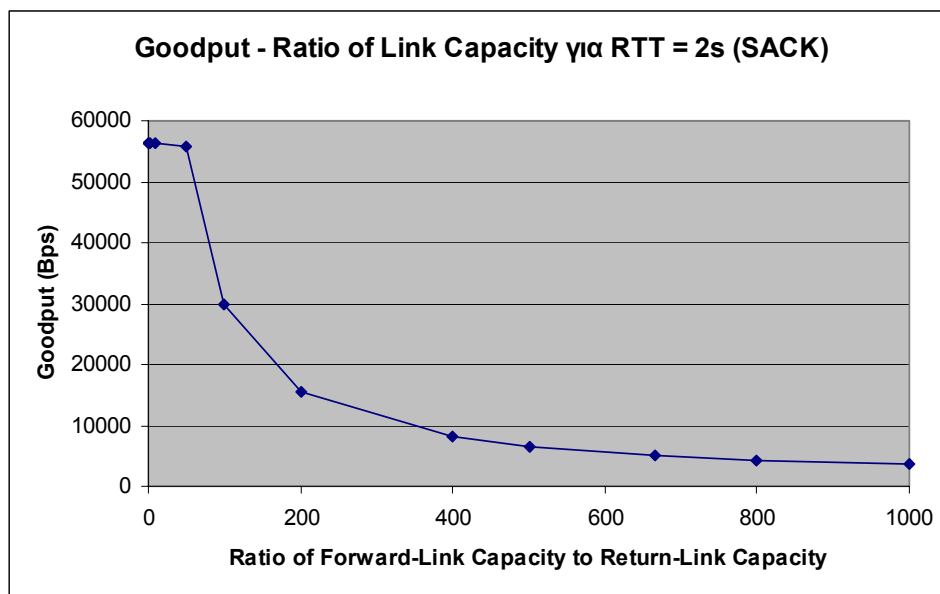


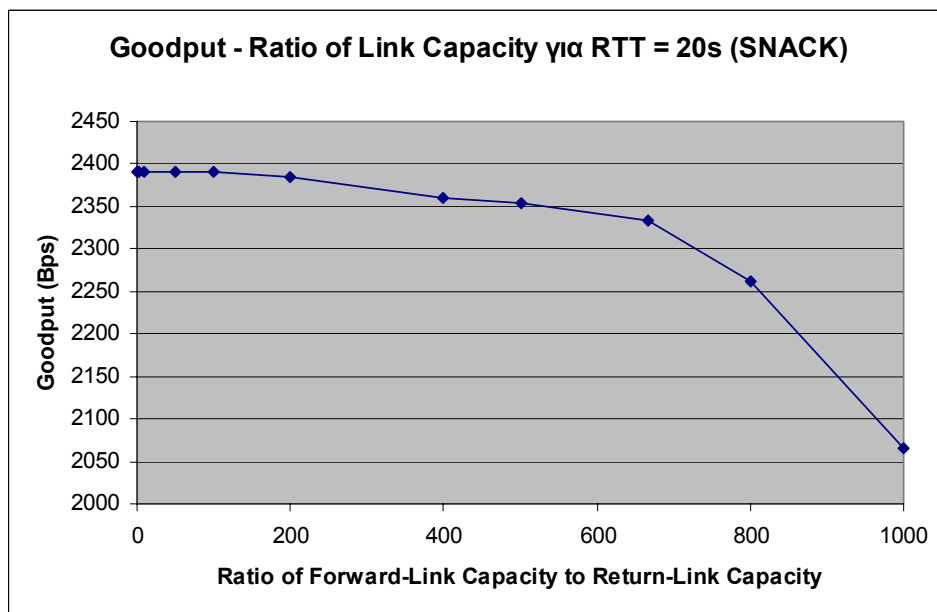
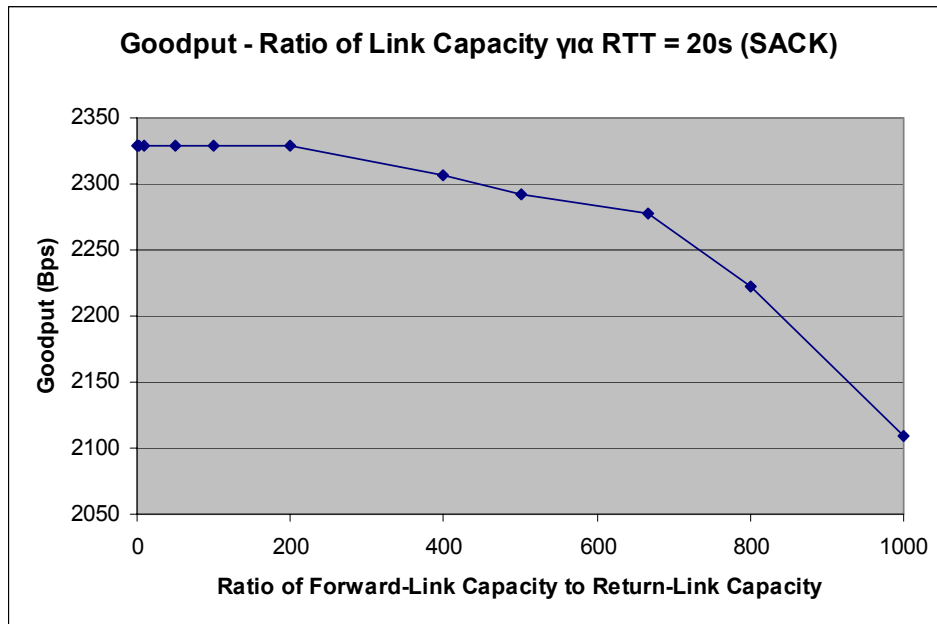
Παρατηρούμε πως για μικρές καθυστερήσεις η απόδοση είναι σημαντικά μεγαλύτερη, κάτι που είναι αναμενόμενο. Επίσης, σημειώνεται πως πέρα από κάποια συγκεκριμένη τιμή χωρητικότητας (στην περίπτωση μας 5MB), η αύξηση της χωρητικότητας δεν έχει κάποια επίδραση στην τιμή της απόδοσης.

4.1.5. Ασύμμετρα κανάλια

Στο σενάριο που ακολουθεί προσομοιώνεται μια ασύμμετρη τοπολογία τύπου simple, με λόγο χωρητικότητας καναλιού δεδομένων προς καναλιού επιβεβαιώσεων που κυμαίνεται από 2:1 έως 1000:1. Η εφαρμογή είναι μεταφορά μεγάλου όγκου δεδομένων, η διαχείριση της ουράς δεδομένων είναι DropTail. Εκτελούμε πειράματα με χωρητικότητα forward καναλιού 1Mb, για διάφορες τιμές καθυστερήσεων, τόσο για το πρωτόκολλα μεταφοράς TCP SACK, όσο για το πρωτόκολλο NewReno με SNACK.

Ακολουθούν γραφικές παραστάσεις της απόδοσης συναρτήσει του λόγου του καναλιού δεδομένων προς το κανάλι επιβεβαιώσεων για τιμές καθυστέρησης 2s και 20s.



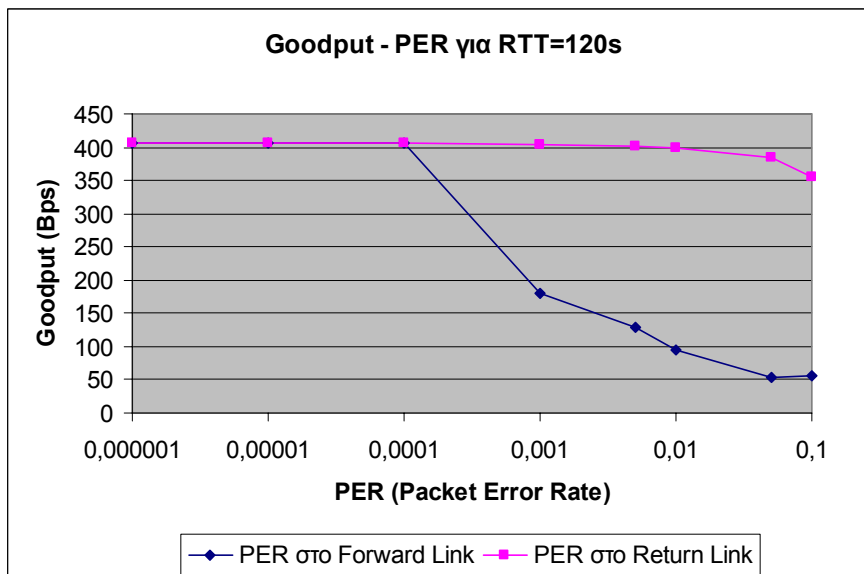
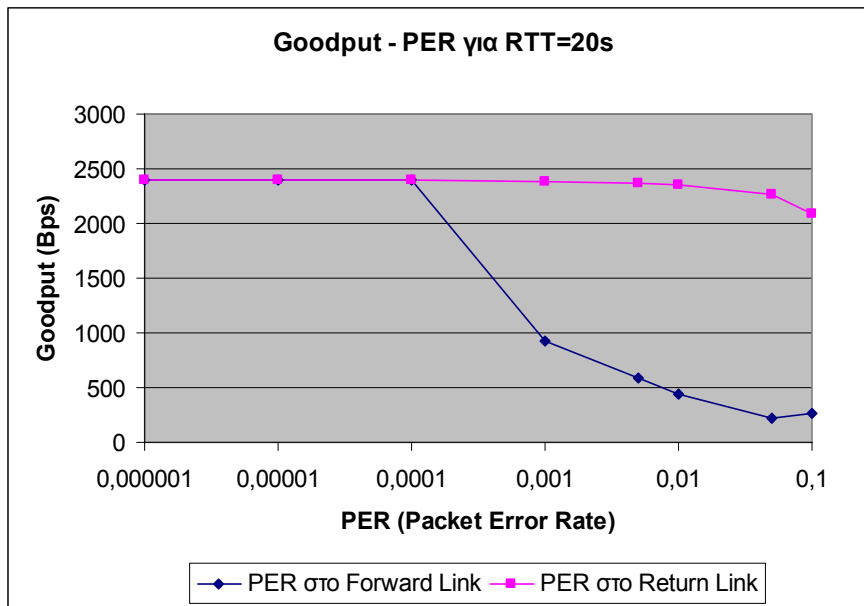


Σε όλες τις περιπτώσεις παρατηρούμε πως η πραγματική απόδοση επηρεάζεται σημαντικά από το λόγο των δύο καναλιών και μειώνεται σημαντικά, όσο ο λόγος αυτός αυξάνεται.

4.1.6. Σφάλματα στο forward link και στο return link

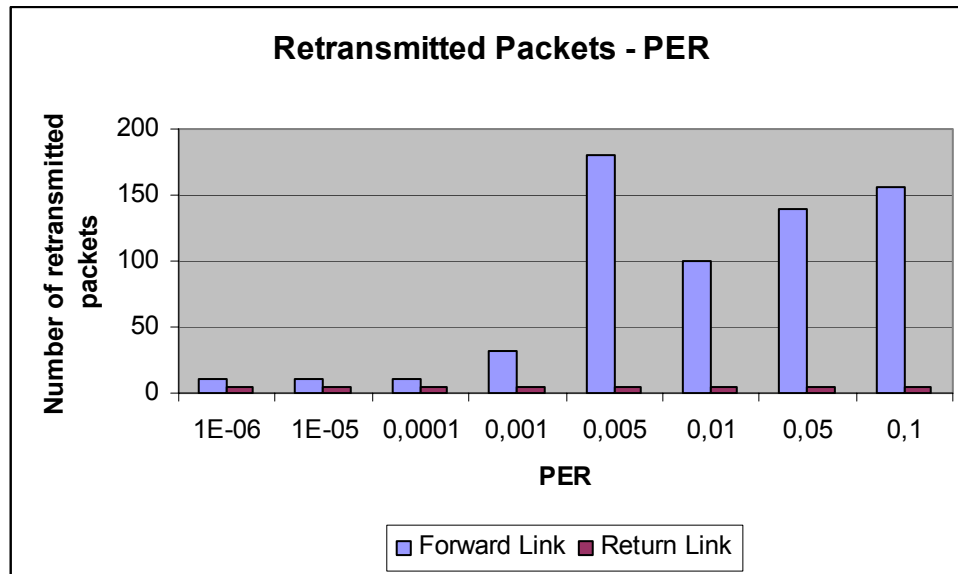
Στο σενάριο αυτό προσομοιώνουμε μια τοπολογία ίδια με αυτή που περιγράψαμε στο τρίτο σενάριο, με τη μόνη διαφορά πως πλέον τα σφάλματα δεν εμφανίζονται με την ίδια πιθανότητα και στα δύο κανάλια (forward link και return link). Στην πρώτη περίπτωση τα σφάλματα εμφανίζονται στο forward link, ενώ το return link δεν παρουσιάζει καθόλου σφάλματα. Δηλαδή, πλέον “χάνονται” μόνο πακέτα και όχι επιβεβαιώσεις. Αντίστοιχα, στη δεύτερη περίπτωση σφάλματα εμφανίζονται μόνο στο return link, με αποτέλεσμα να “χάνονται” μόνο επιβεβαιώσεις.

Παρακάτω δίνονται γραφικές παραστάσεις του Goodput σε συνάρτηση με το PER και για τις δύο περιπτώσεις για τιμές καθυστερήσεων 20s και 120s.



Είναι προφανές πως η απώλεια πακέτων επηρεάζει αρνητικά την απόδοση, σε αντίθεση με την απώλεια επιβεβαιώσεων, η οποία έχει μικρή (και για μικρά PER μηδενική) αρνητική επίδραση στην τελική τιμή της απόδοσης.

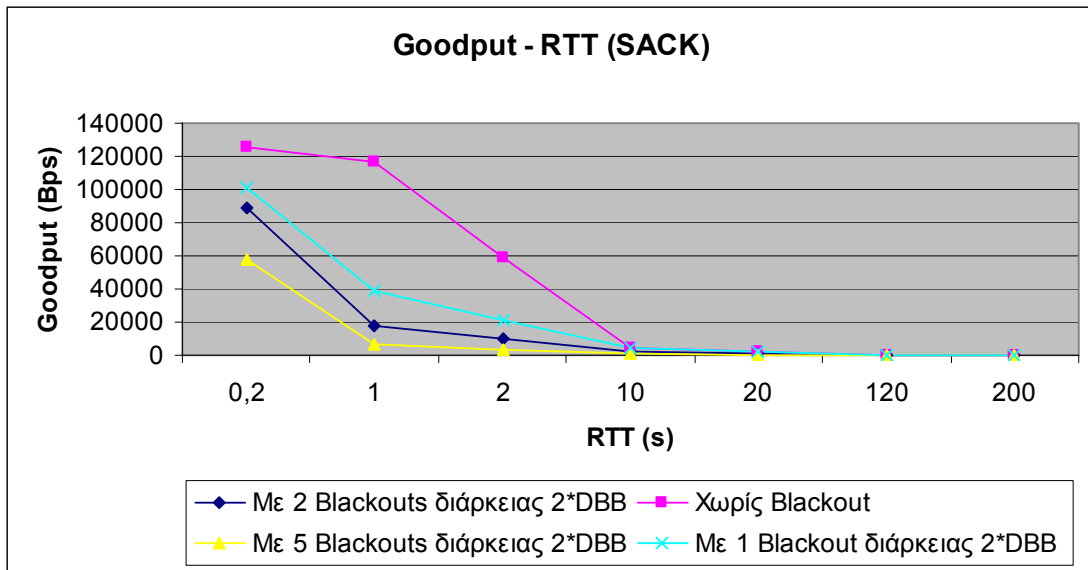
Αξιοσημείωτη είναι ακόμη η μεγάλη διαφορά στο πλήθος των retransmitted packets για τις δύο αυτές περιπτώσεις. Στην περίπτωση που “χάνονται” πακέτα, το πλήθος αυτών που θα επαναμεταδοθούν είναι πολύ μεγάλο, σε αντίθεση με την περίπτωση που “χάνονται” οι επιβεβαιώσεις, οπότε ο αριθμός των retransmitted packets είναι σχεδόν αμελητέος. Παρακάτω φαίνεται γραφικά αυτή η διαφορά.



