



**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

**“ΜΕΛΕΤΗ ΦΑΙΝΟΜΕΝΟΥ PACKETREORDERING ΣΤΟ  
ΠΡΩΤΟΚΟΛΛΟ TCP”**



Καραμπά Ναταλία  
ΑΜ 1703

Αταζάρ Τζίλι Ανν  
ΑΜ 1651

Επιβλέπων καθηγητής: Δρ Παξιμάδης Θ. Κωνσταντίνος

Ναύπακτος 2018

## Contents

ΚΕΦΑΛΑΙΟ 1 .....	8
1.1 ΕΙΣΑΓΩΓΗ ΣΤΟ TCP .....	8
1.2 ΤΟ ΜΟΝΤΕΛΟ ΥΠΗΡΕΣΙΑΣ ΤΟΥ TCP .....	8
1.3 ΤΟ ΠΡΩΤΟΚΟΛΛΟ TCP .....	10
1.4 Η ΚΕΦΑΛΙΑΔΑ ΤΜΗΜΑΤΟΣ ΤΟΥ TCP .....	11
1.5 ΕΓΚΑΘΙΔΡΥΣΗ ΣΥΝΔΕΣΕΩΝ ΣΤΟ TCP .....	15
1.6 ΑΠΟΔΕΣΜΕΥΣΗ ΣΥΝΔΕΣΕΩΝ ΣΤΟ TCP .....	16
1.7 ΜΟΝΤΕΛΟΠΟΙΗΣΗ ΔΙΑΧΕΙΡΙΣΗΣ ΣΥΝΔΕΣΕΩΝ ΤΟΥ TCP .....	16
1.8 ΠΟΛΙΤΙΚΗ ΜΕΤΑΔΟΣΗΣ ΣΤΟ TCP .....	17
ΚΕΦΑΛΑΙΟ 2 .....	19
2.1 ΈΛΕΓΧΟΣ ΣΥΜΦΟΡΗΣΗΣ ΛΕΙΤΟΥΡΓΙΩΝ ΤΟΥ TCP .....	19
2.2 ΑΙΤΙΕΣ PACKET REORDERING .....	21
2.2.1 ΠΟΛΛΑΠΛΗ ΔΡΟΜΟΛΟΓΗΣΗ ΣΕ ΕΠΙΠΕΔΟ ΠΑΚΕΤΟΥ (PACKET LEVEL MULTIPATH ROUTING) .....	21
2.2.2 Η ΚΙΝΗΣΗ ΤΗΣ ΔΙΑΔΡΟΜΗΣ (ROUTE FLUTTERING ROUTING) .....	21
2.2.3 ΕΝΥΠΑΡΧΩΝ ΠΑΡΑΛΛΗΛΙΣΜΟΣ ΣΤΟΥΣ ΣΥΓΧΡΟΝΟΥΣ ΥΨΗΛΗΣ ΤΑΧΥΤΗΤΑΣ ΔΡΟΜΟΛΟΓΗΤΕΣ (INHERENT PARALLELISM IN MODERN HIGH SPEED ROUTERS) .....	22
2.2.4 ΑΝΑΜΕΤΑΔΟΣΗ ΣΕ ΕΠΙΠΕΔΟ ΣΥΔΕΣΗΣ (LINK LAYER RETRANSMISSION) .....	23
2.2.5 ΠΑΥΣΗ ΠΡΟΩΘΗΣΗΣ ΤΟΥ ΔΡΟΜΟΛΟΓΗΤΗ (ROUTER FOR WARDING LULLS) .....	23
2.3 ΟΙ ΕΠΙΠΤΩΣΕΙΣ ΤΟΥ PACKET REORDERING ΣΤΟ TCP .....	24
2.3.1 ΠΑΡΑΣΙΤΙΚΕΣ ΑΝΑΜΕΤΑΔΟΣΕΙΣ ΤΜΗΜΑΤΟΣ (SPURIOUS SEGMENT RETRANSMISSION) .....	24
2.3.2 ΚΡΑΤΩΝΤΑΣ ΤΟ ΠΑΡΑΘΥΡΟ ΣΥΜΦΟΡΗΣΗΣ ΑΔΙΚΑΙΟΛΟΓΗΤΑ ΜΙΚΡΟ (KEEPING CONGESTION WINDOW UNNECESSARILY SMALL) .....	25
2.3.3 ΑΠΩΛΕΙΑ ACK-ΧΡΟΝΙΣΜΟΥ (LOSS OF ACK-CLOCKING) .....	25
2.3.4 ΜΕΙΩΣΗ ΕΚΤΙΜΩΜΕΝΟΥ RTT ΚΑΙ RTO (UNDERSTANDING ESTIMATED RTT AND RTO) .....	26
2.4 ΤΑΞΙΝΟΜΗΣΗ ΛΥΣΕΩΝ ΑΝΑΔΙΑΤΑΞΗΣ ΓΙΑ ΤΟ TCP .....	27
2.4.1 ΤΑΚΤΙΚΗ ΠΡΟΣΕΓΓΙΣΗ (ORDINAL APPROACH) .....	28
ΣΥΜΦΩΝΙΑ ΚΑΤΑΣΤΑΣΗΣ (STATE RECONCILIATION) .....	29
ΌΡΙΟ ΠΡΟΣΑΡΜΟΓΗΣ (THRESHOLD ADJUSTMENT) .....	32
2.4.2 ΧΡΟΝΙΚΗ ΠΡΟΣΕΓΓΙΣΗ (TEMPORAL APPROACH) .....	39

ΑΝΑΒΟΛΗ ΑΠΟΚΡΙΣΗΣ (RESPONSE POSTPONEMENT).....	39
ΑΝΑΜΕΤΑΔΟΣΗ ΜΕ ΤΗ ΛΗΞΗ ΤΟΥ ΧΡΟΝΟΥ (RETRANSMISSION TIMEOUT).....	41
ΚΕΦΑΛΑΙΟ 3 .....	44
3.1 PACKET REORDERING ΣΕ ΔΙΚΤΥΑ ΠΟΛΛΑΠΛΩΝ ΔΙΑΔΡΟΜΩΝ.....	44
3.2 ΠΟΣΟ ΚΟΙΝΟ ΦΑΙΝΟΜΕΝΟ ΕΙΝΑΙ ΤΟ PACKET REORDERING.....	44
3.3 ΠΡΟΤΕΙΝΟΜΕΝΕΣ ΛΥΣΕΙΣ .....	47
3.3.1 ΛΥΣΗ 1: ΛΥΝΟΝΤΑΣ ΤΟ ΠΡΟΒΛΗΜΑ ΣΕ ΕΝΑ ΚΑΤΩΤΕΡΟ ΕΠΙΠΕΔΟ .....	48
• Αναδιάταξη buffer (Reorder buffer): .....	48
• Κωδικοποίηση του Δικτύου (Networking code):.....	49
3.3.2 ΛΥΣΗ 2: ΔΥΝΑΜΙΚΗ ΠΡΟΣΑΡΜΟΓΗ ΤΟΥ DUPTHRESH.....	50
3.3.3 ΛΥΣΗ 3: ΚΑΘΥΣΤΕΡΩΝΤΑΣ ΤΗΝ ΕΜΦΑΝΙΣΗ ΤΟΥ ΕΛΕΓΧΟΥ ΣΥΜΦΟΡΗΣΗΣ.....	51
3.3.4 ΛΥΣΗ 4: ΑΝΙΧΝΕΥΣΗ ΚΑΙ ΑΝΑΚΑΜΨΗ ΑΠΟ ΨΕΥΔΕΙΣ ΑΝΑΜΕΤΑΔΟΣΕΙΣ .....	51
• Χρονισμός του ACK (ACK timing).....	52
• Χρονοσφραγίδα (Timestamping):.....	52
• DSACK: .....	53
3.4 ΠΑΡΑΛΛΑΓΕΣ ΤΟΥ TCP.....	53
Πίνακας 1: TCP variants .....	54
• TCP Eifel.....	54
• TCP-LPC .....	55
• TCP Westwood.....	56
• TCP-BA.....	56
• RR-TCP.....	58
• TCP-PR (Persistent Packet Reordering).....	58
• TCP-DCR (Delayed Congestion Response).....	58
• TCP-NCR (Non Congestion Response ) .....	59
• TCP/NC (Network Coding).....	59
3.5 ΚΡΙΤΙΚΗ ΑΝΑΛΥΣΗ.....	60
3.5.1 ΧΡΟΝΙΚΑ ΌΡΙΑ .....	62
ΚΕΦΑΛΑΙΟ 4 .....	70
4.1 ΑΝΑΔΙΑΤΑΞΗ ΠΑΚΕΤΩΝ ΣΕ ΔΙΚΤΥΑ ΥΨΗΛΩΝ ΤΑΧΥΤΗΤΩΝ .....	70
4.2 ΒΑΣΙΚΑ ΘΕΜΑΤΑ ΤΟΥ PACKET REORDERING .....	71

4.3 ΠΡΟΤΥΠΑ ΠΑΚΕΤΩΝ ΑΝΑΔΙΑΤΑΞΗΣ.....	72
4.4 ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΗΣ ΑΝΑΔΙΑΤΑΞΗΣ ΠΑΚΕΤΩΝ ΣΕ ΔΙΚΤΥΑ ΥΨΗΛΩΝ ΤΑΧΥΤΗΤΩΝ	75
4.5 ΕΠΙΠΤΩΣΗ ΤΟΥ PACKET REORDERING ΣΕ ΔΙΚΤΥΑ ΥΨΗΛΩΝ ΤΑΧΥΤΗΤΩΝ .....	79
4.6 ΕΠΙΠΤΩΣΗ ΤΩΝ ΥΦΙΣΤΑΜΕΝΩΝ ΕΝΙΣΧΥΣΕΩΝ ΤΟΥ TCP .....	82
<b>ΚΕΦΑΛΑΙΟ 5 .....</b>	<b>87</b>
<b>5.1 ΕΠΙΜΟΝΗ ΑΝΑΔΙΑΤΑΞΗ ΠΑΚΕΤΩΝ ΣΤΟ TCP.....</b>	<b>87</b>
<b>5.2 ΒΑΣΙΚΟΣ ΑΛΓΟΡΙΘΜΟΣ .....</b>	<b>87</b>
5.3 ΑΚΡΑΙΕΣ ΑΠΩΛΕΙΕΣ .....	91
5.4 ΑΠΟΔΟΣΗ ΧΩΡΙΣ PACKET REORDERING: PERFORMANCE ΚΑΙ FAIRNESS.....	93
5.5 ΑΠΟΔΟΣΗ ΥΠΟ ΤΗΝ ΑΝΑΔΙΑΤΑΞΗ ΠΑΚΕΤΩΝ ΣΕ ΣΥΓΚΡΙΣΗ ΜΕ ΑΛΛΕΣ ΜΕΘΟΔΟΥΣ ..	98
<b>ΚΕΦΑΛΑΙΟ 6 .....</b>	<b>101</b>
6.1 ΣΥΜΠΕΡΑΣΜΑΤΑ ΣΧΕΤΙΚΑ ΜΕ ΤΗΝ ΑΠΟΔΟΣΗ ΤΩΝ ΠΑΡΑΛΛΑΓΩΝ TCP .....	101
Βιβλιογραφία .....	105

## Ευχαριστίες

Θα θέλαμε να ευχαριστήσουμε τον κύριο Κωνσταντίνο Παξιμάδη που μας υποστήριξε όλο αυτό το διάστημα και μας καθοδήγησε στην επιτυχή ολοκλήρωση της πτυχιακής εργασίας.



## ΠΕΡΙΛΗΨΗ

Η εργασία αυτή αναφέρεται στο φαινόμενο του packet reordering στο πρωτόκολλο TCP (Transport Control Protocol).

Το πρώτο κεφάλαιο αναφέρεται στο πρωτόκολλο TCP. Αναλύεται η δομή του και επεξηγείται ο τρόπος λειτουργίας του.

Στο δεύτερο κεφάλαιο αναλύεται ο έλεγχος συμφόρησης και επεξηγούνται αιτίες και λύσεις του packet reordering.

Στο τρίτο κεφάλαιο περιγράφεται το packet reordering σε δίκτυα πολλαπλών διαδρομών (multipath networks). Αναλύονται λύσεις και παραλλαγές του TCP καθώς και πειράματα που δείχνουν το φαινόμενο αυτό.

Στο τρίτο κεφάλαιο περιγράφεται το φαινόμενο του packet reordering σε δίκτυα υψηλών ταχυτήτων (high-speed networks).

Στο πέμπτο κεφάλαιο αναλύεται μία παραλλαγή του TCP, το TCP-PR (TCP for persistent packet reordering – επιμονή στην αναδιάταξη πακέτων).

Τέλος, το έκτο κεφάλαιο αναφέρει περιληπτικά τα προηγούμενα κεφάλαια, και παρουσιάζονται συμπεράσματα.

## **ΚΕΦΑΛΑΙΟ 1**

### **1.1 ΕΙΣΑΓΩΓΗ ΣΤΟ TCP**

Ο σχεδιασμός του Πρωτοκόλλου Ελέγχου Μετάδοσης (TCP - TransmissionControlProtocol) έγινε με γνώμονα τη διασφάλιση της αξιοπιστίας κατά τη μετάδοση δεδομένων από αποστολέα σε παραλήπτη[1]. Το διαδίκτυο δεν είναι ίδιο με ένα δίκτυο, καθώς παράμετροι, όπως τοπολογία, εύρος ζώνης, μέγεθος πακέτων κ.ά., είναι διαφορετικά. Ο σχεδιασμός του TCP είναι τέτοιος, έτσι ώστε να μπορεί να τροποποιείται στα χαρακτηριστικά ενός διαδικτύου και να ανταπεξέρχεται πολλά είδη αστοχιών.

Κάθε μηχανή, που δέχεται το TCP, διαθέτει μία οντότητα μεταφοράς TCP, η οποία διαθέτει τις λειτουργίες διαδικασίας βιβλιοθήκης, διεργασίας χρήστη ή και τμήμα του πυρήνα. Σε όλες τις παραπάνω λειτουργίες, ο ρόλος της οντότητας του TCP είναι η διαχείριση των διαδρομών του TCP και η διασύνδεση με το επίπεδο του IP. Ο ρόλος μίας οντότητας TCP είναι να παραλαμβάνει ροές δεδομένων χρήστη, από τις τοπικές διεργασίες, και να τις διαχωρίζει σε τμήματα που έχουν μέγιστο μέγεθος 64 KB και να τα αποστέλει ως ξεχωριστά πακέτα IP.

Το επίπεδο IP δεν διασφαλίζει τη σωστή παράδοση των πακέτων. Για το λόγο αυτό, το TCP διαθέτει χρονόμετρα και εκπέμπει τα πακέτα, όταν είναι αναγκαίο. Επίσης, ρόλος του TCP είναι να φτάνουν τα πακέτα στη σωστή σειρά. Με άλλα λόγια, ο ρόλος του TCP είναι να παρέχει αξιοπιστία στους χρήστες, κάτι το οποίο δεν διαθέτει το IP.

### **1.2 ΤΟ ΜΟΝΤΕΛΟ ΥΠΗΡΕΣΙΑΣ ΤΟΥ TCP**



Για την αξιοποίηση της υπηρεσίας του TCP, ο αποστολέας και ο παραλήπτης αναπτύσσουν τερματικά σημεία, τις υποδοχές. Κάθε υποδοχή διαθέτει έναν αριθμό υποδοχής (διεύθυνση), η οποία περιλαμβάνει τη διεύθυνση IP του host (υπολογιστή υπηρεσίας) και ένα 16μπιτο αριθμό, με τοπική σημασία για το host, ο οποίος ονομάζεται θύρα (port). Η υπηρεσία του TCP λειτουργεί, όταν δημιουργεί μία σύνδεση ανάμεσα σε μία υποδοχή, στη μηχανή αποστολής και μία υποδοχή στη μηχανή λήψης.

Μια υποδοχή έχει την δυνατότητα να αξιοποιείται σε πολλές συνδέσεις ταυτόχρονα, δηλαδή σε μία υποδοχή ενδέχεται να τερματίζουν παραπάνω από δύο συνδέσεις.

Μία σύνδεση TCP απαρτίζεται από μία ροή byte και όχι από ροή μηνυμάτων. Η διατήρηση των ορίων των μηνυμάτων, από άκρο σε άκρο, δεν είναι εφικτή. Αν παραδείγματος χάριν, η διαδικασία του αποστολέα αναπτύξει τέσσερις εγγραφές των 512 byte, σε μία διαδρομή TCP, τα δεδομένα αυτά ενδέχεται να παραδοθούν στον παραλήπτη ως τέσσερα τμήματα των 512 byte ή δύο των 1024 byte ή ένα του 2048 byte ή κάπως αλλιώς [1]. Ο παραλήπτης δεν έχει την δυνατότητα να μάθει τον τρόπο με τον οποίο γράφτηκαν τα δεδομένα.

Στην περίπτωση που μία εφαρμογή μεταφέρει δεδομένα στο TCP, το TCP έχει τη δυνατότητα να αποστείλλει κατευθείαν ή επίσης μπορεί να αποθηκεύσει προσωρινά (με σκοπό να συλλέξει περισσότερη ποσότητα για ομαδική αποστολή), όπου καθορίζεται με την εκάστοτε περίπτωση. Υπάρχουν περιστάσεις, όπου η εφαρμογή επιθυμεί την άμεση αποστολή των δεδομένων. Αυτό επιτυγχάνεται με τη χρήση της σημαίας ΩΘΗΣΗ (push), η οποία αναφέρει στο TCP την άμεση μετάδοση.

Αξιοσημείωτο χαρακτηριστικό της υπηρεσίας TCP είναι τα επείγοντα (urgentdata). Σε περίπτωση που ένας διαδραστικός χρήστης πατήσει τα πλήκτρα Del ή Ctrl+C για να σταματήσει έναν απομακρυσμένο υπολογισμό, η εφαρμογή

του αποστολέα στέλνει κάποιες πληροφορίες ελέγχου στο TCP μαζί με τη σημαία ΕΠΕΙΓΟΝ (urgent). Με αυτή την πράξη το TCP σταματά τη συλλογή δεδομένων και μεταδίδει κατευθείαν όλα τα δεδομένα που έχει για τη σύνδεση αυτή.

Όταν παραλαμβάνονται τα επείγοντα δεδομένα, η εφαρμογή του παραλήπτη σταματά τη λειτουργία του, έτσι ώστε να επικεντρωθεί στη δοχή δεδομένων, με σκοπό να βρει τα επείγοντα δεδομένα. Ενώ το τέλος των επειγόντων δεδομένων επισυνάπτεται, έτσι ώστε να ενημερώνεται η εφαρμογή που σταματάνε, αντίθετα η αρχή των επειγόντων δεδομένων δεν επισυνάπτεται, καθώς η εφαρμογή είναι υπεύθυνη να την εντοπίσει.

### **1.3ΤΟ ΠΡΩΤΟΚΟΛΛΟ TCP**

Ένα κύριο συστατικό, που χαρακτηρίζει το TCP και επικρατεί σε μεγάλο βαθμό στο σχεδιασμό του πρωτοκόλλου, είναι πως κάθε byte μίας σύνδεσης TCP διαθέτει το δικό του 32μπιτο αριθμό ακολουθίας.

Τα δεδομένα μεταφέρονται από τις οντότητες αποστολής και λήψης TCP σε μορφή τμημάτων[1]. Το τμήμα TCP (TCPsegment) απαρτίζεται από μία κεφαλίδα, η οποία ακολουθείται από δεδομένα. Η κεφαλίδα έχει το σταθερό μέγεθος των 20 byte, ενώ τα δεδομένα τα οποία ακολουθούν, είναι από 0 ή περισσότερα bytes. Το μέγεθος των τμημάτων ορίζεται από το λογισμικό του TCP. Το μέγεθος των τμημάτων περιορίζεται μόνο από δύο παράγοντες:

- Το ωφέλιμο φορτίο των 65.515 byte του IP και,
- Τη μέγιστη μονάδα μεταφοράς ή MTU (MaximumTransferUnit), όπου κάθε τμήμα πρέπει να χωράει. Η MTU είναι 1500 byte (ωφέλιμο φορτίο ethernet).

Το TCP έχει ως βασικό μηχανισμό ελέγχου ροής, αυτό του κυλιόμενου παραθύρου. Κατά την αποστολή ενός τμήματος, ο αποστολέας βάζει σε χρήση και ένα χρονόμετρο. Μόλις το τμήμα αποσταλεί, το TCP του παραλήπτη επιστρέφει

ένα τμήμα με έναν αριθμό επιβεβαίωσης, ο οποίος ισούται με τον αριθμό ακολουθίας του αμέσως επόμενου αναμενόμενου από τον παραλήπτη segment δεδομένων. Εάν ο χρόνος αναμονής του αποστολέα λήξει πριν λάβει την επιβεβαίωση, αποστέλει ξανά το τμήμα.

Αν και το πρωτόκολλο φαίνεται απλό, υπάρχουν κάποιες ιδιαίζουσες περιπτώσεις. Τα τμήματα ενδέχεται να φτάσουν εκτός σειράς, με αποτέλεσμα να μην μπορούν να επιβεβαιώσουν. Επίσης, υπάρχουν περιπτώσεις, που ο χρόνος αναμονής λήγει, με αποτέλεσμα όταν αναμεταδίδει ο αποστολέας να περιέχει διαφορετικές περιοχές bytes.

Το TCP χρειάζεται να αντιμετωπίζει αυτά τα προβλήματα και να βρίσκει αποδοτικότερες λύσεις.

#### **1.4Η ΚΕΦΑΛΙΔΑ ΤΜΗΜΑΤΟΣ ΤΟΥ TCP**

Η αρχή κάθε τμήματος TCP είναι μία κεφαλίδα με σταθερό μέγεθος 20 byte. Η κεφαλίδα, σταθερού μεγέθους, έχει την δυνατότητα να ακολουθείται από επιλογές κεφαλίδας[1]. Στη συνέχεια μπορεί να ακολουθούν το μέγιστο 65.535 byte δεδομένων, εκ των οποίων τα 20 byte ανήκουν στην κεφαλίδα IP και άλλα 20 byte ανήκουν στην κεφαλίδα TCP. Δηλαδή, 65.495 byte είναι δεδομένα. Εκτός από το παραπάνω τμήμα, υπάρχουν και τμήματα με μηδενικά δεδομένα, τα οποία η χρήση τους έγγυται συνήθως για επιβεβαιώσεις και μηνύματα ελέγχου.

Η κεφαλίδα TCP χωρίζεται στα εξής πεδία:

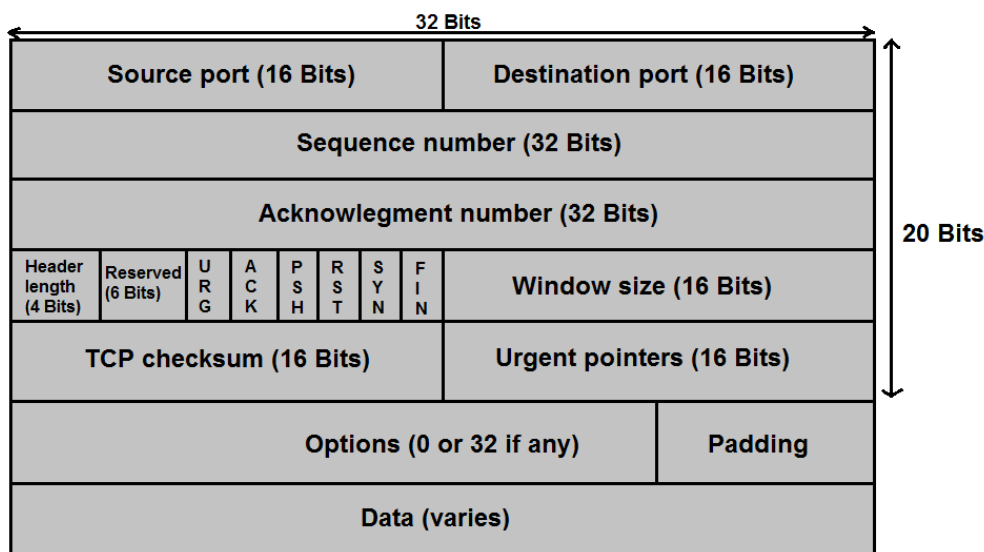
Η θύρα προέλευσης (sourceport) και η θύρα προορισμού (destinationport). Οι παραπάνω θύρες διέπουν τα τοπικά τερματικά σημεία της σύνδεσης. Ο συνδυασμός της θύρας και της διεύθυνσης IP του υπολογιστή υπηρεσίας, διαμορφώνουν ένα τερματικό σημείο των 48 byte. Η θύρα προέλευσης μαζί με τη θύρα προορισμού προσδιορίζουν μονοσήμαντα τη σύνδεση.

Αριθμός ακολουθίας (sequencenumber) και Αριθμός επιβεβαίωσης (acknowledgmentnumber), όπου ο acknowledgmentnumber προσδιορίζει το επόμενο byte που αναμένεται και όχι το byte που μόλις παρελήφθη. Το μήκος του κάθε πεδίου είναι 32 bit.

Το μήκος της κεφαλίδας TCP (TCPheaderlength) δείχνει το σύνολο των 32μπιτων λέξεων, που υπάρχουν στην κεφαλίδα TCP. Η χρησιμότητα αυτής της πληροφορίας είναι στο πεδίο Επιλογές (Options), του οποίου το μήκος του είναι μεταβλητό. Οπότε το ίδιο ισχύει και για την κεφαλίδα. Το πεδίο options είναι αυτό που επισημαίνει την αρχή των δεδομένων μέσα στο τμήμα, τα οποία αριθμούνται σε 32μπιτες λέξεις. Των μέγεθος των 32μπιτων λέξεων είναι ίσο με το μήκος της κεφαλίδας σε λέξεις. Οπότε η κατάληξη είναι ακριβώς η ίδια.

Στη συνέχεια της κεφαλίδας TCP υπάρχει ένα 6μπιτο πεδίο, το οποίο δεν αξιοποιείται. Λόγω της καλά σχεδιασμένης λειτουργίας του TCP, το συγκεκριμένο πεδίο δεν χρειάστηκε να επέμβει καθόλου, αντίθετα με άλλα πρωτόκολλα που θα είχαν αξιοποιήσει το πεδίο αυτό για να διορθώσει αρχικά τους σφάλματα.

Εν συνεχεία υπάρχουν έξι σημαίες, οι οποίες το μέγεθός τους είναι 1 bit η κάθε μία.



Σχήμα 1.1 TCPHeader

Η σημαία URG (urgent): Όταν η τιμή της σημαίας URG είναι 1, αυτό σημαίνει ότι ο Δείκτης επιγόντων χρησιμοποιείται (urgentpointer). Ο urgentpointer έχει τη δυνατότητα να δείχνει την απόσταση που βρίσκονται τα επείγοντα δεδομένα από τον τρέχοντα αριθμό ακολουθίας. Η λειτουργία αυτή έχει τη δυνατότητα να αντικαθιστά τα μηνύματα διακοπής.

Η σημαία ACK (acknowledgment): Όταν η τιμή του ACK είναι 1, αυτό δείχνει την εγκυρότητα του Αριθμού Επιβεβαίωσης (acknowledgmentnumber), ενώ στην περίπτωση που το ACK έχει την τιμή 0 ο αριθμός επιβεβαίωσης παραβλέπεται.

Η σημαία PSH (push): Στην συσκευριμένη σημαία στοχοποιούνται τα δεδομένα, στα οποία έχει γίνει χρήση της λειτουργίας ΩΘΗΣΗ. Το bitPSH έχει τη δυνατότητα να δίνει εντολή στον παραλήπτη να παραδώσει τα δεδομένα στην εφαρμογή κατά την άφιξή τους.

Η σημαία RST (reset): η χρήση της παραπάνω σημαίας είναι να επαναφέρει τη σύνδεση, η οποία έχει περιπλέξει, εξαιτίας παραγόντων, όπως κατάρρευση ενός υπολογιστή υπηρεσίας. Επιπλέον, η χρήση της σημαίας RST, είναι η μη αποδοχή ενός μη έγκυρου τμήματος ή η άρνηση σε μία προσπάθεια ανοίγματος σύνδεσης.

Η σημαία SYN(synchronize): Η χρήση της σημαίας SYN είναι για την εγκατάσταση συνδέσεων. Κατά την αίτηση σύνδεσης, τα bitSYN και ACK έχουν τιμές 1 και 0 αντίστοιχα, έτσι ώστε αν φανεί πως το πεδίο επιβεβαίωσης δεν είναι σε χρήση. Η απάντηση σύνδεσης έχει την τιμή 1 και για το bitSYN και για το bitACK. Με άλλα λόγια η τιμή της Σημαίας SYN υποδεικνύει τα μηνύματα ΑΙΤΗΣΗ ΣΥΝΔΕΣΗΣ και ΑΠΟΔΟΧΗ ΣΥΝΔΕΣΗΣ. Επιπλέον, το bitACK έχει τη δυνατότητα να διακρίνει αυτά τα δύο μηνύματα.

Η σημαία FIN (finish): Η χρήση της γίνεται για τον τερματισμό μίας σύνδεσης. Η σημαία αυτή αποφασίζει αν ο αποστολέας δεν έχει άλλα δεδομένα να

αποστέλλει. Αν και όταν τερματίζει μία σύνδεση, η διεργασία που την τερμάτισε εξακολουθεί να λαμβάνει δεδομένα.

Ο έλεγχος ροής στο TCP πραγματοποιείται μέσω ενός κυλιόμενου παραθύρου, του οποίου το μέγεθός του μεταβάλλεται[1]. Το πεδίο Μέγεθος παραθύρου (windowssize) δείχνει την ποσότητα των byte που μπορούν να σταλούν. Το μέγεθος του παραθύρου, όταν έχει την τιμή 0, συνεπάγεται πως είναι έγκυρο. Η τιμή αυτή δείχνει πως έχουν ληφθεί τα byte, μέχρι και το byte Αριθμός επιβεβαίωσης 1. Τη στιγμή εκείνη, όμως ο παραλήπτης μπρεί να μην επιθυμεί να λάβει άλλα δεδομένα. Οπότε έχει την δυνατότητα παύσης και μπορεί να επιτρέψει αργότερα την αποστολή, μεταδίδοντας ένα κομμάτι με τον ίδιο Αριθμό Επιβεβαίωσης και ένα μη μηδενικό πεδίο Μεγέθους Παραθύρου.

Για παραπάνω σιγουριά και αξιοπιστία, δίνεται ένα άθροισμα Ελέγχου (checksum). Το άθροισμα ελέγχου διαθέτει την κεφαλίδα, τα δεδομένα και την εικονική ψευδοκεφαλίδα. Σε αυτόν τον υπολογισμό, το πεδίο Άθροισμα ελέγχου του TCP δηλώνεται μηδέν και στο πεδίο δεδομένων προστίθεται ένα μηδενικό byte, στην περίπτωση που είναι περιττός αριθμός.

Το πεδίο Options διαθέτει μία μέθοδο προσθήκης βοηθητικών λειτουργιών, οι οποίες δεν αναπληρώνονται από την κανονική κεφαλίδα. Η πιο βασική επιλογή είναι η δυνατότητα που δίνεται σε κάθε υπολογιστή υπηρεσίας να εκχωρεί το μέγιστο ωφέλιμο φορτίο TCP που μπορεί να δεχθεί. Η αξιοποίηση μεγάλων τμημάτων αποφέρει το μεγαλύτερη απόδοση από εκείνη, των μικρών τμημάτων, λόγω της 20μπιτης κεφαλίδας, που έχει την δυνατότητα να αξιοποιείται σε περισσότερα δεδομένα. Αυτό όμως, μπορεί να μην είναι δυνατό σε μικρούς υπολογιστές υπηρεσίας, που ενδέχεται να μην είναι σε θέση να δεχθούν μεγάλα τμήματα.

Όταν δημιουργείται μία σύνδεση, δίνεται η δυνατότητα σε κάθε πλευρά να δηλώσει το μέγιστο όριό της και να δει κα του απέναντί του. Αν δεν αξιοποιηθεί η

επιλογή αυτή, τότε δηλώνεται αυτόματα το μέγεθος του 536 byte. Όλοι οι υπολογιστές υπηρεσίας είναι αναγκαίο να δέχονται τμήματα TCP με μέγιστο  $536+20=556$ byte., όπου δεν χρειάζεται να είναι ίδιο και στις δύο κατευθύνσεις.

## **1.5 ΕΓΚΑΘΙΔΡΥΣΗ ΣΥΝΔΕΣΕΩΝ ΣΤΟ TCP**

Η εγκαθίδρυση των συνδέσεων επιτυγχάνεται μέσω της τριπλής χειραψίας[1]. Για να γίνει μία τέτοια ενέργεια, χρειάζεται η μία πλευρά να αναμένει για μία εισερχόμενη σύνδεση με τη χρήση των λειτουργιών ΑΚΡΟΑΣΗ και ΑΠΟΔΟΧΗ.

Η άλλη πλευρά ενεργεί μέσω της λειτουργίας ΣΥΝΔΕΣΗ κατά την οποία καθορίζεται η διεύθυνση IP και η θύρα, η οποία επιθυμεί να συνδεθεί με το μεγαλύτερο κομμάτι του τμήματος TCP, που θέλει να αποδεχθεί κάποια δεδομένα του χρήστη. Η λειτουργία ΣΥΝΔΕΣΗ στέλνει ένα κομμάτι TCP με ενεργοποιημένο το bit ΣΥΓ και απενεργοποιημένο το bit ΕΠΙ και αναμένει απάντηση.

Κατά την άφιξη του τμήματος στον προορισμό του, η τοπική οντότητα TCP ελέγχει αν έχει ήδη λειτουργήσει η ΑΚΡΟΑΣΗ στο πεδίο Θύρα Προορισμού. Αν η ΑΚΡΟΑΣΗ δεν έχει λειτουργήσει, τότε αποστέλεται απάντηση με ενεργοποιημένο το bit ΕΠΑ, έτσι ώστε να μην επιτραπεί η σύνδεση.

