

ΑΠΟΤΙΜΗΣΗ ΑΠΟΔΟΣΗΣ ΔΙΚΤΥΩΝ ΠΟΥ ΧΡΗΣΙΜΟΠΟΙΟΥΝ
ΑΞΙΟΠΙΣΤΑ ΠΡΩΤΟΚΟΛΛΑ ΜΕΤΑΦΟΡΑΣ ΚΑΙ
ΑΞΙΟΠΙΣΤΑ ΠΡΩΤΟΚΟΛΛΑ ΣΥΝΔΕΣΗΣ

Ιωάννης Κόμνιος

Μεταπτυχιακή Διατριβή
Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Δημοκρίτειο Πανεπιστήμιο Θράκης

Ξάνθη
Οκτώβριος 2009

Περίληψη

Το Διαδίκτυο έχει καταφέρει τη διασύνδεση ενός μεγάλου αριθμού διαφορετικών δικτύων τα οποία βασίζουν τη λειτουργία τους σε ξεχωριστές επικοινωνιακές τεχνολογίες και προσανατολίζονται στην κάλυψη κάποιων ιδιαίτερων δικτυακών συνθηκών. Ωστόσο, η λειτουργία του Διαδικτύου βασίζεται σε κάποιες βασικές παραδοχές, οι οποίες καταστρατηγούνται σε ορισμένα δίκτυα. Οι παραδοχές αυτές συνοπτικά περιλαμβάνουν τις σχετικά χαμηλές καθυστερήσεις μετάδοσης, την ύπαρξη συνεχούς από-άκρο-σε-άκρο συνδεσιμότητας και τον χαμηλό ρυθμό λαθών. Τα κλασικά πρωτόκολλα μεταφοράς που αναπτύχθηκαν για το Διαδίκτυο, όπως το Transmission Control Protocol, δεν λειτουργούν αποδοτικά σε δίκτυα στα οποία καταστρατηγούνται οι παραπάνω παραδοχές, όπως είναι τα ασύρματα και τα δορυφορικά δίκτυα. Έτσι, αναπτύχθηκαν νέες τεχνικές που παρέχουν αξιοπιστία σε διαφορετικά επίπεδα για να καλύψουν τις ανάγκες των δικτύων αυτών.

Στην παρούσα εργασία μελετώνται εκτενώς οι μηχανισμοί αξιοπιστίας επιπέδου μεταφοράς και επιπέδου σύνδεσης σε ασύρματα και δορυφορικά δίκτυα και επισημαίνονται τα πλεονεκτήματα της κάθε μεθόδου. Εξετάζεται η επιβάρυνση που συνεπάγεται η εφαρμογή μηχανισμών αξιοπιστίας σε δύο επίπεδα και η αντίστοιχη βελτίωση της απόδοσης που επιτυγχάνεται. Αρχικά, μέσω κάποιων πειραμάτων αναφοράς, εξετάζεται η επίδραση των αξιόπιστων πρωτοκόλλων μεταφοράς και σύνδεσης σε επίγεια ασύρματα δίκτυα, τα οποία εμφανίζουν σφάλματα ή συμφόρηση. Στη συνέχεια, προχωρούμε σε σενάρια καινοτομίας, στα οποία εξετάζεται η απόδοση αξιόπιστων μηχανισμών σε δύο επίπεδα σε δορυφορικά δίκτυα, τα οποία χαρακτηρίζονται από υψηλές τιμές καθυστέρησης και ρυθμούς σφαλμάτων. Ακόμη, μελετάται η επίδραση του πλήθους των ενδιάμεσων κόμβων ανάμεσα στον αποστολέα και τον παραλήπτη στην τελική απόδοση του δικτύου, όταν η καθυστέρηση διάδοσης παίρνει μεγάλες τιμές.

Για την εκτέλεση του συνόλου των πειραμάτων χρησιμοποιήθηκε ο προσομοιωτής δικτύων Network Simulator 2 (NS-2 version 2.33) σε λειτουργικό σύστημα Ubuntu 8.04. Συνοπτικά, καταλήγουμε στα παρακάτω συμπεράσματα όσον αφορά την εφαρμογή αξιόπιστων πρωτοκόλλων στα δικτυακά επίπεδα μεταφοράς και σύνδεσης.

Παρατηρήσαμε ότι, ακόμη και για μικρούς ρυθμούς σφαλμάτων στο δίκτυο, η χρήση μηχανισμών αξιοπιστίας μόνο στο επίπεδο μεταφοράς οδηγεί σε ραγδαία πτώση της απόδοσης. Ωστόσο, η πτώση αυτή δεν είναι τόσο απότομη χρησιμοποιώντας αξιόπιστα πρωτόκολλα επιπέδου σύνδεσης.

Για μικρό επίπεδο συμφόρησης παρατηρήσαμε ότι η χρήση αξιόπιστων μηχανισμών σε δύο επίπεδα οδηγεί σε επιβάρυνση και, συνεπώς, σε μείωση της απόδοσης. Αντίθετα, για μεγάλα επίπεδα συμφόρησης τα αξιόπιστα πρωτόκολλα μεταφοράς δεν είναι αρκετά αποδοτικά, γεγονός που αντισταθμίζεται από την εφαρμογή αξιοπιστίας στο επίπεδο σύνδεσης.

Σε δορυφορικά δίκτυα, τα οποία χαρακτηρίζονται από υψηλές τιμές καθυστέρησης διάδοσης και σφάλματα, το πρωτόκολλο TCP επηρεάζεται σημαντικά για αυξανόμενο ρυθμό σφαλμάτων και δεν μπορεί να λειτουργήσει αποδοτικά. Αντίθετα, η εφαρμογή αξιοπιστίας στο επίπεδο σύνδεσης αντισταθμίζει την αδυναμία του TCP, επαναμεταδίδοντας τα χαμένα πακέτα πριν το αντιληφθεί το επίπεδο μεταφοράς και διατηρώντας, έτσι, την απόδοση του συστήματος σε υψηλά επίπεδα.

Όσον αφορά την επίδραση του πλήθους των ενδιάμεσων κόμβων ανάμεσα στον αποστολέα και τον παραλήπτη, χρησιμοποιώντας το μηχανισμό ARQ παρατηρούμε μια ελάχιστη μείωση της απόδοσης για αυξανόμενο πλήθος ενδιάμεσων κόμβων, ανεξάρτητα από το ρυθμό σφαλμάτων που εμφανίζει το δίκτυο. Αντίθετα, χρησιμοποιώντας το πρωτόκολλο σύνδεσης Snoop και για μεγάλους ρυθμούς σφαλμάτων παρατηρούμε ότι η απόδοση του συστήματος αυξάνεται για αυξανόμενο πλήθος ενδιάμεσων κόμβων.

Με λίγα λόγια, αποδεικνύεται πως τα υπάρχοντα αξιόπιστα πρωτόκολλα μεταφοράς δεν λειτουργούν αποδοτικά σε δορυφορικά δίκτυα. Για το λόγο αυτό, είναι απαραίτητη η ταυτόχρονη εφαρμογή μηχανισμών αξιοπιστίας στο επίπεδο σύνδεσης, οι οποίοι θα αντισταθμίζουν τις αδυναμίες του επιπέδου μεταφοράς και θα διατηρούν την απόδοση σε υψηλά επίπεδα. Σε κάθε περίπτωση θα πρέπει να λαμβάνονται υπόψη συγκεκριμένες παράμετροι του δικτύου (ρυθμός σφαλμάτων, πλήθος κόμβων κτλ.) για την εφαρμογή των καταλληλότερων πρωτοκόλλων και μηχανισμών.

Abstract

Internet has managed to interconnect a number of networks that rely on different communication technologies and cope with special network conditions. However, the Internet is based on certain principles, which may be violated by some networks. These principles, in essence, include low transmission delay values, continuous end-to-end connectivity and low error rates. The classic transport protocols that were developed for the Internet, like Transmission Control Protocol, do not perform efficiently in networks where the aforementioned principles are violated, such as wireless and satellite networks. Thus, new techniques were developed to provide reliability in different network layers and, therefore, satisfy the needs of these networks.

In this thesis, we extensively study reliability mechanisms in both transport and link layers, in wireless and satellite networks, and we underline the advantages of each method. We study the overhead that entails the implementation of reliability mechanisms in two distinct network layers and the correspondent performance increase. Initially, through a set of reference scenarios, we study the impact of reliable transport and link layer protocols in terrestrial wireless networks, which suffer from errors or congestion. Next, we continue with a set of innovation scenarios, where we study the performance of reliable protocols in satellite networks, which are characterized by high propagation delay and error rate values. Moreover, we study the impact that the number of intermediate nodes between the sender and the receiver has on system performance, when propagation delay values are high.

For the conduction of the simulation experiments, Network Simulator 2 (NS-2 version 2.33) and operating system Ubuntu 8.04 were used. In summary, we reach the following conclusions, as far as applying reliable protocols in transport and link layers is concerned.

We noticed that, even for low error rate values, the implementation of reliability mechanisms only on transport layer leads to a rapid performance decrease. However, this decrease is not so sharp when we incorporate reliable link layer protocols as well.

For low network congestion conditions, we concluded that the implementation of reliability mechanisms simultaneously in two layers leads to overhead and, thus, performance decrease. On the other hand, for high congestion conditions, reliable transport layer protocols are not efficient, a fact that is balanced by the implementation of reliability in link layer.

In satellite networks, which are characterized by high propagation delay values and errors, TCP protocol is considerably affected for increasing error rate and cannot perform well. However, implementation of reliable protocols on link layer compensates for the inability of TCP, by retransmitting the lost segments before transport layer detects the loss and, thus, keeping system efficiency high.

As far as the impact that the number of intermediate nodes between the sender and the receiver has on performance is concerned, we notice a slight decrease in performance for increasing number of intermediate nodes when applying the ARQ mechanism, independent of the error rate of the network. On the contrary, when implementing Snoop link layer protocol for high error rates, we note that system performance rises for increasing number of intermediate nodes.

In a few words, we prove that the existing reliable transport protocols do not perform efficiently in satellite networks. This is the reason why implementing reliable mechanisms in link layer is necessary too, in order to deal with the inefficiencies of transport layer and maintain performance values high. In any case, certain parameters of the network (error rate, number of nodes etc.) should be taken into consideration for the selection of the most suitable protocols and mechanisms.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον καθηγητή κ. Β. Τσαουσίδα για την πολύτιμη βοήθεια και καθοδήγηση κατά την εκπόνηση της μεταπτυχιακής μου διατριβής.

Επίσης, θα ήθελα να ευχαριστήσω τους συναδέλφους της ερευνητικής ομάδας ComNet, Θ. Αμανατίδη, Σ. Διαμαντόπουλο, Θ. Κουτσογιάννη, Γ. Παπαστεργίου, Χ. Σαμαρά και Α. Τσιολιαρίδου για την άψογη συνεργασία και υποστήριξη.

Περιεχόμενα

1. Εισαγωγή	20
2. Πρωτόκολλα μεταφοράς	23
2.1. Transmission Control Protocol (TCP).....	23
2.1.1. Χαρακτηριστικά του TCP	23
2.1.1.1. Επικεφαλίδα	25
2.1.1.2. Εγκαθίδρυση σύνδεσης	27
2.1.1.3. Τερματισμός σύνδεσης.....	29
2.1.1.4. Αξιοπιστία και ταξινόμηση	30
2.1.1.5. Χρονικό Διάστημα Επαναποστολής	31
2.1.1.6. Καθυστερημένες επιβεβαιώσεις	33
2.1.2. Έλεγχος ροής και Έλεγχος Συμφόρησης.....	33
2.1.2.1. Έλεγχος ροής.....	33
2.1.2.2. Έλεγχος και Αποφυγή Συμφόρηση	34
2.1.2.3. Κριτήρια κατάστασης ελέγχου συμφόρησης	36
2.1.2.4. Μηχανισμοί Αργό Ξεκίνημα και Αποφυγή Συμφόρησης του TCP	37
2.1.2.5. Μηχανισμοί Γρήγορης Επαναμετάδοσης και Γρήγορης Ανάκαμψης	40
2.1.2.6. Το Μοντέλο Ανάλυσης για τον Έλεγχο Συμφόρησης	41
2.1.2.7. Προσθετική αύξηση / Πολλαπλασιαστική μείωση	43
2.1.3. Εκδόσεις του πρωτοκόλλου μεταφοράς TCP	45
2.1.3.1. TCP Tahoe	45
2.1.3.2. TCP Reno	46
2.1.3.3. TCP New Reno.....	46
2.1.3.4. TCP SACK	47
2.1.3.5. TCP Vegas.....	48
2.1.3.6. TCP Real	49

2.1.4. Ενεργός Διαχείριση της Ουράς.....	51
2.1.4.1. DropTail.....	51
2.1.4.2. Random Early Drop (RED)	52
2.2. User Datagram Protocol (UDP)	54
2.2.1. Χαρακτηριστικά UDP	54
2.2.1.1. Επικεφαλίδα	55
2.2.1.2. Εφαρμογές	57
2.2.2. Πλεονεκτήματα του UDP.....	58
2.3. Σύγκριση των πρωτοκόλλων TCP και UDP	59
3. Πρωτόκολλα σύνδεσης.....	61
3.1. Πρωτόκολλο Snoop	61
3.2. Automatic Repeat reQuest (ARQ)	68
3.2.1. Pure ARQ	68
3.2.1.1. Πρωτόκολλα επαναμετάδοσης	69
3.2.2. Hybrid-ARQ.....	71
3.3. Forward Error Correction (FEC).....	72
4. Δορυφορικά δίκτυα και μηχανισμοί αξιοπιστίας	75
4.1. Δορυφορικά δίκτυα.....	75
4.1.1. Τύποι δορυφόρων	75
4.1.2. Διαφορές δικτυακών χαρακτηριστικών ανάμεσα στο Διαδίκτυο και τα δορυφορικά δίκτυα	77
4.1.3. Προβλήματα που παρουσιάζει το TCP στο Διάστημα	79
4.2. Μηχανισμοί αξιοπιστίας σε ασύρματα και δορυφορικά δίκτυα	81
4.2.1. Μηχανισμοί από άκρο σε άκρο	82
4.2.2. Μηχανισμοί διαχωρισμού σύνδεσης.....	84
4.2.3. Μηχανισμοί στο επίπεδο σύνδεσης	86
4.2.4. Σύγκριση των διαφόρων μηχανισμών	88

5. Εργαστηριακή Μεθοδολογία	91
5.1 Τοπολογία των πειραμάτων και γενικές παράμετροι	91
5.2 Μονάδες Μέτρησης της Απόδοσης	93
5.2.1. Bandwidth Efficiency	93
5.2.2. Delay x Bandwidth Product	95
5.3. Δείκτης Κατανάλωσης Ενέργειας	97
5.4. Σενάρια προσομοίωσης και θεωρητική ανάλυση	98
5.4.1. Επίγεια δίκτυα που εμφανίζουν σφάλματα	98
5.4.2. Δίκτυα που εμφανίζουν συμφόρηση	98
5.4.3. Επέκταση σε δορυφορικά περιβάλλοντα	99
5.4.4. Αύξηση του πλήθους των ενδιάμεσων κόμβων	100
6. Πειραματικά Αποτελέσματα	101
6.1. Σενάρια αναφοράς	101
6.1.1. Επίγεια δίκτυα που εμφανίζουν σφάλματα	101
6.1.1.1. Πειράματα χρησιμοποιώντας πρωτόκολλο σύνδεσης ARQ και τα πρωτόκολλα μεταφοράς TCP New Reno και TCP SACK	102
6.1.1.2. Πειράματα χρησιμοποιώντας πρωτόκολλο σύνδεσης Snoop και τα πρωτόκολλα μεταφοράς TCP New Reno και TCP SACK	104
6.1.2. Δίκτυα που εμφανίζουν συμφόρηση	105
6.1.2.1. Πειράματα χρησιμοποιώντας πρωτόκολλο σύνδεσης ARQ και το πρωτόκολλο μεταφοράς TCP New Reno	106
6.1.2.2. Πειράματα χρησιμοποιώντας πρωτόκολλο σύνδεσης Snoop και το πρωτόκολλο μεταφοράς TCP New Reno	107
6.2. Σενάρια καινοτομίας	109
6.2.1. Επέκταση σε δορυφορικά περιβάλλοντα	109
6.2.1.1. Πειράματα χρησιμοποιώντας πρωτόκολλο σύνδεσης ARQ και τα πρωτόκολλα μεταφοράς TCP New Reno και TCP SACK	109

6.2.1.2. Πειράματα χρησιμοποιώντας πρωτόκολλο σύνδεσης Snoop και το πρωτόκολλο μεταφοράς TCP New Reno.....	113
6.2.2. Αύξηση του πλήθους των ενδιάμεσων κόμβων	116
6.2.2.1. Πειράματα χρησιμοποιώντας πρωτόκολλο σύνδεσης ARQ και το πρωτόκολλο μεταφοράς TCP New Reno.....	116
6.2.2.2. Πειράματα χρησιμοποιώντας πρωτόκολλο σύνδεσης Snoop και το πρωτόκολλο μεταφοράς TCP New Reno.....	120
6.3. Συμπεράσματα και σχόλια	124
Βιβλιογραφία	127

Συντομογραφίες

ACK	Acknowledgment
adwnd	Advertised Window
AIMD	Additive Increase / Multiplicative Decrease
ARQ	Automatic Repeat reQuest
AQM	Active Queue Management
BDP	Bandwidth Delay Product
BER	Bit Error Rate
Bps	Bytes per second
BW	Bandwidth
CRC	Cyclic Redundancy Check
cwnd	Congestion Window
DACK	Duplicate Acknowledgment
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
ECN	Explicit Congestion Notification
ELN	Explicit Loss Notification
FEC	Forward Error Correction
FIFO	First In First Out
FTP	File Transfer Protocol
HARQ	Hybrid Automatic Repeat reQuest
HTTP	Hypertext Transfer Protocol
I-TCP	Indirect Transmission Control Protocol

IP	Internet Protocol
LBA	Last Byte Acked
LBS	Last Byte Sent
LEO	Low Earth Orbit
M-TCP	Mobile Transmission Control Protocol
Mbps	Megabits per second
MSR	Mobile Support Router
MSS	Maximum Sending Size
NAK	Negative Acknowledgment
NFS	Network File Server
NS	Network Simulator
OSI	Open System Interconnection
RED	Random Early Detection
RIP	Routing Information Protocol
RTO	Retransmission Timeout
RTT	Round Trip Time
SACK	Selective Acknowledgment
SMART	Simple Method to Aid ReTransmissions
SNMP	Simple Network Management Protocol
SR-ARQ	Selective Repeat Automatic Repeat reQuest
SRP	Selective Repeat Protocol
ssthresh	Slow Start Threshold
SW-ARQ	Stop-and-Wait Automatic Repeat reQuest

TCP	Transmission Control Protocol
TULIP	TCP and UDP over Lightweight Internet Protocol
UDP	User Datagram Protocol
VoIP	Voice over Internet Protocol

Πίνακας Σχημάτων

Σχήμα 2.1. Επικεφαλίδα του TCP.....	26
Σχήμα 2.2. Τριπλή «χειραψία»	28
Σχήμα 2.3 Διάγραμμα μετάβασης καταστάσεων.....	28
Σχήμα 2.4. Διάγραμμα μετάβασης καταστάσεων για τον τερματισμό σύνδεσης με το TCP	29
Σχήμα 2.5. Τέσσερα ενδεχόμενα κατά την επικοινωνία	31
Σχήμα 2.6 Αργό Ξεκίνημα	38
Σχήμα 2.7. Το throughput ως Συνάρτηση του Φορτίου του δικτύου	42
Σχήμα 2.8. Το Μοντέλο του Συστήματος Ελέγχου	42
Σχήμα 2.9. Γεωμετρική παράσταση της σύγκλισης με τον αλγόριθμο AIMD.....	44
Σχήμα 2.10. Πιθανότητα απόρριψης πακέτων για αυξανόμενο μέσο μήκος ουράς για τον αλγόριθμο διαχείρισης ουράς RED	53
Σχήμα 2.11 Επικεφαλίδα UDP	55
Σχήμα 3.1 Τοπολογία εφαρμογής του πρωτοκόλλου Snooper	61
Σχήμα 3.2-3.11 Παράδειγμα αποστολής δεδομένων χρησιμοποιώντας το πρωτόκολλο Snooper.....	64-68
Σχήμα 3.12 Κωδικοποίηση και αποκωδικοποίηση δεδομένων χρησιμοποιώντας Forward Error Correction.....	73
Σχήμα 4.1 Αρχιτεκτονική δορυφορικού δικτύου	76
Σχήμα 4.2 Μέγιστη τιμή καθυστέρησης για αυξανόμενη απόσταση από την επιφάνεια της Γης.....	78
Σχήμα 4.3 Μηχανισμός από άκρο σε άκρο	83
Σχήμα 4.4 Μηχανισμός διαχωρισμού σύνδεσης.....	85
Σχήμα 4.5 Μηχανισμός στο επίπεδο σύνδεσης	87
Σχήμα 5.1 Ενδεικτική τοπολογία πειραμάτων	92
Σχήμα 5.2 Bandwidth X Delay Product	97

Σχήμα 6.1 Τοπολογία πειραμάτων	102
Σχήμα 6.2 Goodput - Error Rate για καθυστέρηση διάδοσης 100ms και αποστολή δεδομένων 20MB	102
Σχήμα 6.3 Goodput - Error Rate για καθυστέρηση διάδοσης 200ms και αποστολή δεδομένων 20MB	103
Σχήμα 6.4 Goodput - Error Rate για καθυστέρηση διάδοσης 100ms και αποστολή δεδομένων 20MB	103
Σχήμα 6.5 Goodput - Error Rate για καθυστέρηση διάδοσης 100ms και αποστολή δεδομένων 20MB	104
Σχήμα 6.6 Goodput - Error Rate για καθυστέρηση διάδοσης 100ms και αποστολή δεδομένων 20MB	104
Σχήμα 6.7 Τοπολογία πειραμάτων	105
Σχήμα 6.8 Goodput - Error Rate για καθυστέρηση διάδοσης 100ms και αποστολή δεδομένων 2MB	106
Σχήμα 6.9 Goodput – Total Transmitted Data για καθυστέρηση διάδοσης 100ms.	107
Σχήμα 6.10 Goodput - Error Rate για καθυστέρηση διάδοσης 100ms και αποστολή δεδομένων 2MB	108
Σχήμα 6.11 Goodput - Total Transmitted Data για καθυστέρηση διάδοσης 100ms	108
Σχήμα 6.12 Τοπολογία πειραμάτων	109
Σχήμα 6.13 Goodput – Καθυστέρηση διάδοσης για μηδενικό ρυθμό σφαλμάτων	110
Σχήμα 6.14 Goodput – Καθυστέρηση διάδοσης για ρυθμό σφαλμάτων 1%	110
Σχήμα 6.15 Goodput – Καθυστέρηση διάδοσης για ρυθμό σφαλμάτων 5%	111
Σχήμα 6.16 Goodput – Καθυστέρηση διάδοσης για ρυθμό σφαλμάτων 10%	111
Σχήμα 6.17 Goodput – Ρυθμός σφαλμάτων για καθυστέρηση διάδοσης 400ms ...	112
Σχήμα 6.18 Goodput – Ρυθμός σφαλμάτων για καθυστέρηση διάδοσης 600ms ...	112
Σχήμα 6.19 Goodput – Ρυθμός σφαλμάτων για καθυστέρηση διάδοσης 400ms ...	112
Σχήμα 6.20 Τοπολογία πειραμάτων	113
Σχήμα 6.21 Goodput – Καθυστέρηση διάδοσης για μηδενικό ρυθμό σφαλμάτων	113
Σχήμα 6.22 Goodput – Καθυστέρηση διάδοσης για ρυθμό σφαλμάτων 1%	114

Σχήμα 6.23 Goodput – Καθυστέρηση διάδοσης για ρυθμό σφαλμάτων 5%	114
Σχήμα 6.24 Goodput – Καθυστέρηση διάδοσης για ρυθμό σφαλμάτων 10%	115
Σχήμα 6.25 Goodput – Ρυθμός σφαλμάτων για καθυστέρηση διάδοσης 400ms ...	115
Σχήμα 6.26 Goodput – Ρυθμός σφαλμάτων για καθυστέρηση διάδοσης 600ms ...	115
Σχήμα 6.27 Τοπολογία πειραμάτων	116
Σχήμα 6.28 Goodput – Συνολικό πλήθος κόμβων για μηδενικό ποσοστό σφαλμάτων	117
Σχήμα 6.29 Goodput – Συνολικό πλήθος κόμβων για ποσοστό σφαλμάτων 1%	117
Σχήμα 6.30 Goodput – Συνολικό πλήθος κόμβων για ποσοστό σφαλμάτων 10% ..	118
Σχήμα 6.31 Επαναμεταδόσεις χαμένων πακέτων από το επίπεδο μεταφοράς συναρτήσκει του ρυθμού σφαλμάτων.....	118
Σχήμα 6.32 Goodput – Συνολικό πλήθος κόμβων για καθυστέρηση διάδοσης 100ms.....	119
Σχήμα 6.33 Goodput – Συνολικό πλήθος κόμβων για καθυστέρηση διάδοσης 400ms.....	119
Σχήμα 6.34 Goodput – Συνολικό πλήθος κόμβων για καθυστέρηση διάδοσης 800ms.....	119
Σχήμα 6.35 Τοπολογία πειραμάτων	120
Σχήμα 6.36 Goodput – Συνολικό πλήθος κόμβων για μηδενικό ρυθμό σφαλμάτων	121
Σχήμα 6.37 Goodput – Συνολικό πλήθος κόμβων για ρυθμό σφαλμάτων 1%	121
Σχήμα 6.38 Goodput – Συνολικό πλήθος κόμβων για ρυθμό σφαλμάτων 5%	122
Σχήμα 6.39 Goodput – Συνολικό πλήθος κόμβων για ρυθμό σφαλμάτων 10%	122
Σχήμα 6.40 Επαναμεταδόσεις χαμένων πακέτων από το επίπεδο μεταφοράς συναρτήσκει του ρυθμού σφαλμάτων.....	123

Πίνακας 2.1 Γνωστές θύρες UDP	56
Πίνακας 2.2 Σύγκριση πρωτοκόλλων μεταφοράς TCP και UDP	59
Πίνακας 2.3 Εφαρμογές και αντίστοιχο πρωτόκολλο μεταφοράς.....	60
Πίνακας 4.1 Χαρακτηριστικά επίγειων και δορυφορικών επικοινωνιών	79
Πίνακας 4.2 Τροποποιήσεις που απαιτούνται για την εφαρμογή του κάθε πρωτοκόλλου	90

Κεφάλαιο 1

Εισαγωγή

Το Διαδίκτυο (Internet) αποτέλεσε και αποτελεί το κοινό σημείο αναφοράς για πολλά ετερογενή δίκτυα τα οποία καταφέρει και διασυνδέει με επιτυχία. Η επιτυχία του Διαδικτύου είναι αδιαμφισβήτητη, καθώς πολλές συσκευές της καθημερινής μας ζωής (προσωπικοί υπολογιστές, υπολογιστές παλάμης, έξυπνες συσκευές, VoIP τηλεφωνία) αποτελούν μέρος του Διαδικτύου, των οποίων το πλήθος και το εύρος εφαρμογών συνεχώς αυξάνεται.

Η συνδεσιμότητα του Διαδικτύου βασίζεται πρωτίστως σε ενσύρματες συνδέσεις και για το λόγο αυτό τα θέματα που είχε να αντιμετωπίσει ήταν καθαρά θέματα σταθερότητας, ετερογένειας, δικαιοσύνης σε εύρος ζώνης και ελέγχου συμφόρησης για την αποφυγή καταστάσεων καταστροφικής συμφόρησης. Οι ενσύρματες συνδέσεις χαρακτηρίζονται από συνεχή από-άκρο-σε-άκρο συνδεσιμότητα, σχετικά χαμηλές καθυστερήσεις διάδοσης ανάμεσα στον αποστολέα και τον παραλήπτη, χαμηλό ρυθμό λαθών και σχετικά συμμετρικές ταχύτητες μετάδοσης αμφίδρομων συνδέσεων. Οι συνθήκες αυτές αποτελούν τις παραδοχές πάνω στις οποίες σχεδιάστηκε και συνεχίζει να λειτουργεί το Διαδίκτυο. Το πρωτόκολλο μεταφοράς Transmission Control Protocol είναι το πιο διαδεδομένο πρωτόκολλο σήμερα στο Διαδίκτυο και χρησιμοποιείται από την μεγάλη πλειονότητα των δικτυακών εφαρμογών, παρέχοντας αξιόπιστη επικοινωνία από άκρο σε άκρο και υψηλές τιμές απόδοσης.

Παρόλα αυτά οι παραπάνω παραδοχές δεν ισχύουν σε όλα τα είδη δικτύων όπως για παράδειγμα τα ad-hoc δίκτυα, τα ασύρματα δίκτυα αισθητήρων, τα υποβρύχια ακουστικά δίκτυα, τις δορυφορικές και διαστημικές επικοινωνίες. Τα χαρακτηριστικά των δικτύων αυτών δεν επιτρέπουν στο πρωτόκολλο TCP να λειτουργήσει αποδοτικά και απαιτούν να αναπτυχθούν νέα πρωτόκολλα και μηχανισμοί. Οι μηχανισμοί αξιοπιστίας που έχουν αναπτυχθεί για ασύρματα και δορυφορικά δίκτυα χωρίζονται σε διαφορετικές κατηγορίες ανάλογα με τον τρόπο λειτουργία τους. Σημαντικότερη κατηγορία αποτελούν τα πρωτόκολλα επιπέδου σύνδεσης, τα οποία παρέχουν

αξιοπιστία χαμηλότερου επιπέδου, αντισταθμίζοντας τα προβλήματα του επιπέδου μεταφοράς.

Στην παρούσα εργασία γίνεται μια περιγραφή των βασικών μηχανισμών παροχής αξιοπιστίας, τόσο σε επίπεδο μεταφοράς, όσο σε επίπεδο σύνδεσης. Στόχος είναι η μελέτη των μηχανισμών αξιοπιστίας σε διαφορετικά επίπεδα, ειδικά σε περιβάλλοντα που εμφανίζουν υψηλές καθυστερήσεις διάδοσης και ρυθμούς σφαλμάτων, όπως είναι τα δορυφορικά δίκτυα. Παράλληλα μελετάται η επίδραση του πλήθους των ενδιάμεσων κόμβων, ανάμεσα στον αποστολέα και τον παραλήπτη, στην τελική απόδοση του συστήματος.

Στα πλαίσια της παρούσας εργασίας υλοποιήθηκαν δύο αξιόπιστα πρωτόκολλα σύνδεσης στο περιβάλλον του προσομοιωτή δικτύων Network Simulator 2 (NS-2) και εκτελέστηκαν κατάλληλα πειράματα για την μελέτη της επίδρασης τους στην απόδοση διαφόρων τύπων δικτύων. Στα πειράματα αυτά μετράται και συγκρίνεται η απόδοση του δικτύου για δύο διαφορετικούς τρόπους λειτουργίας: εφαρμογή αξιοπιστίας μόνο στο επίπεδο μεταφοράς και ταυτόχρονη εφαρμογή αξιοπιστίας στα επίπεδα μεταφοράς και σύνδεσης.

Επισκόπηση της μεταπτυχιακής διατριβής

Παρακάτω παρατίθεται μια σύντομη περιγραφή του περιεχομένου των επόμενων κεφαλαίων της μεταπτυχιακής διατριβής:

Στο **κεφάλαιο 2** γίνεται μια ανασκόπηση των δύο βασικότερων πρωτοκόλλων μεταφοράς που χρησιμοποιούνται σήμερα στο Διαδίκτυο: το αξιόπιστο πρωτόκολλο μεταφοράς Transmission Control Protocol και το μη αξιόπιστο User Datagram Protocol.

Στο **κεφάλαιο 3** παρουσιάζονται οι σημαντικότεροι μηχανισμοί παροχής αξιοπιστίας στο επίπεδο σύνδεσης. Συγκεκριμένα, αναλύεται η συμπεριφορά του πρωτοκόλλου Snoop και των μηχανισμών Automatic Repeat request και Forward Error Correction.

Στο **κεφάλαιο 4** περιγράφονται τα χαρακτηριστικά των ασύρματων και δορυφορικών δικτύων και επισημαίνονται οι διαφορές από τα επίγεια δίκτυα. Ακόμη, κατηγοριοποιούνται οι διάφοροι μηχανισμοί αξιοπιστίας ανάλογα με τον τρόπο λειτουργία τους.

Στο **κεφάλαιο 5** αναλύεται η μεθοδολογία των πειραμάτων, εξηγούνται οι μονάδες μέτρησης της απόδοσης που χρησιμοποιήθηκαν και περιγράφονται τα σενάρια που εκτελέστηκαν, περιλαμβάνοντας μια σύντομη θεωρητική ανάλυση των αναμενόμενων αποτελεσμάτων.

Το **κεφάλαιο 6** περιλαμβάνει αναλυτικά αποτελέσματα των διαφόρων πειραμάτων που εκτελέστηκαν και συνοψίζει τα βασικά συμπεράσματα της μεταπτυχιακής διατριβής.

Κεφάλαιο 2

Πρωτόκολλα μεταφοράς

Το δικτυακό επίπεδο μεταφοράς είναι μια ομάδα μεθόδων και πρωτοκόλλων που είναι υπεύθυνα για την ενθυλάκωση των μπλοκ δεδομένων εφαρμογής (application data blocks) σε μονάδες δεδομένων, κατάλληλες για μετάδοση πάνω από τη δικτυακή υποδομή. Αντίστοιχα, εκτελεί και την αντίστροφη διαδικασία της εξαγωγής των δεδομένων που θα παραδοθούν στην εφαρμογή. Ουσιαστικά, το επίπεδο μεταφοράς αποτελεί το μέσο που εγκαθιδρύει μια απευθείας, εικονική δίοδο επικοινωνίας ανάμεσα σε έναν αποστολέα και έναν παραλήπτη για τη μετάδοση των δεδομένων και είναι υπεύθυνο για την παράδοση τους στην κατάλληλη εφαρμογή του παραλήπτη. Οι λειτουργίες του επιπέδου μεταφοράς συμπεριλαμβάνουν τη στατιστική πολύπλεξη των δεδομένων από διαφορετικές εφαρμογές και τη μετάδοση τους πάνω από το μέσο μεταφοράς. Ανάλογα με τις ανάγκες της εκάστοτε εφαρμογής, μπορούν να χρησιμοποιηθούν τόσο αξιόπιστα, όπως είναι το Transmission Control Protocol, όσο και μη αξιόπιστα πρωτόκολλα μεταφοράς, όπως είναι το User Datagram Protocol.

2.1. Transmission Control Protocol (TCP)

2.1.1. Χαρακτηριστικά του TCP

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 την 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 τον σχεδιασμό νέων πρωτοκόλλων και αρχιτεκτονικών και επαναξιολόγηση των υπάρχοντων προτύπων. Ακόμη μεγαλύτερα προβλήματα παρουσιάζει η εφαρμογή του TCP στο Διάστημα, λόγω των ιδιαίτερων χαρακτηριστικών του. 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].