4.1.7. Διακοπή συνδεσιμότητας

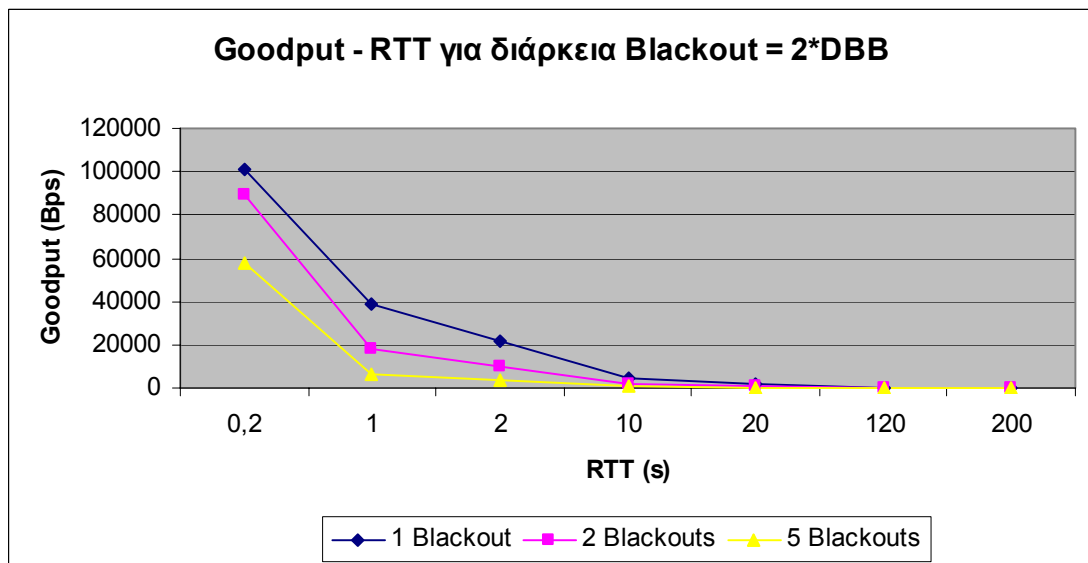
Στο σενάριο αυτό προσομοιώνεται μια τοπολογία ίδια με αυτή του πρώτου σεναρίου (για το πρωτόκολλο TCP SACK), με τη διαφορά ότι πλέον η συνδεσιμότητα δεν είναι συνεχόμενη, αλλά εμφανίζονται διακοπές συνδεσιμότητας (blackouts). Όπως είναι λογικό, οι διακοπές συνδεσιμότητας αποτελούν ένα πρόβλημα για τη σύνδεση μας και επηρεάζουν αρνητικά την απόδοση του πρωτοκόλλου.

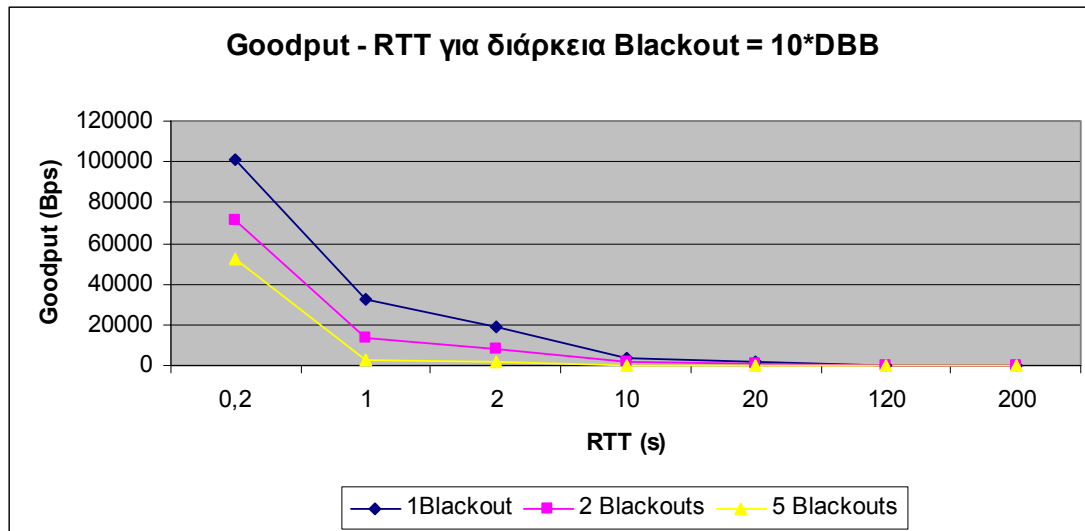
Παρακάτω δίνεται η γραφική παράσταση της πραγματικής απόδοσης σε συνάρτηση με την καθυστέρηση για μηδενική, μια, δύο και πέντε διακοπές συνδεσιμότητας, διάρκειας RTT η καθεμία.



Είναι προφανές πως όσο αυξάνεται το πλήθος των blackouts, τόσο μειώνεται η απόδοση του πρωτοκόλλου.

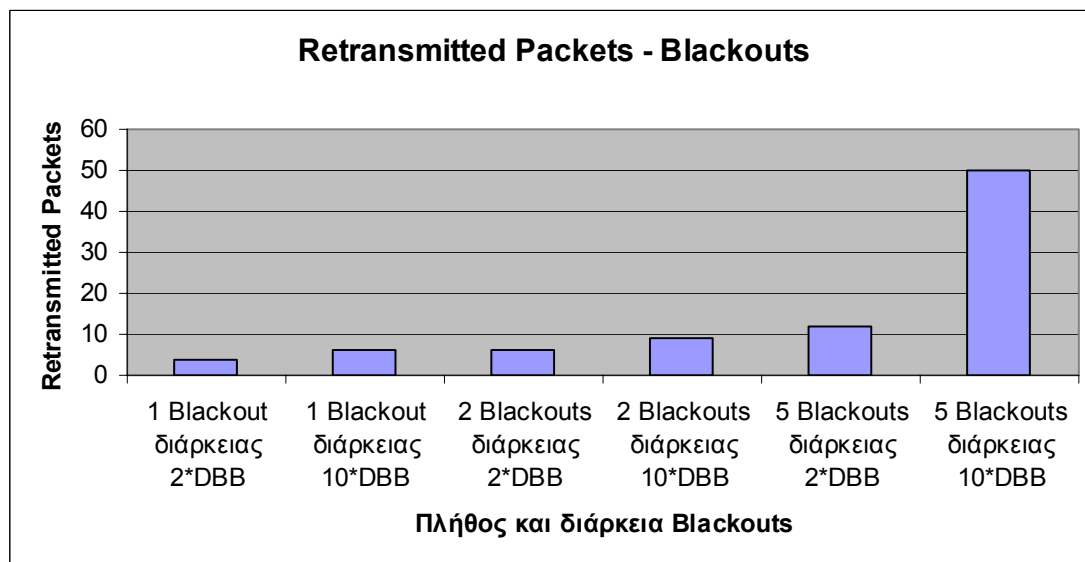
Ακολουθούν δύο γραφικές παραστάσεις του Goodput συναρτήσει της καθυστέρησης για διαφορετική διάρκεια διακοπών συνδεσιμότητας.





Συγκρίνοντας τα αποτελέσματα από τις παραπάνω δυο κυματομορφές γίνεται κατανοητό πως, εκτός από το πλήθος των διακοπών συνδεσιμότητας, σημαντικό ρόλο παίζει ο χρόνος που αυτές διαρκούν. Παρατηρούμε πως για μια μόνο διακοπή συνδεσιμότητας τα αποτελέσματα είναι σχεδόν ίδια, άσχετα από τη διάρκεια της. Αντίθετα, αν έχουμε περισσότερα blackouts (και συνεπώς ο συνολικός χρόνος μη λειτουργίας είναι μεγαλύτερος), η απόδοση μειώνεται αισθητά με την αύξηση της διάρκειας της διακοπής.

Στο παρακάτω γράφημα δίνεται το πλήθος των πακέτων που επαναμεταδίδονται συναρτήσει του πλήθους και της διάρκειας των blackouts.

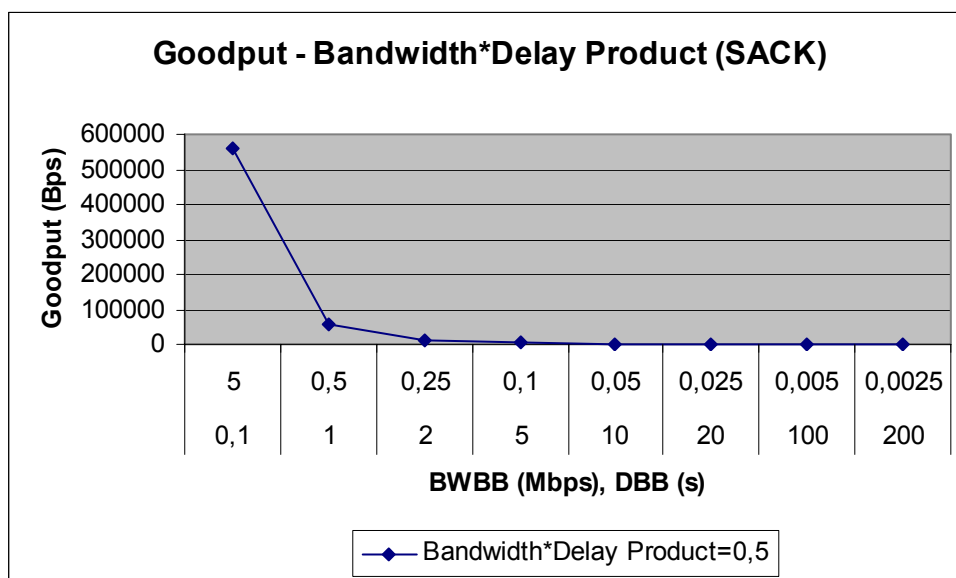


Παρατηρείται μια συνεχής αύξηση των Retransmitted Packets με την αύξηση του πλήθους και της διάρκειας των διακοπών συνδεσιμότητας. Από ενεργειακή άποψη, αυτό σημαίνει πως έχουμε μεγαλύτερη άσκοπη κατανάλωση ενέργειας για αυξανόμενους συνολικούς χρόνους διακοπής συνδεσιμότητας.

4.1.8. Bandwidth X Delay Product

Στο σενάριο αυτό προσομοιώνονται επτά διαφορετικές περιπτώσεις, οι οποίες έχουν το ίδιο Delay X Bandwidth Product, αλλά διαφορετική καθυστέρηση και χωρητικότητα κύριας σύνδεσης. Ο τύπος της εφαρμογής που “τρέχουν” οι χρήστες του δικτύου είναι μεταφορά μεγάλου όγκου δεδομένων (Bulk Data Transfer) και η διαχείριση της ουράς δεδομένων των καταχωρητών του δικτύου γίνεται βάσει της μεθόδου Drop Tail. Υποθέτουμε ότι το σενάριο μας δεν παρουσιάζει καθόλου σφάλματα και χρησιμοποιούμε την έκδοση του πρωτοκόλλου μεταφοράς TCP SACK. Ο συνολικός χρόνος της προσομοίωσης 200 φορές μεγαλύτερος από την τιμή της καθυστέρησης.

Τα αποτελέσματα της προσομοίωσης δίνονται στο διάγραμμα που ακολουθεί.



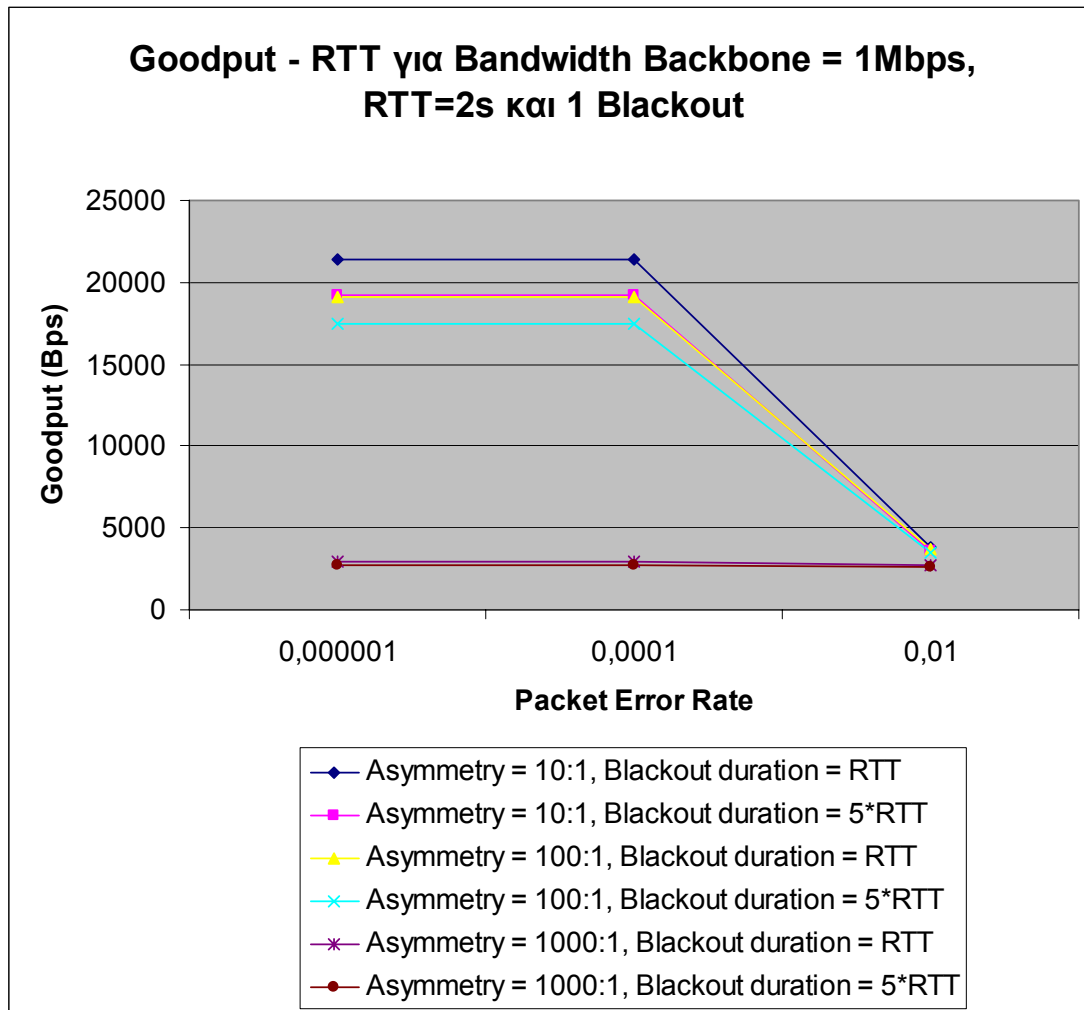
Είναι προφανές πως σημαντικότερο ρόλο στην απόδοση του πρωτοκόλλου παίζει η τιμή της καθυστέρησης, παρά η χωρητικότητα της κύριας σύνδεσης. Μάλιστα, η απόδοση μειώνεται ραγδαία όσο η καθυστέρηση αυξάνεται, άσχετα από την αύξηση της χωρητικότητας. Αιτία αποτελεί ο Ack-clocking μηχανισμός μετάδοσης του TCP.

4.1.9. Συνδυασμός προβλημάτων (PER, Asymmetry, Blackout)

Στο σενάριο αυτό προσομοιώνεται μια τοπολογία τύπου simple, όπου η χωρητικότητα της κύριας σύνδεσης είναι 1Mbps. Ο τύπος της εφαρμογής που “τρέχουν” οι χρήστες του δικτύου είναι μεταφορά μεγάλου όγκου δεδομένων (Bulk Data Transfer) και η διαχείριση της ουράς δεδομένων των καταχωρητών του δικτύου γίνεται βάσει της μεθόδου Drop Tail. Η έκδοση του πρωτοκόλλου μεταφοράς που χρησιμοποιούμε είναι η TCP SACK και ο χρόνος της προσομοίωσης ισούται με 200 φορές την τιμή της καθυστέρησης. Σε κάθε περίπτωση παρατηρείται μια διακοπή συνδεσιμότητας διαφορετικής διάρκειας.