Αν υπάρχει κάποια διεργασία σε αναμονή στην θύρα αυτή, τότε λαμβάνει το εισερχόμενο τμήμα TCP. Στην προκειμένη περίπτωση, η διεργασία αυτή έχει τη δυνατότητα να αποδεχθεί ή να απορρίψει τη σύνδεση. Στην περίπτωση που γίνει αποδοχή, επιστρέφει ένα τμήμα επιβεβαίωσης.

Στην περίπτωση που υπάρχουν δύο υπολογιστές υπηρεσίας που επιχειρούν ταυτόχρονη εγκαθίδρυση, γίνεται εγκαθίδρυση μίας μόνο σύνδεσης, καθώς οι συνδέσεις αναγνωρίζονται μοναδικά.

## **1.6 ΑΠΟΔΕΣΜΕΥΣΗ ΣΥΝΔΕΣΕΩΝ ΣΤΟ TCP**

Για να γίνει αποδέσμευση μίας σύνδεσης, οποιοδήποτε άκρο έχει τη δυνατότητα να στείλει ένα τμήμα TCP με ενεργοποιημένο το bitFIN [1]. Όταν γίνει επιβεβαίωση του σήματος FIN, τότε δεν γίνεται να μεταδοθούν νέα δεδομένα προς την κατεύθυνση αυτή. Όμως μπορούν να μεταφέρονται δεδομένα για άπειρο χρόνο προς την αντίθετη κατεύθυνση. Η σύνδεση αποδεσμεύεται πλήρως όταν κλείσουν και οι δύο κατευθύνσεις. Για την οριστική αποδέσμευση μίας σύνδεσης, χρειάζονται τέσσερα τμήματα TCP. Ένα μήνυμα FIN και ένα μήνυμα ACK από κάθε κατεύθυνση. Το σύνολο των μηνυμάτων μπορεί να πέσει σε τρία, καθώς το πρώτο μήνυμα ACK και το δεύτερο FIN έχουν την ευχέρεια να βρίσκονται στο ίδιο τμήμα.

Το «πρόβλημα των δύο στρατών» αντιμετωπίζεται με τη χρήση χρονομέτρων. Όταν αποσταλλεί το μήνυμα FIN, η απάντηση πρέπει να φτάσει σε χρόνο ίσο με το διπλάσιο της μέγιστης διάρκειας ζωής των πακέτων. Αν δεν συμβεί, τότε ο αποστολέας του μηνύματος FIN αποδεσμεύει τη σύνδεση. Η απέναντι πλευρά, από τη στιγμή που δεν έχει επικοινωνία, θα τελειώσει και ο δικός της χρόνος αναμονής.

## **1.7 ΜΟΝΤΕΛΟΠΟΙΗΣΗ ΔΙΑΧΕΙΡΙΣΗΣ ΣΥΝΔΕΣΕΩΝ ΤΟΥ TCP**

Οι διαδικασίες για την εγκαθίδρυση και την αποδέσμευση των συνδέσεων γίνονται μέσω 11 καταστάσεων μία μηχανής πεπερασμένης κατάστασης[1].

Οι συνδέσεις στην αρχή είναι σε κατάσταση ΚΛΕΙΣΤΗ (closed). Η κατάσταση αυτή μεταβάλλεται, όταν εκτελέσει ένα άνοιγμα, είτε αυτό είναι παθητικό είτε ενεργητικό. Αν η απέναντι πλευρά εκτελέσει ένα αντίστροφο, από αυτήν την πλευρά, άνοιγμα, τότε εγκαθιδρύεται μία σύνδεση και η κατάσταση μεταβένει σε



ΕΓΚΑΘΙΔΡΥΣΗ (established). Η κατάσταση μπορεί να μεταβεί ξανά σε ΚΛΕΙΣΤΗ, όταν μία από τις δύο πλευρές ξεκινήσει την αποδέσμευση.

## **1.8 ΠΟΛΙΤΙΚΗ ΜΕΤΑΔΟΣΗΣ ΣΤΟ TCP**

Όπως είναι γνωστό, η διαχείριση παραθύρων στο TCP, αλλά και στα περισσότερα πρωτόκολλα μετάδοσης δεδομένων, δεν είναι άμεσα συνδεδεμένη με τις επιβεβαιώσεις[1]. Αν λόγου χάρη, έστω ο παραλήπτης διαθέτει περιοχή προσωρινής αποθήκευσης 4096 byte, και ο αποστολέας μεταδώσει 2048 byte, όπου επιβεβαιώνονται από τον παραλήπτη. Στη φάση αυτή, αφού ο παραλήπτης έχει μόνο 2048 byte διαθέσιμη περιοχή προσωρινής αποθήκευσης, θα προβάλει ένα παράθυρο 2048 byte, το οποίο ξεκινά από το επόμενο byte, το οποίο βρίσκεται σε αναμονή.

Αφού ο αποστολέας μεταδώσει άλλα 2048 byte και επιβεβαιωθεί, το παράθυρο που προβάλεται είναι μηδενικό. Σε αυτή την περίπτωση, ο αποστολέας χρειάζεται να σταματήσει τη μετάδοση, έτσι ώστε μόλις η διεργασία εφαρμογής, στον υπολογιστή υπηρεσίας του παραλήπτη, αποδεσμεύσει μερικά δεδομένα από την προσωρινή μνήμη, το TCP θα έχει την ευχέρεια να προβάλει μεγαλύτερο παράθυρο.

Σχετικά με την περίπτωση που το παράθυρο είναι μηδενικό, ο αποστολέας δεν έχει την δυνατότητα να μεταδώσει δεδομένα, παρά μόνο σε δύο περιπτώσεις:

- Μπορεί να γίνει αποστολή επειγόντων δεδομένων. Αυτό δίνει τη δυνατότητα στον χρήστη να σταματήσει τη διεργασία που εκτελείται στην απομακρυσμένη περιοχή.

- Ο αποστολέας έχει τη δυνατότητα να στείλει 1 byte, έτσι ώστε ο παραλήπτης να προβάλει ξανά το επόμενο byte που περιμένει μαζί με το μέγεθος παραθύρου. Η επιλογή αυτή διατίθεται αυστηρά από το TCP, έτσι ώστε να

αποφευχθούν καταστάσεις, σε περίπτωση που χαθεί ποτέ μία ανακοίνωση μεγέθους παραθύρου.

Οι αποστολές δεν αναγκάζονται να στείλουν τα δεδομένα μόλις παραληφθούν από την εφαρμογή. Το ίδιο ισχύει και για τους παραλήπτες, δεν χρειάζεται αν στείλουν επιβεβαίωση άμεσα.

## ΚΕΦΑΛΑΙΟ 2

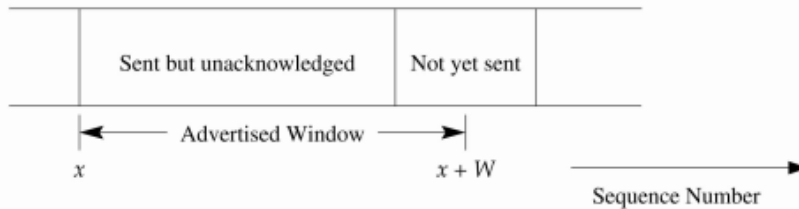
### 2.1 ΈΛΕΓΧΟΣ ΣΥΜΦΟΡΗΣΗΣ ΛΕΙΤΟΥΡΓΙΩΝ ΤΟΥ TCP

Για να επιτευχθεί καλή απόδοση, είναι απαραίτητο να ελέγχεται η συμφόρηση του δικτύου, έτσι ώστε ο αριθμός των πακέτων μέσα στο Διαδίκτυο να είναι χαμηλός. Εφαρμόστηκαν διάφορα μέτρα ελέγχου συμφόρησης [3] στο TCP, για να περιορίσουν το ρυθμό των δεδομένων που εισέρχονται στο Διαδίκτυο, ρυθμίζοντας το μέγεθος του παράθυρου συμφόρησης *cwnd* (*congestionwindow*), και τον αριθμό των μη αναγνωρισμένων τμημάτων που επιτρέπεται να σταλούν.

Τα μέτρα αυτά περιλαμβάνουν αργή εκκίνηση (*slowstart*), αποφυγή της κυκλοφοριακής συμφόρησης (*congestionavoidance*), γρήγορη αναμετάδοση (*fastretransmit*) και γρήγορη ανάκτηση (*fastrecovery*). Όταν εγκαθιστάται μια νέα σύνδεση, το TCP θέτει το *cwnd* σε ένα. Στην αργή εκκίνηση, το *cwnd* αυξάνεται κατά ένα, κάθε φορά που έχουμε μία επιβεβαίωση και το *ssthresh* (*Slow-StartThreshold*) TCP χρησιμοποιεί την απώλεια ως ένδειξη της συμφόρησης του δικτύου. Για να χαρακτηριστεί ένα τμήμα ως απώλεια κατά τη διάρκεια της μεταφοράς, μια πηγή πρέπει να περιμένει αρκετά μεγάλο χρονικό διάστημα χωρίς να λάβει ένα ACK για το τμήμα αυτό. Έτσι, ένα χρονόμετρο αναμετάδοσης συνδέεται με κάθε πακέτο που μεταδίδεται και ένα χρονόμετρο ενημερώνει πότε λήγει ο χρόνος και σηματοδοτεί την απώλεια ενός τμήματος.

Η περίοδος αναμετάδοσης του χρονικού ορίου RTO (*RetransmissionTimeout*) καθορίζεται από το άθροισμα του μέσου όρου των υπολογισμένων φυσιολογικών κινήσεων και από ένα πολλαπλάσιο της μέσης απόκλισης του RTT (*RoundTripTime/ Χρόνος μετάδοσης μετ' επιστροφής*) [4]. Όταν λήξει ο χρόνος, το *ssthresh* έχει οριστεί να μειώσει στη μέση τα εκκρεμή δεδομένα που στέλλονται

στο δίκτυο. Η διαδικασία της αργής εκκίνησης γίνεται ξεκινώντας με το cwnd να ισούται με ένα μέχρι το cwnd να προσεγγίσει το ssthresh.



Σχήμα 2.1 Απεικόνιση του χώρου του αριθμού ακολουθίας της πηγής και του αναδύομενου παραθύρου

Στη συνέχεια μπαίνει σε λειτουργία η αποφυγή συμφόρησης, όπου το cwnd αυξάνεται κατά ένα για κάθε RTT. Όταν τα δεδομένα ενός τμήματος, που φτάνουν είναι μεγαλύτερα από το αναμενόμενο ένα, ο παραλήπτης βρίσκει ένα κενό στον αύξοντα αριθμό ακολουθίας και έτσι αμέσως στέλνει ένα δεύτερο ACK (δηλαδή ένα ACK με τον επόμενο αναμενόμενο αριθμό ακολουθίας του segment δεδομένων) στην πηγή. Αν το κανάλι επικοινωνίας είναι σε τάξη, η λήψη ενός αντιγράφου του ACK συνεπάγεται με την απώλεια ενός τμήματος. Όταν η πηγή λαμβάνει τρία αντίγραφα ACKs, το fastretransmit ενεργοποιείται έτσι ώστε το κομμάτι που έχει χαθεί να επανεκπεμφθεί πριν από τη λήξη του χρόνου αναμετάδοσης. Το fastrecovery λειτουργεί ως ακόλουθος του fastretransmit.

Το fastretransmit υποδηλώνει την παρουσία ήπιας συμφόρησης του δικτύου. Το ssthresh ορίζει στο μισό το ποσό των εκκρεμών δεδομένων που αποστέλλονται στο δίκτυο. Δεδομένου ότι η λήψη ενός αντιγράφου ACK δηλώνει την αποχώρηση ενός τμήματος από το δίκτυο, το cwnd έχει οριστεί ως το σύνολο του ssthresh συν τον αριθμό των διπλότυπων ACKs που λήφθηκαν. Όταν ένα ACK φτάνει για κάποιο καινούργιο πακέτο, το cwnd επανέρχεται στο ssthresh, έτσι ώστε να αποφευχθεί η κυκλοφοριακή συμφόρηση.

Το packetreordering αναφέρεται στη συμπεριφορά του δικτύου, όπου η σειρά ορισμένων πακέτων στην ίδια ροή μεταβάλλεται, όταν τα πακέτα αυτά μεταφέρονται μέσα στο δίκτυο. Με άλλα λόγια, η σειρά λήψης μιας ροής πακέτων (ή τμημάτων) διαφέρει από τη σειρά αποστολής του.

## **2.2 ΑΙΤΙΕΣPACKETREORDERING**

### **2.2.1**

#### **ΠΟΛΛΑΠΛΗΔΡΟΜΟΛΟΓΗΣΗΣΕΠΙΠΕΔΟΠΑΚΕΤΟΥ(PACKETLEVEL MULTIPATHROUTING)**

Η πολλαπλή δρομολόγηση, είναι μία τεχνική εξισορρόπησης, η οποία κατανέμει το φορτίο στο δίκτυο, προκειμένου να ελαττωθεί η υπερφόρτωσή του. Έχει αποδειχθεί ότι η πολλαπλή δρομολόγηση εξισορροπεί το φορτίο εμφανώς καλύτερα από τη μονή δρομολόγηση και παρέχει καλύτερη απόδοση όσον αφορά την υπερφόρτωση και την ταχύτητα μέσω ενσύρματων/ασύρματων δικτύων.

Το πακέτο πολλαπλής διαδρομολόγησης επιτρέπει στα δεδομένα της ίδιας ροής να προωθηθούν μέσα από πολλές διαδρομές στον προορισμό, ώστε να επιτευχθεί η εξισορρόπηση του φορτίου σε εναλλασσόμενα πακέτα δικτύων. Ωστόσο, αυτά τα πακέτα μπορεί να ανασυνταχθούν κατά την άφιξη τους στον προορισμό, εξαιτίας των διαφοροποιήσεων που υπάρχουν στις καθυστερήσεις των διαδρομών .

### **2.2.2 Η ΚΙΝΗΣΗ ΤΗΣ ΔΙΑΔΡΟΜΗΣ (ROUTEFLUTTERINGROUTING)**

Η κίνηση της διαδρομής είναι ένα δικτυακό φαινόμενο, όπου η διαδρομή σε έναν συγκεκριμένο προορισμό ταλαντεύεται ανάμεσα σε μια σειρά από διαθέσιμες διαδρομές. Αυτό προκύπτει από την αστάθεια της πορείας, η οποία οφείλεται σε αβέβαιες συνδέσεις και βαριά φορτία ή ρήξη της κίνησης. Οι μετρήσεις του κόστους σύνδεσης που χρησιμοποιούνται στη δρομολόγηση των αλγόριθμων έχει σχέση με καθυστερήσεις ή συμφόρηση, όπου έχει προκληθεί κατά τη διάρκεια των συνδέσεων του δικτύου. Αυτό οδηγεί επίσης σε τοπολογικές αλλαγές στο ασύρματο περιβάλλον.

### **2.2.3 ΕΝΥΠΑΡΧΩΝ ΠΑΡΑΛΛΗΛΙΣΜΟΣ ΣΤΟΥΣ ΣΥΓΧΡΟΝΟΥΣ ΥΨΗΛΗΣ ΤΑΧΥΤΗΤΑΣ ΔΡΟΜΟΛΟΓΗΤΕΣ (INHERENT PARALLELISM IN MODERN HIGH SPEED ROUTERS)**

Οι σύγχρονοι δρομολογητές υποστηρίζουν ένα σύνολο συνδέσεων, όπου πακέτα της ίδιας ροής μπορούν να αποσταλούν με χαμηλότερη ταχύτητα και με πολύ φθηνότερες πολλαπλές και παράλληλες συνδέσεις, που είναι ενωμένες με τον επόμενο σε σειρά δρομολογητή. Για να αλλαχθούν τα πακέτα σε υψηλής ταχύτητας, ο δρομολογητής εξοικονομεί λειτουργία, ώστε οι εξερχόμενες θύρες που συνδέονται με τον επόμενο σε σειρά δρομολογητή να βρίσκονται σε αδράνεια. Αυτό γίνεται μόνο όταν δεν υπάρχουν εκκρεμή πακέτα που πρέπει να μεταβιβαστούν σε αυτόν το δρομολογητή. Αφού τα πακέτα μπορεί να διαφέρουν σε μέγεθος και οι συνδέσεις μπορεί να έχουν διάφορα εύρη ζώνης, τα πακέτα μπορούν να παραδοθούν σε διαφορετικό χρόνο και ως εκ τούτου φτάνουν στο γειτονικό δρομολογητή με διαφορετική σειρά από αυτή που εστάλησαν.

Η χρήση πολλαπλών ανέξοδων εφαρμογών με συγκεκριμένα ολοκληρωμένα κυκλώματα (ASICS: applicationspecificintegratedcircuits) για το πρωτόκολλο του διαδικτύου(IP)μπορεί να δημιουργήσει την ευκαιρία να επιταχυνθεί η μεταβιβαζόμενη ταχύτητα. Έτσι, ακόμα και όταν υπάρχει μια μόνο εξερχόμενη θύρα σύνδεσης με τον επόμενο σε σειρά δρομολογητή, τα πακέτα επεξεργάζονται από διαφορετικά ASICS και μπορούν να αναδιαταχθούν.

#### **2.2.4 ΑΝΑΜΕΤΑΔΟΣΗ ΣΕ ΕΠΙΠΕΔΟ ΣΥΝΔΕΣΗΣ (LINKLAYERRETRANSMISSION)**

Οι μηχανισμοί αναμετάδοσης σε επίπεδο σύνδεσης [4] έχουν προταθεί για να ανακτηθούν αποτελεσματικά οι απώλειες μετάδοσης που οφείλονται στο υψηλό ποσοστό σφαλμάτων στα κανάλια των ασύρματων δικτύων. Τέτοια reorderedpackets στέλνονται μόνο όταν ανιχνευθούν οι απώλειες. Αυτά τα πακέτα μπορούν στη συνέχεια να εναλλαχθούν με άλλα πακέτα που ανήκουν στην ίδια ροή.

#### **2.2.5 ΠΑΥΣΗ ΠΡΟΩΘΗΣΗΣ ΤΟΥ ΔΡΟΜΟΛΟΓΗΤΗ (ROUTERFORWARDINGLULLS)**

Μερικοί δρομολογητές μπορούν να διακόψουν τη δραστηριότητα προώθησής τους για πακέτα ρυθμίσεων (bufferedpackets), όταν επεξεργάζονται ενημερώσεις για τη δρομολόγηση. Αυτά τα ρυθμισμένα πακέτα που εμπλουτίζονται με νέες αφίξεις, προκαλούν το packetreordering.

## **2.3 ΟΙ ΕΠΙΠΤΩΣΕΙΣ ΤΟΥ PACKETREORDERING ΣΤΟ TCP**

Το TCP βασίζεται στη χρήση ενός αθροιστικού ACK για να ανακοινώσει την παραλαβή του τμήματος. Ο ρυθμός με τον οποίο μια πηγή λαμβάνει ACKs δείχνει πόσο γρήγορα μπορεί να εισάγει τα τμήματα TCP στο δίκτυο και στοσυνδεδεμένο προορισμό του. Με συνεχές και ουσιαστικό packetreordering, το TCP αναμεταδίδει ψευδώς τα τμήματα, κρατά το παράθυρο συμφόρησης του άσκοπα μικρό, χάνει τον ACK-χρονισμό του, και υποτιμά το εκτιμώμενοRTT .

### **2.3.1 ΠΑΡΑΣΙΤΙΚΕΣ ΑΝΑΜΕΤΑΔΟΣΕΙΣ ΤΜΗΜΑΤΟΣ (SPURIOUSSEGMENTRETRANSMISSION)**

Το packetreordering οδηγεί τα αρχικά πακέτα δεδομένων κάποιων ερχόμενων τμημάτων, να διαφέρουν από τα αναμενόμενα ενός προορισμού. Δηλαδή, ο προορισμός βρίσκει ένα κενό στην ακολουθία κατά τη λήψη του τμήματος. Στη συνέχεια, δημιουργεί διπλά ACKs και τα στέλνει στη συνδεόμενη πηγή του. Όταν η πηγή λαμβάνει διαδοχικά τρία τέτοια διπλά ACKs, αναμεταδίδεται απώλεια του ήδη υπάρχοντος τμήματος (αν και στην πραγματικότητα δεν υπάρχει ζημιά). Το συνεχές και ουσιαστικό packetreordering προκαλεί συχνά σε ορισμένα τμήματα TCP να αναμεταδοθούν ψευδώς και χωρίς λόγο, με αποτέλεσμα την κλασική κατάρρευση της συμφόρησης [5].



### **2.3.2 ΚΡΑΤΩΝΤΑΣ ΤΟ ΠΑΡΑΘΥΡΟ ΣΥΜΦΟΡΗΣΗΣ ΑΔΙΚΑΙΟΛΟΓΗΤΑ ΜΙΚΡΟ (KEEPING CONGESTION WINDOW UNNECESSARILY SMALL)**

Το fastrecovery ενεργοποιείται πάντα με το fastretransmit. Το ψευδές fastretransmit όχι μόνο παράγει ακόμα περισσότερο περιττό φόρτο εργασίας στο δίκτυο και στον προορισμό, αλλά και μειώνει στο μισό το παράθυρο συμφόρησης. Έτσι, το παράθυρο συμφόρησης παραμένει μικρό σε σχέση με το διαθέσιμο εύρος ζώνης της διαδρομής του, με συνεχές και ουσιαστικό packetreordering.

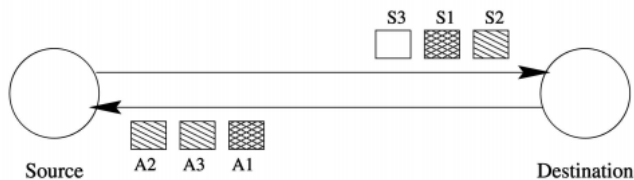
### **2.3.3 ΑΠΩΛΕΙΑ ACK-ΧΡΟΝΙΣΜΟΥ (LOSS OF ACK-CLOCKING)**

Το packetreordering δεν προκαλεί μόνο τα τμήματα των δεδομένων, αλλά και τα ACKs, να καταλήγουν σε έναν προορισμό εκτός σειράς. Το προηγούμενο φαινόμενο ονομάζεται αναδιάταξη προώθησης διαδρομής, ενώ το δεύτερο είναι γνωστό ως αντίστροφη διαδρομή αναδιάταξης[6]. Μια απεικόνιση της αναδιάταξης προώθησης της διαδρομής και της αντίστροφης διαδρομής αναδιατάξεως φαίνεται στο σχήμα 2.2.

Υποθέτοντας ότι τα τμήματα στέλνονται από την πηγή με τη σειρά S1, S2, S3, αλλά το τμήμα S1 φτάνει στον προορισμό μετά το τμήμα S2. Αυτό αντιπροσωπεύει την αναδιάταξη προώθησης της διαδρομής. Το ACK A1 φτάνει μετά το ACKs A2 και A3 στην πηγή. Αυτό απεικονίζει τη διαδρομή αντίστροφης αναδιάταξης.

Η χρονομέτρηση του ACK ή αυτό- χρονομέτρηση αναφέρεται στην ιδιότητα που ο δέκτης μπορεί να δημιουργήσει ACKs όχι γρηγορότερα από ό,τι τα τμήματα

των δεδομένων μπορούν να περάσουν μέσα από το δίκτυο. Για την αναδιάταξη προώθησης της διαδρομής, ένα ACK για πολλά νέα τμήματα, το οποίο ακολουθεί μια σειρά διπλών ACKs, μπορεί με τη σειρά του να επιτρέψει σε μια πηγή να εισάγει πολλά εκκρεμή τμήματα στα δίκτυα. Ακόμα και όταν δεν υπάρχει τμήμα δεδομένων που αναδιατάσσεται, τα αποδιοργανωμένα ACKs οδηγούν σε μια πηγή μετάδοσης, πολλά τμήματα μαζί και όχι ένα ή δύο τμήματα ανά ACK. Αυτό προκαλεί την απώλεια ACKχροניσμού και μια πολύ πιο δυναμική κίνηση, η οποία μπορεί να οδηγήσει σε παροδική συμφόρηση του δικτύου και στην κατάρρευση της συμφόρησης από τα πακέτα που δεν έχουν παραδοθεί.



Σχήμα 2.2: Αναδιάταξη προώθησης της διαδρομής και της αντίστροφης διαδρομής αναδιατάξης

### 2.3.4 ΜΕΙΩΣΗ ΕΚΤΙΜΩΜΕΝΟΥ RTT ΚΑΙ RTO (UNDERSTATING ESTIMATED RTT AND RTO)

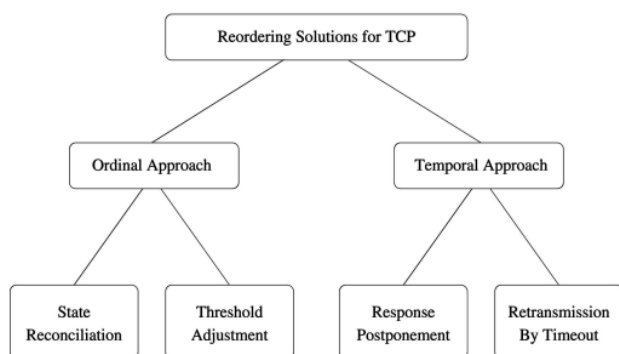
Κάθε φορά που ένα τμήμα αναμεταδίδεται, μια πηγή δεν μπορεί να προσδιορίσει εάν ένα ληφθέν ACK είναι από την πρώτη μετάδοση ή από την αναμετάδοση του τμήματος. Ο αλγόριθμος του Karn [7] μειώνει το πρόβλημα, απορρίπτοντας όλα τα μετρούμενα δείγματα RTT μέχρι ένα ACK να αναγνωρίσει ένα τμήμα που δεν έχει αναμεταδοθεί. Αφού ένα fastretransmit είναι πιθανό να

ανταποκριθεί σε ένα τμήμα που έχει μεγαλύτερη καθυστέρηση στη διαδρομή, η χρήση του αλγόριθμου του Karn αποσκοπεί σε μια θέση δειγματοληψίας των μεγάλων δειγμάτων RTT [8]. Με το συνεχές και ουσιαστικό packetreordering, τα δείγματα αυτά θα πρέπει να απορρίπτονται. Έτσι, τα εκτιμώμενα RTT και RTO είναι υποβαθμισμένα.

## 2.4 ΤΑΞΙΝΟΜΗΣΗ ΛΥΣΕΩΝ ΑΝΑΔΙΑΤΑΞΗΣ ΓΙΑ ΤΟ TCP

Για να είναι ένα περιβάλλον δικτύου σε σειρά, ένας προορισμός λαμβάνει τμήματα με την ίδια σειρά όπως στην αποστολή τους. Ο προορισμός αντιλαμβάνεται την εμφάνιση μιας απώλειας πακέτου, όταν φτάνει ένα μη αναμενόμενο τμήμα. Στη συνέχεια μπορούν να ενσωματωθούν οι πληροφορίες των τμημάτων που λείπουν στα επόμενα ACKs σε μια πηγή, έτσι ώστε η πηγή να μπορεί να αναμεταδώσει τα χαμένα τμήματα στον προορισμό. Ακόμα και όταν ένα τμήμα που φτάνει είναι ένα τμήμα που λήφθηκε πρόσφατα και δεν είναι το αναμενόμενο, ένας προορισμός δεν μπορεί να προσδιορίσει κατά πόσον το αναμενόμενο τμήμα έχει χαθεί στο δίκτυο ή είναι απλά αναδιαταγμένο. Η πηγή και ο προορισμός πρέπει να συλλέξουν επιπλέον πληροφορίες από το δίκτυο για να γίνει διάκριση μεταξύ αυτών των δύο πιθανοτήτων.

Οι αλγόριθμοι που συζητήθηκαν μπορούν να εφαρμοστούν σε έναν πελάτη TCP για να δημιουργήσει συμφόρηση αντιδράσεων όταν συμβαίνει το packetreordering, και σε ένα δρομολογητή που συμμετέχει για να αναφέρουν την πληροφορία της απώλειας πακέτων σε έναν πελάτη TCP. Η ταξινόμηση που χρησιμοποιείται στην έρευνά απεικονίζεται στο σχήμα 2.3.



Σχήμα 2.3: Ταξινόμηση λύσεων αναδιάταξης για το TCP

Έχουν γίνει κατηγοριοποιηθεί οι λύσεις αναδιάταξης για το TCP σε δύο διαφορετικές στρατηγικές, δηλαδή, την τακτική προσέγγιση και τη χρονική προσέγγιση.

#### 2.4.1 ΤΑΚΤΙΚΗ ΠΡΟΣΕΓΓΙΣΗ (ORDINAL APPROACH)

Η τακτική προσέγγιση είναι μια συλλογή από μεθόδους που επεξεργάζονται τις πληροφορίες των πακέτων και των ACKs που λαμβάνονται σε σειρά ώστε να προκύψει ένα συμπέρασμα και μια πιο κατάλληλη αντίδραση συμφόρησης όταν συμβαίνει packetreordering. Η τακτική προσέγγιση μπορεί να διαιρεθεί περαιτέρω σε αλγορίθμους για συμφωνία, και για την προσαρμογή ορίου. Για να γίνει συμφωνία, ο TCP πελάτης ανακαλύπτει ποια κατηγορία ή ποια ACK έχουν αναδιαταχθεί. Στη συνέχεια αντιδρά, με την ανάκτηση παλαιότερων αντιδράσεων συμφόρησης και με την απενεργοποίηση μελλοντικών αντιδράσεων συμφόρησης για κάποια χρονική περίοδο. Για την προσαρμογή του ορίου, ο πελάτης TCP ψάχνει για ένα κατάλληλο εφεδρικό όριο αναγνώρισης, dupthresh, για να το αποφύγει προληπτικά, όποτε είναι δυνατόν, προκαλώντας ένα ψευδές

fastretransmit και fastrecovery, καθώς και ένα χρονικό όριο αναμετάδοσης ταυτόχρονα.

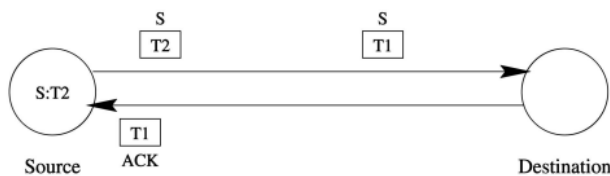
## ΣΥΜΦΩΝΙΑ ΚΑΤΑΣΤΑΣΗΣ (STATERECONCILIATION)

**ΑλγόριθμοςEifel:** ΟιLudwig και Katz πρότειναν τον αλγόριθμο Eifel [9] ,για να εξαλειφθεί η ασάφεια αναμετάδοσης και να λύσει τα προβλήματα απόδοσης που προκαλούνται από ψευδείς αναμεταδόσεις. Μια πηγή χρησιμοποιεί την επιλογή χρονοσφαγίδαςTCP, για να εισάγει την τρέχουσα χρονοσφαγίδα στην κεφαλίδα του κάθε εξερχόμενου τμήματος σε έναν προορισμό. Όταν ο προορισμός στέλνει ACKs, συμπεριλαμβάνει τις αντίστοιχες χρονικές σφραγίδες στα ACKs. Για να εξαλειφθεί η ασάφεια της αναμετάδοσης, η πηγή αποθηκεύει πάντα τη χρονοσφαγίδα της πρώτης αναμετάδοσης ενός τμήματος. Όταν το πρώτο ACK για το τμήμα που έχει αναμεταδοθεί, φτάνει, η πηγή συγκρίνει τη χρονική σφραγίδα του ACK με την αποθηκευμένη χρονοσφαγίδα. Εάν η αποθηκευμένη χρονική σφραγίδα είναι μεγαλύτερη, η αναμετάδοση θεωρείται ψευδής .

Το σχήμα 2.4 δείχνει πώς λειτουργεί ο αλγόριθμος Eifel. Όταν η πηγή στέλνει το τμήμαS την πρώτη φορά σε Ωρα T1, εισάγει την τρέχουσα χρονοσφαγίδα T1 στην κεφαλίδα του τμήματος. Τη χρονική στιγμή t2, η πηγή αρχίζει μια απόκριση συμφόρησης αναμεταδίδοντας το Τμήμα S. Το αρχικό τμήμα διαφέρει με αυτό που αναμεταδόθηκε αφού το μεταγενέστερο περιέχει τη χρονοσφαγίδα T2 αντί του T1. Όταν ο προορισμός λαμβάνει πρώτα το αρχικό Τμήμα S, στέλνει ένα ACK με τη χρονοσφαγίδα του τμήματος, δηλαδή, T1. Όταν φτάνει το ACK για το τμήμα, η πηγή διαπιστώνει ότι η επαναλαμβανόμενη χρονοσφαγίδα, T1, είναι μικρότερη από την αποθηκευμένη, T2. Επομένως η αναμετάδοση χαρακτηρίζεται ως ψευδής. Για την επίλυση των προβλημάτων που προκλήθηκαν από τις ψευδείς

αναμεταδόσεις, η πηγή αποθηκεύει επίσης τις τρέχουσες τιμές του αργού ορίου εκκίνησης, *ssthresh*, και το μέγεθος του παραθύρου συμφόρησης, *cwnd*, όταν ένα τμήμα επανεκπέμπεται την πρώτη φορά. Όταν εντοπιστεί η ψευδής αναμετάδοση που έχει ως αποτέλεσμα την μεμονωμένη αναμετάδοση των παλαιότερων τμημάτων που ήταν σε εκκρεμότητα, η πηγή αποκαθιστά το *ssthresh* και το *cwnd* με τις αποθηκευμένες τιμές.

Έχει αποδειχθεί ότι αυτή η τεχνική είναι απλή και αποτελεσματική στη βελτίωση της απόδοσης του TCP με πορεία αναδιάταξης. Ωστόσο, οι ρήξεις των τμημάτων TCP μπορεί να εισβάλλουν στο δίκτυο όταν η κατάσταση έχει αποκατασταθεί. Εκτός αυτού, το σύστημα δεν λειτουργεί όταν αναμεταδοθούν και τα αρχικά αναδιαταγμένα τμήματα.



