ΠΑΝΕΠΙΣΤΗΜΙΟ
ΙΩΑΝΝΙΝΩΝ

ΣΧΟΛΗ ΠΛΗΡΟΦΟΡΙΚΗΣ & ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ & ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΠΜΣ ΠΛΗΡΟΦΟΡΙΚΗΣ & ΔΙΚΤΥΩΝ

ΜΕΤΑΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΜΕΛΕΤΗ ΤΟΥ ΦΑΙΝΟΜΕΝΟΥ «ΚΑΤΑΡΡΕΥΣΗΣ ΤΟΥ ΔΙΚΤΥΟΥ»ΣΕ ΔΙΚΤΥΑ IP

Πηνελόπη Τσοχαντάρη

Επιβλέπουσα: Σπυριδούλα Μαργαρίτη

ΕΔΙΠ, Α

Τόπος έκδοσης, Μήνας, Έτος ολοκλήρωσης

STUDY OF CONGESTION COLLAPSE IN IP NETWORKS

Εγκρίθηκε από τριμελή εξεταστική επιτροπή

Τόπος, Ημερομηνία

ΕΠΙΤΡΟΠΗ ΑΞΙΟΛΟΓΗΣΗΣ

1. Επιβλέπουσα καθηγήτρια

   Σπυριδούλα Μαργαρίτη

   ΕΔΙΠ, Α

2. Μέλος επιτροπής

   Βαρτζιώτης Φώτιος,

   Επίκουρος καθηγητής

3. Μέλος επιτροπής

   Ελευθέριος Στεργίου,

   Αναπληρωτής καθηγητής

Δήλωση μη λογοκλοπής

Δηλώνω υπεύθυνα και γνωρίζοντας τις κυρώσεις του Ν. 2121/1993 περί Πνευματικής Ιδιοκτησίας, ότι η παρούσα μεταπτυχιακή εργασία είναι εκ ολοκλήρου αποτέλεσμα δικής μου ερευνητικής εργασίας, δεν αποτελεί προϊόν αντιγραφής ούτε προέρχεται από ανάθεση σε τρίτους. Όλες οι πηγές που χρησιμοποιήθηκαν (κάθε είδους, μορφής και προέλευσης) για τη συγγραφή της περιλαμβάνονται στη βιβλιογραφία.

Τσοχαντάρη Πηνελόπη

ΕΥΧΑΡΙΣΤΙΕΣ

Η παρούσα διπλωματική εργασία εκπονήθηκε στα πλαίσια του μεταπτυχιακού προγράμματος «Π.Μ.Σ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΔΙΚΤΥΩΝ» της Σχολής Πληροφορικής και Τηλεπικοινωνιών του Πανεπιστημίου Ιωαννίνων. Με την ολοκλήρωση της παρούσας διπλωματικής εργασίας, θα ήθελα να ευχαριστήσω την καθηγήτρια μου κα. Σπυριδούλα Μαργαρίτη, για την επιστημονική και συμβουλευτική καθοδήγηση, που μου προσέφερε σε όλα τα στάδια εκπόνησης της εργασίας. Επίσης, θα ήθελα να ευχαριστήσω την οικογένεια μου, για την υποστήριξη και την συμπαράσταση τους σε όλη αυτή την προσπάθεια.

ΠΕΡΙΛΗΨΗ

Τα τελευταία χρόνια, η ραγδαία εξάπλωση της χρήσης των δικτύων οδήγησε στη συχνότερη εμφάνιση της συμφόρησης. Η συμφόρηση, στο πλαίσιο των δικτύων, είναι ένας όρος που περιγράφει μια κατάσταση δικτύου όπου η ζήτηση για έναν (ή περισσότερους) πόρους του δικτύου είναι μεγαλύτερη από αυτή που μπορεί να διαχειριστεί. Τυπικά, η χρήση ενός πόρου, ή με άλλα λόγια ο φόρτος που μπορεί να εξυπηρετεί είναι μέχρι τη μέγιστη χωρητικότητά του. Κάθε αίτημα πρόσθετης ζήτησης είτε μπλοκάρεται είτε αναγκάζεται να περιμένει μέχρι ο πόρος να γίνει διαθέσιμος κάποια στιγμή στο μέλλον. Όταν συμβαίνει αυτό, ο πόρος λέγεται ότι παρουσιάζει συμφόρηση. Σε ένα δίκτυο με συμφόρηση, τα δεδομένα που είναι αδύνατον να φτάσουν στην τελική τους πορεία χωρίς να υπάρξουν καθυστερήσεις και απώλειες, ενώ ο χρόνος απόκρισης επιβραδύνεται και μειώνεται η απόδοση δικτύου. Στην εργασία αυτή μελετούμε το πρόβλημα της συμφόρησης, τους παράγοντες από όπου προέρχεται και παρουσιάζουμε τους τρόπους για την αποφυγή και την αντιμετώπισης της. Στην προσπάθεια αυτή, θα εξετάσουμε διάφορους αλγόριθμους ελέγχου συμφόρησης ακολουθώντας διαφορετικές προσεγγίσεις, μέσω προσομοίωσης ώστε να καταλήξουμε στον πιο αποδοτικό και στον πιο δραστικό μηχανισμό για την μείωσης της συμφόρησης. Τέλος παρουσιάζουμε τα αποτελέσματα της προσομοίωσης με στόχο την κατανόηση της λειτουργίας των πρωτοκόλλων αντιμετώπισης της συμφόρησης και τη λήψη κατάλληλων μέτρων για την αντιμετώπιση της. Η επιλογή των μέτρων κατά της συμφόρησης εξαρτάται από τον τύπο του δικτύου, το είδος της συμφόρησης και τις ανάγκες των χρηστών. Συχνά, η καλύτερη λύση είναι μια συνδυαστική προσέγγιση που συνδυάζει διάφορες τεχνικές και προσεγγίσεις για την αντιμετώπιση της συμφόρησης.

Λέξεις-κλειδιά: Συμφόριση, tcp, bbr, cubic

ABSTRACT

In recent years, the rapid spread of network usage has led to a more frequent occurrence of congestion. Congestion, in the context of networks, is a term that describes a network state where a node or a network has so much data that it is impossible for it to reach its final destination without delays in the queue and data and packet loss. In a congested network, the response time slows down with reduced network performance. Congestion typically appears in networks when the traffic volume exceeds the network or specific connection capacity. In this paper, we will focus on the congestion problem, the factors from which it originates, and we will try to find ways to avoid and address it. In this effort, we will examine various congestion control algorithms following different approaches, through simulation, in order to determine the most efficient and effective method for reducing congestion. Finally, we present the results of the simulation with the goal of understanding the functioning of congestion control protocols and taking appropriate measures for dealing with it. The choice of congestion control measures depends on the type of network, the type of congestion, and the user's needs. Often, the best solution is a combined approach that merges different techniques and strategies to address congestion.

Keywords: Congestion, tcp, bbr, cubic

Περιεχόμενα

Πίνακας Εικόνων

1 Εισαγωγή

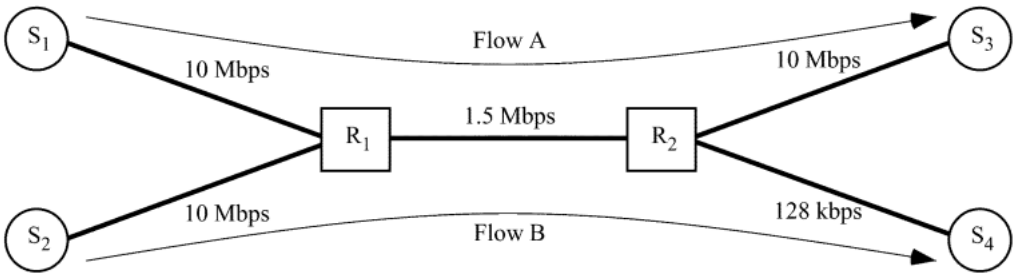
1.1 Γενικά

Η ραγδαία ανάπτυξη των τεχνολογιών Πληροφορικής και δικτύων διαμόρφωσε ένα διαδικτυακό υποσύστημα, το Διαδίκτυο, για την μεταφορά τεράστιου όγκου δεδομένων, σύμφωνα με τις απαιτήσεις των χρηστών. Σήμερα, το Διαδίκτυο συνδέει εκατομμύρια χρήστες και συσκευές και υποστηρίζει πλήθος υπηρεσιών που έχουν απαιτήσεις για απρόσκοπτη διασύνδεση και σε υψηλό εύρος ζώνης. Το διαδίκτυο λειτουργεί μέσω ενός συνόλου πρωτοκόλλων, όπως το *TCP*, που εξασφαλίζουν την αξιόπιστη και αποδοτική μεταφορά δεδομένων από έναν υπολογιστή σε έναν άλλο. Σε ένα δίκτυο βασιζόμενο στο πρωτόκολλο TCP, πολλαπλές συνεδρίες ή χρήστες ανταγωνίζονται για τους πόρους του δικτύου. Το TCP οργανώνει τα δεδομένα σε ροές και για να υποστηρίξει την υπηρεσία της μεταφοράς από άκρο σε άκρο εκμεταλλεύεται τους πόρους δικτύου. Το δίκτυο επιτρέπει οι πόροι να διαμοιράζονται και να κατανέμονται. Κάθε πόρος έχει συγκεκριμένη *χωρητικότητα* υπό την έννοια του μέγιστου *φορτίου* που μπορεί να εξυπηρετήσει. Βέβαια, η εύρεση μιας «καλής» κατανομής του φορτίου στους διαθέσιμους αλλά πεπερασμένους πόρους του δικτύου είναι μια δύσκολή υπόθεση. Αν η ζήτηση ξεπεράσει τη διαθέσιμη χωρητικότητα, τότε τα αντίστοιχα αιτήματα είτε μπλοκάρονται είτε τίθενται σε αναμονή εωσότου υπάρξει διαθεσιμότητα. Η εμφάνιση μιας τέτοιας συνθήκης στο δίκτυο αναφέρεται ως συμφόρηση.

Τα τελευταία χρόνια, η ραγδαία εξάπλωση της χρήσης των δικτύων οδήγησε στη συχνότερη εμφάνιση της συμφόρησης. Η συμφόρηση είναι μια ανεπιθύμητη κατάσταση, που προκύπτει όταν ο αριθμός των αιτημάτων για μεταφορά δεδομένων στο δίκτυο υπερβαίνει τις δυνατότητες επεξεργασίας και μεταφοράς από τους διαθέσιμους πόρους του δικτύου. Για να περιοριστεί το φαινόμενο αυτό είναι αναγκαίος ο σχεδιασμός και η εφαρμογή αποτελεσματικών αλγορίθμων ελέγχου της συμφόρησης.

Σύμφωνα με τους Cardwell κ.ά. [1], η συμφόρηση είναι αποτέλεσμα των επιλογών σχεδιασμού του TCP κατά την εμφάνιση του, την δεκαετία του 1980. Στα πρώτα χρόνια της λειτουργίας του TCP, βασική θεώρηση ήταν πως η απώλεια πακέτων οφειλόταν στην

συμφόρηση. Σήμερα ο έλεγχος συμφόρησης με βάση τις απώλειες δεν είναι αποδοτικός καθώς επιπλέον χρησιμοποιούνται προσωρινοί καταχωρητές. Εξάλλου, η συμφόρηση δεν είναι πρόβλημα έλλειψης πόρων αλλά πως αυτοί οι πόροι θα διανεμηθούν δυναμικά [2].



Εικόνα 1-1: Παράδειγμα εμφάνισης συμφόρησης σε δίκτυο [3].

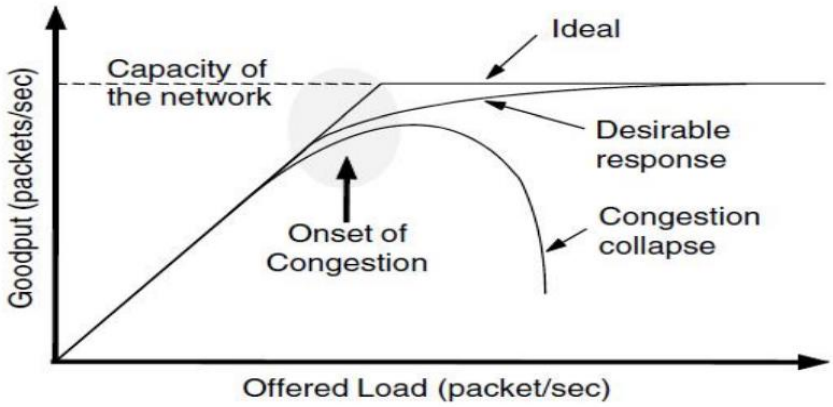
Ένα παράδειγμα συμφόρησης απεικονίζεται στην Εικόνα 1-1. Όπως παρατηρούμε, οι ροές των δεδομένων που φαίνονται στην Εικόνα 1-1 αναφέρονται σε δυο μη ανταποκρινόμενες ροές (ροή Α και ροή Β) οι οποίες ανταγωνίζονται για το εύρος ζώνης σε ένα δίκτυο που περιέχει δύο συνδέσμους συμφόρησης οι οποίοι διευθύνονται από μηχανισμό δίκαιου προγραμματισμού στους δρομολογητές Στον πρώτο σύνδεσμο συμφόρησης διασφαλίζεται ότι κάθε ροή λαμβάνει το μισό από το διαθέσιμο εύρος ζώνης του συνδέσμου (750 kb/s). Στον δεύτερο σύνδεσμο συμφόρησης μεγάλο μέρος της κίνησης από τη ροή Β απορρίπτεται λόγω της περιορισμένης χωρητικότητας του συνδέσμου (128 kb/s). Έτσι, για τη ροή Α ο ρυθμό μεταφοράς είναι 750 kb/s και η ροή Β πετυχαίνει ρυθμό μεταφοράς 128 kb/s.

Αυτό που παρατηρείται είναι ότι έχει συμβεί κατάρρευση λόγω συμφόρησης, επειδή τα πακέτα της ροής Β, τα οποία τελικά απορρίπτονται στον δεύτερο σύνδεσμο συμφόρησης , περιορίζουν τον ρυθμό μεταφοράς της ροής Α μέσω του πρώτου συνδέσμου συμφόρησης. Επίσης, ενώ και οι δύο ροές λαμβάνουν ίσο εύρος ζώνης στον πρώτο σύνδεσμο συμφόρησης, οι κατανομές τους δεν είναι δίκαιες.

Στην Εικόνα 1-2 απεικονίζονται οι επιπτώσεις της συμφόρησης. Όταν τα δεδομένα που στέλνονται στο δίκτυο είναι λιγότερα από την μεταφορική του ικανότητα τότε παραδίδεται (σχεδόν) το σύνολο των πακέτων. Μια αύξηση στο πλήθος των απεσταλμένων πακέτων οδηγεί σε ανάλογη αύξηση στον όγκο των παραδιδόμενων πακέτων. Όμως, αν το προσφερόμενο φορτίο πλησιάσει την χωρητικότητα του δικτύου, θα γεμίσουν περιστασιακά οι καταχωρητές των δρομολογητών και κάποια πακέτα θα χαθούν. Τα πακέτα που χάνονται καταναλώνουν μέρος της χωρητικότητας χωρίς όμως να παραδίδονται, με αποτέλεσμα ο

αριθμός των παραδομένων πακέτων να πέφτει κάτω από την ιδανική καμπύλη. Το φαινόμενο αυτό χαρακτηρίζεται ως κατάρρευση δικτύου [3].

Η διαχείριση της συμφόρησης είναι μια πρόκληση για τους σχεδιαστές και διαχειριστές του δικτύου. Από την πρώτη εμφάνισή της, το 1984, οι ερευνητές και οι σχεδιαστές των δικτύων επιζητούν λύσεις για το πρόβλημα αυτό.



Εικόνα 1-2: Οι επιπτώσεις της συμφόρησης στο δίκτυο

1.2 Το πρόβλημα της συμφόρησης

Τον Οκτώβριο του '86, οι επιστήμονες παρατήρησαν μια σειρά από «καταρρεύσεις λόγω συμφόρησης» του δικτύου με τη ρυθμοαπόδοση να μειώνεται από 32 Kbps σε 40 bps. Που οφείλεται αυτό το φαινόμενο; Η αύξηση του φορτίου σε κομβικά σημεία του δικτύου, οδηγεί σε μείωση της διέλευσης καθώς: α) πακέτα που ενώ έχουν καταναλώσει πόρους του δικτύου απορρίπτονται β)πακέτα που δεν παραδόθηκαν χρειάζεται να αναμεταδοθούν, όμως αυξάνουν το φορτίο. Την ίδια περίοδο, ο van Jacobson εισήγαγε αρχές ελέγχου συμφόρησης και πρακτικές λύσεις που βρίσκουν εφαρμογή στο πρωτόκολλο TCP. Ακολούθησαν πολλές πρωτοποριακές εργασίες, προτάθηκαν καινοτόμα εργαλεία (όπως το traceroute) και αναπτύχθηκαν αρκετές επεκτάσεις του TCP.

Τα σημερινά δίκτυα, επίσης αντιμετωπίζουν προβλήματα καθυστέρησης και φτωχής απόδοσης ως αποτέλεσμα της συμφόρησης [4].

1.3 **Αντικείμενο διπλωματικής**

Σήμερα τα δίκτυα μεταγωγής πακέτων έχουν συμπληρώσει πάνω από μισό αιώνα ζωής, όμως εξακολουθούν να υποφέρουν από το πρόβλημα της συμφόρησης καθώς η λειτουργία τους βασίζεται στην πολυπλεξία και την κοινή χρήση πόρων. Για παράδειγμα στο διαδίκτυο, σε περιπτώσεις αυξημένης κυκλοφορίας πακέτων ένας κοινόχρηστος πόρος (π.χ., σύνδεσμος, δρομολογητής), αποτελεί πεδίο ανταγωνισμού και καθώς δεν μπορεί να ανταποκριθεί στη ζήτηση μπορεί να οδηγήσει στην κατάρρευση δικτύου λόγω συμφόρησης. Το πιο σημαντικό πρωτόκολλο που χρησιμοποιείται για τη μετάδοση των δεδομένων στο διαδίκτυο είναι το TCP [6]. Το TCP διαθέτει μηχανισμούς για την αντιμετώπιση της συμφόρησης που στο πέρασμα του χρόνου βελτιώνονται και προσαρμόζονται στις ανάγκες των χρηστών. Βέβαια, και δεδομένης της συνεχής τεχνολογικής εξέλιξης, αυτοί οι μηχανισμοί δεν ικανοποιούν πλήρως τις απαιτήσεις το χρηστών, ενώ παράλληλα εμφανίζονται νέα προβλήματα που εντείνουν το πρόβλημα της συμφόρησης και μετριάζουν την απόδοση του δικτύου.

Σε αυτό το πλαίσιο, η παρούσα μελέτη, αποσκοπεί στην εξέταση των προσεγγίσεων που προτάθηκαν διαχρονικά για την επίλυση του προβλήματος της συμφόρησης ως βελτιώσεις ή παραλλαγές του πρωτοκόλλου TCP.

Σκοπός της εργασίας αυτής είναι η μελέτη των μηχανισμών ελέγχου ή μετριασμού της συμφόρησης στα σημερινά δίκτυα.