Για το παραπάνω σενάριο εκτελούμε μετρήσεις για PER που κυμαίνεται από 10^{-6} έως 10^{-2} , λόγο χωρητικότητας των δύο ασύμμετρων καναλιών που κυμαίνεται από 10:1

μέχρι 1000:1 και διάρκεια blackout ίση με RTT και 5RTT. Τα αποτελέσματα που παίρνουμε απεικονίζονται στο παρακάτω γράφημα.



Καταρχήν παρατηρούμε πως η τιμή της απόδοσης παραμένει σταθερή σε όλες τις περιπτώσεις, ανεξάρτητα από το ρυθμό σφαλμάτων ανά πακέτο, για σφάλματα μέχρι 10^{-4} . Για PER μεγαλύτερης τάξης σημειώνεται ραγδαία πτώση της απόδοσης.

Για μικρούς λόγους χωρητικότητας των δύο καναλιών, η πραγματική απόδοση επηρεάζεται σημαντικά από τη διάρκεια της διακοπής συνδεσιμότητας. Βλέπουμε πως τόσο για λόγο 10:1, όσο για λόγο 100:1, η απόδοση μειώνεται αισθητά για μεγάλη διάρκεια blackouts. Αντίθετα, για λόγο καναλιού δεδομένων προς κανάλι επιβεβαιώσεων ίσο με 1000:1, η τελική απόδοση δεν επηρεάζεται καθόλου ούτε από το ρυθμό σφαλμάτων ανά πακέτο, ούτε από τη διάρκεια του blackout.

Όπως ήταν αναμενόμενο, παρατηρούμε, ακόμη, ότι για την ίδια διάρκεια διακοπής συνδεσιμότητας, η απόδοση μειώνεται για αυξανόμενο λόγο ασυμμετρίας των δύο καναλιών.

Τέλος, αξίζει να σημειωθεί πως για μεγάλο ρυθμό σφαλμάτων ανά πακέτο (της τάξης του 10^{-2}) όλες οι περιπτώσεις καταλήγουν στην ίδια τιμή απόδοσης, ανεξάρτητα από τη διάρκεια των blackouts και το λόγο ασυμμετρίας.

4.2. Συμπεράσματα και σχόλια

Οι διάφορες εκδόσεις των πρωτοκόλλων μεταφοράς TCP πετυχαίνουν πολύ καλή απόδοση σε ενσύρματα δίκτυα, δεν είναι όμως κατάλληλες για ενσύρματα δίκτυα και, ιδιαίτερα, για ένα περιβάλλον όπως το Διαπλανητικό Διαδίκτυο.

Καταλήξαμε στο συμπέρασμα πως η απόδοση των πρωτοκόλλων μεταφοράς TCP μειώνεται, καθώς ο χρόνος καθυστέρησης μετ' επιστροφής RTT αυξάνεται, και για τιμές RTT μεγαλύτερες από 10 δευτερόλεπτα, η απόδοση σχεδόν μηδενίζεται. Αυτό καθιστά τα ισχύοντα πρωτόκολλα μεταφοράς κατάλληλα μόνο για μικρές αποστάσεις, αλλά μη αποδοτικά (τουλάχιστον χωρίς τροποποιήσεις) για μεγάλες αποστάσεις από τη Γη.

Αποδείχθηκε πως σημαντικό ρόλο στην απόδοση των πρωτοκόλλων παίζει ο ρυθμός σφαλμάτων ανά πακέτο. Παρατηρήθηκε πως για PER μέχρι της τάξης του 10^{-4} η απόδοση δεν επηρεάζεται από σφάλματα. Όμως, για μεγαλύτερες τιμές παρατηρείται ραγδαία αύξηση της επιβάρυνσης και, συνεπώς, πτώση της απόδοσης, ανεξάρτητα από τη τιμή της καθυστέρησης. Το γεγονός αυτό οδηγεί σε σημαντική αύξηση των πακέτων που επαναμεταδίδονται και, κατά συνέπεια, στην ενέργεια που καταναλώνεται.

Από τα πειράματα που εκτελέστηκαν για απώλειες πακέτων δεδομένων, καταλήξαμε στο συμπέρασμα πως η απόδοση επηρεάζεται σημαντικά (αρνητικά) από τις απώλειες πακέτων. Αντίθετα, η απώλεια επιβεβαιώσεων, όσο μεγάλη και να είναι, δεν επηρεάζει σχεδόν καθόλου τη τελική απόδοση του πρωτοκόλλου.

Ένα άλλο συμπέρασμα στο οποίο καταλήξαμε είναι ότι η ασυμμετρία των καναλιών δεδομένων-επιβεβαιώσεων οδηγεί σε ακαταλληλότητα των ισχυόντων πρωτοκόλλων TCP. Αυτό προκύπτει από τη μελέτη και των δύο πρωτοκόλλων μεταφοράς (TCP SACK και TCP New Reno με SNACK) για διάφορες τιμές καθυστέρησης. Σε όλες τις περιπτώσεις η αύξηση της ασυμμετρίας καταλήγει σε κατακόρυφη πτώση της απόδοσης. Τα ισχύοντα πρωτόκολλα μεταφοράς φαίνονται να πετυχαίνουν καλή απόδοση μόνο για λόγους ασυμμετρίας μέχρι 50:1.

Όσον αφορά τις διακοπές συνδεσιμότητας, παρατηρήθηκε ότι η απόδοση των πρωτοκόλλων επηρεάζεται τόσο από το πλήθος των διακοπών, όσο από τη διάρκεια τους. Ουσιαστικά, η απόδοση είναι αντιστρόφως ανάλογη του συνολικού χρόνου των blackout που εμφανίζονται σε μια σύνδεση.

Συγκριτικά, καταλήξαμε στο συμπέρασμα πως για μεγάλους ρυθμούς σφαλμάτων ανά πακέτο, τα σφάλματα υπερισχύουν τόσο των διακοπών συνδεσιμότητας, όσο της ασυμμετρίας των καναλιών. Γενικότερα, η εφαρμογή των ισχυόντων πρωτοκόλλων μεταφοράς σε διαστημικό περιβάλλον για μεγάλες αποστάσεις κρίνεται ακατάλληλη. Θα πρέπει να χρησιμοποιηθούν εναλλακτικοί μηχανισμοί, οι οποίοι θα αντιμετωπίζουν τις προκλήσεις που θέτει το εχθρικό διαστημικό περιβάλλον.

Κεφάλαιο 5

Σχετική Έρευνα

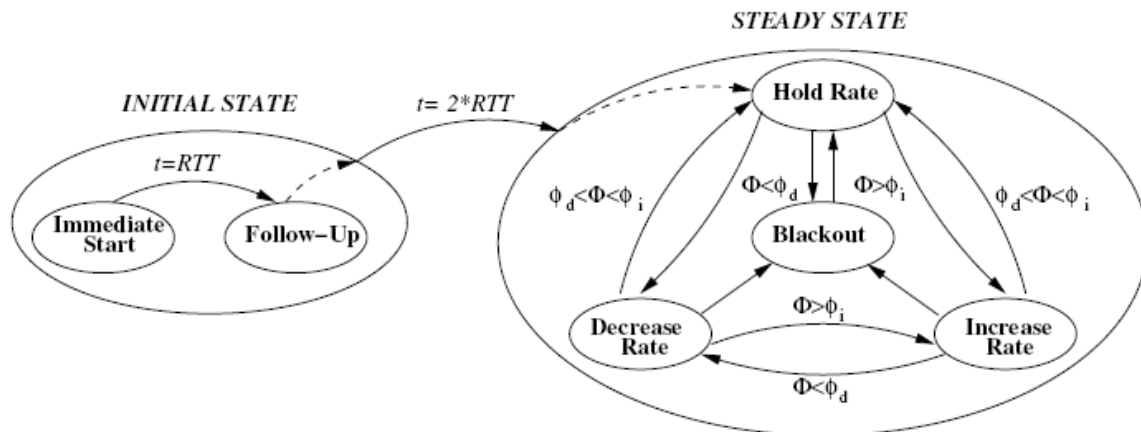
Τα τελευταία χρόνια έχουν γίνει πολλές προσπάθειες να βρεθούν νέα πρωτόκολλα μεταφοράς τα οποία μπορούν να αντιμετωπίσουν τα προβλήματα που δημιουργεί το εχθρικό Διαπλανητικό Διαδίκτυο. Στο κεφάλαιο αυτό περιγράφονται τέσσερις διαφορετικές εναλλακτικές προτάσεις. Το αξιόπιστο πρωτόκολλο μεταφοράς TP Planet που έχει ως σκοπό τη μεταφορά μεγάλου όγκου δεδομένων παρουσιάζεται στη παράγραφο 5.1. Στην επόμενη παράγραφο, αναλύεται το πρωτόκολλο Space Communications Protocol Standards – Transport Protocol (SCPS-TP), το οποίο σχεδιάστηκε και εφαρμόστηκε από τη NASA σε συνεργασία με το Υπουργείο Αμύνης των Η.Π.Α. Στη παράγραφο 5.3 δίνεται μια σύντομη περιγραφή του πρωτοκόλλου μεταφοράς RCP Planet, το οποίο έχει σχεδιαστεί για τη μεταφορά δεδομένων σε πραγματικό χρόνο σε περιβάλλοντα με μεγάλο Bandwidth X Delay Product και υψηλούς ρυθμούς σφαλμάτων. Τέλος, στη παράγραφο 5.4 περιγράφεται το πρωτόκολλο μεταφοράς Saratoga, το οποίο χρησιμοποιείται τα τελευταία χρόνια ενεργά για την αξιόπιστη μεταφορά δεδομένων από δορυφόρους στη Γη.

5.1. TP Planet

Το TP Planet [18] είναι ένα αξιόπιστο πρωτόκολλο μεταφοράς που εφαρμόζεται κυρίως στους κόμβους του InterPlanetary Backbone Network, δηλαδή ο αποστολέας και ο παραλήπτης του TP Planet είναι ο σταθμός-βάση στη Γη και η πλανητική πύλη που συνδέεται στους δορυφόρους αναμετάδοσης που κινούνται σε τροχιά γύρω από τους πλανήτες του εξωτερικού Διαστήματος. Ο σκοπός του TP Planet είναι να πετύχει υψηλή απόδοση και αξιόπιστη μεταφορά δεδομένων αντιμετωπίζοντας τις προκλήσεις του InterPlanetary Backbone Network. Για να αντιμετωπίσει τα προβλήματα που προκύπτουν από τις υψηλές τιμές των καθυστερήσεων διάδοσης, το TP Planet εφαρμόζει ένα νέο έλεγχο συμφόρησης Προσθετικής Αύξησης / Πολλαπλασιαστικής Μείωσης AIMD από άκρο σε άκρο, ο οποίος βασίζεται στο ρυθμό (rate-based), και του οποίου οι παράμετροι AIMD προσαρμόζονται για να αντιμετωπίσουν τη μείωση της απόδοσης. Δυο νέοι αλγόριθμοι, το Initial State και το Steady State, αποτελούν τη δομή του πρωτοκόλλου TP Planet.

Στο Steady State ένας νέος μηχανισμός ανίχνευσης και ελέγχου συμφόρησης εφαρμόζεται για να ελαχιστοποιήσει τις εσφαλμένες αποφάσεις σχετικά με τη συμφόρηση εξαιτίας των υψηλών ρυθμών σφαλμάτων που παρουσιάζουν οι συνδέσεις. Για να μειωθούν οι επιδράσεις των διακοπών λειτουργίας στην απόδοση, το TP Planet ενσωματώνει τη διαδικασία του Blackout State στη λειτουργία του πρωτοκόλλου. Το πρόβλημα της ασυμμετρίας εύρους ζώνης αντιμετωπίζεται με τη χρήση της επιλογής των Καθυστερημένων Επιλεκτικών Επιβεβαιώσεων (delayed SACK). Η αποτίμηση της απόδοσης μέσω πειραμάτων προσομοίωσης αποκαλύπτει πως το TP Planet πετυχαίνει σημαντικά υψηλές τιμές απόδοσης και αντιμετωπίζει τις προκλήσεις του InterPlanetary Backbone Network.

Στην παρακάτω εικόνα δίνεται το διάγραμμα καταστάσεων του πρωτοκόλλου TP Planet, το οποίο εξηγείται αναλυτικά παρακάτω.



Σχήμα 5.1 Διάγραμμα καταστάσεων του TP Planet

Initial State

Ο αλγόριθμος Slow Start που χρησιμοποιείται στα υπάρχοντα πρωτόκολλα TCP, όπως αναφέρθηκε και προηγουμένως, φαίνεται ανεπαρκής για συνδέσεις στο Απομακρυσμένο Διάστημα. Προς αποφυγή της μείωσης της απόδοσης, εξαιτίας του αλγορίθμου Slow Start, το TP Planet εφαρμόζει τον αλγόριθμο Initial State, ο οποίος εκμεταλλεύεται τους πόρους της σύνδεσης με ένα γρήγορο και ελεγχόμενο τρόπο. Ο αλγόριθμος αποτελείται από δύο βασικές διαδικασίες, την Immediate Start και τη Follow-Up.

Immediate Start

Σκοπός της φάσης Immediate Start είναι να εξετάσει το δίκτυο με γρήγορο και ελεγχόμενο τρόπο, έτσι ώστε ο ρυθμός μετάδοσης να αυξηθεί γρήγορα, σύμφωνα με την ανατροφοδότηση που παίρνουμε από τον παραλήπτη.

Σημαντικό στοιχείο του Immediate Start αποτελεί η μετάδοση πακέτων NIL, με σκοπό της εξέταση της πραγματικής κατάστασης των πόρων της σύνδεσης μετά την έναρξη της. Ο λόγος που επιλέχθηκαν τα πακέτα NIL είναι ότι μπορούν να διαχειριστούν στον παραλήπτη και να χρησιμοποιηθούν κατά την επαναμετάδοση πακέτων που χάθηκαν.

Υποθέτοντας πως οι δορυφόροι αναμετάδοσης που χρησιμοποιούνται ως δρομολογητές στο InterPlanetary Backbone Network έχουν τη δυνατότητα να διαμορφώνουν την ουρά σύμφωνα με προτεραιότητας, τα χαμηλής προτεραιότητας πακέτα NIL απορρίπτονται πρώτα σε περίπτωση συμφόρησης. Για το λόγο αυτό, η μετάδοση πακέτων NIL δεν επηρεάζει την απόδοση της πραγματικής μετάδοσης δεδομένων. Αν δεν υπάρχει συμφόρηση, τότε τα πακέτα NIL παραλαμβάνονται και επιστρέφονται οι επιβεβαιώσεις από τον παραλήπτη, γεγονός που αποκαλύπτει πως υπάρχουν και άλλοι ανεκμετάλλευτοι πόροι στη σύνδεση.

Follow-Up Phase

Στη φάση αυτή, η ανατροφοδότηση για τα πακέτα που μεταδόθηκαν κατά τη φάση του Immediate Start αρχίζει να παραλαμβάνεται από τον αποστολέα. Για να εκμεταλλευτούν οι ελάχιστοι πόροι του καναλιού των επιβεβαιώσεων στις ασύμμετρες συνδέσεις στο Απομακρυσμένο Διάστημα, ο παραλήπτης δεν επιβεβαιώνει όλα τα πακέτα που παραλαμβάνει. Αφού τα πακέτα NIL μεταδόθηκαν για να εξετάσουν τη κατάσταση της σύνδεσης, κάθε λαμβανόμενο πακέτο NIL αποτελεί ένδειξη για ύπαρξης αχρησιμοποίητων πόρων στη σύνδεση.

Επιπρόσθετα, κατά τη φάση του Follow-Up, ο αποστολέας αρχίζει να στέλνει πακέτα NIX χαμηλής και υψηλής προτεραιότητας. Τα πακέτα NIX αν και είναι πολύ μικρότερα από τα πακέτα δεδομένων, μεταδίδονται με τον ίδιο ρυθμό. Σκοπός τους είναι η πρόληψη συμφόρησης και η λήψη κατάλληλων αποφάσεων κατά τη φάση του Steady State.