Σχήμα 2.4 Λειτουργία του αλγόριθμου Eifel

**TCP- DOOR:** Οι Wang και Zhang ανέπτυξαν το TCP για ανίχνευση εκτός σειράς και απόκρισης (TCP- DOOR) [10]. Στα γεγονότα εκτός σειράς θεωρούνται ότι συνεπάγονται αλλαγές δρομολογίων στα δίκτυα, κάτι το οποίο συμβαίνει συχνά σε κινητά *ad hoc* δίκτυα. Ο αριθμός ακολουθίας των πακέτων TCP και ο αύξον αριθμός των εφεδρικών ACK, ή οι τρέχουσες χρονικές σφραγίδες, εισάγονται σε κάθε δεδομένα και τμήματα ACK, αντίστοιχα, για την ανίχνευση αναδιατασσομένων δεδομένων και πακέτων ACK. Όταν ανιχνεύονται γεγονότα εκτός σειράς, μια πηγή μπορεί είτε να απενεργοποιήσει προσωρινά τον έλεγχο συμφόρησης ή να εκτελέσει ανάκτηση κατά τη διάρκεια της αποφυγής συμφόρησης. Με την προσωρινή απενεργοποίηση του ελέγχου συμφόρησης, η

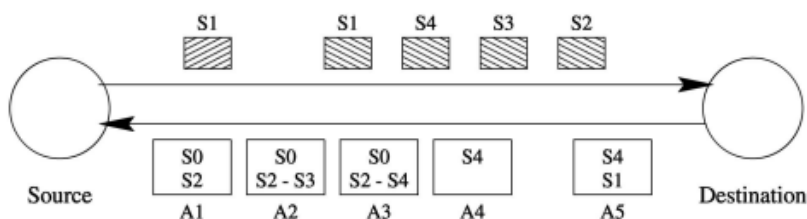
πηγή θα διατηρήσει την κατάσταση σταθερή για ένα χρονικό διάστημα,  $t_1$  δευτερόλεπτα, μετά την ανίχνευση ενός συμβάντος εκτός σειράς. Με την άμεση επαναφορά κατά τη διάρκεια αποφυγής συμφόρησης, η πηγή ανακτά αμέσως την κατάσταση πριν από την αντίδραση της συμφόρησης, η οποία έχει χρησιμοποιηθεί πριν από  $t_2$  δευτερόλεπτα.

Ωστόσο, το TCP-DOOR δεν κάνει διάκριση μεταξύ της κανονικής πορείας αναδιάταξης ή της αντίστροφης πορείας αναδιάταξης. Οι αποκρίσεις είναι κατάλληλες για να μειωθούν κάποια προβλήματα απόδοσης, που προκλήθηκαν από την προς τα εμπρός πορεία αναδιάταξης. Δεν βοηθούν στη μείωση της μεγάλης κίνησης, και μάλιστα επιβαρύνουν τη συμφόρηση του δικτύου υπό την αντίστροφη πορεία αναδιάταξης. Εκτός αυτού, το TCP-DOOR δεν αποδίδει καλά σε πυκνό περιβάλλον δικτύου με σημαντικό συνεχές packetreordering. Απενεργοποιεί τον έλεγχο συμφόρησης για ένα χρονικό διάστημα κάθε φορά που ανιχνεύεται ένα στοιχείο εκτός σειράς, το οποίο μπορεί να οδηγήσει σε κατάρρευση της συμφόρησης από πακέτα που δεν έχουν παραδοθεί.[5]

**DSACKTCP:** Ο Floyd και οι άλλοι συζήτησαν τη χρήση της διπλής επιλεκτικής αναγνώρισης (SACK) [11], για τον εντοπισμό τμημάτων αναδιάταξης και απέσυραν τη συσχετισμένη ανταπόκριση ψευδούς συμφόρησης. Το DSACK είναι μια επέκταση της επιλεκτικής αναγνώρισης (SACK) για το TCP. Στόχος της είναι να χρησιμοποιήσει την επιλογή SACK για διπλότυπα τμήματα. Το πρώτο κομμάτι του πεδίου επιλογής του SACK χρησιμοποιείται για να αναφέρει τον αριθμό ακολουθίας ενός διπλότυπου τμήματος το οποίο έχει προκαλέσει το ACK. Όταν ανιχνεύεται η συμφόρηση, το cwnd αποθηκεύεται πριν γίνει μείωση. Όταν μια πηγή διαπιστώνει ότι έχει κάνει ψευδή αντίδραση συμφόρησης, με βάση την άφιξη ενός DSACK, εκτελεί αργή εκκίνηση για να αυξηθεί το τρέχον cwnd στο αποθηκευμένο cwnd πριν από την αποφυγή της συμφόρησης. Εκτελώντας αργή εκκίνηση κατά τη διάρκεια αποκατάστασης της κατάστασης, επιτρέπει στο TCP να

ξαναζητήσει ACK-χρονομέτρηση και να αποφύγει τη μεγάλη κυκλοφορία μέσα στο δίκτυο.

Το σχήμα 2.5 δείχνει ότι το DSACK χρησιμοποιείται για την ανίχνευση packetreordering. Υποθέτοντας ότι ο Τομέας S1 είναι αναδιαταγμένος με τέτοιο τρόπο, έτσι ώστε να φτάνει μετά το τμήμα S4 στον προορισμό. Το τελευταίο τμήμα που αναγνωρίστηκε είναι το Τμήμα S0. Στην περίπτωση αυτή, ο προορισμός στέλνει τρία αντίγραφα ACKs A1, A2 και A3 (με την ίδια αθροιστική ACK για το Τμήμα S0, αν και τα πεδία επιλογής SACK διαφέρουν) προς την πηγή, έτσι ώστε το Τμήμα S1 να αναμεταδοθεί (υποθέτοντας ότι το dupthresh είναι τρία). Όταν ο προορισμός λαμβάνει το τμήμα S1 που αναμεταδόθηκε, στέλνει ένα αντίγραφο ACK A5 για το Τμήμα S4, αλλά το πρώτο σύνολο του πεδίου επιλογής του SACK αναγνωρίζει την άφιξη ενός αντιγράφου του τομέα S1. Η πηγή τότε ξέρει ότι ο Τομέας S1 έχει αναμεταδοθεί ψευδώς λόγω του packetreordering. Αυτή η μέθοδος μπορεί εύκολα να συνδυαστεί με ένα σύστημα χρησιμοποιώντας τα στοιχεία του DSACK για να ρυθμιστεί το dupthresh για δραστική πρόληψη προκαλώντας ψευδείς αντιδράσεις συμφόρησης στο μέλλον. Ο αλγόριθμος Blanton-Allman , το RR-TCP και ο αλγόριθμος Leung-Ma, έχουν υιοθετήσει αυτή την τεχνική για την ανίχνευση.



Σχήμα 2.5 Ανίχνευση packet reordering μέσω DSACK

## ΌΡΙΟ ΠΡΟΣΑΡΜΟΓΗΣ (THRESHOLDADJUSTMENT)



**Ο αλγόριθμος Lee-Park-Choi (Sender-SideSolution):** Ο Lee και οι άλλοι [12] πρότειναν μια λύση από τη μεριά του αποστολέα για να βελτιώσει την επίδοση του TCP για την προς τα εμπρός διαδρομή αναδιάταξης σε πολλαπλές διαδρομές. Το dupthresh προβλέπεται να αυξηθεί λογαριθμικά με τον αριθμό των διαδρομών που χρησιμοποιούνται. Έτσι, μία πηγή πρέπει να λάβει ένα μεγαλύτερο αριθμό των διπλότυπων ACKs πριν ενεργοποιηθεί αντίδραση συμφόρησης, όταν οι περισσότερες διαδρομές χρησιμοποιούνται ταυτόχρονα για τη μετάδοση μιας ροής TCP.

Ωστόσο, όταν χρησιμοποιείται δρομολόγηση πολλαπλών διαδρομών σε επίπεδο πακέτου, για τη μετάδοση δεδομένων, το επίπεδο του packetreordering μπορεί να εξαρτάται από τις διαφορές στις καθυστερήσεις διαδρομής και στο πως τα πακέτα που ανήκουν σε μια και μόνο ροή διανέμονται σε αυτές τις διαδρομές. Έτσι, δεν υπάρχει άμεσος συσχετισμός μεταξύ του dupthresh και του αριθμού των διαδρομών που συμμετέχουν.

**Οι αλγόριθμοι Blanton-Allman:** Οι Blanton και Allman πρότειναν τρεις εναλλακτικές λύσεις για να ρυθμίσουν δυναμικά το dupthresh. Η πρώτη εναλλακτική λύση, συμβολίζεται ως Blanton-Allman: INC (Increase), να αυξηθεί το dupthresh από κάποια σταθερά, κάθε φορά που ανιχνεύεται ένα ψευδές fastretransmit. Η δεύτερη εναλλακτική, συμβολίζεται ως Blanton- Allman: AVG (Average), να αυξηθεί το dupthresh με τη λήψη του μέσου όρου του τρέχοντος dupthresh και τον αριθμό των διπλότυπων ACKs που απαιτούνται για να αποσαφηνιστεί η αναδιάταξη από την απώλεια, όταν ανιχνεύεται μια ψευδής αναμετάδοση. Η τρίτη εναλλακτική, που συμβολίζεται ως Blanton-Allman: EWMA, είναι να εισχωρήσει το dupthresh σε ραγδαία αυξανόμενο μέσο όρο κίνησης (EWMA: Exponentially Weighted Moving Average) του μήκους των ληφθέντων στοιχείων αναδιάταξης.

Για όλους αυτούς τους αλγόριθμους, το dupthresh επανέρχεται σε τρία, μετά τη λήξη του χρόνου αναμετάδοσης, προκειμένου να μειώσει τις μελλοντικές δαπανηρές λήξεις χρόνου αναμετάδοσης. Οι μελέτες επίσης επέκτειναν τον περιορισμένο αλγόριθμο εκπομπής, ο οποίος επιτρέπει σε μια πηγή να στείλει ένα καινούργιο τμήμα με την παραλαβή των πρώτων δύο διπλότυπων ACKs, έτσι ώστε ένα νέο τμήμα θα μπορούσε να σταλεί σε κάθε δύο διπλότυπα ACKs που λήφθηκαν αργότερα [13]. Αυτό βοηθά στη διατήρηση της χρονομέτρησης του ACK και αποφεύγει την έγχυση μεγάλης κυκλοφορίας όταν ένα ACK φτάνει για ένα μεγάλο τμήμα.

Τα αποτελέσματα της προσομοίωσής τους έδειξαν ότι, σε σύγκριση με το προεπιλεγμένο dupthresh στο τρία, οι προτεινόμενες τεχνικές βελτίωσαν την απόδοση σύνδεσης και μείωσαν τον αριθμό των περιττών αναμεταδόσεων. Ωστόσο, οι αλγόριθμοί τους έχουν τρεις σημαντικές ελλείψεις. Αρχικά, η προσαρμογή του dupthresh για κάποιους προτεινόμενους αλγόριθμους δεν είναι αρκετά προσαρμοστική στη δυναμική συμπεριφορά των στοιχείων αναδιάταξης. Δεύτερον, δεν υπάρχει προσαρμοστικός μηχανισμός για τη σημαντική μείωση του dupthresh, εκτός από τον τρίτο αλγόριθμο. Αυτοί οι αλγόριθμοι αποτυγχάνουν να προσαρμόσουν ένα κατάλληλο dupthresh για να επιτύχει ισορροπία μεταξύ του κόστους ενός ψευδούς fastretransmit και της λήξης του χρονικού ορίου αναμετάδοσης. Επίσης, είναι αδύνατον να ψάξουν για μια κατάλληλη, αλλά μειωμένη τιμή dupthresh όταν η έκταση της αναδιάταξης μειώνεται. Τρίτον, η επαναφορά των dupthresh σε τρία μετά τη λήξη του χρόνου αναμετάδοσης καταστρέφει όλο το ιστορικό των πληροφοριών όσον αφορά την αναδιάταξη προώθησης της διαδρομής στα δίκτυα. Παίρνει επιπλέον χρόνο για να επιτρέψει στο dupthresh να αυξηθεί μέχρι την επιθυμητή τιμή.

**RR-TCP:** Ο Zhang και οι άλλοι επινόησαν την ισχυρή αναδιάταξη του TCP (RR-TCP) ως επέκταση των αλγορίθμων Blanton-Allman[14], αλλά διαφέρουν με

τρεις τρόπους. Αρχικά, το RR-TCP χρησιμοποιεί έναν διαφορετικό μηχανισμό για να ρυθμίσει δραστικά το dupthresh. Διατύπωσαν μια συνδυασμένη συνάρτηση κόστους για τα χρονικά όρια των αναμεταδόσεων, για τα ψευδή fastretransmit, και για την περιορισμένη μετάδοση, ώστε να προσαρμοστεί η αποφυγή του ψευδούς fastretransmit (αναλογία FA). Η αναλογία FA, η οποία αντιπροσωπεύει το τμήμα των στοιχείων αναδιάταξης που πρέπει να αποφεύγονται, προκειμένου να ελαχιστοποιηθεί το κόστος, μπορεί στη συνέχεια να χρησιμοποιηθεί για να βρεθεί το αντίστοιχο dupthresh. Έτσι, αυτό παρέχει ένα μηχανισμό για να αυξήσει ή να μειώσει δραστικά το dupthresh, αλλάζοντας την αναλογία FA με βάση τις τρέχουσες συνθήκες του δικτύου.

Δεύτερον, στις μελέτες προτείνεται μια άλλη εκτεταμένη έκδοση του περιορισμένου αλγόριθμου μετάδοσης. Η επέκταση αυτή επιτρέπει σε μια πηγή να στείλει μέχρι και ένα χρονισμένο ACK στο επιπλέον παράθυρο συμφόρησης των δεδομένων. Τρίτον οZhang πρότεινε μια ιδέα για να διορθωθεί το σφάλμα δειγματοληψίας απέναντι στα RTT δείγματα για τις εκτιμήσεις του RTT και RTO. Στον αλγόριθμο του Karn αντί να παρακαμφθούν τα δείγματα για τα τμήματα που έχουν αναμεταδοθεί, λαμβάνεται ένα δείγμα RTT για κάθε τμήμα που έχει αναμεταδοθεί, παίρνοντας τη μέση τιμή των RTTs τόσο για τις πρώτες όσο και για τις δεύτερες μεταδόσεις του τμήματος αυτού.

Τα αποτελέσματα της προσομοίωσης έδειξαν ότι το RR-TCP θα μπορούσε να βελτιώσει σημαντικά την απόδοση του TCP σε αναδιατασσόμενα δίκτυα. Όταν 1 με 2 τις εκατό των τμημάτων επιλέχθηκαν τυχαία για να αντιμετωπίσουν μια μεγαλύτερη καθυστέρηση (σύμφωνα με μια κανονική κατανομή), το RR-TCP θα μπορούσε να βελτιώσει την απόδοση σύνδεσης κατά περισσότερο από 50 τις εκατό και 150 τις εκατό σε σύγκριση με τους αλγόριθμους Blanton-Allman (συμπεριλαμβανομένης της χρονοκαθυστέρησης του αλγόριθμου fastretransmit) και το SACKTCP, αντίστοιχα.

Ωστόσο, το RR-TCP πρέπει να διατηρήσει ένα ιστόγραμμα αναδιάταξης για να αποθηκεύσει την αναδιάταξη των πληροφοριών. Πρέπει επίσης να σαρώσει και να ενημερώσει το ιστόγραμμα για κάθε τμήμα που αναδιατάσσεται.

**Ο αλγόριθμος των Leung-Ma:** Οι Leung και Ma [15] πρότειναν να βελτιώσουν τη δυναμικότητα του TCP απέναντι σε συνεχή packetreordering με το να επεκτείνουν τους αλγόριθμους των Blanton-Allman [14]. Πρώτον, οι Leung και Ma πρότειναν τη χρήση ενός EWMA και τον τρόπο απόκλισης της έκτασης των τμημάτων αναδιάταξης. Με τη συμπερίληψη της μέσης απόκλισης του μήκους, το dupthresh επιλέγεται για να αποφευχθεί η δημιουργία ενός ορισμένου τμήματος ψευδούς fastretransmit και να προλάβει τη δαπανηρή αναμετάδοση αφού λήξει ο χρόνος. Αυτό μοιράζεται την ίδια σχεδιαστική φιλοσοφία όπως το RR-TCP, αλλά συνεπάγεται με λιγότερες υπολογιστικές και με λιγότερη επιβάρυνση αποθηκευτικού χώρου. Δεύτερον, μπήκε ένα φράγμα του dupthresh για να αποφευχθούν τα χρονικά όρια αναμετάδοσης.

Για να αποφευχθεί η λήξη του χρόνου για ένα χαμένο τμήμα, ένα ACK πρέπει να παραληφθεί από μία πηγή για το τμήμα που αναμεταδόθηκε πριν λήξει ο χρόνος. Ο μέγιστος αριθμός των διπλών ACKs που λήφθηκαν πριν από την ενεργοποίηση ενός fastretransmit, μπορεί να υπολογιστεί ότι πληρεί τα προαναφερθέντα κριτήρια. Τρίτον, οι Leung και Ma πρότειναν επίσης ένα μηχανισμό για τη μείωση του dupthresh για τις αναμεταδόσεις με όριο λήξης, δεδομένου ότι η εμφάνιση ενός χρονομέτρου αναμετάδοσης θα μπορούσε να σημαίνει ότι το dupthresh ήταν πάρα πολύ μεγάλο.

Τα αποτελέσματα της εφαρμογής έδειξαν ότι ο αλγόριθμος των Leung και Ma θα μπορούσε να βελτιώσει την απόδοση σύνδεσης κατά τουλάχιστον 35 τις εκατό και θα μπορούσε να μειώσει τα περιττά fastretransmit κατά 6 τις εκατό σε σύγκριση με τους αλγορίθμους των Blanton και Allman (συμπεριλαμβάνοντας τον

αλγόριθμο της λήξης χρόνου fastretransmit). Όταν συγκριθεί με το RR-TCP, ο αλγόριθμος των Leung και Ma επιτυγχάνει παρόμοια απόδοση από την άποψη της απόδοσης της σύνδεσης και των περιπτώσεων fastretransmit, αλλά χρειάζονται αρκετά λιγότεροι υπολογισμοί και μικρότερος αποθηκευτικός χώρος.

**RN-TCP:** Οι Sathiseelan και Radzik ανέπτυξαν την ειδοποίηση αναδιάταξης του TCP (RN-TCP)[16], για να χρησιμοποιήσουν πληροφορίες σχετικά με τμήματα που έχουν χαθεί στους δρομολογητές, κατά μήκος των διαδρομών μετάδοσης που συνδέουν μια πηγή σε έναν προορισμό. Εάν ένας δρομολογητής έχει χάσει κάποια πακέτα δεδομένων που ανήκουν σε μια ροή, είναι καλό να θυμηθούμε τους γνωστούς μέγιστους και ελάχιστους αριθμούς ακολουθίας των χαμένων πακέτων που ανήκουν στη ροή αυτή. Αυτό σημαίνει ότι ο δρομολογητής έχασε μερικά, αν όχι όλα, τα πακέτα της ροής με αριθμούς ακολουθίας εντός αυτού του φάσματος. Όταν ένα πακέτο δεδομένων περνά μέσα από το δρομολογητή και υπάρχει απώλεια εισόδου για τη ροή του εν λόγω πακέτου, ο δρομολογητής εισάγει την απώλεια εισόδου στο πακέτο και στη συνέχεια απομακρύνεται από το δρομολογητή. Σε περίπτωση που το πακέτο περιέχει μια απώλεια καταχώρησης που έχει συλλεχθεί από ανοδικούς δρομολογητές, ο δρομολογητής αυτός θα υπολογίσει το κατάλληλο εύρος των αριθμών ακολουθίας με τα πακέτα που απορρίφθηκαν πριν βάλει ταμπέλες στα πακέτα αυτά. Ένας προορισμός στέλνει ένα ACK για την άφιξη του τμήματος. Μπορεί να οριστεί κάποια «αναδιάταξη» του ACK ανάλογα με το αν η διαφορά μεταξύ των αναδιαταγμένων τμημάτων προκαλούνται από την προώθηση αναδιάταξη διαδρομής ή όχι.

Όταν ο προορισμός λαμβάνει ένα τμήμα δεδομένων, χρησιμοποιεί τον αύξοντα αριθμό του εισερχόμενου τμήματος, την απώλεια που αποθηκεύεται στο εισερχόμενο τμήμα, και τον αύξοντα αριθμό των τελευταίων τμημάτων που έχουν ληφθεί στη σειρά μνήμης για να ελεγχθούν ποια τμήματα έχουν αναδιαταχθεί ή να

χαθεί. Αν ο προορισμός θεωρεί ότι το κενό μεταξύ των αναδιαταγμένων τμημάτων προκαλούνται από την αναδιάταξη τμημάτων, το 'αναδιαταγμένο' κομμάτι του αντίστοιχου ACK θα ρυθμιστεί. Μια πηγή χρησιμοποιεί δύο διαφορετικούς κανόνες για να προκαλέσει fastretransmit, ανάλογα με το αν το κομμάτι «αναδιάταξης» είναι ρυθμισμένο ή όχι. Εάν το τρίτο διπλό ACK έχει ορίσει ένα αναδιαταγμένο κομμάτι, η πηγή προβλέπει ότι το αναμενόμενο τμήμα έχει αναδιαταχθεί. Η πηγή θα προκαλέσει αναμετάδοση όταν λαμβάνει  $k$ , περισσότερα διπλά ACKs με το ορισμένο κομμάτι 'αναδιάταξης' ή με το διπλό ACK χωρίς το ορισμένο κομμάτι 'αναδιάταξης'. Αλλιώς, η πηγή αναμεταδίδει αμέσως το προαναφερθέν τμήμα που έχει χαθεί. Ωστόσο, το RN-TCP έχει τρία μειονεκτήματα.

Αρχικά, το RNTCP απαιτεί από όλους τους συμμετέχοντες δρομολογητές να αποθηκεύουν, να υπολογίζουν και να εισάγουν τις πληροφορίες που έχουν χαθεί σε πακέτα για προώθηση. Δεν είναι πρακτικό σε ένα ετερογενές περιβάλλον δικτύου, όπως το Διαδίκτυο, όπου οι συσκευές δικτύου διατηρούνται από διαφορετικούς οργανισμούς και γενικά αναβαθμίζονται σταδιακά. Δεύτερον, η τιμή του  $k$  εξαρτάται από το εύρος της απώλειας των πακέτων και την αναδιάταξη μέσα στα δίκτυα. Ένας μηχανισμός εξακολουθεί να χρειάζεται για να καθορίσει μια κατάλληλη τιμή του  $k$ . Τρίτον, το RN-TCP δε μπορεί να στηριχθεί στην αντοχή του για αντίστροφη πορεία αναδιάταξης. Ο αλγόριθμος που αποστέλλεται είναι ευαίσθητος στην σειρά με την οποία λαμβάνονται τα ACKs, αφού η διαφορετική σειρά των ACK που φτάνουν μπορεί να προκαλέσει διαφορετικούς κανόνες για fastretransmit, εμποδίζοντας έτσι την αποτελεσματικότητα του RN-TCP για να βελτιώσει την απόδοση του TCP για το packetreordering.

## 2.4.2 ΧΡΟΝΙΚΗ ΠΡΟΣΕΓΓΙΣΗ (TEMPORAL APPROACH)

Η χρονική προσέγγιση αντιπροσωπεύει μια ομάδα από τεχνικές που αποφεύγει προληπτικά την πρόκληση ψευδών αποκρίσεων συμφόρησης με το να τις αναβάλλει για μια χρονική περίοδο. Οι αποκρίσεις αυτές θα απαντηθούν μόνο όταν λήξει το αντίστοιχο χρονόμετρο. Η χρονική προσέγγιση μπορεί να ταξινομηθεί περαιτέρω σε αλγορίθμους για την αναβολή ανταπόκρισης και σε αλγορίθμους για την αναμετάδοση του χρονικού ορίου. Για την αναβολή απόκρισης, ένας πελάτης TCP καθυστερεί προκαλώντας μια αντίδραση συμφόρησης για μια χρονική περίοδο. Κατά τη διάρκεια της χρονικής περιόδου, η απάντηση θα ακυρωθεί όταν θα συναχθεί εκδήλωση για την έναρξη της απάντησης και δεν πρέπει να προκαλεστεί από τη συμφόρηση. Για την αναμετάδοση με χρονικό όριο, ο πελάτης TCP δημιουργεί μια κατάλληλη απόκριση συμφόρησης μόνο όταν λήξει το όριο χρόνου.

## ΑΝΑΒΟΛΗ ΑΠΟΚΡΙΣΗΣ (RESPONSE POSTPONEMENT)

**Αλγόριθμος Lee-Park-Choi** (Λύση από τη μεριά του δέκτη): Ο Lee και οι άλλοι [12] πρότειναν μια λύση από τη μεριά του δέκτη για τη βελτίωση της απόδοσης του TCP και την προώθηση αναδιάταξης. Ένας προορισμός στέλνει τα καθυστερημένα ACKs για τις αφίξεις των reordered packets. Παράγει ένα ACK σε κάθε δύο αφίξεις τμημάτων ή όταν λήξει ο χρόνος ενός αργοπορημένου ACK. Επιπλέον, ο προορισμός στέλνει αμέσως ACKs για να αναμεταδοθεί το τμήμα, έτσι ώστε να αποφευχθεί η λήξη χρόνου αναμετάδοσης. Ωστόσο, αφού ο χρόνος μεταξύ δύο διαδοχικών αφίξεων των τμημάτων μπορεί να εξαρτάται από τη σύνδεση συμφόρησης μιας διαδρομής, δεν υπάρχει άμεσος συσχετισμός μεταξύ του εύρους ζώνης διαδρομής της συμφόρησης και των διαφορών σε διαδρομές που

καθυστερούν. Έτσι, η υποβληθείσα αναβολή απόκρισης δεν είναι προσαρμοστική ως προς την έκταση του packetreordering, εξαιτίας του επιπέδου των πακέτων πολλαπλών διαδρομών.

**Ο αλγόριθμος του Paxson:** Ο Paxson [17] περιέγραψε μια ιδέα εισαγωγής ενός επιπλέον χρόνου αναμονής, πριν ένας προορισμός δημιουργήσει ένα διπλότυπο ACK από το εκ των προτέρων κενό ακολουθίας. Η καθυστέρηση στην παραγωγή ACK παρέχει την ευκαιρία για τον προορισμό να δει αν είναι απαραίτητο να στείλει ένα αντίγραφο ACK. Ωστόσο, δεν παρέχει κάποιο μηχανισμό, για να καθοριστεί ο τρόπος με τον οποίο θα οριστεί ένα κατάλληλο χρονικό διάστημα, για να προσαρμοστούν στο επίπεδο του packetreordering που πραγματοποιείται.

**Αλγόριθμος χρονικής καθυστέρησης για fastretransmit:** Οι Blanton και Allman ανέπτυξαν τον αλγόριθμο της χρονικής καθυστέρησης του fastretransmit, που συμβολίζεται ως Blanton-Allman: DEL, για να αναβάλουν την αντίδραση της συμφόρησης με την παρουσία των διαδρομών αναδιάταξης. Αφού παραλειφθούν τρία αντίγραφα ACKs, μια πηγή περιμένει για ένα επιπλέον χρονικό διάστημα πριν από την ενεργοποίηση αντίδρασης συμφόρησης. Μπορεί να θεωρηθεί ως επέκτασή της. Ωστόσο, διαφέρουν, αφού ο αλγόριθμος Paxson είναι ένας αλγόριθμος βασισμένος στο δέκτη, αλλά η χρονική καθυστέρηση του fastretransmit του αλγόριθμου είναι ένας αλγόριθμος που βασίζεται στον αποστολέα. Όταν φτάνει ένα ACK για το τμήμα απωλειών, η εκκρεμούσα αντίδραση συμφόρησης είναι ξεκάθαρη. Το χρονικό διάστημα αυξάνεται από κάποια σταθερά, κάθε φορά που ανιχνεύεται ψευδής αναμετάδοση. Έτσι, αυτή η μέθοδος είναι παρόμοια με τους αλγορίθμους που προτείνονται, για να ρυθμιστεί δυναμικά το dupthresh, και μοιράζεται τα σχετικά πλεονεκτήματα και τους περιορισμούς της.

**TCP-DCR:** Οι Bhandarkar και Reddy επινόησαν την όψιμη αντίδραση συμφόρησης του TCP (TCP-DCR), για να βελτιωθεί η ευρωστία του TCP και για



να μην υπάρχουν γεγονότα κυκλοφοριακής συμφόρησης. Βελτιώνει τον αλγόριθμο του χρόνου καθυστέρησης `fastretransmit` [18], καθυστερώντας την αντίδραση συμφόρησης για ένα χρονικό διάστημα, αφού ληφθεί το πρώτο διπλό ACK. Οι συγγραφείς πρότειναν να οριστεί αυτό το διάστημα σε ένα RTT, έτσι ώστε να έχουν αρκετό χρόνο για την αναδιάταξη προώθησης διαδρομής, λόγω των σε σειρά αναμεταδόσεων για την ανάκτηση της απώλειας. Για να διατηρηθεί η χρονομέτρηση του ACK, το TCPDCR στέλνει ένα νέο τμήμα δεδομένων κατά την παραλαβή κάθε διπλού ACK. Τα αποτελέσματα της προσομοίωσης έδειξαν ότι το TCP-DCR απέδωσε πολύ καλύτερα από το SACKTCP .

Το TCP-DCR κατάφερε 10 φορές περισσότερο όσον αφορά τη διοχέτευση από το SACKTCP, όταν περισσότερο από το 5% των πακέτων ήταν αργοπορημένα σύμφωνα με μια κανονική κατανομή αμελητέας κυκλοφοριακής συμφόρησης. Ωστόσο, η καθυστέρηση σύνδεσης συμφόρησης που έχει επιλεχθεί είναι τουλάχιστον ίση με την υψηλότερη δυνατή καθυστέρηση αναδιάταξης για τα πειράματά τους. Αυτό σημαίνει ότι το γεγονός της αναδιάταξης είναι απίθανο να διαρκέσει περισσότερο από το διάστημα της καθυστέρησης αντίδρασης της κυκλοφοριακής συμφόρησης. Το προτεινόμενο διάστημα μπορεί να μην είναι κατάλληλη επιλογή για τη δρομολόγηση πολλαπλών διαδρομών, αφού τα πακέτα που αναδιατάσσονται βασίζονται κυρίως στις διαφορές της καθυστέρησης της διαδρομής, ενώ το εκτιμώμενο RTT είναι ένας σταθμισμένος μέσος όρος των RTT, με βάση την κατανομή της κυκλοφορίας.

## **ΑΝΑΜΕΤΑΔΟΣΗ ΜΕ ΤΗ ΛΗΞΗ ΤΟΥ ΧΡΟΝΟΥ (RETRANSMISSIONTIMEOUT)**

**TCP-PR:** Ο Bohacek και οι άλλοι πρότειναν το TCP για το συνεχές packetreordering (TCP-PR) [19], να αναπτυχθεί το χρονόμετρο RTO, ώστε να βελτιώσει την απόδοση του TCP για το συνεχές packetreordering. Αντί να παρακολουθείται το EWMA του μέσου RTT, το TCP-PR χρησιμοποίησε ένα μη εξομαλυμένο, εκθετικά σταθμισμένο μέγιστο δυνατό RTT. Με αυτόν τον τρόπο, οι αιχμές στο RTT μπορούν να αντικατοπτριστούν άμεσα στην εκτίμηση του RTT για κάποιο χρονικό διάστημα. Όταν ανιχνεύεται απώλεια τμήματος, το cwnd έχει οριστεί στο ήμισυ του cwnd κατά τη στιγμή της αποστολής του τμήματος. Μετά διεξάγεται η αποφυγή της συμφόρησης. Οι μεταγενέστερες περιστασιακές απώλειες τμημάτων, που θα ανιχνευθούν στο ίδιο παράθυρο συμφόρησης, δεν θα προκαλέσουν καμία περαιτέρω μείωση του cwnd για να αποφευχθεί η υπερβολική αντίδραση της συμφόρησης. Όταν περισσότερο από το μισό της αξίας των τμημάτων ενός παραθύρου συμφόρησης που προκύπτει πρέπει να χαθεί, το cwnd έχει οριστεί σε ένα και γίνεται η διαδικασία αργής εκκίνησης.

Το σημαντικότερο πλεονέκτημα του TCP-PR είναι ότι οι νέες εκτιμώμενες RTT και RTO είναι πολύ αποτελεσματικές στη θωράκιση του αποτελέσματος του packetreordering, λόγω διαφορών στις καθυστερήσεις πορείας, δεδομένου ότι επινοήθηκε από το δείγμα του μέγιστου δυνατού RTT. Ένα άλλο πλεονέκτημα για το TCP-PR είναι ότι είναι σε θέση να διατηρήσει τη χρονομέτρηση του ACK με την παρουσία packetreordering. Ο λόγος είναι ότι ένα τμήμα που εκκρεμεί μπορεί να σταλεί εφόσον ο αριθμός των εκκρεμών τμημάτων για να αναγνωριστούν είναι μικρότερος από το μέγεθος του παραθύρου συμφόρησης.