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 την 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:

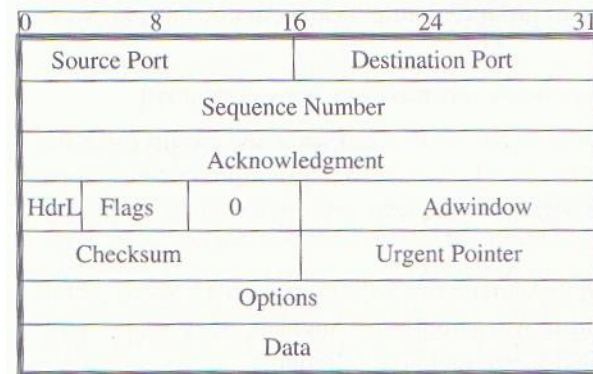
- Εγκαθίδρυση και τερματισμός σύνδεσης (Connection establishment/termination)
- Ανίχνευση της διαθέσιμης χωρητικότητας (Available bandwidth)
- Αξιόπιστη μεταφορά δεδομένων με επαναμεταφορά (Retransmission)
- Επαναταξινόμηση εκτός-σειράς πακέτων
- Έλεγχος ροής (Flow control)
- Έλεγχος και αποφυγή συμφόρησης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

Το Transmission Control Protocol αποτελεί έναν σύνδεσμο όπου από τη μια 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 μια 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.

2.1.1.1. 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 2.1 [5].



Σχήμα 2.1. 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 ποιον 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 την 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 την σκοπιμότητά 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 την 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 την χρήση έξι διαφορετικών συμβολικών μονάδων (σημαιών - Flags):

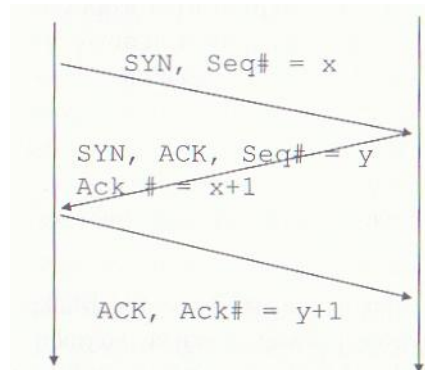
- **SYN, FIN:** Χρησιμοποιούνται στην εγκαθίδρυση και τον τερματισμό σύνδεσης (Connection establishment/ termination)
- **Reset:** Χρησιμοποιείται όταν ο παραλήπτης παίρνει πακέτα που δε μπορεί να κατατάξει.
- **Push:** Σηματοδοτεί άμεση παράδοση των πακέτων στην εφαρμογή
- **URGPNTTR:** Χρησιμοποιείται για να διαχωρίσει ένα επείγον τμήμα του πακέτου, σε συνδυασμό με το πεδίο URGPNTR, το οποίο περιλαμβάνει ένα δείκτη που καθορίζει μέχρι πιο 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.

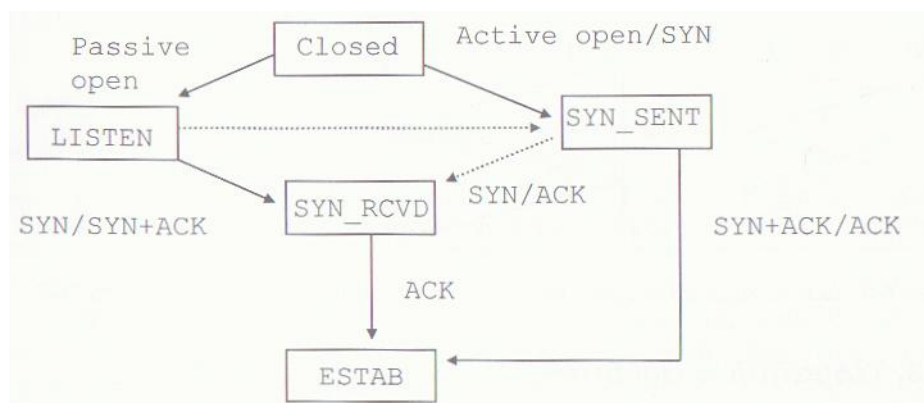
2.1.1.2. Εγκαθίδρυση σύνδεσης (Connection Establishment)

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 υπολογίζουν το AdWindow, αν υπάρχουν κοινές επιλογές κ.λ.π.



Σχήμα 2.2. Τριπλή «χειραψία»

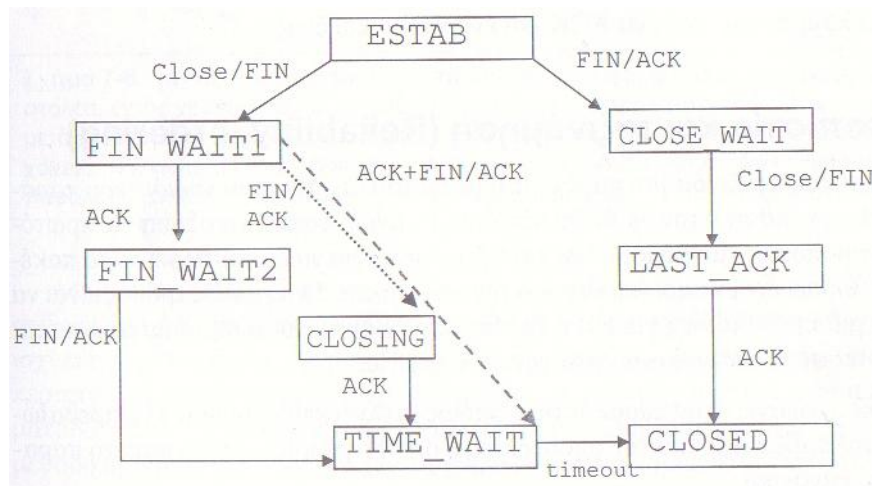


Σχήμα 2.3 Διάγραμμα μετάβασης καταστάσεων

Η αρχική επικοινωνία γίνεται με έναν αλγόριθμο που ονομάζεται 3-way handshake (τρίδρομη ή τριπλή χειραψία). Όπως φαίνεται από το σχήμα 2.2, 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 την μεταφορά των δεδομένων. Το νούμερο αυτό επιλέγεται τυχαία, 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.

2.1.1.3. 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 2.4). Όταν την παραλάβει, περνάει στην κατάσταση FIN_WAIT, όπου αναμένει την 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 2.4. Διάγραμμα μετάβασης καταστάσεων για τον τερματισμό σύνδεσης με το TCP

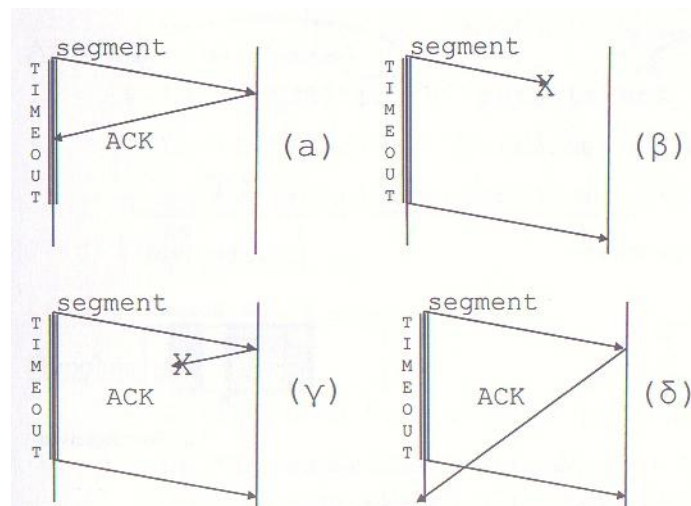
2.1.1.4. Αξιοπιστία και ταξινόμηση (Reliability, Ordering)

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 μιας 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 μπουν 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 2.5 φαίνονται και σε γραφική αναπαράσταση τα τέσσερα αυτά πιθανά ενδεχόμενα κατά την διάρκεια μιας επικοινωνίας.



Σχήμα 2.5. Τέσσερα ενδεχόμενα κατά την επικοινωνία

Για να δημιουργήσουμε ένα μηχανισμό αξιοπιστίας σε ένα πρωτόκολλα χρειαζόμαστε:

- α.** Στην πρώτη περίπτωση μόνο επιβεβαίωση.
- β.** Στην δεύτερη περίπτωση έναν μηχανισμό χρονικού περιθωρίου που θα σηματοδοτήσει την επαναποστολή.
- γ.** Στην τρίτη περίπτωση έναν 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.

2.1.1.5. 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 τον 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:

$$\text{EstimatedRTT} = a \times \text{EstimatedRTT} + b \times \text{SampleRTT}$$

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 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.

2.1.1.6. 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.

2.1.2. 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 την 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.

2.1.2.1. Έλεγχος ροής (Flow Control)

Η αξιοπιστία και η ταξινόμηση στο TCP γίνονται μέσω του μηχανισμού του ολισθαίνοντος 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 τον ρυθμό με τον οποίο διαβάζει τα δεδομένα ο παραλήπτης από τον buffer του

συμβαίνει 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.

2.1.2.2. 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 χωρητικότητων των εισερχόμενων ζεύξεων ξεπερνάει την 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 δημιουργείται η ουρά μέσα στον buffer του κόμβου. Ο buffer μπορεί τελικά να υπερχειλίσει και να αρχίσει να «ρίχνει» τα εισερχόμενα πακέτα. Αν οι αποστολείς δεν μειώσουν την ταχύτητα με την οποία στέλνουν δεδομένα, τότε έχουμε μια 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 ούτε τη δυνατότητα, ούτε 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 την 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 μεταδώσει τέτοια ποσότητα δεδομένων, ώστε να αποφύγει και την υπερφόρτωση του παραλήπτη και την 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.

2.1.2.3. Κριτήρια κατάστασης ελέγχου συμμόρφωσης

Για την αξιολόγηση των καταστάσεων ελέγχου συμμόρφωσης 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 μια απλή σταθερή κατάσταση. Αντ' αυτού, το σύστημα φτάνει σε μια «δυναμική ισορροπία» (equilibrium) στην οποία ταλαντώνεται γύρω από μια βέλτιστη

κατάσταση. Όσο μικρότερο είναι το μέγεθος αυτής της ταλάντωσης, τόσο πιο ομαλό είναι το ποσοστό αποστολής.

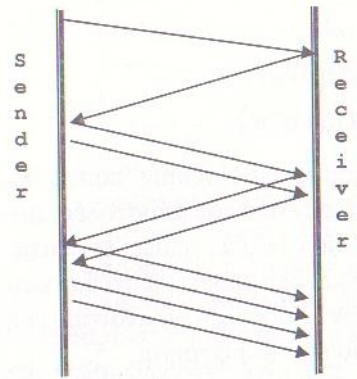
Καθυστέρηση Ουράς - Queuing Delay: Αν η ουρά (queue) μέσα στον buffer του κόμβου είναι συνεχώς άδεια, τότε δε γίνεται η καλύτερη δυνατή εκμετάλλευση της ζεύξης. Το bottleneck link είναι πλήρως εκμεταλλεύσιμο όταν η ουρά στον buffer του κόμβου δεν είναι άδεια. Από την άλλη, ένα μεγάλο μήκος ουράς σημαίνει 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.

2.1.2.4. 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 τον 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 έναν 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 2.6 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. Μια 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 μια αλλαγή τακτικής. Το κατώφλι αυτό είναι το γνωστό `slow start threshold (sssthresh)` και σημαίνει το τέλος της εκθετικής αύξησης και την αρχή μιας 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 μια 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, καθώς 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 μια σύνδεση και άρχισαν να χρησιμοποιούν το δίκτυο επικοινωνίας.

Όλες αυτές οι αμφιβολίες και οι αβεβαιότητες οδηγούν στην επιλογή μιας συντηρητικής στρατηγικής, έτσι ώστε να ελεγχθεί η συμφόρηση και να αποφευχθεί μια κατάσταση καταστροφικής συμφόρησης (`congestive collapse`). Η αρχική έκδοση του TCP, το Old Tahoe, ακολουθούσε στρατηγική υποχώρησης στο αρχικό `cwnd`, με ταυτόχρονη μείωση του `sssthresh` 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 επιπλέον μηχανισμοί που χρησιμοποιούν το TCP Tahoe και το TCP Reno είναι οι Fast Retransmit και Fast Recovery.

2.1.2.5. Μηχανισμοί Γρήγορης Επαναμετάδοσης και Γρήγορης Ανάκαμψης

Ο μηχανισμός γρήγορης επαναμετάδοσης (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 γρήγορης ανάκαμψης (Fast Recovery). Αυτός στηρίζεται στη λογική της «περιορισμένης συμφόρησης», η οποία δικαιολογεί μια 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 μια ακραία στρατηγική, όπως η ρύθμιση του παραθύρου στην αρχική του τιμή. Έτσι, ο μηχανισμός Fast Recovery

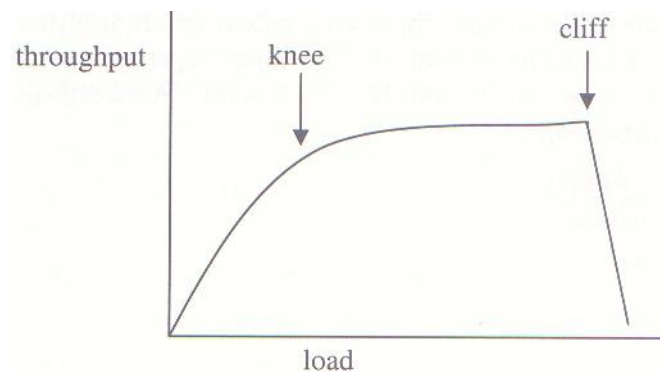
δεν ρυθμίζει το παράθυρο στην αρχική του τιμή, αλλά στο μισό της προηγούμενης τιμής του. Ομοίως υποχωρεί και το `ssthresh`, κάτι που αναγκάζει τον αποστολέα να περάσει αμέσως στη φάση της προσθετικής αύξησης, σε αντίθεση με τον αρχικό μηχανισμό που ρυθμίζει το `ssthresh` στη μισή της προηγούμενης τιμής 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 τον μηχανισμό Fast Retransmit ονομάζεται TCP Tahoe [8]. Η έκδοση που εκτός από τον Fast Retransmit διαθέτει και τον Fast Recovery ονομάζεται TCP Reno [9]. Όλες αυτές οι εκδόσεις συνυπάρχουν στο Διαδίκτυο και, φυσικά, μπορούν να επικοινωνούν μεταξύ τους. Αυτό συμβαίνει επειδή ο έλεγχος συμφόρησης είναι μια 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.

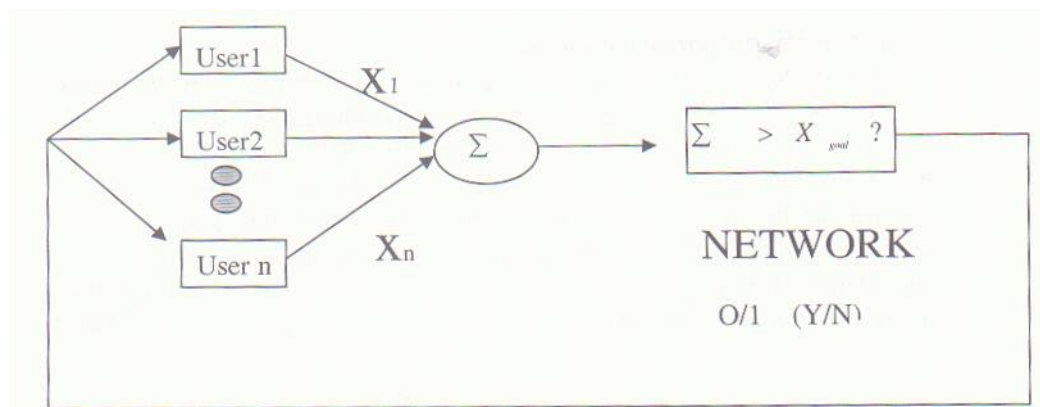
2.1.2.6. Το Μοντέλο Ανάλυσης για τον Έλεγχο Συμφόρησης

Στην ενότητα αυτή δίνεται μια περιγραφή της συμπεριφοράς του throughput ως συνάρτηση του φορτίου που υπάρχει στο δίκτυο, όπως φαίνεται στο Σχήμα 2.7. 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 έχει 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 τον χρόνο απόκρισης.



Σχήμα 2.7. Το throughput ως Συνάρτηση του Φορτίου του δικτύου



Σχήμα 2.8. Το Μοντέλο του Συστήματος Ελέγχου

Στο Σχήμα 2.8 παρουσιάζεται ένα μοντέλο συστήματος ελέγχου η χρηστών οι οποίοι μοιράζονται το δίκτυο. Εδώ υποτίθεται μια λειτουργία διακριτού χρόνου με τον 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 μια 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.

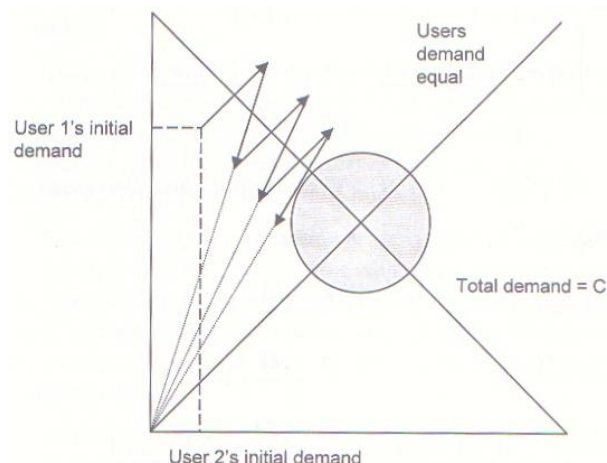
Ένα μειονέκτημα αυτού του μοντέλου είναι ότι υποθέτει ίσα RTT για όλα τα flows, κάτι το οποίο δεν ισχύει σε πραγματικά συστήματα.

2.1.2.7. 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 μια 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 μια 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 2.9 βλέπουμε τους άξονες x και y των χρηστών 1 και 2 αντίστοιχα. Οι χρήστες χρησιμοποιούν διαφορετικά παράθυρα, γεγονός που σημαίνει ότι ο πρώτος χρήστης με το μεγαλύτερο παράθυρο ξεκίνησε νωρίτερα την επικοινωνία μέσω του δικτύου και το παράθυρο του μεγάλωσε κατά την 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 ξεπερνούμε 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, βρισκόμενοι όμως πάνω στον άξονα της δικαιοσύνης. Θέλουμε, δηλαδή, να κινηθούμε περίπου στο εσωτερικό του σκιασμένου κύκλου του σχήματος 2.9.



Σχήμα 2.9. 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 (adwnd) 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 την 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 .

2.1.3. Εκδόσεις του πρωτοκόλλου μεταφοράς TCP

2.1.3.1. 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 (sssthresh) 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 sssthresh 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 τον 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.

2.1.3.2. 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.

2.1.3.3. TCP New Reno

Το TCP New Reno [9] είναι μια τροποποίηση του TCP Reno που 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.

2.1.3.4. 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 ξανασταλούν. 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 ξανασταλεί, 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.

2.1.3.5. 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.

2.1.3.6. TCP Real

Το TCP Real [16, 17] είναι ένα πειραματικό πρωτόκολλο το οποίο αποτελεί μια προσέγγιση που βασίζεται στον παραλήπτη (receiver - oriented). Το πρωτόκολλο διαθέτει μια στρατηγική μετάδοσης δεδομένων βασιζόμενη 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 μια 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, έτσι, μπορεί να επιλέξει κάθε φορά μια στρατηγική ανάκαμψης η οποία βασίζεται στην 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 μια συντηρητική στρατηγική ανάκαμψης. Στην αντίθετη περίπτωση, όπου καλλιεργείται συμφόρηση στο δίκτυο, ο παραλήπτης ενημερώνει τον αποστολέα να μειώσει το ρυθμό αποστολής δεδομένων.

2.1.4. Ενεργός Διαχείριση της Ουράς - Active Queue Management

Παρόλο που ο έλεγχος συμφόρησης του TCP είναι απαραίτητος και καλά σχεδιασμένος, δεν είναι ικανός να παρέχει ικανοποιητικές υπηρεσίες κάτω από όλες τις συνθήκες. Η Ενεργός Διαχείριση της Ουράς - Active Queue Management (AQM) προτάθηκε για να δώσει μια λύση σε περίπτωση συμφόρησης στην ουρά του bottleneck.

2.1.4.1. DropTail

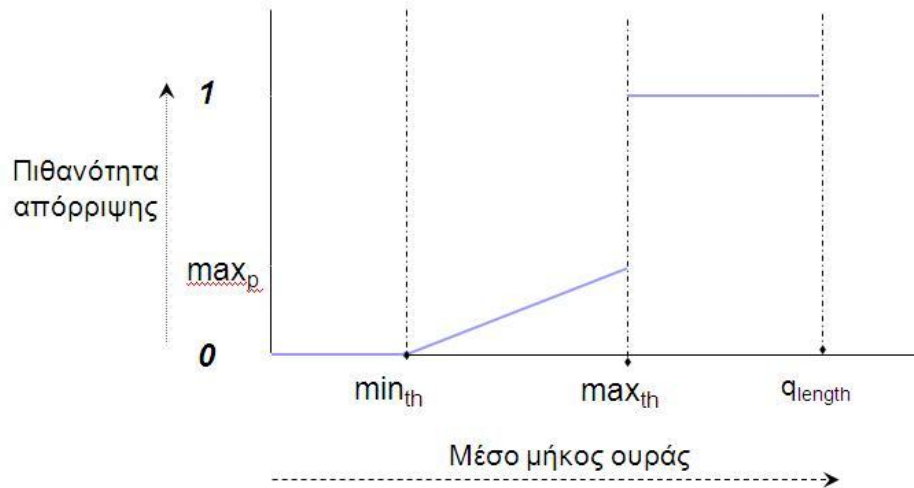
Ο πιο τυπικός αλγόριθμος που χρησιμοποιείται σήμερα στην διαχείριση των ουρών είναι ο αλγόριθμος DropTail ή αλλιώς FIFO (First In, First Out). Δηλαδή, όπως φαίνεται και από το όνομα του αλγορίθμου, τα πρώτα πακέτα που απορρίπτονται σε περίπτωση συμφόρησης είναι αυτά που εισήλθαν στην ουρά πρώτα. Ο απλός DropTail buffer στους δρομολογητές έχει δύο σημαντικά μειονεκτήματα:

- Σε ορισμένες περιπτώσεις ο droptail buffer επιτρέπει σε μια μόνο σύνδεση ή σε λίγα flows να μονοπωλούν την ουρά, μην αφήνοντας άλλες συνδέσεις να χρησιμοποιήσουν τους πόρους του buffer.
- Η στρατηγική του DropTail τείνει να κρατάει την ουρά σχεδόν γεμάτη για μεγάλο χρονικό διάστημα, αφού αρχίζει να ρίχνει πακέτα μόνο όταν η ουρά υπερχειλίζει. Με αυτόν τον τρόπο δημιουργείται μεγάλο queuing delay, κάτι που επηρεάζει αρνητικά τις διαλογικές εφαρμογές όπως το Telnet ή κάποιες μεταφορές Web. Συνήθως το μέγεθος του buffer του bottleneck είναι αρκετά μεγάλο, ώστε να αποφεύγεται η απώλεια δεδομένων όταν αυτά φτάνουν σε μεγάλα ποσά. Εντούτοις, είναι επιθυμητό το μέσο μήκος της ουράς να είναι αρκετά μικρότερο από το μέγεθος του buffer.

Το AQM καλείται να ξεπεράσει αυτά τα δύο μειονεκτήματα. Ο μηχανισμός Random Early Detection (RED) ρίχνει πακέτα τυχαία πριν γεμίσει η ουρά του buffer. Η παραμετροποίηση του RED δεν είναι προκαθορισμένη.

2.1.4.2. Random Early Drop (RED)

Ο μηχανισμός RED [18, 19] χρησιμοποιείται για την αποφυγή συμφόρησης σε δίκτυα μεταγωγής πακέτων και υλοποιείται στον δρομολογητή. Ο δρομολογητής ανιχνεύει ανερχόμενη συμφόρηση υπολογίζοντας τον μέσο όρο του μεγέθους της ουράς. Ο μέσος όρος υπολογίζεται χρησιμοποιώντας ένα χαμηλοπερατό φίλτρο με έναν εκθετικού βάρους κινούμενο μέσο όρο. Ο μέσος όρος του μεγέθους της ουράς συγκρίνεται με δύο κατώφλια, ένα ελάχιστο και ένα μέγιστο. Όταν ο μέσος όρος του μεγέθους της ουράς είναι μικρότερος από το ελάχιστο κατώφλι, κανένα πακέτο δε σημειώνεται. Όταν ο μέσος όρος του μεγέθους της ουράς είναι μεγαλύτερος από το μέγιστο κατώφλι, τότε κάθε πακέτο που φτάνει στον δρομολογητή σημειώνεται-μαρκάρεται. Αν τα μαρκκαρισμένα πακέτα ρίχνονται από την ουρά του δρομολογητή ή αν όλοι οι αποστολείς είναι υποστηρίζουν αυτό το μηχανισμό, τότε ο μέσος όρος του μεγέθους της ουράς δεν θα ξεπεράσει το μέγιστο κατώφλι. Όταν ο μέσος όρος του μεγέθους της ουράς είναι μεταξύ του ελάχιστου και του μέγιστου κατωφλίου, τότε κάθε πακέτο που φτάνει στον δρομολογητή σημειώνεται-μαρκάρεται με μια πιθανότητα p_a , που p_a είναι συνάρτηση του μέσου όρου του μεγέθους της ουράς avg . Κάθε φορά που ένα πακέτο μαρκάρεται, η πιθανότητα το πακέτο να προέρχεται από μια συγκεκριμένη σύνδεση είναι ευθέως ανάλογη προς το ποσοστό του εύρους ζώνης που καταλαμβάνει η συγκεκριμένη σύνδεση στο δρομολογητή. Ο μηχανισμός RED αποτελείται από δύο διαφορετικούς αλγόριθμους. Ο ένας είναι ο αλγόριθμος που υπολογίζει το μέσο όρο του μεγέθους της ουράς και καθορίζει το πόσο ανεκτικός θα είναι ο δρομολογητής στο ρυθμό παραλαβής νέων πακέτων. Ο δεύτερος είναι ο αλγόριθμος που υπολογίζει την πιθανότητα μαρκκαρίσματος των πακέτων, δηλαδή πόσο συχνά ο δρομολογητής θα μαρκάρει τα εισερχόμενα πακέτα, ανάλογα με το επίπεδο της συμφόρησης την συγκεκριμένη στιγμή. Ο στόχος για το δρομολογητή είναι να μαρκάρει εισερχόμενα πακέτα με σταθερό ρυθμό, δηλαδή με σταθερό κενό ανάμεσα σε δύο μαρκκαρίσματα, έτσι ώστε να αποφύγει τις «προκαταλήψεις» προς ορισμένες συνδέσεις και να μαρκάρει πακέτα τόσο συχνά, όσο χρειάζεται για να ελέγχει το μέγεθος της ουράς.



Σχήμα 2.10. Πιθανότητα απόρριψης πακέτων για αυξανόμενο μέσο μήκος ουράς για τον αλγόριθμο διαχείρισης ουράς RED

2.2. User Datagram Protocol (UDP)

Το πρωτόκολλο μεταφοράς UDP [20] αποτελεί ένα από τα κύρια μέρη του σημερινού Διαδικτύου και παρουσιάζει την πιο απλή λειτουργία σε σχέση με τα υπόλοιπα πρωτόκολλα που λειτουργούν στο επίπεδο μεταφοράς, χωρίς να εγγυάται αξιοπιστία παράδοσης δεδομένων. Τα πακέτα UDP που αποστέλλονται από έναν υπολογιστή μπορεί να φτάσουν στον παραλήπτη με λάθος σειρά, διπλά ή να μην φτάσουν καθόλου, εάν το δίκτυο έχει μεγάλο φόρτο. Η λειτουργία του επικεντρώνεται κυρίως στην επικοινωνία διεργασιών που ανήκουν σε διαφορετικούς υπολογιστές. Το πρωτόκολλο UDP, λοιπόν, προσθέτει στα δεδομένα έναν αποπολυπλέκτη, ο οποίος επιτρέπει σε διαφορετικές διεργασίες εφαρμογών κάποιου υπολογιστή να μοιραστούν το δίκτυο.

Ουσιαστικά, το πρωτόκολλο UDP παραλαμβάνει μηνύματα από τις διεργασίες, προσθέτει σε αυτά τα πεδία των θυρών αποστολής και προορισμού για την πολυπλεξία και την αποπολυπλεξία, προσθέτει δύο ακόμη πεδία και στέλνει το «πακέτο» που δημιουργείται στο επίπεδο δικτύου. Το επίπεδο δικτύου ενθυλακώνει το πακέτο σε ένα αυτοδύναμο πακέτο IP και προσπαθεί, χρησιμοποιώντας υπηρεσίες best effort, να το παραδώσει στον προορισμό του. Αν το πακέτο φτάσει στον προορισμό του, το πρωτόκολλο UDP χρησιμοποιεί τις θύρες αποστολής και προορισμού για την παράδοση των δεδομένων στη σωστή εφαρμογή. Προφανώς, το πρωτόκολλο UDP δεν απαιτεί εγκαθίδρυση σύνδεσης ανάμεσα στον αποστολέα και τον παραλήπτη πριν την αποστολή δεδομένων. Για το λόγο αυτό, το πρωτόκολλο UDP χαρακτηρίζεται ως πρωτόκολλο που δεν απαιτεί σύνδεση (connectionless).

2.2.1. Χαρακτηριστικά UDP

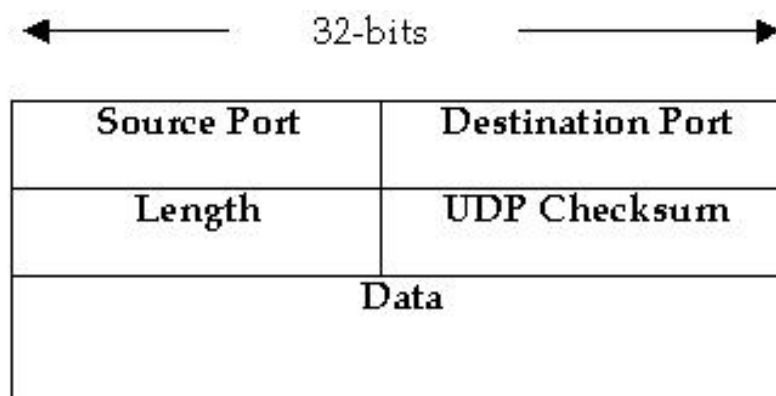
Η λογική σχεδιασμού του UDP, ως πρωτοκόλλου μηδενικής λειτουργικότητας, δεν είναι τυχαία και ο λόγος είναι η ανάγκη ύπαρξης τόσο από την πλευρά διάφορων εφαρμογών, όσο και από την πλευρά του προγραμματιστή ενός πρωτοκόλλου για γρήγορη μετάδοση δεδομένων χωρίς αξιοπιστία μεταφοράς. Έτσι, παρέχει μια μέθοδο αποστολής δεδομένων ανάμεσα σε προγράμματα εφαρμογής, παρέχοντας

τους ελάχιστους δυνατούς μηχανισμούς. Το πρωτόκολλο UDP δίνει την δυνατότητα επιλεκτικού σχεδιασμού πρωτοκόλλων, όταν η λειτουργικότητα του TCP είναι περιορισμένη.

Το πρωτόκολλο UDP δεν χρησιμοποιεί καμία μορφή ελέγχου συμφόρησης. Όμως, έλεγχος συμφόρησης απαιτείται για την αποφυγή της εισαγωγής του δικτύου σε κατάσταση συμφόρησης, όπου η απόδοση μειώνεται κατακόρυφα. Στην περίπτωση που όλοι οι χρήστες μετέδιδαν ένα βίντεο υψηλού ρυθμού χωρίς να χρησιμοποιούν κάποια μορφή ελέγχου συμφόρησης, οι δρομολογητές θα υπερχείλιζαν και κανείς δεν θα μπορούσε να παρακολουθήσει βίντεο. Έτσι, η έλλειψη ελέγχου συμφόρησης στο πρωτόκολλο UDP μπορεί να οδηγήσει σε σημαντικά προβλήματα.

2.2.1.1. Επικεφαλίδα

Η επικεφαλίδα του UDP, όπως φαίνεται στην παρακάτω εικόνα, περιλαμβάνει ελάχιστα πεδία.



Σχήμα 2.11 Επικεφαλίδα UDP

Από τα τέσσερα πεδία της επικεφαλίδας υποχρεωτικά είναι μόνο τα Destination Port και Length.

Το πεδίο **Source Port** αντιπροσωπεύει την θύρα αποστολής και όταν χρειάζεται θα πρέπει να θεωρηθεί ως η θύρα στην οποία θα σταλεί απάντηση. Αν δεν χρησιμοποιηθεί η τιμή που παίρνει είναι το 0.

Το πεδίο **Destination Port** αντιπροσωπεύει την θύρα προορισμού και είναι υποχρεωτικό πεδίο.

Ο συνδυασμός Source-Destination Port αντιπροσωπεύει τον αποπολυπλέκτη της δικτυακής εφαρμογής. Συνήθως η θύρα, δηλαδή η διεργασία που περιμένει να εξυπηρετήσει κάποιο χρήστη, είναι προκαθορισμένη, τουλάχιστον στο αρχικό στάδιο της επικοινωνίας. Η θύρα μπορεί να υλοποιηθεί στο λειτουργικό σύστημα ως ουρά μηνυμάτων. Κάποιες εφαρμογές χρησιμοποιούν στατικές θύρες αποστολής και προορισμού, αποκλειστικά χρησιμοποιούμενες από αυτές, ενώ άλλες εφαρμογές χρησιμοποιούν δυναμικές θύρες. Λόγω του μεγέθους των κεφαλίδων των θυρών (2 Bytes), οι επιλέξιμοι αριθμοί θυρών κυμαίνονται από 0 μέχρι 65535· τιμές μεγαλύτερες από 49151 αναπαριστούν δυναμικές θύρες.

Στον πίνακα που ακολουθεί δίνονται κάποιες από τις πιο γνωστές θύρες UDP που χρησιμοποιούνται σήμερα στο Διαδίκτυο.

Θύρα	Όνομα	Περιγραφή
7	Echo	Echo input back to server
11	Users	System statistics – active users
13	Daytime	Time of day (ASCII)
17	Quote	Quote of the day
19	Chargen	Character generator
37	Time	System time (seconds since 1970)
43	Nickname	Who is
67	Bootps	BOOTP or DHCP Server
68	Bootpc	BOOTP or DHCP Client
88	Kerberow	Kerberos security service
123	Ntp	Network Time Protocol (NTP)
161	Snmp	Simple Network Management Protocol (SNMP)

Πίνακας 2.1 Γνωστές θύρες UDP

Το πεδίο **Length** έχει μέγεθος 16-bit και προσδιορίζει το μέγεθος σε bytes όλου του πακέτου header + Data. Το ελάχιστο μέγεθος του μπορεί να είναι 8-byte, αφού αυτό είναι το μέγεθος του header.

Το πεδίο **Checksum** υπολογίζει το άθροισμα ελέγχου της επικεφαλίδας του πρωτοκόλλου, του περιεχομένου του πεδίου δεδομένων (data) και επιπλέον της ψευδοεπικεφαλίδας (pseudoheader) που απαρτίζεται από τα πεδία length, IP Source Address και IP destination Address του πρωτοκόλλου IP. Η χρήση των πεδίων αυτών αποτρέπει την παράδοση μηνυμάτων σε λάθος διεύθυνση IP, σε περίπτωση που η διεύθυνση αυτή είχε αλλάξει κατά την διάρκεια της επικοινωνίας.

2.2.1.2. Εφαρμογές

Παρότι η κίνηση στο Διαδίκτυο που αποδίδεται στο πρωτόκολλο UDP αποτελεί ένα πολύ μικρό μέρος της συνολικής κίνησης του Διαδικτύου, αρκετές θεμελιώδεις εφαρμογές του Διαδικτύου κάνουν χρήση του, συμπεριλαμβανομένων των: Domain Name System (DNS), Simple Network Management Protocol (SNMP), Dynamic Host Configuration Protocol (DHCP) και το Routing Information Protocol (RIP).