Δημιουργία σύνδεσης

Τα συμβατικά πρωτόκολλα TCP εκτελούν την τριπλή χειραψία (three-way handshake) για τη δημιουργία μιας νέας σύνδεσης. Τα υπάρχοντα πρωτόκολλα στέλνουν αιτήσεις σύνδεσης στην αρχή της σύνδεσης και δεν μεταδίδουν καθόλου δεδομένα μέχρι να φτάσει στον παραλήπτη η επιβεβαίωση της σύνδεσης. Η προσέγγιση αυτή καταλήγει σε μεγάλο ανεκμετάλλευτο εύρος ζώνης για διάρκεια τουλάχιστον ενός RTT σε συνδέσεις στο Απομακρυσμένο Διάστημα. Για την αποφυγή της ανεπάρκειας αυτής, ο αποστολέας δεν περιμένει την επιβεβαίωση της αίτησης σύνδεσης, αλλά αρχίζει τη μετάδοση δεδομένων κατά το Initial State, θεωρώντας τη διαδικασία της σύνδεσης δεδομένη. Αν η αίτηση απορριφθεί, τότε η σύνδεση τερματίζεται μετά από ένα RTT. Ως αποτέλεσμα, το TP Planet πετυχαίνει καλύτερη αξιοποίηση των πόρων της σύνδεσης στην πρώιμη φάση της σύνδεσης, μέσω του αλγορίθμου Initial State.

Steady State

Ο αποστολέας εγκαταλείπει το Initial State για χρόνο $t = 2 \cdot \text{RTT}$ και παραμένει στο Steady State μέχρι το τερματισμό της σύνδεσης. Κατά τη διάρκεια του Steady State ο αποστολέας μπορεί να βρίσκεται σε μια από τις εξής τέσσερις καταστάσεις: Increase Rate, Decrease Rate, Hold Rate και Blackout, όπως φαίνεται στο διάγραμμα καταστάσεων του TP Planet.

Κατά την έναρξη του Steady State ο αποστολέας πηγαίνει στην κατάσταση Hold Rate, όπου δεν συμβαίνει καμία αλλαγή στο ρυθμό μετάδοσης. Κατά τη φάση του Steady State, το TP Planet εφαρμόζει ένα νέο σύστημα ελέγχου συμφόρησης, μέσω του οποίου αποφασίζονται οι μεταβάσεις ανάμεσα σε αυτές τις καταστάσεις στο Steady State. Έτσι, ο ρυθμός μετάδοσης δεδομένων μπορεί να αυξηθεί, να μειωθεί ή να παραμείνει σταθερός, ανάλογα με την ισχύουσα κατάσταση.

Έλεγχος συμφόρησης (Congestion Control)

Κατά τη διάρκεια του Steady State, ο αποστολέας του TP Planet μεταδίδει συνεχώς πακέτα NIX χαμηλής και υψηλής προτεραιότητας για λόγους ανίχνευσης συμφόρησης. Τα πακέτα NIX διαφέρουν από τα πακέτα NIL, που χρησιμοποιούνται στο Initial State, όσον αφορά το μέγεθος και τη λειτουργικότητα τους.

Τα πακέτα NIX χαμηλής και υψηλής προτεραιότητας μεταδίδονται ταυτόχρονα με ρυθμό ίσο με το ρυθμό μετάδοσης δεδομένων. Σκοπός είναι η λήψη «υποστήριξης» στις αποφάσεις που αφορούν τη συμφόρηση μέσω σύγκρισης των στατιστικών λήψης πακέτων NIX τόσο χαμηλής, όσο υψηλής προτεραιότητας. Αφού τα πακέτα έχουν το ίδιο μέγεθος και μεταδίδονται με τον ίδιο ρυθμό, υφίστανται τον ίδιο ρυθμό απώλειας πακέτων λόγω σφαλμάτων στις διαστημικές συνδέσεις. Όμως, υποθέτοντας ότι οι δορυφόροι αναμετάδοσης που λειτουργούν ως δρομολογητές στο InterPlanetary Backbone Network έχουν δυνατότητα διαχείρισης ουράς με προτεραιότητες, τα πακέτα NIX χαμηλής προτεραιότητας απορρίπτονται πρώτα σε περίπτωση συμφόρησης.

Ο παραλήπτης του TP Planet μετρά το πλήθος των πακέτων NIX χαμηλής (n_{LOW}) και υψηλής (n_{HIGH}) προτεραιότητας που λαμβάνει σε ένα κυλιόμενο παράθυρο χρόνου TW και υπολογίζει το λόγο:

$$\Phi = n_{LOW} / n_{HIGH}$$

Ολόκληρος ο έλεγχος συμφόρησης του TP Planet καθορίζεται σύμφωνα με τη τιμή του Φ . Οι μεταβάσεις ανάμεσα στις καταστάσεις Increase Rate, Decrease Rate και Hold Rate εκτελούνται βασιζόμενες στο Φ . Για την αποφυγή μεταβάσεων που δεν απαιτούνται, οι αποφάσεις λαμβάνονται μέσω σύγκρισης του Φ με ένα προκαθορισμένο κατώφλι μείωσης ρυθμού ϕ_d και ένα αύξησης ρυθμού ϕ_i .

Συνοψίζοντας, ο μηχανισμός ελέγχου συμφόρησης γίνεται ως εξής:

- $\Phi < \phi_d$: Στη περίπτωση αυτή ο αποστολέας του TP Planet συμπεράνει πως παρατηρείται συμφόρηση στη σύνδεση. Έτσι, ο αποστολέας μεταβαίνει στην κατάσταση Decrease Rate όπου ο ρυθμός μετάδοσης μειώνεται πολλαπλασιαστικά.
- $\phi_d \leq \Phi \leq \phi_i$: Στη περίπτωση αυτή, ο ρυθμός μετάδοσης δεδομένων διατηρείται σταθερός μέχρι τη λήψη της επόμενης ανατροφοδότησης από τον παραλήπτη.
- $\Phi > \phi_i$: Ο αποστολέας του TP Planet συμπεράνει πως δεν παρατηρείται συμφόρηση. Κατά συνέπεια, αυξάνει το ρυθμό μετάδοσης δεδομένων προσθετικά.

Το νέο rate-based σύστημα Προσθετικής Αύξησης / Πολλαπλασιαστικής Μείωσης

Όπως έχει αναφερθεί, τα ισχύοντα πρωτόκολλα TCP πετυχαίνουν πολύ χαμηλή απόδοση σε συνδέσεις με εξαιρετικά υψηλές καθυστερήσεις διάδοσης, κυρίως εξαιτίας της λειτουργίας τους που βασίζεται σε παράθυρα. Η απόδοση των πρωτοκόλλων TCP που βασίζονται σε παράθυρα και των σχεδίων που βασίζονται στο ρυθμό είναι αντιστρόφως ανάλογη του RTT και της τετραγωνικής ρίζας του RTT αντίστοιχα. Έτσι, τα συστήματα ελέγχου συμφόρησης που βασίζονται στο ρυθμό είναι περισσότερο εύρωστα σε περιβάλλοντα με υπερβολικές καθυστερήσεις διάδοσης από ότι οι μηχανισμοί που βασίζονται σε παράθυρα. Για το λόγο αυτό, το TP Planet χρησιμοποιεί έλεγχο συμφόρησης AIMD που βασίζεται στο ρυθμό.

Blackout State

Οι διακοπές συνδεσιμότητας εξαιτίας απώλειας της οπτικής γραμμής από τροχιακές αποκρύψεις οδηγεί σε εξάρσεις απώλειας πακέτων και, συνεπώς, μείωσης της απόδοσης. Για να παρέχει αξιόπιστη απόδοση, το TP Planet υιοθετεί την επιλογή SACK για να αντιμετωπίσει τις εξάρσεις απώλειας.

Εξαιτίας πιθανής ανεπάρκειας του πλήθους των μπλοκ SACK στο πεδίο επιλογής SACK για μεγάλες τιμές διάρκειας διακοπής συνδεσιμότητας, το TP Planet περιλαμβάνει μηχανισμό timeout. Για την αποφυγή μείωσης της απόδοσης, αναπτύχθηκε η κατάσταση Blackout State και ενσωματώθηκε στο Steady State.

Κατά τη διάρκεια διακοπής της συνδεσιμότητας, ο αποστολέας συνεχίζει αν στέλνει πακέτα NIX υψηλής και χαμηλής προτεραιότητας, χωρίς να αλλάξει το ρυθμό μετάδοσης. Ανάλογα με το χρόνο που διαρκεί η διακοπή λειτουργίας, το TP Planet χρησιμοποιεί διαφορετικούς τρόπους αντιμετώπισης του blackout.

5.2. SCPS-TP

Το πρωτόκολλο SCPS-TP [26], [38] αποτελείται από το τυπικό TCP προσαυξημένο με ένα πλήθος από προεκτάσεις και προσαυξήσεις, οι οποίες αποτελούνται από αλλαγές τόσο σε επίπεδο εφαρμογών, όσο σε επίπεδο προδιαγραφών.

Αντιμετώπιση των διαφόρων πηγών απώλειας

Το κλειδί για τη βελτιστοποίηση της απόδοσης του TCP είναι η ταυτοποίηση της πηγής απώλειας πακέτων και η κατάλληλη ανάλογη αντίδραση. Στο διαστημικό περιβάλλον υπάρχουν τουλάχιστον τρεις πηγές απωλειών: η συμφόρηση του δικτύου, η φθορά και οι διακοπές συνδεσιμότητας. Η κατάλληλη αντιμετώπιση για κάθε μια είναι διαφορετική και το πρωτόκολλο SCPS-TP χρησιμοποιεί ξεχωριστή αντιμετώπιση για την καθεμία. Το SCPS-TP έχει δύο μηχανισμούς για να καθορίσει την πηγή απώλειας πακέτων.

Όπως το τυπικό TCP, το SCPS-TP κάνει μια προκαθορισμένη υπόθεση όσον αφορά την πηγή της απώλειας πακέτων, αν δεν υπάρχουν συγκεκριμένες πληροφορίες. Η προκαθορισμένη υπόθεση του TCP είναι οι απώλειες προκαλούνται από συμφόρηση· παρόλα αυτά, η προκαθορισμένη υπόθεση του SCPS-TP είναι μια παράμετρος η οποία τίθεται από ένα διαχειριστή του δικτύου ή μια εφαρμογή. Υπό κάποιες συνθήκες, όταν το εύρος ζώνης του δικτύου διαχειρίζεται προσεκτικά, η συμφόρηση είναι απίθανη και είναι λογικό το SCPS-TP να υποθέσει πως κάθε απώλεια πακέτων οφείλεται σε σφάλματα.

Ο δεύτερος μηχανισμός για την ταυτοποίηση της πηγής απώλειας πακέτων είναι η συγκεκριμένη σηματοδότηση. Ο παραλήπτης ή κάποια άλλα στοιχεία του δικτύου (δρομολογητές ή σταθμοί στη Γη) μπορούν να στέλνουν συγκεκριμένα σήματα στον αποστολέα του TCP, όσον αφορά τη πηγή απώλειας πακέτων.

Απώλεια που προκαλείται από συμφόρηση

Όταν ενεργοποιείται ο έλεγχος συμφόρησης, το SCPS-TP χρησιμοποιεί τους αλγόριθμους ελέγχου συμφόρησης του TCP Vegas για να ελαχιστοποιήσει την απώλεια και να διευκολύνει τη χρήση μεγάλων παραθύρων. Για την επίτευξη βέλτιστης απόδοσης το TCP εξαρτάται από τη ρύθμιση του παραθύρου του παραλήπτη σε μια τιμή μεγαλύτερη από το Bandwidth X Delay Product του δικτύου, αλλά αρκετά μικρή για την αποφυγή συμφόρησης του δικτύου. Ο αλγόριθμος ελέγχου συμφόρησης του TCP Vegas περιορίζει τον εαυτό του, αυτορυθμίζεται και αποφεύγει τη συμφόρηση του δικτύου χωρίς να οδηγεί τη σύνδεση στο σημείο κορεσμού, ακόμη και όταν χειρίζεται μεγάλα παράθυρα.

Το SCPS-TP χρησιμοποιεί το διαφορετικό αλγόριθμο Slow Start του TCP Vegas, σύμφωνα με τον οποίο το παράθυρο συμφόρησης διπλασιάζεται για κάθε δεύτερο RTT, σε αντίθεση με κάθε RTT όπως συμβαίνει με το τυπικό TCP. Το SCPS-TP τροποποιεί τον αλγόριθμο αυτό παρέχοντας έναν επιπρόσθετο μηχανισμό που οδηγεί στη μετάβαση από τη φάση εκθετικής αύξησης του παραθύρου συμφόρησης στη φάση της αποφυγής συμφόρησης. Το SCPS-TP «μπαίνει» στη φάση αποφυγής συμφόρησης σύμφωνα με την προκαθορισμένη τεχνική του TCP Vegas για την εύρεση του σημείου στο οποίο η απόδοση βαθμιαία σταματάει να αυξάνεται. Επιπρόσθετα, το SCPS-TP «μπαίνει» στη φάση αποφυγής συμφόρησης μόλις το μέγεθος του παραθύρου συμφόρησης γίνει ίσο με το Bandwidth X Delay Product του δικτύου. Ο μηχανισμός αποφυγής συμφόρησης του TCP Vegas συνεχίζει να μετρά την απόδοση που επιτυγχάνεται και προσαρμόζει ανάλογα το μέγεθος του παραθύρου συμφόρησης κατά τη διάρκεια της σύνδεσης.

Ο έλεγχος συμφόρησης επικαλείται ως αντίδραση στην απώλεια πακέτων όταν η συμφόρηση θεωρείται η προκαθορισμένη πηγή απώλειας πακέτων και όταν ένα μήνυμα Explicit Congestion Notification (ECN) παραλαμβάνεται. Κάθε ενδιαμέσος δρομολογητής ή παραλήπτης μπορεί να στείλει ένα συγκεκριμένο σήμα για να ενημερώσει τον αποστολέα του SCPS-TP ότι συμβαίνει συμφόρηση.

Απώλεια που προκαλείται από φθορά

Στις διαστημικές συνδέσεις τα σφάλματα bits είναι η πρωταρχική αιτία απώλειας πακέτων και η συμφόρηση είναι απίθανη, με προσεκτικό προγραμματισμό της χρήσης της σύνδεσης. Στις περιπτώσεις αυτές, το SCPS-TP απενεργοποιεί την προκαθορισμένη υπόθεση ότι η συμφόρηση είναι η πηγή της απώλειας και δεν επικαλείται τον έλεγχο συμφόρησης ως αντίδραση. Για να αποφευχθεί η υπέρβαση της χωρητικότητας της σύνδεσης, το SCPS-TP χρησιμοποιεί ένα μηχανισμό ρύθμισης του ρυθμού ανοιχτού βρόχου με κουπόνι. Αναγνωρίζουμε τέσσερις φάσεις στη διαχείριση της φθοράς: εντοπισμός του σημείου έναρξης, σηματοδότηση, αντίδραση και εντοπισμός του σημείου τερματισμού.

Διακοπή συνδεσιμότητας

Στις διαστημικές συνδέσεις υποθέτουμε πως όταν μια σύνδεση βρίσκεται εκτός σύνδεσης προς τη μια κατεύθυνση, βρίσκεται επίσης εκτός σύνδεσης και προς την άλλη. Θεωρούμε τις ίδιες τέσσερις διαφορετικές προσεγγίσεις όσον αφορά τη διαχείριση των διακοπών συνδεσιμότητας, όπως με τη φθορά.

Ασυμμετρία καναλιών

Το TCP βασίζεται σε μια ροή επιβεβαιώσεων η οποία επιβεβαιώνει συνεχώς πακέτα στον αποστολέα. Οι συνδέσεις επικοινωνίας ανάμεσα σε ένα διαστημόπλοιο και τη Γη τυπικά διαμορφώνονται με ένα κανάλι υψηλού ρυθμού προς τη μια κατεύθυνση και ένα μικρότερου ρυθμού προς την αντίθετη κατεύθυνση · αυτές χρησιμοποιούνται τυπικά για telemetry και telecommand αντίστοιχα. Αν ένας δορυφόρος μεταδίδει δεδομένα αξιόπιστα πάνω από ένα κανάλι telemetry, ο όγκος της κίνησης των επιβεβαιώσεων που παράγει το TCP μπορεί εύκολα να ξεχειλίζει το αντίστροφο κανάλι που μεταφέρει τις επιβεβαιώσεις. Για την επίλυση του προβλήματος, το SCPS-TP απορρίπτει την απαίτηση το TCP να στέλνει επιβεβαιώσεις για κάθε δεύτερο πακέτο που λαμβάνεται. Το SCPS-TP επίσης «χαλαρώνει» την απαίτηση να επιβεβαιώνει αμέσως το TCP κάθε πακέτο που λαμβάνεται όταν υπάρχει μια ουρά εκτός σειράς. Αντίθετα, ο παραλήπτης SCPS-TP καθυστερεί τις επιβεβαιώσεις για μια ρυθμιζόμενη περίοδο χρόνου που σχετίζεται με την εκτίμηση του RTT.