Υπάρχουν δύο περιορισμοί για το TCP-PR. Πρώτον, για να διατηρηθεί μια σταθερή κλιμάκωση (η οποία είναι μικρότερη από ένα) από τις αιχμές RTT ανά RTT, ο συντελεστής κλίμακας αυξάνεται με τον παράγοντα  $\frac{1}{cwnd}$ . Ωστόσο, αυτό κάνει το TCP-PR υπολογιστικά δαπανηρό, αφού μια σειρά εκθετικών

υπολογισμών πρέπει να εκτελείται σε κάθε άφιξη του ACK. Δεύτερον, ο προτεινόμενος εκτιμητής RTT μπορεί να είναι υπερβολικά ευαίσθητος σε αιχμές στο RTT. Ξαφνικά υψηλά δείγματα RTT, που μπορούν να προκληθούν από τη δρομολόγηση κύκλων, οι οποίοι οφείλονται σε τοπολογικές αλλαγές, μπορεί να διευρύνουν σημαντικά το εκτιμώμενο RTT για κάποιο χρονικό διάστημα. Αυτό μπορεί να αναβάλει ουσιαστικά την αναμετάδοση του τομέα, αν πράγματι χάθηκε κατά τη διάρκεια αυτού του χρόνου.

## **ΚΕΦΑΛΑΙΟ 3**

### **3.1 PACKET REORDERING ΣΕ ΔΙΚΤΥΑ ΠΟΛΛΑΠΛΩΝ ΔΙΑΔΡΟΜΩΝ**

Σε ένα ιδανικό δίκτυο, τα πακέτα δεν θα μπορούσαν ποτέ να αναδιαταχθούν, για δύο λόγους. Πρώτον, η αρχική σειρά μετάδοσης κατά πάσα πιθανότητα είναι η σειρά με την οποία ο αποστολέας επιθυμεί να παραδώσει τα πακέτα στον παραλήπτη. Δεύτερον, η λήψη των πακέτων σε σειρά απλοποιεί το σχεδιασμό του δέκτη.

Η δρομολόγηση πολλαπλών διαδρομών παραμένει ως επί το πλείστον αχρησιμοποίητη, κυρίως του λόγου αυτού. Τα κέντρα δεδομένων του δικτύου είναι πολύ διαφορετικά από το Διαδίκτυο. Η τοπολογία του Διαδικτύου είναι ακανόνιστη, και υπάρχουν πολλές διαδρομές μεταξύ των ζευγαριών των τελικών προορισμών. Οι λανθάνοντες χρόνοι μετρούνται σε δεκάδες έως εκατοντάδες χιλιοστά του δευτερολέπτου. Οι δρομολογητές που χρησιμοποιούνται στο διαδίκτυο συνήθως έχουν πολύ μεγάλα, σε βάση DRAM (Dynamic random-access memory), buffers πακέτων.

Σε αντίθεση, τα κέντρα δεδομένων του δικτύου συνήθως έχουν μια κανονική τοπολογία, με πολλαπλές διαδρομές μεταξύ των ζευγαριών των τελικών προορισμών. Οι χρονικές καθυστερήσεις είναι συνήθως 100 ms. Και το κέντρο δεδομένων των switches αλλάζει συχνά, ώστε να έχει ένα ελάχιστο ποσό από πακέτα buffers στη βάση SRAM (Static random-access memory).

### **3.2 ΠΟΣΟ ΚΟΙΝΟ ΦΑΙΝΟΜΕΝΟ ΕΙΝΑΙ ΤΟ PACKET REORDERING**

Από περίπου 2.880 συνδέσεις TCP σε έναν μόνο server, οι περισσότερες εκ των οποίων μεταφέρονται με λιγότερο από 100 KB δεδομένων, ο Mogul διαπίστωσε ότι περίπου 125, ή το 4,3%, των εν λόγω συνδέσεων είχαν τουλάχιστον ένα reorderedpacket. Επιπλέον, η κατανομή ακολούθησε ένα νόμο δύναμης. Ο Mogul δεν παρέχει πληροφορίες για το βαθμό της αναδιάταξης, δηλαδή, πόσο αργά ήταν τα reorderedpackets όταν τελικά έφτασαν. Χωρίς αυτά τα δεδομένα, δεν είναι δυνατόν να προσδιοριστεί ο αριθμός των ψευδών αναμεταδόσεων.

Ο Paxson ανέλυσε 20.000 συνδέσεις του TCP μεταξύ 35 κόμβων στο Διαδίκτυο και διαπίστωσε ότι το packetreordering ήταν στην πραγματικότητα κοινό στο Internet με το 36% όλων των συνδέσεων, που έχουν τουλάχιστον ένα reorderedpacket, και 2% (0,6%) όλων των πακέτων δεδομένων (ACK) που φθάνουν αργά. Στην πραγματικότητα, μία σύνδεση είχε το 36% με reorderedpackets.

Ο Paxson διαπίστωσε ότι η αναδιάταξη δεν παρατηρήθηκε ομοιόμορφα και μερικές περιοχές παρουσίασαν ένα πολύ υψηλότερο βαθμό του packetreordering από άλλες. Επίσης διαπίστωσε ότι σε ορισμένες περιπτώσεις, οι συνδέσεις με μεγάλα επίπεδα packetreordering δεν είχαν καμία απώλεια πακέτων και ψευδείς αναμεταδόσεις λόγω του fastretransmit. Βρήκε ότι για κάθε 22 “καλές” αναμεταδόσεις, λόγω της απώλειας πακέτων, υπήρχε μια “κακή” αναμετάδοση, λόγω του ότι δεν ήταν ξεκάθαρο το packetreordering με την απώλεια πακέτων. Σταματά τον υπολογισμό του χαμένου throughput λόγω αυτών των ψευδών αναμεταδόσεων.

Οι Bennett, Partridge, και Shectman ανέλυσαν υπερφορτώσεις των ping πακέτων ICMP μεταξύ των προορισμών που περνούν από τον ίδιο δρομολογητή και ήταν σε θέση να συσχετίσουν το βαθμό αναδιάταξης με τη διαμόρφωση και το στιγμιαίο φορτίο αυτού του δρομολογητή. Διαπίστωσαν ότι πάνω από το 90% των "συνδέσεων" είχαν υποστεί το φαινόμενο του packetreordering. Με κάποια έννοια,

αυτό το πείραμα είναι ασύμφωνο με τις προηγούμενες μελέτες του TCP, αφού οι δρομολογητές μπορούν να αντιμετωπίζουν τα πακέτα ICMP με διαφορετικό τρόπο από ό, τι τα πακέτα TCP. Επιπλέον, το μικρό μέγεθος των ring πακέτων αυξάνει μόνο τις πιθανότητες αναδιάταξης λόγω του παραλληλισμού, δεδομένου ότι περισσότερα rings θα ταιριάζουν στον αγωγό από τι τα μεγαλύτερα τμήματα του TCP.

Οι Iannaccone, Jaiswal και Diot διεξήγαγαν μια πολύ μεγαλύτερη και πιο ολοκληρωμένη μελέτη πολλών εκατομμυρίων ροών του TCP, που περνά μέσα από ένα ενιαίο πυρήνα δρομολογητή. Βρήκαν ότι λιγότερο από το 5% των ροών του TCP βιώνουν το φαινόμενο του packetreordering [22] και λιγότερο από 2% των ατομικών πακέτων που βρίσκονται μέσα σε αυτές τις ροές. Αυτό διαφωνεί με προηγούμενες μελέτες. Βρήκαν επίσης ότι μόνο το 40% των reorderedpackets θα προκαλέσει fastretransmit. Μόνο το 10% των reorderedpackets καθυστέρησαν κατά 8 πακέτα ή και περισσότερο.

Οι Laor και Gendel μιμήθηκαν την αναδιάταξη σε έναν σύνδεσμο δικτύου και έδειξαν ότι οι ροές με ένα πολύ μικρό RTT μικρότερο από 1 ms είναι πιο ισχυρές στα packetreordering και έχουν υψηλότερη απόδοση με την πάροδο του χρόνου[23]. Αυτό οφείλεται πιθανότατα στο γεγονός ότι με ένα μικρό RTT, το παράθυρο συμφόρησης του αποστολέα δε γίνεται ποτέ μεγάλο, και έτσι παίρνει ένα πολύ μικρότερο χρονικό διάστημα για να ανακτηθεί από μια μείωση στο παράθυρο συμφόρησης. Αυτά εξυπηρετούν τα δίκτυα του κέντρου δεδομένων που έχουν ένα μικρό εύρος ζώνης καθυστέρησης.

Ο Mogul πρότεινε ότι το packetreordering μπορεί να προκληθεί από «πολλαπλές διαδρομές μέσω του Διαδικτύου»[20] αλλά δεν το επεξεργάζεται λεπτομερώς. Ο Paxson ισχυρίζεται ότι οι γρήγορες ταλαντώσεις των διαδρομών μπορεί να οδηγήσουν σε packetreordering και παρέχει στοιχεία που αποδεικνύουν ότι αυτά τα δυο συνδέονται στενά[21]. Ο Paxson παρέχει επίσης στοιχεία ότι οι

δρομολογητές μπορεί μερικές φορές να "παγώσουν" κατά την επεξεργασία μια ενημέρωσης δρομολόγησης, η οποία μπορεί να οδηγήσει σε packetreordering. Για τους λόγους αυτούς, ο Paxson κατηγοριοποιεί το packetreordering ως παθολογικό.

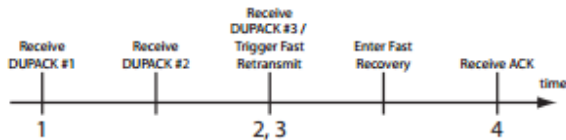
Οι Bennett, Partridge, και Shectman[24] υποστηρίζουν ότι το packetreordering είναι στην πραγματικότητα εντελώς φυσιολογική συμπεριφορά και μπορεί να εξηγηθεί απλά ως επιπτώσεις παραλληλισμού μέσα στους δρομολογητές καθώς και ως διαγράμμιση σε όλες τις παράλληλες εξωτερικές συνδέσεις, όπου και τα δυο χρησιμοποιούνται συνήθως για την κλιμάκωση του εύρους ζώνης. Επιπλέον, η αναδιάταξη αυξάνεται καθώς αυξάνεται το φορτίο.

### **3.3 ΠΡΟΤΕΙΝΟΜΕΝΕΣ ΛΥΣΕΙΣ**

Πολλές λύσεις έχουν προταθεί για τη βελτίωση της απόδοσης του TCP υπό υψηλούς βαθμούς packetreordering. Η λογική για τον διαχωρισμό των λύσεων από την αρχική τους υλοποίηση είναι ότι στις περισσότερες περιπτώσεις, πολλές λύσεις μπορούν να συνδυαστούν σε ένα ενιαίο σύστημα.

Υπάρχουν περίπου τέσσερις διαφορετικοί τύποι λύσεων που περιγράφονται παρακάτω. Η πρώτη λύση είναι να λυθεί το πρόβλημα του packetreordering σε ένα κατώτερο επίπεδο στη στοίβα πρωτοκόλλου, ώστε το TCP να μη λαμβάνει καθόλου DUPACKs. Η δεύτερη λύση είναι να προσαρμοστεί δυναμικά το dupthresh, στον αριθμό των DUPACKs, που απαιτείται για την ενεργοποίηση του fastretransmit. Η τρίτη λύση είναι να εισαχθεί fastretransmit, καθυστερώντας την έναρξη του fastrecovery. Η τέταρτη λύση είναι να ανιχνευτεί όταν μια αναμετάδοση είναι ψευδής και να αποκατασταθεί το αρχικό cwnd και ssthresh. Αυτό συνεπάγεται ότι είναι σε θέση να διαφοροποιηθεί μεταξύ των καλών

αναμεταδόσεων και των ψευδών. Οι λύσεις αυτές απεικονίζονται σε ένα χρονοδιάγραμμα στο σχήμα 3.1.



Σχήμα 3.1 Χρονοδιάγραμμα για τις λύσεις

### 3.3.1 ΛΥΣΗ 1: ΛΥΝΟΝΤΑΣ ΤΟ ΠΡΟΒΛΗΜΑ ΣΕ ΕΝΑ ΚΑΤΩΤΕΡΟ ΕΠΙΠΕΔΟ

Για λόγους ανάπτυξης, μπορεί να μην είναι δυνατή η τροποποίηση του TCP, έτσι οι λύσεις σε κατώτερα επίπεδα γίνονται ελκυστικές.

- **Αναδιάταξηbuffer (Reorder buffer):** Τα περισσότερα switches και δρομολογητές έχουν σχεδιαστεί για να διατηρήσουν την τάξη των πακέτων, αν και προσθέτει κόστος και πολυπλοκότητα. Παρά το γεγονός αυτό, τα πακέτα εξακολουθούν να φτάνουν στους τελικούς παραλήπτες αναδιαταγμένα.

Ένας τρόπος για τη βελτίωση της απόδοσης του TCP, χωρίς να τροποποιείται το TCP, είναι να αλλαχθεί η σειρά των πακέτων πριν τα λάβει το TCP, χρησιμοποιώντας μια ξεχωριστή αναδιάταξηbuffer. Αυτό θα αποτρέψει τα reordered packets να φτάσουν στο δέκτη, αποτρέποντας έτσι τα DUPACKs να φτάσουν στον αποστολέα και να προκαλέσουν ψευδή αναμετάδοση. Θα αποτρέψει επίσης τα αναδιαταγμένα ACKs να φτάσουν στον αποστολέα.

Η αναδιάταξηbuffer θα μπορούσε να εφαρμοστεί στο λογισμικό ως ένα επίπεδο με απόσταση 3.5 μεταξύ IP και TCP. Επίσης θα μπορούσε να εφαρμοστεί στο



λογισμικό NIC ως ένα νέο επίπεδο 2.5, παρόμοιο με το μηχανισμό μείωσης φορτίου του TCP.

- **Κωδικοποίηση του Δικτύου (Networking code):** Ένας άλλος τρόπος για να προστατευθεί το TCP από την αναδιάταξη είναι να χρησιμοποιηθεί κωδικοποίηση δικτύου. Η κωδικοποίηση δικτύου μπορεί να οριστεί ως «κωδικοποίηση πάνω από το φυσικό επίπεδο σε ένα δίκτυο πακέτων»[25]. Γενικά, η κωδικοποίηση δικτύου επιτρέπει σε κάθε κόμβο ή switch σε ένα δίκτυο να εξαγάγει τις πληροφορίες από ένα πακέτο, να το ανακατεύσει με πληροφορίες από μια τοπική μνήμη των πακέτων, και στη συνέχεια να δημιουργήσει ένα άλλο νέο κωδικοποιημένο πακέτο, ώστε να μεταδοθεί στο επόμενο hop του δικτύου. Το κωδικοποιημένο πακέτο περιέχει πληροφορίες από το σύνολο των μικτών πακέτων. Ο δέκτης αυτών των κωδικοποιημένων πακέτων μπορεί να τα διαβιβάσει, όπως είναι, ή να τα αποκωδικοποιήσει όταν φτάσουν αρκετά κωδικοποιημένα πακέτα.

Το δίκτυο κωδικοποίησης αφαιρεί αποτελεσματικά την έννοια της διάταξης από μια σειρά τμημάτων του TCP. Αν δεν υπάρχει διάταξη, τότε δεν μπορεί να υπάρξει αναδιάταξη. Αν και η κωδικοποίηση του δικτύου έχει δείξει ελπιδοφόρα αποτελέσματα για ασύρματα δίκτυα, μπορεί να μην είναι η καλύτερη λύση για ένα ενσύρματο δίκτυο. Υπάρχουν σημαντικά ανταλλάγματα που πρέπει να εξεταστούν. Η καθυστέρηση της επικοινωνίας αυξάνεται, επειδή ο κόμβος αποκωδικοποίησης πρέπει να συλλέξει αρκετά πακέτα, πριν ανακτηθούν τα αρχικά πακέτα. Επίσης, απαιτούνται μεγαλύτερες ρυθμίσεις στον αποστολέα για να κρατήσει τα πρωτότυπα πακέτα που έχουν ανακατευτεί, και στο δέκτη για να κρατήσει τα μικτά πακέτα που πρέπει να αποκωδικοποιηθούν. Ωστόσο, η κωδικοποίηση δικτύου παρέχει πολλά οφέλη.

Σε ορισμένες τοπολογίες και πρότυπα επικοινωνίας, η κωδικοποίηση δικτύου μπορεί πραγματικά να αυξήσει την απόδοση μειώνοντας το συνολικό αριθμό των απαιτούμενων hops-by-hops αναμεταδόσεων. Η κωδικοποίηση του δικτύου, και συγκεκριμένα η διαγραφή κωδικοποίησης, μπορεί να καλύψει την απώλεια πακέτων, με την προσθήκη περιττών πληροφοριών στα πακέτα. Η αρχική σειρά των πακέτων μπορεί να αποκωδικοποιηθεί ακόμη και αν λαμβάνεται ένα υποσύνολο κωδικοποιημένων πακέτων. Η κωδικοποίηση του δικτύου αυξάνει ελαφρώς την ασφάλεια, διότι ένας ωτακουστής πρέπει να συλλέξει αρκετά κωδικοποιημένα πακέτα, πριν μπορέσει να κωδικοποιήσει οποιοδήποτε αρχικό πακέτο.

### **3.3.2 ΛΥΣΗ 2: ΔΥΝΑΜΙΚΗ ΠΡΟΣΑΡΜΟΓΗ ΤΟΥ DUPTHRESH**

Το 1988, σε ενσύρματη πορεία Διαδικτύου 56 Kb/s με τρεις δρομολογητές, ο Jacobson διαπίστωσε ότι τρία πανομοιότυπα DUPACKs ήταν ένας καλός εκτιμητής για την απώλεια πακέτων. Ο αριθμός "3" ήταν ενσωματωμένος σε κάθε εφαρμογή του TCP για την επόμενη δεκαετία, με το ερώτημα γιατί αυτός ο αριθμός ήταν ο καλύτερος.

Ακριβώς όπως το RTO του TCP (λήξη χρονικού ορίου αναμετάδοσης) προσαρμόζεται δυναμικά ανάλογα με το εκτιμώμενο RTT, θα μπορούσε κανείς να φανταστεί ότι η προσαρμογή του dupthresh εξαρτάται από τον μετρούμενο βαθμό αναδιάταξης του δικτύου. Αυτό θα μπορούσε ενδεχομένως να μειώσει τον αριθμό των ψευδών αναμεταδόσεων, διότι θα μπορούσε να διαρκέσει περισσότερο από τρία DUPACKs και να προκαλέσει fastretransmit. Δυστυχώς, αυτό θα καθυστερήσει επίσης ένα fastretransmit όταν συμβαίνει απώλεια πακέτων, οπότε

θα πρέπει να ληφθεί μέριμνα, ώστε να μην είναι πάρα πολύ ανεκτικοί στο packetreordering.

### **3.3.3 ΛΥΣΗ 3: ΚΑΘΥΣΤΕΡΩΝΤΑΣ ΤΗΝ ΕΜΦΑΝΙΣΗ ΤΟΥ ΕΛΕΓΧΟΥ ΣΥΜΦΟΡΗΣΗΣ**

Μια άλλη δυνατότητα για τον αποστολέα είναι να αποφύγει το fastrecovery αμέσως μετά το fastretransmit. Αυτό επιτρέπει στον αποστολέα να έχει περισσότερο χρόνο για να περιμένει ένα ACK για ένα καθυστερημένο τμήμα και έχει τη δυνατότητα να μην επικαλεστεί έλεγχο συμφόρησης. Το μειονέκτημα είναι ότι, κάνει το TCP να ανταποκρίνεται λιγότερο στην απώλεια πακέτων.

### **3.3.4 ΛΥΣΗ 4: ΑΝΙΧΝΕΥΣΗ ΚΑΙ ΑΝΑΚΑΜΨΗ ΑΠΟ ΨΕΥΔΕΙΣ ΑΝΑΜΕΤΑΔΟΣΕΙΣ**

Η πιο καταστροφική επίδραση μιας ψευδούς αναμετάδοσης είναι ότι ο αποστολέας μειώνει άσκοπα το παράθυρο συμφόρησής του κατά έναν παράγοντα 2. Μια απλή βελτίωση στο TCP είναι να ανιχνεύσει αυτές τις ψευδείς αναμεταδόσεις και να αναιρεθεί το παράθυρο συμφόρησης στην κατάσταση που βρισκόταν ακριβώς πριν πραγματοποιηθεί το fastretransmit.

Το επιχείρημα έχει ως εξής: αφού το packetreordering δεν είναι απώλεια πακέτων, η διαδρομή μεταξύ του αποστολέα και του δέκτη δεν είναι κορεσμένη και ως εκ τούτου, το παράθυρο συμφόρησης δεν πρέπει να μειωθεί.

Αν και η επαναφορά από μόνη της δεν θα αποτρέψει τη χαμένη απόδοση και το προστιθέμενο λάθος χρόνο από μια ψευδή αναμετάδοση, τα μεγαλύτερα οφέλη προέρχονται από την αποκατάσταση του παραθύρου συμφόρησης. Διαφορετικά,

θα χρειαστούν πολλά ACKs για να επαναφέρουν το παράθυρο στην αρχική του τιμή. Αυτό βέβαια προϋποθέτει ότι ο αποστολέας μπορεί να καθορίσει πότε μια αναμετάδοση είναι ψευδής. Το πρότυπο TCP Reno στερείται μιας τέτοιας τεχνικής, η οποία έχει αναφερθεί ως ασάφεια αναμετάδοσης[26].

- **Χρονισμός του ACK (ACK timing)** Εάν ένα ACK "επανέρχεται πάρα πολύ σύντομα" μετά από μια αναμετάδοση στη συνέχεια υπάρχει εξαιρετικά μεγάλη πιθανότητα το ACK να ανήκει στην αρχική μετάδοση και όχι την αναμετάδοση. Επομένως, συνέβη το packet reordering και όχι απώλεια πακέτων, έτσι η αναμετάδοση ήταν ψευδής. Μια πρόταση είναι να χρησιμοποιηθεί το 75% του εκτιμώμενου RTT . Δεν είναι σαφές τι θα πρέπει να είναι αυτή η τιμή. Στην πραγματικότητα, οι άλλες επιλογές είναι πιο ισχυρές τεχνικές που δεν υπόκεινται στις διακυμάνσεις του RTT.

- **Χρονοσφραγίδα (Timestamping):** Το TCP υποστηρίζει την επιλογή να τοποθετηθεί μια χρονοσφραγίδα σε ένα τμήμα δεδομένων ή σε ένα τμήμα ACK. Αν ο αποστολέας τοποθετήσει μια χρονοσφραγίδα σε ένα τμήμα αναμεταδόσεων δεδομένων, τότε ένα από τα δύο πράγματα μπορεί να συμβούν. Κατ 'αρχάς, ο αποστολέας θα μπορούσε να λάβει ένα ACK με την ίδια χρονοσφραγίδα. Αυτό θα σήμαινε ότι το αρχικό τμήμα των δεδομένων είχε χαθεί, αλλά λήφθηκε το τμήμα που αναμεταδόθηκε. Δεύτερον, ο αποστολέας θα μπορούσε να λάβει ένα ACK με μια παλαιότερη χρονοσφραγίδα. Αυτό θα σήμαινε ότι το αρχικό τμήμα δεδομένων καθυστέρησε από το δίκτυο και έφτασε εκτός σειράς. Άρα, η αναμετάδοση ήταν ψευδής και το παράθυρο συμφόρησης μπορεί με ασφάλεια να επανέλθει. Ένα πρόβλημα με τη χρήση χρονικών σφραγίδων σε κάθε τμήμα και κάθε ACK είναι ότι κάθε χρονοσφραγίδα καταναλώνει 12 επιπλέον bytes.

Σε κάθε περίπτωση, ο αποστολέας χρειάζεται να θυμάται μόνο τις χρονικές σημάνσεις που σχετίζονται με την αναμετάδοση των τμημάτων και όχι από όλους τους τομείς, αν και μερικές παραλλαγές του TCP όπως το TCPVegas στην πραγματικότητα δεν παρακολουθεί χρονικές σφραγίδες για όλα τα μη αναγνωρισμένα τμήματα.

- **DSACK:** Το DSACK [27] επιτρέπει σε ένα δέκτη να κοινοποιήσει τον αποστολέα, κάθε φορά που λαμβάνει ένα αντίγραφο τμήματος, καθώς και τον αύξοντα αριθμό του εν λόγω τμήματος. Αν ο αποστολέας παρακολουθεί ποια τμήματα έχουν αναμεταδοθεί, τότε ο αποστολέας μπορεί να προσδιορίσει πότε έχει συμβεί το packet reordering. Το DSACK είναι μια επέκταση του SACK και λόγω της δημοτικότητας των SACKs, καθώς και της αποτελεσματικότητας της επικοινωνίας του, το DSACK είναι κοινώς αποδεκτό ως η καλύτερη τεχνική για την ανίχνευση ψευδών αναμεταδόσεων.

### 3.4 ΠΑΡΑΛΛΑΓΕΣ ΤΟΥ TCP

Πολλές παραλλαγές του TCP έχουν προταθεί την τελευταία δεκαετία για να λυθεί το πρόβλημα του packetreordering στο TCP. Μερικές παραλλαγές του TCP έχουν σχεδιαστεί ειδικά για να λύσουν το πρόβλημα του packetreordering σε δίκτυα πολλαπλών διαδρομών. Άλλοι έχουν σχεδιαστεί για να λύσουν το πρόβλημα της κατάρρευσης πακέτων σε ασύρματα δίκτυα, αν και συχνά βρίσκουν λύση από τον ίδιο χώρο. Ο πίνακας 1 συνοψίζει τις παραλλαγές του TCP.

Το γεγονός ότι οι δύο διαφορετικές κοινότητες που εργάζονται στην ίδια περιοχή, καθώς και το γεγονός ότι ο Σύλλογος Μηχανημάτων Υπολογισμού (ACM Association for Computing Machinery: Είναι μια παγκόσμια κοινότητα που ειδικεύεται στην πληροφορική, και αποτελεί τη μεγαλύτερη ερευνητική και

εκπαιδευτική κοινότητα πληροφορικής.) και το Ινστιτούτο Ηλεκτρολόγων και Ηλεκτρονικών Μηχανικών (IEEE Institute of Electrical and Electronics Engineers: Είναι ένας από τους μεγαλύτερους οργανισμούς με επαγγελματίες τεχνολογίας, που ειδικεύεται στην εξέλιξη της ηλεκτρικής και ηλεκτρολογικής μηχανικής, των τηλεπικοινωνιών, της επιστήμης των υπολογιστών και των ενοποιημένων προτύπων) συχνά δεν συνεργάζονται όπως θα έπρεπε, έχει οδηγήσει σε επανάληψη των εργασιών και εκ νέου ανακάλυψη αυτών των βασικών λύσεων. Επομένως, δεν πρέπει να εντυπωσιαζει το γεγονός αν μία παραλλαγή του TCP που προτείνεται αργότερα, είναι στην πραγματικότητα λιγότερο εξελιγμένη από μία προτεινόμενη που έχει προταθεί νωρίτερα.

Συγκρίνονται μόνο τα χαρακτηριστικά υψηλού επιπέδου αυτών των παραλλαγών του TCP. Μια σύγκριση επιδόσεων καθορίζει τα σχετικά πλεονεκτήματα και μειονεκτήματα των διαφορετικών λύσεων που απαιτούν πειραματισμό.

Name	Year	Extends	S1	S2	S3	S4
TCP Eifel	2000	TCP Reno				✓
TCP-LPC	2002	TCP Reno		✓	✓	
TCP Westwood	2002	TCP Reno				✓
TCP-BA	2002	TCP Eifel, SACK		✓	✓	✓
RR-TCP	2003	TCP-BA		✓		✓
TCP-PR	2003	TCP Reno				✓
TCP-DCR	2004	TCP SACK			✓	
TCP-NCR	2006	TCP-BA, RR-TCP, TCP-DCR		✓		
TCP/NC	2009	TCP Vegas	✓			

Πίνακας 1: TCP variants

- **TCP Eifel**

Το TCP Eifel [26] σχεδιάστηκε για να βελτιώσει την απόδοση του TCP σε όλες τις ασύρματες συνδέσεις, όπου επαναφέρει την κατάσταση ελέγχου της συμφόρησης με τέτοιο τρόπο, ώστε οι εκρήξεις πακέτων να αποφεύγονται, προκαλώντας στον αποστολέα να συνεχίσει να μειώνει το παράθυρο συμφόρησής του. Το TCP Eifel χρησιμοποιεί τη λύση 4: τον εντοπισμό και την ανάκτηση από

ψευδείς αναμεταδόσεις και ψευδή χρονικά όρια. Για την ανίχνευση, το TCPEifel χρησιμοποιεί χρονοσφραγίδες, που είναι ενσωματωμένες σε κάθε τμήμα δεδομένων και κάθε τμήμα του ACK. Όταν έχει βρεθεί ψευδής αναμετάδοση ή ψευδής λήξη χρονικού ορίου, το TCPEifel αποκαθιστά το cwnd και το ssthresh στις προηγούμενες τιμές τους.

Η λύση αυτή μπορεί να προκαλέσει στον αποστολέα να μεταδώσει μια “έκρηξη” (burst) πακέτων μετά την αποκατάσταση του παραθύρου συμφόρησης, η οποία είναι ανεπιθύμητη. Συνιστάται να χρησιμοποιηθεί ένας ξεχωριστός «ρυθμός έκρηξης» για την εξισορρόπησή του[5]. Είναι γνωστό ότι η αναδιάταξη των ACKs μπορεί επίσης να οδηγήσει σε υπερφορτώσεις οπότε η πρόταση αυτή θα πρέπει να εξεταστεί.

Οι βελτιώσεις του TCPEifel χρησιμοποιούνται μόνο όταν συμβαίνει το packet reordering στο δίκτυο σε αρκετά μεγάλο βαθμό, για να προκαλέσει ψευδείς αναμεταδόσεις. Ένα πείραμα που θα ήταν χρήσιμο είναι να μεταβληθεί η συχνότητα και οι βαθμοί αναδιάταξης σε μια ροή του TCP, ενώ σχεδιάζει τη μέση απόδοση. Το TCPEifel υλοποιήθηκε ως τροποποίηση της υφιστάμενης στοίβας του TCP. Για να εκτελεστούν τα πειράματα αυτά, ένα νέο πρωτόκολλο Layer 2,5 εφαρμόστηκε μεταξύ του επιπέδου δικτύου (IP) και του επιπέδου σύνδεσης (PPP). Αυτό το επίπεδο ονομάστηκε "hiccup" και χρησιμοποιήθηκε τεχνητά για την καθυστέρηση ορισμένων πακέτων, προκαλώντας έτσι την αναδιάταξη. Τα πειράματα διεξήχθησαν με την άμεση σύνδεση δύο μηχανών με τη σειριακή γραμμή ενός 9.6 Kb/s.

- **TCP-LPC** (Transmission Control Protocol –Local Procedure Call )

Το TCP-LPC υποβλήθηκε 13 μήνες μετά τη δημοσίευση του TCPEifel. Το TCP-LPC χρησιμοποιεί τη λύση 2 για τον αποστολέα και τη λύση 3 για το δέκτη. Για τον αποστολέα, όρισαν στατικά το dupthresh, με βάση τον αριθμό των

πιθανών διαδρομών μεταξύ του αποστολέα και του δέκτη. Ισχυρίζονται ότι η εύρεσή τους υποστηρίζεται από προσομοίωση. Ο δέκτης περιμένει ένα μικρό χρονικό διάστημα πριν μεταδώσει ένα DUPACK. Αυτό είναι ως επί το πλείστον ισοδύναμο με τον αποστολέα που είναι σε αναμονή πριν από την ενεργοποίηση του fastretransmit, με εξαίρεση ότι δε θα εφαρμοσθεί περιορισμένη μετάδοση.

- **TCP Westwood**

Το TCPWestwood σχεδιάστηκε για ασύρματα δίκτυα. Το TCPWestwood είναι μοναδικό μεταξύ των παραλλαγών του TCP που μελετήθηκαν εδώ. Αντί να αποφεύγονται ψευδείς αναμεταδόσεις, ή η ανάκτηση από αυτές, απλά τις αγνοεί. Αντί αυτού, ο αποστολέας του TCPWestwood υπολογίζει το πραγματικό εύρος ζώνης από άκρο σε άκρο με δειγματοληψία των ACKs. Εάν αυτότο στιγμιαίο εύρος ζώνηςACK είναι μεγάλο, τότε το TCPWestwood υποθέτει ότι το fastretransmit δεν υποδεικνύει συμφόρηση, και έτσι μειώνει το παράθυρο συμφόρησης κατά ένα μικρό ποσοστό. Η λήξη του χρονικού ορίου θα έδειχνε σοβαρά επίπεδα συμφόρησης, αρκετά για να σταματήσει τη ροή των ACKs. Το TCPWestwood, στη συνέχεια, θα μειώσει δραστικά το παράθυρο συμφόρησης. Οι προσομοιώσεις του TCPWestwood δείχνουν ότι αποδίδει καλά σε ενσύρματα, πολλαπλών διαδρομών, δίκτυα.

Ωστόσο, αυτές οι προσομοιώσεις δείχνουν επίσης ότι το TCPReno και TCPSACK λειτουργούν εξίσου καλά, όταν το RTT είναι λιγότερο από 20 ms και όταν χρησιμοποιούνται μόνο 2 ή 3 διαδρομές. Για 4 διαδρομές, το TCPWestwood εκτελεί μια διάταξη καλύτερου μεγέθους.

- **TCP-BA(Bandwidth Aware)**

Το TCP-BA βελτιώνει άμεσα το TCPEifel διατηρώντας τη λύση4, αλλά προσθέτοντας στη λύση 2, την ικανότητα να αυξάνεται δυναμικά το dupthresh,



ανάλογα με το βαθμό του packetreordering στο δίκτυο. Αξιολογούνται τρεις τεχνικές για την αύξηση του dupthresh μετά από μια παραπλανητική αναμετάδοση: σταθερή αύξηση, αύξηση σε σχέση με το μήκος του συμβάντος αναδιάταξης και ένα εκθετικά σταθμισμένο κινητό μέσο όρο του μήκους του γεγονότος αναδιάταξης (exponentiallyweightedmovingaveragechart- EWMA). Τα δύο πρώτα έχουν καλές επιδόσεις, αλλά όχι το EWMA.