Για παράδειγμα, όταν μια εφαρμογή DNS θέλει να κάνει μια αίτηση, δημιουργεί ένα DNS μήνυμα αίτησης και το στέλνει στη UDP socket. Χωρίς εκτέλεση εγκαθίδρυσης σύνδεσης, το UDP προσθέτει επικεφαλίδες στο μήνυμα και το στέλνει στο επίπεδο δικτύου. Το επίπεδο δικτύου ενθυλακώνει το πακέτο UDP σε ένα αυτοδύναμο πακέτο, το οποίο στέλνει στον διακομιστή ονομάτων (name server). Η εφαρμογή DNS του κόμβου που έκανε αίτηση περιμένει μέχρι να λάβει απάντηση. Αν δεν παραλάβει κάποια απάντηση (πιθανόν το πρωτόκολλο UDP έχασε την αίτηση ή την απάντηση), προσπαθεί να στείλει την αίτηση σε έναν άλλο διακομιστή ονομάτων ή ενημερώνει την εφαρμογή ότι δεν έλαβε απάντηση.

Οι εφαρμογές audio και video streaming χρησιμοποιούν κατά κόρον πακέτα UDP. Για τις εφαρμογές αυτές είναι πολύ σημαντικό τα πακέτα να παραδοθούν στον παραλήπτη σε σύντομο χρονικό διάστημα, ούτως ώστε να μην υπάρχει διακοπή στην ροή του ήχου ή της εικόνας. Κατά συνέπεια, προτιμάται το πρωτόκολλο UDP διότι είναι αρκετά γρήγορο, παρόλο που υπάρχει η πιθανότητα μερικά πακέτα UDP να χαθούν. Στην περίπτωση που χαθεί κάποιο πακέτο, οι εφαρμογές αυτές διαθέτουν ειδικούς μηχανισμούς διόρθωσης και παρεμβολής, ούτως ώστε ο τελικός χρήστης να μην παρατηρεί καμία αλλοίωση ή διακοπή στην ροή του ήχου και της

εικόνας λόγω του χαμένου πακέτου. Κλασικό παράδειγμα προγραμμάτων που χρησιμοποιούν το πρωτόκολλο UDP είναι οι εφαρμογές πραγματικού χρόνου πχ. media streaming, παιχνίδια στο Διαδίκτυο, [VoIP](#) κτλ. Σε αντίθεση με το πρωτόκολλο TCP, το UDP υποστηρίζει broadcasting, δηλαδή την αποστολή ενός πακέτου σε όλους τους υπολογιστές ενός δικτύου, και multicasting, δηλαδή την αποστολή ενός πακέτου σε κάποιους συγκεκριμένους υπολογιστές ενός δικτύου. Η τελευταία δυνατότητα χρησιμοποιείται πολύ συχνά στις εφαρμογές audio και video streaming έτσι ώστε μια ροή ήχου ή εικόνας να μεταδίδεται ταυτόχρονα σε πολλούς συνδρομητές

2.2.2. Πλεονεκτήματα του UDP

Το πρωτόκολλο UDP έχει κάποια πλεονεκτήματα σε σχέση με το TCP με ένα από αυτά, όπως αναφέραμε ήδη, την δυνατότητα που παρέχει στους προγραμματιστές να χρησιμοποιούν επιλεκτικά τις υπηρεσίες που χρειάζονται και να αποφεύγουν, έτσι, υπηρεσίες όπως η διάταξη πακέτων και ο έλεγχος συμφόρησης. Επίσης, είναι προφανές ότι η επεξεργασία που χρειάζεται η επικεφαλίδα του UDP είναι ελάχιστη, καθιστώντας το ένα γρήγορο πρωτόκολλο επικοινωνίας. Σε αυτό τον ισχυρισμό συνεισφέρει και η παρατήρηση ότι το UDP μειώνει το συνολικό χρόνο επικοινωνίας, αφού δεν χρειάζεται σύνδεση, ούτε και περιορίζεται από την προοδευτική αύξηση της ροής του TCP. Επιπλέον, το UDP απαιτεί μικρότερο διαχειριστικό κόστος, αφού έχει μικρότερη επικεφαλίδα από το TCP. Βέβαια, γίνεται αντιληπτό ότι όλα τα παραπάνω πλεονεκτήματα του UDP αποβαίνουν σε βάρος της λειτουργικότητας του.

2.3. Σύγκριση των πρωτοκόλλων TCP και UDP

Ο πίνακας που ακολουθεί περιλαμβάνει τα σημαντικότερα χαρακτηριστικά των πρωτοκόλλων μεταφοράς TCP και UDP, με τέτοιο τρόπο ώστε να γίνονται εύκολα αντιληπτές οι διαφορές τους, όσον αφορά ένα πλήθος χαρακτηριστικών.

	TCP	UDP
Εγκαθίδρυση σύνδεσης	Απαιτείται και καταναλώνει χρόνο	Δεν απαιτείται
Εγγυημένη μετάδοση μηνυμάτων	Επιστρέφει επιβεβαιώσεις (ACKs)	Δεν επιστρέφει ACKs και, συνεπώς, ο παραλήπτης δεν μπορεί να ενημερώσει σχετικά με το ποια πακέτα έχουν παραδοθεί επιτυχώς. Τα χαμένα πακέτα δεν επαναμεταδίδονται
Σειρά πακέτων	Τα πακέτα θα φτάσουν στον παραλήπτη με την ίδια σειρά με την οποία στάλθηκαν. Στην περίπτωση που λείπει ένα πακέτο και έρθουν επόμενα πακέτα, τότε αυτά κατακρατούνται στην προσωρινή μνήμη μέχρι να φτάσει το πακέτο που λείπει, οπότε και αναδιατάσσονται και εμφανίζονται με την σωστή σειρά στον παραλήπτη	Τα πακέτα UDP δεν αριθμούνται και, κατά συνέπεια, δεν υπάρχει συγκεκριμένη σειρά με την οποία θα πρέπει να φτάσουν στον παραλήπτη
Έλεγχος ροής	Ο παραλήπτης μπορεί να ενημερώσει τον αποστολέα να μειώσει το ρυθμό του	Δεν υπάρχει
Έλεγχος συμφόρησης	Οι επιβεβαιώσεις παρέχουν δυνατότητα ελέγχου της συμπεριφοράς των αποστολέων	Δεν υπάρχει, λόγω έλλειψης επιβεβαιώσεων
Ταχύτητα μετάδοσης δεδομένων	Το πρωτόκολλο TCP θεωρείται αργό, δεδομένου του γεγονότος ότι χρειάζονται τουλάχιστον 3 πακέτα για την εγκαθίδρυση της σύνδεσης, πριν ακόμη μεταδοθεί οποιοδήποτε πακέτο δεδομένων. Επίσης, οι μηχανισμοί αξιοπιστίας που υλοποιεί έχουν σημαντικό αντίκτυπο στην ταχύτητα μετάδοσης δεδομένων	Το πρωτόκολλο UDP είναι πολύ γρήγορο, διότι δεν εφαρμόζει μηχανισμούς αξιοπιστίας επικοινωνίας.

Πίνακας 2.2 Σύγκριση πρωτοκόλλων μεταφοράς TCP και UDP

Ο πίνακας που ακολουθεί περιλαμβάνει τις πιο γνωστές δικτυακές εφαρμογές και το πρωτόκολλο μεταφοράς που χρησιμοποιεί η καθεμία.

Εφαρμογή	Πρωτόκολλο Επιπέδου Εφαρμογής	Πρωτόκολλο Επιπέδου Μεταφοράς
Ηλεκτρονικό ταχυδρομείο	SMTP	TCP
Απομακρυσμένη σύνδεση	Telnet	TCP
Παγκόσμιος Ιστός	HTTP	TCP
Μεταφορά αρχείων	FTP	TCP
Απομακρυσμένος διακομιστής αρχείων	NFS	Τυπικά UDP
Μετάδοση πολυμέσων	-	Τυπικά UDP
Τηλεφωνία μέσω Διαδικτύου	-	Τυπικά UDP
Διαχείριση δικτύου	SNMP	Τυπικά UDP
Πρωτόκολλα δρομολόγησης	RIP	Τυπικά UDP
Ονοματολογία	DNS	Τυπικά UDP

Πίνακας 2.3 Εφαρμογές και αντίστοιχο πρωτόκολλο μεταφοράς

Όπως είναι αναμενόμενο, εφαρμογές που απαιτούν αξιόπιστη μετάδοση δεδομένων, όπως είναι το e-mail, ο Παγκόσμιος Ιστός και η απομακρυσμένη σύνδεση σε έναν υπολογιστή, χρησιμοποιούν το αξιόπιστο πρωτόκολλο μεταφοράς TCP. Υπάρχουν, όμως, πολλές σημαντικές εφαρμογές που χρησιμοποιούν το μη αξιόπιστο μεταφοράς UDP. Για παράδειγμα, οι ενημερώσεις των πινάκων δρομολόγησης RIP χρησιμοποιούν το UDP, αφού ενημερώσεις αποστέλλονται περιοδικά και, στην περίπτωση που κάποιες χαθούν, θα αντικατασταθούν από νεότερες.

Κεφάλαιο 3

Πρωτόκολλα σύνδεσης

Τα πρωτόκολλα επιπέδου σύνδεσης είναι υπεύθυνα για τη μετάδοση δεδομένων ανάμεσα σε γειτονικούς κόμβους σε ένα δίκτυο ευρείας ζώνης ή ανάμεσα σε κόμβους του ίδιου τοπικού δικτύου, παρέχοντας ταυτόχρονα δυνατότητα ανίχνευσης και διόρθωσης σφαλμάτων που συμβαίνουν στο φυσικό επίπεδο. Το επίπεδο σύνδεσης ασχολείται με την τοπική παράδοση των πλαισίων (frames) ανάμεσα σε συσκευές του ίδιου δικτύου. Τόσο η δρομολόγηση, όσο η διευθυνσιοδότηση είναι λειτουργίες υψηλότερου επιπέδου, επιτρέποντας στα πρωτόκολλα επιπέδου σύνδεσης να συγκεντρωθούν στην τοπική παράδοση. Όταν δύο συσκευές χρησιμοποιούν το ίδιο μέσο ταυτόχρονα, είναι πιθανό να συμβούν συγκρούσεις πλαισίων. Τα πρωτόκολλα σύνδεσης προσδιορίζουν πως οι συσκευές ανιχνεύουν και ανακάμπτουν από τις συγκρούσεις αυτές. Έτσι, το επίπεδο σύνδεσης μπορεί να παρέχει αξιόπιστη ή μη αξιόπιστη μετάδοση δεδομένων. Στο κεφάλαιο αυτό περιγράφονται τρία από τα πιο σημαντικά αξιόπιστα πρωτόκολλα σύνδεσης: το πρωτόκολλο Snoop, το Automatic Repeat Request και το Forward Error Correction.

3.1. Πρωτόκολλο Snoop

Το πρωτόκολλο Snoop [21, 22, 23] αποτελεί μια αξιόπιστη λύση στο πρόβλημα της μείωσης του παραθύρου συμφόρησης του TCP, εξαιτίας υψηλών ρυθμών σφαλμάτων (BER) σε ασύρματο περιβάλλον. Αυτό το πετυχαίνει διατηρώντας ταυτόχρονα την από-άκρο-σε-άκρο σημασιολογία του TCP. Ένα τυπικό σενάριο εφαρμογής του πρωτοκόλλου Snoop φαίνεται στο σχήμα 3.1.



Σχήμα 3.1 Τοπολογία εφαρμογής του πρωτοκόλλου Snooper

Το Snooper εκτελεί τοπικές επαναμεταδόσεις χρησιμοποιώντας καταχωρητές επιπέδου σύνδεσης στο σταθμό βάση, από όπου περνούν όλα τα πακέτα που μεταδίδονται από το ενσύρματο προς το ασύρματο κανάλι. Έτσι, μόνο το δικτυακό λογισμικό στο σταθμό βάση χρειάζεται τροποποιήσεις. Ένας πράκτορας Snooper τοποθετείται σε ένα σταθμό βάση, ο οποίος έχει ως κύριες λειτουργίες την ανίχνευση απωλειών και την ανάκαμψη μέσω επαναμεταδόσεων. Η βασική ιδέα είναι ότι ο πράκτορας Snooper αποθηκεύει προσωρινά όλα τα μη επιβεβαιωμένα δεδομένα TCP και εκτελεί τοπικές επαναμεταδόσεις με βάση στρατηγικές, οι οποίες εξαρτώνται από τις TCP επιβεβαιώσεις από τον κινητό κόμβο και τις λήξεις των χρονικών περιθωρίων. Ο πράκτορας Snooper αποκρύπτει τις διπλές επιβεβαιώσεις από τον αποστολέα προς τον παραλήπτη και εκτελεί επαναμεταδόσεις των χαμένων πακέτων. Έτσι, χρησιμοποιώντας διπλές επιβεβαιώσεις για την ανίχνευση απωλειών πακέτων και εκτελώντας τοπικές επαναμεταδόσεις μόλις ανιχνευθεί απώλεια, ο πράκτορας Snooper προστατεύει τον αποστολέα από τα ιδιαίτερα χαρακτηριστικά του ασύρματου δικτύου.

Τα πακέτα που παραδίδονται στο σταθμό βάση και έχουν ως προορισμό τον κινητό κόμβο, περνούν μέσω του πράκτορα Snooper ανάμεσα στο επίπεδο σύνδεσης και το επίπεδο IP. Ο πράκτορας Snooper διατηρεί αντίγραφα των πακέτων που στάλθηκαν από τον αποστολέα και έχουν προωθηθεί, αλλά δεν έχουν ακόμη επιβεβαιωθεί από τον παραλήπτη. Όταν ένα νέο πακέτο φτάνει από τον αποστολέα, ο πράκτορας το προσθέτει στη προσωρινό χώρο αποθήκευσης που διαθέτει και το προωθεί κανονικά. Ο πράκτορας Snooper αδειάζει το χώρο αποθήκευσης του χρησιμοποιώντας τις νέες επιβεβαιώσεις που παραλαμβάνει από τον παραλήπτη.

Η λήψη διπλοεπιβεβαιώσεων ή λήξη ενός χρονικού περιθωρίου οδηγεί σε επαναμετάδοση του χαμένου πακέτου στον παραλήπτη, εάν το πακέτο έχει αποθηκευτεί στο χώρο που διαθέτει ο πράκτορας Snoop. Ακόμη, ο πράκτορας Snoop αποκρύπτει από τον αποστολέα τις διπλές επιβεβαιώσεις που αντιστοιχούν σε ασύρματες απώλειες, ώστε να αποφεύγονται λανθασμένες χρήσεις του μηχανισμού ελέγχου συμφόρησης που σχετίζονται με το μηχανισμό γρήγορης επαναμετάδοσης. Αν ο πράκτορας δεν έχει αποθηκεύσει το πακέτο, τότε γίνεται αντιληπτό ότι η απώλεια προκλήθηκε λόγω συμφόρησης. Οι αντίστοιχες διπλές επιβεβαιώσεις προωθούνται στον αποστολέα TCP, επιτρέποντας τη χρήση των μηχανισμών γρήγορης επαναμετάδοσης και ελέγχου συμφόρησης ως συνήθως.

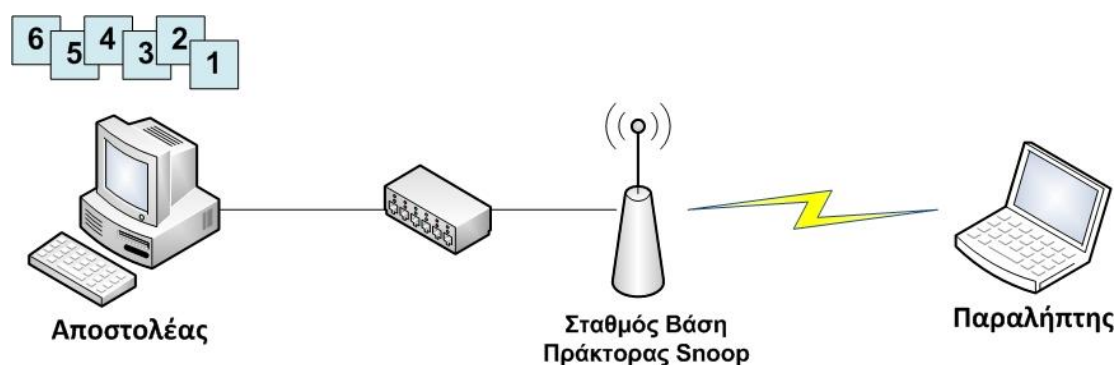
Το πρωτόκολλο Snoop πετυχαίνει τις παραπάνω λειτουργίες χρησιμοποιώντας δύο βασικές συναρτήσεις τη **snoop_data()** και τη **snoop_ack()**.

Η συνάρτηση **snoop_data()** επεξεργάζεται και αποθηκεύει προσωρινά τα πακέτα που προορίζονται για τον κινητό κόμβο, ενώ ταυτόχρονα ασχολείται με τα πακέτα που επαναμεταδίδονται ή φτάνουν εκτός σειράς στον παραλήπτη. Όταν ένα πακέτο φτάνει εντός σειράς και έχει αριθμό ακολουθίας μεγαλύτερο από το προηγούμενο πακέτο, το πρωτόκολλο Snoop το αποθηκεύει στο σταθμό βάση και το προωθεί στον κινητό κόμβο. Ο χρόνος μετ' επιστροφής για τη μετάδοση του πακέτου υπολογίζεται χρησιμοποιώντας ένα τοπικό ρολόι. Αν ένα πακέτο φτάσει εκτός σειράς και με αριθμό ακολουθίας μικρότερο από το τελευταίο πακέτο που επιβεβαιώθηκε, αυτό σημαίνει ότι ένα επαναμεταδιδόμενο πακέτο από τον αποστολέα έφτασε στο σταθμό βάση. Αν το πακέτο έχει ήδη σταλεί στον κινητό κόμβο, το πρωτόκολλο Snoop επιστρέφει μια επιβεβαίωση στον αποστολέα, ώστε να μην λήξει το χρονικό περιθώριο του, αλλιώς προωθεί το πακέτο στον κινητό κόμβο και το αποθηκεύει, ενώ επαναμεταδίδεται από τον αποστολέα.

Η συνάρτηση **snoop_ack()** επεξεργάζεται τις επιβεβαιώσεις που παραλαμβάνονται από τον κινητό κόμβο και εκτελεί επαναμεταδόσεις. Όταν παραλαμβάνεται μια επιβεβαίωση από τον κινητό κόμβο, το πρωτόκολλο Snoop αναγνωρίζει αν η επιβεβαίωση είναι αυθεντική, πλαστή ή διπλή. Αν είναι μια αυθεντική επιβεβαίωση, δηλαδή είναι εντός σειράς, οι τοπικοί καταχωρητές αδειάζουν, υπολογίζεται ο χρόνος μετ' επιστροφής και η επιβεβαίωση προωθείται στον

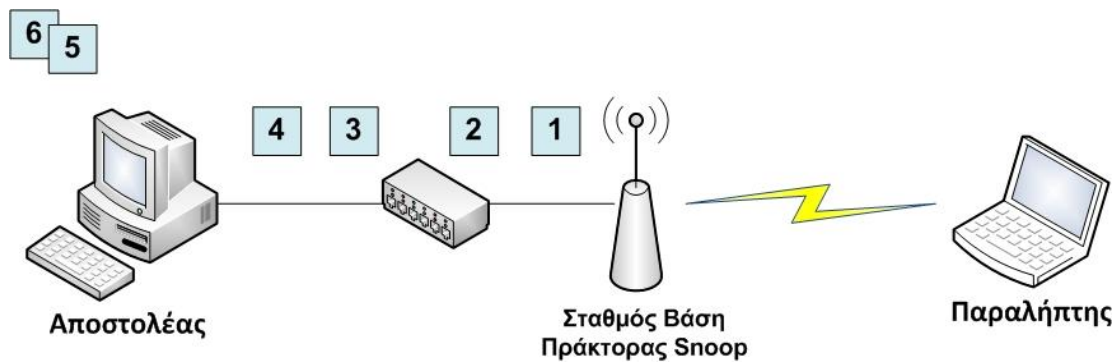
αποστολέα. Αν είναι μια επιβεβαίωση με αριθμό ακολουθίας μικρότερο από τον τελευταίο που παραλήφθηκε, τότε είναι πλαστή επιβεβαίωση και απορρίπτεται. Αν μια διπλή επιβεβαίωση παραλαμβάνεται για ένα πακέτο το οποίο δεν βρίσκεται στον προσωρινό χώρο αποθήκευσης ή έχει σημειωθεί για επαναμετάδοση από τον αποστολέα, τότε δρομολογείται στον αποστολέα. Αν ληφθεί μια διπλή επιβεβαίωση για ένα πακέτο το οποίο βρίσκεται αποθηκευμένο στο προσωρινό χώρο του πράκτορα Snoop, τότε επαναμεταδίδεται το πακέτο στον κινητό κόμβο. Αν παραληφθεί μια διπλή επιβεβαίωση για ένα χαμένο πακέτο, τότε αυτή απορρίπτεται, αποτρέποντας έτσι μη αναγκαίες επαναμεταδόσεις.

Στις εικόνες που ακολουθούν απεικονίζεται μια τυπική μετάδοση δεδομένων χρησιμοποιώντας το πρωτόκολλο Snoop. Η τοπολογία (Σχήμα 3.2) αποτελείται από έναν σταθερό κόμβο-αποστολέα, έναν σταθμό βάση-πράκτορα Snoop και έναν κινητό κόμβο-παραλήπτη. Υποθέτουμε ότι ο αποστολέας έχει να αποστείλει 9 πακέτα δεδομένων.



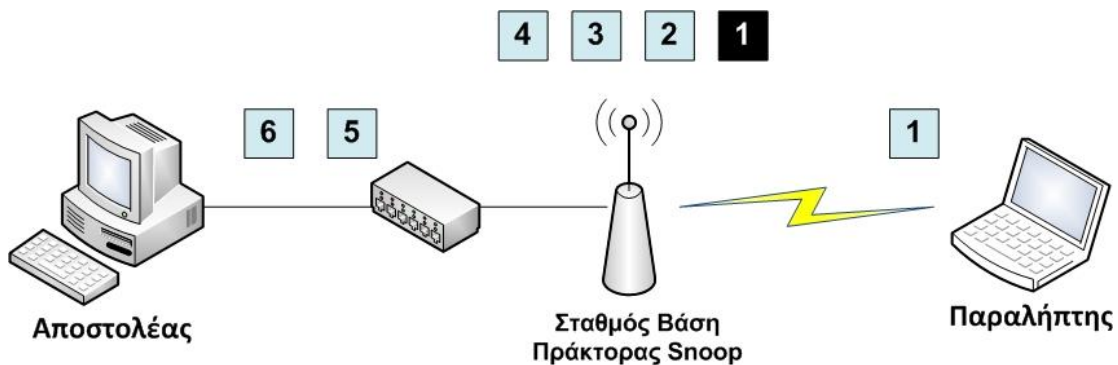
Σχήμα 3.2 Παράδειγμα αποστολής δεδομένων
χρησιμοποιώντας το πρωτόκολλο Snoop

Ο αποστολέας ξεκινάει την αποστολή των τεσσάρων πρώτων πακέτων, τα οποία παραλαμβάνει και αποθηκεύει προσωρινά ο σταθμός βάση πριν τα προωθήσει στον παραλήπτη. (Σχήμα 3.3)



Σχήμα 3.3 Παράδειγμα αποστολής δεδομένων χρησιμοποιώντας το πρωτόκολλο Snoop

Ο σταθμός βάση προωθεί τα πακέτα 1, 2 και 3 στον παραλήπτη, όμως το πακέτο 1 για κάποιο λόγο χάνεται κατά τη μετάδοση. Ταυτόχρονα, ο αποστολέας μεταδίδει τα πακέτα 5 και 6. (Σχήματα 3.4-3.5)



Σχήμα 3.4 Παράδειγμα αποστολής δεδομένων χρησιμοποιώντας το πρωτόκολλο Snoop



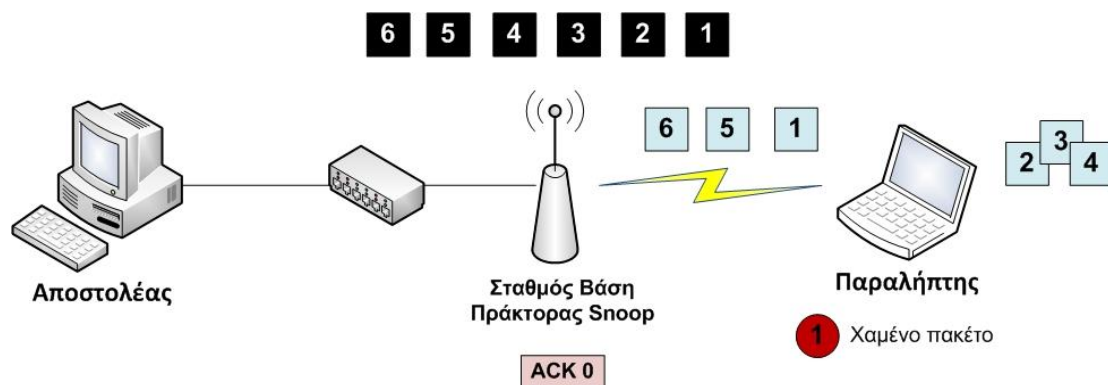
Σχήμα 3.5 Παράδειγμα αποστολής δεδομένων χρησιμοποιώντας το πρωτόκολλο Snoop

Ο παραλήπτης παραλαμβάνει το πακέτο 2 και επιστρέφει την επιβεβαίωση ότι περιμένει να παραλάβει το πρώτο πακέτο. Την ίδια στιγμή ο σταθμός βάση μεταδίδει το πακέτο 4. (Σχήμα 3.6)



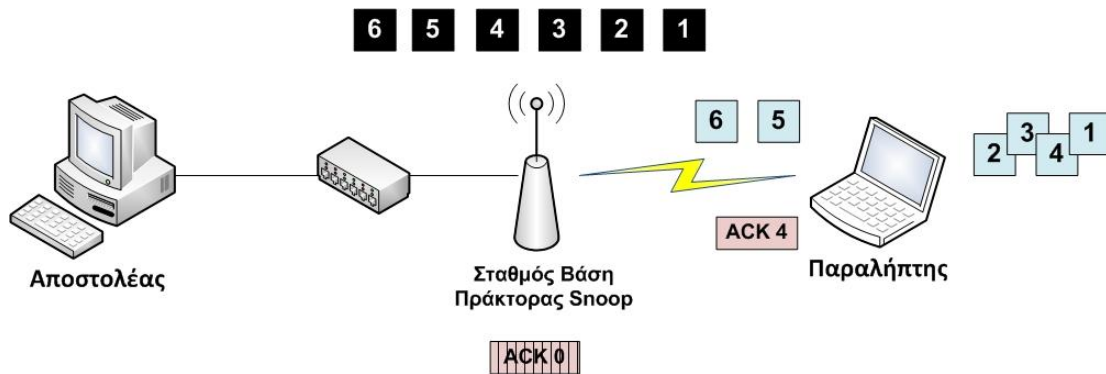
Σχήμα 3.6 Παράδειγμα αποστολής δεδομένων χρησιμοποιώντας το πρωτόκολλο Stop-and-Wait

Μόλις ο σταθμός βάση παραλάβει την επιβεβαίωση, αντιλαμβάνεται ότι έχει χαθεί το πακέτο 1. Έτσι, το επαναμεταδίδει αμέσως, χρησιμοποιώντας το αντίγραφο που είχε αποθηκεύσει στην προσωρινή του μνήμη, ακολουθούμενο από τα πακέτα 5 και 6. (Σχήμα 3.7) Με τον τρόπο αυτό ο αποστολέας δεν αντιλαμβάνεται την απώλεια του πρώτου πακέτου και, συνεπώς, δεν χρειάζεται να μειώσει το ρυθμό αποστολής του.



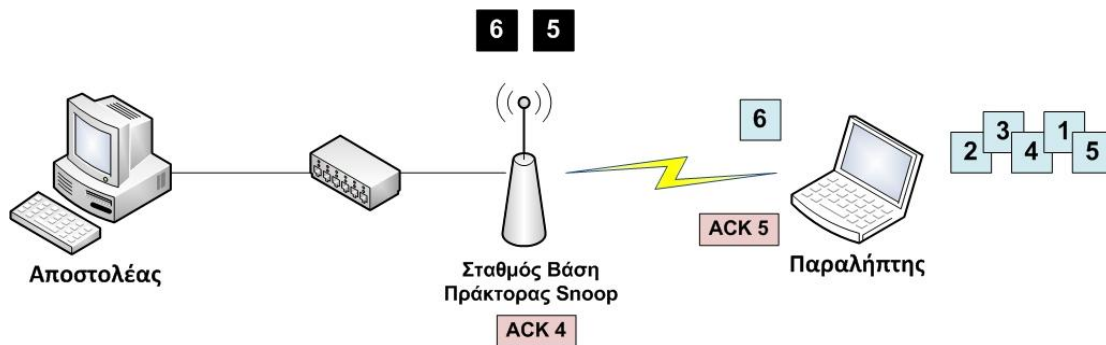
Σχήμα 3.7 Παράδειγμα αποστολής δεδομένων χρησιμοποιώντας το πρωτόκολλο Stop-and-Wait

Όταν ο παραλήπτης παραλάβει το πακέτο 1, επιστρέφει μια επιβεβαίωση στο σταθμό βάση μια συνοπτική επιβεβαίωση για τα πρώτα τέσσερα πακέτα που παρέλαβε. (Σχήμα 3.8)

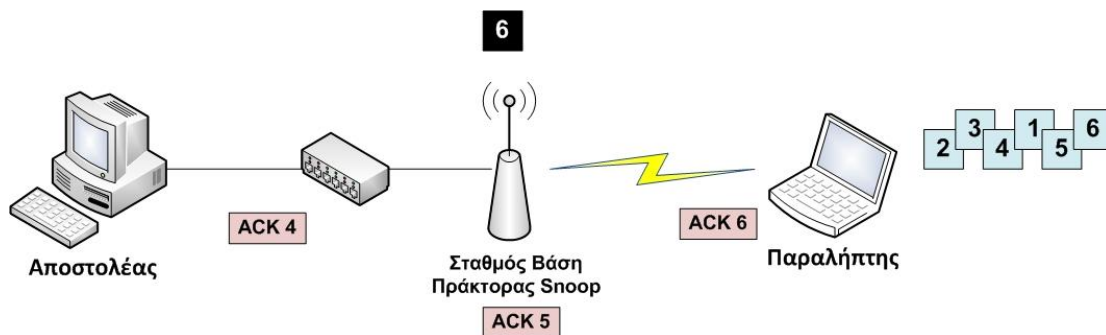


Σχήμα 3.8 Παράδειγμα αποστολής δεδομένων χρησιμοποιώντας το πρωτόκολλο Stop-and-Wait

Ο σταθμός βάση παραλαμβάνει τη συνοπτική επιβεβαίωση για τα πρώτα τέσσερα πακέτα και την προωθεί στον αποστολέα. Ταυτόχρονα, ο παραλήπτης επιβεβαιώνει τα πακέτα 5 και 6 και προωθεί τις αντίστοιχες επιβεβαιώσεις προς τον αποστολέα (Σχήματα 3.9-3.10)

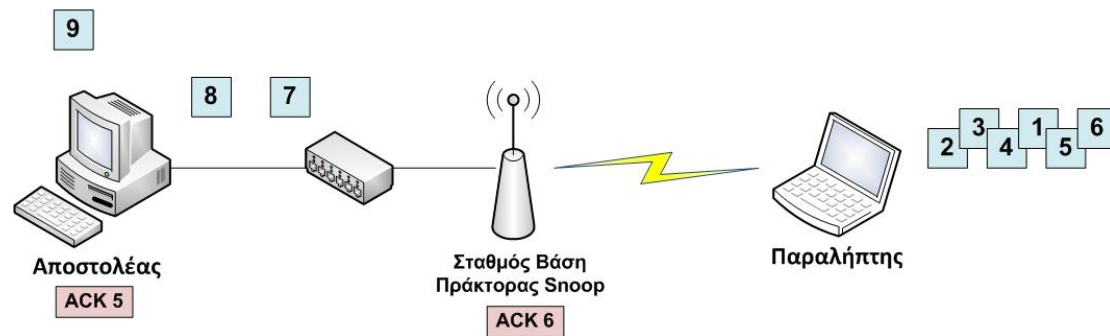


Σχήμα 3.9 Παράδειγμα αποστολής δεδομένων χρησιμοποιώντας το πρωτόκολλο Stop-and-Wait



Σχήμα 3.10 Παράδειγμα αποστολής δεδομένων χρησιμοποιώντας το πρωτόκολλο Stop-and-Wait

Ο αποστολέας παραλαμβάνει τις επιβεβαιώσεις που προέρχονται από τον αποστολέα και συνεχίζει κανονικά την αποστολή δεδομένων (Σχήμα 3.11)



Σχήμα 3.11 Παράδειγμα αποστολής δεδομένων
χρησιμοποιώντας το πρωτόκολλο Snoop

3.2. Automatic Repeat reQuest (ARQ)

Στο επίπεδο σύνδεσης, ο μηχανισμός ARQ (Automatic Repeat reQuest) [24, 25, 26, 27, 28, 29] λειτουργεί σε μπλοκ δεδομένων, γνωστά ως πλαίσια (frames), και προσπαθεί να τα παραδώσει από τον αποστολέα προς τον παραλήπτη πάνω από το επίπεδο σύνδεσης. Κανάλι αποτελεί το φυσικό επίπεδο σύνδεσης πάνω από το οποίο λειτουργεί το πρωτόκολλο σύνδεσης. Στην απλούστερη μορφή του ένα κανάλι μπορεί να είναι μια απευθείας σύνδεση φυσικού επιπέδου ανάμεσα σε δύο κόμβους.

Οι πιο σημαντικές μέθοδοι ελέγχου λαθών για συνδέσεις με ανατροφοδότηση είναι οι εξής:

- ARQ
- Type-I HARQ
- Type-II HARQ

3.2.1. Pure ARQ

Στα πρωτόκολλα ARQ με την ανίχνευση σφαλμάτων που δεν μπορούν να διορθωθούν σε ένα πακέτο, ο παραλήπτης απορρίπτει το πακέτο αυτό. Έτσι, ανιχνεύοντας τα χαμένα πακέτα στο επίπεδο μεταφοράς, χάρη στην μη ολοκληρωμένη ακολουθία αρίθμησης των δεδομένων, ο παραλήπτης ζητά επαναμετάδοση των χαμένων πακέτων. Οι πιο συχνά χρησιμοποιούμενοι κώδικες ανίχνευσης λαθών για ARQ πρωτόκολλα είναι οι κώδικες κυκλικού ελέγχου πλεονασμού CRC (Cyclic Redundancy Check).

Το ARQ, παρόλο που δεν καταναλώνει μεγάλη χωρητικότητα, αυξάνει τη καθυστέρηση μετ' επιστροφής RTT, γεγονός που μπορεί να μειώσει την απόδοση του TCP, αν δεν γίνει σωστά. Ένα άλλο πρόβλημα του ARQ είναι η αλληλεπίδραση του με τα χρονικά περιθώρια του TCP. Ένα χρονικό περιθώριο επαναμετάδοσης μπορεί να λήξει, ενώ το χαμένο πακέτο μεταδίδεται πάνω από το ασύρματο δίκτυο.