Πιο συγκεκριμένα στόχος της εργασίας είναι να:

- διερευνήσει τους μηχανισμούς που συμβάλλουν στη δημιουργία συμφόρησης
- παρουσιάσει βασικότερες τεχνικές αντιμετώπισης του προβλήματος που εφαρμόστηκαν μέχρι σήμερα
- επισημάνει τους περιορισμούς τους
- παρουσιάσει ενδεικτικά πειραματικά αποτελέσματα για θεμελιώδεις μετρικές απόδοσης δικτύου
- ερευνήσει μηχανισμούς αντιμετώπισης της συμφόρησης
- Παρουσιάσει τις σημαντικότερες προσπάθειες αντιμετώπισης και διαχείρισης της συμφόρησης

- Αξιολογήσει την αποτελεσματικότητά τους όταν εφαρμόζονται σε δυναμικά περιβάλλοντα με βάση την βιβλιογραφία.
- Παρέχει μια λεπτομερή επισκόπηση των μειονεκτημάτων και των πλεονεκτημάτων τους.
- Εκτιμήσει μέσω προσομοίωσης την απόδοση επιλεγμένων στρατηγικών

1.4 Δομή εργασίας

Η εργασία αυτή αποτελείται από 6 κεφάλαια.

Στο Κεφάλαιο 1 παρουσιάζεται το πρόβλημα της συμφόρησης, ένα παράδειγμα για την κατανόηση της κατάρρευσης της συμφόρησης και αναλύεται το αντικείμενο της διπλωματικής εργασίας. Στο κεφάλαιο 2 παρουσιάζεται η ιστορία του TCP αλλά και των μηχανισμών ελέγχου της συμφόρησης. Επίσης παρουσιάζεται η έννοια της συμφόρησης. Στο κεφάλαιο 3 αναλύονται οι τρόποι με της οποίους μπορεί να αναγνωριστεί η συμφόρηση. Στο κεφάλαιο 4 παρουσιάζεται η αξιολόγηση της απόδοσης του δικτύου σε συνθήκες συμφόρησης. Στο κεφάλαιο 5 παρουσιάζονται τα αποτελέσματα της εργασίας. Στο κεφάλαιο 6 παρουσιάζονται τα συμπεράσματα στα οποία κατέληξα.

2 Θεωρητικό υπόβαθρο

2.1 Η ιστορία του TCP και των μηχανισμών ελέγχου συμφόρησης

Η ιστορία του Διαδικτύου ξεκίνησε με την ανάπτυξη των ηλεκτρονικών υπολογιστών στη δεκαετία του 1950. Αυτό άρχισε με την επικοινωνία point-to-point μεταξύ mainframe υπολογιστών και τερματικών, επεκτάθηκε σε point-to-point συνδέσεις ανάμεσα σε υπολογιστές και στη συνέχεια σε μεταγωγή πακέτων .

Στη συνέχεια 1969, η Αμερικανική Υπηρεσία Ερευνών Προηγμένων Εργασιών (ARPA) δημιούργησε το Advanced Research Projects Agency Network (ARPANET), το πρώτο δίκτυο μεταγωγής πακέτων στον κόσμο που θεωρείται ως ο πρόγονος του διαδικτύου. Το ARPANET αναπτύχθηκε από την Υπηρεσία Προηγμένων Ερευνητικών Προγραμμάτων Άμυνας(DARPA) του Υπουργείου Άμυνας των ΗΠΑ και χρησιμοποιήθηκε από

πανεπιστήμια. Η μεταγωγή πακέτων είναι σήμερα η κυρίαρχη μορφή επικοινωνίας δεδομένων παγκοσμίως. Η ανάπτυξη του ξεκίνησε το 1969 και το δίκτυο έγινε λειτουργικό το 1970. Η μεταγωγή πακέτων του ARPANET βασίστηκε σε ένα σχέδιο του Lawrence Roberts του Εργαστηρίου Lincoln και εισάγει τη διαμοίραση των δικτυακών πόρων [2].

Κύριος σκοπός του ARPANET ήταν η δημιουργία ενός αξιόπιστου και ανθεκτικού συστήματος επικοινωνίας μεταξύ υπολογιστών, προκειμένου να εξασφαλιστεί η ανταλλαγή πληροφοριών και η διαμοιρασμός πόρων μεταξύ επιστημονικών και ερευνητικών οργανισμών.

Το ARPANET οδήγησε στην ανάπτυξη πρωτοκόλλων για δικτύωση, όπου πολλαπλά χωριστά δίκτυα θα μπορούσαν να ενωθούν σε ένα. Το ARPANET χρησιμοποίησε το πρωτόκολλο επικοινωνίας Network Control Protocol (NCP), το οποίο αργότερα εξελίχθηκε στο Transmission Control Protocol (TCP), το οποίο συνδυάστηκε με το Internet Protocol (IP) για τη δημιουργία του TCP/IP, που αποτελεί τη βάση του σύγχρονου Διαδικτύου. Το ARPANET αποτελεί ένα κρίσιμο στάδιο στην εξέλιξη των υπολογιστών και των δικτύων επικοινωνίας, καθώς σηματοδότησε την αρχή της παγκόσμιας δικτύωσης και του Διαδικτύου όπως το γνωρίζουμε σήμερα.[7]

Κατά τη δεκαετία του '70, δημιουργήθηκαν τα πρώτα ιδρύματα και πρότυπα που διαμόρφωσαν το μοντέλο του σημερινού διαδικτύου. Αναπτύχθηκε το πρωτόκολλο TCP/IP το οποίο έγινε η βάση του διαδικτύου. Οι δημιουργοί του TCP, συμπεριλαμβανομένων των Vinton Cerf και Bob Kahn, σχεδίασαν το πρωτόκολλο με σκοπό την αξιόπιστη μετάδοση δεδομένων στο ARPANET, το πρόγονο του σύγχρονου Διαδικτύου. Σύμφωνα με τους δημιουργούς του, το TCP αντιμετωπίζει με ευελιξία τη συμφόρηση «χάρη στη στιβαρή στρατηγική αναμετάδοσης και ανίχνευσης αντιγράφων» [8].

Το 1978 το TCP χωρίζεται σε δύο ξεχωριστά πρωτόκολλα: TCP και IP . Αυτό σηματοδοτεί τη γέννηση της σουίτας TCP/IP. Το 1980, το διαδίκτυο εξαπλώθηκε πέρα από τα ακαδημαϊκά και στρατιωτικά περιβάλλοντα και έγινε πιο ευρέως προσβάσιμο σε επιχειρήσεις και άλλους φορείς.

Λίγο αργότερα το 1982 το TCP/IP υιοθετείται ως το πρότυπο πρωτόκολλο για το ARPANET. Την επόμενη χρόνια κατά την 1η Ιανουαρίου 1983 σηματοδοτείται η αρχή της ευρείας χρήσης του TCP/IP και τη γέννηση του σύγχρονου διαδικτύου. Η αναγνώριση του TCP/IP ως κυρίαρχο πρωτόκολλο έγινε στα μέσα της δεκαετίας του 1990.Το 1986, όπως περιγράφει ο Van Jacobson στην εργασία του “Congestion avoidance and control”, το δίκτυο υπέφερε

συχνά από συμφόρηση[7]. Αυτή η διαπίστωση έστρεψε τον συγγραφέα προς την εξεύρεση λύσεων και πρότεινε μια σειρά από μηχανισμούς ελέγχου συμφόρησης με βάση το «παράθυρο συμφόρησης».

Το 1988 εμφανίζεται TCP Tahoe, δημιουργία του Jacobson. Το TCP Tahoe είναι μία τεχνική ελέγχου συμφόρησης TCP, που χρησιμοποιείται όταν ο αποστολέας λαμβάνει τρεις διπλές επιβεβαιώσεις. Αυτό το συγκεκριμένο TCP σχεδιάστηκε γύρω από τη λίμνη Tahoe και ως εκ τούτου ονομάστηκε TCP Tahoe. Το 1990 δημιουργείται το TCP Reno και δημοσιεύονται τα RFC 1122 και RFC 1123, τα οποία καθορίζουν τα πρότυπα για το TCP και το IP.

Το 1994 παρουσιάζεται η ρητή ειδοποίηση συμφόρησης (ECN). Είναι μια τεχνική που χρησιμοποιείται στο πρωτόκολλο δικτύου, κυρίως στο (TCP), για να βελτιώσει τη διαχείριση της συμφόρησης στο δίκτυο. Η ECN επιτρέπει τον ακριβέστερο έλεγχο της κυκλοφορίας δεδομένων και την ανίχνευση της συμφόρησης πριν προκληθούν προβλήματα από απώλειες δεδομένων στο δίκτυο [9].

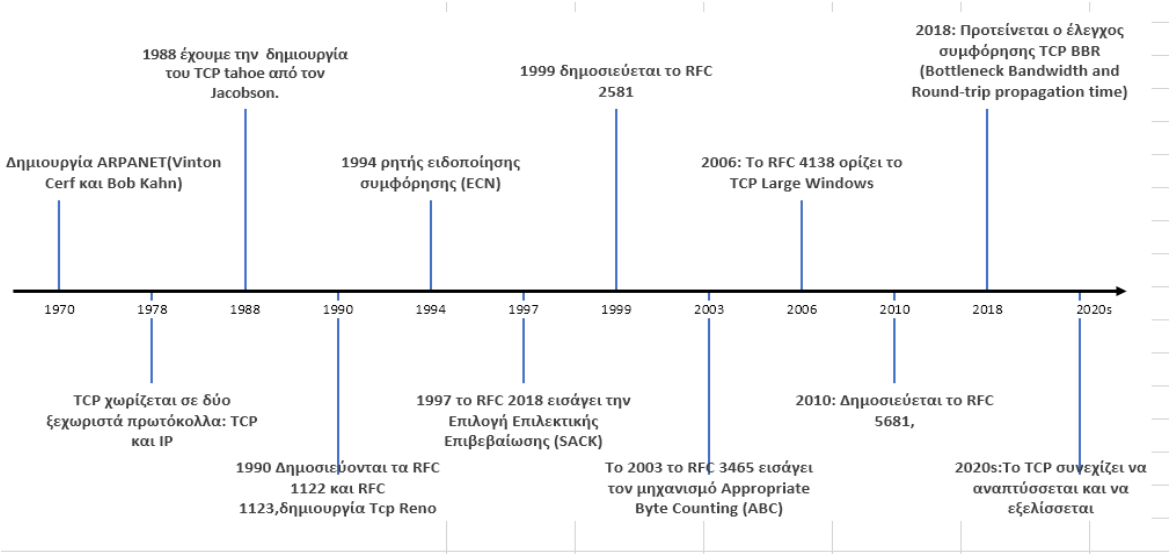
Το 1997 το RFC 2018 εισάγει την Επιλογή Επιλεκτικής Επιβεβαίωσης (SACK), βελτιώνοντας την ικανότητα του TCP να ανακάμπτει από την απώλεια πακέτων.

Το (1998), οι Floyd και Fall παρατηρούν ότι η αυξανόμενη ανάπτυξη κίνησης χωρίς έλεγχο συμφόρησης από άκρο σε άκρο μπορεί να οδηγήσει σε συμφόρηση, με αποτέλεσμα οι υπερφορτωμένοι σύνδεσμοι να αποστέλλουν πακέτα που θα απορριφθούν αργότερα στο δίκτυο, προκαλώντας ενδεχομένως κατάρρευση του δικτύου. Έχει επίσης αποδειχθεί (Kelly, Maulloo & Tan, 1998) ότι θα μπορούσε να επιτευχθεί με απλούς αλγόριθμους ελέγχου ρυθμού, μια δίκαιη κατανομή χρησιμοποιώντας κανόνες αύξησης και μείωσης παρόμοιους με αυτούς που περιγράφονται από τους Chiu και Jain (1989) και τον Jacobson (1988) και εφαρμόζονται στο TCP. Οι Crowcroft και Oechslin (1998) έχουν προτείνει τρόπους ρύθμισης των παραμέτρων του πρωτοκόλλου TCP για να επιτύχουν δίκαιη αναλογική κατανομή. Επίσης έχουν παρουσιάσει αποτελέσματα για τη δυνατότητα ύπαρξης ενός TCP όπου θα υπολογίζει τον κόστος χρησιμοποιώντας παραμέτρους ελέγχου συμφόρησης που μπορούν να τροποποιηθούν από τους τελικούς κόμβους με τρόπο συμβατό με τα κίνητρα [11].

Από τα μέσα της δεκαετίας του 1990 το διαδίκτυο έχει αλλάξει σημαντικά τη ζωή των ανθρώπων, χάρη στην πρόσβαση που παρέχει σε χιλιάδες εφαρμογές (ηλεκτρονικό ταχυδρομείο, ανταλλαγή άμεσων μηνυμάτων, Voice over Internet Protocol (VoIP), μεταφορά αρχείων, πλοήγηση στο διαδίκτυο, και πολλά άλλα.

Το 1999 δημοσιεύεται το RFC 2581, το οποίο επικαιροποιεί τους μηχανισμούς ελέγχου συμφόρησης του TCP

Το TCP από την πρώτη χρήση του μέχρι και σήμερα, αποτελεί το περισσότερο σημαντικό πρωτόκολλο του διαδικτύου. Τα πρώτα χρόνια της χρήσης του, το TCP αντιμετώπιζε με ευελιξία τη συμφόρηση «χάρη στη στιβαρή στρατηγική επαναμετάδοσης και ανίχνευσης αντιγράφων» σύμφωνα με τους Cerf και Kahn [Cerf και Kahn1974]. Την περίοδο αυτή – πρώιμη εποχή του διαδικτύου (έως 1990) – το κύριο σημείο συμφόρησης αποτελούσαν οι συνδέσεις μεγάλων αποστάσεων[10]. Τη δεκαετία του 1990, δηλαδή τη φάση ανάπτυξης του διαδικτύου, η πρόσβαση μέσω τηλεφώνου αντιμετώπισε πολλές δυσκολίες εξ αιτίας των χαμηλών ρυθμών μετάδοσης δεδομένων. Μετά το 2000, η ζήτηση για πρόσβαση στο διαδίκτυο καλύπτεται με ευρυζωνικές συνδέσεις, όμως η ανάγκη για μαζική μεταφορά δεδομένων, αποτελεί περιοριστικό παράγοντα. Από το 2000 και μετά το διαδίκτυο και η προσβασιμότητα συνεχίζουν να εξελίσσονται με γρήγορους ρυθμούς. Το διαδίκτυο έχει γίνει αναπόσπαστο μέρος της καθημερινής ζωής, παρέχοντας πληροφορίες, ψυχαγωγία, επικοινωνία και ηλεκτρονικό εμπόριο σε παγκόσμια κλίμακα [6] .



Εικόνα 2-1 Η εξέλιξη των πρωτοκόλλων TCP για την αντιμετώπιση της συμφόρησης

Το 2003 το RFC 3465 εισάγει τον μηχανισμό Appropriate Byte Counting (ABC) για τον έλεγχο συμφόρησης του TCP.

Το 2004 το FAST TCP (επίσης γραμμένο FastTCP) είναι ένας αλγόριθμος αποφυγής συμφόρησης TCP που στοχεύει ειδικά σε συνδέσεις μεγάλων αποστάσεων, υψηλής

καθυστέρησης, που αναπτύχθηκε στο Netlab του Ινστιτούτου Τεχνολογίας της Καλιφόρνια και τώρα διατίθεται στο εμπόριο από τη FastSoft. Η FastSoft εξαγοράστηκε από την Akamai Technologies το 2012.

2006: Το RFC 4138 ορίζει το TCP Large Windows, το οποίο επιτρέπει πιο εκτεταμένη κλιμάκωση παραθύρων. Το TCP CUBIC είναι ένας αλγόριθμος αποφυγής συμφόρησης δικτύου για TCP που μπορεί να επιτύχει συνδέσεις υψηλού εύρους ζώνης μέσω δικτύων πιο γρήγορα και αξιόπιστα ενόψει της υψηλής καθυστέρησης από τους προηγούμενους αλγόριθμους. Βοηθά στη βελτιστοποίηση δικτύων μεγάλου όγκου, η πρώτη υλοποίηση CUBIC κυκλοφόρησε στον πυρήνα Linux 2.6.13. Από την έκδοση 2.6.19 του πυρήνα, το CUBIC αντικαθιστά το BIC-TCP ως τον προεπιλεγμένο αλγόριθμο ελέγχου συμφόρησης TCP στον πυρήνα του Linux.

Το 2008 το TCP CUBIC αποκτά ευρεία προβολή.

2010: Δημοσιεύεται το RFC 5681, το οποίο επικαιροποιεί τους αλγορίθμους ελέγχου συμφόρησης του TCP.

2018: Προτείνεται ο έλεγχος συμφόρησης TCP BBR (Bottleneck Bandwidth and Round-trip propagation time) για τη βελτιστοποίηση των επιδόσεων του δικτύου.

2020s: Το TCP συνεχίζει να αναπτύσσεται και να εξελίσσεται, με συνεχείς προσπάθειες για τη βελτίωση της απόδοσής του και την προσαρμογή του στις μεταβαλλόμενες συνθήκες του δικτύου. Καθ' όλη τη διάρκεια της ιστορίας του, το TCP παρέμεινε ένα θεμελιώδες πρωτόκολλο για την αξιόπιστη μετάδοση δεδομένων στο Διαδίκτυο. Έχει υποστεί πολυάριθμες βελτιστοποιήσεις και βελτιώσεις για να ανταποκριθεί στις αυξανόμενες απαιτήσεις της σύγχρονης δικτύωσης, συμπεριλαμβανομένων των συνδέσεων υψηλής ταχύτητας και των ποικίλων εφαρμογών. Η σουίτα TCP/IP εξακολουθεί να αποτελεί τη ραχοκοκαλιά του Διαδικτύου, εξασφαλίζοντας την αξιόπιστη παράδοση πακέτων δεδομένων σε παγκόσμια δίκτυα.[6][9]

2.2 Η λειτουργία του πρωτοκόλλου TCP

Το TCP βρίσκεται πάνω από το πρωτόκολλο IP. Το TCP δημιουργεί και διατηρεί σύνδεση μεταξύ του αποστολέα και του παραλήπτη πριν από την ανταλλαγή δεδομένων. Αυτό εξασφαλίζει τη σταθερότητα και την αξιοπιστία της μετάδοσης. Τα πακέτα που μεταδίδονται

φθάνουν ακέραια στον προορισμό τους χωρίς λάθη και με σωστή σειρά. Οι πιο γνωστές εφαρμογές που χρησιμοποιούν το πρωτόκολλο TCP είναι το πρωτόκολλο HTTP , η αλληλογραφία ηλεκτρονικού ταχυδρομείου (email), και η μετάδοση αρχείων καθώς τους προσφέρει αξιοπιστία και ακεραιότητα των δεδομένων. Το TCP είναι ένα αξιόπιστο πρωτόκολλο που εγγυάται ότι τα πακέτα θα παραδοθούν στον προορισμό τους, θα φτάσουν με τη σειρά με την οποία στάλθηκαν και ότι τα περιεχόμενα των πακέτων θα φτάσουν αναλλοίωτα.

Η βασική λειτουργία του συγκεκριμένου πρωτοκόλλου είναι η παροχή αξιόπιστης υπηρεσίας με σύνδεση για τη μεταφορά δεδομένων. Το πρωτόκολλο επιτρέπει τη μετάδοση της πληροφορίας προς δυο κατευθύνσεις (εκπομπός, δέκτης). Τα δεδομένα οργανώνονται σε τμήματα (πακέτα) για να μεταδοθούν. Το κάθε τμήμα περιέχει την επικεφαλίδα και το ωφέλιμο φορτίο, με το μέγεθός του να μην ξεπερνά τα 1500 bytes. Το λογισμικό TCP του λειτουργικού συστήματος έχει την ευθύνη για την εγκαθίδρυση και τον τερματισμό των συνδέσεων από άκρο σε άκρο, καθώς και για τη μεταφορά δεδομένων.