Αξιολογείται επίσης η λύση 3: αναμένεται ένα σύντομο χρονικό διάστημα μετά τη λήψη του τρίτου DUPACK πριν προκληθεί ένα fastretransmit. Σε πειράματα που έχουν γίνει δείχνουν ότι η λύση 3 αποδίδει καλύτερα από άποψη απόδοσης και περιττών αναμεταδόσεων. Δυστυχώς, η λύση 3 απαιτεί ένα επιπλέον χρονόμετρο. Μια μικρή διαφορά μεταξύ του TCP-BA και του TCPEifel είναι ότι το TCP-BA χρησιμοποιεί DSACK παρά TCP χρονοσφραγίδες για την ανίχνευση ψευδών αναμεταδόσεων.

Για να λυθεί το πρόβλημα υπερφόρτωσης, χρησιμοποιείται περιορισμένη μετάδοση, η οποία επιτρέπει στον αποστολέα να μεταδίδει νέα τμήματα του TCP, όποτε τα DUPACKs φτάνουν σε κατάσταση fast retransmit. Κανονικά, η περιορισμένη μετάδοση επιτρέπει μόνο την αποστολή των νέων δεδομένων για τα δύο πρώτα DUPACKs. Η επέκτασή τους επιτρέπει στο TCP-BA να στείλει ένα νέο τμήμα του TCP, όταν φτάνουν δύο επιπλέον DUPACKs. Σε αντίθεση με το TCPEifel, αυτά τα πειράματα διεξήχθησαν εξ ολοκλήρου σε προσομοίωση με τη χρήση του προσομοιωτή ns-2. Επεκτείνουν τον προσομοιωτή να εισάγει έναν δρομολογητή που αναδιατάσσει τα πακέτα εσωτερικά.

Ωστόσο, το μοντέλο αναδιάταξης είναι πολύ απλό και δεν προσπαθεί να διαμορφώσει έναν πραγματικό δρομολογητή. Αντί αυτού, περιμένει μέχρι να σχηματιστεί μια σειρά από πακέτα και τυχαία να ανταλλαχθούν δύο πακέτα στη σειρά. Με την προσομοίωση, φαίνεται ότι ένα γεγονός packet reordering ανά δευτερόλεπτο είναι αρκετό για να μειώσει την απόδοση κατά 7,4%. Καθορίστηκε

ότι οι λύσεις τους 2 και 3 βελτίωσαν μόνο την απόδοση κατά περίπου 1%, και συμπεραίνεται, ότι “η επίπτωση κάποιου συστήματος επανόρθωσης έχει ελάχιστη απόδοση και ότι η αύξηση της απόδοσης προέρχεται από τη λύση 4, όταν μια αναμετάδοση είναι καθορισμένη να είναι ψευδής”.

- **RR-TCP (ReorderingRobust)**

Το RR-TCP επεκτείνει το TCP-BA, προσθέτοντας τη δυνατότητα να μειώσει το dupthresh καθώς και να το αυξήσει. Αυτό είναι σημαντικό, δεδομένου ότι το TCP-BA θα συνεχίσει να αυξάνει το dupthresh, μέχρι που τελικά θα υπάρχουν αρκετά DUPACKs να φτάσουν και θα γίνει λήξη του χρονικού ορίου, στο σημείο όπου το TCP-BA θα επαναφέρει dupthresh στο 3. Το RR-TCP δεν υιοθέτησε τη λύση 3.

Το RR-TCP εκτείνει επίσης την περιορισμένη μετάδοση, για να στείλει μέχρι ένα ολόκληρο παράθυρο συμφόρησης δεδομένων, όταν είναι στην κατάσταση fastretransmit. Αυτό είναι πιο επιθετικό από το TCP-BA, το οποίο μεταδίδει μόνο ένα νέο τμήμα του TCP για κάθε δύο DUPACKs που θα ληφθούν.

- **TCP-PR (Persistent Packet Reordering) (Συνεχής Αναδιάταξη)**

Το TCP-PR είναι παρόμοιο με το TCPWestwood στο γεγονός ότι αγνοεί τα DUPACKs που θα προκαλέσουν fastretransmit. Αντίθετα, το TCP-PR σφραγίζει χρονικά κάθε τμήμα, και τα ACKs που δεν φτάνουν πριν λήξει το χρονικό όριο, αναμεταδίδονται. Αυτό είναι ένα πολύ απλό σχέδιο, αλλά θα μπορούσε να υποφέρει από προβλήματα υλοποίησης, ιδιαίτερα στις συνδέσεις υψηλής ταχύτητας.

- **TCP-DCR (Delayed Congestion Response) (Καθυστέρηση Αντίδρασης Συμφόρησης)**

Το TCP-DCR έχει σχεδιαστεί για τα ασύρματα δίκτυα. Το TCP-DCR εκτείνει το TCPSACK και εφαρμόζει τη λύση 3, καθυστερεί την έναρξη του ελέγχου συμφόρησης μετά τη λήψη τριών DUPACKs. Η καθυστέρηση έχει ρυθμιστεί στο RTT, η οποία είναι ακριβώς κάτω από την τιμή του RTO.

- **TCP-NCR (Non Congestion Response )** (Χωρίς Αντίδραση Συμφόρησης)

Το TCP-NCR είναι μια σύνθεση του TCP-BA, του RR-TCP και του TCP-DCR. Ουσιαστικά, το TCP-NCR εφαρμόζει μόνο τη λύση 2, δυναμικά μεταβαλλόμενο dupthresh. Ωστόσο, το dupthresh έχει οριστεί να είναι ίσο με την τιμή των δεδομένων ενός παράθυρου συμφόρησης, δηλαδή ένα RTT. Αξίζει να σημειωθεί ότι λείπει η λύση 4. Οι μελετητές προτείνουν ότι το TCP-NCR μπορεί να συνδυαστεί με οποιοδήποτε άλλο TCP που διαθέτει τη λύση 4.

Το TCP-NCR προσφέρει δύο τρόπους περιορισμένης μετάδοσης. Η προσεκτική περιορισμένη μετάδοση είναι ουσιαστικά η ίδια συμπεριφορά όπως το TCP-BA, όπου ένα νέο τμήμα δεδομένων αποστέλλεται για κάθε δύο DUPACKs. Η επιθετική περιορισμένη μετάδοση είναι ίδια με το RR-TCP, όπου ένα νέο τμήμα δεδομένων στέλνεται για κάθε DUPACK.

Το RFC4653 δεν παρουσιάζει τα αποτελέσματα της προσομοίωσης, μόνο μια περιγραφή και μια διασάφηση.

- **TCP/NC (Network Coding)** (Κωδικοποίηση Δικτύου)

Το TCP/NC σχεδιάστηκε για να ξεπεράσει απώλειες συνδέσεων σε ασύρματα δίκτυα. Θα μπορούσε επίσης να είναι κατάλληλο για την επίλυση του προβλήματος του packetreordering σε ενσύρματα δίκτυα πολλαπλών διαδρομών. Το TCP / NC μπορεί να χρησιμοποιηθεί με οποιαδήποτε υλοποίηση του TCP.

Χρησιμοποιεί το TCPVegas ως βασική έκδοση, διότι καταγράφει το χρόνο που κάθε πακέτο στέλνεται και υπολογίζει μια εκτίμηση του RTT για κάθε μετάδοση.

Αυτό επιτυγχάνεται με την παρακολούθηση του χρόνου κάθε επιβεβαίωσης ACK προς τον αποστολέα. Ωστόσο, το TCP/NC μπορεί να χρειαστεί να τροποποιηθεί σε μεγάλο βαθμό, για να λειτουργήσει καλά για ενσύρματα δίκτυα, δεδομένου ότι το TCP/NC μεταδίδει επιπλέον πακέτα, για να καλύψει την απώλεια πακέτων, τα οποία θα μπορούσαν να οδηγήσουν σε συμφόρηση.

### 3.5 ΚΡΙΤΙΚΗ ΑΝΑΛΥΣΗ

Χωρίς ρεαλιστικές μετρήσεις πραγματικών ή προσομοιωμένων κέντρων δεδομένων δικτύου, είναι δύσκολο να πραγματοποιηθεί μια ποσοτική ανάλυση.

Πρώτον, δεδομένου του μήκους των καλωδίων Ethernet στα κέντρα δεδομένων, τη καθυστέρηση μέσω ατομικών switches πακέτων και τον αριθμό αναπηδήσεων των switches μεταξύ των τελικών προορισμών, ένα τυπικό κέντρο δεδομένων RTT είναι 100  $\mu$ s.

Δεύτερον, το Ethernet ορίζει ένα μέγιστο μέγεθος ωφέλιμου φορτίου των 1500 *bytes*<sup>2</sup>. Αυτό το μέγεθος ωφέλιμου φορτίου αντιστοιχεί σε 1538B "στο ενσύρματο", το οποίο κάνει 12.304  $\mu$ s για τη μετάδοση. Ως εκ τούτου, το μέγεθος του αγωγού (μέγιστο μέγεθος παραθύρου) είναι 8 πακέτα.

Πχ. Η Alice μεταδίδει στον Bob και έχει φτάσει σε μια σταθερή κατάσταση των 8 πακέτων ανά παράθυρο. Κατά τη διάρκεια των 7 RTTs, Η Alice θα στείλει  $8 * 7 = 56$  πακέτα στον Bob, με την απουσία πλαισίου συμφόρησης, κατάρρευσης και packetreordering. Τώρα σκεφτείτε τι θα συνέβαινε εάν κατά τη διάρκεια του πρώτου RTT, ένα από τα πακέτα της Alice καθυστερούσε με ακριβώς τρία πλαίσια. Αυτό θα μπορούσε να προκαλέσει στον Bob να δημιουργήσει τρία DUPACKs, τα οποία θα μπορούσαν να προκαλέσουν την Alice να εισέλθει σε fastrecovery. Κατά τη διάρκεια του δεύτερου RTT, ο μηχανισμός fastretransmit της Alice θα (ψευδώς) αναμετάδιδε το αργοπορημένο πλαίσιο. Αυτό θα μείωνε τον

αριθμό των καλών πακέτων από μια ιδανική των  $16(8*2)$  έως  $15(8+7)$ . Κατά τη διάρκεια του τρίτου RTT, η Alice έχει ήδη εξέλθει από το fastrecovery και έχει εισέλθει ξανά σε CongestionAvoidance, αλλά με το cwnd να μειώνεται κατά ένα παράγοντα 2.

Έτσι, κατά τη διάρκεια του τρίτου RTT, η Alice θα μεταδίδει μόνο 4 frames αντί για 8. Ο αριθμός αυτός θα αυξηθεί κατά 1 για κάθε RTT, μέχρι το 7ο RTT, όταν η Alice έχει αυξήσει το cwnd πίσω στο αρχικό και μέγιστο 8 frames. Ο συνολικός αριθμός των καλών frames που μεταδίδονται κατά τη διάρκεια αυτής της περιόδου θα είναι  $8+7+4+5+6+7+8=45$ , το οποίο είναι το 80% του ιδανικού αριθμού των 56 frames.

Από την παραπάνω ανάλυση, οι ψευδείς λογαριασμοί αναμετάδοσης για ένα μόνο απλό χαμένο frame, αλλά και το ανάλογο FastRecovery και το αντίστοιχο μισό του παραθύρου συμφόρησης, αντιστοιχούν σε 10 χαμένα frames. Για WANs με πολύ μεγαλύτερα RTTs, η διαφορά αυτή είναι ακόμη πιο έντονη. Από την ανάλυση αυτή, προβλέπεται από τη λύση 4 ότι, ο εντοπισμός και η ανάκτηση από ψευδείς αναμεταδόσεις, είναι πιθανό να είναι η πιο σημαντική λύση που απασχολείται σε κάθε εφαρμογή στο κέντρο δεδομένων του TCP υψηλών επιδόσεων. Οι λύσεις 2 και 3, οι οποίες τροποποιούν το dupthresh για να αποφευχθούν οι ψευδείς αναμεταδόσεις, πιθανότατα θα έχουν πολύ μικρό όφελος, δεδομένης της παρουσίας της λύσης 4.

Η λύση 1, είτε η ρύθμιση αναμετάδοσης ή η κωδικοποίηση του δικτύου, είναι απίθανο να δώσει μια ικανοποιητική λύση για το αρχικό πρόβλημα. Ο κύριος λόγος είναι η έλλειψη της οικονομίας του μηχανισμού. Οποιαδήποτε λύση 1 θα είναι να επαναλαμβάνει τη λειτουργικότητα που υπάρχει ήδη στο TCP. Στην περίπτωση της ρύθμισης αναμετάδοσης, θα επαναλαμβανόταν η ρύθμιση αναδιάταξης του TCP, και θα κρύβονταν σημαντικές λεπτομέρειες από το TCP. Θα καθυστερούσε επίσης την παράδοση πακέτων στο TCP, διογκώνοντας έτσι το

RTT και αυξάνοντας τη διακύμανση του RTT και την αντίστοιχη φόρτωση της κυκλοφορίας. Η κωδικοποίηση δικτύου είναι καλή στη θεωρία, αλλά όπως φάνηκε από την παρούσα πρόταση για την κωδικοποίηση του δικτύου με το TCP, αγνοεί εντελώς τον έλεγχο συμφόρησης. Η προσπάθεια να παράγουν ένα κωδικοποιημένο δίκτυο TCP με τον έλεγχο συμφόρησης είναι κατά πάσα πιθανότητα πολύ περισσότερη από την απλή εφαρμογή της λύσης 4, χρησιμοποιώντας το DSACK.

### 3.5.1 ΧΡΟΝΙΚΑ ΌΡΙΑ

Για τα δίκτυα ενός κέντρου δεδομένων (datacenternetworks), θα πρέπει να αποφευχθούν μεγάλα χρονικά όρια με κάθε κόστος. Χρήζει έμφασης, ότι ένα χρονικό όριο είναι απαραίτητο, όταν είναι αδύνατη το fastretransmit, όπως επίσης όταν έχει χαθεί ολόκληρο το πακέτο του παραθύρου του αποστολέα, ή η παθολογική περίπτωση, όταν το dupthresh έχει αυξηθεί τόσο πολύ που είναι αδύνατο να προκαλέσει fastretransmit. Το αρχικό ελάχιστο χρονικό όριο της περιόδου του TCP,  $RTO_{min}$ , ήταν περίπου 1 δευτερόλεπτο. Αργότερα οι υλοποιήσεις στο TCP μείωσαν τα  $RTO_{min}$  σε 500 ms, και πιο πρόσφατα σε 200 ms. Ωστόσο, ακόμη και τα 200 ms είναι πολύ μεγάλα για τα κέντρα δεδομένων. Να σημειωθεί ότι αυτή είναι μια ελάχιστη χρονική περίοδος λήξης, και το πραγματικό χρονικό όριο είναι ελαφρώς μεγαλύτερο από το εκτιμώμενο RTT, αλλά ποτέ λιγότερο από το  $RTO_{min}$ .

Υποθέτοντας ότι ένα RTT των 100 ms, τα 200 ms μετρούν για 2.000 RTTs (μέθοδος των τριών), διάστημα κατά το οποίο θα μπορούσαν να σταλούν συνολικά 16.000 πακέτα. Εάν ένας αποστολέας χάνει όλο το παράθυρο των 8 πακέτων, στη συνέχεια, από τη μεριά της σύνδεσης του, αυτό είναι ισοδύναμο με την απώλεια

16.000 πακέτων, αν και η διασύνδεση του δικτύου μπορεί να χρησιμοποιήσει αυτό το χρόνο αδράνειας για να στείλει πακέτα από άλλες συνδέσεις TCP.

Για την αποφυγή χρονικών ορίων, η λύση 2 πρέπει να χρησιμοποιηθεί πολύ προσεκτικά για να αποφευχθεί η παθολογική υπόθεση που αναφέρθηκε προηγουμένως. Η λύση 3 μπορεί να προσαρμοστεί για να προκαλέσει `fastretransmit` ακριβώς πριν από ένα συγκεκριμένο χρονικό διάστημα, το οποίο είναι ιδανικό. Αυτό υπαινίσσεται ότι η λύση 3 θα μπορούσε να είναι πιο εύκολη, ώστε να εφαρμοστεί στην πράξη και είναι πιο πιθανό να αποδώσει σε ένα ευρύ φάσμα των δικτύων και περιπτώσεων χρήσης. Επίσης, η λύση 1 πρέπει να χρησιμοποιείται με προσοχή, ώστε να μην καθυστερήσει πακέτα περισσότερο από το χρονικό όριο, γεγονός που περιπλέκει το σχεδιασμό της λύσης 1, δεδομένου ότι δεν μπορούν να έχουν πρόσβαση στα κατ'εκτίμηση RTT του TCP και στην αντίστοιχη τιμή του χρονικού ορίου. Οι λύσεις 2 και 3 θα έχουν άμεση πρόσβαση και δεν θα υποφέρουν από αυτό το πρόβλημα διασύνδεσης. Κάποιος πρέπει επίσης να αμφισβητήσει τη χρησιμότητα ενός χρονικού ορίου που είναι 2.000 φορές μεγαλύτερο από το RTT. Πολλοί έχουν πειραματιστεί με τη μείωση των `RTOmin` σε 200  $\mu$ s και έχουν αναφέρει πολύ καλή απόδοση, και έλυσαν ένα διαφορετικό αλλά σημαντικό πρόβλημα του δικτύου κέντρου δεδομένων που ονομάζεται `incast`. Το `incast` είναι μία κατάσταση κατά την οποία εμφανίζεται μεγάλη μείωση της απόδοσης, τη στιγμή που πολλοί servers επικοινωνούν με ένα μόνο παραλήπτη. Κατά την διάρκεια που διεξάγεται η επικοινωνία, ο παραλήπτης στέλνει συνεχώς αιτήματα για παραλαβή δεδομένων από πολλούς servers. Οι servers, όταν λάβουν τα αιτήματα του παραλήπτη ξεκινούν να του στέλνουν στην ίδια στιγμή μεγάλο όγκο πληροφοριών με αποτέλεσμα να μειώνεται η απόδοση του παραλήπτη στο επίπεδο της εφαρμογής, καθώς επίσης μειώνεται αρκετά το μέγεθος σε σχέση με τις δυνατότητες της σύνδεσης.

### 3.6 ΠΡΟΤΕΙΝΟΜΕΝΑ ΠΕΙΡΑΜΑΤΑ

Μία ομάδα ερευνητών έχει κατασκευάσει ένα μικρό δίκτυο κατανομής (fat-treedatacenternetnetwork), με πολλούς κλάδους του κέντρου δεδομένων, που τους επιτρέπει να διεξάγουν πειράματα του TCP και packetreordering. Η τοπολογία δικτύου περιγράφει πως διασυνδέονται τα switchesandhosts. Παρουσιάζονται σε γράφημα με συνδέσμους μεταξύ τους που τα ενώνει. Η τοπολογία είναι εξίσου σημαντική τόσο για το κόστος όσο και για την απόδοση του κέντρου δεδομένων. Η τοπολογία επηρεάζει μια σειρά από ανταλλαγές σχεδιασμού, συμπεριλαμβανομένων των επιδόσεων, το σύστημα συσκευασίας, την ποικιλομορφία του μονοπατιού, και το πλεόνασμα, το οποίο, με τη σειρά του, επηρεάζει την ανθεκτικότητα του δικτύου σε σφάλματα. Το σχήμα 3.2 δείχνει την τοπολογία του δικτύου για την πλατφόρμα δοκιμών, η οποία αποτελείται από 37 υπολογιστές σε μια ενιαία βάση. Κάθε υπολογιστής καταλαμβάνει μία μονάδα rack.

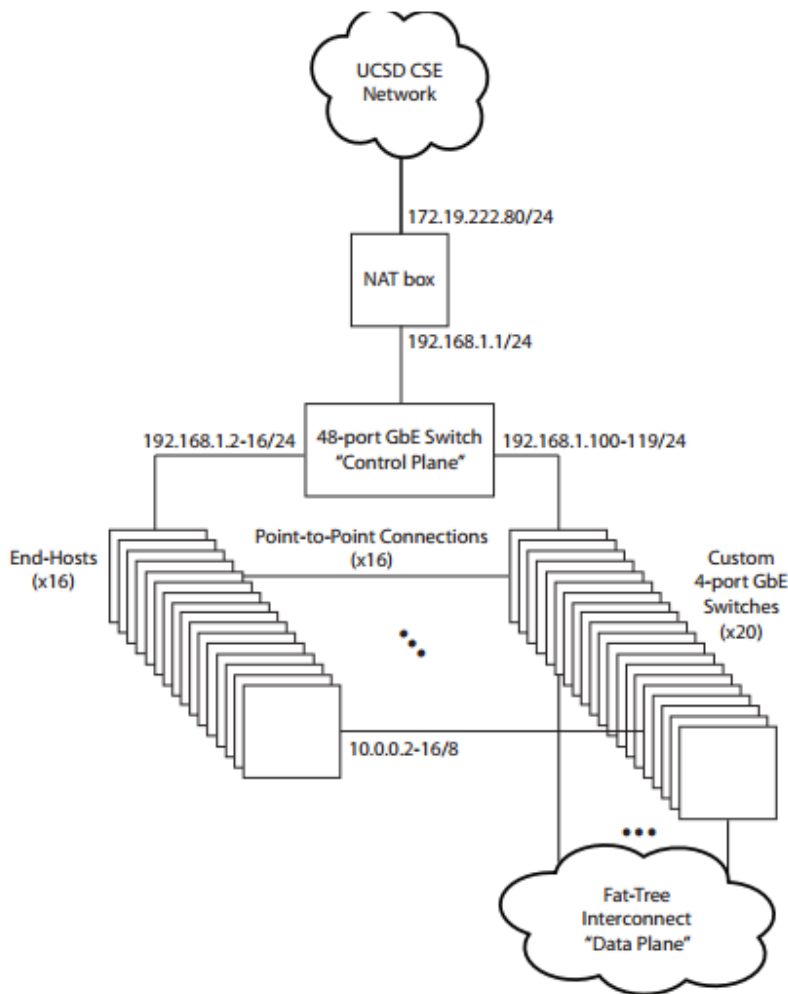
Ένας υπολογιστής, ονομάζεται κουτί NAT (NATbox), χρησιμοποιείται ως μεταφραστής διευθύνσεων του δικτύου. Το NATbox απομονώνει την πλατφόρμα δοκιμών από την ξένη κίνηση στο δίκτυο UCSDCSE, ενώ επιτρέπει επίσης στους υπόλοιπους υπολογιστές να έχουν πρόσβαση στο Internet, για να κατεβάσουν ενημερώσεις λογισμικού. Το NATbox χρησιμοποιείται για να συγκεντρώσει λογισμικό, ως αποθήκη αρχείων, να διαμορφώσει τα αρχεία, να συγκεντρώσει τα διαδικαί αρχεία και τα αποτελέσματα των δοκιμών, και να επιτρέπει την άμεση πρόσβαση του SSH στους υπόλοιπους υπολογιστές στην πλατφόρμα δοκιμών. Τέλος, το NAT box φιλοξενεί έναν Ανοικτό ελεγκτή Ροής, κατά τη διάρκεια των πραγματικών πειραμάτων.

Οι υπόλοιποι 36 υπολογιστές χωρίζονται σε 16 τελικούς προορισμούς και 20 προσαρμοσμένα switches 4 θυρών του GigabitEthernet (GbE). Οι τελικοί



προορισμοί εκτελούν τυπικές εφαρμογές του κέντρου δεδομένων, όπως MapReduce, καθώς και τα σημεία αναφοράς της απόδοσης του δικτύου, όπως το Iperf. Κάθε προσαρμοσμένο switch τρέχει πρόγραμμα Ανοιχτής Ροής που ονομάζεται secchan και συνδέεται με τον ελεγκτή Ανοιχτής Ροής στο NATbox. Οι προσαρμοσμένα switches περιέχουν επίσης τις τροποποιήσεις του switch στο secchan πρόγραμμα, καθώς και τροποποιήσεις στον εξοπλισμό στο NetFPGA [28]. Η πλατφόρμα περιλαμβάνει δύο ξεχωριστά δίκτυα, ένα επίπεδο ελέγχου και ένα επίπεδο δεδομένων. Όλοι οι 37 υπολογιστές συνδέονται με το επίπεδο ελέγχου μέσω ενός 48-θυρου GbEswitch3. Το επίπεδο ελέγχου χρησιμοποιείται για συνδέσεις ssh, για την παροχή πρόσβασης στο Internet, για ενημερώσεις λογισμικού, καθώς και για τη δημιουργία των πειραμάτων. Κατά τη διάρκεια των πειραμάτων, το επίπεδο ελέγχου χρησιμοποιείται για να επιτρέψει στους υπολογιστές να συντονιστούν με το NATbox. Οι τελικοί προορισμοί γράφουν τα αποτελέσματά τους στο NAT box και τα προσαρμοσμένα switches συντονίζονται με τον Ελεγκτή Ροής στο NATbox. Το επίπεδο δεδομένων χρησιμοποιείται μόνο κατά τη διάρκεια των πειραμάτων.

Κάθε τελικός προορισμός έχει άμεση σύνδεση με έναν από τα προσαρμοσμένα switches. Οι υπόλοιπες θύρες από τα προσαρμοσμένα switches χρησιμοποιούνται για να σχηματίσουν τη διασύνδεση δικτύου κατανομής. Αφού δίκτυο κατανομής πολλαπλών σταδίων είναι ένα switchLayer 2, μόνο οι τελικοί προορισμοί έχουν διευθύνσεις IP στο επίπεδο των δεδομένων. Οι διευθύνσεις IP είναι μέρος της ιδιωτικής περιοχής διευθύνσεων 10.0.0.0/8.



Σχήμα 3.2 Τοπολογία του δικτύου για την πλατφόρμα δοκιμών στο packetreordering

### 3.6.1 ΑΛΓΟΡΙΘΜΟΣ LOADBALANCING ΑΝΑ-ΠΑΚΕΤΟ

Η εξισορρόπηση φόρτου (loadbalancing) σε κάθε πακέτο δεν πρέπει να είναι ίδια σε κάθε πακέτο . Μόνο τα πακέτα που εισέρχονται σε ένα switch σε μια θύρα που έχει φορά προς τα κάτω, και προορίζονται για μια ανοδική-θύρα, θα πρέπει το φόρτος να είναι ισορροπημένο. Μια συνέπεια αυτού είναι ότι το loadbalancing συμβαίνει μόνο στα άκρα και σε συγκεντρωμένα switches, όχι στον πυρήνα τους.

Ένας απλός αλγόριθμος loadbalancing θα ήταν να διατηρήσει έναν ενιαίο μετρητή που αντιστοιχεί σε μια ανοδική θύρα εξόδου, για να διαβιβάσει το επόμενο πακέτο μετά από αυτό. Αυτός ο μετρητής θα κρατήσει γύρω του όλες τις ανοδικές θύρες που ήδη απαριθμούνται. Ενώ απλά και άμεσα, ο αλγόριθμος αυτός μπορεί να μην αποδώσει καλά στην πράξη. Λαμβάνοντας υπόψη το σενάριο όπου δύο διαφορετικές ροές TCP που φθάνουν σε ένα τελευταίο switch με θύρες καθοδικές, ένα στη θύρα 0 και ένα στη θύρα 1, διατηρώντας και οι δυο 1Gb/s. Το απλό loadbalancing δεν μπορεί να διανεμίει τα πακέτα από αυτές τις δύο ροές ομοιόμορφα σε όλες τις θύρες και ανοδικές και καθοδικές. Ο συγχρονισμός μπορεί να συμβεί, όταν όλα τα πακέτα που φθάνουν στη θύρα 0 προωθούνται στη θύρα 2, και όλα τα πακέτα που φθάνουν στη θύρα 1 διαβιβάζονται στη θύρα 3.

Η παρούσα παθολογική κατάσταση δεν είναι η προβλεπόμενη συμπεριφορά του loadbalancing. Ωστόσο, για άγνωστους ακόμα λόγους, αυτή η παθολογική περίπτωση δεν μπορεί να συμβεί στην πράξη, καθιστώντας τον απλό αλγόριθμο μια πολύ καλή επιλογή. Κάποιος μπορεί επίσης να φανταστεί πιο περίπλοκη πολιτική προγραμματισμού, όπως τουλάχιστον τη φορτωμένη θύρα εξόδου, τη θύρα εξόδου που χρησιμοποιήθηκε πρόσφατα, και την τυχαία ομοιομορφία της. Προτείνεται η εφαρμογή πολλών από αυτούς τους αλγορίθμους ανά πακέτο loadbalancing σε λογισμικό χρησιμοποιώντας ένα NetFPGA, και τη λειτουργία αυτών των αλγορίθμων στην πλατφόρμα δοκιμών.

Αφού το δίκτυο κατανομής με πολλούς κλάδους είναι ένα κανονικό δίκτυο, αναμένεται ότι όλοι οι αλγόριθμοι θα αποδώσουν εξίσου καλά σε ένα άνευ φορτίου δίκτυο με μία μόνο ροή TCP μεταξύ δύο τελικών προορισμών. Όλα τα πακέτα θα πρέπει να φτάσουν στη σειρά, κάνοντας την αναδιάταξη του TCP περιττή. Η πραγματική δοκιμασία θα έρθει με διασταυρωμένη κυκλοφορία και πολλαπλές ροές TCP από πολλαπλά ζεύγη τελικών προορισμών. Σε αυτή την περίπτωση, υποθέται ότι οι πιο περίπλοκοι αλγόριθμοι, και η τυχαία ομοιομορφία,

θα αποδώσουν καλύτερα στην αποφυγή συμφόρηση συνδέσεων από τους πιο απλούς αλγόριθμους. Για να δημιουργηθεί διασταύρωση, προτείνεται η χρήση πολλαπλών περιπτώσεων Iperf για διάφορα ζεύγη τελικών προορισμών, για να επιτρέψει την κλιμάκωση του πειράματος από μια ενιαία ροή έως O (162) ροές.

### **3.6.2 ΠΑΡΑΛΛΑΓΕΣ ΤΟΥ TCP**

Ο πυρήνας του Linux περιλαμβάνει τις ακόλουθες 10 παραλλαγές του TCP: Το TCP Reno, TCP BIC, TCP CUBIC, TCP HYBLA, TCP Illinois, Scalable TCP, TCP Vegas, TCP VenO, TCP Westwood, και TCP Yeah. Κάθε παραλλαγή έχει διαφορετικό στόχο και ιστορία, όπως η βελτίωση της απόδοσης σε καθυστέρηση μεγάλου εύρους ζώνης δικτύων ή ασύρματων δικτύων. Καμία από αυτές τις παραλλαγές δε σχεδιάστηκε ρητά για να βελτιώσει την απόδοση πολλαπλών διαδρομών, όπως το TCP Eifel ή RR-TCP. Ωστόσο, υπάρχουν ήδη αποτελέσματα της προσομοίωσης, όπου το TCP Westwood αποδίδει καλά σε δίκτυα πολλαπλών διαδρομών. Και αυτό μπορεί να είναι η περίπτωση που άλλα "ασύρματα TCPs", όπως το TCP VenO ή το TCP HYBLA εκτελούν επίσης καλά. Ανεξάρτητα από αυτό, θα ήταν σημαντικό να είναι γνωστό ποια TCPs αποδίδουν ιδιαίτερα καλά ή ιδιαίτερα άσχημα σε δίκτυα πολλαπλών διαδρομών.

### **3.6.3 ΑΡΓΟΠΟΡΗΜΕΝΕΣ ΑΝΑΓΝΩΡΙΣΕΙΣ**

Οι αργοπορημένες αναγνωρίσεις εισήχθησαν για να μειωθούν οι υπολογιστικές απαιτήσεις ενός δέκτη TCP. Κανονικά, ένας δέκτης TCP θα αναγνώριζε κάθε τμήμα που λήφθηκε αμέσως. Οι καθυστερημένες αναγνωρίσεις επιτρέπουν σε ένα δέκτη TCP να περιμένει έως και 200 ms πριν από την αποστολή βεβαίωσης ή την παραλαβή των δύο τμημάτων, οποιοδήποτε και από τα δύο συμβεί πρώτο. Οι καθυστερημένες αναγνωρίσεις αυξάνουν τη θεωρητική πραγματική απόδοση των

αμφίδρομων TCP ροών στο GigabitEthernet, από 897Mb/s σε 921Mbs/s. Ωστόσο, η μείωση του συνολικού αριθμού των παραδοχών που μεταδίδεται δεν μπορεί να λειτουργήσει καλά στο κέντρο δεδομένων . Υποθέτοντας ότι, αυτό θα παρέχει μια βελτίωση στην απόδοση μεταξύ 0% και 5% για όλα τα πειράματα, λόγω της ικανότητας του αποστολέα του TCP να λάβει ACKs και DUPACKs αμέσως, είναι σε θέση να ανακτήσει γρηγορότερα μετά από ένα συμβάν συμφόρησης. Το δίλημμα είναι η αυξημένη χρήση εύρους ζώνης από τα επιπλέον τμήματα ACK τα οποία θα μειώσουν την απόδοση.

### **3.6.4 RTOmin**

Έχει προταθεί ότι το RTOmin κάνει περισσότερο κακό παρά καλό στα δίκτυα του κέντρου δεδομένων και θα πρέπει να εξαλειφθεί. Ο λόγος είναι ότι τα RTOmin είναι της τάξης των 200ms, ενώ το RTT σε ένα κέντρο δεδομένων του δικτύου είναι συνήθως 100μs. Το RTOmin παρακάμπτει αποτελεσματικά την εκτίμηση του χρονοδιακόπτη του RTO για το χρόνο που πρέπει να αποκλειστεί και αναγκάζει το χρονόμετρο RTO να περιμένει πολύ περισσότερο, σε αυτή την περίπτωση 2.000 φορές περισσότερο.