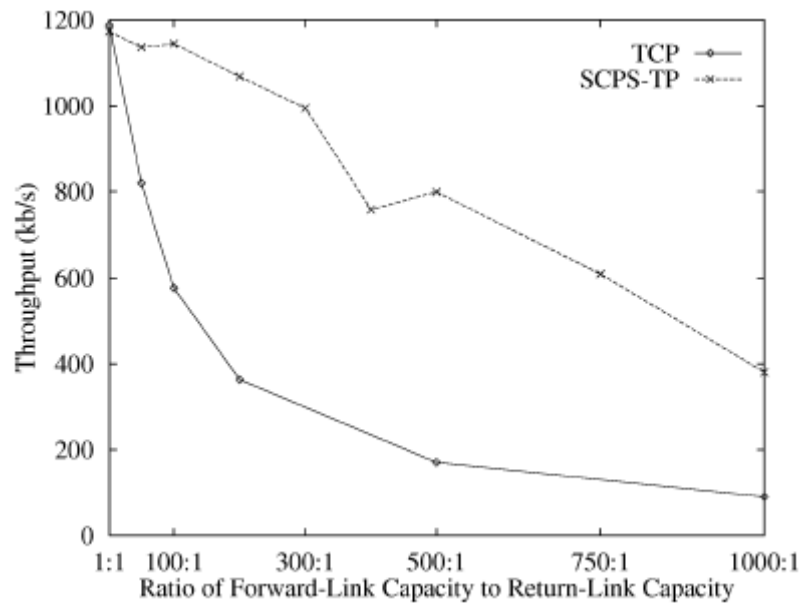
Σημειώνεται ακόμη πως ανάλογα με τη συχνότητα λήψης επιβεβαιώσεων, τη τιμή του χρονομετρητή επαναμετάδοσης και τα μεγέθη των παραθύρων που χρησιμοποιούνται, ο αλγόριθμος Fast Retransmit, με τον οποίο ο αποστολέας ανιχνεύει απώλεια μετρώντας πολλαπλές επιβεβαιώσεις, μπορεί να απενεργοποιηθεί.

Περιορισμένη χωρητικότητα

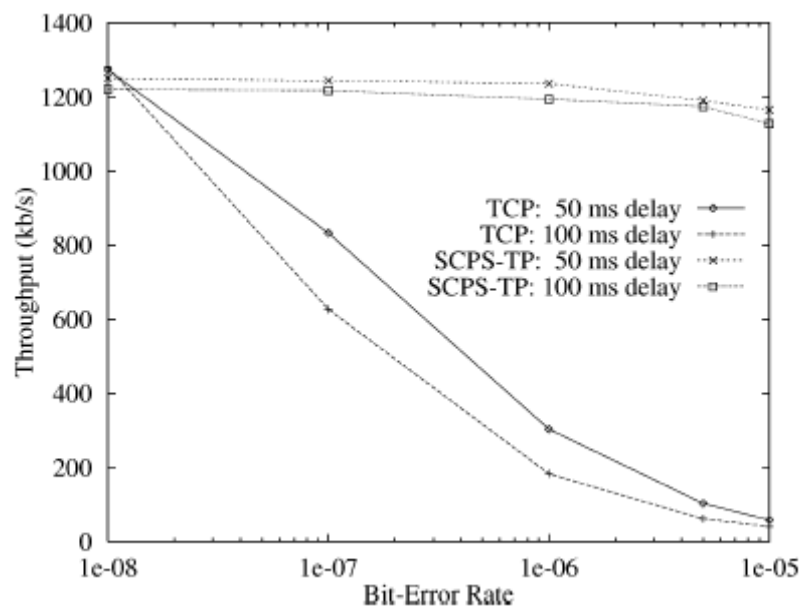
Το SCPS-TP χρησιμοποιεί δύο μηχανισμούς για τη βελτίωση της απόδοσης σε περιβάλλοντα όπου η χωρητικότητα είναι περιορισμένη:

- Την SCPS-TP συμπίεση της επικεφαλίδας και
- Την επιλογή SCPS-TP Selective Negative Acknowledgments (SNACK)

Παρακάτω δίνονται δύο συγκριτικές γραφικές παραστάσεις του πρωτοκόλλου μεταφοράς TCP και του πρωτοκόλλου SCPS-TP. Στην πρώτη συγκρίνονται τα δύο πρωτόκολλα για αυξανόμενη ασυμμετρία καναλιών, ενώ στη δεύτερη συγκρίνονται για διαφορετικούς ρυθμούς σφαλμάτων ανά bit για καθυστέρηση ίση με 50ms και 100ms.



Σχήμα 5.2 Ασυμμετρία καναλιών: SCPS-TP vs TCP

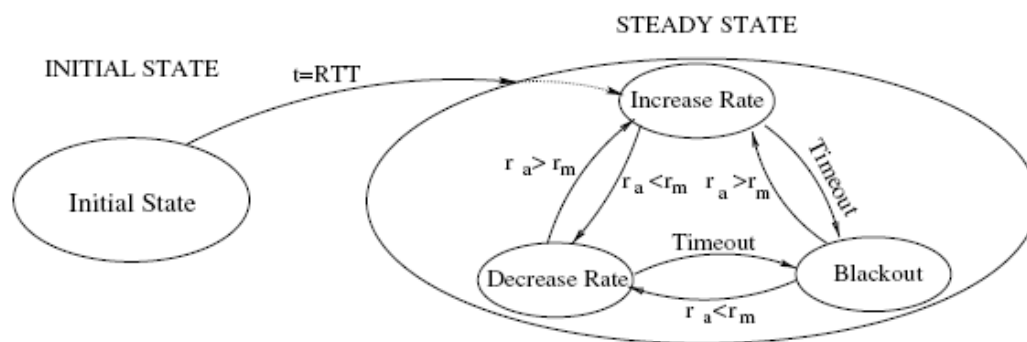


Σχήμα 5.3 Σφάλματα bit: SCPS-TP vs TCP

5.3. RCP Planet

Το RCP Planet [39], [40], [41] είναι ένα πρωτόκολλο ελέγχου του ρυθμού, το οποίο προτείνεται για την αντιμετώπιση των προκλήσεων που παρουσιάζουν τα πολυμεσικά πρωτόκολλα μεταφοράς στο InterPlanetary Backbone Network, όπου τα τελικά σημεία του αποστολέα και του παραλήπτη είναι δύο δορυφόροι αναμετάδοσης οι οποίοι κινούνται σε τροχιά γύρω από πλανήτες. Το RCP Planet «τρέχει» πάνω από το επίπεδο IP (Internet Protocol) και δεν απαιτεί καμία συγκεκριμένη τροποποίηση στα χαμηλότερα επίπεδα του συνόλου πρωτοκόλλων TCP/IP.

Το RCP Planet αποτελείται από δύο καταστάσεις, τη κατάσταση Initial State και τη κατάσταση Steady State, όπως φαίνεται στην παρακάτω εικόνα.



Σχήμα 5.4. Διάγραμμα καταστάσεων του RCP Planet

Οι βασικές λειτουργίες του RCP Planet συνοψίζονται παρακάτω.

FEC στο επίπεδο πακέτων

Για την ανάκαμψη από απώλειες πακέτων εξαιτίας συμφόρησης ή σφαλμάτων στις συνδέσεις του Διαπλανητικού Διαδικτύου, χρησιμοποιούνται οι κώδικες Tornado για FEC (Forward Error Correction) σε επίπεδο πακέτων λόγω της πολύ γρήγορης κωδικοποίησης και αποκωδικοποίησης τους. Αν και οι κώδικες Tornado απαιτούν ελαφρώς περισσότερα πακέτα κωδικοποίησης για την ανακατασκευή των αρχικών δεδομένων, το μειονέκτημα αυτό αντισταθμίζεται με τη χρήση μπλοκ FEC μεγάλου μεγέθους και, συνεπώς, μικρότερης επιβάρυνσης FEC. Ακόμη, οι κώδικες Tornado χρησιμοποιούν μόνο τελεστές αποκλειστικού-ή (XOR), οι οποίοι τους καθιστούν εύκολα εφαρμόσιμους στην πράξη. Το μήκος των μπλοκ FEC επιλέγεται κατάλληλα ώστε να ελαχιστοποιήσει την επιβάρυνση FEC για διαφορετικούς ρυθμούς απώλειας πακέτων.

Initial State

Στην κατάσταση Initial State είναι δύσκολο να καθοριστεί ο αρχικός ρυθμός αποστολής, αφού στην αρχή δεν είναι διαθέσιμες πληροφορίες σχετικά με τη

σύνδεση. Συντηρητικά θέτουμε τον αρχικό ρυθμό της εφαρμογής ίσο με τον ελάχιστο που απαιτείται, ώστε να μην εισάγουμε πάρα πολλά πακέτα στο δίκτυο. Ακόμη, ο ρυθμός απώλειας πακέτων στο Initial State δεν είναι γνωστός και, για το λόγο αυτό χρησιμοποιείται η πιο πρόσφατη τιμή r_h ως μια προσέγγιση του τρέχοντα ρυθμού απώλειας, ώστε να προσδιοριστεί το μήκος n των μπλοκ FEC. Παρόλα αυτά, ο ακριβής ρυθμός απώλειας πακέτων ίσως να μην είναι ίσος με τον r_h . Για να αντιμετωπιστεί η χειρότερη συνθήκη του δικτύου, συντηρητικά επιλέγουμε ένα ρυθμό απώλειας πακέτων r_1 πολύ μεγαλύτερο από το r_h και υπολογίζουμε το αντίστοιχο μήκος μπλοκ FEC n' . Το n' χρησιμοποιείται ως το ακριβές μήκος μπλοκ FEC για την κωδικοποίηση δεδομένων. Αφού τα $n-n'$ πλεονάζοντα πακέτα είναι ουσιαστικά επιπρόσθετος πλεονασμός που χρησιμοποιείται για την αντιμετώπιση μιας πιθανής χειρότερης κατάστασης του δικτύου, αποστέλλονται με χαμηλή προτεραιότητα. Τα πακέτα χαμηλής προτεραιότητας απορρίπτονται πρώτα κατά της συμφόρησης και, έτσι, δεν επηρεάζεται η κίνηση κανονικών δεδομένων κατά τη συμφόρηση. Τα εναπομείναντα πακέτα n μεταδίδονται με υψηλή προτεραιότητα.

Νέος μηχανισμός εξέτασης του ρυθμού

Η εξέταση του ρυθμού (probing) είναι ένας μηχανισμός για τη μέτρηση του παρατηρούμενου ρυθμού στη πλευρά του παραλήπτη για τον καθορισμό της διαθέσιμης χωρητικότητας. Η νέα μέθοδος εξέτασης του ρυθμού εκτελείται σε κάθε μπλοκ FEC, δηλαδή με κάθε μπλοκ FEC, ένα προκαθορισμένο πλήθος από πακέτα, το οποίο αποκαλείται ακολουθία probing, αποστέλλονται πρώτα με υψηλό ρυθμό, το λεγόμενο ρυθμό εξέτασης r_p . Τα εναπομείναντα πακέτα στο μπλοκ FEC αποστέλλονται χρησιμοποιώντας τον τρέχον ρυθμό αποστολής του αποστολέα r_s . Κάποια πακέτα μπορούν να απορριφθούν από την πύλη (gateway) εξαιτίας του περιορισμού της χωρητικότητας του δικτύου και, έτσι, ο παρατηρούμενος ρυθμός στον παραλήπτη r_o είναι η διαθέσιμη χωρητικότητα. Το μήκος της ακολουθίας probing είναι μια σχεδιαστική παράμετρος και επιλέγεται κατάλληλα. Μια άλλη σχεδιαστική παράμετρος είναι ο ρυθμός εξέτασης r_p . Στη κατάσταση Initial State δεν είναι διαθέσιμη καμία πληροφορία σχετικά με τη κατάσταση του δικτύου και, έτσι, το r_p τίθεται με τέτοιο τρόπο ώστε να μπορεί να φτάσει τη διαθέσιμη χωρητικότητα όσο γρηγορότερα γίνεται. Στην κατάσταση Initial State ο ρυθμός εξέτασης ενημερώνεται προσαρμοστικά σύμφωνα με την κατάσταση του δικτύου.

Νέα μέθοδος ελέγχου του ρυθμού

Για την ομαλή παράδοση των πολυμέσων, το RCP Planet εφαρμόζει μια νέα μέθοδο ελέγχου του ρυθμού, η οποία βασίζεται στο μηχανισμό εξέτασης του ρυθμού. Με την παραλαβή μιας επιβεβαίωσης από τον παραλήπτη, ο παρατηρούμενος ρυθμός r_o γίνεται γνωστός, γεγονός που αποκαλύπτει τη διαθέσιμη χωρητικότητα του δικτύου. Ο αντίστοιχος διαθέσιμος ρυθμός των μέσων (media rate) r_a υπολογίζεται από το r_o , το οποίο είναι το ανώτερο όριο του τρέχοντα ρυθμού. Αν $r_a \geq r_m$, όπου r_m είναι ο τρέχον ρυθμός των μέσων, η χωρητικότητα δεν αξιοποιείται πλήρως, δηλαδή ο ρυθμός r_m πρέπει να αυξηθεί. Το επιπλέον μέγεθος $r_a - r_m$ αυξάνεται γραμμικά σε ένα RTT σε σχέση με το χρόνο, ώστε ο ρυθμός των μέσων να αυξάνεται ομαλά για τη μείωση των πιθανοτήτων συμφόρησης του δικτύου. Από την άλλη μεριά, αν $r_a < r_m$, δηλαδή ο τρέχον ρυθμός είναι πολύ υψηλός, ο αποστολέας πρέπει να

«οπισθοχωρήσει» και να μειώσει το ρυθμό του, δηλαδή ο ρυθμός των μέσων μειώνεται πολλαπλασιαστικά.

Blackout State

Για τη μείωση του υποβιβασμού της απόδοσης εξαιτίας διακοπών συνδεσιμότητας, το RCP Planet ενσωματώνει τη κατάσταση Blackout State στη λειτουργία του πρωτοκόλλου. Ο αποστολέας συμπεραίνει τη διακοπή και σταματάει να αποστέλλει πακέτα δεδομένων αν δεν παραλάβει καμία επιβεβαίωση για ένα συγκεκριμένο χρονικό διάστημα. Με παρόμοιο τρόπο, ο παραλήπτης επίσης συμπεράνει τη διακοπή συνδεσιμότητας και αρχίζει να μεταδίδει τις λεγόμενες Μηδενικές Επιβεβαιώσεις (Zero ACKs). Αφού η καθυστέρηση είναι πολύ μεγάλη, η επίδραση της διακοπής στην απόδοση αλλάζει με τη σχετική θέση εμφάνισης της συμφόρησης σε σχέση με τον παραλήπτη. Οι Μηδενικές Επιβεβαιώσεις και οι Επιβεβαιώσεις εν Πτήση (on-fly ACKs) όταν έχουμε διακοπή συνδεσιμότητας χρησιμοποιούνται ώστε ο αποστολέας να αποκτήσει ακριβείς πληροφορίες σχετικά με τη κατάσταση της διακοπής και να λειτουργήσει αντίστοιχα.

FEC σε επίπεδο μπλοκ στις επιβεβαιώσεις

Για την αντιμετώπιση του προβλήματος της ασυμμετρίας των χωρητικότητων των καναλιών στις συνδέσεις του InterPlanetary Backbone, το RCP Planet υιοθετεί τον FEC σε επίπεδο μπλοκ στις επιβεβαιώσεις, δηλαδή μόνο μια επιβεβαίωση αποστέλλεται για ένα ολόκληρο μπλοκ FEC. Αν το μέγεθος του μπλοκ FEC είναι αρκετά μεγάλο, το πρόβλημα της ασυμμετρίας μπορεί να λυθεί από τον FEC σε επίπεδο μπλοκ στις επιβεβαιώσεις. Οι καθυστερημένες επιβεβαιώσεις μπορούν, επίσης, να χρησιμοποιηθούν για την περαιτέρω μείωση του πλήθους των επιβεβαιώσεων στο κανάλι επιστροφής, δηλαδή αποστέλλεται μόνο μια επιβεβαίωση για ένα συγκεκριμένο πλήθος μπλοκ FEC. Στη περίπτωση αυτή, ο παρατηρούμενος ρυθμός και ο τρέχον ρυθμός απώλειας πακέτων είναι οι μέσες τιμές των πολλαπλών μπλοκ FEC.