Το TCP εργάζεται ως εξής: το κάθε πακέτο δεδομένων αριθμείται. Ο παραλήπτης και ο αποστολέας, αλλά όχι οι ενδιάμεσοι υπολογιστές, παρακολουθούν τους αριθμούς των πακέτων και ανταλλάσσουν μεταξύ τους πληροφορίες. Ο παραλήπτης λαμβάνει το πρώτο πακέτο, το δεύτερο, κλπ. Σε περίπτωση που παρουσιαστεί κάποιο πρόβλημα στο δίκτυο είτε χαθεί κάποιο πακέτο κατά τη διάρκεια της μετάδοσης, το ξαναζητάει και ο αποστολέας είναι υπεύθυνος για την αναμετάδοση του. Ο παραλήπτης ελέγχει επίσης αν το περιεχόμενο των πακέτων φτάνει σωστά. Με αυτή τη μέθοδο δεν πραγματοποιούνται έλεγχοι μέσω των ενδιάμεσων υπολογιστών και έτσι εξασφαλίζεται αξιοπιστία και ταχύτητα.

Το TCP περιλαμβάνει μηχανισμούς για:

- Δημιουργία σύνδεσης
- Μεταφορά δεδομένων
- Τερματισμό σύνδεσης
- Υπολογισμός του Round-trip time & του timeout
- Επανεκπομπή μετά από καθορισμένο χρονικό όριο
- Έλεγχος ροής

- Έλεγχος συμφόρησης

2.2.1 Δημιουργία σύνδεσης

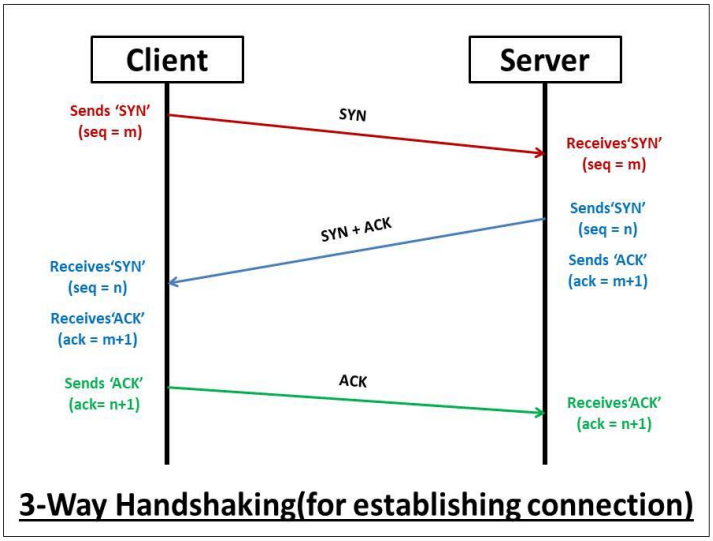
Το "3-way handshake" (τριπλή χειραψία) είναι μια διαδικασία που χρησιμοποιείται από το TCP για να δημιουργήσει μια σύνδεση ανάμεσα σε δύο υπολογιστές (ή σε έναν υπολογιστή και έναν διακομιστή) σε ένα δίκτυο. Η τριπλή χειραψία αποτελείται από τρία στάδια ή μηνύματα που ανταλλάσσονται μεταξύ του αποστολέα (client) και του παραλήπτη (server) για να καθιερώσουν μια σύνδεση TCP. Τα τρία στάδια που απεικονίζονται στην Εικόνα 2-2 ,είναι τα εξής:

Αίτηση (SYN): Ο αποστολέας ξεκινά τη σύνδεση αποστέλλοντας ένα πακέτο με τη σημαία (flag) SYN (Synchronize) στον παραλήπτη. Αυτό το πακέτο περιέχει τον αριθμό ακολουθίας (sequence number) που χρησιμοποιεί ο αποστολέας.

Απάντηση (SYN-ACK): Ο παραλήπτης λαμβάνει την αίτηση, επιβεβαιώνει την παραλαβή της με το flag ACK (Acknowledge), και απαντά με ένα πακέτο που περιέχει τη σημαία SYN και τη σημαία ACK. Αυτό το πακέτο περιέχει επίσης έναν αριθμό ακολουθίας για τον παραλήπτη καθώς και έναν αριθμό ακολουθίας για τον αποστολέα.

Επιβεβαίωση (ACK): Ο αποστολέας λαμβάνει την απάντηση του παραλήπτη και αποστέλλει ένα τελευταίο πακέτο με τη σημαία ACK. Αυτό το πακέτο περιέχει τον αριθμό ακολουθίας του παραλήπτη και επιβεβαιώνει την επιτυχή σύνδεση των δύο μερών.

Αφού ολοκληρωθεί η τριπλή χειραψία, η σύνδεση TCP έχει δημιουργηθεί επιτυχώς, και τα δεδομένα μπορούν να ανταλλαχθούν μεταξύ του αποστολέα και του παραλήπτη. Αυτή η διαδικασία είναι σημαντική για την εξασφάλιση της αξιοπιστίας και της σειριακής παράδοσης των δεδομένων σε μια σύνδεση TCP.



Εικόνα 2-2 Διαδικασία Τριπλής χειραψίας

2.2.2 Μεταφορά δεδομένων

Κατά την ανταλλαγή των δεδομένων εφαρμόζονται δύο μηχανισμοί: α) ο έλεγχος ροής και β) ο έλεγχος συμφόρησης.

Α) Ο έλεγχος ροής περιλαμβάνει τη διαχείριση του όγκου των δεδομένων που αποστέλλονται σε ένα δίκτυο για την αποφυγή καθυστερήσεων και απωλειών που προκαλούνται από συμφόρηση.

Β) ο έλεγχος συμφόρησης αντιμετωπίζει τη συμφόρηση δικτύου μειώνοντας τη ροή δεδομένων κατά την ανίχνευση συμφόρησης, αποφεύγοντας έτσι τη μείωση της απόδοσης του δικτύου.

Οι μέθοδοι *ARQ (Automatic Repeat reQuest)* είναι μια τεχνική επικοινωνίας η οποία χρησιμοποιείται για τη διασφάλιση της αξιόπιστης μεταφοράς δεδομένων σε δίκτυα. *One Outside Packet ARQ:* Αυτή η μέθοδος συνήθως αναφέρεται σε μια απλή στρατηγική όπου, αν ένα πακέτο δεν ληφθεί ή είναι κατεστραμμένο, ο παραλήπτης ζητά ξανά την αποστολή μόνο εκείνου του πακέτου.

2.2.3 Τερματισμός σύνδεσης

Ο τερματισμός μιας σύνδεσης TCP γίνεται μέσω μιας σειράς μηνυμάτων μεταξύ του αποστολέα (client) και του παραλήπτη (server). Αυτή η διαδικασία γνωστή ως "τετράδα χειραψίας" (Four-Way Handshake) περιλαμβάνει τέσσερα στάδια. Ας δούμε πώς λειτουργεί:

Αίτημα (FIN): Ένας από τους δύο συνδεδεμένους κόμβους (client ή server) εκκινεί τον τερματισμό της σύνδεσης από πλευράς του, στέλνει ένα πακέτο με το FIN ενεργοποιημένο,

Επιβεβαίωση (ACK): Ο άλλος κόμβος λαμβάνει το μήνυμα FIN και απαντά με ένα πακέτο που έχει το ACK (Acknowledgment) ενεργοποιημένο. Αυτό επιβεβαιώνει τη λήψη του μηνύματος FIN.

Αίτημα (FIN): Ο δεύτερος κόμβος (που δεν ξεκίνησε τον τερματισμό) μπορεί αν θέλει να τερματίσει να αποστέλλει ένα πακέτο FIN.

Επιβεβαίωση (ACK): Ο πρώτος κόμβος απαντά με ένα πακέτο που έχει τα flags FIN και ACK ορισμένα, επιβεβαιώνοντας, ότι σύνδεση έχει τερματιστεί.

Μια σύνδεση μπορεί να είναι "half-open", δηλαδή μόνο η μια πλευρά να έχει τερματίσει. Η πλευρά που έχει τερματίσει δεν μπορεί να στείλει πλέον δεδομένα, ενώ η άλλη μπορεί.

Τέλος, υπάρχει πιθανότητα αν και πολύ σπάνια οι δύο host να στέλνουν πακέτα FIN ο ένας στον άλλο ταυτόχρονα. Μετά από αυτό, όλοι εγκρίνουν το FIN που δέχονται στο πακέτο ACK. και διακόπτονται και οι δυο συνδέσεις. Είναι σημαντικό να σημειωθεί ότι ο κάθε κόμβος μπορεί να ξεκινήσει τον τερματισμό ανεξάρτητα. Μετά τον τερματισμό όμως καμία από τις δύο πλευρές δεν μπορεί να στείλει περαιτέρω δεδομένα στη σύνδεση.

2.2.4 Υπολογισμός του Round-Trip Time & του timeout

Υπολογισμός RTT

Ο χρόνος μετάδοσης μετά επιστροφής, γνωστός και ως RTT (Round-Trip Time), αντιπροσωπεύει τον χρόνο που απαιτείται για ένα πακέτο δεδομένων να ταξιδέψει από τον αποστολέα σε έναν προορισμό και να επιστρέψει πίσω. Είναι ένας σημαντικός παράγοντας για την αξιολόγηση της απόδοσης ενός δικτύου και την κατανόηση της καθυστέρησης που αντιμετωπίζουν τα πακέτα δεδομένων κατά τη μετάδοσή τους. Υπολογίζουμε τον RTT από τον τύπο:

$$EstimatedRTT \equiv (1 - a) * Estimated + a * SampleRTT$$

Όπου α=0,125

Ο υπολογισμός της συγκεκριμένης παραμέτρου είναι απαραίτητη έτσι ώστε να γνωρίζουμε αν ένα πακέτο έχει καθυστερήσει ή ακόμα και αν έχει χαθεί. Ο χρόνος αυτός μπορεί να είναι εξαιρετικά μεταβλητός μακροπρόθεσμα ή βραχυπρόθεσμα. Προς αποφυγήν αυτού του προβλήματος μπορούμε να κάνουμε τα εξής

- να χρησιμοποιούμε εκθετικά σταθμισμένος κινητός μέσος όρος (Exponentially Weighted Moving Average - EWMA)
- να εκτιμούμε την απόκλιση, καθώς και την αναμενόμενη τιμή
- να υποθέτουμε ότι το πακέτο χάθηκε όταν ο χρόνος ξεπερνά την λογική απόκλιση

Το RTT μπορεί να επηρεάζεται από πολλούς παράγοντες, όπως η απόσταση μεταξύ του αποστολέα και του προορισμού, η ταχύτητα μετάδοσης του μέσου επικοινωνίας (π.χ., φυσικό καλώδιο ή ασύρματο), αλλά και το φορτίο που μεταφέρεται (π.χ., φυσικό καλώδιο ή ασύρματο), οι καθυστερήσεις στο δρομολογητή, καθώς και άλλοι παράγοντες του δικτύου. Είναι σημαντικό να μετράτε και να λαμβάνεται υπόψη κατά τον σχεδιασμό και την αξιολόγηση των δικτυακών εφαρμογών, καθώς μπορεί να έχει σημαντικές επιδράσεις στην απόδοση και την αντίληψη των χρηστών.

Ο χρόνος μετάδοσης μετ'επιστροφης υπολογίζεται στο πρωτόκολλο ελέγχου μεταφοράς TCP με τη χρήση επιβεβαίωσης. Το RTT παρέχει πληροφορίες σχετικά με τον χρόνο που απαιτείται για ένα πακέτο δεδομένων να φύγει από τον αποστολέα, να φτάσει στον προορισμό, και να επιστρέψει πίσω στον αποστολέα.[9]

2.2.5 Επανεκπομπή μετά από καθορισμένο χρονικό όριο

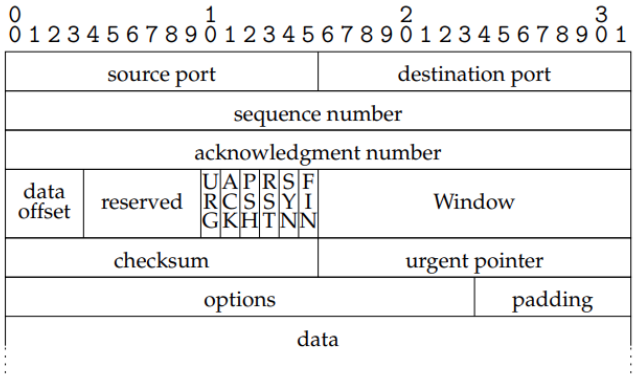
Αν ένα segment (δηλαδή ένα τμήμα TCP που αποτελείται από τα bytes δεδομένων προς αποστολή και την επικεφαλίδα) δεν επιβεβαιωθεί, θα πρέπει να επαναμεταδοθεί. Η αναμετάδοση είναι μια σύνθετη διαδικασία που βασίζεται στην εκτίμηση του χρόνου μετάβασης μετά επιστροφής και την εφαρμογή εκθετικής οπισθοχώρησης σε περίπτωση συμφόρησης. Τα ζητήματα που τίθενται εδώ είναι:

α) ο υπολογισμός του RTT

β) και πώς θα αναμεταδοθούν τα πακέτα

2.3 Έλεγχος ροής στο TCP

Στόχος αυτής της λειτουργίας είναι ο περιορισμός του ρυθμού αποστολής πακέτων από τον αποστολέα, έτσι ώστε ο παραλήπτης να είναι σε θέση να τα χειριστεί. Οι τεχνικές που εφαρμόζονται για το σκοπό αυτό βασίζονται σε πρωτόκολλα το κυλιόμενου παραθύρου (Sliding window). Το μήκος του παραθύρου (έστω w) ορίζεται στο αντίστοιχο πεδίο της επικεφαλίδας (Εικόνα 2-3) και είναι ανάλογο της ποσότητας (ή του χώρου που διαθέτει) πληροφορίας που μπορεί να χειριστεί ο παραλήπτης [9]. Ο κανόνας επιβάλλει να στέλνονται μη επιβεβαιωμένα έως και w πακέτα.



Εικόνα 2-3 Η δομή του TCP segment

Ένα από τα κυριότερα μέρη της δομής του TCP segment είναι η επικεφαλίδα (TCP header).Το μέγεθος της επικεφαλίδας μπορεί να είναι από 5 words το ελάχιστο μέχρι και 15 words για το μέγιστο. Στη συνέχεια εξηγούμε τα υπόλοιπα μέρη από τα οποία αποτελείται ένα TCP segment.

Source Port: προσδιορίζει τον αριθμό για τη θύρα αποστολής

Destination Port: προσδιορίζει τον αριθμό για τη port (θύρα) του παραλήπτη

Sequence Number: προσδιορίζει τον αριθμό ακολουθίας όπου

- Εάν υπάρχει η SYN flag τότε είναι ο αρχικός αριθμός ακολουθίας (ISN - initial sequence number) και η πρώτη octet δεδομένων του πακέτου είναι ο ISN+1.
- Εάν δεν υπάρχει η SYN flag, τότε η πρώτη octet δεδομένων είναι ο αριθμός ακολουθίας.

Acknowledgment number: Προσδιορίζει τον επόμενο sequence number (αριθμό ακολουθίας) που αναμένει ο αποστολέας

Data offset: Είναι ο αριθμός από words μεγέθους 32 bit στην επικεφαλίδα TCP (TCP header). Καθορίζει το μέγεθος της επικεφαλίδας (πολλαπλάσιο του 32) και επομένως δείχνει και την αρχή των δεδομένων

Reserved: Πεδίο 6 bit για μελλοντική χρήση. Η τιμή των bit πρέπει να είναι 0.

Flags: Περιέχει 6 bit - σημαίες:

Window: Ο αριθμός από octets δεδομένων (bytes) που επιθυμεί να δεχτεί ο αποστολέας του πακέτου

Checksum:Το πεδίο μεγέθους 16 bit χρησιμοποιείται για έλεγχο λαθών στην επικεφαλίδα και στα δεδομένα.

Options: Μεταβλητή, η οποία καθορίζει ειδικές επιλεγόμενες ρυθμίσεις και μπορεί να καταλάβει χώρο στο τέλος της επικεφαλίδας TCP (TCP header)

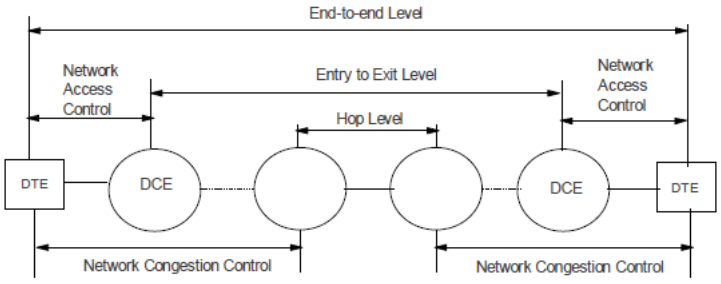
Urgent pointer: Εάν είναι ενεργοποιημένο το URG bit ελέγχου, τότε αυτό το πεδίο δείχνει τον αριθμό ακολουθίας (sequence number)της *octet* που βρίσκεται αμέσως μετά το τελευταίο byte από τα επείγοντα δεδομένα [11].

2.4 Έλεγχος συμφόρησης στο TCP

Ο έλεγχος συμφόρησης σε δίκτυα είναι ένα ενεργό ερευνητικό πεδίο από τη δεκαετία του 1980 έως και σήμερα. Πλήθος επιστημονικών εργασιών μελετούν το φαινόμενο και διακρίνουν τέσσερα διαφορετικά σημεία που μπορεί να ελεγχθεί η συμφόρηση[2]:

- Έλεγχος σε επίπεδο hop
- Έλεγχος σε επίπεδο εισόδου-εξόδου
- Έλεγχος σε επίπεδο δικτύου
- Έλεγχος από άκρο σε άκρο

Στην Εικόνα(2-4) απεικονίζονται τα σημεία του δικτύου που μπορεί να γίνει έλεγχος συμφόρησης.



Εικόνα 2-4 Σημεία του δικτύου που μπορεί να γίνει έλεγχος συμφόρησης. Πηγή [2].

Το TCP υλοποιεί αλγορίθμους συμφόρησης από άκρο σε άκρο που συνήθως υπολογίζουν το παράθυρο συμφόρησης ή το ρυθμό μετάδοσης από το ρυθμό λήψης των επιβεβαιώσεων. Η μη άφιξη επιβεβαιώσεων σηματοδοτεί αυξημένες καθυστερήσεις ή απώλειες πακέτων. Από μια τέτοια κατάσταση συμπεραίνεται η ύπαρξη συμφόρησης στο δίκτυο.

Όταν ο έλεγχος γίνεται σε επίπεδο δικτύου, οι δρομολογητές παρέχουν ανατροφοδότηση στα τελικά συστήματα είτε χρησιμοποιώντας ένα απλό bit που υποδεικνύει την ύπαρξη συμφόρησης, είτε δηλώντας ρητά το ρυθμό με τον οποίο ο αποστολέας θα στείλει τα δεδομένα του.

Υπό το πρίσμα της θεωρίας ελέγχου, η συμφόρηση στα δίκτυα μπορεί να ελεγχθεί με συστήματα ανοικτού ή κλειστού βρόχου. Ουσιαστικά, το σύστημα μοντελοποιείται έως ένα δίκτυο ουρών αναμονής και εξυπηρετητών στο οποίο ρυθμίζεται η ποσότητα των δεδομένων. Στα συστήματα ανοικτού βρόχου, η είσοδος δεν επηρεάζεται από την έξοδο του συστήματος. Αντίθετα, στα συστήματα κλειστού βρόχου η είσοδος καθορίζεται από την ανατροφοδότηση που λαμβάνεται από την έξοδο του συστήματος.

Πληθώρα λύσεων έχουν προταθεί για τον έλεγχο της συμφόρησης. Οι βασικότερες από αυτές τις προσεγγίσεις παρουσιάζονται στο επόμενο κεφάλαιο.