Πιθανολογείται ότι η εξάλειψη των RTOmin θα κάνει τις λύσεις 1, 2 και 3 περιττές. Αν το χρονικό όριο είναι της τάξης του ενός RTT, τότε θα μπορούσε κανείς απλά να αγνοήσει τα DUPACKs, το fastretransmit, και το fastrecovery και να επιτρέψει ένα χρονικό όριο, όταν τα πακέτα καθυστερούν για υπερβολικά μεγάλο χρονικό διάστημα μιας χρονικής περιόδου. Ωστόσο, η λύση 4 μπορεί να χρησιμοποιηθεί για την ανίχνευση ψευδών αναμεταδόσεων και να αποκαταστήσει τις αρχικές cwnd και ssthresh μεταβλητές σχεδόν αμέσως μετά, πράγμα που σημαίνει ότι χάνεται πολύ μικρή απόδοση, λόγω του ότι είναι προσωρινά σε αργή εκκίνηση.

## ΚΕΦΑΛΑΙΟ 4

### 4.1 ΑΝΑΔΙΑΤΑΞΗ ΠΑΚΕΤΩΝ ΣΕ ΔΙΚΤΥΑ ΥΨΗΛΩΝ ΤΑΧΥΤΗΤΩΝ

Τα τελευταία χρόνια ένας αριθμός παραλλαγών TCP υψηλής ταχύτητας, όπως το HSTCP (HighSpeedTCP), STCP (StreamControlTransmissionProtocol) , BIC (BinaryIncreaseCongestioncontrol), και FAST (FASTAQM(ActiveQueueManagement) ScalableTCP), έχουν προταθεί για την αντιμετώπιση του προβλήματος της υπολειτουργίας του TCP σε δίκτυα υψηλής ταχύτητας και μεγάλης απόστασης.

Αυτές οι παραλλαγές του TCP υψηλής ταχύτητας τροποποιούν τους αλγόριθμους για αποφυγή της κυκλοφοριακής συμφόρησης του TCP ώστε να γίνουν πιο επιθετικοί σε δίκτυα υψηλής ταχύτητας. Ταυτόχρονα εξακολουθούν να χρησιμοποιούν τους αλγόριθμους για γρήγορη αναμετάδοση (fastretransmit) και γρήγορη ανάκτηση (fastrecovery), όπως το TCP, επιτρέπουν στο TCP να ανιχνεύσει και να ανακτήσει την απώλεια πακέτων πριν λήξει το χρονικό όριο. Ωστόσο, είναι γνωστό ότι οι αλγόριθμοι fastretransmit και fastrecovery ενδέχεται να παρερμηνεύσουν το packetreordering ως απώλεια πακέτων, και ως εκ τούτου το TCP να μην αποδίδει καλά σε δίκτυα με μεγάλο βαθμό packetreordering [29]. Επομένως, οι παραλλαγές υψηλής ταχύτητας του TCP, που χρησιμοποιούν τα ίδια fastretransmit και fastrecovery, έχει ως αποτέλεσμα οι αλγόριθμοι να μην μπορούν να επιτύχουν την αναμενόμενη υψηλή απόδοση όταν γίνεται packetreordering.

Τα αποτελέσματα των μετρήσεων στο Διαδίκτυο δείχνουν ότι σε ένα μικρό ποσοστό της κίνησης του TCP τα πακέτα δεν φτάνουν με τη σωστή σειρά, και στην πραγματικότητα, μόνο πολύ λίγα αναδιαταγμένα πακέτα προκαλούν στο TCP γρήγορη εκ νέου αποστολή και fastrecovery. Αυτό εξηγεί γιατί το TCP μπορεί να επιτύχει μια ικανοποιητική απόδοση στις περισσότερες περιπτώσεις. Ωστόσο,

αρκετές πρόσφατες μελέτες δείχνουν μια ισχυρή συσχέτιση μεταξύ της κατανομής του διαστήματος μεταξύ πακέτων (inter-packetspacing) και του packetreordering. Άρα, η μικρότερη inter-packetspacing μπορεί να αυξήσει την πιθανότητα του packetreordering. Αυτό το αποτέλεσμα σημαίνει ότι για το ίδιο μέγεθος των πακέτων, όσο υψηλότερος είναι ο ρυθμός αποστολής των πακέτων, τόσο μεγαλύτερη είναι η πιθανότητα packetreordering.

Υπάρχει κάποια συζήτηση σχετικά με το αν το packetreordering είναι μια παθολογική συμπεριφορά του Διαδικτύου, και αν αυτό το ζήτημα πρέπει να αντιμετωπιστεί σχεδιάζοντας ένα ελεύθερο δίκτυο αναδιάταξης ή σχεδιάζοντας μια αναδιάταξη ανεκτή του TCP. Η εργασία αυτή επικεντρώνεται στο φαινόμενο του χαρακτηρισμού των packetreordering σε δίκτυα υψηλής ταχύτητας, καθώς και τις επιπτώσεις του στις πρόσφατες παραλλαγές του TCP υψηλής ταχύτητας. Επιπλέον αξιολογείται η αποτελεσματικότητα των υφιστάμενων βελτιώσεων αναδιάταξης του TCP.

## **4.2 ΒΑΣΙΚΑ ΘΕΜΑΤΑ ΤΟΥ PACKETREORDERING**

Το packetreordering είναι ένα φαινόμενο στο οποίο πακέτα με υψηλότερους αριθμούς ακολουθίας λήφθηκαν νωρίτερα από εκείνα με μικρότερους αριθμούς ακολουθίας [30].

Το packetreordering στα δίκτυα μπορεί να προκληθεί από δύο βασικούς λόγους. Πρώτον, λόγω του τοπικού παραλληλισμού μέσα σε έναν δρομολογητή πακέτων, το οποίο είναι μια πολλά υποσχόμενη προσέγγιση για την κατασκευή δρομολογητών. Δεύτερον, λόγω της εξισορρόπησης φορτίου μεταξύ πολλαπλών συνδέσμων. Οι πολλαπλές συνδέσεις με ελαφρώς διαφορετικές καθυστερήσεις συνδέσμου μπορούν να επιφέρουν σημαντικό packetreordering. Το TCP επιχειρεί να διακρίνει το packetreordering από την απώλεια πακέτων, χρησιμοποιώντας τον

αριθμό των dupACKs (εις διπλούν αναγνώρισεις). Ένα ACK παράγεται από ένα δέκτη TCP για να ενημερώσει στον αποστολέα TCP τον επόμενο αριθμό ακολουθίας που αναμένεται να ληφθεί.

Στο RFC 2581 [29] προτείνεται ότι ο αποστολέας TCP πρέπει να εξετάσει τρία ή περισσότερα dupACKs ως ένδειξη ότι ένα πακέτο έχει χαθεί, με βάση την υπόθεση ότι το packetreordering μπορεί να προκαλέσει μόνο ένα ή δύο dupACKs. Ωστόσο, αν ένα packetreordering προκαλεί τρία ή περισσότερα dupACKs, τότε το TCP το παρερμηνεύει ως χαμένο πακέτο. Κατά συνέπεια, το TCP πραγματοποιεί άσκοπα τη διαδικασία fastretransmit [31], για να αναμεταδώσει το πακέτο που ερμήνευσε ως χαμένο, και πάλι άσκοπα, χρησιμοποιεί τη διαδικασία fastrecovery για να μειώσει το ρυθμό αποστολής. Άρα είναι φανερό ότι το TCP δεν αποδίδει καλά σε δίκτυα με έντονο packetreordering.

### **4.3 ΠΡΟΤΥΠΑ ΠΑΚΕΤΩΝ ΑΝΑΔΙΑΤΑΞΗΣ**

Εδώ παρουσιάζεται ένα νέο μοντέλο packetreordering, το οποίο μπορεί να χρησιμοποιηθεί για να δημιουργηθούν σχετικά ρεαλιστικές ακολουθίες reorderedpackets. Επιπλέον το μοντέλο αυτό, μπορεί να χρησιμοποιηθεί για τη μελέτη της επίδρασης μιας συγκεκριμένης ιδιότητας του packetreordering στην απόδοση του TCP.

#### **A. Περιορισμοί των υπάρχοντων μοντέλων packetreordering**

Αρκετές μέθοδοι έχουν προταθεί για την προσομοίωση του packetreordering σε προηγούμενες μελέτες. Τα μοντέλα μπορούν να ταξινομηθούν σε δύο κατηγορίες και οι δύο κατηγορίες έχουν τους δικούς τους περιορισμούς. [32]

Μία κατηγορία μοντέλων αναδιατάσσει μόνο ένα πακέτο κάθε φορά, εναλλάσσοντας δύο πακέτα μεταξύ τους σε μια ουρά δρομολογητή, ή με τη χρήση

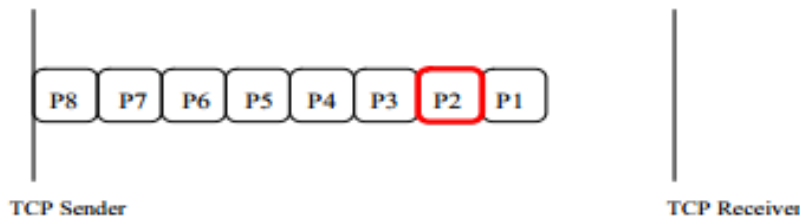


του στοιχείου hiccup. Ωστόσο, στην πράξη, μία ομάδα πακέτων, μπορεί να αναδιαταχθεί ταυτόχρονα, αντί για μόνο ένα πακέτο.

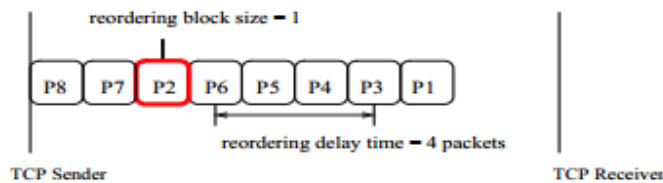
Η άλλη κατηγορία των μοντέλων δημιουργεί κάθε φορά πολλαπλά reorderedpackets, επεκτείνοντας το εσφαλμένο μοντέλο NS-2 να καθυστερήσει ένα ρυθμιζόμενο ποσοστό των πακέτων, ή αλλάζοντας την καθυστέρηση της σύνδεσης περιοδικά. Αν και αυτά τα μοντέλα μπορούν να παράγουν πιο ρεαλιστικό reorderedtraffic, είναι δύσκολο να απομονωθεί η επίπτωση μιας συγκεκριμένης ιδιότητας ενός packetreordering στην απόδοση του TCP από εκείνη των άλλων ιδιοτήτων. Το IETF (InternetEngineeringTaskForce- Τακτική Δύναμη Μηχανικών Internet) αναπτύσσει αυτή τη στιγμή μετρήσεις, για να συλλάβει τις περιστατικά και χαρακτηριστικά των packetreordering στο Διαδίκτυο. Ωστόσο, αυτές οι μετρήσεις είναι πιο κατάλληλες για την περιγραφή του packetreordering, που πραγματοποιείται σε μια ροή αντί για τη δημιουργία packetreordering.

## **B. Προτεινόμενα μοντέλα packetreordering**

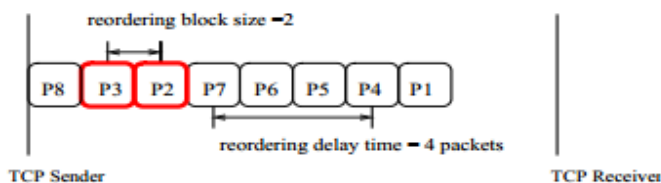
Λαμβάνοντας υπόψη τους περιορισμούς των τωρινών μοντέλων packetreordering, προτείνεται ένα νέο packetreorderinggenerator, που υλοποιείται με την επέκταση του μοντέλου σφάλματος NS-2. Το προτεινόμενο packetreorderinggenerator διαμορφώνει το φαινόμενο του packetreordering με τρεις παραμέτρους. Αυτό δίνει τη δυνατότητα να μελετηθούν οι επιπτώσεις της κάθε μίας από τις τρεις παραμέτρους, πάνω στην απόδοση των παραλλαγών TCP υψηλής ταχύτητας. Το packetreorderinggenerator μπορεί να περιγραφεί από τις ακόλουθες τρεις παραμέτρους:



Σχήμα 4.1 Αρχική ακολουθία χωρίς να αναδιατάσσονται τα πακέτα



Σχήμα 4.2 Νέα ακολουθία πακέτων αφού το πακέτο 2 αναδιαταχθεί με μέγεθος packetreordering = 2 και delaytime = 4 πακέτα



Σχήμα 4.3 Νέα ακολουθία πακέτων αφού τα πακέτα 2 και 3 αναδιαταχθούν, με μέγεθος packetreordering = 2 και reorderingdelaytime = 4 πακέτα

- Διάστημα Αναδιάταξης: το χρονικό διάστημα μεταξύ δύο διαδοχικών γεγονότων packetreordering.
- Reorderingdelaytime (Χρονική καθυστέρηση αναδιάταξης): το χρονικό διάστημα από το πρώτο packetreordering σε περίπτωση αναδιάταξης στο παλαιότερο προωθημένο πακέτο με μεγαλύτερο αριθμό ακολουθίας.
- Μέγεθος reorderingblock: ο αριθμός των πακέτων που αναδιατάσσεται ως σύνολο.

Το `packetreorderinggenerator` λειτουργεί ως ειδικός δρομολογητής με μία θύρα εισόδου και μία θύρα εξόδου, και προωθεί όλα τα εισερχόμενα πακέτα στη θύρα εξόδου. Για να εισάγει το `packetreordering`, έχει καθυστερήσει ένα σύνολο πακέτων μεγέθους `reorderingblock` για κάποιο `reorderingdelaytime` για κάθε διάστημα αναδιάταξης, το οποίο ονομάζεται φαινόμενο αναδιάταξης. Όλα τα άλλα πακέτα προωθούνται αμέσως χωρίς καμία καθυστέρηση. Και οι τρεις παράμετροι θα μπορούσαν να είναι τυχαίες μεταβλητές ακολουθώντας συγκεκριμένες διανομές.

Ένα `reorderingdelaytime` μετριέται σε δύο διαφορετικές μονάδες. Υποτίθεται ότι το  $t(P_i)$  υποδηλώνει την ώρα που το πακέτο  $P_i$  αναχωρεί από ένα κύκλωμα. Επιπλέον, το  $P_x$  είναι το πρώτο `reorderedpacket` σε ένα συμβάν αναδιάταξης, και το  $P_y$  είναι το παλαιότερο πακέτο που έφυγε από το κύκλωμα μεταξύ όλων των πακέτων, με μεγαλύτερο αριθμό ακολουθίας (δηλαδή,  $t(P_y) < t(P_x)$  και  $y > x$ ). Πρώτον, το `reorderingdelaytime` μετριέται ως χρονικό διάστημα  $t(P_y) - t(P_x)$ . Δεύτερον, μετράται ως ο αριθμός των πακέτων που διαβιβάζεται μεταξύ του  $t(P_y)$  και του  $t(P_x)$ .

Να σημειωθεί ότι μπορεί να καθοριστεί μία από αυτές τις δύο τιμές από μια άλλη, αν ο ρυθμός αποστολής των πακέτων είναι γνωστός και υποθέτοντας την ίδια χρονική στιγμή άφιξης για όλα τα πακέτα. Και οι δύο μονάδες (δηλαδή, δευτερόλεπτα και πακέτα) που χρησιμοποιούνται σε αυτό το άρθρο, βασίζονται σε ποιο από το καθένα είναι πιο βολικό.

#### **4.4 ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΗΣ ΑΝΑΔΙΑΤΑΞΗΣ ΠΑΚΕΤΩΝ ΣΕ ΔΙΚΤΥΑ ΥΨΗΛΩΝ ΤΑΧΥΤΗΤΩΝ**

Οι λογικές τιμές για τρεις παραμέτρους βρίσκονται αναλύοντας τα πειραματικά αποτελέσματα που διεξήχθησαν από τον Gharai et al. [33] το 2004. Οι Gharai et al. μέτρησαν την εμφάνιση του `packet reordering` μεταδίδοντας ροές UDP με διάφορα

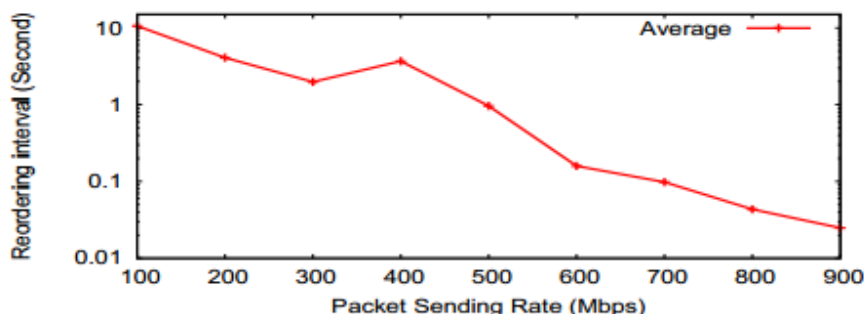
μεγέθη πακέτων για ένα λεπτό με ρυθμό που κυμαίνεται από 1Mbps έως 900Mbps μεταξύ Washington DC, Pittsburgh και Los Angeles πάνω από το DARPA SuperNet. Αναλύονται όλα τα πειράματά τους και παρατηρούνται παρόμοια αποτελέσματα για πειράματα με διαφορετικά μεγέθη πακέτων. Περιορίζεται από το διάστημα, κάτω από το οποίο αναλύεται και παρουσιάζεται τα ευρήματα μόνο για πειράματα με μέγεθος πακέτου 1500 bytes.

Το σχήμα 4.4 δείχνει το διάστημα αναδιάταξης, που είναι κατά μέσο όρο για όλα τα διαστήματα αναδιάταξης, σε όλα τα πειράματα, με μέγεθος πακέτου 1500 byte. Δείχνει σαφώς ότι όσο υψηλότερος είναι ο ρυθμός αποστολής πακέτων, τόσο μικρότερο είναι το διάστημα αναδιάταξης. Έτσι είναι τα περισσότερα γεγονότα αναδιάταξης. Για παράδειγμα, το μέσο διάστημα αναδιάταξης στα 100Mbps είναι περίπου 10 δευτερόλεπτα, ενώ αυτό στα 900Mbps είναι μόνο περίπου 0,02 δευτερόλεπτα. Αυτό συμβαδίζει με τις παρατηρήσεις στο [33]. Παρατηρείται επίσης ότι ακόμη και στο ίδιο πείραμα, το διάστημα αναδιάταξης δεν είναι σταθερό. Για παράδειγμα, η αθροιστική κατανομήσυνάρτησης (CDF: CumulativeDistributionFunction) των διαστημάτων αναδιάταξης, σε ένα πείραμα με ρυθμό 900Mbps που φαίνεται στο σχήμα 4.5, υποδεικνύει ότι τα μέγιστα και ελάχιστα διαστήματα αναδιάταξης που παρατηρήθηκαν από την ίδια ροή μπορεί να διαφέρουν περισσότερο από 100 φορές. Αυτό υποδηλώνει ότι προηγούμενα αποτελέσματα προσομοίωσης που λήφθηκαν με ένα σταθερό διάστημα αναδιάταξης ενδέχεται να μην αντανakλούν την πραγματική απόδοση του TCP παρουσία packetreordering.

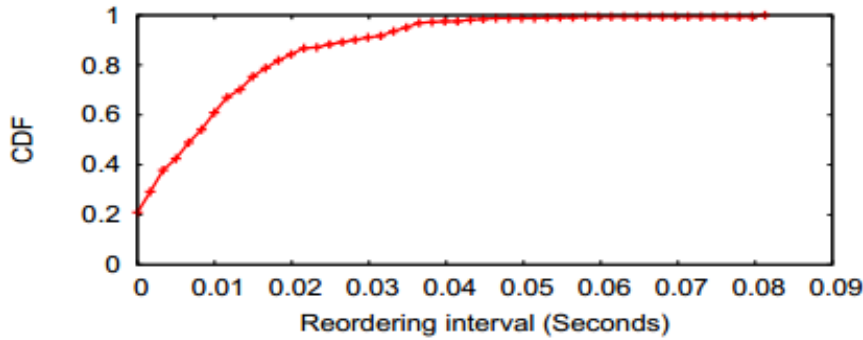
Το Σχήμα 4.6 δείχνει το μέσους και μέγιστο reorderingdelaytime που μετριέται με τον πρώτο υπολογισμό του μέσου και του μέγιστου reorderingdelaytime για κάθε πείραμα και στη συνέχεια τον υπολογισμό του μέσου όρου μεταξύ όλων των πειραμάτων με τον ίδιο ρυθμό αποστολής. Ωστόσο, συνεχίζεται να είναι εμφανή

ότι το μέσο και το μέγιστο reorderingdelaytime ελαφρά να αυξάνονται όσο αυξάνεται ο ρυθμός αποστολής πακέτων. Για παράδειγμα, το μέσο reorderingdelaytime στα 100Mbps είναι μόνο 1 πακέτο, ενώ ο αριθμός των 900Mbps είναι μεγαλύτερος από 2 πακέτα. Επιπλέον, στις περισσότερες περιπτώσεις, το μέσο reorderingdelaytime είναι μικρότερος από 3 πακέτα. Δηλαδή, ένας σημαντικός αριθμός περιπτώσεων αναδιάταξης οδηγεί σε λιγότερα από τρία dupACKs, και στη συνέχεια δεν προκαλούν ψευδή fastretransmit. Ωστόσο, μια ροή μεγάλης ταχύτητας είναι πιο πιθανό να προκαλέσει γεγονότα αναδιάταξης με reorderingdelaytime αρκετά μεγάλο, έτσι ώστε να προκαλέσει ψευδή fastretransmit.

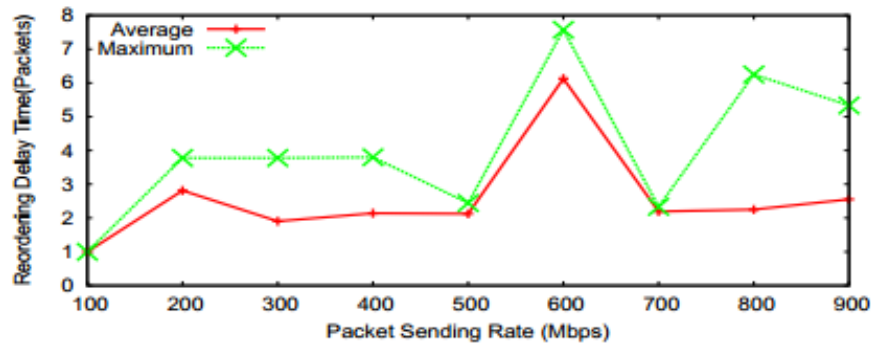
Το σχήμα 4.7 δείχνει το μέσο και το μέγιστο μέγεθος reorderingblock. Σημειώνεται ότι το μέσο μέγεθος reorderingblock είναι σχετικά αναισθητο στο ρυθμό αποστολής πακέτων και είναι πάντα πολύ κοντά στο 1. Δηλαδή, στις περισσότερες περιπτώσεις αναδιάταξης, μόνο ένα πακέτο αναδιατάσσεται, ακόμη και για μια ροή με υψηλό ρυθμό αποστολής. Επίσης, φαίνεται καθαρά ότι το μέγιστο μέγεθος reorderingblock αυξάνεται, καθώς αυξάνεται ο ρυθμός. Αυτό σημαίνει ότι είναι πιο πιθανό να αναδιαταχθεί ένας μεγαλύτερος αριθμός πακέτων ως σύνολο για μια ροή με υψηλότερο ρυθμό αποστολής.



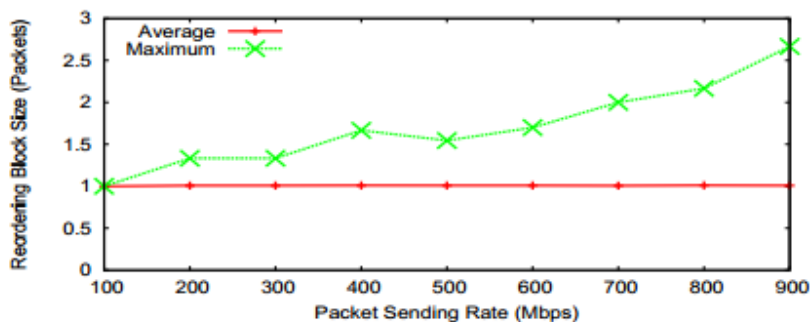
Σχήμα 4.4 Όσο υψηλότερος είναι ο ρυθμός αποστολής πακέτων, τόσο μικρότερο είναι το μέσο διάστημα αναδιάταξης. Δηλαδή, τόσο περισσότερα τα γεγονότα αναδιάταξης



Σχήμα 4.5 Η Αθροιστική Κατανομή Συνάρτησης(CDF) των διαστημάτων αναδιάταξης σε ένα πείραμα με ρυθμό 900 Mbps



Σχήμα 4.6 Το μέσο και το μέγιστο reorderingdelaytime αυξάνεται ελαφρά καθώς αυξάνεται ο ρυθμός αποστολής πακέτων



Σχήμα 4.7 Παρόλο που το μέσο μέγεθος reorderingblock παραμένει σχεδόν το ίδιο για όλους τους ρυθμούς αποστολής, το μέγιστο μέγεθος reorderingblock αυξάνεται όσο αυξάνεται ο ρυθμός

## **4.5 ΕΠΙΠΤΩΣΗ ΤΟΥ PACKETREORDERING ΣΕ ΔΙΚΤΥΑ ΥΨΗΛΩΝ ΤΑΧΥΤΗΤΩΝ**

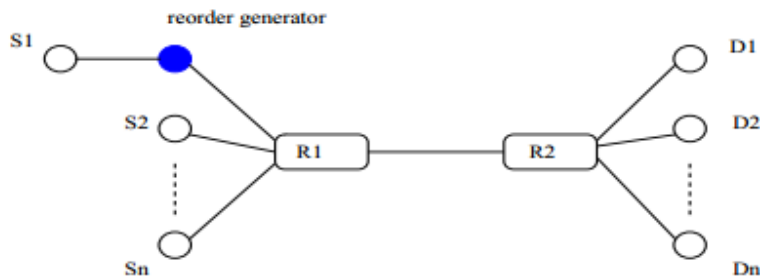
### **A. Εγκατάσταση προσομοίωσης**

Το Σχήμα 4.8 δείχνει την εγκατάσταση προσομοίωσης NS-2. Το reorderinggenerator εισάγεται μεταξύ της υψηλής ταχύτητας του TCP αποστολέα S1 και του δρομολογητή R1, έτσι ώστε μόνο η σύνδεση TCP μεταξύ S1 και D1 να βιώνει packetreordering. Το bandwidth και η μονόδρομη καθυστέρηση του συνδέσμου συμφόρησης έχουν οριστεί σε 1000 Mbps και 50 ms, αντίστοιχα. Κάθε πηγή και αποδέκτης είναι συνδεδεμένοι με το σύνδεσμο συμφόρησης μέσω διαφόρων συνδέσμων πρόσβασης, με τυχαίες καθυστερήσεις, που κυμαίνονται από 0,1 ms έως 0,9 ms για να μετριάσουν την φάση επίδρασης. Για να αυξηθεί η δυναμική της κυκλοφορίας και να μειωθεί η φάση επίδρασης, προσομοιώνονται διάφορα είδη σχεδίων κυκλοφορίας και στις δύο κατευθύνσεις.

Τόσο το διάστημα αναδιάταξης όσο και το reorderingdelaytime είναι τυχαίες μεταβλητές, με κατανομή περίπου που ακολουθεί το CDFS τους αντίστοιχα. Εκτός και αν αναφέρεται διαφορετικά, δηλαδή ότι οι μέσοι όροι του διαστήματος αναδιάταξης και το reorderingdelaytime έχουν οριστεί σε 0,02 δευτερόλεπτα και 20 μs αντίστοιχα, οι οποίοι αντιστοιχούν στις μετρηθείσες μέσες τιμές τους σε 900Mbps. Το μέγεθος του packetreordering ρυθμίζεται σε σταθερή τιμή του 1 πακέτου.

Χρησιμοποιείται το TCP / SACK ως παράγοντας για τις συνδέσεις του TCP (εκτός από την αναδιάταξη που είναι ανεκτή στους αλγόριθμους, που χρειάζονται

ειδικούς παράγοντες του TCP). Το TCP / SACK επιλέγεται για υψηλότερες επιδόσεις, σε σύγκριση με το πρωτόκολλο TCP / Reno και TCP / NewReno, και οι περισσότεροι από τους αλγόριθμους αναδιάταξης βασίζονται στο TCP / SACK.



Σχήμα 4.8 Εγκατάσταση προσομοίωσης NS-2

## **B. Αποτελεσματα Προσομοίωσης**

Το σχήμα 4.9 δείχνει την επίδραση των διαστημάτων αναδιάταξης στις επιδόσεις του BIC και του HSTCP. Μεταβάλλοντας το μέσο διάστημα αναδιάταξης από 0,01 δευτερόλεπτα σε 0,5 δευτερόλεπτα και ρυθμίζοντας τις άλλες δύο παραμέτρους στις τιμές που δίνονται στην Ενότητα 4.5-A, γίνονται οι ακόλουθες τρεις παρατηρήσεις. Κατ' αρχάς, ένα μικρότερο διάστημα αναδιάταξης οδηγεί σε χαμηλότερη απόδοση. Δεύτερον, η απόδοση του BIC πέφτει απότομα, όταν το διάστημα αναδιάταξης είναι μικρότερο από 0,02 δευτερόλεπτα. Αυτό οφείλεται στο γεγονός ότι, όταν το διάστημα αναδιάταξης είναι αρκετά μικρό, έτσι ώστε μια ροή να μπορεί να παρουσιάσει περισσότερα από ένα συμβάντα αναδιάταξης εντός ενός RTT, δεν μπορεί να ανακάμψει αποτελεσματικά από ψευδή fastretransmit και fastrecovery. Τρίτον, το HSTCP πάσχει περισσότερο από το φαινόμενο του packetreordering σε σύγκριση με το BIC. Για παράδειγμα, με ένα διάστημα αναδιάταξης 0,5 δευτερολέπτων, το BIC μπορεί να επιτύχει ταχύτητα διασύνδεσης 400Mbps. Ωστόσο, το HSTCP μπορεί να επιτύχει μόνο



10Mbps. Αυτό οφείλεται στο γεγονός ότι το BIC είναι πιο επιθετικό από το HSTCP σε δίκτυα υψηλής ταχύτητας.

Το σχήμα 4.10 αξιολογεί το αντίκτυπο των `reorderingdelaytime`. Μεταβάλλοντας τον μέσο `reorderingdelaytime` από 0 έως 100  $\mu$ s και ρυθμίζοντας τις άλλες δύο παραμέτρους στις τιμές που δίδονται στο κεφάλαιο 4.5-A, σημειώνεται ότι τόσο το BIC όσο και το HSTCP επιτυγχάνουν πολύ υψηλές παροχές, όταν δεν υπάρχει `packetreordering`.

Ωστόσο, ακόμη και ένα πολύ μικρό `reorderingdelaytime` μπορεί να υποβαθμίσει σημαντικά την απόδοση τους. Αυτό οφείλεται στο γεγονός ότι το μέσο διάστημα αναδιάταξης έχει οριστεί σε 0,02 δευτερόλεπτα. Αυτό το διάστημα είναι τόσο μικρό, που μια ροή μπορεί να παρουσιάσει περισσότερα από ένα συμβάντα αναδιάταξης εντός ενός RTT. Με ένα μικρό `reorderingdelaytime`, παρόλο που ένα συμβάν αναδιάταξης μόνο δεν μπορεί να οδηγήσει σε ψευδή αναμετάδοση TCP, δύο ή περισσότερα συμβάντα αναδιάταξης είναι πιο πιθανό να προκαλέσουν λανθασμένο `retransmittTCP` και `fastrecovery`. [33]

Δεύτερον, παρατηρείται ότι η περαιτέρω αύξηση του `reorderingdelaytime` μειώνει ελαφρώς την απόδοση. Διαισθητικά, όταν ενεργοποιηθεί λανθασμένα το `fastretransmit` και το `fastrecovery`, ένα μεγαλύτερο `reorderingdelaytime` καθιστά την ροή να παραμείνει σε `fastrecovery` λίγο παραπάνω, γεγονός που δεν αλλάζει υπερβολικά την απόδοση.

Το σχήμα 4.11 παρουσιάζει τις διελεύσεις των BIC και HSTCP για διαφορετικά μεγέθη `reorderingblock`. Όπως φαίνεται από το σχήμα, η επίπτωση του μεγέθους του `reorderingblock` είναι αρκετά παρόμοια με το `reorderingdelaytime`. Ακόμη και ένα μικρό μέγεθος `reorderingblock` μπορεί να ρίξει σημαντικά την απόδοση των BIC και HSTCP. Ωστόσο, η περαιτέρω αύξηση του μεγέθους του `reorderingblock` ελαττώνει ελαφρώς την απόδοση. Ο λόγος είναι

ότι ο μηχανισμός TCP/SACK μπορεί να ανακάμψει γρήγορα από πολλαπλά πακέτα που έχουν χαθεί ή να αναδιατάξει πακέτα σε ένα παράθυρο συμφόρησης, εκτός εάν ενεργοποιηθούν μερικά από τα επόμενα ACKs.

#### **4.6 ΕΠΙΠΤΩΣΗ ΤΩΝ ΥΦΙΣΤΑΜΕΝΩΝ ΕΝΙΣΧΥΣΕΩΝ ΤΟΥ TCP**

**A. Αξιολόγηση των υφιστάμενων βελτιώσεων TCP που είναι ανεκτικές στις αναπροσαρμογές.**

Σε αυτή την ενότητα αξιολογείται η αποτελεσματικότητα των τρέχοντων αλγορίθμων αναδιάταξης σε δίκτυα υψηλών ταχυτήτων, προσομοιώνοντας τα TCP-NCR, AVG-DEV και TCP-RR, οι οποίοι είναι τρεις πρόσφατοι προτεινόμενοι αλγόριθμοι για να καταστεί το TCP πιο εύρωστο στο packetreordering. Χρησιμοποιούν ελαφρώς διαφορετικές μεθόδους για την αποτροπή της λανθασμένης γρήγορης αναμετάδοσης και της γρήγορης ανάκτησης TCP. Τα αποτελέσματα προσομοίωσης δείχνουν ότι λειτουργούν καλύτερα από άλλους παρόμοιους αλγόριθμους αναδιάταξης στα περιβάλλοντα προσομοίωσης που έχουν μελετηθεί.