Αποτελέσματα προσομοιώσεων δείχνουν ότι το RCP Planet πετυχαίνει υψηλή απόδοση, δικαιοσύνη και είναι ανεκτικό στη καθυστέρηση, αντιμετωπίζοντας τις προκλήσεις του Διαπλανητικού Διαδικτύου.

5.4. Saratoga

Σκοπός του Saratoga [42], [43], [44] είναι να παρέχει μια μόνο ροή ανάμεσα σε δύο χρήστες, πολυπλέκοντας πολλαπλές ξεχωριστές συνεδρίες μέσω μιας αποκλειστικής σύνδεσης. Το Saratoga έχει σχεδιαστεί για να επιτυγχάνει υψηλές τιμές απόδοσης, μεταφέροντας όσο το δυνατόν περισσότερο φορτίο στη σύνδεση. Η αξιοποίηση της σύνδεσης είναι ένας συγκεκριμένος σχεδιαστικός στόχος · η διαχείριση της συμφόρησης του Διαδικτύου για ροές που ανταγωνίζονται όχι.

Το Saratoga είναι ένα UDP πρωτόκολλο μεταφοράς αρχείων το οποίο βασίζεται στο ρυθμό (rate-based) και είναι ικανό να μεταφέρει μεγάλα αρχεία. Το Saratoga χρησιμοποιείται ενεργά από το 2004 για τη μεταφορά δεδομένων εικόνων από αστερισμούς παρακολούθησης καταστροφών (DMC – Disaster Monitoring Constellation) στους σταθμούς στη Γη. Ως τώρα έχουν εκτοξευτεί πέντε δορυφόροι, τέσσερις από τους οποίους λειτουργούν σε χαμηλή τροχιά (low Earth orbit) και μεταφέρουν στη Γη αρχεία εικόνων για απόσταση δείγματος από το έδαφος (Ground Sample Distance) ίση με 32 μέτρα ή ακόμη καλύτερη.



Σχήμα 5.5. Παράδειγμα εικόνας DMC – Νέα Ορλεάνη Η.Π.Α. μετά τον καταστροφικό τυφώνα Κατρίνα. Το πράσινο χρώμα απεικονίζει τη βλάστηση, ενώ το κόκκινο τις πλημμυρισμένες περιοχές.

Το Saratoga αρχικά σχεδιάστηκε για να μεταφέρει αρχεία μεγέθους μόνο μέχρι 4 GB, αλλά εξελίχθηκε ώστε να συμβαδίζει με τις ανάγκες για μεγαλύτερη ανάλυση και, φυσικά, ακόμη μεγαλύτερων αρχείων εικόνων. Τρεις ακόμη δορυφόροι παρακολούθησης καταστροφών DMC βρίσκονται τώρα υπό κατασκευή.

Αρχικά το Saratoga σχεδιάστηκε για να μεταφέρει αρχεία από τους δορυφόρους στους σταθμούς-βάσεις μέσα σε ένα χρονικό περιθώριο 12 λεπτών ή μικρότερο. Αυτός ο περιορισμένος χρόνος επαφής οδήγησε στην ανάγκη για όσο το δυνατόν περισσότερα δεδομένα χωρίς σφάλματα. Το Saratoga βασίζεται στον έλεγχο αθροίσματος (checksum) UDP για βασικό έλεγχο σφαλμάτων και μπορεί να εφαρμόσει έλεγχο αθροίσματος από άκρο σε άκρο. Αν και το Saratoga αρχικά αναπτύχθηκε για μεταφορά αρχείων, έχει πλέον αποδειχθεί πως είναι χρήσιμο και για ανταλλαγές πακέτων (bundle) DTN.

Χαρακτηριστικά του Saratoga

Παρακάτω δίνονται κάποια χαρακτηριστικά του Saratoga:

- Ενώ το TCP είναι ακατάλληλο για ασύμμετρες συνδέσεις όπου η ασυμμετρία ξεπερνάει το 50:1, το Saratoga με τη χρήση Αρνητικών Επιβεβαιώσεων (Negative Acknowledgments) είναι κατάλληλο για ασυμμετρία συνδέσεων μέχρι 170:1.
- Απαιτείται κάποιος μηχανισμός ανάκαμψης από απώλειες πακέτων εξαιτίας σφαλμάτων στο κανάλι. Οι απώλειες αυτές, όμως, δεν αναγκάζουν τον αποστολέα να μειώσει το ρυθμό αποστολής.
- Είναι σημαντική η υψηλή αξιοποίηση της σύνδεσης, έτσι ώστε να επιτευχθεί μέγιστη χρησιμότητα των περιορισμένων και σποραδικά διαθέσιμων χρονικών περιόδων συνδεσιμότητας.
- Δεν χρειάζεται σχεδόν καμία ρύθμιση του ελέγχου συμφόρησης, αφού οι κόμβοι χρησιμοποιούν αποκλειστικά συνδέσεις χωρίς να ανταγωνίζονται άλλους κόμβους ή ακόμη και τους εαυτούς τους, αφού πολλαπλές συνεδρίες πολυπλέκονται σε μια ροή.
- Στο μοντέλο αυτό, μεμονωμένα payloads χρησιμοποιούν εναλλασσόμενα τη σύνδεση για να παραδώσουν τις πληροφορίες που έχουν, πριν τεθούν εκτός λειτουργίας για να εξοικονομήσουν ενέργεια.
- Οι ουρές εισόδου και εξόδου δεν είναι πλέον τόσο σημαντικές αφού ο μηχανισμός επαναμετάδοσης του Saratoga μπορεί να χειρίζεται απευθείας ένα αρχείο που διατηρείται στην κύρια μνήμη και, έτσι, επιτρέπει μικρότερες ουρές στις συνδέσεις και γρηγορότερη ανάκαμψη.
- Ο Forward Error Correction (FEC) χρησιμοποιείται για την επίτευξη μιας αξιόπιστης μετάδοσης μέσω ενός καναλιού και αφορά κυρίως το φυσικό επίπεδο. Η στρατηγική των αρνητικών επιβεβαιώσεων (NAK – Negative Acknowledgments) απλών πακέτων που βασίζεται στο ARQ (Automatic Repeat-Request) χρησιμοποιείται από το Saratoga για την αντιμετώπιση λαθών που δεν χειρίζονται από το FEC.

Βιβλιογραφία

- [1] “Διαδικτυακά πρωτόκολλα” Βασίλης Τσαουσίδης. Κλειδάριθμος, 2004.
- [2] “Receiver-oriented and measurement-based Transmission Control Protocol in Heterogeneous (Wired/Wireless) Networks” Chi Zhang, Ph.D. Thesis in Computer Science of Northeastern University, 2003.
- [3] “On the efficiency and fairness of AIMD-based Congestion Avoidance Algorithms” Adrian Lahanas, Ph. D. Thesis in Computer Science of Northeastern University, 2003.
- [4] “Computer Network Programming – TCP Overview”, Dr. Sam Hsu, Computer and Science Engineering, Florida Atlantic University.
- [5] “Transmission Control Protocol”, J. Postel, RFC 793, September 1981.
- [6] “TCP/IP Illustrated, Volume 1: The Protocols”, W. Stevens, Addison-Wesley, 1994.
- [7] “Analysis of the Increase and Decrease Algorithms for Congestion Avoidance in Computer Networks”, D.-M. Chiu and R.Jain, Computer Networks and ISDN Systems, 17(1):1-14, June 1989.
- [8] “TCP Slow Start, Congestion Avoidance, Fast Retransmit and Fast Recovery Algorithms”, W. Stevens, RFC 2001, January 1997.
- [9] “The New Reno Modification to TCP’s Fast Recovery Algorithm”, S.Floyd and M. Handley, RFC 2582, April 1999.
- [10] “Simulation Based Comparisons of Tahoe, Reno and SACK TCP”, K. Fall and S. Floyd, Computer Communication Review, 26 (3) : 5-21, July 1996.
- [11] “Congestion Avoidance and Control”, V. Jacobson, In Proceedings of ACM SIGCOMM 88, August 1998.
- [12] “TCP Congestion Control”, M. Allman, V. Paxson and W. Stevens, RFC 2581, April 1999.
- [13] “TCP Selective Acknowledgment Options”, M. Mathis, J. Mahdavi, S. Floyd and A. Romanow, RFC 2018, April 1996.
- [14] “TCP Friendly Rate Control (TFRC): Protocol Specification”, M. Handley, J. Padhye, S. Floyd and J. Widmer, Internet draft, May 2001.
- [15] “TCP Vegas: End to End Congestion Avoidance on a Global Internet”, L.S.Brakmo and L.L. Peterson, IEEE Journal on Selected Areas in Communications, 13 (8): 1465-1480, October 1995.

- [16] “The interrelation of TCP Responsiveness and Smoothness in heterogeneous networks”, Zhang C. and V. Tsaoussidis, The 7th IEEE Symposium on Computers and Communications, ISCC 2002, IEEE Computer Society Press, Taormina, Italy, July 2002.
- [17] “TCP Probing: Towards an Error Control Schema with Energy and Throughput Performance Gains”, Tsaoussidis V., H. Badr and R. Verma, In Proceedings of the 7th International Conference on Network Protocols, October 1999.
- [18] “TP Planet: A Reliable Transport Protocol for InterPlanetary Internet”, Ozgur B. Akan, Jian Fang and Ian F. Akyildiz, 2003.
- [19] InterPlanetary Internet, <http://www.ece.gatech.edu/research/labs/bwn/deepspace/>
- [20] “Delay Tolerant Networking: An Approach to InterPlanetary Internet”, Scott Burleigh, Adrian Hooke, Leigh Torgerson, Kevin Fall, Vint Cerf, Bob Durst, Keith Scott and Howard Weiss, IEEE Communication Magazine, June 2003.
- [21] “Performance Analysis Of Snack In Satellite Networks Through Simulation”, Divya Anantharaman, 2004.
- [22] “High-performance bulk data transfers with TCP”, Matei Ripeanu.
- [23] Space Telecommand, http://www.atis.org/tg2k/_space_telecommand.html
- [24] Space Telemetry, http://www.atis.org/tg2k/_space_telemetry.html
- [25] “Performance of TCP Protocols in Deep Space Communication Networks”, Ozgur B. Akan, Jian Fang and Ian F. Akyildiz, IEEE Communication Letters, Vol.6, No.11, November 2002.
- [26] “TCP Extensions for Space Communications”, Robert C. Burst, Gregory J. Miller and Eric J. Travis, Wireless Networks 3 p. 389-403, J.C. Baltzer AG, Science Publishers, 1997.
- [27] “FEC (Forward Error Correction) Performance Evaluation Method and Poisson Error Generator”, Kuroda Masahiro and Furuya Takashi.
- [28] Forward Error Correction, http://en.wikipedia.org/wiki/Forward_error_correction
- [29] “Compact FEC Schemes”, <http://www.faqs.org/rfcs/rfc3695.html>
- [30] “Extending Internet into Space”
- [31] “Performance of Multimedia Rate Control Protocols in InterPlanetary Internet”, Jian Fang and Ozgur B. Akan, IEEE Communication Letters, 2003.

- [32] “When TCP Breaks: Delay-and Disruption-Tolerant Networking”, Stephen Farrell, Vinny Cahill, Dermot Geraghty, Ivor Humphreys and Paul McDonald, IEEE Computer Society, July-August 2006.
- [33] “The ns manual”, Kevin Fall and Kannan Varadhan, June 2007.
- [34] NS-2 Network Simulator, <http://www.isi.edu/nsnam/ns/>
- [35] “Wireless Networking and Systems”, CSE 590, NS Tutorial.
- [36] “Performance Analysis of SNACK in Satellite Networks Through Simulation”, Divya Anantharaman, December 2004.
- [37] SNACK Extension, http://www.geocities.com/divya_anantharaman/snack.html
- [38] “Report On ESA Network Infrastructure Protocol Standards and Services: Limitations and Capabilities for Internetworking”
- [39] “InterPlaNetary Internet: state-of-the-art and research challenges”, Ian F. Akyildiz, Ozgur B. Akan , Chao Chen, Jian Fang and Weilian Su, Computer Networks 43 (2003) 75–112.
- [40] “RCP-Planet: a rate control protocol for InterPlaNetary Internet”, Jian Fang and Ian F. Akyildiz, International Journal of Satellite Telecommunications and Networking, Vol. 25, Issue 2, P. 167-194.
- [41] “An Experimental Analysis of Rate-based and Window-based Transmission Mechanisms over Simulated Space-Internet Links”, Ru H. Wang, Stephen Horan, Bhanu Gutha, Bo Sun and Pranay K. R. Reguri, IEEE Globecom, 2005.
- [42] “Saratoga: a Delay-Tolerant Networking convergence layer with efficient link utilization”, Lloyd Wood, Wesley M. Eddy, Will Ivancic, Jim McKim and Chris Jackson, Delay Tolerant Networking session, Third International Workshop on Satellite and Space Communications, September 2007.
- [43] “Introducing Saratoga fast file transfer from space”, Lloyd Wood, Delay Tolerant Networking Workshop, May 2007.
- [44] “Saratoga: a bundle convergence layer”, Lloyd Wood, Delay Tolerant Networking session, July 2007.

ΠΑΡΑΡΤΗΜΑ

Παράρτημα

Στο παράρτημα περιλαμβάνονται τα ακριβή αποτελέσματα από τα πειράματα που πραγματοποιήθηκαν για να μελετηθεί η απόδοση του TCP σε διαστημικό περιβάλλον. Συγκεκριμένα, η σειρά με την οποία παρουσιάζονται τα αποτελέσματα των πειραμάτων είναι η εξής:

- Επίδραση της καθυστέρησης
- Ρυθμός σφαλμάτων ανά πακέτο
- Χωρητικότητα της κύριας σύνδεσης
- Ασύμμετρα κανάλια
- Σφάλματα στο forward link και στο return link
- Διακοπή συνδεσιμότητας
- Bandwidth X Delay Product
- Συνδυασμός προβλημάτων (PER, Asymmetry, Blackout)

Επίδραση της καθυστέρησης

Application	Bulk Data Transfer
TCP	SACK
Queue	DropTail
Metric	Alltraced
Error	No error
Blackout	No blackout
Bandwidth Backbone	1 Mbps

DBB (s)	BWxDelay	RTT (s)	TT (s)	Throughput (Bps)	Goodput (Bps)	Ret.packets
0,01	1250	0,02	2	184100	177000	0
0,05	6250	0,1	10	132292	127200	0
0,1	12500	0,2	20	125062	120250	0
0,2	25000	0,4	40	120823	116175	0
0,3	37500	0,6	60	119150	114566	0
0,4	50000	0,8	80	118105,5	113562	0
0,45	56250	0,9	90	117763,1	113233	0
0,5	62500	1	100	116324,39	111850	0
0,55	68750	1,1	110	105862,9	101790	0
1	125000	2	200	58687,4	56430	0
3	375000	6	600	8075,73	7763	2
5	625000	10	1000	4844,439	4658	2
10	1250000	20	2000	2422,739	2329	3
30	3750000	60	6000	807,593	776	5
60	7500000	120	12000	403,8	388	6
100	12500000	200	20000	20,449	9	202
200	25000000	400	40000	12,8	4	302
300	37500000	600	60000	10,25	3	402
450	56250000	900	90000	7,977	2	502
600	75000000	1200	120000	7,7	1	702
1000	125000000	2000	200000	6,679	0	1102
2000	250000000	4000	400000	5,915	0	2102
3000	375000000	6000	600000	5,8316	0	3202
4000	500000000	8000	800000	5,6612	0	4202
5000	625000000	10000	1000000	5,6619	0	5302
6000	750000000	12000	1200000	5,5766	0	6302