2.5 Συμφόρηση

Η συμφόρηση είναι ένα φαινόμενο που εμφανίζεται σε ένα δίκτυο μεταγωγής πακέτων όταν πολλαπλές σύνοδοι ανταγωνίζονται για τους πόρους του δικτύου. Στους πόρους ενός δικτύου περιλαμβάνονται το εύρος ζώνης, ο δρομολογητής, και η χωρητικότητα του καταχώρηση του δρομολογητή. Εάν η ζήτηση για έναν πόρο υπερβαίνει τη χωρητικότητα, εμφανίζεται συμφόρηση. Η συμφόρηση εκδηλώνεται με αυξημένες καθυστερήσεις πακέτων ή απώλειες πακέτων.

3 Αντιμετώπιση της συμφόρησης στο TCP

Ο έλεγχος συμφόρησης είναι χρήσιμος για να αποτραπεί η υπερφόρτωση των πόρων του δικτύου /διαδικτύου που θα οδηγήσει στο μπλοκάρισμα των δικτύων. Είναι μια διαδικασία που αποτελεί μια συνεχή πρόκληση για τα δίκτυα επικοινωνίας αλλά και ένα μεγάλο πρόβλημα το οποίο εμποδίζει την ομαλή λειτουργία του δικτύου[12]. Γι' αυτό το λόγο θα πρέπει να βρεθούν τρόποι έτσι ώστε να μειωθεί το φαινόμενο αυτό καθώς δεν είναι δυνατόν να αποφευχθεί.

Οι μηχανισμοί ελέγχου συμφόρησης που εφαρμόζονται στο TCP περιλαμβάνουν

- το Adaptive time out
- το slow-start
- congestion avoidance
- fast retransmit
- τον αλγόριθμο fast recovery

3.1 Πως θα αναγνωρίσουμε τη συμφόρηση ;

Υπάρχουν διάφοροι τρόποι που μπορούμε να αναγνωρίσουμε και να προλάβουμε ενδεχομένως την συμφόρηση ενός δικτύου. Παρακάτω παρουσιάζονται κάποιοι από τους σημαντικότερους:

1. timeout (loss): απώλεια πακέτων και λήξη αναμετάδοσης
2. dupACK (loss likely): Διπλές επιβεβαιώσεις στην περίπτωση αυτή όταν ο δέκτης δεν λαμβάνει επιβεβαίωση για το πακέτο που έχει στείλει αντιλαμβάνεται ότι στο δίκτυο υπάρχει συμφόρηση καθώς το πακέτο δεν έχει φθάσει στο προορισμό του.
3. queueing delay: Ο χρόνος που το πακέτο περιμένει στην ουρά μέχρι να μπορεί να υποβληθεί σε επεξεργασία. Εάν τα πακέτα φτάνουν πιο γρήγορα από τον

δρομολογητή τότε τα πακέτα θα βρίσκονται στην ουρά μέσα σε ένα εισερχόμενο bufferInflight.

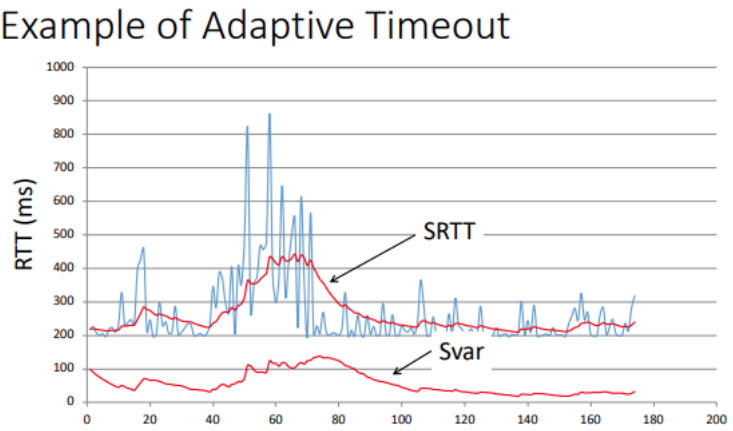
4. mark (needs ECN): Μέσω σήματος το οποίο χρησιμοποιεί το ECN [14], [13]

3.2 ACK clocking

Στον μηχανισμό αυτό το κάθε ACK εντός σειράς προωθεί το ολισθένον παράθυρο. Με αυτό τον τρόπο μπορεί να εισαχθεί ένα νέο τμήμα στο δίκτυο. Ο μηχανισμός ελέγχου ονομάζεται «χρονισμός» των πακέτων. Μετά την αποστολή ενός πλήρους μεγέθους παραθύρου από bytes, το TCP πρέπει να περιμένει για ACKs. Έτσι, τα πακέτα λέγεται ότι «χρονίζονται» από τα ACKs..Το παράθυρο αυτό ονομάζεται παράθυρο συμφόρησης ή αλλιώς cwnd και ο ρυθμός του υπολογίζεται από τον τύπο cwnd/RTT.Το TCP στέλνει μικρές ριπές τμημάτων έτσι ώστε το δίκτυο να μπορεί να της διαχειριστεί χωρίς να δημιουργήσει συμφόρηση.

3.3 Adaptive Time Out

Ο αλγόριθμος TCP timeout είναι μια διαδικασία που χρησιμοποιείται στο TCP για την αποφυγή για την εκτίμηση του RTT. Όπως φαίνεται στην Εικόνα 3.1, η εξομαλυμένη εκτίμηση του RTT, βασίζεται σε συνεχείς ενημερώσεις των εκτιμήσεων με τη βοήθεια του κινητού μέσου όρου. Το RTT, είναι μια εκτίμηση του χρόνου που απαιτείται για να στείλει και να λάβει ένα πακέτο από τον παραλήπτη και προσαρμόζει το χρόνο αναμονής με βάση αυτό το χρόνο. Το TCP παρέχει αξιοπιστία είτε χρησιμοποιώντας timeout αναμετάδοσης που βασίζεται στον υπολογισμό του χρονικού ορίου ή με μηχανισμό γρήγορης αναμετάδοσης που εξαρτάται από τη λήψη τριών διαδοχικών επιβεβαιώσεων. Ωστόσο η αναμονή για παρατεταμένο χρονικό διάστημα ή αναμονή για τρεις διπλές επιβεβαιώσεις, θα μπορούσαν να αυξήσουν τη συνολική καθυστέρηση και να μειώσουν τη μετάδοση δεδομένων. Επομένως, πρέπει να υπολογίσουμε μια κατάλληλη τιμή χρονικού ορίου έτσι ώστε να μπορούμε να υπολογίσουμε την πραγματική αιτία της απώλειας. Αυτό μπορεί να υπολογιστεί με τη χρήση του συγκεκριμένου αλγορίθμου.



Εικόνα 3-1 παράδειγμα adaptive time out [8]

3.4 End-host side

3.4.1 Slow Start

Η στρατηγική της αργής εκκίνησης χρησιμοποιείται στο πρωτόκολλο ελέγχου μεταφοράς TCP (Transmission Control Protocol) και έχει σκοπό τη βελτιστοποίηση της απόδοσης κατά τη μετάδοση δεδομένων σε δίκτυα.

Η διαδικασία λειτουργεί ως εξής:

- Στην αρχή της σύνδεσης ορίζεται το cwnd σε ένα πακέτο ή περισσότερα. Ο αποστολέας ξεκινά με ένα μικρό αριθμό πακέτων που μπορεί να στείλει προτού λάβει επιβεβαίωση (ACK) από τον παραλήπτη.
- Σε κάθε επιβεβαίωση δεδομένων, αυξάνουμε το cwnd κατά ένα πακέτο
- Η διαδικασία αυτή επαναλαμβάνεται έως ότου ο αποστολέας δεν λαμβάνει πλέον επιβεβαίωση από τον παραλήπτη, πράγμα που σημαίνει ότι είτε εντοπίζεται συμφόρηση είτε έχει συμπληρωθεί το όριο παραθύρου.

Σε αυτή τη διαδικασία υπάρχει ένας κανόνας που θα πρέπει να εφαρμόζεται για τη σωστή λειτουργία της σύνδεσης. Ο κανόνας λέει ότι : το πλήθος των απεσταλμένων και μη επιβεβαιωμένων bytes πρέπει είναι μικρότερα ή ίσα με το cwnd. bytes ≤ cwnd. Ο αποστολέας λαμβάνει την απώλεια σήματος ως σήμα συμφόρησης δικτύου. Συμπεραίνουμε λοιπόν ότι όταν δεν έχουμε απώλεια πακέτων ο ρυθμός ανάπτυξης της ¨αργής¨ εκκίνησης αυξάνεται εκθετικά. Η μόνη περίπτωση που μπορεί να είναι αργός είναι στο σημείο εκκίνησης όπου έχουμε 1 MSS.

3.4.2 Congestion Anoidance

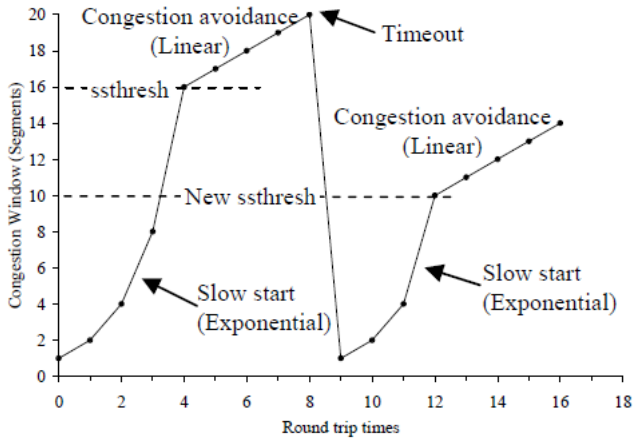
Η αποφυγή της συμφόρησης αποτελείται από 2 κύριες συνιστώσες

1. Μηχανισμός προσδιορισμού της τρέχουσας συμφόρησης
2. Μηχανισμός που επιβραδύνει την ταχύτητα αποστολής σε περίπτωση συμφόρησης

Στην ουσία η αποφυγή συμφόρησης λειτουργεί ως εξής:

Όταν ο αποστολέας αντιληφθεί την πιθανότητα συμφόρησης

- μειώνει τον ρυθμό με το οποίο αποστέλλει τα δεδομένα έτσι ώστε να αποφευχθεί η συμφόρηση.
- θέτει το cwnd στο μισό της τρέχουσας τιμής και
- για κάθε επιβεβαίωση που λαμβάνεται, το πακέτο αυξάνεται κατά cwdn/1 και το μέγεθος του παραθύρου κατά 1,αυξάνεται δηλαδή γραμμικά με το RTT.



Εικόνα 3-2: TCP slow start and congestion avoidance phase πηγή[15] .

Η Εικόνα 3-2 δείχνει τις αλλαγές στο παράθυρο συμφόρησης (cwnd) κατά τη διάρκεια της φάσης αργής εκκίνησης και της φάσης αποφυγής συμφόρησης. Αυτό που συμβαίνει είναι ότι όταν ξεκινά μια νέα σύνδεση, το TCP θέτει το cwnd σε 1 πακέτο και στη συνέχεια το αυξάνει εκθετικά σε 2, 4, 8 και ούτω καθεξής. Η παρούσα μέγιστη τιμή του cwnd είναι 20, επομένως η θέση του νέου ssthresh μετά από timeout τίθεται σε 10. Το TCP λειτουργεί στη φάση αργής εκκίνησης μέχρι το μέγεθος του παραθύρου να φτάσει το ssthresh, το οποίο

είναι 16. Τώρα η φάση αποφυγής συμφόρησης ξεκινά με γραμμική αύξηση του cwnd, και όταν φτάσει στην μέγιστη τιμή του cwnd (δηλαδή 20), συμβαίνει timeout και ξεκινά και πάλι η φάση αργής εκκίνησης, με το νέο ssthresh να είναι 10.

3.4.3 End-to-End Rate-Based Congestion Control

Οι μηχανισμοί ελέγχου συμφόρησης από άκρο σε άκρο (end-to-end) είναι πολύ σημαντικοί για την ανθεκτικότητα και τη σταθερότητα του Διαδικτύου. Ο πιο γνωστός τρόπος για την μεταφορά δεδομένων είναι μέσω TCP. Επομένως, η ύπαρξη "φιλικής προς το TCP" συμπεριφοράς είναι σημαντική για τις νέες εφαρμογές.

Ωστόσο, η εμφάνιση εφαρμογών που δεν ελέγχουν τη συμφόρηση απειλεί τη ομαλή λειτουργία του TCP και την πιθανή κατάρρευση λόγω συμφόρησης.. Γι' αυτό το λόγο έχει δημιουργηθεί ένα Πρωτόκολλο Προσαρμογής Ρυθμού (RAP) που είναι φιλικό προς το TCP και λειτουργεί μέσω ενός αλγορίθμου αύξησης προσθετικού και μείωσης πολλαπλασιαστικού (AIMD). Είναι κατάλληλο για την αναπαραγωγή ροών σε πραγματικό χρόνο μέσω unicast και άλλες αξιόπιστες εφαρμογές. Ο κύριος στόχος του είναι να είναι δίκαιο και φιλικό προς το TCP, ενώ διαχωρίζει τον έλεγχο συμφόρησης του δικτύου από την αξιοπιστία της εφαρμογής.

Αυτό που έχει παρατηρηθεί είναι ότι όταν εφαρμόζεται το πρωτόκολλο RAP το εύρος ζώνης μοιράζεται συνήθως ομοιόμορφα μεταξύ της κίνησης TCP και RAP.

Ο μηχανισμός του πρωτοκόλλου RAP είναι κυρίως υλοποιημένος στην πηγή. Μια πηγή RAP στέλνει πακέτα δεδομένων με αριθμούς ακολουθίας, και ένας αποδέκτης RAP επιβεβαιώνει κάθε πακέτο, παρέχοντας ανατροφοδότηση από άκρο σε άκρο. Κάθε πακέτο επιβεβαίωσης (ACK) περιλαμβάνει τον αριθμό ακολουθίας του αντίστοιχου προδοθέντος πακέτου δεδομένων. Χρησιμοποιώντας την ανατροφοδότηση, η πηγή RAP μπορεί να εντοπίσει απώλειες και να μετρήσει τον χρόνο round-trip (RTT).

Για τον σχεδιασμό ενός μηχανισμού προσαρμογής ρυθμού, πρέπει να αντιμετωπιστούν τρία ζητήματα. Αυτά είναι η συνάρτηση απόφασης, ο αλγόριθμος αύξησης/μείωσης και η συχνότητα απόφασης.

Συνάρτηση απόφασης: Ο τρόπος που λειτουργεί η συνάρτηση απόφασης είναι η εξής: αν εντοπιστεί συμφόρηση μειώνεται ο ρυθμός μετάδοσης. Το RAP λαμβάνει τις απώλειες ως ένδειξη συμφόρησης και έτσι χρησιμοποιεί χρονικά όρια στο χώρο ακολουθίας για να ανιχνεύσει τις απώλειες.

Αλγόριθμος αύξησης/μείωσης αύξησης/μείωσης: Εδώ χρησιμοποιείται ένας αλγόριθμος αύξησης/μείωσης AIMD. Σε περίπτωση που δεν υπάρχει απώλεια πακέτων, ο ρυθμός μετάδοσης αυξάνεται προσθετικά. Ωστόσο, όταν ανιχνεύεται συμφόρηση (όπως μέσω απωλειών ή καθυστερήσεων), ο ρυθμός μετάδοσης μειώνεται πολλαπλασιαστικά.

Συχνότητα απόφασης: Η συχνότητα απόφασης καθορίζει πόσο συχνά πρέπει να αλλάζει ο ρυθμός μετάδοσης. Η βέλτιστη συχνότητα προσαρμογής εξαρτάται από την καθυστέρηση ανατροφοδότησης. Επειδή η καθυστέρηση RTT είναι μια σημαντική παράμετρος, η συχνότητα προσαρμογής του ρυθμού δεν πρέπει να είναι μεγαλύτερη από μία φορά ανά RTT. Αν ο ρυθμός προσαρμογής αλλάζει πολύ συχνά, μπορεί να δημιουργήσει ταλαντώσεις και αστάθεια στον ρυθμό μετάδοσης.

Συνοψίζοντας το RAP είναι ένας σημαντικός και αποδοτικός αλγόριθμος ως αναφορά τη δικαιοσύνη του δικτύου. Καθώς έχει την ικανότητα να προσαρμόζει τον ρυθμό μετάδοσης δεδομένων, ώστε να εκμεταλλεύεται στο μέγιστο τους διαθέσιμους πόρους. Με τον τρόπο αυτό επιτυγχάνει υψηλή απόδοση και σταθερότητα αποφεύγοντας τη συμφόρηση. [16]

3.4.4 Fast Recovery

Όταν το TCP παρατηρεί ότι ένα πακέτο δεδομένων έχει χαθεί και λαμβάνει αναφορές επιβεβαίωσης (acknowledgments) για άλλα πακέτα με μεγαλύτερο αριθμό ακολουθίας από το χαμένο, είναι πιθανό ότι το πακέτο είναι απλά καθυστερημένο και δεν έχει χαθεί πραγματικά. Σε αυτήν την περίπτωση, το TCP υιοθετεί το Fast Recovery για να αποφύγει την πλήρη επαναφορά του παραθύρου (window) και να διατηρήσει την ροή των δεδομένων. Στη διαδικασία Fast Recovery, με τη χρήση του cwnd γίνεται πολλαπλασιαστική μείωση και προσθετική αύξηση. Για να εξασφαλιστεί μια καλή κατανομή πακέτων στο δίκτυο, θα πρέπει το TCP να ακολουθεί ένα νόμο AIMD.

Ο αλγόριθμος *AIMD (Additive Increase, Multiplicative Decrease)* χρησιμοποιείται για να ρυθμίσει την ποσότητα των δεδομένων που στέλνονται από τον αποστολέα σε μια σύνδεση TCP, ώστε να αποφευχθεί η συμφόρηση στο δίκτυο. Η λειτουργία του χωρίζεται σε 2 μέρη:

Additive Increase (Προσθήκη κατά την αύξηση): Όταν δεν παρατηρείται συμφόρηση στο δίκτυο, το μέγεθος του παράθυρου αποστολής αυξάνεται αργά και σταθερά, συνήθως με την προσθήκη ενός πακέτου για κάθε ολοκληρωμένο κύκλο (RTT).

Multiplicative Decrease (Πολλαπλασιαστική μείωση): Όταν εντοπιστεί συμφόρηση στο δίκτυο (π.χ. μέσω απώλειας πακέτων ή λήψης διπλών ACKs), το μέγεθος του παράθυρου αποστολής μειώνεται δραστικά, συχνά κατά το ήμισυ (π.χ. μειώνεται το παράθυρο αποστολής TCP κατά 50%) [17].

Ο αποστολέας χρησιμοποιεί ένα παράθυρο συμφόρησης ή cwnd για να ορίσει τον ρυθμό(=cwnd/RTT). Στη συνέχεια χρησιμοποιεί αργή εκκίνηση για την αύξηση του ρολογιού ACK, με τη μέθοδο της προσθετικής αύξησης. Μετά από time out ο αποστολέας ξεκινάει πάλι αργά με cwnd=1 καθώς δεν έχει ρολόι ACK. Με τη διαδικασία του fast recovery ο αποστολέας μπορεί να επιδιορθώσει την απώλεια ενός τμήματος γρήγορα, συνήθως πριν από ένα time out.

3.4.5 Fast Retransmit

Στη περίπτωση αυτή ο αλγόριθμος αυτός λειτουργεί ως εξής: όταν ο αποστολέας λαμβάνει το 3ο διπλότυπο ACK, υποθέτει ότι το πακέτο έχει χαθεί και μεταδίδει ξανά αυτό το πακέτο χωρίς να περιμένει τη λήξη ενός χρονοδιακόπτη αναμετάδοσης. Μετά την αναμετάδοση, ο αποστολέας συνεχίζει την κανονική μετάδοση δεδομένων. Έτσι μπορεί να διορθωθεί γρήγορα η απώλεια ενός τμήματος πριν από ένα χρονικό όριο.