3.2.1.1. Πρωτόκολλα επαναμετάδοσης

Κατά την επιλογή ενός πρωτοκόλλου επαναμετάδοσης, ο σχεδιαστής θα πρέπει να ισορροπήσει ανάμεσα στην πολυπλοκότητα του σχεδιασμού και την απόδοση του συστήματος. Τα τρία βασικά πρωτόκολλα επαναμετάδοσης είναι τα εξής:

- Stop-and-wait
- Go-back-N
- Selective Repeat

Stop-and-wait: Στο πρωτόκολλο stop-and-wait [30] ο αποστολέας στέλνει ένα πακέτο και περιμένει μια επιβεβαίωση. Ο παραλήπτης απαντά στέλνοντας την επιβεβαίωση, αν το πακέτο παραληφθεί χωρίς σφάλματα, ή στέλνοντας μια αίτηση επαναμετάδοσης, αν το πακέτο περιείχε ένα ανιχνεύσιμο λάθος. Ο αποστολέας παραμένει αδρανής, ενώ περιμένει την επιβεβαίωση.

Δύο ειδών λάθη μπορεί να συμβούν με τη χρήση του πρωτοκόλλου stop-and-wait ARQ. Πρώτον, το πακέτο δεδομένων που φτάνει στον παραλήπτη μπορεί να είναι κατεστραμμένο. Ο παραλήπτης ανιχνεύει το σφάλμα και απορρίπτει το πακέτο. Ο αποστολέας περιμένει μέχρι ένα συγκεκριμένο χρονικό περιθώριο για την επιβεβαίωση, αφού μεταδώσει ένα πακέτο δεδομένων. Αν το χρονικό περιθώριο λήξει και δεν έχει παραληφθεί κάποια επιβεβαίωση, τότε το πακέτο μεταδίδεται ξανά.

Δεύτερον, υπάρχει πιθανότητα η επιβεβαίωση να είναι κατεστραμμένη. Έστω ότι ένας αποστολέας A αποστέλλει ένα πακέτο δεδομένων. Το πακέτο παραλαμβάνεται επιτυχώς από τον παραλήπτη B, ο οποίος απαντά με μια επιβεβαίωση (ACK). Η επιβεβαίωση, όμως, καταστρέφεται κατά την μετάδοση της, με αποτέλεσμα να μην μπορεί ο αποστολέας A να την αναγνωρίσει και, συνεπώς, να λήξει το χρονικό περιθώριο και να επαναμεταδώσει το ίδιο πακέτο. Η δεύτερη μετάδοση του ίδιου πακέτου γίνεται δεκτή από τον παραλήπτη, ο οποίος πλέον έχει παραλάβει το ίδιο πακέτο δύο φορές. Για την αποφυγή του προβλήματος αυτού, τα πακέτα δεδομένων ονομάζονται διαδοχικά 0 ή 1 και οι θετικές επιβεβαιώσεις έχουν την μορφή ACK0 και ACK1. Διατηρώντας τον μηχανισμό του κυλιόμενου παραθύρου, αν μια επιβεβαίωση ACK0 επιβεβαιώνει την παραλαβή ενός πακέτου που είχε αρίθμηση 1, τότε ο αποστολέας αναγνωρίζει ότι περιμένει ένα πακέτο με αρίθμηση 0.

Το κύριο πλεονέκτημα του SW-ARQ πρωτοκόλλου είναι ότι δεν υπάρχει ανάγκη για προσωρινή αποθήκευση των πακέτων στον αποστολέα ή στον παραλήπτη. Το κύριο μειονέκτημα του πρωτοκόλλου είναι ότι πετυχαίνει χαμηλή απόδοση, ειδικά όταν η καθυστέρηση διάδοσης έχει υψηλές τιμές (δορυφορικές επικοινωνίες).

Go-back-N: Αν υπάρχει η δυνατότητα προσωρινής αποθήκευσης στον αποστολέα, το πρωτόκολλο Go-back-N [31] μπορεί να εφαρμοστεί. Στο πρωτόκολλο αυτό ο αποστολέας μεταδίδει πακέτα με ένα συνεχόμενο ρυθμό. Όταν ο παραλήπτης ανιχνεύσει ένα λάθος, στέλνει ένα αίτημα επαναμετάδοσης για το συγκεκριμένο πακέτο και περιμένει να παραλάβει το αντίγραφο του. Ταυτόχρονα, ο παραλήπτης αγνοεί όλα τα ακόλουθα πακέτα μέχρι να παραλάβει το αντίγραφο και, έτσι,

αποφεύγεται η αποθήκευση στον παραλήπτη. Ο αποστολέας απαντάει επαναμεταδίδοντας το ζητούμενο πακέτο και όλα τα ακόλουθα πακέτα για το λόγο αυτό είναι απαραίτητη η αποθήκευση στον αποστολέα.

Selective Repeat: Αν υπάρχει δυνατότητα αποθήκευσης τόσο στον αποστολέα, όσο και στον παραλήπτη, μπορούμε να εφαρμόσουμε ένα πρωτόκολλο SR-ARQ [32]. Ο αποστολέας μεταδίδει μια συνεχόμενη ακολουθία πακέτων και απαντά σε αιτήσεις επαναμετάδοσης αποστέλλοντας το ζητούμενο πακέτο και συνεχίζοντας τη μετάδοση από εκεί που σταμάτησε.

3.2.2. Hybrid-ARQ (HARQ)

Τα πρωτόκολλα HARQ [33] αποτελούν συνδυασμό των μηχανισμών FEC και ARQ και μπορούν τόσο να ανιχνεύσουν, όσο να διορθώσουν σφάλματα. Έτσι, περιορίζεται το πλήθος των επαναμεταδόσεων που δημιουργούνται σε δίκτυα που εμφανίζουν υψηλό ρυθμό σφαλμάτων και μειώνεται σημαντικά το φορτίο στο κανάλι επιστροφής και η καθυστέρηση παράδοσης. Τα πρωτόκολλα HARQ πετυχαίνουν απόδοση παρόμοια με αυτή των συστημάτων FEC, ενώ ταυτόχρονα παρέχουν αξιοπιστία αντίστοιχη αυτής των πρωτοκόλλων ARQ.

Τα πρωτόκολλα HARQ χωρίζονται σε απλούστερα πρωτόκολλα τύπου I και πιο πολύπλοκα πρωτόκολλα τύπου II.

Τύπου-I HARQ: Τα πρωτόκολλα τύπου-I HARQ μπορούν να εφαρμοστούν χρησιμοποιώντας συστήματα ενός ή δύο κωδικών. Σε ένα σύστημα δύο κωδικών τα δεδομένα πρώτα κωδικοποιούνται χρησιμοποιώντας έναν κώδικα υψηλού ρυθμού ανίχνευσης λαθών (συχνά χρησιμοποιούνται κώδικες CRC) και αποκωδικοποιούνται χρησιμοποιώντας και πάλι έναν κώδικα FEC.

Τύπου-II HARQ: Τα πρωτόκολλα τύπου-II HARQ είναι συστήματα που συνδυάζουν διαφορετικούς κώδικες με τον περιορισμό ότι το πλήθος των συνδυασμένων πακέτων δεν θα ξεπεράσει τα δύο. Χρησιμοποιώντας κυμαινόμενο μέγεθος

πλεονασμού, τα πρωτόκολλα αυτά προσαρμόζονται στις εναλλασσόμενες συνθήκες του καναλιού. Ο αποστολέας απαντά στις αιτήσεις επαναμετάδοσης αποστέλλοντας επιπλέον bits ισοτιμίας στον παραλήπτη, τα οποία επιτρέπουν την αυξημένη διόρθωση λαθών.

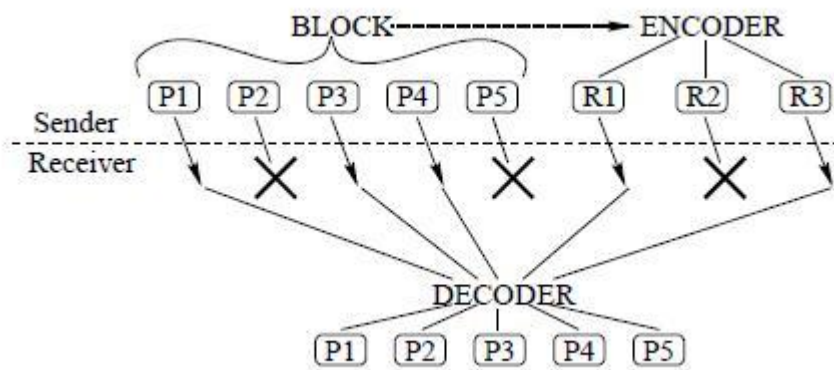
3.3. Forward Error Correction (FEC)

Μια απλή λύση για τη βελτίωση της ποιότητας ενός καναλιού που παρουσιάζει απώλειες, η οποία δεν απαιτεί καμία αλλαγή στο υπάρχον TCP και δεν αλληλεπιδρά με τους μηχανισμούς διόρθωσης λαθών, αποτελεί η χρήση μηχανισμών εκ των προτέρων διόρθωσης λαθών FEC (Forward Error Correction) [34, 35]. Η ιδέα πάνω από το FEC είναι η αποστολή, εκτός από τα αρχικά δεδομένα, κάποιων επιπλέον πληροφοριών, ώστε ένα πακέτο το οποίο έχει υποστεί «φθορά» ενώ μεταδιδόταν, να μπορεί να ανακατασκευαστεί στη πλευρά του παραλήπτη, χωρίς να απαιτείται καμία επαναμετάδοση. Μπορεί κανείς να θεωρήσει το μηχανισμό FEC ως μια αποστολή μαζί με τα αρχικά δεδομένα και κάποιων αντιγράφων τους, ώστε ένα αντίγραφο, εφόσον δεν χαθεί, να μπορεί να χρησιμοποιηθεί, αν το αρχικό πακέτο χαθεί. Αυτό βελτιώνει την ποιότητα του καναλιού που παρουσιάζει σφάλματα, καταναλώνοντας κάποια επιπλέον χωρητικότητα.

Οι κώδικες Forward Error Correction (FEC) χρησιμοποιούνται για να διορθώσουν σχετικά μικρό αριθμό λαθών ή απωλειών δεδομένων σε μια ακολουθία byte. Οι απώλειες δεδομένων αντιμετωπίζονται ευκολότερα από ότι οι διορθώσεις σφαλμάτων, αφού το σημείο που συνέβη η απώλεια είναι εκ των προτέρων γνωστό. Οι κώδικες FEC αποκρύπτουν από το πρωτόκολλο μεταφοράς τα λάθη που μπορούν να διορθωθούν, μειώνοντας έτσι τις επαναμεταδόσεις από τον αποστολέα.

Έστω ότι τα δεδομένα ομαδοποιούνται για τη δημιουργία ενός μπλοκ δεδομένων μεγέθους k . Ο αποστολέας εισάγει k πακέτα στον κωδικοποιητή του, ο οποίος παράγει τα επιπλέον πακέτα για το μπλοκ αυτό. Ο παραλήπτης χρησιμοποιεί τον αποκωδικοποιητή του για να ξαναδημιουργήσει τα δεδομένα χρησιμοποιώντας οποιονδήποτε συνδυασμό k πακέτων, είτε προέρχονται από τα αρχικά δεδομένα,

είτε από τα επιπλέον πακέτα που παράχθηκαν από τον κωδικοποιητή. Ένα παράδειγμα χρήσης κωδικοποιητή και αποκωδικοποιητή FEC δίνεται παρακάτω.



Σχήμα 3.12 Κωδικοποίηση και αποκωδικοποίηση δεδομένων χρησιμοποιώντας Forward Error Correction

Αρχικά, ο αποστολέας ομαδοποιεί 5 πακέτα δεδομένων σε ένα μπλοκ (P1-P5). Εισάγει τα πακέτα αυτά στον κωδικοποιητή του και παράγει 3 επιπλέον πακέτα δεδομένων (R1-R3) τα οποία αποστέλλει στον παραλήπτη μαζί με τα αρχικά δεδομένα. Μόλις ο παραλήπτης παραλάβει οποιαδήποτε 5 πακέτα (είτε είναι αρχικά δεδομένα, είτε επιπλέον πακέτα), ενεργοποιεί τον αποκωδικοποιητή του και ανακάμπτει τα χαμένα πακέτα. Στο συγκεκριμένο παράδειγμα, τα πακέτα P2, P5 και R2 δεν φτάνουν ποτέ στον παραλήπτη, αλλά αναδημιουργούνται χρησιμοποιώντας τα υπόλοιπα πακέτα.

Έχει αποδειχθεί ότι υπάρχει ένα βέλτιστο επίπεδο επιπλέον δεδομένων που μπορούν να προστεθούν, πάνω από το οποίο η απόδοση του TCP μειώνεται αντί να βελτιώνεται, αν και η μείωση αυτή είναι πιο αργή από το κέρδος σε απόδοση που πετυχαίνουμε όταν προσθέτουμε τις πρώτες μονάδες επιπλέον δεδομένων. Το άλλο μειονέκτημα του FEC είναι ότι απαιτεί κάποιο χρόνο επεξεργασίας για την κωδικοποίηση και την αποκωδικοποίηση των επιπλέον δεδομένων. Ο αλγόριθμος αποκωδικοποίησης είναι ικανός να ανιχνεύσει περισσότερα λάθη από αυτά που μπορεί να διορθώσει, ενώ ο μηχανισμός FEC χρησιμοποιεί δύο είδη κωδικοποιήσεων: την κωδικοποίηση μπλοκ και την ελικοειδή κωδικοποίηση.

Παρόλα αυτά, τα πλεονεκτήματα του FEC είναι πάρα πολλά και, έτσι, το κάνουν μια ελκυστική λύση, παρόλο το κόστος του. Τα κατεστραμμένα πακέτα διορθώνονται αμέσως, γεγονός που εξαλείφει τις διακυμάνσεις στο RTT που προκαλούν οι τοπικές επαναμεταδόσεις, και όλα τα πακέτα παραδίδονται με την σειρά που μεταδόθηκαν πάνω από το κανάλι. Με τον τρόπο αυτό αποφεύγεται η δημιουργία μιας διπλής επιβεβαίωσης, αν το χαμένο πακέτο επαναμεταδοθεί τοπικά και τα πακέτα που ακολουθούν συνεχίζουν να προωθούνται στον τελικό προορισμό.

Κεφάλαιο 4

Δορυφορικά δίκτυα και μηχανισμοί αξιοπιστίας

Οι εξελίξεις στις τεχνολογίες του διαστήματος τη τελευταία δεκαετία έχουν οδηγήσει στην ραγδαία αύξηση του πλήθους των δορυφόρων και των επιστημονικών αποστολών στο Διάστημα. Οι αποστολές αυτές παράγουν σημαντική ποσότητα επιστημονικών δεδομένων, τα οποία αποστέλλονται στη Γη. Για την επιτυχή μεταφορά των επιστημονικών δεδομένων και την αξιόπιστη επικοινωνία, οι διαστημικές αποστολές έχουν παρουσιάσει σημαντικές προκλήσεις σχετικά με την ανάπτυξη νέων πρωτοκόλλων και αρχιτεκτονικών των διαστημικών δικτύων νέας γενιάς. Τα υπάρχοντα δικτυακά πρωτόκολλα δεν είναι κατάλληλα για χρήση σε δορυφορικά δίκτυα, λόγω των ιδιαίτερων χαρακτηριστικών του Διαστήματος. Για το λόγο αυτό, έχει προταθεί ένα πλήθος εναλλακτικών, αξιόπιστων μηχανισμών και πρωτοκόλλων.

4.1. Δορυφορικά δίκτυα

4.1.1. Τύποι δορυφόρων

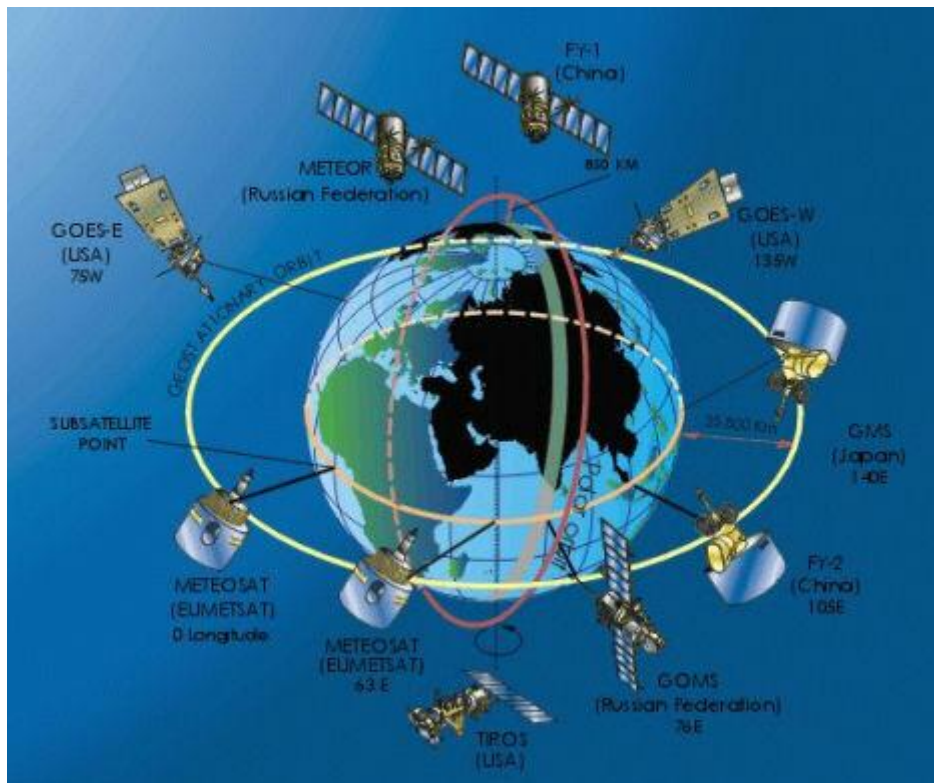
Υπάρχουν 3 τύποι δορυφόρων επικοινωνίας και χωρίζονται σε κατηγορίες ανάλογα με την τροχιά που κάνουν:

- **Γεωστατικοί δορυφόροι (Geostationary satellites):** Βρίσκονται σε τροχιά γύρω από τον Ισημερινό σε ύψος 22.000 μίλια και περισσότερο. Σε αυτό το ύψος (σε σχέση με τον ήλιο) πραγματοποιούν μια περιφορά της γης σε 24 ώρες. Οι δορυφόροι αυτοί μένουν σταθεροί πάνω από την Γη και έχουν την δυνατότητα από αυτό το ύψος να «βλέπουν» το 40% της Γης. Τρεις τέτοιοι δορυφόροι με διαφορά 120 μοιρών καλύπτουν τον γνωστό πολιτισμένο κόσμο.
- **Δορυφόροι χαμηλής τροχιάς (Low earth orbit - LEO):** Υπάρχει ένα μεγάλος «στόλος» από τέτοιους δορυφόρους οι οποίοι περιστρέφονται κυκλικά σε

ύψος μερικών εκατοντάδων μιλίων και περνούν κοντά από τους γεωγραφικούς πόλους. Κάθε περιφορά διαρκεί από 90 λεπτά μέχρι μερικές ώρες. Η τροχιά τους έχει ρυθμισθεί κατά τέτοιο τρόπο, ώστε από κάθε σημείο της γης την κάθε στιγμή να φαίνεται ένας.

- ο **Δορυφόροι ελλειπτικής τροχιάς (Elliptical orbit satellites):** Αυτοί οι δορυφόροι κινούνται με μεγάλη ταχύτητα, όταν βρίσκονται χαμηλά, και κινούνται αργά όταν βρίσκονται σε μεγάλη απόσταση και σε μεγάλο ύψος από την Γη. Χρησιμοποιούνται για επαγγελματικούς και κυβερνητικούς σκοπούς και απαιτούν κατευθυντικές κεραίες για την συνεχή παρακολούθηση του ίχνους.

Μια τυπική αρχιτεκτονική δικτύου σε δορυφορικό περιβάλλον που αποτελείται από ένα πλήθος δορυφόρων φαίνεται στην παρακάτω εικόνα.



Σχήμα 4.1 Αρχιτεκτονική δορυφορικού δικτύου

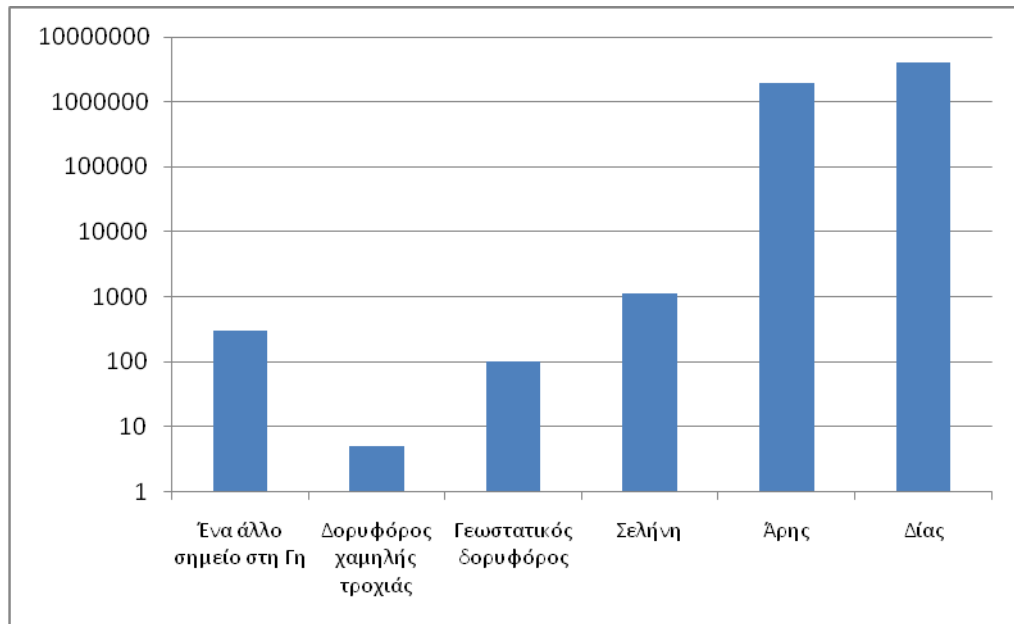
4.1.2. Διαφορές δικτυακών χαρακτηριστικών ανάμεσα στο Διαδίκτυο και τα δορυφορικά δίκτυα

Στο Διαδίκτυο επικρατεί η χρήση του συνόλου πρωτοκόλλων TCP/IP, 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 την επικοινωνία ενός κόμβου στη Γη με έναν δορυφόρο είναι απαραίτητη η προσαρμογή της στοίβας πρωτοκόλλων στα χαρακτηριστικά του δορυφορικού περιβάλλοντος, το οποίο γενικά χαρακτηρίζεται από:

- Μεγάλες τιμές καθυστέρησης διάδοσης. Οι συνδέσεις επικοινωνίας στο Διάστημα μπορεί να έχουν μεγάλες καθυστερήσεις διάδοσης. Για παράδειγμα, ο χρόνος μετ' επιστροφής από άκρο σε άκρο σε ένα δίκτυο επικοινωνίας μεταξύ της Γης και της Σελήνης φτάνει τα δύο δευτερόλεπτα. Στον παρακάτω πίνακα δίνονται κάποιες ενδεικτικές τιμές καθυστερήσεων.



Σχήμα 4.2 Μέγιστη τιμή καθυστέρησης για αυξανόμενη απόσταση από την επιφάνεια της Γης

- Σχετικά χαμηλούς ρυθμούς δεδομένων.
- Υψηλούς ρυθμούς σφαλμάτων στις συνδέσεις.
- Πιθανές χρονικές περιόδους αποσύνδεσης της λήψης ή της μετάδοσης, εξαιτίας των τροχιών που κινούνται οι δορυφόροι.
- Κεντρικά διαχειριζόμενη πρόσβαση στο κανάλι επικοινωνίας με ουσιαστικά καμία πιθανότητα συμφόρησης.

Ο επόμενος πίνακας [36] συνοψίζει τις διαφορές που παρουσιάζουν οι γήινες και οι δορυφορικές επικοινωνίες, όσον αφορά τα δικτυακά χαρακτηριστικά.

	Επίγειες επικοινωνίες	Δορυφορικές επικοινωνίες
Ρυθμός Σφαλμάτων Bit (BER)	Τυπικά μικρότερος από 10^{-9}	Τυπικά 10^{-4}
Καθυστέρηση διάδοσης RTT	Millisecond	Δευτερόλεπτα
Συνδεσιμότητα	Συνεχόμενη	Διακοπτόμενη
Στόχοι επικοινωνίας	Δίκαιη πρόσβαση Υψηλή απόδοση Υψηλή αξιοπιστία	Μέγιστη δυνατή απόδοση
Πηγή απώλειας δεδομένων	Συμφόρηση	Συμφόρηση Διακοπή σύνδεσης Φθορά

Πίνακας 4.1 Χαρακτηριστικά επίγειων και δορυφορικών επικοινωνιών

4.1.3. Προβλήματα που παρουσιάζει το TCP στο Διάστημα

Οι αρχικές εκδόσεις του πρωτοκόλλου μεταφοράς TCP (Transmission Control Protocol) έδειχναν να μην πετυχαίνουν καλή απόδοση στα δίκτυα επικοινωνιών στο Διάστημα [37, 38]. Ο κυρίαρχος παράγοντας στην μείωση της απόδοσης είναι οι υψηλές τιμές της καθυστέρησης διάδοσης που εμφανίζουν οι δορυφορικές συνδέσεις. Αυτό οφείλεται στο τρόπο λειτουργίας του μηχανισμού που βασίζεται στα παράθυρα και χρησιμοποιείται από το πρωτόκολλο TCP κατά την διάρκεια του αργού ξεκινήματος (Slow Start) και των αλγορίθμων αποφυγής συμφόρησης (Congestion Avoidance Algorithms).

Ακόμη, 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. 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 τον 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. Ο αλγόριθμος ελέγχου συμφόρησης του TCP λειτουργεί σωστά όταν η απώλεια προκαλείται από συμφόρηση, αλλά καταλήγει μόνο σε μείωση της απόδοσης σε θορυβώδεις συνδέσεις.

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.

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 στην 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 την 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.

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 επαναμετάδοση 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.

4.2. Μηχανισμοί αξιοπιστίας σε ασύρματα και δορυφορικά δίκτυα

Το πρωτόκολλο μεταφοράς TCP είναι το πλέον αποδεδειγμένα αξιόπιστο πρωτόκολλο μεταφοράς στα ενσύρματα δίκτυα και το πιο συχνά χρησιμοποιούμενο πρωτόκολλο για εφαρμογές στο Διαδίκτυο, αφού μπορεί να διαχειριστεί αποδοτικά τις απώλειες πακέτων, κυρίως λόγω της δικτυακής συμφόρησης. Το TCP, που είναι σχεδιασμένο κυρίως για ενσύρματα δίκτυα και για σταθμούς βάσεις, παρέχει αξιόπιστη μετάδοση δεδομένων από-άκρο-σε-άκρο ανάμεσα σε δύο εφαρμογές που εκτελούνται σε δύο απομακρυσμένους υπολογιστές. Αυτό το πετυχαίνει προσθέτοντας υπηρεσίες πάνω από το επίπεδο IP, το οποίο δεν εγγυάται αξιόπιστη παράδοση των πακέτων.

Η αυξανόμενη ανάγκη για κινητικότητα ανάμεσα στους χρήστες, οδήγησε στην απαίτηση για δημιουργία ασύρματων δικτύων. Έτσι, δημιουργήθηκε η ανάγκη για την προσαρμογή του TCP στα ασύρματα δίκτυα, μιας και το ασύρματο κανάλι υποφέρει από έλλειψη αξιοπιστίας εξαιτίας των υψηλών ρυθμών σφαλμάτων.

Το TCP υποθέτει ότι η απώλεια ενός πακέτου γενικά σημαίνει την ύπαρξη συμφόρησης στο δίκτυο. Έτσι, όταν εφαρμόζεται πάνω από ασύρματα δίκτυα, ελαττώνει το ρυθμό μετάδοσης, μειώνοντας το μέγεθος του παραθύρου συμφόρησης και, συνεπώς, το πλήθος των πακέτων δεδομένων που μπορεί να στείλει ο αποστολέας. Ο πιο ενδεδειγμένος τρόπος ανάκαμψης από απώλειες είναι ο αλγόριθμος του TCP Reno. Η απώλεια ενός πακέτου ανιχνεύεται από τη λήξη ενός χρονικού περιθωρίου (timeout) ή μια διπλή επιβεβαίωση. Η λήξη ενός χρονικού περιθωρίου συμβαίνει όταν ο αποστολέας TCP δεν παραλαμβάνει επιβεβαιώσεις από τον παραλήπτη για ένα προκαθορισμένο χρονικό διάστημα. Όταν λήξει το χρονικό περιθώριο, το TCP χειρίζεται την κατάσταση ως δικτυακή συμφόρηση και εκτελεί το μηχανισμό αργού ξεκινήματος (slow start). Στη δεύτερη περίπτωση, όταν ο αποστολέας παραλάβει μια διπλή επιβεβαίωση, αναγνωρίζει ότι ο παραλήπτης παρέλαβε κάποια πακέτα εκτός σειράς και αντιδρά εισερχόμενος σε μηχανισμό γρήγορης επαναμεταφοράς (fast retransmit) και γρήγορης ανάκαμψης (fast recovery).

Το κύριο πρόβλημα του TCP Reno είναι ότι δεν μπορεί να ξεχωρίσει τις απώλειες εξαιτίας συμφόρησης από τις μη-συμφορητικές απώλειες. Έτσι, οι απώλειες που εμφανίζονται εξαιτίας της ασύρματης φύσης του δικτύου, θεωρούνται απώλειες λόγω συμφόρησης και το TCP αντιδρά μειώνοντας το παράθυρο συμφόρησης, οδηγώντας σε σημαντική μείωση της απόδοσης και πολύ υψηλές καθυστερήσεις πάνω από το ασύρματο κανάλι.

Υπάρχουν δύο διαφορετικές προσεγγίσεις για τη βελτίωση της απόδοσης του TCP πάνω από ασύρματα και δορυφορικά δίκτυα:

- είτε αποκρύπτοντας από τον αποστολέα τις μη συμφορητικές απώλειες
- είτε προσαρμόζοντας τον αποστολέα, ώστε να καταλαβαίνει ότι κάποιες απώλειες δεν οφείλονται σε συμφόρηση και να αποφεύγει τη χρήση αλγορίθμων ελέγχου συμφόρησης για μη-συμφορητικές απώλειες.

Έχουν προταθεί διάφοροι μηχανισμοί βασισμένοι [23, 39, 40] στις παραπάνω προσεγγίσεις. Οι μηχανισμοί αυτοί μπορούν να διαχωριστούν σε τρεις κατηγορίες:

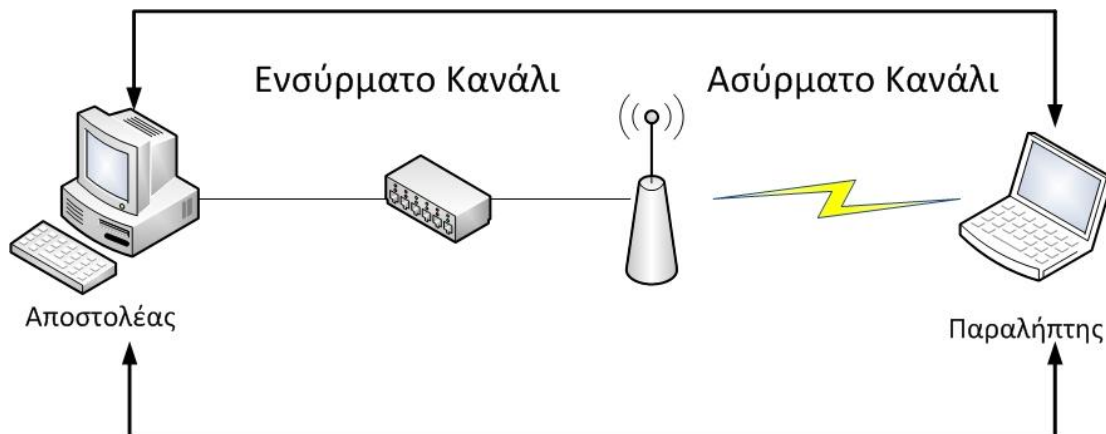
- μηχανισμοί από άκρο σε άκρο (end-to-end)
- μηχανισμοί διαχωρισμού σύνδεσης (split-connection)
- μηχανισμοί επιπέδου σύνδεσης (link layer)

4.2.1. Μηχανισμοί από άκρο σε άκρο

Στη μέθοδο από άκρο σε άκρο ο TCP αποστολέας χειρίζεται τις απώλειες μέσω δύο τεχνικών. Πρώτον, χρησιμοποιεί κάποια μορφή επιλεκτικών επιβεβαιώσεων ή γρήγορης επαναμετάδοσης για να επιτρέψει στον αποστολέα να ανακάμψει από πολλαπλές απώλειες πακέτων σε ένα παράθυρο, χωρίς να οδηγηθεί σε λήξη του χρονικού περιθωρίου. Δεύτερον, γίνεται προσπάθεια διαχωρισμού στη πλευρά του αποστολέα ανάμεσα στις απώλειες λόγω συμφόρησης και στις υπόλοιπες μορφές συμφόρησης χρησιμοποιώντας ρητή ειδοποίηση απώλειας ELN (Explicit Loss

Notification) [41] ή ρητή ειδοποίηση συμφόρησης ECN (Explicit Congestion Notification) [42].

Κάποιες προτάσεις που βασίζονται στο μηχανισμό αυτό είναι οι εξής: SACK, ELN, NAK, SMART, WTCP, TCP Probing, TCP Santa Cruz, ACK Pacing κ.α.



Σχήμα 4.3 Μηχανισμός από άκρο σε άκρο

SMART: Η απλή αυτή μέθοδος (Simple Method to Aid ReTransmissions) [43] διαχωρίζει τον έλεγχο ροής από τον έλεγχο σφαλμάτων, χρησιμοποιώντας δύο διαφορετικά παράθυρα· το παράθυρο ελέγχου λαθών που βρίσκεται στον παραλήπτη και χρησιμοποιείται για την αποθήκευση των πακέτων που παραλήφθηκαν εκτός σειράς και το παράθυρο ελέγχου ροής στον αποστολέα για την αποθήκευση πακέτων που δεν έχουν ακόμη επιβεβαιωθεί. Στο TCP κάθε πακέτο επιβεβαίωσης μεταφέρει τη συνοπτική επιβεβαίωση και τον αριθμό ακολουθίας (sequence number) του πρώτου πακέτου που «καλύπτει» η συγκεκριμένη επιβεβαίωση. Με τον τρόπο αυτό ο αποστολέας ενημερώνεται σχετικά με τα πακέτα που χάθηκαν, έτσι ώστε να μπορεί να τα επαναμεταδώσει επιλεκτικά. Στο μηχανισμό αυτό χρησιμοποιούνται ειδικές μέθοδοι για την εύρεση της χαμένης επαναμετάδοσης και την αποφυγή της εξάρτησης από τα χρονικά περιθώρια του TCP.

NAK: Όταν μια επιβεβαίωση αποστέλλεται, αρνητικές επιβεβαιώσεις NAK (Negative Acknowledgment) [44] μπορούν να συμπεριληφθούν στην επικεφαλίδα TCP για να υποδείξουν ένα πακέτο που παραλήφθηκε με σφάλματα. Έτσι, ο αποστολέας να

μπορεί να επαναμεταδώσει γρήγορα το πακέτο, χωρίς να τροποποιήσει το μέγεθος του παραθύρου συμφόρησης. Αυτό ισχύει με την υπόθεση ότι το πακέτο που περιέχει σφάλματα μπορεί ακόμη να φτάσει στον προορισμό του μαζί με τη διεύθυνση του αποστολέα.