Application	Bulk Data Transfer
TCP	New Reno SNACK
Queue	DropTail
Metric	Alltraced
Error	No error
Blackout	No blackout
Bandwidth Backbone	1 Mbps
Buffer Size	120

DBB (s)	RTT (s)	TT (s)	Throughput (Bps)	Goodput (Bps)	Ret.packets
0,01	0,02	2	184100	177000	0
0,05	0,1	10	132292	127200	0
0,1	0,2	20	125062	120250	0
0,2	0,4	40	120823	116175	0
0,3	0,6	60	119150	114566	0
0,4	0,8	80	118105,5	113562	0
0,45	0,9	90	117763,1	113233	0
0,5	1	100	116324,39	111850	0
0,55	1,1	110	105862,9	101790	0
1	2	200	58687,4	56430	0
3	6	600	8181,46	7863	3
5	10	1000	4926,6	4737	2
10	20	2000	2487,74	2390	5
30	60	6000	824,93	792	7
60	120	12000	423,3	406	11
100	200	20000	250,66	240	9
200	400	40000	24,42	9	560
300	600	60000	21,2	6	820
450	900	90000	18,03	4	1133
600	1200	120000	18,58	3	1703
1000	2000	200000	17,38	2	2887
2000	4000	400000	16,42	1	5860
3000	6000	600000	16,58	0	9113
4000	8000	800000	16,3	0	12086
5000	10000	1000000	16,32	0	15253
6000	12000	1200000	16,25	0	18323

Ρυθμός σφαλμάτων ανά πακέτο

Application	Bulk Data Tranfer
TCP	SACK
Queue	DropTail
Metric	Alltraced
Blackout	No blackout

BWBB	1Mbps			
DBB	100ms			
RTT	200ms			
TT	20s			
Error	Throughput (Bps)	Goodput (Bps)	Overhead (%)	Ret.Packets
0,000001	125062	120250	3,85	0
0,00001	125062	120250	3,85	0
0,0001	125062	120250	3,85	0
0,001	117106	112500	3,93	2
0,005	82162	78500	4,46	10
0,01	49454	46800	5,37	15
0,05	15448	14000	9,37	18
0,1	6818	5700	16,40	19

BWBB	1Mbps			
DBB	1s			
RTT	2s			
TT	200s			
Error	Throughput (Bps)	Goodput (Bps)	Overhead (%)	Ret.Packets
0,000001	58687,4	56430	3,85	0
0,00001	58687,4	56430	3,85	0
0,0001	58687,4	56430	3,85	0
0,001	19874,6	19080	4,00	6
0,005	11065,8	10595	4,25	9
0,01	6328,6	6010	5,03	15
0,05	1960,8	1785	8,97	21
0,1	1264,2	1045	17,34	36

BWBB	1Mbps			
DBB	10s			
RTT	20s			
TT	2000s			
Error	Throughput (Bps)	Goodput (Bps)	Overhead (%)	Ret.Packets
0,000001	2422,739	2329	3,87	3
0,000001	2422,739	2329	3,87	3
0,0001	2422,739	2329	3,87	3
0,001	1497,66	1438	3,98	6
0,005	798,259	763	4,42	10
0,01	497,69	472	5,16	15
0,05	199,759	182	8,89	23
0,1	134,24	111	17,31	38

BWBB	1Mbps			
DBB	60s			
RTT	120s			
TT	12000s			
Error	Throughput (Bps)	Goodput (Bps)	Overhead (%)	Ret.Packets
0,000001	403,8	388	3,91	6
0,000001	403,8	388	3,91	6
0,0001	403,8	388	3,91	6
0,001	249,36	239	4,15	9
0,005	132,96	127	4,48	13
0,01	82,7	78	5,68	18
0,05	32,69	29	11,29	26
0,1	22,2	18	18,92	41

Application	Bulk Data Transfer
TCP	New Reno SNACK
Queue	DropTail
Metric	Alltraced
Blackout	No blackout

BWBB	1Mbps			
DBB	100ms			
RTT	200ms			
TT	20s			
Error	Throughput (Bps)	Goodput (Bps)	Overhead (%)	Ret.Packets
0,000001	125062	120250	3,85	0
0,000001	125062	120250	3,85	0
0,0001	125062	120250	3,85	0
0,001	109254	98550	9,80	130
0,005	85646	74600	12,90	155
0,01	68538	58600	14,50	146
0,05	17164	14800	13,77	35
0,1	5674	4350	23,33	24

BWBB	1Mbps			
DBB	1s			
RTT	2s			
TT	200s			
Error	Throughput (Bps)	Goodput (Bps)	Overhead (%)	Ret.Packets
0,000001	58687,4	56430	3,85	0
0,00001	58687,4	56430	3,85	0
0,0001	58687,4	56430	3,85	0
0,001	16208,6	15075	6,99	102
0,005	9157,39	8360	8,71	89
0,01	8252,6	7045	14,63	178
0,05	2127,19	1830	13,97	44
0,1	1264,2	995	21,29	46

BWBB	1Mbps			
DBB	10s			
RTT	20s			
TT	2000s			
Error	Throughput (Bps)	Goodput (Bps)	Overhead (%)	Ret.Packets
0,000001	2487,739	2390	3,93	5
0,00001	2487,739	2390	3,93	5
0,0001	2487,739	2390	3,93	5
0,001	979,22	924	5,64	37
0,005	525,77	483	8,13	46
0,01	460,259	420	8,75	47
0,05	246,56	202	18,07	72
0,1	246,56	193	21,72	91

BWBB	1Mbps			
DBB	60s			
RTT	120s			
TT	12000s			
Error	Throughput (Bps)	Goodput (Bps)	Overhead (%)	Ret.Packets
0,000001	423,3	406	4,09	11
0,00001	423,3	406	4,09	11
0,0001	423,3	406	4,09	11
0,001	174,74	165	5,57	37
0,005	138,59	130	6,20	43
0,01	77,58	69	11,06	69
0,05	39,71	33	16,90	66
0,1	24,026	18	25,08	56

BWBB	1Mbps			
DBB	300s			
RTT	600s			
TT	60000s			
Error	Throughput (Bps)	Goodput (Bps)	Overhead (%)	Ret.Packets
0,000001	21,2046	6	71,70	820
0,00001	21,2046	6	71,70	820
0,0001	21,2046	6	71,70	820
0,001	21,2046	6	71,70	820
0,005	21,274	6	71,80	827
0,01	20,649	6	70,94	801
0,05	18,327	5	72,72	739
0,1	16,923	4	76,36	686

BWBB	1Mbps			
DBB	600s			
RTT	1200s			
TT	120000s			
Error	Throughput (Bps)	Goodput (Bps)	Overhead (%)	Ret.Packets
0,000001	18,576	3	83,85	1703
0,00001	18,576	3	83,85	1703
0,0001	18,576	3	83,85	1703
0,001	18,594	3	83,87	1705
0,005	18,481	3	83,77	1694
0,01	18,403	3	83,70	1698
0,05	16,808	3	82,15	1560
0,1	15,647	2	87,22	1476

Χωρητικότητα της κύριας σύνδεσης

DBB	100ms
RTT	200ms
TT	20s
Application	Bulk Data Tranfer
TCP	SACK
Queue	DropTail
Metric	alltraced
Error	No error
Blackout	No blackout

BWBB (Mbps)	Throughput (Bps)	Goodput (Bps)
0,1	18410	17700
0,2	30422	29250
0,4	54290	52200
0,5	66146	63600
0,7	89806	86350
0,9	113310	108950
1	125062	120250
5	581622	559250
10	586874	564300
20	586874	564300
30	586874	564300
40	586874	564300
50	586874	564300

DBB	1s
RTT	2s
TT	200s
Application	Bulk Data Tranfer
TCP	SACK
Queue	DropTail
Metric	alltraced
Error	No error
Blackout	No blackout

BWBB (Mbps)	Throughput (Bps)	Goodput (Bps)
0,1	12506,2	12025
0,2	24164,59	23235
0,4	47242,19	45425
0,5	58162,19	55925
0,7	58411,8	56165
0,9	58661,4	56405
1	58687,4	56430
5	58687,4	56430
10	58687,4	56430
20	58687,4	56430
30	58687,4	56430
40	58687,4	56430
50	58687,4	56430

DBB	10s
RTT	20s
TT	2000s
Application	Bulk Data Tranfer
TCP	SACK
Queue	DropTail
Metric	alltraced
Error	No error
Blackout	No blackout

BWBB (Mbps)	Throughput (Bps)	Goodput (Bps)
0,1	2422,739	2329
0,2	2422,739	2329
0,4	2422,739	2329
0,5	2422,739	2329
0,7	2422,739	2329
0,9	2422,739	2329
1	2422,739	2329
5	2422,739	2329
10	2422,739	2329
20	2422,739	2329
30	2422,739	2329
40	2422,739	2329
50	2422,739	2329

DBB	60s
RTT	120s
TT	12000s
Application	Bulk Data Tranfer
TCP	SACK
Queue	DropTail
Metric	alltraced
Error	No error
Blackout	No blackout

BWBB (Mbps)	Throughput (Bps)	Goodput (Bps)
0,1	403,8	388
0,2	403,8	388
0,4	403,8	388
0,5	403,8	388
0,7	403,8	388
0,9	403,8	388
1	403,8	388
5	403,8	388
10	403,8	388
20	403,8	388
30	403,8	388
40	403,8	388
50	403,8	388

DBB	300s
RTT	600s
TT	60000s
Application	Bulk Data Tranfer
TCP	SACK
Queue	DropTail
Metric	alltraced
Error	No error
Blackout	No blackout

BWBB (Mbps)	Throughput (Bps)	Goodput (Bps)
0,1	10,25	3
0,2	10,25	3
0,4	10,25	3
0,5	10,25	3
0,7	10,25	3
0,9	10,25	3
1	10,25	3
5	10,25	3
10	10,25	3
20	10,25	3
30	10,25	3
40	10,25	3
50	10,25	3

DBB	600s
RTT	1200s
TT	120000s
Application	Bulk Data Tranfer
TCP	SACK
Queue	DropTail
Metric	alltraced
Error	No error
Blackout	No blackout

BWBB (Mbps)	Throughput (Bps)	Goodput (Bps)
0,1	7,7	1
0,2	7,7	1
0,4	7,7	1
0,5	7,7	1
0,7	7,7	1
0,9	7,7	1
1	7,7	1
5	7,7	1
10	7,7	1
20	7,7	1
30	7,7	1
40	7,7	1
50	7,7	1

DBB	100ms
RTT	200ms
TT	20s
Application	Bulk Data Tranfer
TCP	New Reno SNACK
Queue	DropTail
Metric	Alltraced
Error	No error
Blackout	No blackout

BWBB (Mbps)	Throughput (Bps)	Goodput (Bps)
0,1	18410	17700
0,2	30422	29250
0,4	54290	52200
0,5	66146	63600
0,7	89806	86350
0,9	113310	108950
1	125062	120250
5	581622	559250
10	586874	564300
20	586874	564300
30	586874	564300
40	586874	564300
50	586874	564300

DBB	1s
RTT	2s
TT	200s
Application	Bulk Data Tranfer
TCP	New Reno SNACK
Queue	DropTail
Metric	Alltraced
Error	No error
Blackout	No blackout

BWBB (Mbps)	Throughput (Bps)	Goodput (Bps)
0,1	12506,2	12025
0,2	24164,59	23235
0,4	47242,19	45425
0,5	58162,19	55925
0,7	58411,8	56165
0,9	58661,4	56405
1	58687,4	56430
5	58687,4	56430
10	58687,4	56430
20	58687,4	56430
30	58687,4	56430
40	58687,4	56430
50	58687,4	56430

DBB	10s
RTT	20s
TT	2000s
Application	Bulk Data Tranfer
TCP	New Reno SNACK
Queue	DropTail
Metric	Alltraced
Error	No error
Blackout	No blackout

BWBB (Mbps)	Throughput (Bps)	Goodput (Bps)
0,1	2471,09	2374
0,2	2487,739	2390
0,4	2487,739	2390
0,5	2487,739	2390
0,7	2487,739	2390
0,9	2487,739	2390
1	2487,739	2390
5	2487,739	2390
10	2487,739	2390
20	2487,739	2390
30	2487,739	2390
40	2487,739	2390
50	2487,739	2390

DBB	60s
RTT	120s
TT	12000s
Application	Bulk Data Tranfer
TCP	New Reno SNACK
Queue	DropTail
Metric	Alltraced
Error	No error
Blackout	No blackout

BWBB (Mbps)	Throughput (Bps)	Goodput (Bps)
0,1	420,96	404
0,2	420,96	404
0,4	423,3	406
0,5	423,3	406
0,7	423,3	406
0,9	423,3	406
1	423,3	406
5	423,3	406
10	423,3	406
20	423,3	406
30	423,3	406
40	423,3	406
50	423,3	406

Ασύμμετρα κανάλια

DBB	50ms
RTT	100ms
TT	10s
Application	Bulk Data Transfer
TCP	SACK
Queue	DropTail
Metric	Alltraced
Error	No error
Blackout	No blackout

Forward-Link Capacity (Kbps)	Return-Link Capacity (Kbps)	Ratio	Throughput (Bps)	Goodput (Bps)
1000	1000	1	132292	127200
1000	500	2	132292	127200
1000	100	10	132084	127000
1000	20	50	74988	72100
1000	10	100	43892	42200
1000	5	200	28188	27100
1000	2,5	400	15812	15200
1000	2	500	12692	12200
1000	1,5	666,66667	9364	9000
1000	1,25	800	7908	7600
1000	1	1000	6244	6000

DBB	500ms
RTT	1s
TT	100s
Application	Bulk Data Transfer
TCP	SACK
Queue	DropTail
Metric	Alltraced
Error	No error
Blackout	No blackout

Forward-Link Capacity (Kbps)	Return-Link Capacity (Kbps)			
		Ratio	Throughput (Bps)	Goodput (Bps)
1000	1000	1	116324,39	111850
1000	500	2	116282,8	111810
1000	100	10	115981,2	111520
1000	20	50	62244,4	59850
1000	10	100	32032,4	30800
1000	5	200	16775,59	16130
1000	2,5	400	9058,79	8710
1000	2	500	7509,19	7220
1000	1,5	666,66667	5949,19	5720
1000	1,25	800	5169,19	4970
1000	1	1000	4389,19	4220

DBB	1s
RTT	2s
TT	200s
Application	Bulk Data Transfer
TCP	SACK
Queue	DropTail
Metric	Alltraced
Error	No error
Blackout	No blackout

Forward-Link Capacity (Kbps)	Return-Link Capacity (Kbps)			
		Ratio	Throughput (Bps)	Goodput (Bps)
1000	1000	1	58687,4	56430
1000	500	2	58687,4	56430
1000	100	10	58609,4	56355
1000	20	50	57969,8	55740
1000	10	100	31127,4	29930
1000	5	200	16021,4	15405
1000	2,5	400	8387,79	8065
1000	2	500	6848,6	6585
1000	1,5	666,66667	5304,19	5100
1000	1,25	800	4534,6	4360
1000	1	1000	3754,59	3610

DBB	10s
RTT	20s
TT	2000s
Application	Bulk Data Transfer
TCP	SACK
Queue	DropTail
Metric	Alltraced
Error	No error
Blackout	No blackout

Forward-Link Capacity (Kbps)	Return-Link Capacity (Kbps)	Ratio	Throughput (Bps)	Goodput (Bps)
1000	1000	1	2422,739	2329
1000	500	2	2422,739	2329
1000	100	10	2422,739	2329
1000	20	50	2422,739	2329
1000	10	100	2422,739	2329
1000	5	200	2422,739	2329
1000	2,5	400	2400,38	2307
1000	2	500	2384,26	2292
1000	1,5	666,66667	2369,69	2278
1000	1,25	800	2312,5	2223
1000	1	1000	2194,46	2109

DBB	50ms
RTT	100ms
TT	10s
Application	Bulk Data Transfer
TCP	New Reno SNACK
Queue	DropTail
Metric	Alltraced
Error	No error
Blackout	No blackout