Η έννοια της συμφόρησης στα δίκτυα υπάρχει από την αρχή της ανάπτυξης των δικτύων. Με την αύξηση της χρήσης των δικτύων και την αυξημένη ζήτηση για μεταφορά δεδομένων, η συμφόρηση έχει γίνει ένα συνηθισμένο πρόβλημα. Για την επίλυση αυτού του προβλήματος απαιτείται η ανάπτυξη νέων τεχνικών και λύσεων για τη διαχείριση του φόρτου κίνησης και την εξασφάλιση υψηλής απόδοσης και ποιότητας υπηρεσιών για τους χρήστες.

Πιο συγκεκριμένα η ιδέα του ελέγχου συμφόρησης TCP για κάθε πηγή προσδιορίζει πόση χωρητικότητα είναι διαθέσιμη στο δίκτυο, ώστε να γνωρίζει πόσα πακέτα μπορεί να μεταφέρει. Το TCP τη χειρίζεται μειώνοντας το μέγεθος του παραθύρου του αποστολέα. Ο αλγόριθμος ελέγχου συμφόρησης TCP περιλαμβάνει διάφορες τεχνικές αύξησης/πολλαπλασιαστικής μείωσης (AIMD), όπως η αργή εκκίνηση και το παράθυρο συμφόρησης (CWND), για την επίτευξη αποφυγής συμφόρησης. Το παράθυρο συμφόρησης (CWND) είναι ένας από τους παράγοντες που καθορίζουν τον αριθμό των byte που μπορούν

να σταλούν ανά πάσα στιγμή. Ο αποστολέας δεν πρέπει να στέλνει δεδομένα μεγαλύτερα από αυτά του μεγέθους του παραθύρου του παραλήπτη. Εάν τα δεδομένα που αποστέλλονται είναι μεγαλύτερα από αυτά του μεγέθους του παραθύρου του δέκτη, τότε προκαλεί αναμετάδοση του TCP λόγω της πτώσης του τμήματος TCP. Ως εκ τούτου, ο αποστολέας θα πρέπει πάντα να στέλνει δεδομένα που είναι μικρότερα ή ίσα με το μέγεθος του παραθύρου του παραλήπτη.

3.5 Network side//διαχείριση ουρών

3.5.1 Congestion Avoidance vs Control

Η διαφορά μεταξύ της αποφυγής συμφόρησης και του ελέγχου συμφόρησης είναι ότι η αποφυγή επικεντρώνεται στην πρόληψη της συμφόρησης πριν αυτή εμφανιστεί, ενώ ο έλεγχος επικεντρώνεται στην αντιμετώπιση της συμφόρησης μόλις αυτή εντοπιστεί.

Το TCP δουλεύει με τον εξής τρόπο όταν πρόκειται για τον έλεγχο συμφόρησης αυξάνει επανειλημμένα το φορτίο του στο δίκτυο για να βρει το σημείο όπου εμφανίζεται συμφόρηση και εγκαταλείπει στη συνέχεια το σημείο αυτό. Δηλαδή το TCP πρέπει να υποστεί απώλειες για να βρει διαθέσιμο εύρος ζώνης στη σύνδεση. Από την άλλη με τη μέθοδο της αποφυγής συμφόρησης , το TCP, προσπαθεί να προβλέψει πότε θα εμφανιστεί συμφόρηση .

Active Queue Management (DECbit, RED, ECN)

Γενικά υπάρχουν δύο διαφορετικές προσεγγίσεις για την αποφυγή της συμφόρησης σε ένα δίκτυο επικοινωνίας. Η πρώτη αναφέρεται στη δυνατότητα του δρομολογητή να βοηθήσει τον τελικό κόμβο να προβλέψει την συμφόρηση. Αυτή η προσέγγιση είναι συχνά αναφέρεται ως *ενεργή διαχείριση ουράς* (AQM). Από την άλλη μεριά η δεύτερη προσέγγιση προσπαθεί να αποφύγει τη συμφόρηση καθαρά από τους τελικούς κόμβους. Η πρώτη προσέγγιση απαιτεί αλλαγές στους δρομολογητές. Η λύση αυτή δεν είναι η πιο αποδεκτή .Καθώς δεν προτιμάμε να κάνουμε αλλαγές στους δρομολογητές. Παρακάτω περιγράφονται κάποιοι βασικοί μηχανισμοί εφαρμόζονται [11] .

3.5.2 DECbit

Ο πρώτος μηχανισμός αναπτύχθηκε για χρήση στο Digital Network Architecture (DNA), ένα δίκτυο χωρίς σύνδεση με πρωτόκολλο μεταφοράς προσανατολισμένο στη σύνδεση. Αυτός ο μηχανισμός θα μπορούσε, επομένως, να εφαρμοστεί και σε TCP και IP. Αυτή η ειδοποίηση υλοποιείται ορίζοντας ένα bit δυαδικής συμφόρησης στα πακέτα που ρέουν μέσω του δρομολογητή. εξ ου και το όνομα DECbit. Στη συνέχεια, ο κεντρικός υπολογιστής προορισμού αντιγράφει αυτό το bit συμφόρησης στο ACK που στέλνει πίσω στην πηγή. Τέλος, η πηγή προσαρμόζει τον ρυθμό αποστολής της έτσι ώστε να αποφεύγεται η συμφόρηση.

3.5.3 Random Early Detection (RED)

Ο δεύτερος μηχανισμός, που ονομάζεται Random Early Detection(RED), είναι παρόμοιος με το DECbit. Κάθε δρομολογητής είναι προγραμματισμένος να παρακολουθεί το μήκος της δικής του ουράς και να ενημερώνει την πηγή ώστε να προσαρμόσει το παράθυρο συμφόρησης όταν εντοπίζει επικείμενη συμφόρηση. Το RED, που σχεδιάστηκε από τους Sally Floyd και Van Jacobson στις αρχές της δεκαετίας του 1990, διαφέρει από DECbit σε δύο βασικά σημεία. Πρώτον, αντί να στέλνει ρητά ένα μήνυμα ειδοποίησης συμφόρησης στην πηγή, το RED συνήθως υλοποιείται για να ενημερώνει έμμεσα την πηγή για συμφόρηση όταν απορρίπτεται κάποιο από τα πακέτα. Το RED έχει σχεδιαστεί για να χρησιμοποιείται με το TCP, το οποίο μπορεί πλέον να έχει timeout ή άλλα μέσα ανίχνευσης απώλειας πακέτων, όπως διπλά ACKs για την ανίχνευση συμφόρησης)-όπως υποδηλώνει το "early" μέρος του ακρωνυμίου RED, η πύλη ενημερώνει την πηγή να μειώσει το παράθυρο συμφόρησης νωρίτερα από το συνηθισμένο, απορρίπτοντας πακέτα νωρίτερα από το συνηθισμένο. Με άλλα λόγια, ο δρομολογητής επιβραδύνει την πηγή απορρίπτοντας μερικά πακέτα πριν εξαντλήσει πλήρως τον αποθηκευτικό χώρο της, έτσι ώστε να μην χρειαστεί να απορρίψει περισσότερα πακέτα αργότερα.

Η δεύτερη διαφορά μεταξύ RED και DECbit έγκειται στις λεπτομέρειες του τρόπου με τον οποίο η RED αποφασίζει πότε και ποια πακέτα θα απορρίψει. Αν λοιπόν θεωρήσουμε μια απλή ουρά FIFO. Αυτό που συμβαίνει στην ουσία είναι αντί να περιμένουμε μέχρι να γεμίσει εντελώς η ουρά και στη συνέχεια να αναγκάσετε όλα τα εισερχόμενα πακέτα να απορριφθούν μπορεί να αποφασιστεί ότι κάθε φορά που το μήκος της ουράς υπερβαίνει ένα ορισμένο επίπεδο απόρριψης , όλα τα εισερχόμενα πακέτα θα απορρίπτονται με μια ορισμένη πιθανότητα απόρριψης. Αυτή η ιδέα ονομάζεται πρώιμη τυχαιοποιημένη

απόρριψη(RED) και ο αλγόριθμος RED καθορίζει τον τρόπο παρακολούθησης του μήκους της ουράς και τις λεπτομέρειες για το πότε απορρίπτονται τα πακέτα.

Οι ουρές που χρησιμοποιούνται στο δίκτυο είναι μέσου μήκους αναμονής και αυτό συμβαίνει διότι η έννοια της συμφόρησης με αυτό τον τρόπο αποτυπώνεται με μεγαλύτερη ακρίβεια. Οι ουρές αναμονής μπορούν να γεμίσουν και να αδειάσουν ξανά σε πολύ σύντομο χρονικό διάστημα εάν στο δίκτυο υπάρχει μεγάλη κινητικότητα .Εάν η ουρά είναι άδεια τον περισσότερο χρόνο, υπάρχει περίπτωση παρερμήνευσης ότι ο δρομολογητής είναι υπερφορτωμένος και ότι ο κεντρικός υπολογιστής πρέπει να επιβραδύνει. Επομένως, ο υπολογισμός του σταθμισμένου τρέχοντος μέσου όρου προσπαθεί να ανιχνεύσει τη μακροπρόθεσμη συμφόρηση φιλτράροντας τις βραχυπρόθεσμες αλλαγές στο μήκος της ουράς.

3.5.4 ECN

Το Explicit Congestion Notification (ECN), που μεταφράζεται ως Ρητή Ειδοποίηση Συμφόρησης, είναι μια τεχνική που χρησιμοποιείται στο πρωτόκολλο δικτύου, κυρίως στο (TCP), για να βελτιώσει τη διαχείριση της συμφόρησης στο δίκτυο. Το ECN επιτρέπει τον ακριβέστερο έλεγχο της κυκλοφορίας δεδομένων και την ανίχνευση της συμφόρησης πριν προκληθούν προβλήματα από απώλειες δεδομένων στο δίκτυο.

Ειδικότερα αυτή η ανάδραση υλοποιείται αντιμετωπίζοντας δύο bit στο TOS πεδίο IP ως bit ECN. Ένα bit ορίζεται από την πηγή για να υποδεικνύει ότι μπορεί να αντιδράσει σε μια ειδοποίηση συμφόρησης. Αυτό ονομάζεται ECTbit (ECN-Capable Transport). Το άλλο bit ορίζεται από τους δρομολογητές κατά μήκος της διαδρομής από άκρο σε άκρο όταν αντιμετωπίζεται συμφόρηση, όπως υπολογίζεται από οποιονδήποτε αλγόριθμο AQM εκτελεί. Αυτό ονομάζεται CEbit.

Επίσης, το ECN περιλαμβάνει την προσθήκη δύο προαιρετικών σημαιών στην κεφαλίδα TCP. Το πρώτο, ECE(ECN-Echo), επικοινωνεί από τον δέκτη στον αποστολέα ότι έχει λάβει ένα πακέτο με το CE σύνολο bit. Το δεύτερο, CWR(Μειωμένο παράθυρο συμφόρησης) επικοινωνεί από τον αποστολέα στον παραλήπτη ότι έχει μειώσει το παράθυρο συμφόρησης. Η υποστήριξη για ECN συνιστάται ιδιαίτερα, αλλά δεν απαιτείται. Επιπλέον, δεν υπάρχει ενιαίος προτεινόμενος αλγόριθμος AQM, αλλά αντίθετα, υπάρχει μια λίστα απαιτήσεων που πρέπει να πληροί ένας καλός αλγόριθμος AQM. Ωστόσο, υπάρχουν ορισμένα σενάρια όπου οι αλγόριθμοι ελέγχου συμφόρησης TCP και οι αλγόριθμοι AQM έχουν σχεδιαστεί για να συνεργάζονται. Αυτό είναι το κέντρο δεδομένων. [13]

3.6 Ο έλεγχος συμφόρησης στις βασικές εκδοχές του TCP

Στη συνέχεια περιγράφονται συνοπτικά οι κύριοι αλγόριθμοι που εισήχθησαν για τον έλεγχο συμφόρησης στο TCP σε μια προσπάθεια να βελτιώσουν την ικανότητα του TCP να διαχειρίζεται τη συμφόρηση του δικτύου.

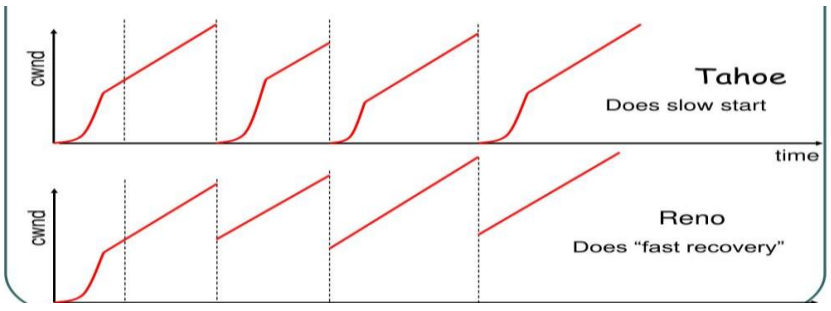
3.6.1 TCP Tahoe/Reno

Το TCP ελέγχει τη μεταφορά δεδομένων έτσι ώστε να είναι αξιόπιστη. Τα πρωτόκολλα Tahoe και Reno θεωρούν το χρονικό όριο αναμετάδοσης (RTO) και τα διπλά ACK ως συμβάντα απώλειας πακέτων, η μόνη διαφορά στη συμπεριφορά τους έγκειται κυρίως ως προς τον τρόπο με τον οποίο αντιδρούν σε διπλότυπα ACK.

Στην περίπτωση του Tahoe εάν ληφθούν τρία διπλά ACK (δηλ. τέσσερα ACK που αναγνωρίζουν το ίδιο πακέτο, τα οποία δεν βασίζονται σε δεδομένα και δεν αλλάζουν το διαφημιζόμενο παράθυρο του δέκτη), το Tahoe εκτελεί γρήγορη αναμετάδοση, ορίζει το όριο αργής εκκίνησης στο ήμισυ του τρέχοντος παραθύρου συμφόρησης, μειώνει το παράθυρο συμφόρησης σε 1 MSS και επαναφέρει την κατάσταση αργής εκκίνησης.

Από την άλλη το Reno εάν ληφθούν τρία διπλά ACK, το Reno θα εκτελέσει μια γρήγορη αναμετάδοση και θα παραλείψει τη φάση αργής εκκίνησης μειώνοντας κατά το ήμισυ το παράθυρο συμφόρησης (αντί να το ρυθμίσει σε 1 MSS όπως το Tahoe), ρυθμίζοντας το ssthresh ίσο με το νέο παράθυρο συμφόρησης και θα εισέλθει σε μια φάση που ονομάζεται γρήγορη ανάκτηση.

Τόσο στο Tahoe όσο και στο Reno, εάν λήξει το χρονικό όριο ACK (χρονικό όριο RTO), χρησιμοποιείται αργή εκκίνηση και οι δύο αλγόριθμοι μειώνουν το παράθυρο συμφόρησης σε 1 MSS.



Εικόνα 3-3 TCP Tachoe vs TCP Reno

3.6.2 TCP NEW RENO

Οι αλγόριθμοι TCP Tahoe και Reno πήραν αναδρομικά το όνομά τους από τις εκδόσεις του λειτουργικού συστήματος 4.3BSD στο οποίο πρωτοεμφανίστηκε ο καθένας (οι οποίοι πήραν το όνομά τους από τη λίμνη Tahoe και την κοντινή πόλη Reno της Νεβάδα). Ο αλγόριθμος Tahoe εμφανίστηκε για πρώτη φορά στο 4.3BSD-Tahoe (το οποίο δημιουργήθηκε για να υποστηρίξει τον μίνι υπολογιστή CCI Power 6/32 "Tahoe"), και αργότερα έγινε διαθέσιμος σε μη-AT&T licensees ως μέρος του 4.3BSD Networking Release. Αυτό εξασφάλισε την ευρεία διανομή και εφαρμογή του. Βελτιώσεις έγιναν στην έκδοση 4.3BSD-Reno και στη συνέχεια κυκλοφόρησαν στο κοινό ως Networking Release 2 και αργότερα ως 4.4BSD-Lite.

Το TCP New Reno, που ορίζεται από το RFC 6582 (το οποίο καταργεί προηγούμενους ορισμούς στα RFC 3782 και RFC 2582), βελτιώνει την αναμετάδοση κατά τη φάση γρήγορης αποκατάστασης του TCP Reno.

Κατά τη διάρκεια της γρήγορης ανάκτησης, για να διατηρηθεί το παράθυρο μετάδοσης γεμάτο, για κάθε διπλότυπο ACK που επιστρέφεται, αποστέλλεται ένα νέο μη απεσταλμένο πακέτο από το τέλος του παραθύρου συμφόρησης.

Η διαφορά από το Reno είναι ότι το New Reno δεν μειώνει κατά το ήμισυ το ssthresh αμέσως, γεγονός που μπορεί να μειώσει το παράθυρο πάρα πολύ εάν προκύψουν πολλαπλές απώλειες πακέτων. Δεν βγαίνει από γρήγορη ανάκτηση και επαναφορά ssthresh μέχρι να αναγνωρίσει όλα τα δεδομένα. Μετά την αναμετάδοση, τα πρόσφατα αναγνωρισμένα δεδομένα έχουν δύο περιπτώσεις:

Πλήρης αναγνώριση: Το ACK αναγνωρίζει όλα τα ενδιάμεσα τμήματα που αποστέλλονται, το ssthresh δεν μπορεί να αλλάξει, το cwnd μπορεί να οριστεί σε ssthresh

Μερική αναγνώριση: Το ACK δεν αναγνωρίζει όλα τα δεδομένα. Αυτό σημαίνει ότι μπορεί να συμβεί άλλη απώλεια, αναμετάδοση του πρώτου μη αναγνωρισμένου τμήματος, εάν επιτρέπεται.

Χρησιμοποιεί μια μεταβλητή που ονομάζεται "ανάκτηση" για να καταγράψει πόσα δεδομένα πρέπει να ανακτηθούν. Μετά από ένα χρονικό όριο αναμετάδοσης, καταγράφει τον υψηλότερο αριθμό ακολουθίας που μεταδίδεται στη μεταβλητή ανάκτησης και εξέρχεται από τη διαδικασία γρήγορης ανάκτησης. Εάν αναγνωριστεί αυτός ο αριθμός ακολουθίας, το TCP επιστρέφει στην κατάσταση αποφυγής συμφόρησης. Όταν δεν υπάρχουν απώλειες πακέτων, αλλά αντίθετα, τα πακέτα αναδιατάσσονται με περισσότερους

από 3 αριθμούς ακολουθίας πακέτων παρουσιάζεται πρόβλημα με το New Reno. Σε αυτή την περίπτωση, το New Reno εισέρχεται λανθασμένα σε γρήγορη ανάκαμψη. Όταν παραδοθεί το αναδιατεταγμένο πακέτο, αποστέλλονται αμέσως διπλές και περιττές αναμεταδόσεις.

Το νέο Reno αποδίδει εξίσου καλά με το SACK σε χαμηλά ποσοστά σφάλματος πακέτων και ουσιαστικά ξεπερνά το Reno σε υψηλά ποσοστά σφάλματος.