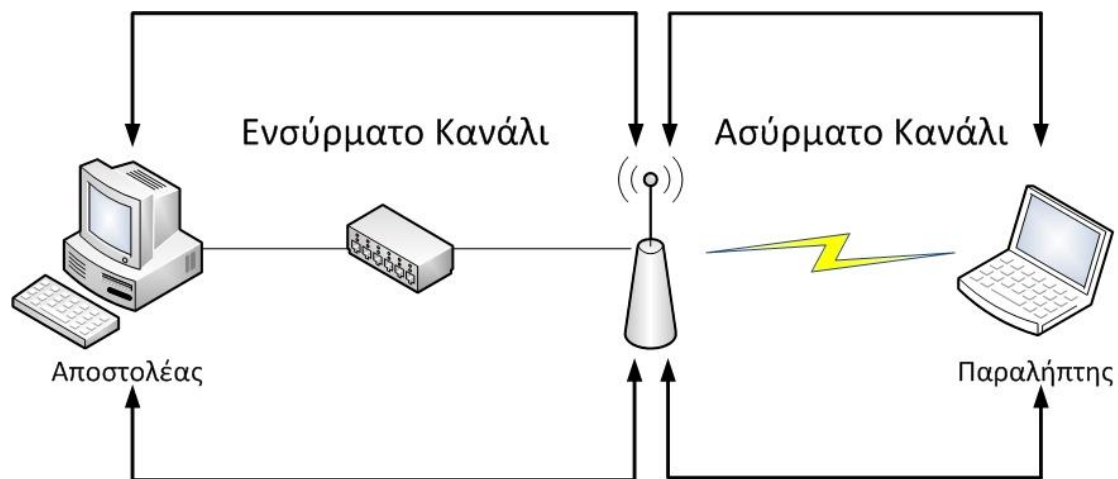
SACK: Το TCP SACK [45] αναπτύχθηκε από τους Floyd και Fall για να αντιμετωπίσει την ανικανότητα του TCP Reno να ανακάμπτει από πολλαπλές απώλειες σε ένα πακέτο δεδομένων. Όπως υποδεικνύει το RFC2018, ένας παραλήπτης SACK είναι ικανός να προσδιορίσει στον αποστολέα ακριβώς ποια πακέτα από την ακολουθία δεδομένων έχουν παραληφθεί σωστά και ποια όχι, χρησιμοποιώντας επιλεκτικές επιβεβαιώσεις (Selective Acknowledgments). Αυτό επιτυγχάνεται χρησιμοποιώντας έναν ειδικό τύπο επιλεκτικών επιβεβαιώσεων, ο οποίος ονομάζεται μπλοκ SACK. Ένα μπλοκ SACK παρέχει στον αποστολέα όλες τις απαραίτητες πληροφορίες για την επαναμετάδοση των πακέτων που χάθηκαν. Με τον τρόπο αυτό μπορεί να αντιμετωπίζει αποδοτικά απώλειες πακέτων στο ασύρματο κανάλι και να επαναμεταδίδει όλα τα χαμένα πακέτα σε ένα RTT, μειώνοντας έτσι τις λήξεις των χρονικών περιθωρίων του TCP.

ELN: Στο ELN [41] ο σταθμός βάση ανιχνεύει την απώλεια πακέτων στο ασύρματο δίκτυο και, τότε, ενημερώνει ρητά τον αποστολέα. Με βάση την ενημέρωση, ο αποστολέας μπορεί να επιλέξει το κατάλληλο έλεγχο συμφόρησης ανάλογα με την αιτία της απώλειας.

4.2.2. Μηχανισμοί διαχωρισμού σύνδεσης

Με τη μέθοδο αυτή υποθέτουμε πως ένας αποστολέας TCP είναι συνδεδεμένος σε έναν κινητό κόμβο-παραλήπτη μέσω ενός σταθμού βάση. Ο αποστολέας είναι «κρυμμένος» από το ασύρματο δίκτυο, τερματίζοντας την TCP σύνδεση στο σταθμό βάση και χρησιμοποιώντας μια ξεχωριστή αξιόπιστη σύνδεση ανάμεσα στο σταθμό βάση και τον παραλήπτη. Υπάρχει, δηλαδή, μια σύνδεση TCP ανάμεσα στον αποστολέα και το σταθμό βάση και μια δεύτερη ανάμεσα στο σταθμό βάση και τον παραλήπτη. Η κάθε σύνδεση μπορεί να χρησιμοποιεί επιλεκτικές (SACK) ή αρνητικές (NAK) επιβεβαιώσεις ή κάποιο εξειδικευμένο πρωτόκολλο που

πετυχαίνει καλή απόδοση πάνω από το ασύρματο δίκτυο. Ο διαχωρισμός της στοίβας πρωτοκόλλων στο σταθμό βάση οδηγεί σε διαχωρισμό των μηχανισμών ελέγχου ροής και συμφόρησης ανάμεσα στην ενσύρματη και την ασύρματη σύνδεση. Η αξιοπιστία από άκρο σε άκρο επιτυγχάνεται από το επίπεδο εφαρμογής. Μερικές από τις προτάσεις που βασίζονται στο μηχανισμό αυτό είναι τα εξής πρωτόκολλα: I-TCP, M-TCP, METP, WAP κ.α.



Σχήμα 4.4 Μηχανισμός διαχωρισμού σύνδεσης

Indirect TCP: Το πρωτόκολλο Indirect TCP (I-TCP) [46] προτάθηκε ως η πρώτη μέθοδος διαχωρισμού σύνδεσης. Στο πρωτόκολλο αυτό ένας κινητός δρομολογητής εξυπηρέτησης MSR (Mobile Support Router) χρησιμοποιείται ως κεντρικό σημείο, λειτουργώντας ως σταθμός βάση. Τα πακέτα που αποστέλλονται από τον αποστολέα αποθηκεύονται στον καταχωρητή του MSR και μετά αποστέλλονται στον κινητό κόμβο. Όταν ο κινητός κόμβος-παραλήπτης κινείται σε διαφορετικές κυψελίδες, οι πληροφορίες που έχει ο MSR-1 σχετικά με τους κόμβους, μεταδίδονται στον MSR-2.

Το πρωτόκολλο I-TCP βοηθάει στο διαχωρισμό του αποστολέα από την ασύρματη σύνδεση, προστατεύοντας έτσι τον αποστολέα από τα χαρακτηριστικά του ασύρματου δικτύου. Στο I-TCP η σημασιολογία από-άκρο-σε-άκρο της σύνδεσης TCP χάνεται, αφού οι επιβεβαιώσεις μπορούν να φτάσουν στον αποστολέα πριν ακόμη παραδοθεί το πακέτο. Κατά τη μετάβαση σε έναν άλλο κινητό δρομολογητή η κατάσταση της σύνδεσης μεταφέρεται στον επόμενο σταθμό βάση. Συνήθως το ασύρματο κανάλι έχει τη μικρότερη χωρητικότητα αναλογικά με το ενσύρματο κομμάτι της σύνδεσης και, έτσι, οι κινητοί δρομολογητές-σταθμοί βάσεις είναι

πολύπλοκοι και θα πρέπει να έχουν μεγάλους χώρους αποθήκευσης δεδομένων για να αντιμετωπίσουν υψηλά επίπεδα συμφόρησης. Αν οι μεταβάσεις αυτές είναι συχνές, τότε η επιβάρυνση που σχετίζεται με αυτή την ανταλλαγή πληροφοριών μπορεί να είναι μεγάλη και να προκαλέσει επιπλέον καθυστέρηση.

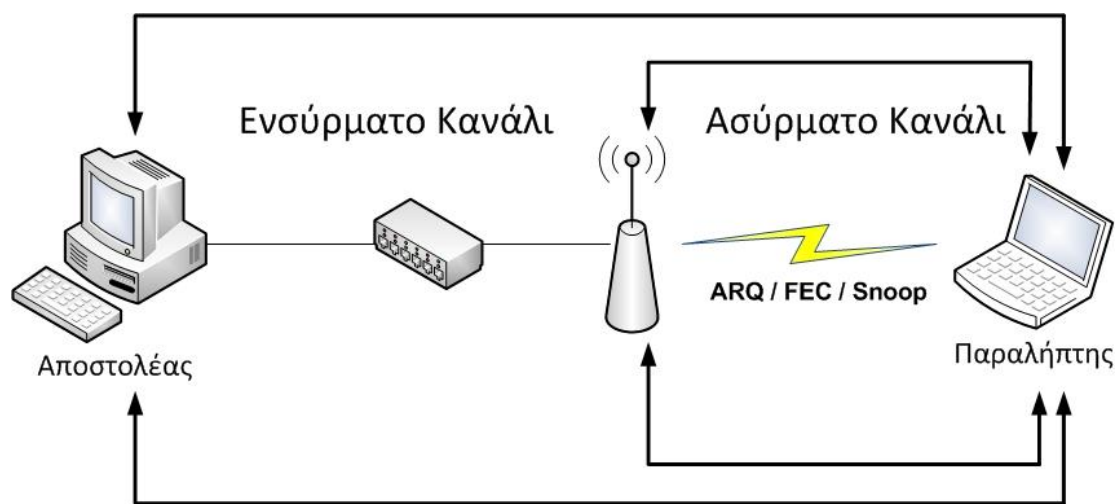
Mobile TCP: Το πρωτόκολλο αυτό υιοθετεί μια ιεραρχία τριών επιπέδων: κινητός κόμβος, κινητοί σταθμοί εξυπηρέτησης, κόμβος επίβλεψης και, τέλος, αποστολέας [47]. Ο κόμβος επίβλεψης και ο αποστολέας συνδέονται μέσω ενός ενσύρματου δικτύου, ενώ οι υπόλοιποι συνδέονται χρησιμοποιώντας το ασύρματο δίκτυο. Ο κόμβος επίβλεψης παραλαμβάνει ένα πακέτο από τον αποστολέα και το στέλνει στον κινητό κόμβο μέσω του σταθμού εξυπηρέτησης. Ο κόμβος επίβλεψης στέλνει μια επιβεβαίωση στον αποστολέα μετά τη λήψη κάθε επιβεβαίωσης από τον κινητό κόμβο. Το πρωτόκολλο είναι σχεδιασμένο ώστε να ανακάμπτει γρήγορα από ασύρματες απώλειες εξαιτίας διακοπών σύνδεσης και να εξαλείφει τα εκτεταμένα timeouts.

SRP: Το πρωτόκολλο επιλεκτικής επαναμετάδοσης SRP (Selective Repeat Protocol) η σύνδεση ανάμεσα στον αποστολέα και το σταθμό βάσης χρησιμοποιεί το κλασικό TCP, ενώ το πρωτόκολλο μεταφοράς ανάμεσα στο σταθμό βάσης και τον κινητό κόμβο τροποποιείται ειδικά για ασύρματα δίκτυα που χρησιμοποιούν επιλεκτικές επαναμεταδόσεις. Αντίθετα με το TCP, το πρωτόκολλο SRP έχει το σημαντικό πλεονέκτημα να μπορεί να ανακάμπτει από περισσότερες από μια απώλειες σε ένα μόνο παράθυρο.

4.2.3. Μηχανισμοί στο επίπεδο σύνδεσης

Η μέθοδος [48, 49] αυτή βρίσκεται ανάμεσα στις προηγούμενες δύο ομάδες. Στη περίπτωση αυτή, οι απώλειες που σχετίζονται με το κανάλι αποκρύπτονται από τον αποστολέα TCP, χρησιμοποιώντας τοπικές επαναμεταδόσεις των χαμένων πακέτων μέσω αυτόματης αίτησης επανάληψης ARQ (Automatic Repeat Request) [24, 25, 26, 27, 28, 29] και εκ των προτέρων διόρθωση λαθών FEC (Forward Error Correction) [34, 35]. Οι μηχανισμοί ARQ και FEC οδηγούν σε μείωση του ρυθμού σφαλμάτων bits BER (Bit Error Rate) και «ταιριάζουν» φυσικά στην αρχιτεκτονική επιπέδων της

στοίβας πρωτοκόλλων. Το κύριο πρόβλημα της προσέγγισης αυτής είναι η πιθανότητα ταυτόχρονης επαναμετάδοσης ανάμεσα στο TCP και το επίπεδο σύνδεσης. Άσκοπες διπλές επαναμεταδόσεις οδηγούν σε σπατάλη πολύτιμης χωρητικότητας και σημαντική μείωση της απόδοσης του TCP από άκρο σε άκρο. Μερικές προτάσεις που βασίζονται στους μηχανισμούς αυτούς είναι οι εξής: Snoop, TULIP, AIRMAIL, MALAW κ.α.



Σχήμα 4.5 Μηχανισμός στο επίπεδο σύνδεσης

Snoop: Το πρωτόκολλο [21, 22, 23] αυτό εισάγει έναν πράκτορα snoop στο σταθμό βάση, ο οποίος παρακολουθεί τα πακέτα που διασχίζουν τη σύνδεση TCP και προς τις δύο κατευθύνσεις. Διατηρεί μια λίστα των πακέτων TCP που έχουν σταλεί στο δίκτυο, αλλά δεν έχουν ακόμη επιβεβαιωθεί από τον παραλήπτη. Όταν ανιχνεύεται μια απώλεια πακέτου, με την άφιξη μιας διπλής επιβεβαίωσης από τον παραλήπτη ή τη λήξη ενός τοπικού χρονικού περιθωρίου, ο πράκτορας επαναμεταδίδει τα χαμένα πακέτα, εφόσον τα έχει αποθηκεύσει. Το γεγονός αυτό μειώνει τις διπλές επιβεβαιώσεις.

TULIP: Το πρωτόκολλο TULIP [50] έχει το χαρακτηριστικό ότι μπορεί να παρέχει αξιοπιστία μόνο για εκείνα τα πακέτα που το απαιτούν, χωρίς όμως να γνωρίζει λεπτομέρειες σχετικά με το πρωτόκολλο στο οποίο παρέχει αξιόπιστες υπηρεσίες. Πιο συγκεκριμένα, το TULIP παρέχει αξιόπιστες υπηρεσίες για πακέτα τα οποία μεταφέρουν δεδομένα TCP και μη αξιόπιστες υπηρεσίες για άλλους τύπους πακέτων όπως τα δεδομένα UDP (π.χ. ενημερώσεις πινάκων δρομολόγησης και

πακέτα DNS) και επιβεβαιώσεις TCP. Το TULIP δεν παρέχει αξιοπιστία στις επιβεβαιώσεις TCP, επειδή οι συνοπτικές επιβεβαιώσεις που ακολουθούν την απώλεια μιας επιβεβαίωσης συμπληρώνουν τις πληροφορίες που χάνονται με την απώλεια αυτής. Ο παραλήπτης αποθηκεύει τα πακέτα στον καταχωρητή του και τα στέλνει στο επόμενο επίπεδο με τη σειρά, αποφεύγοντας έτσι τη δημιουργία διπλών επιβεβαιώσεων στην περίπτωση που κάποιο πακέτο λείπει από την αναμενόμενη ακολουθία πακέτων. Η προσέγγιση αυτή εξαλείφει την ανάγκη για έναν εξυπηρετητή στο επίπεδο μεταφοράς, ο οποίος πρέπει να παρακολουθεί συνεχώς τα TCP πακέτα και να απορρίπτει οποιαδήποτε διπλή επιβεβαίωση συναντήσει.

Ένα σημαντικό χαρακτηριστικό του TULIP είναι η ικανότητα του για τοπικές επαναμεταδόσεις όσων πακέτων χάθηκαν στο ασύρματο δίκτυο, ώστε να αποφύγει μη αναγκαίες και καθυστερημένες επαναμεταδόσεις πακέτων πάνω από ολόκληρο το δίκτυο και την συνακόλουθη μείωση του παραθύρου συμφόρησης του TCP. Ο έλεγχος ροής της σύνδεσης εκτελείται από ένα κυλιόμενο παράθυρο και η αυτόματη επαναμετάδοση των χαμένων πακέτων επιτυγχάνεται από το επίπεδο σύνδεσης του αποστολέα. Τα χαμένα πακέτα ανιχνεύονται στον αποστολέα μέσω ενός διανύσματος bit, το οποίο επιστρέφει από τον παραλήπτη ως μέρος της κάθε επιβεβαίωσης. Αυτό επιτρέπει τη γρήγορη και αποδοτική ανάκαμψη των πακέτων και βοηθάει στη διατήρηση της καθυστέρησης και της απόκλισης της σε χαμηλά επίπεδα. Παρόλα αυτά, το TULIP έχει σχεδιαστεί για αποδοτική επικοινωνία πάνω από half-duplex radio channels, τα οποία είναι διαθέσιμα στα εμπορικά ραδιόφωνα σήμερα.

4.2.4. Σύγκριση των διαφόρων μηχανισμών

Στην περίπτωση πρωτοκόλλων που χρησιμοποιούν μηχανισμούς από άκρο σε άκρο απαιτείται τροποποίηση στον κώδικα του TCP στον κινητό κόμβο-παραλήπτη ή στον σταθερό κόμβο-αποστολέα και, έτσι, δεν είναι εύχρηστο από άποψη συμβατότητας. Αυτό θα οδηγήσει σε επανασύνδεση και επανασύνταξη της υπάρχουσας εφαρμογής στους σταθερούς κόμβους. Αυτό είναι το σημαντικότερο μειονέκτημα ενός μηχανισμού από άκρο σε άκρο.

Στην περίπτωση των μηχανισμών διαχωρισμού σύνδεσης παρέχεται ανάστροφη συμβατότητα με το υπάρχον πρωτόκολλο του ενσύρματου δικτύου. Εδώ δεν απαιτείται καμία τροποποίηση στον σταθερό κόμβο για την επικοινωνία του με τον κινητό κόμβο. Όμως, συχνά ο μηχανισμός αυτός υποφέρει από υψηλή επιβάρυνση στη περίπτωση των διπλών αποστολών στη στοίβα πρωτοκόλλων και απαιτεί πολύ χώρο για αποθήκευση στο σταθμό βάση. Αν και οι μηχανισμοί διαχωρισμού σύνδεσης λύνουν το πρόβλημα που παρουσιάζουν οι μηχανισμοί από άκρο σε άκρο και χειρίζονται αποδοτικά τις διακοπές σύνδεσης, δεν διατηρούν τη σημασιολογία της έννοιας από άκρο σε άκρο του TCP.

Οι μηχανισμοί σε επίπεδο σύνδεσης έχουν το σημαντικό πλεονέκτημα της διατήρησης της σημασιολογίας από άκρο σε άκρο του TCP. Οι άσκοπες λειτουργίες της γρήγορης επαναμετάδοσης και του ελέγχου συμφόρησης μειώνονται, καθώς ο σταθμός βάση μπορεί να χειριστεί τη κατάσταση τοπικά, ενώ η αξιοπιστία του ασύρματου δικτύου αυξάνεται, μειώνοντας το ρυθμό σφαλμάτων bit. Η μέθοδος αυτή λειτουργεί αποδοτικά ακόμη και σε περιβάλλοντα με υψηλό ρυθμό σφαλμάτων. Παρόλα αυτά, συχνές διακοπές συνδέσεις δεν υποστηρίζονται, επειδή στη περίπτωση που ο κινητός κόμβος αποσυνδεθεί, δεν θα παραλάβει τα πακέτα που επαναμεταδόθηκαν. Αν και η μέθοδος αυτή έχει το πλεονέκτημα να ταιριάζει φυσικά στη δομή επιπέδων των δικτυακών πρωτοκόλλων, υπάρχει μεγάλη πιθανότητα να συμβεί το αντίστροφο αποτέλεσμα, χρησιμοποιώντας συγκεκριμένα πρωτόκολλα μεταφοράς.

Γενικότερα, καθένας από τους παραπάνω μηχανισμούς ποικίλει με βάση την αλληλεπίδραση με το TCP, το επίπεδο δικτύου, το επίπεδο σύνδεσης, τη μέθοδο των επαναμεταδόσεων, τον έλεγχο ροής, τη κατάσταση του δικτύου, τη σημασιολογία από-άκρο-σε-άκρο, τη καταλληλότητα σε διαφορετικά δικτυακά περιβάλλοντα, την επιβάρυνση, την επίδραση της καθυστέρησης του καναλιού στο RTT, την ανίχνευση των απωλειών κ.α.

Ο πίνακας που ακολουθεί περιλαμβάνει τα στοιχεία του δικτύου (αποστολέας, σταθμός-βάση, παραλήπτης) στα οποία θα πρέπει να γίνουν αλλαγές για την εφαρμογή του κάθε πρωτοκόλλου.

Πρωτόκολλα	Αποστολέας	Σταθμός Βάση	Παραλήπτης
Από άκρο σε άκρο			
SMART	*		*
NAK	*		*
SACK	*		*
ELN	*		*
WTCP		*	*
Διαχωρισμού σύνδεσης			
I-TCP		*	*
M-TCP		*	*
SRP		*	*
Επιπέδου σύνδεσης			
Snoop		*	*
TULIP	*		*
FEC	*		*
ARQ	*	*	

Πίνακας 4.2 Τροποποιήσεις που απαιτούνται για την εφαρμογή του κάθε πρωτοκόλλου

Κεφάλαιο 5

Εργαστηριακή Μεθοδολογία

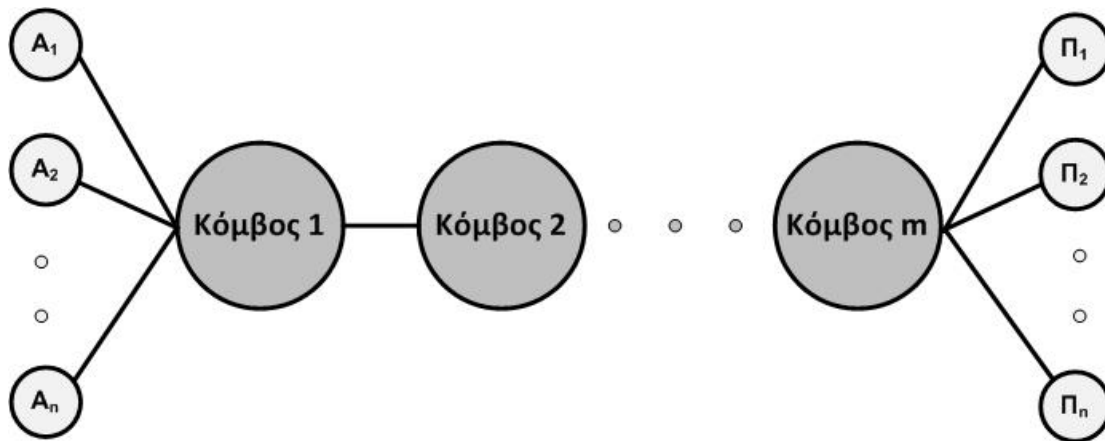
Σκοπός αυτού του κεφαλαίου είναι η περιγραφή του τρόπου διεξαγωγής των εργαστηριακών πειραμάτων, καθώς επίσης και της θεωρητικής ανάλυσης των αναμενόμενων αποτελεσμάτων. Τα πειράματα έχουν ως σκοπό στο σύνολο τους την αξιολόγηση της ταυτόχρονης χρήσης αξιόπιστων πρωτοκόλλων μεταφοράς και σύνδεσης σε δορυφορικό περιβάλλον. Για την μαθηματική αποτύπωση των 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 παραμέτρους 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 την διεξαγωγή των πειραμάτων. Τέλος, η θεωρητική ανάλυση των αναμενόμενων αποτελεσμάτων δίνει έναυσμα για σύγκριση ανάμεσα στα θεωρητικά και πειραματικά αποτελέσματα.

5.1 Τοπολογία των πειραμάτων και γενικές παράμετροι

Τα πειράματα εκτελούνται στον προσομοιωτή δικτύων Network Simulator 2 [51, 52]. Έχουν ως βάση μια απλή τοπολογία string, η οποία ανάλογα με το εκάστοτε πείραμα έχει:

- διαφορετικό πλήθος κόμβων,
- διαφορετικό πλήθος αποστολών,
- διαφορετικό πλήθος παραληπτών και
- διαφορετικές τιμές καθυστέρησης.

Μια ενδεικτική string τοπολογία με n πλήθος αποστολών ($A_1, A_2 \dots A_n$) και παραληπτών ($P_1, P_2 \dots P_n$) και m πλήθος κόμβων απεικονίζεται στο ακόλουθο σχήμα.



Σχήμα 5.1 Ενδεικτική τοπολογία πειραμάτων

Στην παραπάνω τοπολογία ο κάθε κόμβος αναπαριστά έναν δορυφόρο, οι συνδέσεις ανάμεσα στους κόμβους τη σύνδεση ανάμεσα σε δύο δορυφόρους (η χωρητικότητα της οποίας διαφέρει ανάλογα με το εκάστοτε πείραμα) και οι αποστολές και οι παραλήπτες τους αντίστοιχους κόμβους, οι οποίοι μπορεί να βρίσκονται είτε στη Γη (πχ. κέντρο διαστημικών επιχειρήσεων), είτε στο Διάστημα (πχ. αισθητήρας τοποθετημένος σε ένα ρομπότ στην επιφάνεια της Σελήνης).

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

TCP Version → Έκδοση TCP

Queue Type → Τύπος ουράς

Start Time → Χρόνος έναρξης προσομοίωσης

Finish Time → Χρόνος λήξης προσομοίωσης

Error Rate → Ρυθμός σφαλμάτων

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

5.2 Μονάδες Μέτρησης της Απόδοσης

5.2.1. 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 μετράται 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 επαναμετάδοσης 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 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, γεγονός που σημαίνει ότι το πρωτόκολλο δεν είναι σίγουρο ότι θα το καταλάβει πλήρως ή ότι δεν θα προσπαθήσει να το υπερκαλύψει με αποτέλεσμα να χαθούν πακέτα, λαμβάνοντας υπόψη την δυναμική συμπεριφορά του πρωτοκόλλου κατά την διάρκεια της επικοινωνίας.

Γενικά, το Throughput είναι μια αρκετά παρεξηγημένη έννοια. Μπορεί κανείς να το μετρήσει για μια 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 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.

5.2.2. 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.

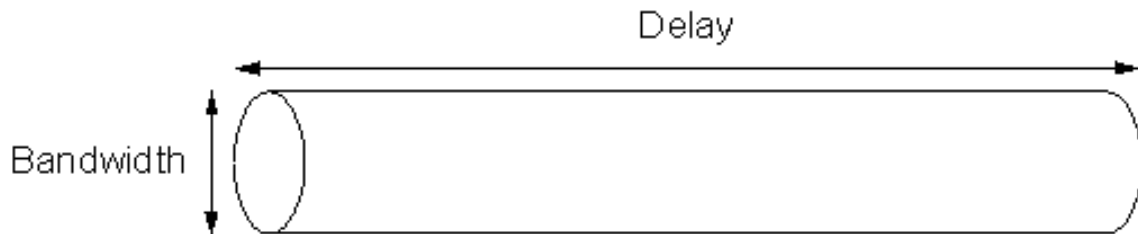
Καθυστέρηση επεξεργασίας (Processing Delay) (ms) είναι η καθυστέρηση που οφείλεται στην επικόλληση των επικεφαλίδων, στην υλοποίηση του δικτύου και στην ενδοεπικοινωνία διαφορετικών πρωτοκόλλων μέχρι να διοχετεύσουν τα δεδομένα στο δίκτυο.

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 την δυναμικότητα ενός δικτύου, 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 την 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 μια άλλη μονάδα, το RTT (Round Trip Time).



Σχήμα 5.2 Bandwidth X Delay Product

5.3. 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 την 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.

5.4. Σενάρια προσομοίωσης και θεωρητική ανάλυση

Στα πειράματα που εκτελέστηκαν έγινε η χρήση των παρακάτω σεναρίων, καθένα από τα οποία εξετάζει διαφορετικά χαρακτηριστικά του δικτύου. Στα σενάρια αναφοράς που αναφέρονται παρακάτω, μελετάται η συμπεριφορά των υπαρχόντων πρωτοκόλλων σύνδεσης και μεταφοράς σε επίγεια δίκτυα. Όσον αφορά τα σενάρια καινοτομίας, η συμπεριφορά των πρωτοκόλλων στις συγκεκριμένες συνθήκες δεν έχει μελετηθεί παλαιότερα, ωστόσο μπορούμε να σχολιάσουμε τα αναμενόμενα πειραματικά αποτελέσματα χρησιμοποιώντας θεωρητικές μεθόδους.

5.4.1. Επίγεια δίκτυα που εμφανίζουν σφάλματα

Σε αυτά τα σενάρια εξετάζεται η απόδοση των διαδικτυακών πρωτοκόλλων μεταφοράς και σύνδεσης για μικρές τιμές καθυστέρησης διάδοσης και μικρές τιμές ρυθμού σφαλμάτων. Τα σενάρια αυτά χρησιμεύουν ως σενάρια αναφοράς στα σενάρια που θα ακολουθήσουν.

Έχει αποδειχθεί από υπάρχουσες εργασίες η ανικανότητα του πρωτοκόλλου TCP να λειτουργήσει αποδοτικά πάνω από ασύρματα δίκτυα τα οποία εμφανίζουν σφάλματα. Για το λόγο αυτό, άλλωστε, προτάθηκαν οι διάφορες μέθοδοι που περιγράφηκαν στο προηγούμενο κεφάλαιο. Έτσι, από τα πειραματικά αποτελέσματα αναμένουμε η συμπεριφορά του πρωτοκόλλου TCP να μειώνεται σημαντικά για αυξανόμενο ρυθμό σφαλμάτων στο δίκτυο, γεγονός που αντισταθμίζεται από την ταυτόχρονη εφαρμογή αξιόπιστων μηχανισμών στο επίπεδο σύνδεσης.

5.4.2. Δίκτυα που εμφανίζουν συμφόρηση

Η συμφόρηση είναι ένα από τα σημαντικότερα προβλήματα στο Διαδίκτυο και έχει απασχολήσει τους ερευνητές για χρόνια. Στα σενάρια αυτά θα εξετάσουμε πως αντιμετωπίζουν τη συμφόρηση δύο αξιόπιστα πρωτόκολλα μεταφοράς και πως

μπορεί να βελτιώσει την απόδοση του συστήματος η ταυτόχρονη εφαρμογή ενός αξιόπιστου πρωτοκόλλου σύνδεσης.

Η λειτουργικότητα των πρωτοκόλλων σύνδεσης μελετήθηκε εκτενώς κατά την ανάπτυξη του Διαδικτύου, στη προσπάθεια να αντισταθμιστούν φαινόμενα συμφόρησης σε ασύρματα δίκτυα. Για τα λόγο αυτό αναμένουμε η απόδοση ενός δικτύου να βελτιώνεται με την ταυτόχρονη χρήση αξιόπιστων μηχανισμών σε δύο επίπεδα.

5.4.3. Επέκταση σε δορυφορικά περιβάλλοντα

Τα σημερινά πρωτόκολλα TCP έχουν σχεδιαστεί για ενσύρματες συνδέσεις, οι οποίες έχουν αμελητέο ρυθμό σφαλμάτων και μικρές τιμές καθυστερήσεων από-άκρο-σε-άκρο. Στις δορυφορικές 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 που κυμαίνονται σε ρυθμούς από 1%έως 30%. Θα εξετάσουμε, ακόμη, τη συμπεριφορά των πρωτοκόλλων μεταφοράς και σύνδεσης στην περίπτωση που η καθυστέρηση διάδοσης παίρνει πολύ μεγάλες τιμές.

Ουσιαστικά, σε αυτό το σετ πειραμάτων μελετούμε τη συμπεριφορά των υπαρχόντων πρωτοκόλλων μεταφοράς και σύνδεσης σε δορυφορικό περιβάλλον. Αναμένουμε οι μηχανισμοί αξιοπιστίας να αντισταθμίζουν τα σφάλματα που δημιουργούν οι δορυφορικές συνθήκες (απώλειες και φθορές πακέτων), χωρίς όμως να αντιμετωπίζουν επαρκώς το πρόβλημα της αυξημένης καθυστέρησης και, κατά συνέπεια, των καθυστερημένων επιβεβαιώσεων που οδηγούν σε μείωση του ρυθμού αποστολής.

5.4.4. Αύξηση του πλήθους των ενδιάμεσων κόμβων

Σκοπός αυτού του σετ πειραμάτων είναι ο έλεγχος της επίδρασης του πλήθους των ενδιάμεσων κόμβων ανάμεσα στον αποστολέα και τον παραλήπτη στην πραγματική απόδοση του συστήματος. Αυξάνοντας το πλήθος των ενδιάμεσων κόμβων και διατηρώντας σταθερή τη καθυστέρηση από-άκρο-σε-άκρο, αναμένουμε να πετύχουμε γρηγορότερες τοπικές επαναμεταδόσεις που θα οδηγήσουν σε βελτίωση της απόδοσης.

Για μεγάλες τιμές καθυστέρησης αναμένουμε η αύξηση του πλήθους των ενδιάμεσων κόμβων να οδηγήσει σε υψηλότερες τιμές απόδοσης. Αυτό δικαιολογείται από το γεγονός ότι η αύξηση του πλήθους των ενδιάμεσων κόμβων, διατηρώντας την ίδια τιμή καθυστέρησης από-άκρο-σε-άκρο, οδηγεί σε μείωση της καθυστέρησης ανάμεσα σε οποιουσδήποτε δύο κόμβους του δικτύου. Έτσι, οι τοπικές επαναμεταδόσεις από το επίπεδο σύνδεσης γίνονται γρηγορότερα, αποφεύγοντας τις συχνές λήξεις χρονικών περιθωρίων (timeouts) των χρονοπρογραμματιστών και επιτρέποντας τα πρωτόκολλα να λειτουργήσουν ομαλά.

Κεφάλαιο 6

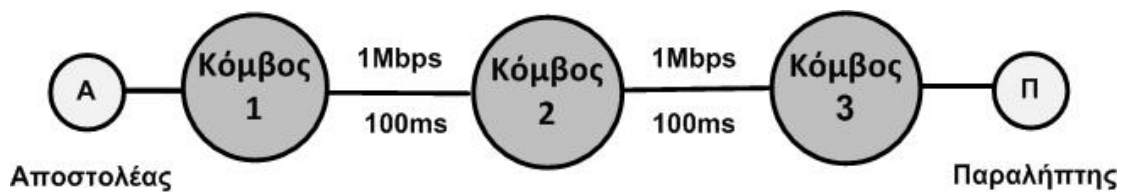
Πειραματικά Αποτελέσματα

Στο κεφάλαιο αυτό παρουσιάζονται τα αποτελέσματα από τα πειράματα που περιγράφηκαν στο προηγούμενο κεφάλαιο. Για κάθε πείραμα γίνεται μια σύντομη περιγραφή των παραμέτρων που ρυθμίσαμε, ακολουθούμενη από γραφικές αναπαραστάσεις των αποτελεσμάτων για την ευκολότερη κατανόηση τους. Σε κάθε πείραμα γίνεται σχολιασμός των αποτελεσμάτων και των παραμέτρων που οδήγησαν στο αποτέλεσμα. Στην τελευταία παράγραφο του κεφαλαίου συνοψίζονται και σχολιάζονται τα σημαντικότερα συμπεράσματα από το σύνολο των πειραμάτων.