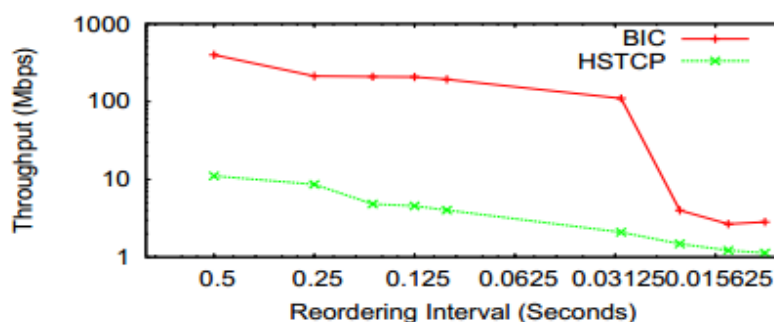
Έχουν αποκτηθεί παρόμοια αποτελέσματα προσομοίωσης τόσο για το BIC όσο και για το HSTCP, και παρακάτω παρουσιάζονται τα αποτελέσματα μόνο για το BIC. Παρακάτω προσομοιώνεται το BIC με κάθε έναν από αυτούς τους αλγόριθμους αναδιάταξης και αξιολογείται η αποτελεσματικότητά τους, καθώς μεταβάλλεται κάθε μια από τις τρεις παραμέτρους αναδιάταξης της γεννήτριας αναδιάταξης. Χρησιμοποιούνται τα ίδια σύνολα παραμέτρων προσομοίωσης όπως στα Σχήματα 4.9, 4.10 και 4.11. Τα Σχήματα 4.12, 4.13 και 4.14 δείχνουν την αποτελεσματικότητα αυτών των τριών αλγορίθμων αναδιάταξης, καθώς

μεταβάλλεται το διάστημα αναδιάταξης, `reorderingdelaytime` και το μέγεθος του `reorderingblock`.

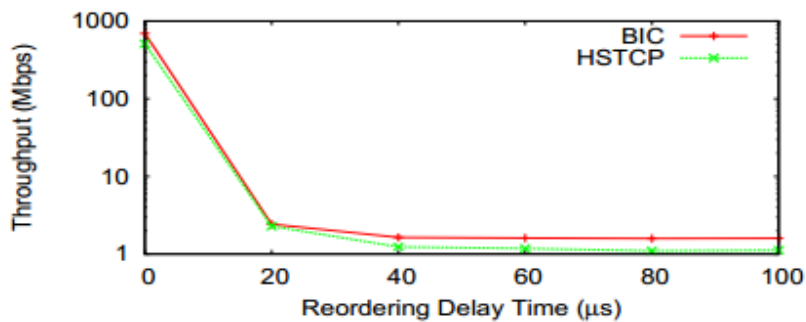
Παρουσιάζεται επίσης η απόδοση του BIC χωρίς αλγόριθμο αναδιάταξης (που αναφέρεται ως BIC-SACK στις εικόνες). Γίνονται οι ακόλουθες τρεις παρατηρήσεις. Πρώτον, όλοι οι αλγόριθμοι αναδιάταξης μπορούν να βελτιώσουν σημαντικά την απόδοση του BIC σε όλα τα σενάρια προσομοίωσης.

Δεύτερον, το TCP-NCR λειτουργεί σταθερά καλύτερα από τους άλλους δύο αλγόριθμους: AVG-DEV και TCP-RR. Αυτό είναι αναμενόμενο, δεδομένου ότι το TCP-NCR έχει το μεγαλύτερο `dupthresh` (δηλ. ο αριθμός των `dupack` για να ενεργοποιήσει το `fastretransmitTCP` και το `recovery`) και το `virtuallayer`.

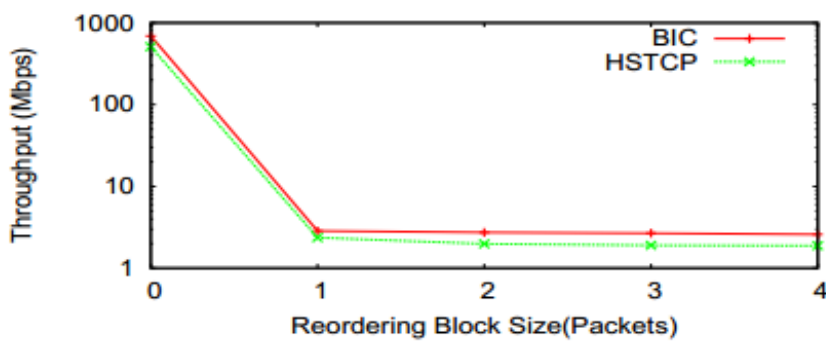
Τρίτον, ακόμα και με το TCP-NCR, το BIC εξακολουθεί να υποφέρει από `packetreordering`, ειδικά με μικρά διαστήματα αναδιάταξης και μεγάλα μεγέθη αναδιάταξης. Για παράδειγμα, το σχήμα 4.14 δείχνει ότι το BIC με TCP-NCR επιτυγχάνει απλή μετάδοση 463Mbps με μέγεθος `reorderingblock` 2 πακέτων, ενώ επιτυγχάνει διασύνδεση 632Mbps με μέγεθος `reorderingblock` 1 πακέτου. Διαισθητικά, παρόλο που το TCP-NCR μπορεί να αποτρέψει αποτελεσματικά τα περισσότερα ψευδή `fastretransmit` και `fastrecovery`, λόγω του `packetreordering`, δεν μπορεί να ανακάμψει γρήγορα από το `fast retransmit` και `fast recovery`, που προκαλείται από την απώλεια πακέτων και το `packetreordering`.



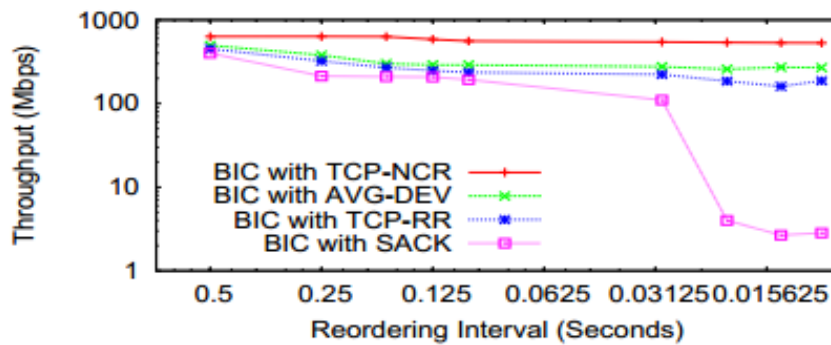
Σχήμα 4.9 Όσο μικρότερος είναι ο χρόνος αναμετάδοσης, τόσο χαμηλότερη είναι η απόδοση των παραλλαγών TCP υψηλής ταχύτητας. Το BIC επιτυγχάνει καλύτερη απόδοση από το HSTCP, αλλά εξακολουθεί να υποφέρει από πολύ μικρά διαστήματα αναδιάταξης.



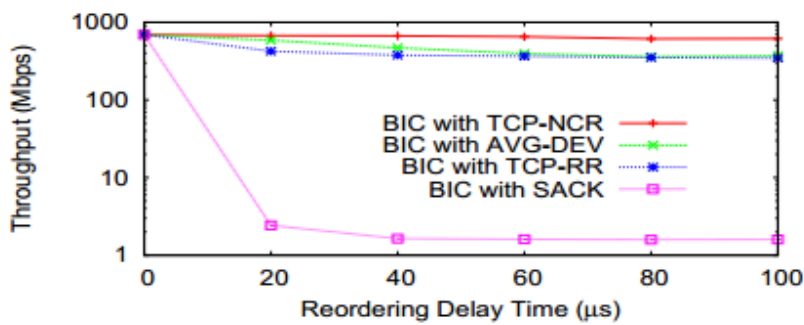
Σχήμα 4.10 Ακόμα και ένα μικρό reorderingdelaytime (όπως 20 μs) μπορεί να μειώνει σημαντικά την απόδοση των BIC και HSTCP. Ωστόσο, η περαιτέρω αύξηση του reorderingdelaytime μειώνει ελαφρώς την απόδοση



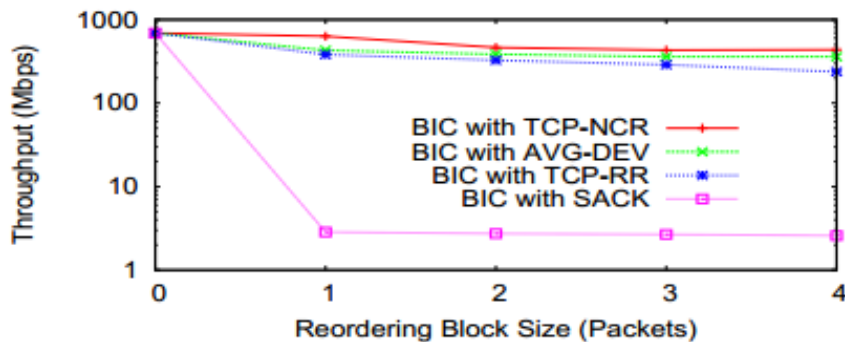
Σχήμα 4.11 Ακόμη και ένα μικρό μέγεθος reorderingblock (όπως 1 πακέτο) μπορεί να μειώνει σημαντικά την απόδοση των BIC και HSTCP. Ωστόσο, η περαιτέρω αύξηση του μεγέθους του reorderingblock ελαττώνει ελαφρώς την απόδοση



Σχήμα 4.12 Οι υπάρχοντες αλγόριθμοι αναδιάταξης μπορούν να βελτιώσουν σημαντικά την απόδοση του BIC. Ωστόσο, το BIC εξακολουθεί να πάσχει από packetreordering με μικρά διαστήματα αναδιάταξης.



Σχήμα 4.13 Οι υπάρχοντες αλγόριθμοι αναδιάταξης μπορούν να βελτιώσουν σημαντικά την απόδοση του BIC. Το BIC μπορεί να επιτύχει πολύ υψηλή απόδοση ακόμα και με μεγάλα reorderingdelaytime.



Σχήμα 4.14 Οι υπάρχοντες αλγόριθμοι αναδιάταξης μπορούν να βελτιώσουν σημαντικά τη απόδοση του BIC. Ωστόσο, το BIC εξακολουθεί να υποφέρει από packetreordering με μεγάλα μεγέθη reorderingblock

### **B.Σχετικές Αναδιατάξεις-Ανορθωτικές βελτιώσεις TCP**

Οι περισσότεροι αλγόριθμοι αναδιάταξης αυξάνουν το dupthresh από την προεπιλεγμένη τιμή 3 σε μεγαλύτερο αριθμό, για να αποτρέψουν την λανθασμένη αναμετάδοση και ανάκτηση του TCP. Οι Blanton και Allman ρυθμίζουν το dupthresh, χρησιμοποιώντας το ιστορικό αναδιάταξης, με βάση τις πληροφορίες DSACK. Το TCP-RR [31] που προτάθηκε από τους Zhan et al., χρησιμοποιεί το D-SACK και μια συνάρτηση κόστους για το χρονικό όριο και το ψευδή fast retransmit για να ρυθμίσει το dupthresh. Το AVG-DEV [34], που αναπτύχθηκε από τους Ma και Leung, προσαρμόζει το dupthresh λαμβάνοντας υπόψη τόσο τον μέσο όρο όσο και την απόκλιση του ιστορικού αναδιάταξης.

Το TCP-NCR [35], που σχεδιάστηκε από τους Bhandarkar et al., ρυθμίζει το μέγεθος ενός παραθύρου συμφόρησης. Το TCP-Aix, που προτάθηκε πρόσφατα από τους Ekstrom et al., ρυθμίζει το dupthresh και διαχωρίζει την απώλεια ανάκαμψης από τον έλεγχο συμφόρησης.

## **ΚΕΦΑΛΑΙΟ 5**

### **5.1 ΕΠΙΜΟΝΗ ΑΝΑΔΙΑΤΑΞΗ ΠΑΚΕΤΩΝ ΣΤΟ TCP**

Στο TCP-PR (TCPforPersistentPacketReordering ) η βασική ιδέα του πρωτοκόλλου είναι να ανιχνεύει απώλειες πακέτων μέσω της χρήσης χρονομέτρων αντί των διπλών ACKs. Αυτό υπαγορεύεται από την παρατήρηση ότι, υπό το συνεχές packetreordering, οι εις διπλούν επιβεβαιώσεις είναι μια κακή ένδειξη απώλειας πακέτων. Επειδή το TCP-PR στηρίζεται αποκλειστικά σε χρονόμετρα για να ανιχνεύσει την απώλεια πακέτων, είναι επίσης ισχυρό στην αναγνώριση ζημιών. Αυτό συμβαίνει επειδή ο αλγόριθμος δεν κάνει διάκριση μεταξύ απωλειών των δεδομένων (σχετικά με την προς τα εμπρός διαδρομή) ή απωλειών ACKs (στην αντίστροφη διαδρομή). Οι προτεινόμενοι αλγόριθμοι απαιτούν μόνο αλλαγές στον αποστολέα TCP και επομένως είναι συμβατοί με οποιοδήποτε δέκτη TCP. Ο TCP-PR αλγόριθμος του αποστολέα εξακολουθεί να βασίζεται στην έννοια του παράθυρου συμφόρησης, αλλά η ενημέρωση του παράθυρου συμφόρησης ακολουθεί ελαφρώς διαφορετικούς κανόνες από το πρότυπο TCP. Ωστόσο, δόθηκε σημαντική προσοχή στο να καταστεί ο αλγόριθμος δίκαιος σε σχέση με άλλες εκδόσεις του TCP για να βεβαιωθεί ότι μπορούν να συνυπάρξουν.

### **5.2 ΒΑΣΙΚΟΣ ΑΛΓΟΡΙΘΜΟΣ**

Τα πακέτα που επεξεργάζεται ο αποστολέας, διατηρούνται σε μία από τις δύο λίστες: Η λίστα να-σταλούν (to-be-sent) περιέχει όλα τα πακέτα των οποίων η μετάδοση είναι σε εκκρεμότητα, περιμένοντας για ένα "άνοιγμα" στο παράθυρο συμφόρησης. Η λίστα αναμονής (to-be-ack) περιέχει τα πακέτα που έχουν ήδη

αποσταλεί, αλλά δεν έχουν ακόμη αναγνωρισθεί. Συνήθως, όταν μια εφαρμογή παράγει ένα πακέτο αρχικά τίθεται στη λίστα to-be-sent. Όταν το παράθυρο συμφόρησης το επιτρέψει, το πακέτο στέλνεται στον παραλήπτη και μετακινείται στη λίστα to-be-ack. Τέλος, όταν ένα ACK για το συγκεκριμένο πακέτο φθάνει από το δέκτη, αφαιρείται από τη λίστα to-be-ack (κάτω από αθροιστικά ACKs, πολλά πακέτα θα αφαιρεθούν ταυτόχρονα από την αναμονή). Εναλλακτικά, όταν ανιχνεύεται ότι ένα πακέτο έχει απορριφθεί, μετακινείται από τη λίστα to-be-ack πίσω στη λίστα to-be-sent.

Όπως αναφέρθηκε παραπάνω, οι απορρίψεις ανιχνεύονται πάντα μέσω χρονομέτρων. Για το σκοπό αυτό, κάθε φορά που ένα πακέτο στέλνεται στον παραλήπτη και τοποθετείται στη λίστα to-be-ack, είναι σφραγισμένο χρονικά. Όταν ένα πακέτο παραμένει στη λίστα to-be-ack πάνω από ένα ορισμένο χρονικό διάστημα χαρακτηρίζεται ως απορριπτέο. Πιο συγκεκριμένα, υποθέτοντας ότι ένα πακέτο απορρίφθηκε σε χρόνο  $t$ , όταν το  $t$  υπερβαίνει τη χρονοσφραγίδα του πακέτου στη λίστα to-be-ack, υπάρχει ένας εκτιμώμενος μέγιστος χρόνος με επιστροφή.

Όταν αποστέλλονται τα πακέτα και λαμβάνονται τα ACKs, μια εκτίμηση  $m_{xrtt}$  του μέγιστου δυνατού χρόνου με επιστροφή ενημερώνεται συνεχώς. Η εκτίμηση που χρησιμοποιείται δίνεται από:  $m_{xrtt} = \beta * e_{wrtt}$ .

όπου  $\beta$  είναι μια σταθερά μεγαλύτερη από 1 και καταγράφει έναν εκθετικά σταθμισμένο μέσο όρο των τελευταίων RTTs. Όταν έρχεται ένα νέο ACK, και ενημερώνεται το  $e_{wrtt}$  ως εξής:  $e_{wrtt} = \max \{ a \frac{1}{cwnd} * e_{wrtt}, sample - rtt \}$  (1),

όπου το  $a$  υποδηλώνει μια θετική σταθερά μικρότερη από 1, το  $cwnd$  υποδηλώνει το τρέχον μέγεθος του παράθυρου συμφόρησης, και το Sample-RTT, το RTT για το πακέτο του οποίου η αναγνώριση μόνο ήταν 5. Ο λόγος για την αύξηση  $a$  στη δύναμη  $\frac{1}{cwnd}$  είναι ότι σε ένα RTT ο τύπος (1) επαναλαμβάνεται



cwnd φορές. Αυτό σημαίνει ότι, για παράδειγμα, αν υπήρχε μια ξαφνική μείωση στο RTT, τότε το ewrtt θα μειωνόταν κατά ένα ποσοστό ( $\alpha$  cwnd1)  $cwnd = \alpha$  ανά RTT, ανεξάρτητα από την τρέχουσα τιμή του παράθυρου συμφόρησης. Η παράμετρος  $\alpha$  μπορεί να ερμηνευθεί ως παράγοντας μνήμης σε μονάδες RTTs. Το ewrtt δεν είναι μια ομαλοποιημένη έκδοση RTT. Ως εκ τούτου, αυτή η προσέγγιση δεν είναι σαν τις πρώτες εκδόσεις του υπολογισμού RTOTCP's που βασίζονται αποκλειστικά σε εκτιμήσεις του μέσου RTT.

Αντίθετα, αν μια παρελθοντική παρατήρηση του RTT είναι μεγάλη, αυτό το μεγάλο RTT είναι υπεύθυνο για την τιμή του ewrtt για λίγο καιρό. Με τον τρόπο αυτό, το ewrtt θα επανεξετάσει κάποιες αιχμές στο RTT. Αυτό είναι παρόμοιο με τον αλγόριθμο VanJacobson, όπου οι τελευταίες διακυμάνσεις RTT έχουν σημαντική επίπτωση στο RTO [35]. Η απόδοση του αλγορίθμου δεν είναι στην πραγματικότητα πολύ ευαίσθητη στις μεταβολές των παραμέτρων  $\beta$  και  $\alpha$ , με την προϋπόθεση ότι έχουν επιλεγεί από τις κατάλληλες κλίμακες.

Υπάρχουν δύο τρόποι για την ενημέρωση του παράθυρου συμφόρησης: slow-start και congestion-avoidance. Ο αποστολέας ξεκινά πάντα σε slow-start και επιστρέφει μόνο σε slow-start μετά από περιόδους ακραίων απωλειών. Σε αυτή τη λειτουργία, το cwnd ξεκινά με την τιμή ένα και αυξάνεται σημαντικά (ένα για κάθε ACK που θα ληφθεί). Μόλις η πρώτη απώλεια ανιχνευθεί, το cwnd γίνεται μισό και ο αποστολέας μετακινείται στην κατάσταση congestion-avoidance, όπου το cwnd αυξάνεται γραμμικά ( $\frac{1}{cwnd}$  για κάθε ACK που θα ληφθεί). Μεταγενέστερες απώλειες προκαλούν περαιτέρω μείωση κατά το ήμισυ του cwnd, χωρίς ο αποστολέας να χρειαστεί να αφήσει τη λειτουργία congestion-avoidance. Ένα σημαντικό αλλά λεπτό σημείο στη μείωση του cwnd κατά το ήμισυ, είναι ότι όταν ένα πακέτο αποστέλλεται, εκτός της χρονοσφαγίδας, αποθηκεύεται στη λίστα to-be-ack και η τρέχουσα τιμή του cwnd. Όταν ανιχνεύεται μια απώλεια πακέτου,

τότε το cwnd τίθεται ίσο με το μισό της τιμής του cwnd τη στιγμή που το πακέτο στάλθηκε και όχι στο μισό της τρέχουσας τιμής του cwnd. Αυτό κάνει τον αλγόριθμο αρκετά ισχυρό στην καθυστέρηση που λαμβάνει χώρα, μεταξύ του χρονικού διαστήματος που χρειάζεται για να εμφανιστεί η απώλεια.

Για να αποφευχθούν μεγάλες απώλειες από την πρόκληση υπερβολικών μειώσεων του cwnd, μόλις μια απώλεια ανιχνευθεί, παίρνεται ένα στιγμιότυπο από τη λίστα to-be-sent και αποθηκεύεται σε μια βοηθητική λίστα που ονομάζεται απομνημόνευση (memorize). Όταν τα πακέτα αναγνωριστούν ή δηλωθούν ως απώλεια, αφαιρούνται από τη λίστα memorize, έτσι ώστε η λίστα αυτή να περιέχει μόνο τα πακέτα που στάλθηκαν πριν το cwnd γίνει μισό και δεν έχουν καταγραφεί ακόμα. Όταν ένα πακέτο σε αυτήν τη λίστα θεωρείται ως απώλεια, το cwnd δε μειώνεται στο μισό. Η λογική για αυτό είναι ότι ο αποστολέας ήδη αντέδρασε με την συμφόρηση που προκλήθηκε από αυτή τη μεγάλη απώλεια. Αυτό το είδος της λογικής βρίσκεται επίσης στο TCP-NewReno και στο TCPSACK.

Στο TCP-PR, τα πακέτα αποστέλλονται μόνο όταν το παράθυρο συμφόρησης το επιτρέπει. Συγκεκριμένα, όταν το cwnd υπερβαίνει τον αριθμό των πακέτων στη λίστα to-be-ack. Στην πράξη, αυτό σημαίνει ότι τα περισσότερα πακέτα αποστέλλονται όταν μια αναγνώριση φτάνει και ένα πακέτο αφαιρείται από τη λίστα to-be-ack. Ως εκ τούτου, το πρωτόκολλο TCP-PR παρουσιάζει τον τύπο του αυτοχρονισμού να είναι κοινός με άλλες εκδόσεις του TCP. Τα πακέτα μπορούν επίσης να αποσταλούν όταν ανιχνευθεί μια απώλεια, αφού όταν αυτό συμβαίνει, ένα πακέτο αφαιρείται επίσης από τη λίστα to-be-ack. Ο ψευδοκώδικας στον Πίνακα 1 αντιστοιχεί στον αλγόριθμο που μόλις περιγράφηκε. Ο Πίνακας 2 συνοψίζει την σημειογραφία που χρησιμοποιείται στον κώδικα.

Event	Code
initialization	<pre> 1  mode := <i>slow-start</i> 2  cwnd := 1 3  ssthresh := +∞ 4  memorize := ∅ </pre>
time > time( <i>n</i> ) + mxrtt (drop detected for packet <i>n</i> )	<pre> 5  remove(to-be-ack, <i>n</i>) 6  add(to-be-sent, <i>n</i>) 7  if not is-in(memorize, <i>n</i>) then /* new drop */ 8      memorize := to-be-ack 9      cwnd := cwnd(<i>n</i>)/2 10     ssthresh := cwnd 11  else /* other drops in burst */ 12     remove(memorize, <i>n</i>) 13  flush-cwnd() </pre>
ack received for packet <i>n</i>	<pre> 14  srtt = max { <math>\alpha^{\frac{1}{cwnd}}</math> × srtt, time - time(<i>n</i>) } 15  mxrtt := <math>\beta</math> × srtt 16  remove(to-be-ack, <i>n</i>) 17  remove(memorize, <i>n</i>) 18  if mode = <i>slow-start</i> and cwnd + 1 ≤ ssthresh then 19     cwnd := cwnd + 1 20  else 21     mode := <i>congestion-avoidance</i> 22     cwnd := cwnd + 1/cwnd 23  flush-cwnd() </pre>
flush-cwnd()	<pre> 24  while cwnd &gt;  to-be-ack  do 25     <i>k</i> = send(to-be-sent) 26     remove(to-be-sent, <i>k</i>) 27     add(to-be-ack, <i>k</i>) 28     time(<i>k</i>) = time </pre>

Πίνακας 1: Ψευδοκώδικας για το TCP-PR

time	current time
time( <i>n</i> )	time at which time packet <i>n</i> was sent
cwnd( <i>n</i> )	congestion window at the time packet <i>n</i> was sent
is-in(list, <i>k</i> )	returns true if the packet <i>k</i> is in the list list
add(list, <i>k</i> )	add the packet <i>k</i> to the list list
remove(list, <i>k</i> )	remove the packet <i>k</i> from list list (if <i>k</i> is not in list do nothing)
list	number of elements in the list list
<i>k</i> = send(list)	send the packet in list list with smallest seq. number, returning the seq. number

Πίνακας 2: Σηματολογία του Πίνακα 1

### 5.3 ΑΚΡΑΙΕΣ ΑΠΩΛΕΙΕΣ

Όταν τα μισά (ή περισσότερα) πακέτα χάνονται μέσα σε ένα παράθυρο, το TCP-NewReno / SACK θα λήξει σε λειτουργία fastrecovery. Αυτό οφείλεται στο γεγονός ότι δε λήφθηκαν αρκετά ACKs, ώστε το παράθυρο συμφόρησης να ανοίξει και να επιτρέψει στον αποστολέα να εκτελέσει τις απαραίτητες αναμεταδόσεις. Τελικά προκύπτει η λήξη του χρονικού ορίου. Όταν συμβαίνει αυτό συνέχεια, τα πρωτόκολλα ξεκινούν μια εκθετική υποχώρηση του χρονικού ορίου, μέχρι τα πακέτα να είναι σε θέση να περάσουν. Η ‘σωστή’ συμπεριφορά του ελέγχου συμφόρησης σε ακραίες απώλειες είναι κάπως αμφιλεγόμενη και ίσως η πιο λογική προσέγγιση είναι να επιτραπεί στην εφαρμογή να αποφασίσει τι θα κάνει σε αυτή την περίπτωση. Ωστόσο, για να είναι και συμβατή με τις προηγούμενες εκδόσεις του TCP, προτείνεται μια έκδοση του TCP-PR που επαναφέρει το cwnd σε ένα και εκτελεί εκθετική υποχώρηση κάτω από ακραίες συνθήκες απώλειας.

Εντοπίζονται ακραίες απώλειες, μετρώντας τον αριθμό των πακέτων που χάνονται σε μια υπερφόρτωση. Αυτό μπορεί να γίνει, χρησιμοποιώντας έναν μετρητή cburst, που προστίθεται κάθε φορά που ένα πακέτο έχει αφαιρεθεί από τη λίστα memorize, εξαιτίας των απωλειών και επαναφέρεται στο μηδέν, όταν η λίστα αυτή αδειάσει. Υπενθυμίζεται ότι, η λίστα αυτή συνήθως διατηρείται άδεια, αλλά όταν υπάρξει μια απώλεια ‘απομνημονεύει’ τα πακέτα που ήταν σε εκκρεμότητα. Στη σκέψη του TCP-NewReno και TCP-Sack, τα πακέτα από αυτή τη λίστα που έχουν δηλωθεί ως απώλεια, δεν οδηγούν σε περαιτέρω μείωση κατά το ήμισυ του παράθυρου συμφόρησης. Για να εξομοιωθεί όσο το δυνατόν καλύτερα τι συμβαίνει με το TCP-NewReno και το Sack, όταν το cburst (και συνεπώς ο αριθμός των απωλειών σε μια υπερφόρτωση) υπερβαίνει το  $\frac{cwnd}{2+1}$ , μπορεί να γίνει επαναφορά του cwnd = 1 και να γίνει μετάβαση σε λειτουργία.

Επιπλέον, και για το fairness (δικαιοσύνη) όσον αφορά τις εφαρμογές του TCPNewReno / SACK που χρησιμοποιούν χρονόμετρα, αυξάνεται το  $mxrtt$  [36] ένα δευτερόλεπτο και καθυστερείται η αποστολή πακέτων από το  $mxrtt$ . Αν περαιτέρω (νέες) απώλειες προκύψουν, ενώ το  $cwnd = 1$ , αντί της διαίρεσης των  $cwnd$  από δύο, διπλασιάζεται το  $mxrtt$ , το οποίο εξομοιώνει την εκθετική οπισθοχώρηση [35].

#### **5.4 ΑΠΟΔΟΣΗΧΩΡΙΣ PACKET REORDERING: PERFORMANCE ΚΑΙ FAIRNESS**

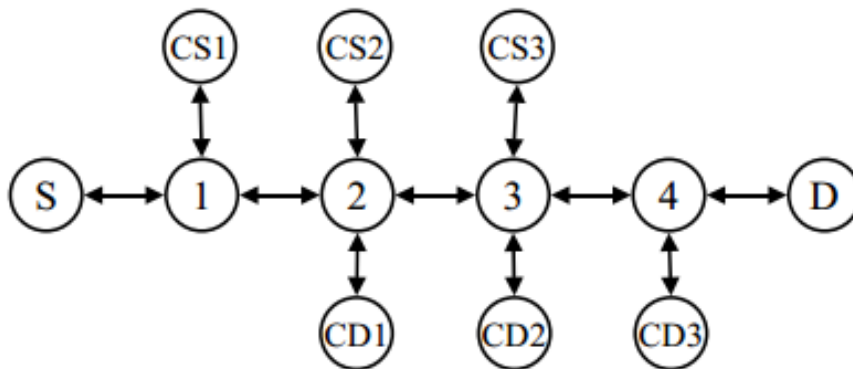
Δύο θέματα που προκύπτουν κατά την εξέταση του TCP-PR πάνω σε δίκτυα που δεν υπάρχει packetreordering: performanceandfairness. Το πρώτο ζήτημα είναι αν το TCP-PR εκτελεί το ίδιο όπως άλλες εφαρμογές του TCP υπό ‘φυσιολογικές’ συνθήκες, δηλαδή, δε γίνεται κανένα packetreordering. Συγκεκριμένα, για μια καθορισμένη τοπολογία και ένα κυκλοφοριακό υπόβαθρο, το TCP-PR μπορεί να επιτύχει παρόμοιο performance ως πρότυπο εφαρμογής του TCP. Η δεύτερη ανησυχία είναι αν το TCP-PR και οι καθορισμένες εφαρμογές TCP μπορούν να συνυπάρξουν το ίδιο. Σε κάποιο βαθμό, το ζήτημα του fairness περιλαμβάνει το ζήτημα αποδόσεων: αν το TCP-PR ανταγωνίζεται αρκετά έναντι των τυπικών εφαρμογών του TCP σε μια παραλλαγή συνθηκών του δικτύου, τότε φαίνεται λογικό ότι το TCP-PR και άλλες εφαρμογές του TCP είναι σε θέση να επιτύχουν παρόμοιο performance (και συνεπώς να εκτελέσουν όμοια) όταν εκτίθενται σε παρόμοιες συνθήκες του δικτύου. Έτσι, ενώ η ενότητα αυτή εστιάζει στο fairness, αντιμετωπίζει έμμεσα το ζήτημα των αποδόσεων.

Σε ένα πείραμα έγιναν εκτεταμένες προσομοιώσεις σε ns-2 για να δείξουν ότι, για ένα ευρύ φάσμα συνθηκών του δικτύου και τοπολογιών, το TCP-PR ισούται με το πρότυπο εφαρμογών TCP. Σε αυτό το κομμάτι, ένα δείγμα των αποτελεσμάτων

της προσομοίωσης παρουσιάζεται με επικεντρωμένη την προσοχή στη συμβατότητα με το πρωτόκολλο TCP- SACK σε πάνω από δύο τοπολογίες.

Η πρώτη τοπολογία που χρησιμοποιήθηκε είναι η τοπολογία χωρίς συγκεκριμένη διάταξη, η οποία είναι επίσης γνωστή ως μεμονωμένης-συμφόρησης (single-bottleneck) καθώς περιλαμβάνει μόνο ένα σύνδεσμο συμφόρησης. Ένας αριθμός μελετών προσομοίωσης χρησιμοποίησε την τοπολογία χωρίς συγκεκριμένη διάταξη για την αξιολόγηση της απόδοσης των πρωτοκόλλων δικτύου. Ένα πρόσφατο παράδειγμα είναι η σύγκριση μεταξύ της απόδοσης (συμπεριλαμβανομένης της ισοτιμίας) του TCP-Sack και την εφαρμογή του "TCP-friendly" τύπου .

Η δεύτερη τοπολογία που χρησιμοποιήθηκε είναι η τοπολογία parking-lot, η οποία όπως και η χωρίς διάταξη, έχει επίσης χρησιμοποιηθεί σε μια σειρά από πρόσφατες μελέτες απόδοσης των πρωτοκόλλων δικτύου. Το Parking-lot είναι μια γενίκευση της τοπολογίας χωρίς διάταξη, καθώς περιλαμβάνει πολλαπλές συνδέσεις συμφόρησης. Το Σχήμα 5.1 παρουσιάζεται η τοπολογία parking-lot που χρησιμοποίησαν, συμπεριλαμβανομένων των κόμβων πηγής και προορισμού για τις ροές διάρκειας TCP- SACK που χαρακτηρίζονται ως διαπεραστικές. Θα πρέπει να επισημανθεί ότι, οι προηγούμενες μελέτες που χρησιμοποίησαν την τοπολογία parking-lot περιέλαβαν τη διαπεραστική κυκλοφορία ροής ανάμεσα σε ζεύγη κόμβων CS1 → CD1, CS2 → CD2, και CS3 → CD3. Έχουν αυτά, καθώς και μεταξύ των CS1 → CD2, CS1 → CD3, και CS2 → CD3.