3.6.3 TCP WITH SACK

Το TCP μπορεί να εμφανίσει κακή απόδοση όταν χαθούν πολλά πακέτα από ένα παράθυρο δεδομένων. Η Επιλεκτική Επιβεβαίωση (SACK) είναι μια στρατηγική που διορθώνει αυτή τη συμπεριφορά όταν υπάρχουν πολλαπλά χαμένα τμήματα. Με την ενεργοποίηση του μηχανισμού SACK, ο παραλήπτης μπορεί να στείλει στον αποστολέα ακριβείς πληροφορίες για ποια τμήματα δεδομένων έχουν παραληφθεί σωστά, ακόμα και αν υπάρχουν χαμένα τμήματα ενδιάμεσα. Η διαδικασία που ακολουθείται είναι ότι ο παραλήπτης του TCP στέλνει πακέτα SACK στον αποστολέα, ενημερώνοντάς τον για τα δεδομένα που έχουν ληφθεί. Ο αποστολέας στη συνέχεια έχει τη δυνατότητα να επαναμεταδώσει τα τμήματα δεδομένων που έχουν χαθεί. Υπάρχουν κάποια εμπειρικά στοιχεία υπέρ της επιλεκτικής επιβεβαίωσης, τα οποία έχουν δείξει ότι χωρίς την δυνατότητα επιλεκτικής επιβεβαίωσης αυξάνεται σημαντικά ο αριθμό των επαναμεταδοθέντων τμημάτων ,με αποτέλεσμα την υψηλή καθυστέρηση στο δίκτυο. Γι αυτό το λόγο Ο μηχανισμός **SACK** μπορεί να βελτιώσει τη διαχείριση των απωλειών στο TCP, μειώνοντας τις αχρείαστες αναμεταδόσεις και βελτιώνοντας την απόδοση του πρωτοκόλλου[18]

3.6.4 COMBOUT TCP WINDOWS

COMBOUT TCP: Σύνθετο TCP (CTCP) είναι ένας αλγόριθμος της Microsoft που παρουσιάστηκε ως μέρος της στοίβας TCP των Windows Vista και Window Server 2008. Έχει σχεδιαστεί για να προσαρμόζει επιθετικά το παράθυρο συμφόρησης του αποστολέα για να βελτιστοποιεί το TCP για συνδέσεις με προϊόντα μεγάλης καθυστέρησης εύρους ζώνης, προσπαθώντας παράλληλα να μην βλάψει τη δικαιοσύνη (όπως μπορεί να συμβεί με το HSTCP). Είναι επίσης διαθέσιμο για Linux, καθώς και για Windows XP και Windows Server 2003 μέσω επείγουσας επιδιόρθωσης.

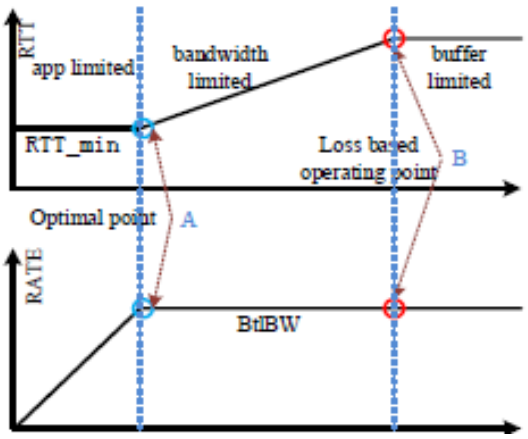
3.6.5 TCP CUBIC

Το CUBIC είναι ένας από τους πιο δημοφιλείς αλγόριθμους ελέγχου/αποφυγής συμφόρησης που χρησιμοποιείται αυτή τη στιγμή. Είναι ένας τυπικός αλγόριθμος ελέγχου συμφόρησης TCP που χρησιμοποιεί μια κυβική συνάρτηση αντί για μια γραμμική συνάρτηση αύξησης παραθύρου συμφόρησης. Ο αλγόριθμος αυτός εφαρμόζεται σε δίκτυα γρήγορων και μεγάλων αποστάσεων με σκοπό τη βελτίωση της επεκτασιμότητας και της σταθερότητας. Το CUBIC έχει υιοθετηθεί ως ο προεπιλεγμένος αλγόριθμος ελέγχου συμφόρησης TCP από τις στοίβες Linux, Windows και Apple. Επίσης, επιτυγχάνει πιο δίκαιη κατανομή εύρους ζώνης μεταξύ ροών με διαφορετικά RTTs και κάνει την ανάπτυξη του παραθύρου συμφόρησης ανεξάρτητη από το RTT. Ένα άλλο χαρακτηριστικό του αλγόριθμου αυτού είναι ότι αυξάνει το μέγεθος του παραθύρου επιθετικά όταν το παράθυρο είναι μακριά από το σημείο κορεσμού και αργά όταν είναι κοντά στο σημείο κορεσμού. Αυτή η δυνατότητα επιτρέπει στο CUBIC να είναι πολύ επεκτάσιμο και ταυτόχρονα εξαιρετικά σταθερό όταν το εύρος ζώνης και το γινόμενο καθυστέρησης του δικτύου είναι μεγάλο.[13].

3.6.6 BBR

Ο αλγόριθμος BBR (Bottleneck Bandwidth and Round-trip propagation time) αναπτύχθηκε στην Google το 2016. Βασίζεται στο εύρος ζώνης συμφόρησης και στο χρόνο μετάδοσης και μετ' επιστροφής (RTT). Ο αλγόριθμος ελέγχου συμφόρησης BBR μετρά το εύρος ζώνης συμφόρησης δικτύου και την ελάχιστη καθυστέρηση σε πραγματικό χρόνο για να υπολογίσει το γινόμενο καθυστέρησης εύρους ζώνης (BDP). Στη συνέχεια προσαρμόζει τον ρυθμό μετάδοσης για να μεγιστοποιήσει την απόδοση και να ελαχιστοποιήσει την καθυστέρηση. Ωστόσο, έχει παρατηρηθεί ότι το BBR εξακολουθεί να έχει ζητήματα όπως η αδικία RTT, ο υψηλός ρυθμός απώλειας πακέτων και η υποβάθμιση της απόδοσης σε μεγάλα buffer. Κάθε επιβεβαίωση παράδοσης πακέτου καταγράφει την ποσότητα των δεδομένων που παραδίδονται κατά το χρονικό διάστημα μεταξύ της μετάδοσης ενός πακέτου δεδομένων και της επιβεβαίωσης αυτού του πακέτου. Όταν εφαρμόστηκε στο You Tube, το BBRv1 απέδωσε κατά μέσο όρο 4% υψηλότερη απόδοση δικτύου και έως και 14% σε ορισμένες χώρες. Το BBR είναι διαθέσιμο για Linux TCP από το Linux 4.9 και διατίθεται επίσης για το QUIC. Το BBRv1 συνυπάρχει καλά με το CUBIC και η έκδοση 2 έχει ως σκοπό να βελτιωθεί έτσι ώστε να περιλαμβάνει πληροφορίες σχετικά με την απώλεια πακέτων και πληροφορίες από ρητή ειδοποίηση συμφόρησης (ECN). Το BBRv2 μπορεί μερικές φορές να έχει χαμηλότερη απόδοση από το BBRv1, θεωρείται γενικά ότι έχει καλύτερη απόδοση. Η έκδοση 3 (BBRv3) διορθώνει δύο σφάλματα στο BBRv2 (πρόωρο

τέλος ανίχνευσης εύρους ζώνης, σύγκλιση εύρους ζώνης) και εκτελεί κάποια ρύθμιση απόδοσης [14].



Εικόνα 3-4 Σημείο λειτουργίας ελέγχου συμφόρησης πηγή[14]

Για τη λειτουργία μιας σύνδεσης TCP με ρυθμό εύρους ζώνης στο σημείο συμφόρησης και με ελάχιστη καθυστέρηση θα πρέπει τα συνολικά δεδομένα σε μετάδοση να είναι ίσα με το γινόμενο εύρους ζώνης-καθυστέρησης (εύρος ζώνης × καθυστέρηση), ή BDP. Πιο συγκεκριμένα αυτό που συμβαίνει στο Bottleneck είναι ότι όταν μία σύνδεση λειτουργεί με τη μέγιστη ροή δεδομένων (throughput) και τη χαμηλότερη καθυστέρηση θα πρέπει να συμβαίνουν τα παρακάτω

1. ισορροπία ρυθμού :η ταχύτητα άφιξης πακέτων στο σημείο συμφόρησης είναι ίση με το BtlBw, και
2. γεμάτος αγωγός: τα συνολικά δεδομένα σε κίνηση είναι ίσα με το BDP (= BtlBw × RTprop). όπου, RTprop (round-trip propagation - χρόνος διάδοσης) και BtlBw (εύρος ζώνης συμφόρησης). Αυτοί οι παράμετροι περιορίζουν την απόδοση της μεταφοράς δεδομένων

Στην πρώτη περίπτωση το σημείο συμφόρησης λειτουργεί αξιόπιστα 100%. Στη δεύτερη περίπτωση παρατηρούμε ότι υπάρχουν αρκετά δεδομένα ώστε να αποφεύγεται η "λιμοκτονία" του σημείου συμφόρησης, χωρίς όμως αγωγός να είναι γεμάτος. Όταν υπάρχει ισορροπία ρυθμού αυτό εξασφαλίζει ότι η ουρά μπορεί να δημιουργηθεί αλλά δεν μπορεί να αλλάξει μέγεθος (π.χ., εάν μια σύνδεση ξεκινήσει στέλνοντας το αρχικό παράθυρο των 10 πακέτων σε ένα BDP 5 πακέτων, και μετά συνεχίσει με τον ακριβή ρυθμό του σημείου

συμφόρησης, τότε 5 από τα 10 πακέτα γεμίζουν τον αγωγό και τα υπόλοιπα σχηματίζουν μια στάσιμη ουρά στο σημείο συμφόρησης που δεν μπορεί να διαλυθεί). Ομοίως, η συνθήκη του γεμάτου αγωγού δεν εγγυάται ότι δεν θα υπάρχει ουρά (π.χ., μια σύνδεση που στέλνει BDP σε εκρήξεις BDP/2 επιτυγχάνει πλήρη αξιοποίηση του σημείου συμφόρησης, αλλά με μέση ουρά BDP/4). Ο μόνος τρόπος για να ελαχιστοποιηθεί η ουρά στο σημείο συμφόρησης και σε όλη τη διαδρομή είναι να πληρούνται και οι δύο συνθήκες ταυτόχρονα.

Οι τιμές του BtlBw και του RTprop αλλάζουν κατά τη διάρκεια μιας σύνδεσης, γι' αυτό θα πρέπει να εκτιμώνται συνεχώς. Το TCP αυτή τη στιγμή παρακολουθεί το RTT (το χρονικό διάστημα από την αποστολή ενός πακέτου δεδομένων μέχρι να επιβεβαιωθεί η λήψη του) επειδή είναι απαραίτητο για την ανίχνευση απώλειας.

Όπου t: RTTt = RTpropt + ηt όπου η ≥ 0 αντιπροσωπεύει τον "θόρυβο" που προκαλείται από τις ουρές στη διαδρομή, Το RTprop μεταβάλλεται μόνο όταν αλλάζει η ίδια η διαδρομή. Επειδή οι μεταβολές στη διαδρομή συμβαίνουν σε χρονικά διαστήματα πολύ μεγαλύτερα από το RTprop, είναι χρήσιμο να υπάρξει ένας εκτιμητής για τις χρονικές στιγμές T [18].

3.7 Σύγκριση αλγορίθμων ελέγχου συμφόρησης

Στον Πίνακα 3-1, συνοψίζουμε τα χαρακτηριστικά των κύριων παραλλαγών του TCP που περιεγράφηκαν παραπάνω [19].

Πίνακας 3-1: Χαρακτηριστικά αλγορίθμων TCP

Algorithm	Year	Features
Tachoe	1988	Slow start, congestion avoidance and a fast retransmit mechanism.
Reno	1990	Loss based, fast recovery mechanism.
New Reno	1999	Fast recovery with retransmissions.

CUBIC	2008	Loss, Slow Start, Congestion Avoidance, Μεγάλο cwnd και η απόδοση βελτιώνεται
BBR	2016	RTT, delay based,
BBR V1	2016	Using RTT, Delay based,
BBRV2	2018	using ECN signals, βελτιωμένη δικαιοσύνη με χαμηλότερους ρυθμούς απώλειας πακέτων

Πίνακας 3-2 Χαρακτηριστικά αλγορίθμων TCP

TCP algorithm Parameters ↓	**RENO**	**NEW RENO**	**TACHOE**	**SACK**	**CUBIC**	**BBR**	**BBR V1**	**BBRV2**
Loss	Yes	Yes	Yes	Yes	Yes	No	No	No
Slow Start	Yes	Yes	Yes	Yes	Yes	No	No	No
Congestion Avoidance	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes
Fast Retransmit	Yes	Yes	Yes	Yes	Yes	No	No	Yes
Fast Recovery	Yes	No	No	No	Yes	No	No	Yes

4 Αξιολόγηση απόδοσης δικτύου σε συνθήκες συμφόρησης

4.1 Μετρικές

Οι μετρικές που χρησιμοποιήσαμε είναι οι παρακάτω:

- **Ρυθμός Αποστολής (Sending Rate):** Υπολογίζουμε τον ρυθμό αποστολής ανά ροή και το συνολικό ρυθμό ως το μέσο bit-rate με βάση το μέγεθος του πακέτου IP σε παραμετροποιήσιμα διαστήματα, όπου το υπολογίζουμε πριν από το σημείο συμφόρησης. Το χρονικό διάστημα Δt πρέπει να είναι αρκετά μικρό για να παρέχει ακριβείς τιμές και αρκετά μεγάλο για να αποφεύγονται οι ταλαντώσεις. Π.χ. αν το Δt είναι τόσο μικρό που φτάνει είτε 1 είτε 0 πακέτα ανά διάστημα, τότε ο υπολογιζόμενος ρυθμός αποστολής θα εναλλάσσεται μεταξύ 0 και του μεγέθους του πακέτου Δt.
- **Ρυθμός μετάδοσης (Throughput) πακέτων/δευτερόλεπτο**:

$$\text{Throughput} = \frac{\frac{W}{RTT} packets}{sec} = \frac{W*MSS}{RTT} bytes/sec$$

Ο νόμος του Little είναι μια θεμελιώδης έννοια στη θεωρία ουράς, που πήρε το όνομά του από τον John D.C. Little. Υπολογίζεται μεταξύ του πλήθους των μη επιβεβαιωμένων πακέτων (W), του μέγιστου μεγέθους τμήματος (MSS), και του χρόνου διαδρομής μετ' επιστροφής (RTT). Ο νόμος αυτός συνδέει τον αριθμό των αντικειμένων (πακέτα) σε ένα σύστημα με τον ρυθμό άφιξής τους και τον χρόνο παραμονής τους στο σύστημα. Στην προκειμένη περίπτωση, το σύστημα είναι το δίκτυο, και ο αριθμός των πακέτων που εκκρεμούν είναι W.

- **Συντελεστή δικαιοσύνης**: Χρησιμοποιούμε τον **Δείκτη Jain** ως συντελεστή δικαιοσύνης με βάση τον ρυθμό αποστολής, για να δείξουμε πόσο δίκαια μοιράζεται το εύρος ζώνης μεταξύ όλων των ροών. Για n ροές, κάθε μία από τις οποίες διαθέτει xi ≥ 0 πόρους, ο τύπος του δείκτη είναι:

$$F = \frac{1}{n} * [\sum_{i=1}^{n} xi]^2 \ / \ \sum_{i=1}^{n} x_i^2$$

Ο δείκτης είναι 1 αν όλες οι ροές λαμβάνουν το ίδιο εύρος ζώνης και 1/n αν μία ροή χρησιμοποιεί όλο το εύρος ζώνης, ενώ οι άλλες δεν λαμβάνουν τίποτα. Ο δείκτης αυτός επιτρέπει την ποσοτικοποίηση της δικαιοσύνης.

- **Χρόνος Διαδρομής (RTT):** Χρησιμοποιούμε την επιλογή **TCP Timestamp** για να μετρήσουμε το τρέχον RTT για κάθε εισερχόμενη αναγνώριση, ο μέσος όρος του RTT υπολογίζεται σε χρονικά διαστήματα.

- **Επαναμεταδόσεις:** μετρά τις επαναμεταδόσεις των τμημάτων TCP στην καταγραφή των πακέτων πριν από το σημείο συμφόρησης.
  Δεδομένα σε μεταφορά (inflight data): αναφέρεται στον αριθμό των bytes που έχουν σταλεί αλλά δεν έχουν ακόμη επιβεβαιωθεί. Λαμβάνουμε αυτή την τιμή υπολογίζοντας τη διαφορά μεταξύ του μέγιστου παρατηρούμενου αριθμού ακολουθίας και των αριθμών επιβεβαίωσης στην καταγραφή πριν από το σημείο συμφόρησης. Αυτή η μετρική είναι χρήσιμη μόνο όταν δεν υπάρχουν επαναμεταδόσεις.
- **Συσσώρευση στον αποθηκευτικό χώρο του σημείου συμφόρησης (bottleneck buffer backlog):** Μετρά το μήκος της συσσώρευσης σε bit σε παραμετροποιήσιμα χρονικά διαστήματα, π.χ. 20 ms.

- **Εσωτερικές τιμές ελέγχου συμφόρησης (congestion control internal values):** οι αλγόριθμοι ελέγχου συμφόρησης διατηρούν διαφορετικές εσωτερικές μετρικές, όπως για παράδειγμα το παράθυρο συμφόρησης ή το όριο της αργής εκκίνησης. Επιπλέον, το bbr παρακολουθεί το εκτιμώμενο εύρος ζώνης του σημείου συμφόρησης και το RTT, καθώς και τους παράγοντες ρυθμού μετάδοσης και μεγέθους παραθύρου. [20]
- **Retransmit rate:** πακέτα που αποστέλλονται ξανά από τον αποστολέα επειδή δεν επιβεβαιώθηκαν από τον παραλήπτη. Αυτό μπορεί να συμβεί για λόγους, όπως συμφόρηση δικτύου, απώλεια πακέτων.
- **Fairness**

  Η κατανόηση του ρυθμού μετάδοσης (throughput) και της δικαιοσύνης των αλγορίθμων ελέγχου συμφόρησης (Congestion Control Algorithm - CCA) προέρχεται κατά ένα μεγάλο βαθμό από μοντέλα και μετρήσεις που (έμμεσα) υποθέτουν ότι η συμφόρηση εμφανίζεται στο τελευταίο κομμάτι του δικτύου. Πρόσφατες μετρήσεις βέβαια δείχνουν ότι η συμφόρηση μπορεί να συμβεί και στον πυρήνα του Διαδικτύου, σε συνδέσεις μεταξύ παροχών, όπου χιλιάδες ροές μοιράζονται συνδέσεις υψηλού εύρους ζώνης. Από μελέτες που έχουν γίνει έχει παρατηρηθεί ότι ο αλγόριθμος BBR επιτυγχάνει ¨καλή¨ δικαιοσύνη. Η δικαιοσύνη είναι από τις πιο σημαντικές ιδιότητες των CCAs, καθώς καθορίζουν την αποτελεσματικότητα με την οποία τα δεδομένα μπορούν να μεταφερθούν μέσω του Διαδικτύου και την ικανότητα πολλαπλών ροών TCP να συνυπάρχουν.

- **Queuing latency:** Η καθυστέρηση στην ουρά (queuing latency) αφορά τον χρόνο που ένα πακέτο παραμένει σε μια ουρά πριν προωθηθεί ή επεξεργαστεί στο δίκτυο. Αυτή η καθυστέρηση εμφανίζεται όταν τα δεδομένα που στέλνονται ξεπερνούν την ικανότητα του δικτύου να τα μεταδώσει ή να τα διαχειριστεί, έτσι ο χρόνος εξυπηρέτησης αυξάνεται ,επομένως και η καθυστέρηση στην ουρά .Ο χρόνος εξυπηρέτησης ισούται με την καθυστέρηση πρόσβασης + καθυστέρηση διάδοσης + καθυστέρηση επεξεργασίας πακέτου + καθυστέρηση αυτόματου αιτήματος επανάληψης (ARQ).