6.1. Σενάρια αναφοράς

6.1.1. Επίγεια δίκτυα που εμφανίζουν σφάλματα

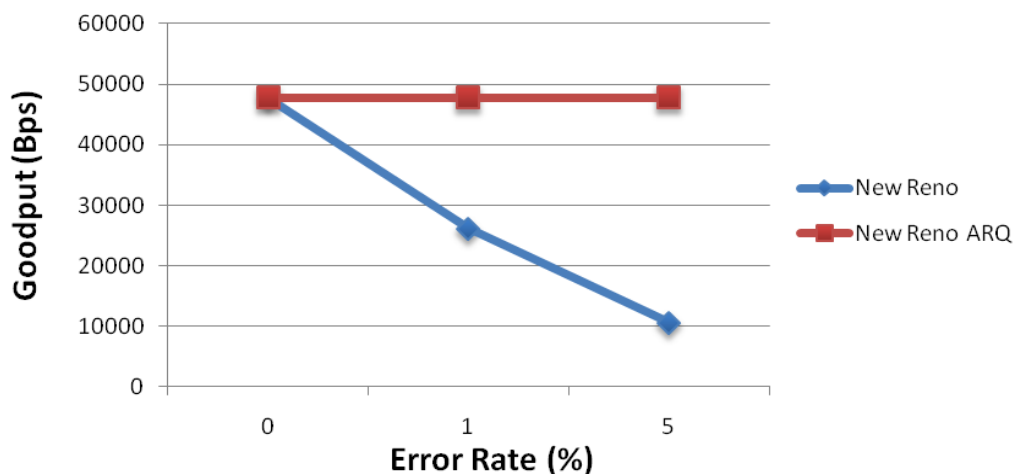
Σε αυτό το σετ πειραμάτων προσομοιώνεται η τοπολογία του σχήματος 6.1, όπου η χωρητικότητα της κάθε σύνδεσης είναι 1Mb. Στον πρώτο κόμβο της τοπολογίας είναι συνδεδεμένος ένας αποστολέας και στον τελευταίο κόμβο ένας παραλήπτης. Το πρωτόκολλο εφαρμογής που χρησιμοποιεί ο αποστολέας είναι το πρωτόκολλο μεταφοράς αρχείων δεδομένων FTP (file transfer protocol) και η διαχείριση της ουράς δεδομένων των καταχωρητών του δικτύου γίνεται βάσει της μεθόδου DropTail. Υποθέτουμε ότι το σενάριο μας προσομοιώνει ένα επίγειο ασύρματο δίκτυο με χαμηλή καθυστέρηση διάδοσης, το οποίο παρουσιάζει ποσοστό σφαλμάτων έως 5%. Προφανώς το δίκτυο δεν εμφανίζει συμφόρηση, αφού υπάρχει μόνο ένας αποστολέας. Τα πρωτόκολλα μεταφοράς που χρησιμοποιούνται είναι το TCP NewReno και το TCP SACK, ενώ όλα τα πειράματα εκτελούνται ξανά με τη χρήση των αξιόπιστων πρωτοκόλλων σύνδεσης ARQ και Snoop. Ο χρόνος της προσομοίωσης είναι αρκετά μεγάλος, ώστε να ολοκληρωθεί η αποστολή των δεδομένων.



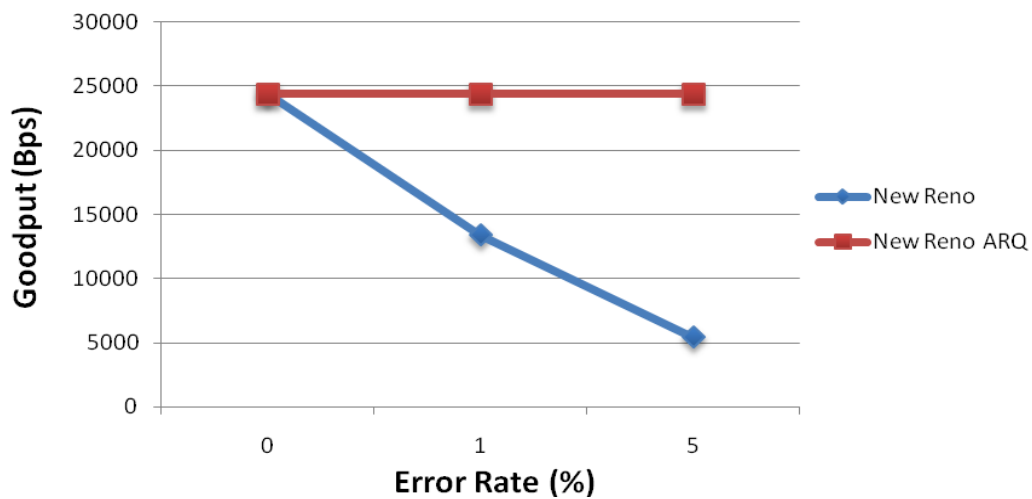
Σχήμα 6.1 Τοπολογία πειραμάτων

6.1.1.1. Πειράματα χρησιμοποιώντας πρωτόκολλο σύνδεσης ARQ και τα πρωτόκολλα μεταφοράς TCP New Reno και TCP SACK

Στις παρακάτω γραφικές παραστάσεις δίνεται η πραγματική απόδοση του συστήματος κατά την αποστολή ενός αρχείου μεγέθους 20MB, όταν η καθυστέρηση διάδοσης ισούται με 100ms και 200ms αντίστοιχα. Παρατηρούμε πως και στις δύο περιπτώσεις με την ταυτόχρονη χρήση του αξιόπιστου πρωτοκόλλου μεταφοράς TCP New Reno και του αξιόπιστου πρωτοκόλλου σύνδεσης ARQ επιτυγχάνουμε σημαντικά μεγαλύτερη πραγματική απόδοση. Η τιμή της απόδοσης, μάλιστα, παραμένει σχεδόν σταθερή, ανεξάρτητα από το ποσοστό απωλειών που εμφανίζεται στο δίκτυο. Αντίθετα, όταν δεν έχουμε αξιοπιστία στο επίπεδο σύνδεσης, παρατηρούμε μια ραγδαία πτώση της πραγματικής απόδοσης για αυξανόμενο ρυθμό σφαλμάτων. Η πτώση αυτή οφείλεται στην αδυναμία του πρωτοκόλλου New Reno να λειτουργήσει αποδοτικά για μεγάλους ρυθμούς σφαλμάτων.

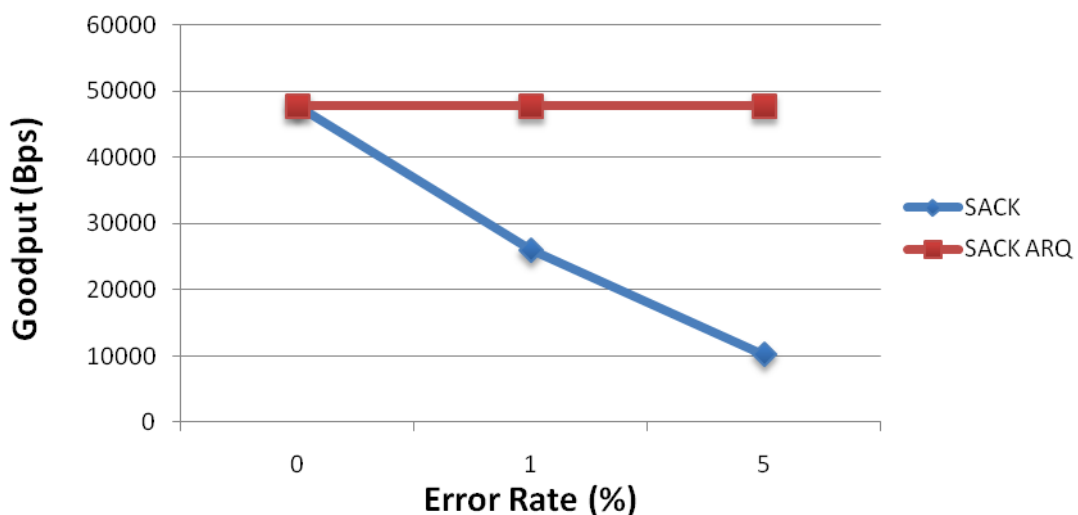


Σχήμα 6.2 Goodput - Error Rate για καθυστέρηση διάδοσης 100ms και αποστολή δεδομένων 20MB



Σχήμα 6.3 Goodput - Error Rate για καθυστέρηση διάδοσης 200ms και αποστολή δεδομένων 20MB

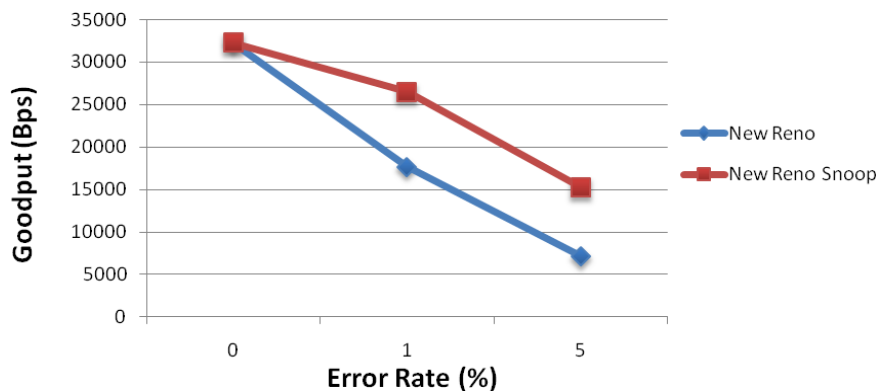
Στο παρακάτω διάγραμμα απεικονίζεται ένα αντίστοιχο πείραμα, αυτή τη φορά χρησιμοποιώντας το αξιόπιστο πρωτόκολλο μεταφοράς TCP SACK. Παρατηρούμε πως η συμπεριφορά του TCP SACK είναι παρόμοια με αυτή του TCP New Reno. Δηλαδή καμία από τις δύο εκδόσεις του πρωτοκόλλου μεταφοράς TCP δεν πετυχαίνει υψηλές τιμές απόδοσης για αυξανόμενο ρυθμό σφαλμάτων. Αντίθετα, με τη ταυτόχρονη χρήση ενός αξιόπιστου πρωτοκόλλου σύνδεσης, το οποίο αντιλαμβάνεται γρηγορότερα τις απώλειες και αποστέλλει άμεσα τα χαμένα δεδομένα, η τιμή της πραγματικής απόδοσης παραμένει σταθερή.



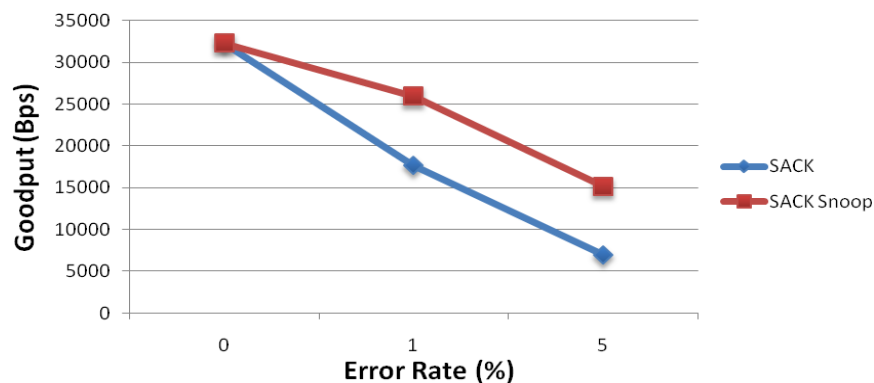
Σχήμα 6.4 Goodput - Error Rate για καθυστέρηση διάδοσης 100ms και αποστολή δεδομένων 20MB

6.1.1.2. Πειράματα χρησιμοποιώντας πρωτόκολλο σύνδεσης Snoop και τα πρωτόκολλα μεταφοράς TCP New Reno και TCP SACK

Αντίστοιχα με τα παραπάνω πειράματα, προσομοιώσαμε την αποστολή δεδομένων 20MB στην τοπολογία της εικόνας 6.1 χρησιμοποιώντας το αξιόπιστο πρωτόκολλο σύνδεσης Snoop σε συνδυασμό με τα πρωτόκολλα μεταφοράς TCP New Reno και TCP SACK. Πράκτορας Snoop θεωρείται ο πρώτος κόμβος της τοπολογίας. Παρατηρούμε πως και στις δύο περιπτώσεις, ακόμη και για ρυθμούς σφαλμάτων της τάξης του 1%, με την εφαρμογή του πρωτοκόλλου Snoop πετυχαίνουμε σημαντικά μεγαλύτερη πραγματική απόδοση. Παρατηρούμε, ωστόσο, πως για αυξανόμενο ρυθμό σφαλμάτων με τη χρήση του πρωτοκόλλου Snoop εμφανίζεται μια σημαντική πτώση στην απόδοση, σε αντίθεση με τη περίπτωση του πρωτοκόλλου σύνδεσης ARQ, όπου η απόδοση παραμένει σταθερή. Αυτό οφείλεται στη ταχύτερη επαναμετάδοση των δεδομένων που χάθηκαν ή αλλοιώθηκαν μέσω του πρωτοκόλλου Automatic Repeat reQuest.



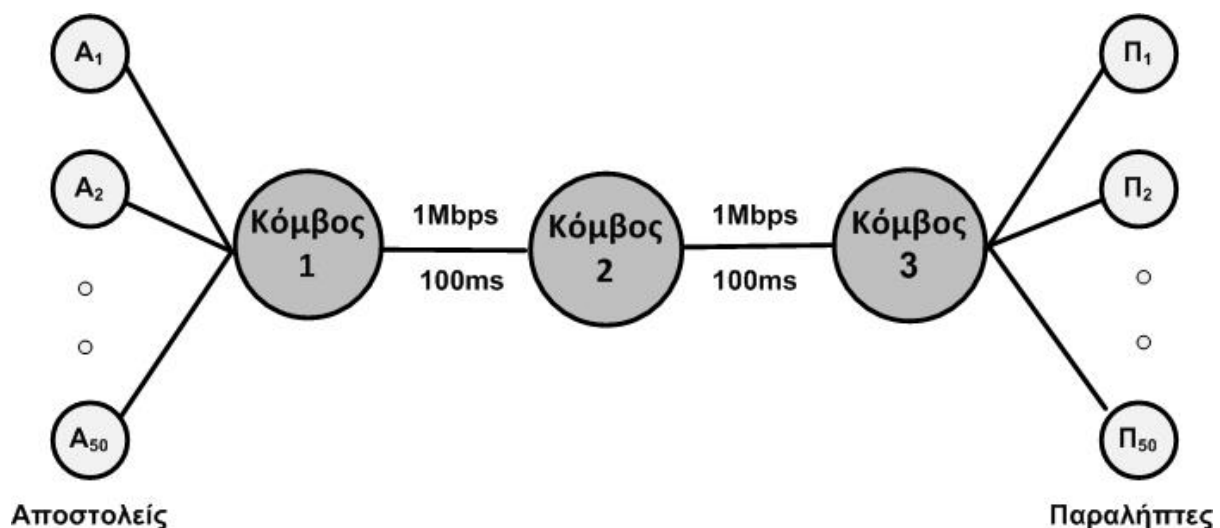
Σχήμα 6.5 Goodput - Error Rate για καθυστέρηση διάδοσης 100ms και αποστολή δεδομένων 20MB



Σχήμα 6.6 Goodput - Error Rate για καθυστέρηση διάδοσης 100ms και αποστολή δεδομένων 20MB

6.1.2. Δίκτυα που εμφανίζουν συμφόρηση

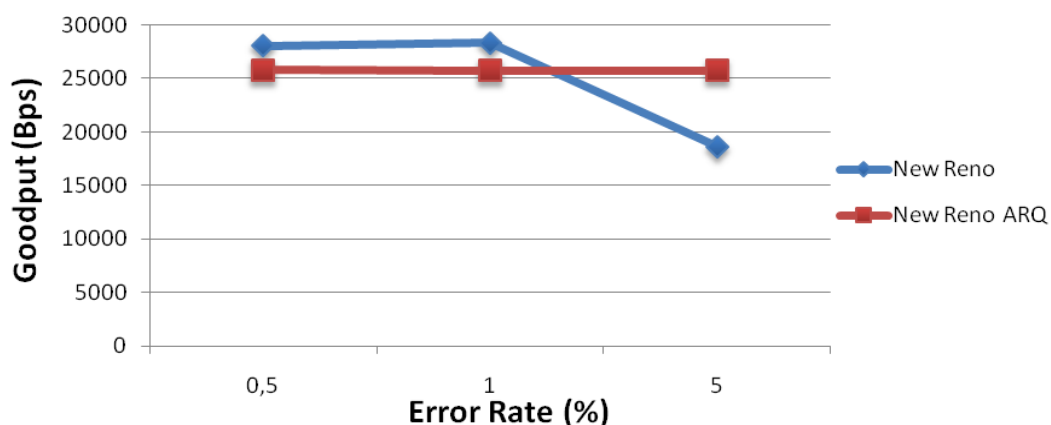
Στα πειράματα που ακολουθούν προσομοιώνεται η τοπολογία της εικόνας 6.7, όπου η χωρητικότητα της κάθε σύνδεσης είναι 1Mb. Στον πρώτο κόμβο της τοπολογίας είναι συνδεδεμένοι 50 αποστολείς, ενώ στον τελευταίο κόμβο 50 παραλήπτες. Το πρωτόκολλο εφαρμογής που χρησιμοποιούν οι χρήστες του δικτύου είναι το πρωτόκολλο μεταφοράς αρχείων δεδομένων FTP (file transfer protocol) και η διαχείριση της ουράς δεδομένων των καταχωρητών του δικτύου γίνεται βάσει της μεθόδου DropTail. Υποθέτουμε ότι το σενάριο μας παρουσιάζει μόνο μικρά ποσοστά σφαλμάτων που κυμαίνονται από 0,5% έως 5%. Αντίστοιχα, το επίπεδο συμφόρησης που εμφανίζεται στο δίκτυο κυμαίνεται ανάλογα με το μέγεθος των δεδομένων που μεταφέρονται στο δίκτυο. Συγκεκριμένα, οι αποστολείς στέλνουν ταυτόχρονα αρχεία μεγέθους 2MB, 5MB, 10MB και 20MB. Για αυξανόμενο μέγεθος αρχείου αυξάνεται το ποσοστό συμφόρησης που εμφανίζεται στο δίκτυο, αφού τα υπόλοιπα χαρακτηριστικά των συνδέσεων (καθυστέρηση, χωρητικότητα κτλ.) παραμένουν σταθερά. Το πρωτόκολλο μεταφοράς που χρησιμοποιείται σε κάθε περίπτωση είναι το TCP NewReno, ενώ τα πειράματα εκτελούνται δύο φορές, πρώτα με τη χρήση των αξιόπιστων πρωτοκόλλων σύνδεσης ARQ και Snoop και έπειτα χωρίς. Ο χρόνος της προσομοίωσης είναι αρκετά μεγάλος, ώστε να ολοκληρώσουν όλοι οι χρήστες τις αποστολές τους.



Σχήμα 6.7 Τοπολογία πειραμάτων

6.1.2.1. Πειράματα χρησιμοποιώντας πρωτόκολλο σύνδεσης ARQ και το πρωτόκολλο μεταφοράς TCP New Reno

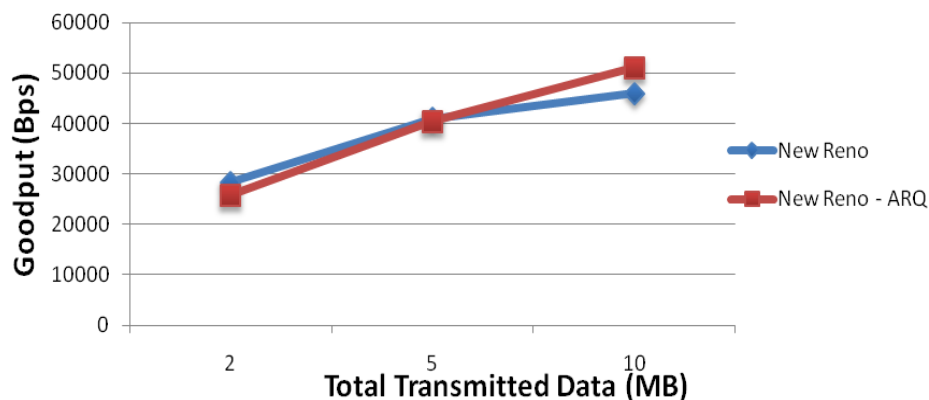
Παρακάτω δίνεται η γραφική παράσταση του Goodput συναρτήσει του ρυθμού σφαλμάτων για μικρή συμφόρηση στο δίκτυο (ο κάθε αποστολέας στέλνει ένα αρχείο μεγέθους 2MB) για καθυστέρηση διάδοσης ίση με 100ms, χρησιμοποιώντας το πρωτόκολλο μεταφοράς New Reno και το αξιόπιστο πρωτόκολλο σύνδεσης ARQ. Παρατηρούμε μια διαφοροποίηση στη συμπεριφορά των πρωτοκόλλων ανάλογα με το ποσοστό σφαλμάτων που εμφανίζει το δίκτυο. Συγκεκριμένα, για πολύ μικρό ρυθμό σφαλμάτων 0,5% ή 1% με τη ταυτόχρονη χρήση αξιόπιστων πρωτοκόλλων σε δύο επίπεδα πετυχαίνουμε ελαφρώς χειρότερη απόδοση από ότι εφαρμόζοντας αξιοπιστία μόνο στο επίπεδο μεταφοράς. Αυτό οφείλεται στο γεγονός ότι το ποσοστό σφαλμάτων είναι τόσο μικρό που μπορεί να το αντιμετωπίσει επαρκώς το πρωτόκολλο TCP. Στην περίπτωση αυτή προσθέτοντας αξιοπιστία στο επίπεδο σύνδεσης, προσθέτουμε επιβάρυνση χωρίς ουσιαστικό κέρδος, αφού το TCP αρκεί για την αντιμετώπιση των απωλειών. Αντίθετα, για μεγαλύτερο ρυθμό σφαλμάτων (5%) παρατηρούμε ότι η απόδοση του TCP μειώνεται σημαντικά, σε αντίθεση με τη σταθερή απόδοση που σημειώνεται όταν χρησιμοποιούμε αξιόπιστους μηχανισμούς σε δύο επίπεδα.



Σχήμα 6.8 Goodput - Error Rate για καθυστέρηση διάδοσης 100ms και αποστολή δεδομένων 2MB

Για αυξανόμενη συμφόρηση στο δίκτυο, διατηρώντας το ποσοστό σφαλμάτων σε χαμηλά επίπεδα της τάξης του 1%, παρατηρούμε ότι το κέρδος από την εφαρμογή αξιόπιστων μηχανισμών σε δύο επίπεδα εξαρτάται από το μέγεθος της

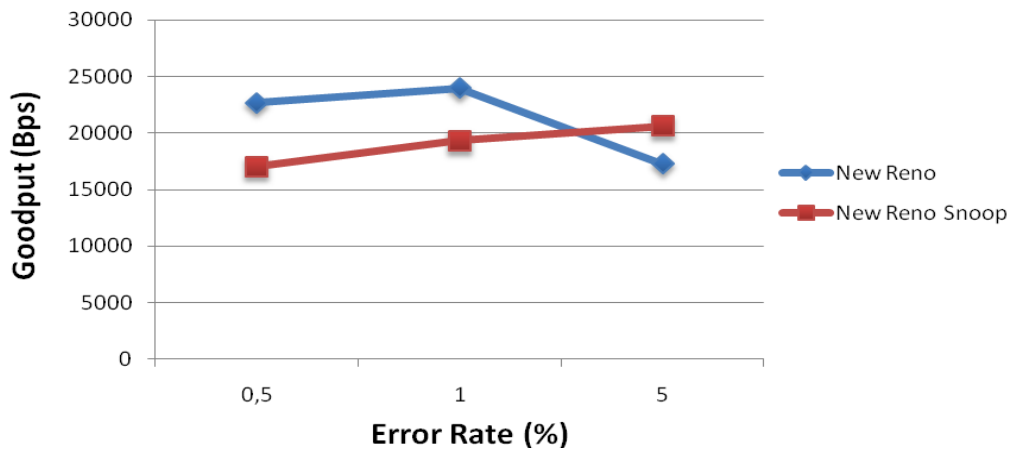
συμφόρησης. (Σχήμα 6.9) Για μικρό μέγεθος συμφόρησης (αποστολή 2MB ή 5MB) παρατηρούμε, όπως προηγουμένως, ότι η απόδοση του συστήματος δεν αυξάνεται χρησιμοποιώντας αξιοπιστία σε δύο επίπεδα. Όμως, για μεγαλύτερη συμφόρηση στο δίκτυο (αποστολή 10MB) παρατηρούμε σημαντικό κέρδος σε όρους απόδοσης. Έτσι, καταλήγουμε στο συμπέρασμα πως η χρήση αξιόπιστων πρωτοκόλλων σύνδεσης είναι αποδοτική μόνο για δίκτυα που εμφανίζουν μεγάλα ποσοστά συμφόρησης.



Σχήμα 6.9 Goodput – Total Transmitted Data για καθυστέρηση διάδοσης 100ms

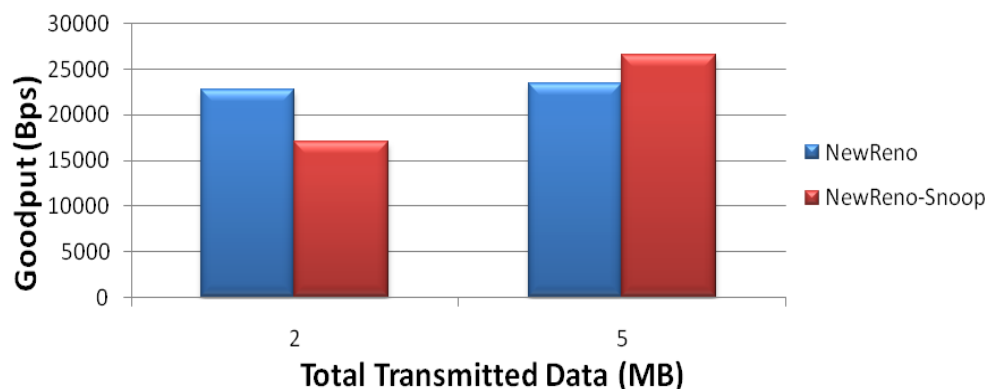
6.1.2.2. Πειράματα χρησιμοποιώντας πρωτόκολλο σύνδεσης Snoop και το πρωτόκολλο μεταφοράς TCP New Reno

Αρχικά σημειώνεται ότι πράκτορας Snoop θεωρείται ο πρώτος κόμβος της τοπολογίας. Παρακάτω δίνεται η γραφική παράσταση του Goodput συναρτήσεως του ρυθμού σφαλμάτων για μικρό ποσοστό συμφόρησης (αποστολή δεδομένων μεγέθους 2MB) και καθυστέρηση διάδοσης 100ms, χρησιμοποιώντας το πρωτόκολλο μεταφοράς New Reno και το πρωτόκολλο σύνδεσης Snoop. Παρατηρούμε ότι για πολύ χαμηλούς ρυθμούς σφαλμάτων δεν είναι αποδοτική η εφαρμογή του αξιόπιστου πρωτοκόλλου σύνδεσης Snoop. Η συμπεριφορά αυτή εξηγείται, ομοίως με τη περίπτωση εφαρμογής του ARQ, στην επιβάρυνση που συνεπάγεται η εφαρμογή αξιοπιστίας σε δύο επίπεδα για πολύ μικρό ρυθμό σφαλμάτων. Αντίθετα, για ποσοστό σφάλματος της τάξης του 1%, με την εφαρμογή του πρωτοκόλλου Snoop πετυχαίνουμε υψηλότερες τιμές απόδοσης, καθιστώντας την εφαρμογή του αποδοτική.



Σχήμα 6.10 Goodput - Error Rate για καθυστέρηση διάδοσης 100ms και αποστολή δεδομένων 2MB

Αντίστοιχα με τα προηγούμενα πειράματα για το πρωτόκολλα ARQ, διατηρώντας το ποσοστό σφαλμάτων σε πάρα πολύ χαμηλά επίπεδα της τάξης του 0,5%, παρατηρούμε ότι το κέρδος από την εφαρμογή αξιόπιστων μηχανισμών σε δύο επίπεδα εξαρτάται από το μέγεθος της συμφόρησης. (Σχήμα 6.11) Συγκεκριμένα, για μικρό μέγεθος συμφόρησης (αποστολή δεδομένων 2MB) παρατηρούμε ότι η απόδοση του συστήματος μειώνεται χρησιμοποιώντας αξιοπιστία σε δύο επίπεδα. Το γεγονός αυτό που οφείλεται στην επιπλέον επιβάρυνση που προσθέτει στο δίκτυο η εφαρμογή αξιοπιστίας στο επίπεδο σύνδεσης, αφού ουσιαστικά δεν χρησιμοποιείται από τη στιγμή που το δίκτυο δεν εμφανίζει συμφόρηση. Αντίθετα, για μεγαλύτερη συμφόρηση στο δίκτυο (αποστολή 10MB) παρατηρούμε σημαντικό κέρδος σε όρους απόδοσης. Τελικά, καταλήγουμε και πάλι στο συμπέρασμα πως για δίκτυα που εμφανίζουν συμφόρηση είναι αποδοτική η χρήση αξιόπιστων πρωτοκόλλων σύνδεσης μόνο για μεγάλα ποσοστά συμφόρησης.



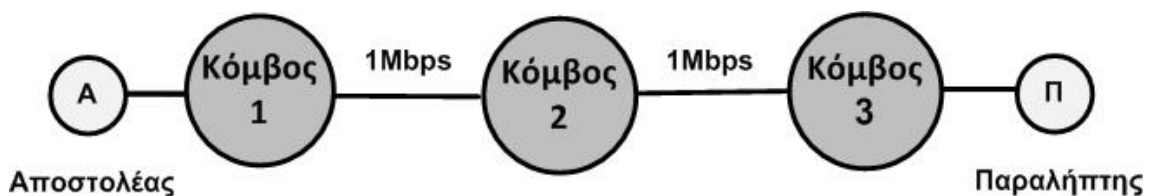
Σχήμα 6.11 Goodput - Total Transmitted Data για καθυστέρηση διάδοσης 100ms

6.2. Σενάρια καινοτομίας

6.2.1. Επέκταση σε δορυφορικά περιβάλλοντα

6.2.1.1. Πειράματα χρησιμοποιώντας πρωτόκολλο σύνδεσης ARQ και τα πρωτόκολλα μεταφοράς TCP New Reno και TCP SACK

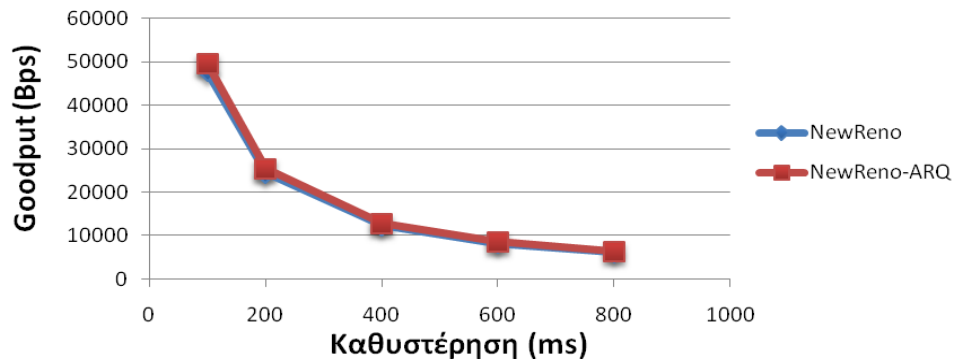
Στο σετ πειραμάτων που ακολουθεί προσομοιώνεται η τοπολογία της εικόνας 6.12, όπου η χωρητικότητα της κάθε σύνδεσης είναι 1Mb. Στον πρώτο κόμβο της τοπολογίας είναι συνδεδεμένος ένας αποστολέας και στον τελευταίο κόμβο ένας παραλήπτης. Το πρωτόκολλο εφαρμογής που χρησιμοποιεί ο αποστολέας είναι το πρωτόκολλο μεταφοράς αρχείων δεδομένων FTP (file transfer protocol) και η διαχείριση της ουράς δεδομένων των καταχωρητών του δικτύου γίνεται βάσει της μεθόδου DropTail. Υποθέτουμε ότι το σενάριο μας προσομοιώνει ένα δορυφορικό δίκτυο με καθυστέρηση διάδοσης που κυμαίνεται από 100ms έως 800ms. Όλα τα πειράματα εκτελούνται για ρυθμό σφαλμάτων που φτάνει το 30% των συνολικών δεδομένων που αποστέλλονται. Τα πρωτόκολλα μεταφοράς που χρησιμοποιούνται είναι το TCP NewReno και το TCP SACK, ενώ όλα τα πειράματα εκτελούνται ξανά με τη χρήση των αξιόπιστων πρωτοκόλλων σύνδεσης ARQ και Snoop. Ο χρόνος της προσομοίωσης είναι αρκετά μεγάλος, ώστε να ολοκληρωθεί η αποστολή των δεδομένων.



Σχήμα 6.12 Τοπολογία πειραμάτων

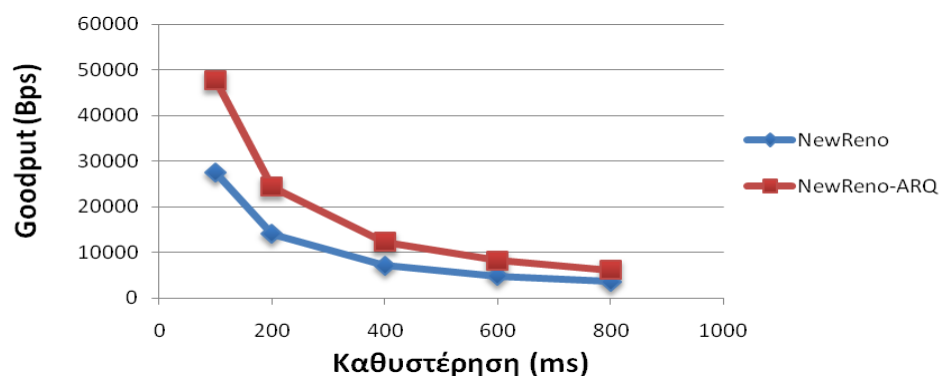
Στις γραφικές παραστάσεις που ακολουθούν απεικονίζεται η πραγματική απόδοση σε σχέση με την καθυστέρηση για αυξανόμενο ρυθμό σφαλμάτων. Συνολικά παρατηρούμε ότι καθώς αυξάνεται η καθυστέρηση διάδοσης, η τιμή της απόδοσης μειώνεται σημαντικά. Το γεγονός αυτό είναι αναμενόμενο εξαιτίας της αδυναμίας του πρωτοκόλλου TCP να λειτουργήσει αποδοτικά σε συνθήκες υψηλής

καθυστερήσης οι επιβεβαιώσεις καθυστερούν να επιστρέψουν στον αποστολέα, ο οποίος εκλαμβάνει το γεγονός ως ένδειξη συμφόρησης και μειώνει άσκοπα το ρυθμό αποστολής. Στην περίπτωση του μηδενικού ρυθμού σφαλμάτων, όπως είναι αναμενόμενο, οι λειτουργίες του μηχανισμού ARQ δεν χρησιμοποιούνται και, συνεπώς, δεν παρατηρείται κάποια διαφορά στην απόδοση.