Σχήμα 5.1 Τοπολογία parking-lot

Ακολουθώντας την προσέγγιση της fairness της TCP-PR σε TCP-SACK κρίνεται από την προσομοίωση ο αριθμός ροών TCP-PR και TCP-SACK. Αυτές οι ροές έχουν μια κοινή πηγή και έναν κοινό προορισμό. Η σταθερή κατάσταση fairness μπορεί να προσδιοριστεί ποσοτικά με ένα μόνο αριθμό, τη μέση φυσιολογική απόδοση. Αν υπάρχουν  $n$  ροές, τότε η φυσιολογική απόδοση ροής του  $i$  είναι

$$T_i = \frac{x_i}{\frac{1}{n} \sum_{j=1}^n x_j} = n x_j$$

όπου η απόδοση,  $x_i$ , είναι το σύνολο των δεδομένων που αποστέλλονται κατά τη διάρκεια των τελευταίων 60 δευτερολέπτων της προσομοίωσης. Η μέση ομαλή απόδοση για ένα συγκεκριμένο πρωτόκολλο είναι η μέση τιμή του  $T_i$ , κατά μέσο όρο σε όλες τις ροές αυτού του πρωτόκολλου. Να σημειωθεί ότι αν  $T_i = 1$ , τότε η ροή  $i$  έχει λάβει την μέση απόδοση. Ομοίως, εάν η μέση ομαλή απόδοση είναι ένα, τότε οι δύο εφαρμογές έλαβαν την ίδια μέση απόδοση.

Το Σχήμα 5.2 δείχνει το φυσιολογικό throughput και το μέσο φυσιολογικό throughput για διάφορους αριθμούς ροών TCP-PR και TCP-SACK. Τα αποτελέσματα από τις τοπολογίες χωρίς διάταξη και τις τοπολογίες parking-

Ιοιφαίνονται στο δεξί και στο αριστερό γράφημα, αντίστοιχα. Σε αυτά τα πειράματα, το TCP-PR α και β σταθεροποιήθηκαν σε 0.995 και 3.0, αντίστοιχα. Από τις γραφικές παραστάσεις, είναι ξεκάθαρο ότι οι δύο εκδόσεις του TCP-PR και TCP-SACK ανταγωνίζονται ισότιμα σε ένα ευρύ φάσμα των συνθηκών κυκλοφορίας και κατά συνέπεια παρουσιάζουν παρόμοια απόδοση.

Ενώ η μέση φυσιολογική throughput περιγράφει τη μέση συμπεριφορά όλων των ροών, ο συντελεστής διακύμανσης περιγράφει τη διακύμανση της απόδοσης. Συγκεκριμένα, αν είναι το I είναι το σύνολο των ροών ενός συγκεκριμένου πρωτοκόλλου. Ο συντελεστής διακύμανσης είναι

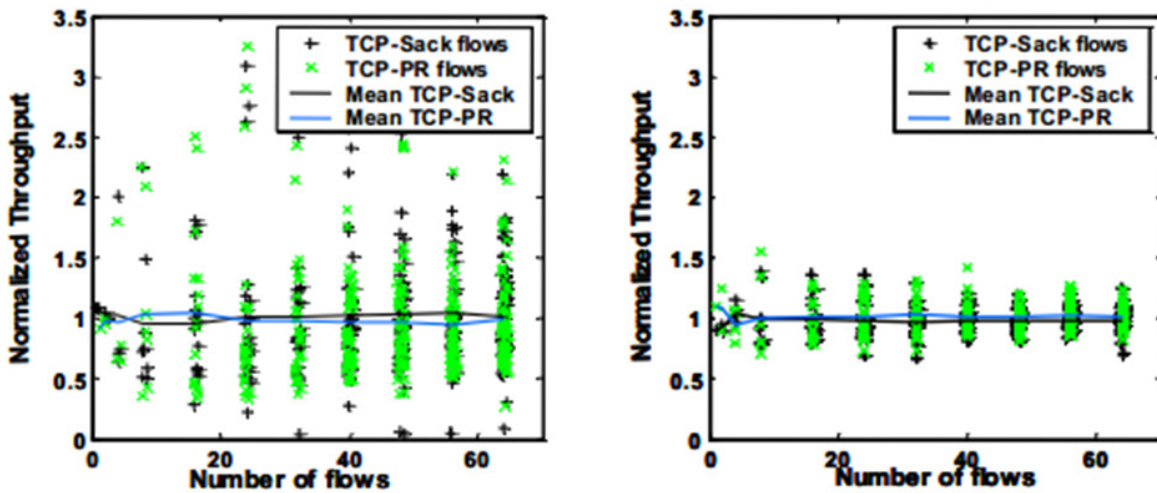
$$CoV = \frac{1}{\sum_{i \in I} T_i} \sqrt{\sum_{i \in I} (T_i - \frac{1}{|I|} \sum_{i \in I} T_i)^2}$$

όπου |I| είναι ο αριθμός των στοιχείων στο σύνολο I. Το σχήμα 5.3 δείχνει το συντελεστή διακύμανσης για δέκα προσομοιώσεις, καθώς και το μέσο συντελεστή μεταβολής για το σύνολο της προσομοίωσης. Από τα σχήματα 5.2 και 5.3, βγαίνει το συμπέρασμα ότι η μέση τιμή και η διακύμανση της throughputs για το TCP-PR και TCP-SACK είναι παρόμοια.

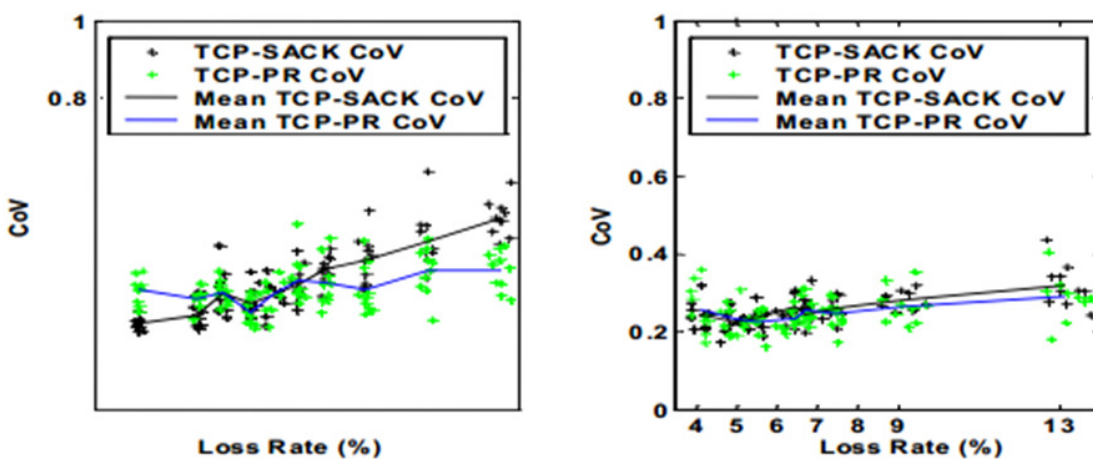
Παραδόξως, το fairness διατηρείται για ένα ευρύ φάσμα α και β. Το σχήμα 5.4 δείχνει τη μέση φυσιολογική απόδοση του TCPSACK για διαφορετικές τιμές α και β. Για αυτές τις προσομοιώσεις, ο αριθμός των ροών διατηρήθηκε σταθερός για 64 ροές συνολικά (32 TCP-SACK και 32 TCP-PR ροές). Σημειώνεται ότι για β = 1, το TCP-SACK παρουσιάζει την καλύτερη απόδοση. Ωστόσο, για το β μεγαλύτερο του 1, και οι δύο υλοποιήσεις επέδειξαν σχεδόν ταυτόσημες αποδόσεις. Ένας μεγάλος αριθμός προσομοιώσεων δείχνει ότι αυτά τα αποτελέσματα είναι σταθερά για διαφορετικά επίπεδα υποβάθρου κυκλοφορίας και για διαφορετικές τοπολογίες. Έχει παρατηρηθεί ότι ακόμα και σε περιπτώσεις όπου η κυκλοφορία διέρχεται, προκαλεί ακραίες συνθήκες απώλειας (πάνω από 15% πιθανότητα απώλειας), το TCP-SACK παίρνει μόνο μέχρι 20% περισσότερη απόδοση όταν β



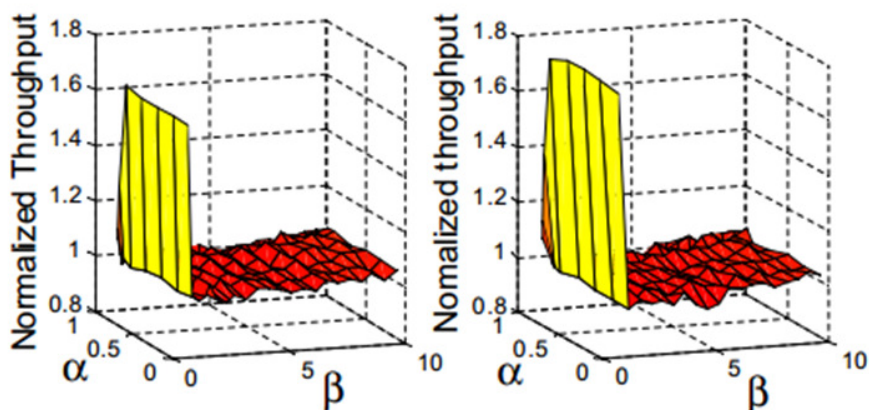
= 10, ενώ τα διερχόμενα είναι τα ίδια για  $1 < \beta < 5$ . Τέτοια ακραία απώλεια δεν είναι ιδιαίτερα σημαντική, δεδομένου ότι η απόδοση του TCP είναι πολύ χαμηλή, όταν η πιθανότητα απώλειας είναι πολύ μεγάλη.



Σχήμα 5.2 Φυσιολογικό throughput και μέσο φυσιολογικό throughput για διάφορους αριθμούς ροών TCP-PR και TCP-SACK



Σχήμα 5.3 Συντελεστή διακύμανσης για δέκα προσομοιώσεις και το μέσο συντελεστή μεταβολής για το σύνολο της προσομοίωσης.



Σχήμα 5.4 Μέση φυσιολογική απόδοση του TCPSACK για διαφορετικές τιμές  $\alpha$  και  $\beta$ . Για αυτές τις προσομοιώσεις, ο αριθμός των ροών διατηρήθηκε σταθερός για 64 ροές συνολικά (32 TCP- SACK και 32 TCP-PR ροές)

Αυτά τα αποτελέσματα υπό κανονικές συνθήκες κυκλοφορίας δε δίνουν τόσα πολλά στοιχεία για το πόσο ιδιαίτερο είναι το TCP-PR, αλλά μάλλον πιστοποιούν την ευρωστία του συστήματος AIMD. Το σημαντικό χαρακτηριστικό του συστήματος AIMD είναι ότι αν δύο ροές ανιχνεύσουν απώλειες με τον ίδιο ρυθμό, τότε τα παράθυρα συμφόρησης θα συγκλίνουν. Το TCP ρέει πάνω από μια τοπολογία, χωρίς διάταξη και συγκλίνει αρκετά γρήγορα, προς το ίδιο εύρος ζώνης. Αν και αυτές οι αποδείξεις βασίζονται σε πρωτόκολλα τα οποία είναι ίδια, επίσης, επισημαίνουν την εγγενή σταθερότητα του συστήματος AIMD που μαρτυρείται από τα αποτελέσματα της προσομοίωσης που παρουσιάζονται εδώ.

## 5.5 ΑΠΟΔΟΣΗ ΥΠΟ ΤΗΝ ΑΝΑΔΙΑΤΑΞΗ ΠΑΚΕΤΩΝ ΣΕ ΣΥΓΚΡΙΣΗ ΜΕ ΑΛΛΕΣ ΜΕΘΟΔΟΥΣ

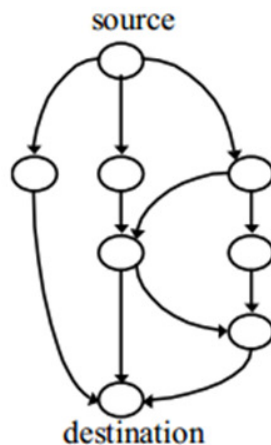
Εκτελούνται εκτεταμένες προσομοιώσεις χρησιμοποιώντας ns-2 για τη σύγκριση της απόδοσης των μεθόδων αυτών για την αντιμετώπιση του συνεχούς packetreordering λόγω της δρομολόγησης πολλαπλών διαδρομών. Η τοπολογία για τη σύγκριση αυτή φαίνεται στο Σχήμα 5.5. Πραγματοποιήθηκαν δυο τάξεων προσομοιώσεις, η πρώτη καθόρισε την καθυστέρηση διάδοσης για κάθε σύνδεσμο σε 10ms, ενώ η δεύτερη καθόρισε την καθυστέρηση διάδοσης σε 60ms.

Πολλές στρατηγικές δρομολόγησης πολλαπλών διαδρομών είναι δυνατές κατά τη διάρκεια αυτής της τοπολογίας. Έχει αναπτυχθεί ένα σύνολο στρατηγικών που βάζει παραμέτρους από μια ενιαία μεταβλητή  $\epsilon$ . Αυτή η παράμετρος ελέγχει το βαθμό στον οποίο λαμβάνεται υπόψη η καθυστέρηση όταν η δρομολόγηση είναι σχεδιασμένη. Ειδικότερα, όταν  $\epsilon = \infty$ , όταν η καθυστέρηση είναι σε ιδιαίτερα μειονεκτική θέση, χρησιμοποιείται η συντομότερη διαδρομή, ενώ όταν  $\epsilon = 0$ , η καθυστέρηση δεν είναι σημαντική και χρησιμοποιείται για την πλήρη δρομολόγηση πολλαπλών διαδρομών, που οδηγεί σε όλες τις ανεξάρτητες διαδρομές από την πηγή στον προορισμό με ίσες πιθανότητες. Οι ενδιάμεσες τιμές του  $\epsilon$  αντιστοιχούν σε συμβιβασμούς μεταξύ των δύο ακραίων περιπτώσεων.

Για μια σταθερή στρατηγική δρομολόγησης (ένα σταθερό  $\epsilon$ ), το TCP-PR και το TCP συζητήθηκε με διάφορα dupthresh\* συστήματα αποζημίωσης όπου κάθε ένα δοκιμάστηκε ξεχωριστά, και ως εκ τούτου, μόνο μια ροή ήταν ενεργή τη φορά [37]. Επιπλέον, για αυτές τις προσομοιώσεις, δεν υπήρχε υπόβαθρο κίνησης. Το σκεπτικό πίσω από αυτές τις επιλογές είναι ότι ο στόχος αυτής της ενότητας είναι, ότι αντί να συγκρίνουν το πώς οι διαφορετικές εκδόσεις του TCP αλληλεπιδρούν μεταξύ τους, ερευνούν πώς οι διαφορετικές μέθοδοι είναι σε θέση να αντιμετωπίσουν το συνεχές packetreordering.

Το σχήμα 5.5 δείχνει την απόδοση για διάφορες τιμές του  $\epsilon$ . Αυτές οι προσομοιώσεις δείχνουν ότι για  $\epsilon = 0$  (πλήρης δρομολόγηση πολλαπλών διαδρομών), η απόδοση από τα περισσότερα από τα άλλα πρωτόκολλα μειώνεται

δραστικά. Για  $\epsilon = 500$  (δρομολόγηση μονής διαδρομής), όλες οι μέθοδοι επιτυγχάνουν την ίδια δρομολόγηση. Η εξαίρεση είναι η χρονική καθυστέρηση ταχείας αποκατάστασης (TD-FR), η οποία εξακολουθεί να επιτυγχάνει μια λογική απόδοση για μικρές τιμές του  $\epsilon$ , αν η καθυστέρηση διάδοσης είναι μικρή. Ωστόσο, καθώς η καθυστέρηση διάδοσης είναι αυξημένη, η απόδοση μειώνεται. Σε κάποιο βαθμό η μείωση αυτή στην απόδοση οφείλεται στην αύξηση του χρόνου μετ' επιστροφής. (Να σημειωθεί ότι στο  $\epsilon = 500$ , όλες οι δρομολογήσεις είναι μικρότερες στα δεξιά παρά στα αριστερά). Ωστόσο, στο  $\epsilon = 0$ , το TD-FR έχει πολύ μεγάλη απώλεια στην απόδοση όταν η καθυστέρηση διάδοσης είναι αυξημένη. Ο λόγος για αυτή την πτώση στην απόδοση είναι ότι το TD-FR κάνει χρήση τόσο του `dupthresh` και όσο και των χρονομέτρων. Συγκεκριμένα, το `dupthresh` είναι μεγαλύτερο όταν ο χρόνος μετ' επιστροφής είναι μεγαλύτερος. Η αύξηση του `dupthresh` μπορεί να οδηγήσει σε έκρηξη.



Σχήμα 5.5 Τοπολογία για τη σύγκριση των υλοποιήσεων TCP.

## ΚΕΦΑΛΑΙΟ 6

Στο κεφάλαιο αυτό παρουσιάζονται συμπεράσματα, που προκύπτουν από τις εργασίες που παρουσιάστηκαν. Κύριος σκοπός ήταν να δωθούν ιδέες και κίνητρα για τη χρήση του δικτύου στο μέγιστο. Στις μελέτες παρατηρείται πως, η μεγαλύτερη συμφόρηση και συνεπώς το packet reordering, παρουσιάζεται όταν το cwnd μεγαλώνει.

### 6.1 ΣΥΜΠΕΡΑΣΜΑΤΑ ΣΧΕΤΙΚΑ ΜΕ ΤΗΝ ΑΠΟΔΟΣΗ ΤΩΝ ΠΑΡΑΛΛΑΓΩΝ TCP

Όπως αναλύθηκε και στο κεφάλαιο 3, το TCP Eifel [26] σχεδιάστηκε για να βελτιώσει την απόδοση του TCP σε όλες τις ασύρματες συνδέσεις. Με τον τρόπο αυτό μπορεί να επαναφέρει την κατάσταση ελέγχου της συμφόρησης με τέτοιο τρόπο, ώστε οι εκρήξεις πακέτων να αποφεύγονται, προκαλώντας στον αποστολέα να συνεχίσει να μειώνει το παράθυρο συμφόρησής του. Το TCP Eifel χρησιμοποιεί τη λύση εντοπισμού και ανάκτησης από ψευδείς αναμεταδόσεις και ψευδή χρονικά όρια. Για την ανίχνευση, το TCP Eifel χρησιμοποιεί χρονοσφραγίδες, που είναι ενσωματωμένες σε κάθε τμήμα δεδομένων και κάθε τμήμα του ACK. Όταν έχει βρεθεί ψευδής αναμετάδοση ή ψευδής λήξη χρονικού ορίου, το TCP Eifel αποκαθιστά το cwnd και το ssthresh στις προηγούμενες τιμές τους.

Επίσης στο κεφάλαιο 3 συζητήθηκε το TCP-LPC. Στο TCP-LPC, από την μεριά του αποστολέα, ορίστηκε στατικά το dupthresh, με βάση τον αριθμό των πιθανών διαδρομών μεταξύ του αποστολέα και του δέκτη. Από την άλλη ο δέκτης

περιμένει ένα μικρό χρονικό διάστημα πριν μεταδώσει ένα DUPACK. Αυτό είναι ως επί το πλείστον ισοδύναμο με τον αποστολέα που είναι σε αναμονή πριν από την ενεργοποίηση του fast retransmit, με εξαίρεση ότι δε θα εφαρμοσθεί περιορισμένη μετάδοση.

Αναφερόμενοι στο TCP Eifel, αναλύθηκε και το TCP-BA, όπου βελτιώνει άμεσα τον προαναφερόμενό του. Αυτό γίνεται με το να προσαρμόζει δυναμικά το dupthresh στον αριθμό των DUPACKs, το οποίο απαιτείται για την ενεργοποίηση του fast retransmit. Επιπλέον, αναμένεται ένα σύντομο χρονικό διάστημα μετά τη λήψη του τρίτου DUPACK πριν προκληθεί ένα fast retransmit. Δυστυχώς, αυτό απαιτεί ένα επιπλέον χρονόμετρο.

Μια μικρή διαφορά μεταξύ του TCP-BA και του TCP Eifel είναι ότι το TCP-BA χρησιμοποιεί DSACK αντί για TCP χρονοσφραγίδες για την ανίχνευση ψευδών αναμεταδόσεων.

Επέκταση του TCP-BA είναι το RR-TCP, το οποίο έχει τη δυνατότητα να αυξομειώνει το dupthresh.

Αυτό είναι σημαντικό, δεδομένου ότι το TCP-BA θα συνεχίσει να αυξάνει το dupthresh, μέχρι που τελικά θα υπάρχουν αρκετά DUPACKs να φτάσουν και θα γίνει λήξη του χρονικού ορίου, στο σημείο όπου το TCP-BA θα επαναφέρει το dupthresh στο 3.

Το RR-TCP εκτείνει επίσης την περιορισμένη μετάδοση, για να στείλει μέχρι ένα ολόκληρο παράθυρο συμφόρησης δεδομένων, όταν είναι στην κατάσταση fast retransmit. Αυτό είναι πιο επιθετικό από το TCP-BA, το οποίο μεταδίδει μόνο ένα νέο τμήμα του TCP για κάθε δύο DUPACKs που θα ληφθούν.

Το TCP-NCR είναι μια σύνθεση του TCP-BA, του RR-TCP και του TCP-DCR. Ουσιαστικά, το TCP-NCR εφαρμόζει μόνο δυναμικά μεταβαλλόμενο dupthresh. Ωστόσο, το dupthresh έχει οριστεί να είναι ίσο με την τιμή των δεδομένων ενός παράθυρου συμφόρησης, δηλαδή ένα RTT. Οι μελετητές προτείνουν ότι το TCP-NCR μπορεί να συνδυαστεί με οποιοδήποτε άλλο TCP που διαθέτει τη δυνατότητα να ανιχνεύσει μια αναμετάδοση που είναι ψευδής και να αποκαταστήσει το αρχικό cwnd και ssthresh.

Μία άλλη παραλλαγή του TCP που αναφέρθηκε στο κεφάλαιο 3 είναι το TCP Westwood το οποίο σχεδιάστηκε για τα ασύρματα δίκτυα. Αξιοσημείωτη ιδιότητα του TCP Westwood είναι η αγνόηση των ψευδών αναμεταδόσεων και ανακτήσεων του δικτύου. Ο αποστολέας του TCP Westwood υπολογίζει το πραγματικό εύρος ζώνης από άκρο σε άκρο με δειγματοληψία των ACKs. Εάν ό στιγμιαίο εύρος ζώνης είναι μεγάλο, τότε το TCP Westwood θεωρεί ότι το fast retransmit δεν υποδεικνύει συμφόρηση, και έτσι μειώνει το παράθυρο συμφόρησης κατά ένα μικρό ποσοστό. Η λήξη του χρονικού ορίου θα έδειχνε σοβαρά επίπεδα συμφόρησης, αρκετά για να σταματήσει τη ροή των ACKs.

Οι προσομοιώσεις του TCP Westwood δείχνουν ότι αποδίδει καλά σε ενσύρματα δίκτυα, πολλαπλών διαδρομών.

Το TCP-PR είναι παρόμοιο με το TCP Westwood στο γεγονός ότι αγνοεί τα DUPACKs που θα προκαλέσουν fast retransmit. Αντίθετα, το TCP-PR σφραγίζει χρονικά κάθε τμήμα, και τα ACKs που δεν φτάνουν πριν λήξει το χρονικό όριο, αναμεταδίδονται. Αυτό είναι ένα πολύ απλό σχέδιο, αλλά θα μπορούσε να υποφέρει από προβλήματα υλοποίησης, ιδιαίτερα στις συνδέσεις υψηλής ταχύτητας.

Το TCP/NC σχεδιάστηκε για να ξεπεράσει απώλειες συνδέσεων σε ασύρματα δίκτυα. Θα μπορούσε επίσης να είναι κατάλληλο για την επίλυση του προβλήματος του packet reordering σε ενσύρματα δίκτυα πολλαπλών διαδρομών. Το TCP / NC μπορεί να χρησιμοποιηθεί με οποιαδήποτε υλοποίηση του TCP. Για να λειτουργήσει καλά στα ενσύρματα δίκτυα μπορεί να χρειαστεί να τροποποιηθεί σε μεγάλο βαθμό, δεδομένου ότι το TCP/NC μεταδίδει επιπλέον πακέτα, για να καλύψει την απώλεια πακέτων, τα οποία θα μπορούσαν να οδηγήσουν σε συμφόρηση.

Η χρήση αλγορίθμων για τον έλεγχο συμφόρησης σε ένα δίκτυο, που παρουσιάζει φαινόμενα συμφόρησης, δείχνει τη σημαντικότητα που έχει ο έλεγχος συμφόρησης στην μετάδοση μηνυμάτων και στη κινητικότητα του δικτύου. Όσο εξελίσσονται τα δίκτυα, είναι αναγκαία η εύρεση νέων ιδεών και τεχνικών για τον έλεγχο της συμφόρησης και συνεπώς του packet reordering. Συνεπώς, είναι σημαντικό να καλυτερεύσει το TCP μελλονικά, έτσι ώστε να διαχειριστεί στο μέγιστο την κυκλοφορία και τη συμφόρηση.

Ενώ ο "περιορισμένος αλγόριθμος μεταφοράς (limitedtransmitalgorithm)" επιχειρεί να μειώσει την συμφόρηση, Η συμφόρηση αυτή παραμένει πρόβλημα για το TD-FR σε συνδέσεις με μεγάλη καθυστέρηση. Αυτές οι προσομοιώσεις αποδεικνύουν την αποτελεσματικότητα των απωλειών των χρονομετρημένων πακέτων ανίχνευσης TCP- PR. Επιβεβαιώνεται ότι τα διπλότυπα ACKs δηλώνουν την απώλεια πακέτων σε μια μονόδρομη δρομολόγηση αλλά η εμφάνισή τους μεταφέρει ελάχιστες πληροφορίες, κατά τη χρήση πολλαπλής δρομολόγησης.

Πρόσφατα προτάθηκε μια άλλη μέθοδος για την προσαρμογή του dupthresh [38]. Δεδομένου ότι η προσομοίωση της εφαρμογής αυτής της μεθόδου δεν είναι ακόμα διαθέσιμη, δεν περιλαμβάνεται σε αυτή τη σύγκριση.



## Βιβλιογραφία

[1] Tanenbaum, A. (2012). *Δίκτυα Υπολογιστών (5<sup>η</sup> έκδοση)*. Αθήνα: Εκδόσεις Κλειδάριθμος.

[2] ArjunaSathiaseelanandTomaszRadzik. *Improving the performance of TCP in the case of packet reordering*. Department of Computer Science, King's College London, Strand, London

[3] M. Allman, V. Paxson, and W. Stevens, *TCP Congestion Control*, IETF RFC 2581, Apr. 1999.

[4] V. Paxson and M. Allman, *Computing TCP's Retransmission Timer*, IETF RFC 2988, Nov. 2000.

[5] S. Floyd and K. Fall, "Promoting the Use of End-to-End Congestion Control in the Internet," *IEEE/ACM Trans. Networking*, vol. 7, no. 4, pp. 458-472, Aug. 1999.

[6] J. Bennett, C. Partridge, and N. Shectman, "Packet Reordering is Not Pathological Network Behavior," *IEEE/ACM Trans. Networking*, vol. 7, no. 6, pp. 789-798, Dec. 1999.

[7] P. Karn and C. Partridge, "Improving Round-Trip Time Estimates in Reliable Transport Protocols," *ACM Trans. Computer Systems*, vol. 9, no. 4, pp. 364-373, Nov. 1991.

- [8] M. Zhang, B. Karp, S. Floyd, and L. Peterson, "RR-TCP: A Reordering-Robust TCP with DSACK," *Proc. IEEE Int'l Conf. Network Protocols (ICNP '03)*, pp. 95-106, Nov. 2003.
- [9] V. Jacobson, R. Braden, and D. Borman, *TCP Extensions for High Performance*, IETF RFC 1323, May 1992.
- [10] F. Wang and Y. Zhang, "Improving TCP Performance over Mobile Ad-Hoc Networks with Out-of-Order Detection and Response," *Proc. ACM MOBIHOC '02*, pp. 217-225, June 2002
- [11] S. Floyd, J. Mahdavi, M. Mathis, and M. Podolsky, *An Extension to the Selective Acknowledgement (SACK) Option for TCP*, IETF RFC 2883, July 2000.
- [12] Y. Lee, I. Park, and Y. Choi, "Improving TCP Performance in Multipath Packet Forwarding Networks," *J. Comm. and Networks*, vol. 4, no. 2, pp. 148-157, June 2002.
- [13] M. Allman, H. Balakrishnan, and S. Floyd, *Enhancing TCP's Loss Recovery Using Limited Transmit*, IETF RFC 3042, Network Working Group, Jan. 2001.
- [14] E. Blanton and M. Allman, "On Making TCP More Robust to Packet Reordering," *ACM SIGCOMM Computer Comm. Rev.*, vol. 32, no. 1, pp. 20-30, Jan. 2002.
- [15] K.-C. Leung and C. Ma, "Enhancing TCP Performance to Persistent Packet Reordering," *J. Comm. and Networks*, vol. 7, no. 3, pp. 385-393, Sept. 2005.
- [16] A. Sathiaseelan and T. Radzik, "Improving the Performance of TCP in the Case of Packet Reordering," *Lecture Notes in Computer Science*, vol. 3079, pp. 63-73, June/July 2004.

[17]M. Laor and L. Gendel, “The Effect of Packet Reordering in a Backbone Link on Application Throughput,” *IEEE Network*, vol. 16, no. 5, pp. 28-36, Sept./Oct. 2002.

[18]V. Paxson, “End-to-End Internet Packet Dynamics,” *IEEE/ACM Trans. Networking*, vol. 7, no. 3, pp. 277-292, June 1999.

[19]S. Bohacek, J.P. Hespanha, J. Lee, C. Lim, and K. Obraczka, “A New TCP for Persistent Packet Reordering,” *IEEE/ACM Trans. Networking*, vol. 14, no. 2, pp. 369-382, Apr. 2006.

[20]Jeffrey C. Mogul. *Observing TCP Dynamics in Real Networks*. In *Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, 1992

[21]Vern Paxson. *End-to-End Internet Packet Dynamics*. *IEEE/ACM Transactions on Networking*, 7(3):277–292, June 1999.

[22]Gianluca Iannaccone, Sharad Jaiswal, and Christophe Diot. *Packet Reordering Inside the Sprint Backbone*. Technical report, Technical Report TR01-ATL-062917, Sprint ATL, 2001.

[23]Michael Laor and Lior Gendel. *The Effect of Packet Reordering in a Backbone Link on Application Throughput*. *IEEE Network*, 16(5):28–36, Sep/Oct 2002

[24]Jon C. R. Bennett, Craig Partridge, and Nicholas Shectman. *Packet Reordering is Not Pathological Network Behavior*. *IEEE/ACM Transactions on Networking*, 7(6):789–798, December 1999.

[25] Tracey Ho and Desmond S. Lun. *Network Coding: an Introduction*. Cambridge University Press, 2008

[26] Reiner Ludwig and Randy H. Katz. *The Eifel Algorithm: Making TCP Robust Against Spurious Retransmissions*. *ACM SIGCOMM Computer Communication Review*, 2000.

- [27] M. Mathis, J. Mahdavi, Sally Floyd, and M. Podolsky. *RFC 2883: An Extension to the Selective Acknowledgment (SACK) Option for TCP*, July 2000.
- [28] Jad Naous, David Erickson, G. Adam Covington, Guido Appenzeller, and Nick McKeown. *Implementing an OpenFlow Switch on the NetFPGA Platform*. In *ANCS '08: Proceedings of the 4<sup>th</sup> ACM/IEEE Symposium on Architectures for Networking and Communications Systems*, pages 1–9, 2008.
- [29] G. M. Allman, V. Paxson and W. Stevens, "TCP Congestion Control", *RFC 2581*, Ap
- [30] J. Mogul, "Observing TCP dynamics in real networks". In *Proceedings of ACM SIGCOMM*, Baltimore, MD, August 1992
- [31] M. Zhang, B. Karp, S. Floyd, and L. Peterson, "RR-TCP: A reordering robust TCP with DSACK". In *Proceedings of IEEE ICNP*, Atlanta, Georgia, November 2003.
- [32] G. Neglia, V. Falletta and G. Bianchi, "Is TCP packet reordering always harmful?", In *Proceeding of MASCOTS*, October 2004.
- [33] L. Gharai, C. Perkins, and T. Lehman, "Packet reordering, high speed networks and transport protocol performance", In *Proceeding of the 13<sup>th</sup> ICCCN*, Chicago, IL, October 2004, pp. 73- 78
- [34] S. Bhandarkar, A. L. N. Reddy, M. Allman, E. Blanton, "Improving the Robustness of TCP to Non-Congestion Events", *RFC 4653*, August 2006
- [35] S. Bohacek, J. P. Hespanha, J. Lee, C. Lim, and K. Obraczka. *TCP-PR: TCP for persistent packet reordering. Extended version. Technical report*, University of California, Santa Barbara, May 2003
- [36] M. Allman and V. Paxson. *Computing TCP's retransmission timer*. *RFC 2988*, page 13, Nov. 2000
- [37] E. Blanton and M. Allman. *On making TCP more robust to packet reordering*. *ACM Computer Communications Review*, 32, 2002

[38] N. Zhang, B. Karp, S. Floyd, and L. Peterson. *RR-TCP: $\beta$ A reordering-robust TCP with DSACK*. Technical Report TR-02-006, ICSI, Berkeley, CA, July 2002.