  Bandwidth Delay Product (BDP): γινόμενο που υπολογίζει πόσα bits μπορεί να στείλει ο αποστολέας πριν φτάσει το πρώτο bit στον δέκτη.[18]

4.2 Ρυθμοαπόδοση.

Η απόδοση για το πρωτόκολλο του συρόμενου παραθύρου υπολογίζεται από τον τύπο:

$$\frac{w}{RTT},$$

όπου w είναι το μέγεθος του παραθύρου και RTT ο ελάχιστος χρόνος μετ' επιστροφής. Αποδεικνύεται ότι η πιο ορθή τιμή του παραθύρου W είναι C*RTT, όπου C είναι η χωρητικότητα του συνδέσμου συμφόρησης κατά μήκος της διαδρομής από τον αποστολέα στον δέκτη. Ένα παράθυρο που είναι μεγαλύτερο από το C* RTT (δηλαδή, το γινόμενο καθυστέρησης - εύρους ζώνης ή BDP) θα συμβάλει μόνο στην καθυστέρηση της ουράς, χωρίς να αυξήσει την απόδοση.

BDP ή Bandwidth Delay Product ονομάζεται η μέγιστη ποσότητα δεδομένων που μπορεί να μεταδοθεί σε μια διαδρομή δικτύου οποιαδήποτε στιγμή. Η εκτίμηση του BDP είναι δύσκολη διαδικασία καθώς οι τιμές των παραμέτρων C και RTT μπορεί να διαφέρουν ανάλογα με τις συνθήκες δικτύου στον αποστολέα και τον δέκτη και τη διαδρομή μεταξύ τους. Επίσης διαφέρουν κατά τη μεταφορά δεδομένων μεταξύ του αποστολέα και του παραλήπτη.

Το C ανάλογα με το πόσο μακριά είναι ο χρήστης από το σημείο πρόσβασης WiFi ή τον σταθμό βάσης κινητής τηλεφωνίας και το RTT εξαρτάται από το εάν ο αποστολέας ή ο δέκτης είναι σε σύνδεση WiFi, Ethernet ή 4G, επειδή κάθε τεχνολογία σύνδεσης έχει διαφορετικές καθυστερήσεις μετάδοσης.

Τέλος, το RTT μπορεί να αλλάξει λόγω αλλαγών στη διαδρομή δικτύου μεταξύ του αποστολέα και του παραλήπτη (π.χ. ένας δρομολογητής αφαιρέθηκε για συντήρηση). Συνοπτικά, ούτε το C ούτε το RTT είναι γνωστά εκ των προτέρων και η ρύθμιση του μεγέθους παραθύρου του πρωτοκόλλου συρόμενου παραθύρου στο BDP δεν είναι πρακτική. Ένας άλλος παράγοντας που μπορεί να προκαλέσει σοβαρά προβλήματα απόδοσης είναι η ρύθμιση του παραθύρου σε λανθασμένη τιμή.

Αυτό συμβαίνει :

Α)εάν το μέγεθος του παραθύρου είναι μεγαλύτερο από το BDP, καθώς έτσι αυξάνεται η καθυστέρηση.(αυτή η καθυστέρηση μπορεί να βλάψει τη δια δραστικότητα για μια ροή που ενδιαφέρεται για τον λανθάνοντα χρόνο μεμονωμένων πακέτων (π.χ. η τερματική εφαρμογή SSH που εκτελείται μέσω TCP),

Β) όταν αυξάνεται η καθυστέρηση στην ουρά, αυξάνεται επίσης ο κίνδυνος λανθασμένου χρονικού ορίου αναμετάδοσης.

Στην πραγματικότητα αυτό που συμβαίνει είναι ότι το πακέτο που αναμεταδίδεται δεν έχει χαθεί ακόμα. είναι απλά κολλημένο σε μια μεγάλη ουρά. Το αποτέλεσμα είναι ένα αντίγραφο του πακέτου. Εάν αυτό συμβεί αρκετά, η απόδοση του επιπέδου μεταφοράς μπορεί να μειωθεί δραστικά επειδή παραδίδονται πολλαπλά αντίγραφα πακέτων στον δέκτη, γεγονός που σπαταλά τη χωρητικότητα της ζεύξης.

Σε αυτό το σενάριο πολλαπλών ροών, είναι δύσκολο για οποιαδήποτε ροή να γνωρίζει με ακρίβεια το πλήθος των ροών (n) . Αυτό καθιστά πιο πιθανό κάθε ροή να χρησιμοποιεί τη λάθος τιμή του W. Η χρήση ενός μεγάλου W οδηγεί σε καθυστερήσεις ουράς ή αναμεταδόσεις. Η χρήση ενός μικρού W οδηγεί σε υπό αξιοποίηση των πόρων του δικτύου. Όλα αυτά συμβαίνουν αν υποθέσουμε ότι έχουμε ρυθμίσει λάθος το μέγεθος του παραθύρου για μια μόνο ροή δεδομένων στη συνέχεια θα δούμε τι συμβαίνει όταν έχουμε περισσότερες ροές στον ίδιο σύνδεσμο συμφόρησης, κάτι που είναι πιο σύνηθες και πιο ρεαλιστικό, εδώ τα προβλήματα επιδεινώνονται.

4.3 Κατάρρευση συμφόρησης

Η χρήση ενός W που είναι πολύ μεγάλο (ανεξάρτητα από το αν ένας αποστολέας επιλέγει ένα μεγάλο W ή αν είναι επειδή υπάρχουν πάρα πολλοί αποστολείς) οδηγεί σε κατάρρευση συμφόρησης. Η κατάρρευση συμφόρησης είναι ένας όρος για μια κατάσταση στην οποία το προσφερόμενο φορτίο, δηλαδή η ζήτηση για τις υπηρεσίες του δικτύου, αυξάνεται, αλλά η συνολική χρησιμότητα του δικτύου στους χρήστες του μειώνεται.

Παρακάτω παρουσιάζονται 3 παραδείγματα κατάρρευσης λόγω συμφόρησης:

Α) Παράδειγμα 1ο

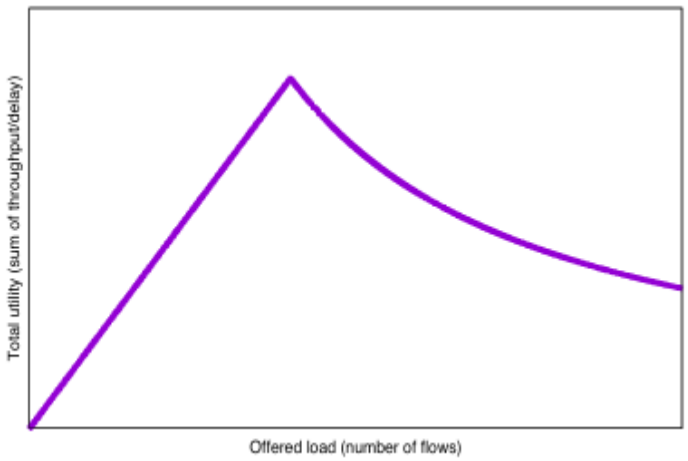
Έστω n ροές χρησιμοποιούν το κυλιόμενο παράθυρο. Η συνολική χρηστικότητα δίνεται από τον τύπο [5]:

$$\frac{C^2}{nW}$$

Αν σχεδιάσουμε τη συνολική χρησιμότητα ως συνάρτηση του n, βλέπουμε ένα γράφημα όπως αυτό στην Εικόνα 4.1.

Το χαρακτηριστικό γνώρισμα της κατάρρευσης συμφόρησης είναι πώς η χρηστικότητα ανεβαίνει μέχρι ένα σημείο και στη συνέχεια κατεβαίνει ή «καταρρέει» απότομα όταν το προσφερόμενο το φορτίο αυξάνεται περαιτέρω. Στο καθεστώς στο οποίο υπάρχει

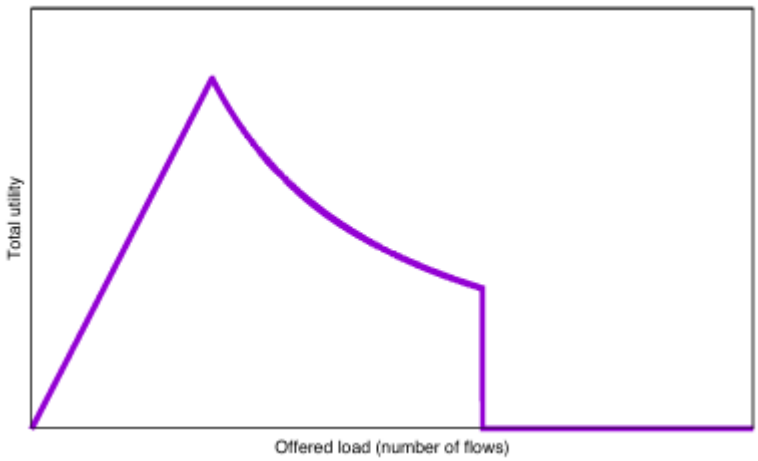
κατάρρευση συμφόρησης, το δίκτυο εξακολουθεί να λειτουργεί. Για παράδειγμα, στο παραπάνω παράδειγμα, ο σύνδεσμος bottleneck εξακολουθεί να χρησιμοποιείται 100% και να παραδίδει πακέτα σε κάθε στιγμή. Όμως, δεν συμβάλει θετικά στην απόδοση του δικτύου. Συγκεκριμένα, σε αυτό το παράδειγμα, υπάρχει φθίνουσα χρηστικότητα για τον χρήστη από την καθυστερημένη παράδοση πακέτων, επειδή η χρησιμότητα είναι αντιστρόφως ανάλογη με τη μέση καθυστέρηση πακέτων.



Εικόνα 4-1 απλό παράδειγμα κατάρρευσης συμφόρησης όπου η χρησιμότητα κάθε ροής είναι ίση με την απόδοση/λανθάνουσα κατάσταση(throughput/latency) υπό την υπόθεση ότι οι σύνδεσμοι έχουν άπειρους Buffers.

Β) Παράδειγμα 2ο

Μπορούμε να τροποποιήσουμε τη χρηστικότητα ώστε να είναι ακόμη πιο ευαίσθητη στην καθυστέρηση του πακέτου. Για παράδειγμα, θα μπορούσαμε να δηλώσουμε ότι εάν η μέση καθυστέρηση πακέτων είναι μεγαλύτερη από ένα ορισμένο όριο, η χρησιμότητα για τον χρήστη είναι μηδέν, ανεξάρτητα από το ποια είναι η απόδοση. Μια τέτοια βοηθητική λειτουργία συλλαμβάνει τις προτιμήσεις των χρηστών του προγράμματος περιήγησης, οι οποίοι περιστασιακά θα εγκαταλείψουν μια αναζήτηση στον ιστό εάν διαρκέσει περισσότερο από μερικά χιλιοστά του δευτερολέπτου. Σε τέτοιες περιπτώσεις, υπάρχει μια ακόμη πιο έντονη μορφή κατάρρευσης συμφόρησης (Εικόνα 4-2), όπου η χρησιμότητα πέφτει στο μηδέν μόλις ο αριθμός των n $\frac{nw}{c} > th$, δηλαδή, η μέση καθυστέρηση υπερβεί το th.



Εικόνα 4-2 Ιδιο παράδειγμα με την εικόνα 4-1 υπό την υπόθεση ότι η χρησιμότητα του είναι μηδέν αν ξεπεραστεί ένα συγκεκριμένο όριο καθυστέρησης

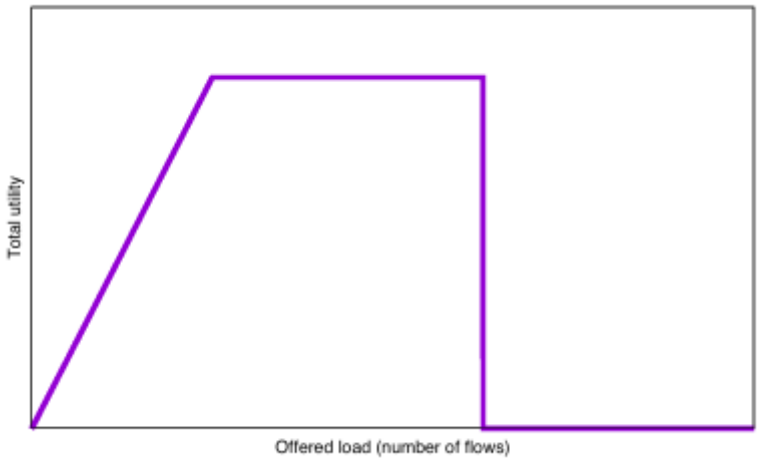
Γ) Παράδειγμα 3ο

Στο τρίτο παράδειγμα κατάρρευσης συμφόρησης εξετάζονται οι αναμεταδόσεις. Σε αυτό το παράδειγμα, η κατάρρευση συμφόρησης θα συμβεί επειδή τα πακέτα αναμεταδίδονται πολλές φορές και λαμβάνονται πολλαπλά αντίγραφα πακέτων στον δέκτη. Αντί να αφαιρέσουμε όλη τη λογική αναμετάδοσης από το πρωτόκολλο συρόμενου παραθύρου, θα υποθέσουμε ότι τα πακέτα αναμεταδίδονται μετά από ένα σταθερό χρονικό όριο T, όπου T > RTT, τροποποιώντας τη συνάρτηση χρηστικότητας, ώστε να αντικατοπτρίζει μόνο τη διεκπεραίωση του επιπέδου μεταφοράς της ροής, όχι την καθυστέρηση. Επειδή όμως τα πακέτα μπορούν να αναμεταδοθούν από τον αποστολέα, είναι πιθανό να υπάρχουν πολλαπλά αντίγραφα του ίδιου πακέτου που παραλαμβάνεται από τον δέκτη. Σε τέτοιες περιπτώσεις, θα μετράμε αυτά τα αντίγραφα μόνο μία φορά. Με άλλα λόγια, εμείς ενδιαφέρονται για τον αριθμό των μοναδικών πακέτων που λαμβάνονται κατά σειρά ανά δευτερόλεπτο.

$$n < \frac{CRTT}{W}$$

Και πάλι, θα δούμε τι συμβαίνει καθώς αυξάνουμε το προσφερόμενο φορτίο: ο αριθμός των ροών n. Δεν υπάρχει καθυστέρηση στην ουρά και, ως εκ τούτου, δεν υπάρχουν χρονικά όρια ή αναμεταδόσεις έτσι η συνολική χρησιμότητα σε αυτό το καθεστώς αυξάνεται γραμμικά με το n όπως και με τα δύο προηγούμενα παραδείγματα. Τα χρονικά όρια που

σημειώθηκαν εξακολουθούν να βρίσκονται στην ουρά και τελικά θα παραδοθούν (ανάκληση του συνδέσμου δεν ρίχνει κανένα πακέτο).



Εικόνα 4-3 Κατάρρευση συμφόρησης όταν η χρησιμότητα εξαρτάται από τον αριθμό των μοναδικών πακέτων που παραδίδονται κατά σειρά και υπάρχουν αναμεταδόσεις στο δίκτυο

Ταυτόχρονα, επειδή τα buffer της σύνδεσης δεν απορρίπτουν πακέτα, τα πακέτα που θα αναμεταδοθούν θα παραδοθούν επίσης. Ως αποτέλεσμα, τα πακέτα που παραδίδονται από τη σύνδεση είναι ένας συνδυασμός των αρχικών πακέτων και των διπλών αναμεταδόσεών τους, πράγμα που σημαίνει ότι η συνολική απόδοση του επιπέδου μεταφοράς θα είναι αυστηρά μικρότερη από το C. Πόσο λιγότερο; Η υποβάθμιση της συνολικής απόδοσης του επιπέδου μεταφοράς εξαρτάται από το πόσο μεγάλος είναι ο λανθάνων χρόνος της ουράς και, επομένως, από τον αριθμό των αναμεταδόσεων. Αυτό, με τη σειρά του, εξαρτάται από το προσφερόμενο φορτίο n. Καθώς το n μεγαλώνει, η συνολική απόδοση του στρώματος μεταφοράς πέφτει. Με άλλα λόγια, υπάρχει ένας βρόχος θετικής ανάδρασης, ο οποίος σύντομα οδηγεί στην αναμετάδοση των περισσότερων πακέτων σε σημείο όπου η χρήσιμη απόδοση είναι σχεδόν μηδενική.

5 Πειραματική αξιολόγηση

Περιπτώσεις Μελέτης

Σε αυτή την ενότητα, θα μελετήσουμε τη συμπεριφορά δύο βασικών τεχνικών αντιμετώπισης της συμφόρησης στο TCP, και συγκεκριμένα το TCP Cubic και το TCP BBR με τη βοήθεια προσομοίωσης και την αναπαραγωγή των αποτελεσμάτων από την εργασία των Scholz et al.[[23], χρησιμοποιώντας το περιβάλλον εξομοίωσης Mininet. Ο κύριος στόχος είναι η σε βάθος κατανόηση της λειτουργίας των πρωτοκόλλων αντιμετώπισης της συμφόρησης και η λήψη μετρήσεων για την αξιολόγησή τους. Σύμφωνα με τις επιστημονικές μελέτες, η αναδημιουργία και επανάληψη των πειραμάτων είναι μια εκπαιδευτική διαδικασία με πολύ σημαντικό ρόλο [29].

5.1 Αναπαραγωγή ερευνητικών δημοσιευμένων αποτελεσμάτων

Στη συγκεκριμένη μελέτη χρησιμοποιούμε την υποδομή που προτείνεται στην εργασία [29] και συγκεκριμένα το περιβάλλον εξομοίωσης Mininet και αναπαράγουμε τα σενάρια εκτέλεσης διαφόρων αλγορίθμων ελέγχου συμφόρησης. Στόχος είναι να κατανοήσουμε τη συμπεριφορά τους και να διαπιστώσουμε κατά πόσο τα αποτελέσματα συμφωνούν με αυτά των συγγραφέων χρησιμοποιώντας διάφορες μετρικές όπως που αφορούν τη δικαιοσύνη, την απόδοση και την καθυστέρηση.