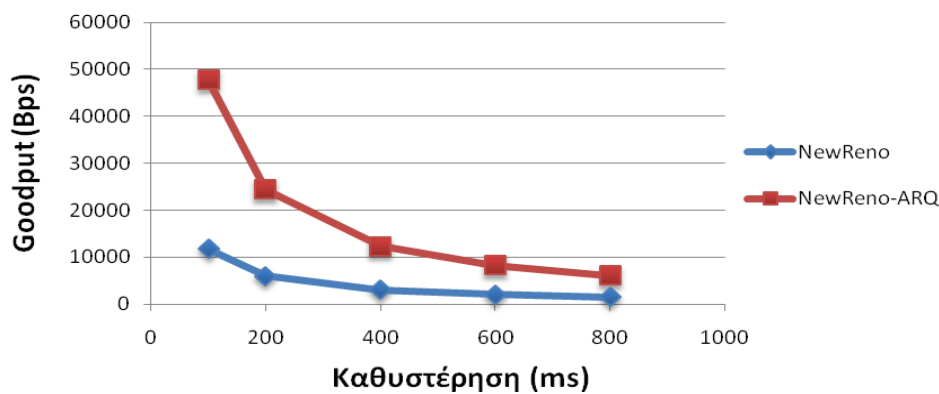
Σχήμα 6.13 Goodput – Καθυστερήση διάδοσης για μηδενικό ρυθμό σφαλμάτων

Ακόμη και για πολύ μικρό ρυθμό σφαλμάτων της τάξης του 1% παρατηρούμε μια διαφοροποίηση σε όρους απόδοσης ανάμεσα στις δύο περιπτώσεις. Με τη χρήση μόνο του αξιόπιστου πρωτοκόλλου μεταφοράς TCP New Reno παρατηρούμε μια σημαντική μείωση στην απόδοση, η οποία αυξάνεται για μεγαλύτερες τιμές καθυστέρησης. Αντίθετα, η ταυτόχρονη χρήση μηχανισμών αξιοπιστίας σε δύο επίπεδα διατηρεί την απόδοση σε σημαντικά υψηλότερα επίπεδα. Ουσιαστικά, οι μεταδόσεις δεδομένων από το επίπεδο σύνδεσης αντισταθμίζουν την αδυναμία αποδοτικής λειτουργίας του πρωτοκόλλου μεταφοράς σε περιβάλλοντα που εμφανίζουν σφάλματα και χαρακτηρίζονται από υψηλές τιμές καθυστερήσεων. Φυσικά, για αυξανόμενες τιμές καθυστέρησης η πραγματική απόδοση μειώνεται και στις δύο περιπτώσεις.

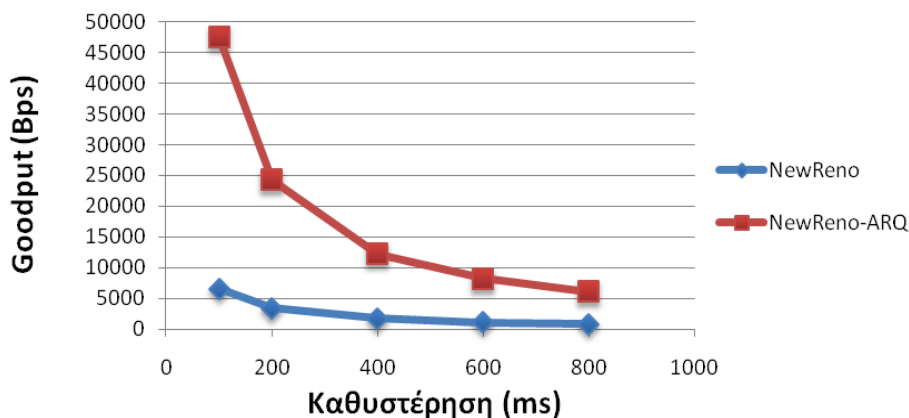


Σχήμα 6.14 Goodput – Καθυστερήση διάδοσης για ρυθμό σφαλμάτων 1%

Για αυξανόμενο ρυθμό σφαλμάτων στο δίκτυο παρατηρούμε μια αύξηση της διαφοράς σε όρους απόδοσης ανάμεσα στις δύο περιπτώσεις. Καθώς αυξάνονται τα σφάλματα στη κάθε σύνδεση, το πρωτόκολλο TCP αδυνατεί να αντισταθμίσει τις απώλειες, οδηγώντας σε συνεχή μείωση της απόδοσης. Αντίθετα, το πρωτόκολλο ARQ αντιμετωπίζει τις απώλειες αυτές, επαναμεταδίδοντας δεδομένα από το επίπεδο σύνδεσης, διατηρώντας την απόδοση σε σχεδόν σταθερά επίπεδα ανεξάρτητα από το ρυθμό σφαλμάτων ανά πακέτο.

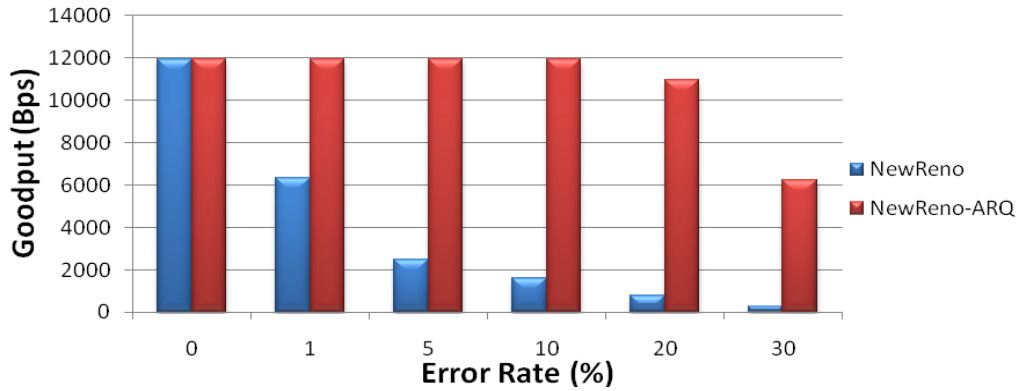


Σχήμα 6.15 Goodput – Καθυστέρηση διάδοσης για ρυθμό σφαλμάτων 5%

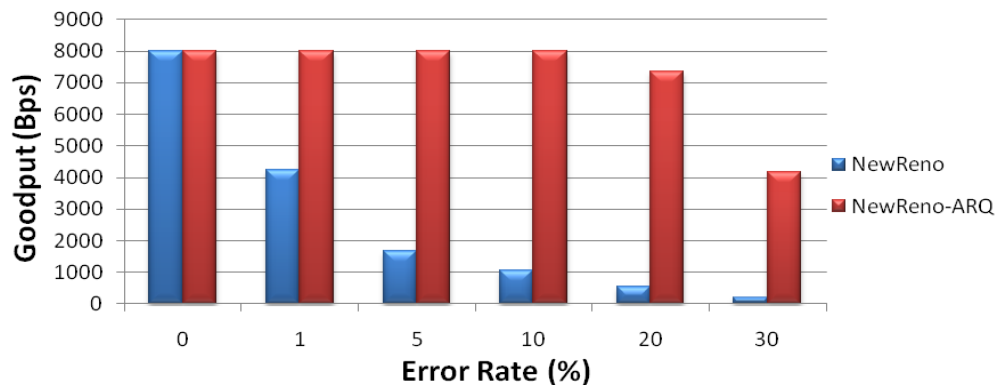


Σχήμα 6.16 Goodput – Καθυστέρηση διάδοσης για ρυθμό σφαλμάτων 10%

Στις γραφικές παραστάσεις που ακολουθούν μπορεί κανείς να παρατηρήσει καλύτερα τη σημαντική διαφορά σε όρους απόδοσης που πετυχαίνουμε με τη χρήση του αξιόπιστου πρωτοκόλλου σύνδεσης ARQ. Για μεγάλες τιμές καθυστέρησης διάδοσης, ακόμη και για ποσοστά σφαλμάτων της τάξης του 1%, η απόδοση που πετυχαίνουμε με τη χρήση αξιόπιστων πρωτοκόλλων σε δύο επίπεδα είναι σχεδόν διπλάσια από ότι στην περίπτωση χρήσης μόνο του αξιόπιστου πρωτοκόλλου NewReno.

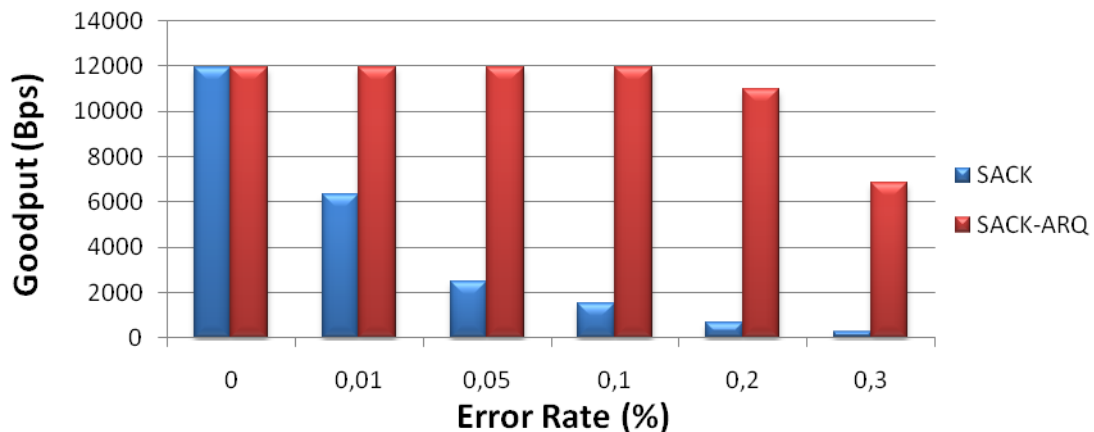


Σχήμα 6.17 Goodput – Ρυθμός σφαλμάτων για καθυστέρηση διάδοσης 400ms



Σχήμα 6.18 Goodput – Ρυθμός σφαλμάτων για καθυστέρηση διάδοσης 600ms

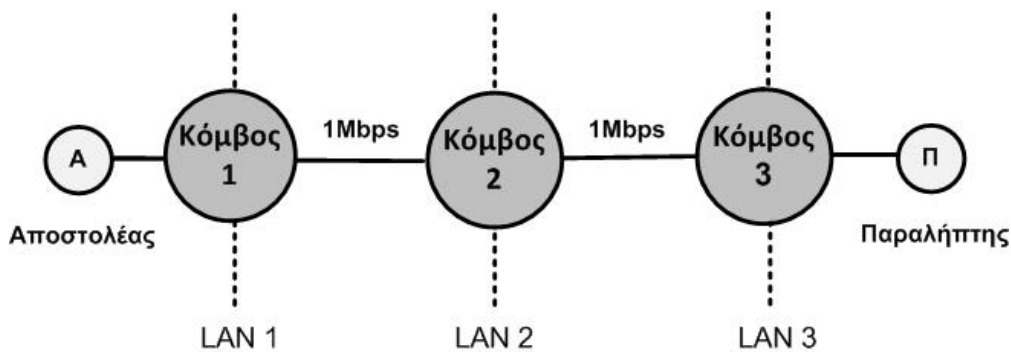
Αντίστοιχα αποτελέσματα παίρνουμε και με τη χρήση του αξιόπιστου πρωτοκόλλου μεταφοράς TCP SACK. Για διαφορετικές τιμές καθυστέρησης διάδοσης παρατηρούμε πως η απόδοση παραμένει σε υψηλά επίπεδα όταν χρησιμοποιούνται αξιόπιστα πρωτόκολλα σε δύο επίπεδα, σε αντίθεση με την περίπτωση χρήσης αξιόπιστου πρωτοκόλλου μεταφοράς μόνο, όπου παρατηρείται σημαντική μείωση της απόδοσης για αυξανόμενο ρυθμό σφαλμάτων.



Σχήμα 6.19 Goodput – Ρυθμός σφαλμάτων για καθυστέρηση διάδοσης 400ms

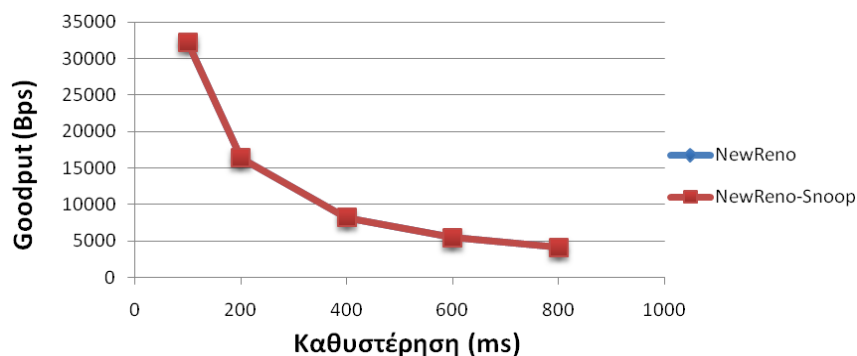
6.2.1.2. Πειράματα χρησιμοποιώντας πρωτόκολλο σύνδεσης Snoop και το πρωτόκολλο μεταφοράς TCP New Reno

Το σενάριο που ακολουθεί προσομοιώνει την τοπολογία της εικόνας 6.20, χρησιμοποιώντας ίδια χαρακτηριστικά με το προηγούμενο σετ πειραμάτων. Η μόνη διαφορά είναι πως αυτή τη φορά το πρωτόκολλο σύνδεσης που χρησιμοποιούμε είναι το πρωτόκολλο Snoop. Πράκτορας Snoop θεωρείται ο πρώτος κόμβος της τοπολογίας. Λόγω προβλημάτων υλοποίησης του πρωτοκόλλου Snoop στον προσομοιωτή δικτύων NS-2, στην παρακάτω τοπολογία οι ενδιάμεσοι κόμβοι υλοποιήθηκαν ως μέρος ενός διαφορετικού τοπικού δικτύου ο καθένας.

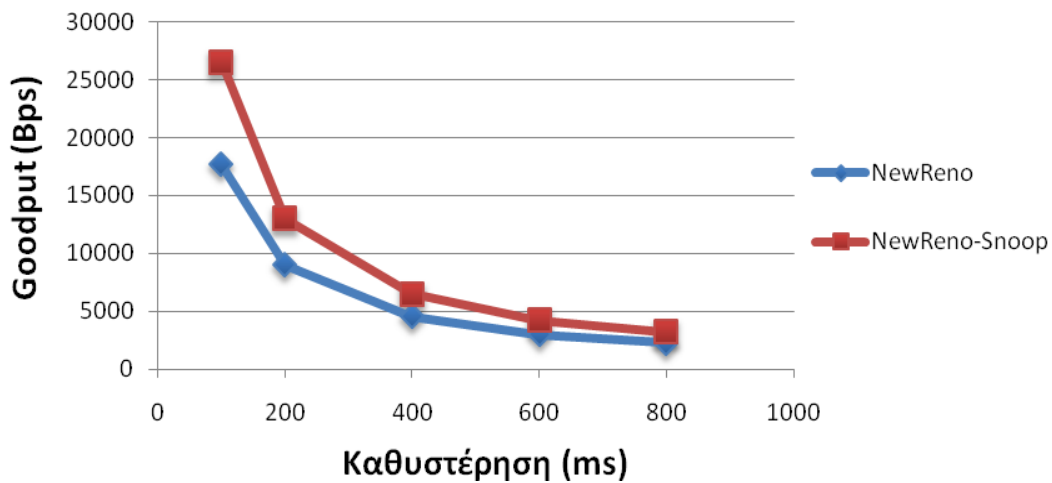


Σχήμα 6.20 Τοπολογία πειραμάτων

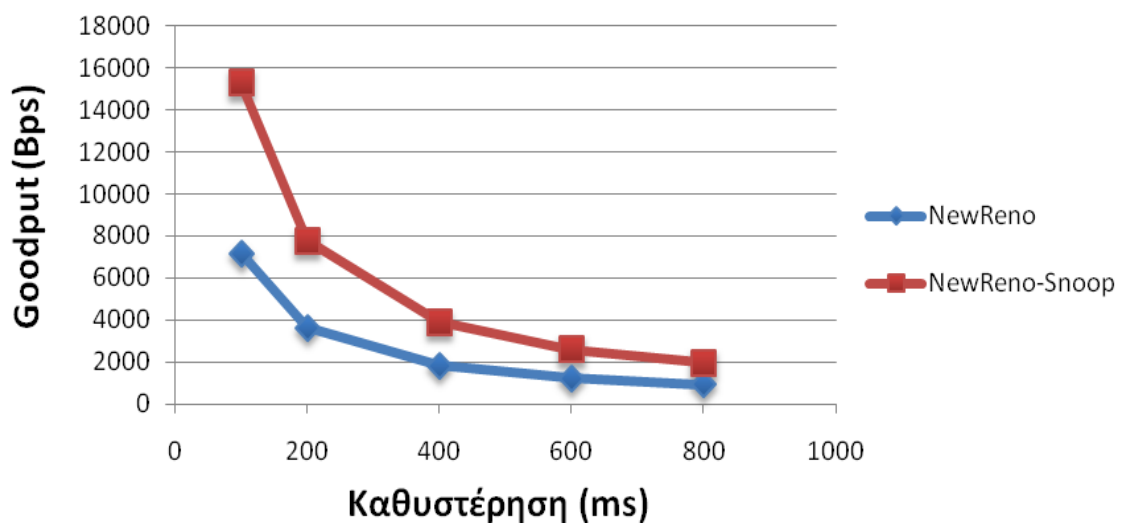
Στις παρακάτω γραφικές παραστάσεις φαίνεται η συμπεριφορά του δικτύου της παραπάνω τοπολογίας για αυξανόμενη καθυστέρηση διάδοσης και ρυθμό σφαλμάτων. Η υπεροχή σε όρους απόδοσης, ανεξάρτητα από το ρυθμό σφαλμάτων, όταν χρησιμοποιούνται ταυτόχρονα ένα αξιόπιστο πρωτόκολλο σύνδεσης και ένα αξιόπιστο πρωτόκολλο μεταφοράς είναι προφανής. Ωστόσο, παρατηρούμε ότι με τη χρήση του πρωτοκόλλου σύνδεσης Snoop πετυχαίνουμε χαμηλότερη απόδοση από ότι με τη χρήση του πρωτοκόλλου ARQ.



Σχήμα 6.21 Goodput – Καθυστέρηση διάδοσης για μηδενικό ρυθμό σφαλμάτων

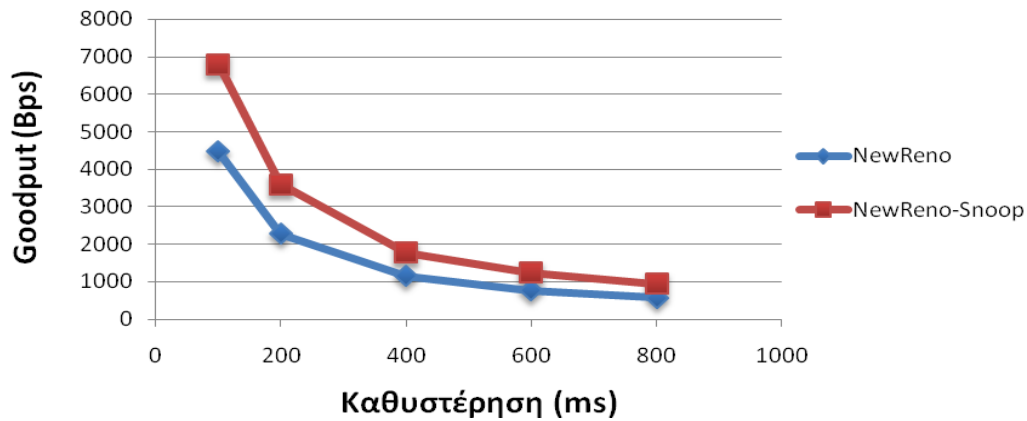


Σχήμα 6.22 Goodput – Καθυστερήση διάδοσης για ρυθμό σφαλμάτων 1%



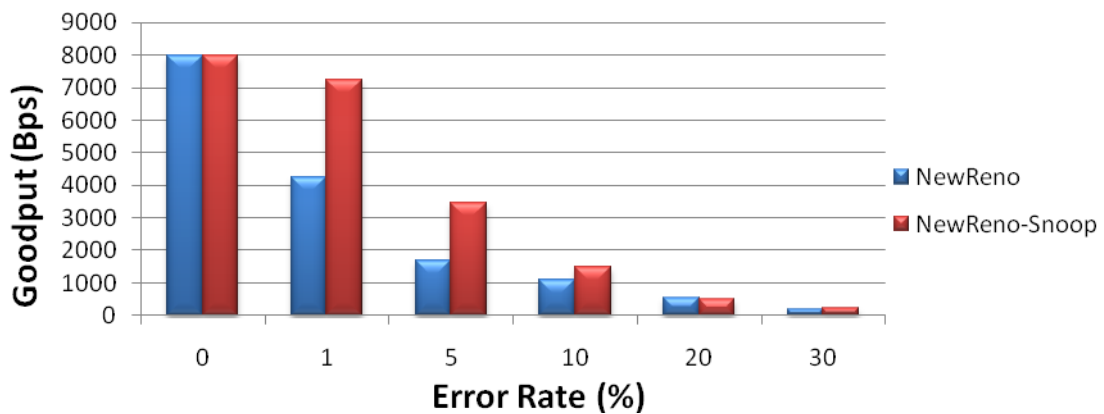
Σχήμα 6.23 Goodput – Καθυστερήση διάδοσης για ρυθμό σφαλμάτων 5%

Από τη γραφική παράσταση που ακολουθεί μπορεί εύκολα να συμπεράνει κάποιος πως για μεγάλες τιμές ποσοστών σφαλμάτων και καθυστερήσεις διάδοσης μεγαλύτερες από 400ms, η εφαρμογή αξιόπιστων μηχανισμών σε δύο επίπεδα δεν βελτιώνει την απόδοση του συστήματος. Αυτό οφείλεται στην αδυναμία και των δύο πρωτοκόλλων να λειτουργήσουν αποδοτικά για τόσο μεγάλες τιμές καθυστερήσεων. Για το λόγο αυτό μεταδίδουν μόνο ελάχιστα δεδομένα, γεγονός που οδηγεί σε πολύ χαμηλά επίπεδα απόδοσης.

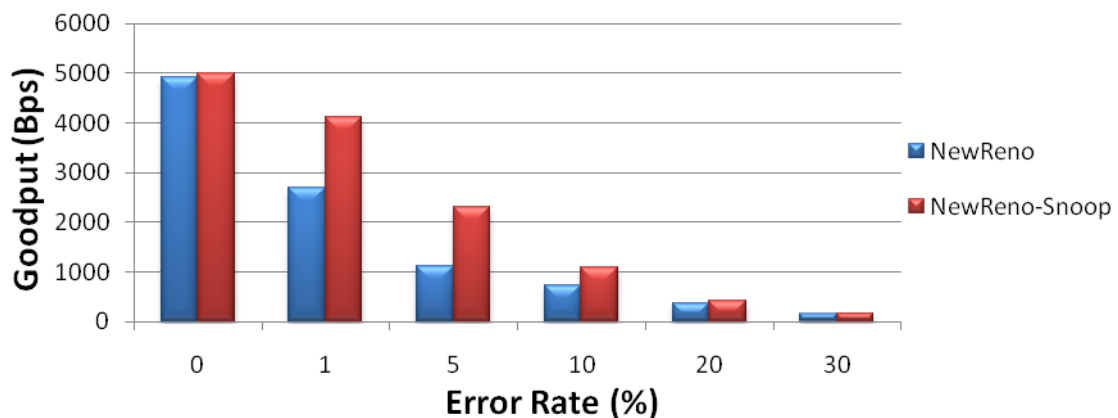


Σχήμα 6.24 Goodput – Καθυστέρηση διάδοσης για ρυθμό σφαλμάτων 10%

Στα γραφήματα που ακολουθούν απεικονίζονται συνοπτικά τα αποτελέσματα για διάφορες τιμές ρυθμού σφαλμάτων και για καθυστέρηση διάδοσης ίση με 400ms και 600ms. Παρατηρούμε μια συνεχή πτώση της απόδοσης με την αύξηση του ρυθμού σφαλμάτων, ενώ ακόμη και για πάρα πολύ μικρούς ρυθμούς σφαλμάτων με τη χρήση του Snoop στο επίπεδο σύνδεσης πετυχαίνουμε σχεδόν διπλάσια τιμή πραγματικής απόδοσης.



Σχήμα 6.25 Goodput – Ρυθμός σφαλμάτων για καθυστέρηση διάδοσης 400ms



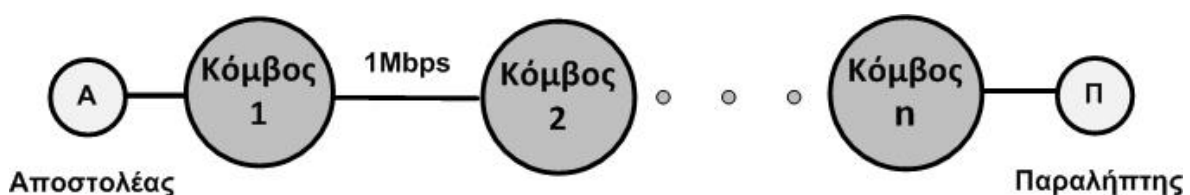
Σχήμα 6.26 Goodput – Ρυθμός σφαλμάτων για καθυστέρηση διάδοσης 600ms

6.2.2. Αύξηση του πλήθους των ενδιάμεσων κόμβων

Στα πειράματα που ακολουθούν θα ελέγξουμε την επίδραση του πλήθους των ενδιάμεσων κόμβων που υπάρχουν ανάμεσα στον αποστολέα και τον παραλήπτη στην τελική απόδοση του συστήματος. Ο έλεγχος θα γίνει επαναλαμβάνοντας ένα σετ σεναρίων, κάθε φορά αυξάνοντας το πλήθος των ενδιάμεσων κόμβων, για μεγάλες τιμές καθυστέρησης διάδοσης.

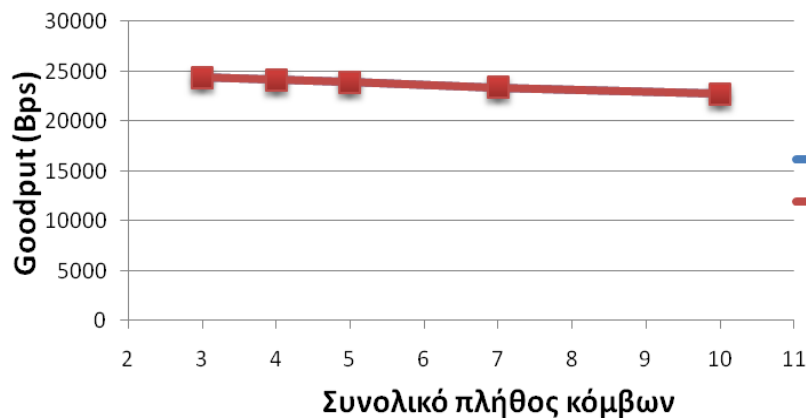
6.2.2.1. Πειράματα χρησιμοποιώντας πρωτόκολλο σύνδεσης ARQ και το πρωτόκολλο μεταφοράς TCP New Reno

Στο σενάριο που ακολουθεί προσομοιώνεται η τοπολογία της εικόνας 6.27, όπου η χωρητικότητα της κάθε σύνδεσης είναι 1Mb. Στον πρώτο κόμβο της τοπολογίας είναι συνδεδεμένος ένας αποστολέας και στον τελευταίο κόμβο ένας παραλήπτης. Το πρωτόκολλο εφαρμογής που χρησιμοποιείται είναι το πρωτόκολλο μεταφοράς αρχείων δεδομένων FTP (file transfer protocol) και η διαχείριση της ουράς δεδομένων των καταχωρητών γίνεται βάσει της μεθόδου DropTail. Υποθέτουμε ότι το σενάριο μας παρουσιάζει ποσοστά σφαλμάτων που κυμαίνονται από 0,5% έως 5%. Το πλήθος των κόμβων, για σταθερή καθυστέρησης διάδοσης από-άκρο-σε-άκρο, ξεκινάει από 3 κόμβους και φτάνει τους 10 κόμβους. Τα πρωτόκολλα μεταφοράς που χρησιμοποιούνται είναι το TCP NewReno και το TCP SACK, ενώ τα πειράματα εκτελούνται δύο φορές, πρώτα με τη χρήση του αξιόπιστου πρωτοκόλλου σύνδεσης ARQ και έπειτα χωρίς. Ο χρόνος της προσομοίωσης είναι αρκετά μεγάλος, ώστε να ολοκληρωθεί η αποστολή των δεδομένων (20 MB).

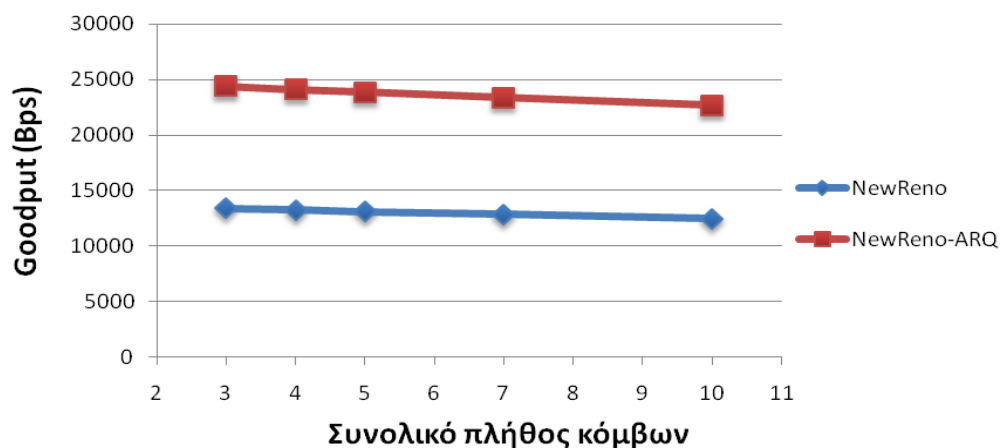


Σχήμα 6.27 Τοπολογία πειραμάτων

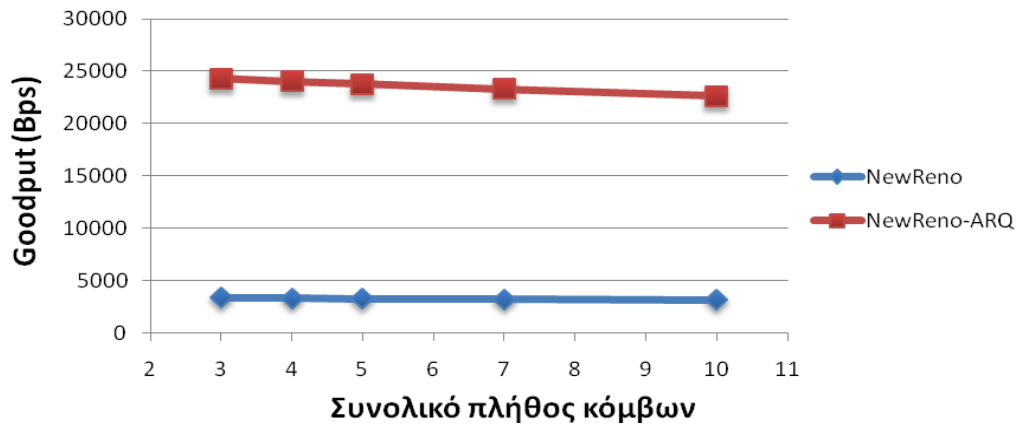
Στις γραφικές παραστάσεις που ακολουθούν απεικονίζεται η συμπεριφορά του συστήματος που περιγράφηκε παραπάνω για διαφορετικό ποσοστό σφάλματος και πλήθος κόμβων, διατηρώντας τη καθυστέρηση διάδοσης σταθερή από-άκρο-σε-άκρο και ίση με 400ms. Παρατηρούμε ότι για αυξανόμενο πλήθος ενδιάμεσων κόμβων σημειώνεται μια μικρή πτώση της πραγματικής απόδοσης του συστήματος. Η πτώση αυτή οφείλεται στην αύξηση των τοπικών επαναμεταδόσεων που συμβαίνουν κατά τη διάρκεια της αποστολής των δεδομένων και στον περιορισμό που θέτει το μέγεθος των καταχωρητών που χρησιμοποιούνται στο δίκτυο. Σε κάθε περίπτωση, πάντως, η απόδοση που πετυχαίνουμε χρησιμοποιώντας το πρωτόκολλο σύνδεσης ARQ είναι πολλαπλάσια της απόδοσης, όταν χρησιμοποιούμε μόνο αξιόπιστο πρωτόκολλο μεταφοράς.



Σχήμα 6.28 Goodput – Συνολικό πλήθος κόμβων για μηδενικό ποσοστό σφαλμάτων

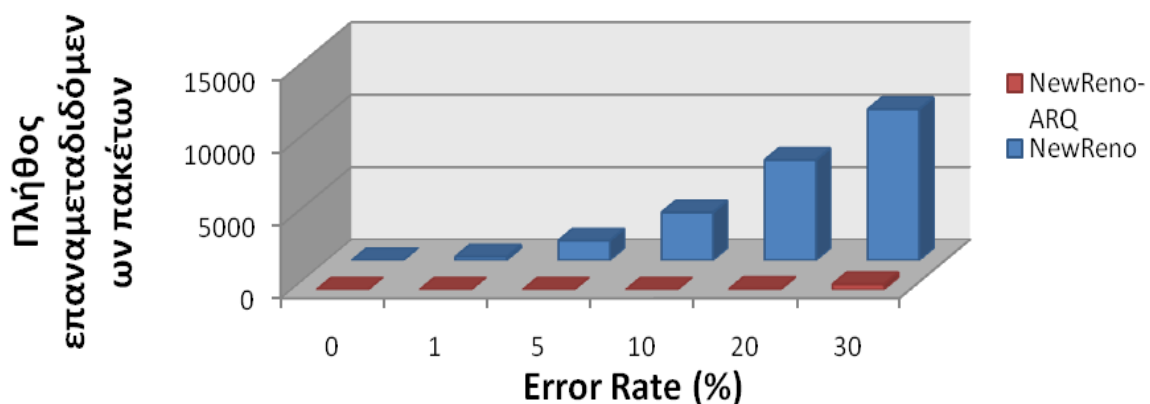


Σχήμα 6.29 Goodput – Συνολικό πλήθος κόμβων για ποσοστό σφαλμάτων 1%



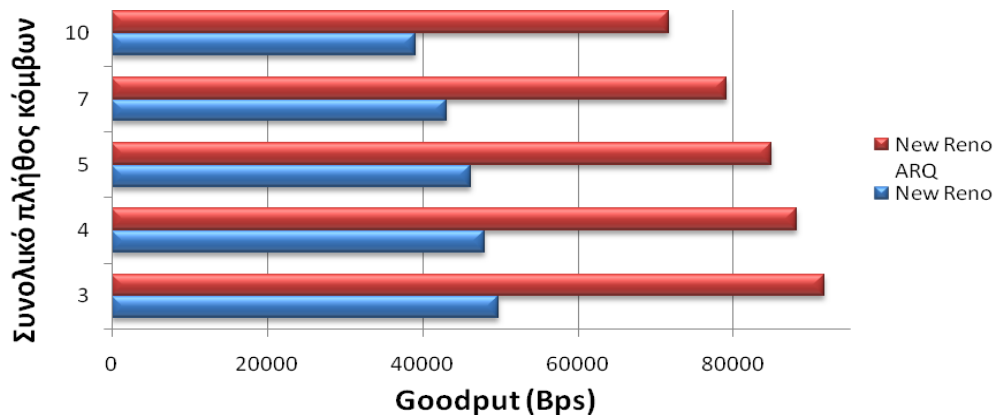
Σχήμα 6.30 Goodput – Συνολικό πλήθος κόμβων για ποσοστό σφαλμάτων 10%

Η διαφορά σε όρους απόδοσης που παρατηρούμε ανάμεσα στις δύο περιπτώσεις είναι πολύ σημαντική και αυξάνεται για μεγαλύτερα ποσοστά σφάλματος για ποσοστό σφάλματος 1% η πραγματική απόδοση χωρίς τη χρήση του ARQ είναι διπλάσια από ότι με τη ταυτόχρονη χρήση του ARQ και TCP, ενώ για ποσοστό σφάλματος 10% η ίδια διαφορά είναι οκταπλάσια. Αυτό δικαιολογείται χάρη στις επαναμεταδόσεις που γίνονται από το επίπεδο σύνδεσης, πριν το TCP New Reno αντιληφθεί την απώλεια των δεδομένων. Για το λόγο αυτό, άλλωστε, όταν εφαρμόζεται το αξιόπιστο πρωτόκολλο σύνδεσης, το συνολικό πλήθος των επαναμεταδόσεων που γίνονται από το επίπεδο μεταφοράς μειώνεται κατακόρυφα. (σχήμα 6.31)

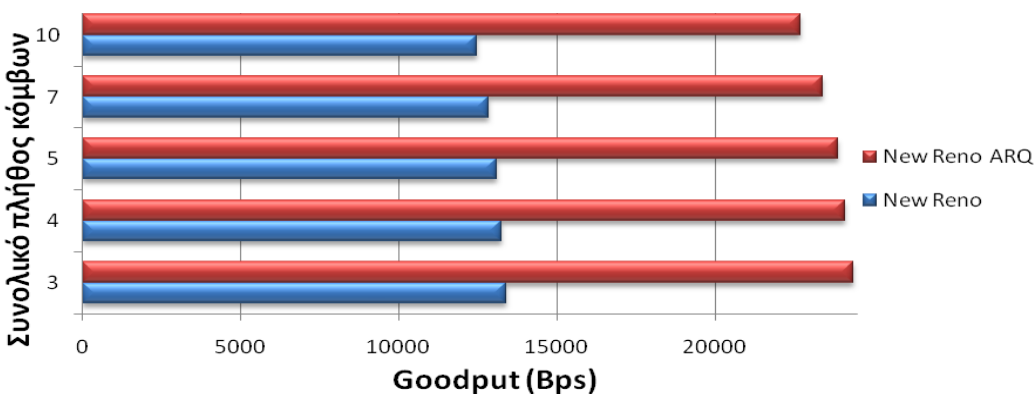


Σχήμα 6.31 Επαναμεταδόσεις χαμένων πακέτων από το επίπεδο μεταφοράς συναρτήσει του ρυθμού σφαλμάτων

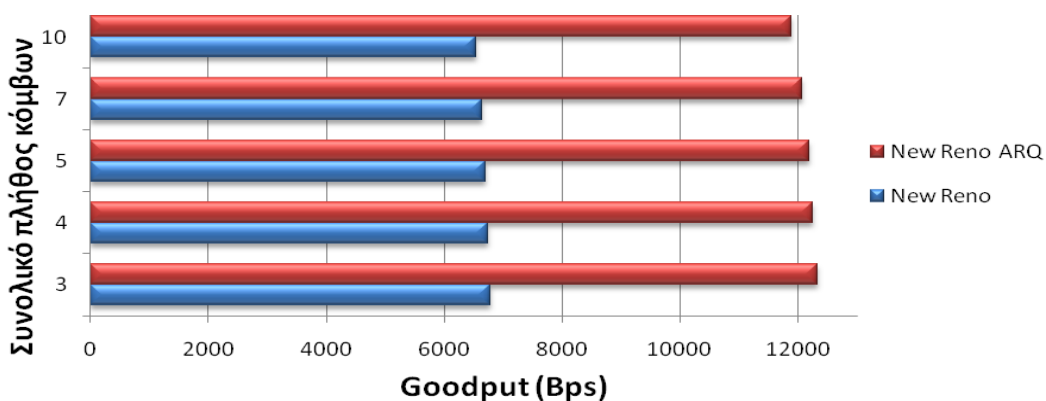
Στις γραφικές παραστάσεις που ακολουθούν απεικονίζεται η απόδοση συναρτήσει του συνολικού πλήθους κόμβων για αυξανόμενη καθυστέρηση διάδοσης από-άκρο-σε-άκρο. Παρατηρούμε πως το πλήθος των ενδιάμεσων κόμβων επηρεάζει την απόδοση του δικτύου μόνο για μικρές τιμές καθυστέρησης διάδοσης. Συγκεκριμένα, για καθυστέρηση διάδοσης 100ms η πραγματική απόδοση μειώνεται για αυξανόμενο πλήθος κόμβων, ενώ για τιμές καθυστέρησης μεγαλύτερες από 400ms παραμένει σχεδόν σταθερή.