Forward-Link Capacity (Kbps)	Return-Link Capacity (Kbps)	Ratio	Throughput (Bps)	Goodput (Bps)
1000	1000	1	132292	127200
1000	500	2	132292	127200
1000	100	10	132084	127000
1000	20	50	74988	72100
1000	10	100	43892	42200
1000	5	200	28188	27100
1000	2,5	400	15812	15200
1000	2	500	12692	12200
1000	1,5	666,66667	9364	9000
1000	1,25	800	7908	7600
1000	1	1000	6244	6000

DBB	500ms
RTT	1s
TT	100s
Application	Bulk Data Tranfer
TCP	New Reno SNACK
Queue	DropTail
Metric	Alltraced
Error	No error
Blackout	No blackout

Forward-Link Capacity (Kbps)	Return-Link Capacity (Kbps)	Ratio	Throughput (Bps)	Goodput (Bps)
1000	1000	1	116324,39	111850
1000	500	2	116282,8	111810
1000	100	10	115981,2	111520
1000	20	50	62244,4	59850
1000	10	100	32032,4	30800
1000	5	200	16775,59	16130
1000	2,5	400	9058,79	8710
1000	2	500	7509,19	7220
1000	1,5	666,66667	5949,19	5720
1000	1,25	800	5169,19	4970
1000	1	1000	4389,19	4220

DBB	1s
RTT	2s
TT	200s
Application	Bulk Data Tranfer
TCP	New Reno SNACK
Queue	DropTail
Metric	Alltraced
Error	No error
Blackout	No blackout

Forward-Link Capacity (Kbps)	Return-Link Capacity (Kbps)	Ratio	Throughput (Bps)	Goodput (Bps)
1000	1000	1	58687,4	56430
1000	500	2	58687,4	56430
1000	100	10	58609,4	56355
1000	20	50	57969,8	55740
1000	10	100	31127,4	29930
1000	5	200	16021,4	15405
1000	2,5	400	8387,79	8065
1000	2	500	6848,6	6585
1000	1,5	666,66667	5304,19	5100
1000	1,25	800	4534,6	4360
1000	1	1000	3754,59	3610

DBB	10s
RTT	20s
TT	2000s
Application	Bulk Data Tranfer
TCP	New Reno SNACK
Queue	DropTail
Metric	Alltraced
Error	No error
Blackout	No blackout

Forward-Link Capacity (Kbps)	Return-Link Capacity (Kbps)	Ratio	Throughput (Bps)	Goodput (Bps)
1000	1000	1	2487,739	2390
1000	500	2	2487,739	2390
1000	100	10	2487,739	2390
1000	20	50	2487,739	2390
1000	10	100	2487,739	2390
1000	5	200	2480,98	2384
1000	2,5	400	2454,98	2359
1000	2	500	2449,26	2353
1000	1,5	666,66667	2427,94	2333
1000	1,25	800	2354,09	2262
1000	1	1000	2150,78	2066

Σφάλματα στο forward link και στο return link

Application	Bulk Data Transfer
TCP	New Reno SNACK
Queue	DropTail
Metric	Alltraced
Blackout	No blackout
Link	Forward

BWBB	1Mbps			
DBB	100ms			
RTT	200ms			
TT	20s			
Error	Throughput (Bps)	Goodput (Bps)	Overhead (%)	Ret.Packets
0,000001	125062	120250	3,85	0
0,00001	125062	120250	3,85	0
0,0001	125062	120250	3,85	0
0,001	109254	98550	9,80	130
0,005	86374	76800	11,08	125
0,01	66770	55950	16,20	165
0,05	21948	17900	18,44	65
0,1	7806	5850	25,06	35

BWBB	1Mbps			
DBB	1s			
RTT	2s			
TT	200s			
Error	Throughput (Bps)	Goodput (Bps)	Overhead (%)	Ret.Packets
0,000001	58687,4	56430	3,85	0
0,00001	58687,4	56430	3,85	0
0,0001	58687,4	56430	3,85	0
0,001	16198,2	15070	6,96	101
0,005	7233,39	6635	8,27	64
0,01	6822,6	6030	11,62	106
0,05	2257,19	1880	16,71	59
0,1	2137,8	1700	20,48	73

BWBB	1Mbps			
DBB	10s			
RTT	20s			
TT	2000s			
Error	Throughput (Bps)	Goodput (Bps)	Overhead (%)	Ret.Packets
0,000001	2487,739	2390	3,93	5
0,00001	2487,739	2390	3,93	5
0,0001	2487,739	2390	3,93	5
0,001	979,74	924	5,69	37
0,005	637,06	594	6,76	38
0,01	504,98	441	12,67	91
0,05	255,92	219	14,43	57
0,1	339,64	259	23,74	137

BWBB	1Mbps			
DBB	60s			
RTT	120s			
TT	12000s			
Error	Throughput (Bps)	Goodput (Bps)	Overhead (%)	Ret.Packets
0,000001	423,3	406	4,09	11
0,00001	423,3	406	4,09	11
0,0001	423,3	406	4,09	11
0,001	188,52	179	5,05	32
0,005	149,86	129	13,92	181
0,01	106,62	94	11,84	100
0,05	67,53	53	21,52	140
0,1	70,3	55	21,76	156

BWBB	1Mbps			
DBB	300s			
RTT	600s			
TT	60000s			
Error	Throughput (Bps)	Goodput (Bps)	Overhead (%)	Ret.Packets
0,000001	21,2046	6	71,70	820
0,00001	21,2046	6	71,70	820
0,0001	21,2046	6	71,70	820
0,001	21,2046	6	71,70	820
0,005	21,17	6	71,66	820
0,01	21,135	6	71,61	819
0,05	19,818	6	69,72	786
0,1	18,847	5	73,47	771

BWBB	1Mbps			
DBB	600s			
RTT	1200s			
TT	120000s			
Error	Throughput (Bps)	Goodput (Bps)	Overhead (%)	Ret.Packets
0,000001	18,576	3	83,85	1703
0,00001	18,576	3	83,85	1703
0,0001	18,576	3	83,85	1703
0,001	18,576	3	83,85	1703
0,005	18,568	3	83,84	1702
0,01	18,429	3	83,72	1698
0,05	18	3	83,33	1680
0,1	17,12	3	82,48	1610

Application	Bulk Data Transfer
TCP	New Reno SNACK
Queue	DropTail
Metric	Alltraced
Blackout	No blackout
Link	Return

BWBB	1Mbps			
DBB	100ms			
RTT	200ms			
TT	20s			
Error	Throughput (Bps)	Goodput (Bps)	Overhead (%)	Ret.packets
0,000001	125062	120250	3,85	0
0,00001	125062	120250	3,85	0
0,0001	125062	120250	3,85	0
0,001	125062	120250	3,85	0
0,005	125062	120250	3,85	0
0,01	125062	120250	3,85	0
0,05	124854	120050	3,85	0
0,1	84504	81250	3,85	1

BWBB	1Mbps			
DBB	1s			
RTT	2s			
TT	200s			
Error	Throughput (Bps)	Goodput (Bps)	Overhead (%)	Ret.packets
0,000001	58687,4	56430	3,85	5
0,00001	58687,4	56430	3,85	5
0,0001	58687,4	56430	3,85	5
0,001	58687,4	56430	3,85	5
0,005	58682,19	56425	3,85	5
0,01	58677	56420	3,85	5
0,05	58229,8	55990	3,85	5
0,1	20223,2	19445	3,85	5

BWBB	1Mbps			
DBB	10s			
RTT	20s			
TT	2000s			
Error	Throughput (Bps)	Goodput (Bps)	Overhead (%)	Ret.packets
0,000001	2487,739	2390	3,93	0
0,00001	2487,739	2390	3,93	0
0,0001	2487,739	2390	3,93	0
0,001	2483,06	2386	3,91	0
0,005	2471,62	2375	3,91	0
0,01	2456,54	2360	3,93	0
0,05	2352,02	2260	3,91	0
0,1	2174,18	2089	3,92	1

BWBB	1Mbps			
DBB	60s			
RTT	120s			
TT	12000s			
Error	Throughput (Bps)	Goodput (Bps)	Overhead (%)	Ret.packets
0,000001	423,3	406	4,09	820
0,00001	423,3	406	4,09	820
0,0001	423,3	406	4,09	820
0,001	421,13	404	4,07	820
0,005	419,31	402	4,13	833
0,01	414,72	398	4,03	834
0,05	400,25	384	4,06	826
0,1	369,56	354	4,21	750

BWBB	1Mbps			
DBB	300s			
RTT	600s			
TT	60000s			
Error	Throughput (Bps)	Goodput (Bps)	Overhead (%)	Ret.packets
0,000001	21,2046	6	71,70	11
0,00001	21,2046	6	71,70	11
0,0001	21,2046	6	71,70	11
0,001	21,2046	6	71,70	11
0,005	21,43	6	72,00	11
0,01	21,447	6	72,02	11
0,05	21,256	6	71,77	11
0,1	19,35	6	68,99	10

Διακοπή συνδεσιμότητας

Application	Bulk Data Transfer				
TCP	SACK				
Queue	DropTail				
Bandwidth Backbone	1 Mbps				
Metric	alltraced				
Error	No error				
Blackout times	1				
Blackout duration	2*DBB				
DBB (s)	RTT (s)	TT (s)	Goodput (Bps)	Throughput (Bps)	Ret.packets
0,1	0,2	20	101400	111126	109
0,5	1	100	38440	39998,8	2
1	2	200	21525	22407	4
5	10	1000	4469	4647,92	3
10	20	2000	2234	2323,998	4
60	120	12000	380	395,48	6
100	200	20000	9	20,194	201

Application	Bulk Data Tranfer				
TCP	SACK				
Queue	DropTail				
Bandwidth Backbone	1 Mbps				
Metric	alltraced				
Error	No error				
Blackout times	1				
Blackout duration	10*DBB				
DBB (s)	RTT (s)	TT (s)	Goodput (Bps)	Throughput (Bps)	Ret.packets
0,1	0,2	20	101400	111126	109
0,5	1	100	32140	33457,2	3
1	2	200	19315	20113,8	5
5	10	1000	3839	3992,8	5
10	20	2000	1919	1996,42	6
60	120	12000	341	355,62	9
100	200	20000	9	19,37	197

Application	Bulk Data Tranfer				
TCP	SACK				
Queue	DropTail				
Bandwidth Backbone	1 Mbps				
Metric	alltraced				
Error	No error				
Blackout times	2				
Blackout duration	2*DBB				
DBB (s)	RTT (s)	TT (s)	Goodput (Bps)	Throughput (Bps)	Ret.packets
0,1	0,2	20	89150	101558	170
0,5	1	100	17830	18585,2	4
1	2	200	10005	10436,6	6
5	10	1000	2042	2171,68	49
10	20	2000	1043	1085,86	6
60	120	12000	173	184,88	52
100	200	20000	9	19,83	199

Application	Bulk Data Tranfer
TCP	SACK
Queue	DropTail
Bandwidth Backbone	1 Mbps
Metric	alltraced
Error	No error

Blackout times	2				
Blackout duration	10*DBB				
DBB (s)	RTT (s)	TT (s)	Goodput (Bps)	Throughput (Bps)	Ret.packets
0,1	0,2	20	71150	82890	171
0,5	1	100	13160	14102,8	40
1	2	200	8120	8486,6	8
5	10	1000	1585	1690,24	45
10	20	2000	812	846,7	9
60	120	12000	145	151,26	12
100	200	20000	8	18,38	191

Application	Bulk Data Tranfer				
TCP	SACK				
Queue	DropTail				
Bandwidth Backbone	1 Mbps				
Metric	alltraced				
Error	No error				
Blackout times	5				
Blackout duration	2*DBB				
DBB (s)	RTT (s)	TT (s)	Goodput (Bps)	Throughput (Bps)	Ret.packets
0,1	0,2	20	58150	70722	197
0,5	1	100	6450	7134,8	41
1	2	200	3240	3588,2	42
5	10	1000	645	741,68	71
10	20	2000	352	370,86	12
60	120	12000	55	63,2	74
100	200	20000	8	18,738	193

Application	Bulk Data Tranfer				
TCP	SACK				
Queue	DropTail				
Bandwidth Backbone	1 Mbps				
Metric	alltraced				
Error	No error				
Blackout times	5				
Blackout duration	10*DBB				
DBB (s)	RTT (s)	TT (s)	Goodput (Bps)	Throughput (Bps)	Ret.packets
0,1	0,2	20	52250	64638	198
0,5	1	100	3010	3505,2	36
1	2	200	1825	2043,8	28
5	10	1000	356	401,68	35
10	20	2000	166	196,18	50
60	120	12000	31	36,35	54
100	200	20000	6	15,47	171

Bandwidth X Delay Product

Application	Bulk	DBB (s)	BWBB (Mbps)	BW*Delay	Goodput	Throughput	Ret. Packets
	Data Transfer	0,1	5	0,5	559250	581622	0
TCP	SACK	1	0,5	0,5	55925	58162,2	0
Queue	DropTail	2	0,25	0,5	11455	11913,4	1
Metric	alltraced	5	0,1	0,5	4582	4765,39	2
Error	noerror	10	0,05	0,5	2291	2383,22	3
Blackout	No blackout	20	0,025	0,5	1145	1191,62	4
		100	0,005	0,5	9	20,19	199
		200	0,0025	0,5	4	12,618	297

Application	Bulk	DBB (s)	BWBB (Mbps)	BW*Delay	Goodput	Throughput	Ret. Packets
	Data Transfer	0,1	5	0,5	559250	581622	0
TCP	New Reno	1	0,5	0,5	55925	58162,2	0
	SNACK	2	0,25	0,5	11535	11996,6	1
Queue	DropTail	5	0,1	0,5	4654	4840,28	2
Metric	alltraced	10	0,05	0,5	2352	2448,22	5
Error	noerror	20	0,025	0,5	1152	1199,68	9
Blackout	No blackout	100	0,005	0,5	236	246,6	9
		200	0,0025	0,5	9	24,29	560

Συνδυασμός προβλημάτων (PER, Asymmetry, Blackout)

Application	Bulk Data Transfer
TCP	SACK
Queue	DropTail
Metric	Alltraced
Bandwidth	1 Mbps
Backbone	
DBB	1
Asymmetry	10
Blackout times	1
Blackout duration	2*DBB

PER	Goodput (Bps)	Throughput (Bps)	Ret.packets
0,000001	21430	22308,2	4
0,0001	21430	22308,2	4
0,01	3810	4040,59	15

Application	Bulk Data Tranfer
TCP	SACK
Queue	DropTail
Metric	Alltraced
Bandwidth	1 Mbps
Backbone	
DBB	1
Asymmetry	10
Blackout times	1
Blackout duration	10*DBB

PER	Goodput (Bps)	Throughput (Bps)	Ret.packets
0,000001	19205	19999,4	5
0,0001	19205	19999,4	5
0,01	3585	384,8	16

Application	Bulk Data Tranfer
TCP	SACK
Queue	DropTail
Metric	Alltraced
Bandwidth	1 Mbps
Backbone	
DBB	1
Asymmetry	100
Blackout times	1
Blackout duration	2*DBB

PER	Goodput (Bps)	Throughput (Bps)	Ret.packets
0,000001	19130	19916,2	4
0,0001	19130	19916,2	4
0,01	3705	3931,4	15

Application	Bulk Data Tranfer
TCP	SACK
Queue	DropTail
Metric	Alltraced
Bandwidth	1 Mbps
Backbone	
DBB	1
Asymmetry	100
Blackout times	1
Blackout duration	10*DBB

PER	Goodput (Bps)	Throughput (Bps)	Ret.packets
0,000001	17520	18247	5
0,0001	17520	18247	5
0,01	3530	3749,4	15

Application	Bulk Data Transfer
TCP	SACK
Queue	DropTail
Metric	Alltraced
Bandwidth	1 Mbps
Backbone	
DBB	1
Asymmetry	1000
Blackout times	1
Blackout duration	2*DBB

PER	Goodput (Bps)	Throughput (Bps)	Ret.packets
0,000001	2955	3094,19	4
0,0001	2955	3094,19	4
0,01	2770	2927,8	9

Application	Bulk Data Transfer
TCP	SACK
Queue	DropTail
Metric	Alltraced
Bandwidth	1 Mbps
Backbone	
DBB	1
Asymmetry	1000
Blackout times	1
Blackout duration	10*DBB

PER	Goodput (Bps)	Throughput (Bps)	Ret.packets
0,000001	2770	2907	5
0,0001	2770	2907	5
0,01	2575	2730,19	10