Σχήμα 6.32 Goodput – Συνολικό πλήθος κόμβων για καθυστέρηση διάδοσης 100ms



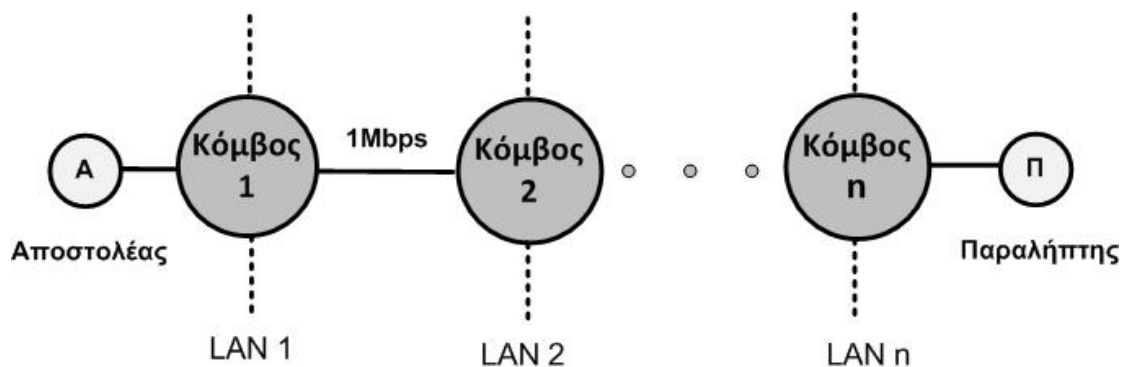
Σχήμα 6.33 Goodput – Συνολικό πλήθος κόμβων για καθυστέρηση διάδοσης 400ms



Σχήμα 6.34 Goodput – Συνολικό πλήθος κόμβων για καθυστέρηση διάδοσης 800ms

6.2.2.2. Πειράματα χρησιμοποιώντας πρωτόκολλο σύνδεσης Snoop και το πρωτόκολλο μεταφοράς TCP New Reno

Τα πειράματα που ακολουθούν προσομοιώνουν την τοπολογία της εικόνας 6.35, χρησιμοποιώντας ίδια χαρακτηριστικά με το προηγούμενο σετ πειραμάτων. Η μόνη διαφορά είναι πως αυτή τη φορά το πρωτόκολλο σύνδεσης που χρησιμοποιούμε είναι το πρωτόκολλο Snoop. Πράκτορας Snoop θεωρείται ο πρώτος κόμβος της τοπολογίας. Λόγω προβλημάτων υλοποίησης του πρωτοκόλλου Snoop στον προσομοιωτή δικτύων NS-2, στην παρακάτω τοπολογία οι ενδιάμεσοι κόμβοι υλοποιήθηκαν ως μέρος ενός διαφορετικού τοπικού δικτύου ο καθένας.

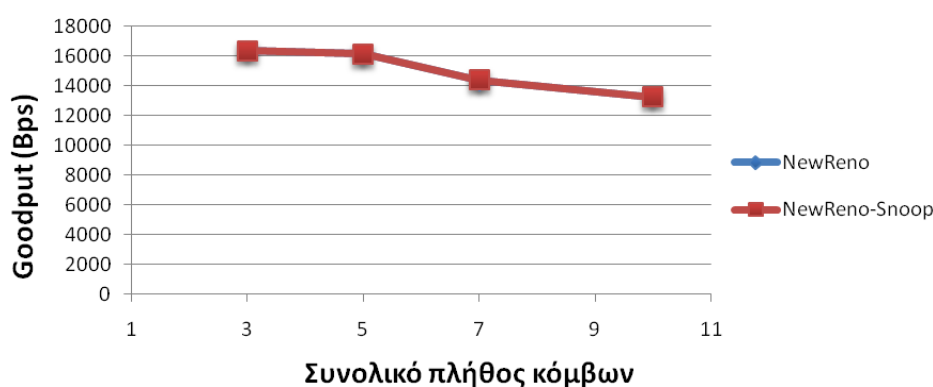


Σχήμα 6.35 Τοπολογία πειραμάτων

Παρακάτω δίνονται οι γραφικές παραστάσεις της απόδοσης σε συνάρτηση με το συνολικό πλήθος κόμβων της τοπολογίας για καθυστέρηση διάδοσης 400ms και ποσοστό σφαλμάτων που φτάνει το 10%. Αρχικά παρατηρούμε πως για ένα κανάλι το οποίο δεν παρουσιάζει σφάλματα η απόδοση του πρωτοκόλλου New Reno σε συνδυασμό με το πρωτόκολλο σύνδεσης Snoop ταυτίζεται με την απόδοση που πετυχαίνουμε χρησιμοποιώντας μόνο το αξιόπιστο πρωτόκολλο μεταφοράς. Ωστόσο, για αυξανόμενο ρυθμό σφαλμάτων παρατηρείται μια διαφορά στην απόδοση (πάντα υπερσχύει η ταυτόχρονη εφαρμογή αξιόπιστων πρωτοκόλλων σε δύο επίπεδα), η οποία μάλιστα αυξάνεται για μεγαλύτερο ποσοστό σφαλμάτων.

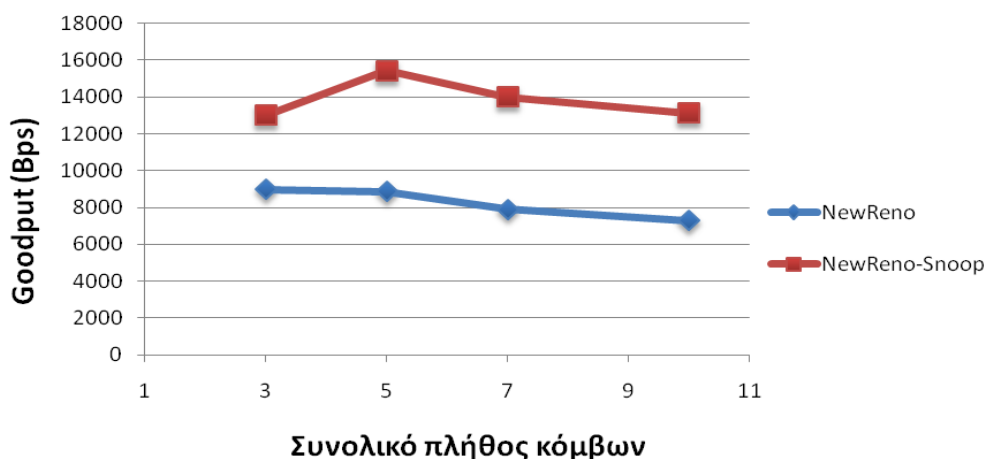
Όσον αφορά το πλήθος των ενδιάμεσων κόμβων, στην περίπτωση εφαρμογής μόνο του αξιόπιστου πρωτοκόλλου μεταφοράς TCP New Reno, παρατηρούμε μια σταδιακή πτώση της απόδοσης για αυξανόμενο πλήθος κόμβων. Η πτώση αυτή οφείλεται στο περιορισμό του μεγέθους των καταχωρητών για μικρό πλήθος

κόμβων η καθυστέρηση ανάμεσα σε δύο κόμβους είναι μεγάλη με αποτέλεσμα οι καταχωρητές να προλαβαίνουν να αποθηκεύσουν όλα τα πακέτα δεδομένων που φτάνουν σε αυτούς, πριν τα προωθήσουν. Αντίθετα, στην περίπτωση των πολλών ενδιάμεσων κόμβων η καθυστέρηση διάδοσης ανάμεσα σε οποιουδήποτε δύο κόμβους είναι σημαντικά μικρότερη, με αποτέλεσμα μεγάλος όγκος δεδομένων να φτάνει γρηγορότερα στους καταχωρητές, οι οποίοι πλέον δεν έχουν τη δυνατότητα να αποθηκεύσουν προσωρινά όλα τα πακέτα και, αναγκαστικά, τα απορρίπτουν. Έτσι, τα χαμένα πακέτα πρέπει να επαναμεταδοθούν, γεγονός που οδηγεί σε μείωση της πραγματικής απόδοσης.



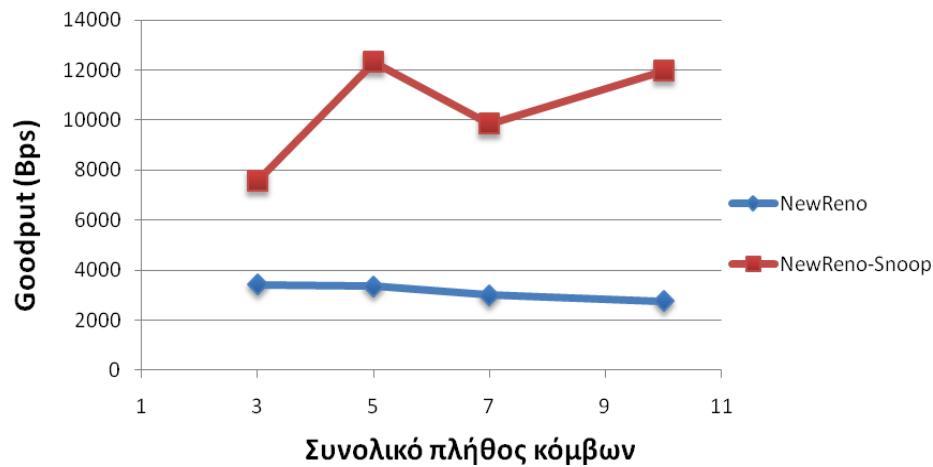
Σχήμα 6.36 Goodput – Συνολικό πλήθος κόμβων για μηδενικό ρυθμό σφαλμάτων

Για ποσοστό σφαλμάτων της τάξης του 1% παρατηρούμε μια σημαντική διαφορά όσον αφορά την πραγματική απόδοση στις δύο περιπτώσεις, με εμφανή υπεροχή του σεναρίου που χρησιμοποιούμε το πρωτόκολλο Snoop. Ακόμη, για αυξανόμενο πλήθος ενδιάμεσων κόμβων παρατηρούμε μια σταδιακή πτώση της απόδοσης.

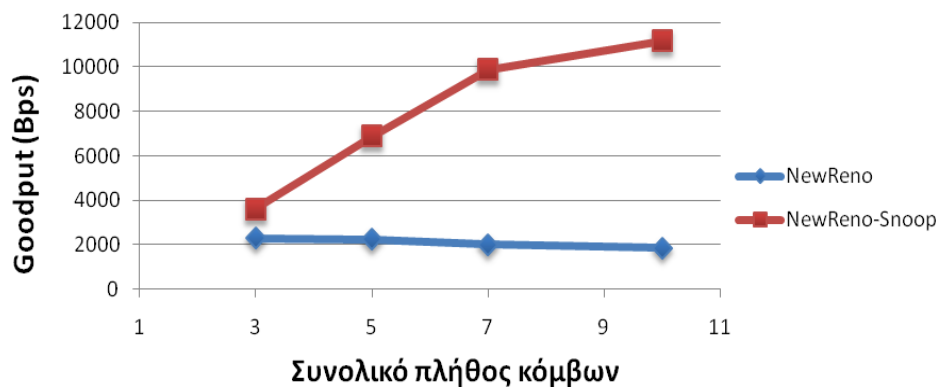


Σχήμα 6.37 Goodput – Συνολικό πλήθος κόμβων για ρυθμό σφαλμάτων 1%

Στην περίπτωση εφαρμογής μόνο του πρωτοκόλλου TCP New Reno, για μεγαλύτερα ποσοστά σφαλμάτων παρατηρούμε, αντίστοιχα με προηγουμένως, μια μικρή μείωση της απόδοσης για αυξανόμενο πλήθος ενδιάμεσων κόμβων.



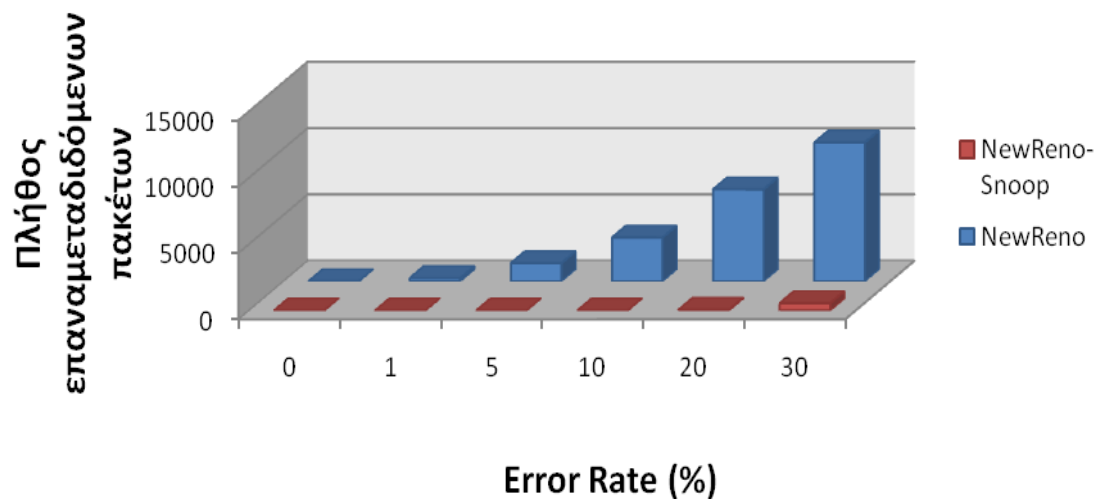
Σχήμα 6.38 Goodput – Συνολικό πλήθος κόμβων για ρυθμό σφαλμάτων 5%



Σχήμα 6.39 Goodput – Συνολικό πλήθος κόμβων για ρυθμό σφαλμάτων 10%

Για μεγάλους ρυθμούς σφαλμάτων και χρησιμοποιώντας αξιοπιστία σε δύο επίπεδα παρατηρούμε ότι η απόδοση που πετυχαίνουμε αυξάνεται για αυξανόμενο πλήθος ενδιάμεσων κόμβων. Το γεγονός αυτό οφείλεται στη μειωμένη καθυστέρηση ανάμεσα σε δύο κόμβους του δικτύου για μεγάλο πλήθος κόμβων, η οποία οδηγεί σε γρηγορότερες επαναμεταδόσεις των χαμένων πακέτων από το επίπεδο μεταφοράς, λόγω εφαρμογής του πρωτοκόλλου Snoop και, συνεπώς, αύξηση της πραγματικής απόδοσης.

Στο διάγραμμα που ακολουθεί απεικονίζονται οι επαναμεταδόσεις των χαμένων πακέτων που γίνονται από το επίπεδο μεταφοράς σε συνάρτηση με το ποσοστό σφαλμάτων και στις δύο περιπτώσεις. Η σημαντική διαφορά στο πλήθος των επαναμεταδόσεων, ειδικά για μεγάλες τιμές σφαλμάτων, δικαιολογείται από τις επαναμεταδόσεις που γίνονται στο επίπεδο σύνδεσης. Οι επαναμεταδόσεις αυτές γίνονται γρηγορότερα από ότι στο επίπεδο μεταφοράς για το λόγο αυτό η πραγματική απόδοση που πετυχαίνουμε χρησιμοποιώντας αξιόπιστο πρωτόκολλο σύνδεσης είναι καλύτερη.



Σχήμα 6.40 Επαναμεταδόσεις χαμένων πακέτων από το επίπεδο μεταφοράς συναρτήσει του ρυθμού σφαλμάτων

6.3. Συμπεράσματα και σχόλια

Από τα πειράματα που εκτελέστηκαν και τα αντίστοιχα αποτελέσματα μπορούμε να εξάγουμε σημαντικά συμπεράσματα σχετικά με την αξιοπιστία διαφορετικών πρωτοκόλλων σε διάφορα επίπεδα του δικτυακού μοντέλου OSI.

Όσον αφορά τα επίγεια δίκτυα, παρατηρήσαμε ότι, ακόμη και για μικρούς ρυθμούς σφαλμάτων, η χρήση μηχανισμών αξιοπιστίας μόνο στο επίπεδο μεταφοράς οδηγεί σε ραγδαία πτώση της απόδοσης. Η πτώση αυτή δεν είναι τόσο απότομη χρησιμοποιώντας το πρωτόκολλο Snoop στο επίπεδο σύνδεσης, ενώ χρησιμοποιώντας τον μηχανισμό ARQ η απόδοση παραμένει σταθερή, ανεξάρτητα από το ρυθμό σφαλμάτων.

Σε συνθήκες χαμηλής δικτυακής συμφόρησης παρατηρήθηκε ότι η χρήση αξιοπιστων μηχανισμών σε δύο επίπεδα οδηγεί σε επιβάρυνση και, συνεπώς, σε μείωση της απόδοσης. Η επιβάρυνση οφείλεται στον επιπλέον χρόνο επεξεργασίας που εισάγουν τα πρωτόκολλα επιπέδου σύνδεσης, τα οποία ουσιαστικά δεν παρέχουν κάποια λειτουργικότητα, αφού ο ρυθμός σφαλμάτων είναι αρκετά χαμηλός, ώστε να μπορεί να αντιμετωπιστεί από τις επαναμεταδόσεις στο επίπεδο μεταφοράς.

Αντίθετα, για μεγάλα επίπεδα συμφόρησης τα αξιόπιστα πρωτόκολλα μεταφοράς δεν είναι αρκετά αποδοτικά, γεγονός που αντισταθμίζεται από την εφαρμογή αξιοπιστίας στο επίπεδο σύνδεσης. Παρατηρήθηκε πως υπάρχει ένα συγκεκριμένο επίπεδο συμφόρησης για το οποίο, εφόσον χρησιμοποιούμε αξιόπιστο πρωτόκολλο μεταφοράς, η εφαρμογή αξιοπιστίας και στο επίπεδο σύνδεσης δεν οδηγεί σε καμία διαφοροποίηση σε όρους απόδοσης.

Για μεγάλες τιμές καθυστέρησης διάδοσης, όπως στα δορυφορικά δίκτυα, χρησιμοποιώντας το πρωτόκολλο TCP παρατηρούμε μια σταδιακή πτώση της απόδοσης. Η μείωση αυτή οφείλεται στη λειτουργία του TCP στις συνθήκες αυτές λόγω της υψηλής καθυστέρησης ο αποστολέας δεν παραλαμβάνει εγκαίρως τις επιβεβαιώσεις των πακέτων που μετέδωσε, θεωρεί εσφαλμένα ότι τα πακέτα

χάθηκαν και επιστρέφει άσκοπα στη φάση του αργού ξεκινήματος, μειώνοντας το ρυθμό αποστολής του.

Για αυξανόμενο ρυθμό σφαλμάτων σε ένα δορυφορικό δίκτυο, το πρωτόκολλο TCP επηρεάζεται σημαντικά και δεν μπορεί να λειτουργήσει αποδοτικά. Αντίθετα, η εφαρμογή αξιοπιστίας στο επίπεδο σύνδεσης αντισταθμίζει την αδυναμία του TCP, επαναμεταδίδοντας τα χαμένα πακέτα πριν το αντιληφθεί το επίπεδο μεταφοράς και, έτσι, διατηρεί την απόδοση του συστήματος σε υψηλά επίπεδα. Ειδικά για πολύ μεγάλους ρυθμούς σφαλμάτων η διαφορά σε όρους απόδοσης είναι μέχρι και εικοσαπλάσια χρησιμοποιώντας το πρωτόκολλο σύνδεσης ARQ. Το πρωτόκολλο Snoop δεν λειτουργεί τόσο αποδοτικά, όσο το ARQ σε δορυφορικά δίκτυα. Ειδικά για μεγάλους ρυθμούς σφαλμάτων δεν λειτουργεί αποδοτικά.

Μελετήθηκε ακόμη η επίδραση του πλήθους των ενδιάμεσων κόμβων σε μια τοπολογία στην τελική απόδοση του συστήματος. Για το πρωτόκολλο ARQ, ανεξάρτητα από το ρυθμό σφαλμάτων, παρατηρείται μια ελάχιστη μείωση της απόδοσης για αυξανόμενο πλήθος ενδιάμεσων κόμβων. Η πτώση αυτή οφείλεται στο περιορισμό που θέτει το μέγεθος των καταχωρητών για μικρό πλήθος κόμβων η καθυστέρηση ανάμεσα σε δύο κόμβους είναι μεγάλη με αποτέλεσμα οι καταχωρητές να προλαβαίνουν να αποθηκεύσουν όλα τα πακέτα δεδομένων που φτάνουν σε αυτούς, πριν τα προωθήσουν. Στην περίπτωση, όμως, πολλών ενδιάμεσων κόμβων η καθυστέρηση διάδοσης ανάμεσα σε οποιουδήποτε δύο κόμβους είναι σημαντικά μικρότερη με αποτέλεσμα μεγάλος όγκος δεδομένων να φτάνει γρηγορότερα στους καταχωρητές, οι οποίοι πλέον δεν έχουν τη δυνατότητα να αποθηκεύσουν προσωρινά όλα τα πακέτα και, αναγκαστικά, τα απορρίπτουν. Έτσι, τα χαμένα πακέτα πρέπει να επαναμεταδοθούν, γεγονός που οδηγεί σε μείωση της πραγματικής απόδοσης. Ωστόσο, για αυξανόμενες τιμές καθυστέρησης το πλήθος των κόμβων επηρεάζει ολοένα και λιγότερο την απόδοση του συστήματος.

Αντίστοιχα, χρησιμοποιώντας το αξιόπιστο πρωτόκολλο σύνδεσης Snoop για μικρό ρυθμό σφαλμάτων, η απόδοση και πάλι μειώνεται για αυξανόμενο πλήθος ενδιάμεσων κόμβων. Παρατηρούμε, όμως, ότι για μεγάλους ρυθμούς σφαλμάτων η απόδοση που πετυχαίνουμε χρησιμοποιώντας το αξιόπιστο πρωτόκολλο σύνδεσης

Σnoor αυξάνεται για αυξανόμενο πλήθος ενδιάμεσων κόμβων. Το γεγονός αυτό οφείλεται στη μειωμένη καθυστέρηση ανάμεσα σε δύο κόμβους του δικτύου για μεγάλο πλήθος κόμβων, η οποία οδηγεί σε γρηγορότερες επαναμεταδόσεις των χαμένων πακέτων από το επίπεδο μεταφοράς λόγω της λειτουργίας του Snoor και, συνεπώς, αύξηση της πραγματικής απόδοσης.

Με λίγα λόγια, αποδεικνύεται πως τα υπάρχοντα αξιόπιστα πρωτόκολλα μεταφοράς δεν λειτουργούν αποδοτικά σε δορυφορικά δίκτυα. Για το λόγο αυτό, είναι απαραίτητη η ταυτόχρονη εφαρμογή μηχανισμών αξιοπιστίας στο επίπεδο σύνδεσης, οι οποίοι θα αντισταθμίζουν τις αδυναμίες του επιπέδου μεταφοράς και θα διατηρούν την απόδοση σε υψηλά επίπεδα. Σε κάθε περίπτωση θα πρέπει να λαμβάνονται υπόψη συγκεκριμένες παράμετροι του δικτύου (ρυθμός σφαλμάτων, πλήθος κόμβων κτλ.) για την εφαρμογή των καταλληλότερων πρωτοκόλλων και μηχανισμών.

Βιβλιογραφία

- [1] Β. Τσαουσίδης, *“Διαδικτυακά πρωτόκολλα”*, Κλειδάριθμος, 2004
- [2] C. Zhang, *“Receiver-oriented and measurement-based Transmission Control Protocol in Heterogeneous (Wired/Wireless) Networks”*, Ph.D. Thesis in Computer Science of Northeastern University, 2003
- [3] A. Lahanas, *“On the efficiency and fairness of AIMD-based Congestion Avoidance Algorithms”*, Ph.D. Thesis in Computer Science of Northeastern University, 2003
- [4] S. Hsu, *“Computer Network Programming–TCP Overview”*, Computer and Science Engineering, Florida Atlantic University
- [5] J. Postel, *“Transmission Control Protocol”*, IETF RFC 793, 1981
- [6] W. Stevens, *“TCP/IP Illustrated, Volume 1: The Protocols”*, Addison-Wesley, 1994
- [7] D. M. Chiu and R. Jain, *“Analysis of the Increase and Decrease Algorithms for Congestion Avoidance in Computer Networks”*, Computer Networks and ISDN Systems, 1989
- [8] W. Stevens, *“TCP Slow Start, Congestion Avoidance, Fast Retransmit and Fast Recovery Algorithms”*, IETF RFC 2001, 1997
- [9] S. Floyd and M. Handley, *“The New Reno Modification to TCP’s Fast Recovery Algorithm”*, IETF RFC 2582, 1999
- [10] K. Fall and S. Floyd, *“Simulation Based Comparisons of Tahoe, Reno and SACK TCP”*, Computer Communication Review, 1996
- [11] V. Jacobson, *“Congestion Avoidance and Control”*, ACM SIGCOMM, 1998

- [12] M. Allman, V. Paxson and W. Stevens, *"TCP Congestion Control"*, IETF RFC 2581, 1999
- [13] M. Mathis, J. Mahdavi, S. Floyd and A. Romanow, *"TCP Selective Acknowledgment Options"*, IETF RFC 2018, 1996
- [14] M. Handley, J. Padhye, S. Floyd and J. Widmer, *"TCP Friendly Rate Control (TFRC): Protocol Specification"*, IETF RFC 3448, 2001
- [15] L. S. Brakmo and L. L. Peterson, *"TCP Vegas: End to End Congestion Avoidance on a Global Internet"*, IEEE Journal on Selected Areas in Communications, 1995
- [16] C. Zhang and V. Tsaoussidis, *"The interrelation of TCP Responsiveness and Smoothness in heterogeneous networks"*, The 7th IEEE Symposium on Computers and Communications, 2002
- [17] V. Tsaoussidis, H. Badr and R. Verma, *"TCP Probing: Towards an Error Control Schema with Energy and Throughput Performance Gains"*, The 7th International Conference on Network Protocols, 1999
- [18] S. Floyd and V. Jacobson, *"Random Early Detection gateways for Congestion Avoidance"*, IEEE/ACM Transactions on Networking, 1993
- [19] B. Braden, D. Clark, J. Crowcroft, B. Davie, S. Deering, D. Estrin, S. Floyd, V. Jacobson, G. Minshall, C. Partridge, L. Peterson, K. Ramakrishnan, S. Shenker, J. Wroclawski and L. Zhang, *"Recommendations on Queue Management and Congestion Avoidance in the Internet"*, IETF RFC 2309, 1998
- [20] J. Postel, *"User Datagram Protocol"*, IETF RFC 768, 1980
- [21] A. S. Chouhan, A. Dhar, N. Sharma and A. Sangodkar, *"Simulation of Snoop Protocol"*, 2007

- [22] S. Vangala and M. A. Labrador, *"Performance of TCP over Wireless Networks with the Snoop Protocol"*, 27th Annual IEEE International Conference on Local Computer Networks - LCN, 2002
- [23] H. Balakrishnan, S. Seshan, E. Amir and R. Katz, *"Improving TCP/IP Performance over Wireless Networks"*, 1st ACM Conference on Mobile Communications and Networking - Mobicom, 1995
- [24] G. Fairhurst and L. Wood, *"Advice to link designers on link Automatic Repeat reQuest (ARQ)"*, IETF RFC 3366, 2002
- [25] A. Chockalingam and M. Zorzi, *"Wireless TCP Performance with link layer FEC/ARQ"*, IEEE International Conference on Communications - ICC, 1999
- [26] Y. Bai, A. T. Ogielski and G. Wu, *"Interactions of TCP and radio link ARQ"*, IEEE Vehicular Technology Conference - VTC, 1999
- [27] A. A. Fawal and C. Barakat, *"Simulation-based study of link-level hybrid FEC/ARQ-SR for wireless links and long-lived TCP traffic"*, Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks – WiOpt, 2003
- [28] J. Wong and V. Leung, *"Improving end-to-end performance of TCP using link-layer retransmissions over mobile internetworks"*, IEEE International Conference on Communications - ICC, 1999
- [29] F. Filali, *"Link-layer fragmentation and retransmission impact on TCP performance in 802.11-based networks"*, The 7th International Conference on Mobile and Wireless Communications Networks - IFIP, 2005
- [30] S. D. Vuyst, K. Tworus, S. Wittevrongel and H. Bruneel, *"Analysis of Stop-and-Wait ARQ for a wireless channel"*, 4OR: A Quarterly Journal of Operations Research Vol. 7-1, 2009

- [31] K. Nagaoka, C. X. Chen and M. Komatsu, *“Throughput Efficiency of Go-Back-N ARQ Protocol in Parallel Multi-Channel with Burst Errors”*, IEICE Transactions on Communication, 2008
- [32] T. Shikama and T. Mizuno, *“Resequencing Schemes for Selective-Repeat ARQ and Their Performance”*, 19th International Conference on Advanced Information Networking and Applications - AINA, 2005
- [33] I. Chlamtaca, C. Petriolib and J. Redic, *“Energy-conserving Selective Repeat ARQ Protocols for Wireless Data”*, The 9th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications – PIMRC, 1998
- [34] M. Luby, L. Vicisano, J. Gemmell, L. Rizzo, M. Handley and J. Crowcroft, *“Forward Error Correction (FEC) Building Block”*, IETF RFC 3452, 2002
- [35] M. Luby and L. Vicisano, *“Compact Forward Error Correction (FEC) Schemes”*, IETF RFC 3695, 2004
- [36] D. Anantharaman, *“Performance Analysis of Snack in Satellite Networks through Simulation”*, Master's thesis in Lamar University, 2004
- [37] O. B. Akan, J. Fang and I. F. Akyildiz, *“Performance of TCP Protocols in Deep Space Communication Networks”*, IEEE Communication Letters Vol.6, 2002
- [38] R. C. Burst, G. J. Miller and E. J. Travis, *“TCP Extensions for Space Communications”*, Wireless Networks 3, Science Publishers, 1997
- [39] A. Sinha, *“Study of Proposed Methods for Improving TCP Performance over Wireless Links”*, Computer Research Repository – CoRR, 2009
- [40] H. Balakrishnan, V. Padmanabhan, S. Seshan and R. H. Katz, *“A comparison of mechanisms for improving TCP performance over wireless links”*, IEEE/ACM Transactions on Networking, 1997

- [41] W. Ding and A. Jamalipour, "*A new explicit loss notification with acknowledgment for wireless TCP*", The 12th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, 2001
- [42] K. Ramakrishnan and S. Floyd, "*The Addition of Explicit Congestion Notification (ECN) to IP*", IETF RFC 2481, 2001
- [43] S. Keshav and S. P. Morgan, "*SMART Retransmission: Performance with Overload and Random Losses*", IEEE Conference on Computer Communications - INFOCOM, 1996
- [44] A. B. Roach, "*A Negative Acknowledgement Mechanism for Signaling Compression*", IETF RFC 4077, 2005
- [45] M. Mathis, J. Mahdavi, S. Floyd and A. Romanow, "*TCP Selective Acknowledgment Options*", IETF RFC 2018, 1996
- [46] A. Bakre and B. R. Badrinath, "*I-TCP: Indirect TCP for Mobile Hosts*", The 15th International Conference on Distributed Computing Systems, 1995
- [47] Z. J. Haas, "*Mobile-TCP: An Asymmetric Transport Protocol Design for Mobile Systems*", IEEE International Conference on Communications, 1997
- [48] N. Vimal and K. Subramanian, "*Survey on wireless link reliability mechanisms using Transmission Control Protocol*", Technical Report, Computer Science, University of North Carolina, 2005
- [49] T. Mahmodi, V. Friderikos, O. Holland and A. H. Aghvami, "*Cross-layer design to improve wireless TCP performance with link-layer adaptation*", IEEE 66th Vehicular Technology Conference - VTC, 2007
- [50] C. Parsa and J. Garcia-Luna-Aceves, "*TULIP: A link-level protocol for improving TCP over wireless links*", IEEE Wireless Communications and Networking Conference - WCNC, 1999

[51] K. Fall and K. Varadhan, “The ns manual”, 2007

[52] NS-2 Network Simulator, Available from <http://www.isi.edu/nsnam/ns/>