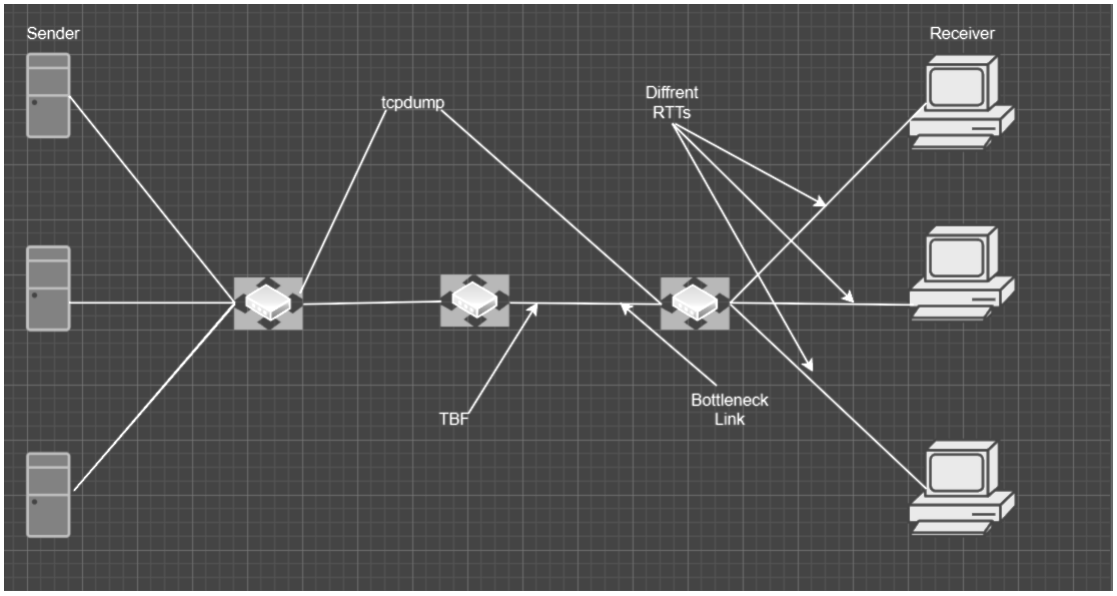
5.1.1 Περιβάλλον προσομοίωσης

Απαραίτητο Λογισμικό

Για την εκτέλεση των πειραμάτων δικτύου και την προσομοίωση των TCP ροών, χρησιμοποιήσαμε το Mininet, μέσω μια εικονικής μηχανής VirtualBox. Το Mininet είναι ένα open source εργαλείο που δημιουργεί ρεαλιστικά εικονικά δίκτυα. Οι κεντρικοί υπολογιστές Mininet χρησιμοποιούν λογισμικό δικτύου Linux. Αυτά τα εργαλεία συνέβαλαν στην αποδοτική διαχείριση του δικτύου και στην προσομοίωση σε εικονικά περιβάλλοντα. Ο υπολογιστής που χρησιμοποιήσαμε έχει τα εξής χαρακτηριστικά CPU Intel Core i3, με συχνότητα 3.60GHz και RAM 16,0GB.

Τοπολογία

Το σύστημα βασίζεται στην τοπολογία Dumbbell (Εικόνα 5-1), η οποία λειτουργεί ως εξής: για κάθε ροή TCP προστίθεται ένα νέο ζευγάρι υπολογιστών, δηλαδή ένας αποστολέας και ένας παραλήπτης, έτσι ώστε να διευκολυνθεί η συλλογή δεδομένων ανά ροή. Και οι δύο πλευρές συνδέονται μέσω τριών μεταγωγέων. Ο κεντρικός μεταγωγέας αποτελεί σημείο συμφόρησης, και είναι αυτός που πραγματοποιεί έλεγχο κίνησης στη διεπαφή του. Οι δύο υπόλοιποι μεταγωγής επιτρέπουν την παρακολούθηση της κίνησης πριν και μετά τον έλεγχο. Η κίνηση από τους παραλήπτες προς τους αποστολείς δεν περιορίζεται, καθώς τα δεδομένα στέλνονται μόνο από τους αποστολείς. Η επιστρεφόμενη ροή αναγνωρίσεων δεν υπερβαίνει το εύρος ζώνης του σημείου συμφόρησης. Όπως φαίνεται και στην εικόνα 5-1 χρησιμοποιείται το NetEm για προστεθεί συγκεκριμένη καθυστέρηση ανά ροή στις συνδέσεις μεταξύ του διακόπτη και των αντίστοιχων παραληπτών, επιτρέποντας ρυθμιζόμενους χρόνους καθυστέρησης (RTTs). Για τον περιορισμό του ρυθμού και τη ρύθμιση του μεγέθους του buffer, χρησιμοποιήθηκε το Token-Bucket Filter (TBF) του Linux. Έτσι επιτρέπεται η συσσώρευση μιας σειράς ρυθμιζόμενων tokens όταν δεν χρειάζονται. Ορίζεται αυτό το μέγεθος του token bucket ώστε να χωράει μόνο ένα πακέτο, επειδή η υπέρβαση του εύρους ζώνης του σημείου συμφόρησης, έστω και για λίγο, επηρεάζει την ικανότητα του BBR να εκτιμά σωστά το εύρος ζώνης συμφόρησης.



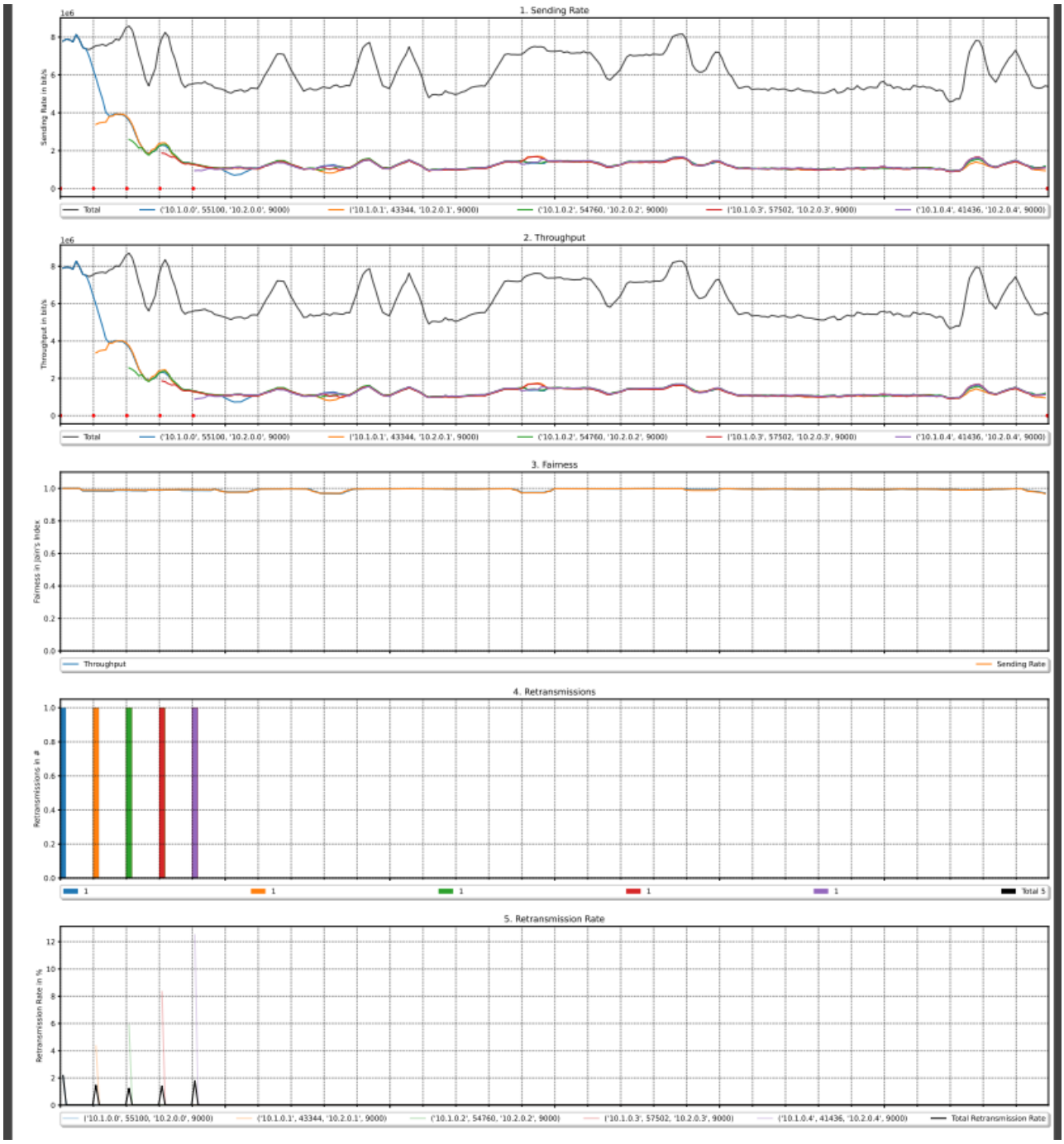
Εικόνα 5-1 Mininet setup with sending and receiving hosts and bot- thence link.

Ροή εργασίας: κάθε πείραμα ελέγχεται μέσω ενός αρχείου διαμόρφωσης που περιγράφει τις ροές. Για κάθε ροή πρέπει να καθοριστεί ο επιθυμητός αλγόριθμος ελέγχου συμφόρησης TCP, ο χρόνος εκκίνησης σε σχέση με την προηγούμενη ροή, το RTT και η διάρκεια της.

Το πλαίσιο ανάλυσης παράγει αρχεία .csv για κάθε υπολογιζόμενο δείκτη, για να την επεξεργασία των δεδομένων. Επιπλέον, δημιουργείται ένα γράφημα που απεικονίζει όλα τα συλλεγμένα δεδομένα με τη μορφή αρχείου .pdf. Μόλις ολοκληρωθεί το πείραμα, δημιουργείται αυτόματα μια αναφορά που περιλαμβάνει 9 γραφήματα που απεικονίζουν τις μετρικές με την πάροδο του χρόνου.

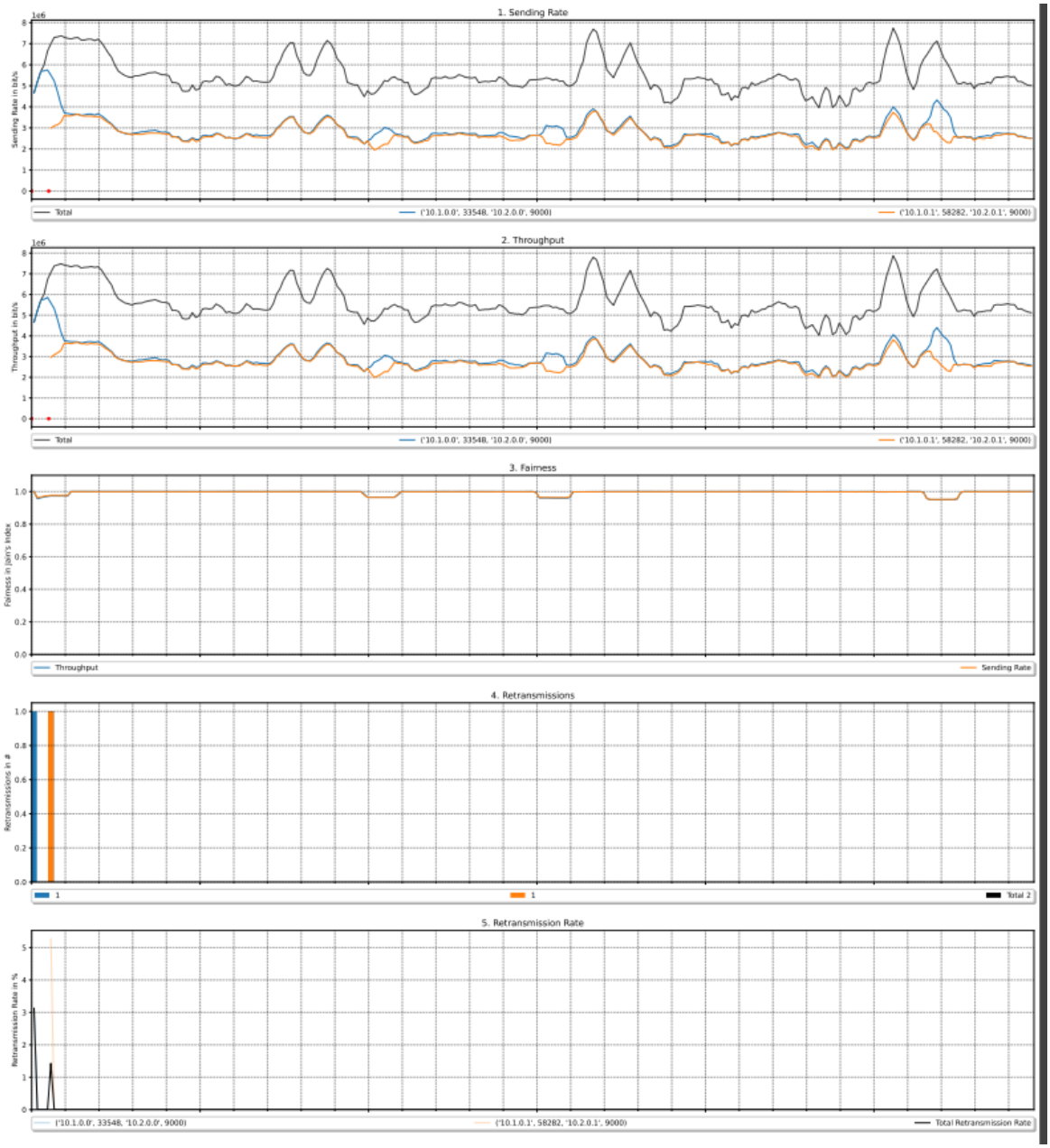
Αποτελέσματα

Παρακάτω παρουσιάζονται τα αποτελέσματα από την αναπαραγωγή των πειραμάτων της εργασίας των Scholz. Στις 2 πρώτες εικόνες παρουσιάζονται τα αποτελέσματα από το πρώτο παράδειγμα που εφαρμόζεται ο αλγόριθμος BBR . Στη συνέχεια παρουσιάζονται τα αποτέλεσμα από το δεύτερο παράδειγμα όπου εφαρμόζονται οι αλγόριθμοι BBR και CUBIC ώστε να συγκριθεί ο τρόπος που λειτουργεί το καθένα.



Εικόνα 5-2 1ο παράδειγμα

Εικόνα 5-3 1ο παράδειγμα συνέχεια



Εικόνα 5-4 2ο παράδειγμα

Εικόνα 5-5 2ο παράδειγμα συνέχεια

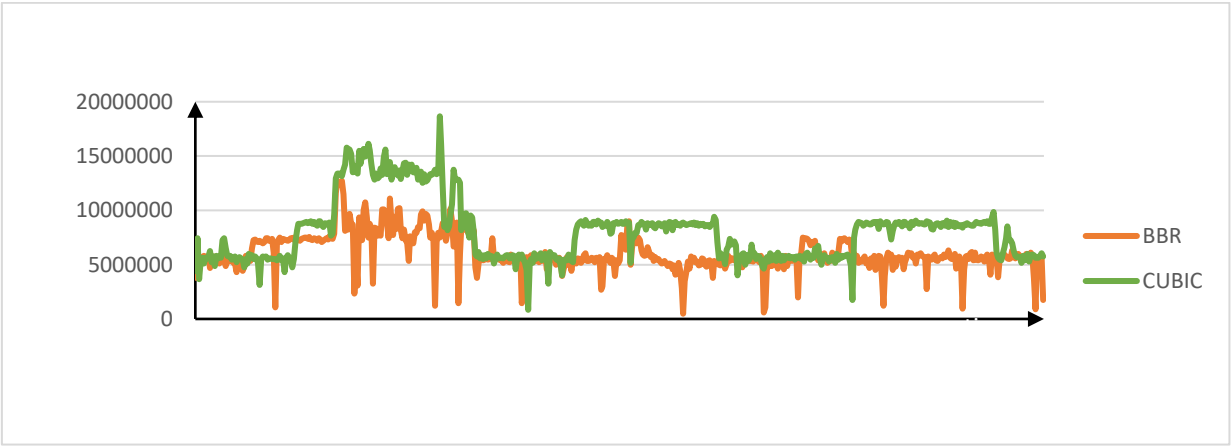
Συγκρίνοντας τα αποτελέσματα της δικής μας εργασίας και της εργασίας που μελετήσαμε, παρατηρούμε ότι είναι παρόμοια μεταξύ τους, όχι όμως εντελώς ίδια. Αυτό είναι λογικό αν σκεφτούμε ότι εργαστήκαμε σε διαφορετικά συστήματα με διαφορετικά χαρακτηριστικά.

5.2 Τροποποίηση σεναρίων προσομοίωσης

Για να κατανοήσουμε τη συμπεριφορά των πρωτοκόλλων BBR και CUBIC τροποποιούμε τις συνθήκες εκτέλεσης των πειραμάτων με τη βοήθεια των παραμέτρων. Συγκεκριμένα εργαστήκαμε σε τέσσερα διαφορετικά σενάρια: α) εξετάσαμε συγκριτικά τη συμπεριφορά των δύο πρωτοκόλλων β) μεταβάλλουμε το εύρος ζώνης γ) μεταβάλουμε το ρυθμό απώλειας πακέτων δ) μεταβάλουμε το μέγεθος του καταχωρητή (buffer size).

5.2.1 Σενάριο 1ο Σύγκριση BBR και CUBIC

Αρχικά, συγκρίνουμε τη συμπεριφορά των αλγορίθμων BBR και CUBIC, ώστε να εξετάσουμε τη διεκπεραιωτή τους ικανότητα (throughput) και την από άκρο σε άκρο καθυστέρηση (RTT) σε συνθήκες συμφόρησης. Σε κάθε περίπτωση θεωρούμε πως στέλνονται απλές ροές.



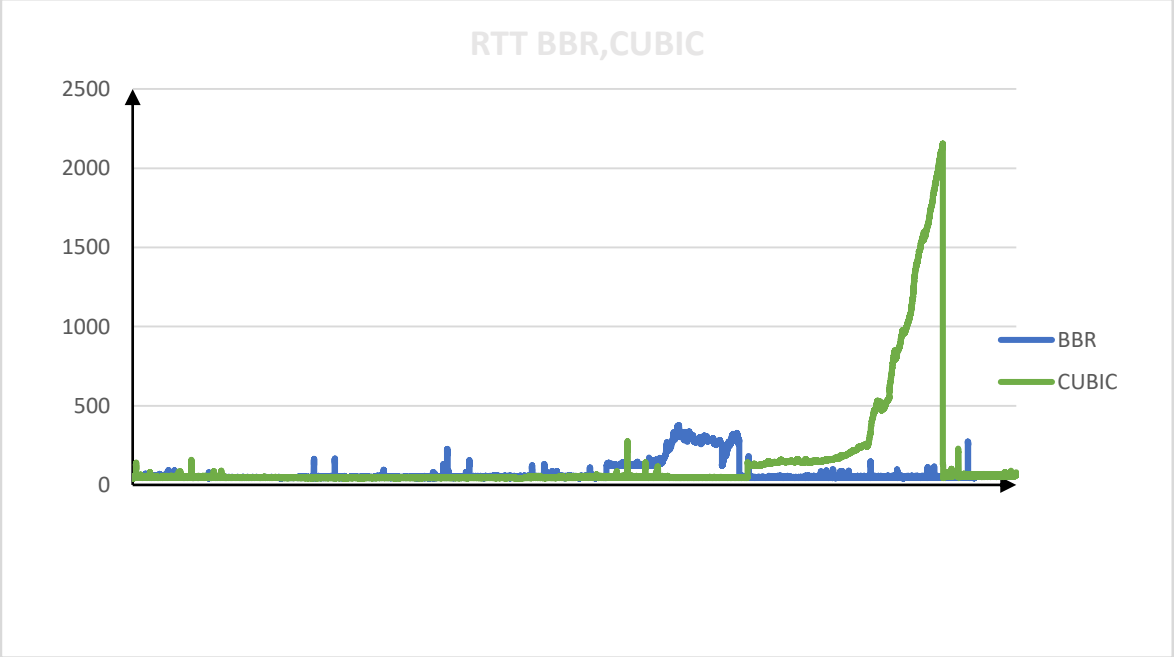
Εικόνα 5-6 Σύγκριση throughput για BBR,CUBIC

Συγκεκριμένα, και όπως φαίνεται στον Πίνακα 5-1, τη χρονική στιγμή "0"μια ροή διάρκειας 110 s εισάγεται στο δίκτυο. Η μέση καθυστέρηση (RTT) ορίζεται στα 40ms. Μετά από 18s το εύρος ζώνης στο σημείο συμφόρησης, ορίζεται σε 20Mbit, και 18s αργότερα σε 10Mbit. Μετά από 20s η μέση καθυστέρηση (RTT) ορίζεται στα 80ms, και λίγο αργότερα (20s) ορίζεται σε 0ms.

Πίνακας 5-1: Ορισμός παραμέτρων 1ου Σεναρίου.

Αρχικές Τιμές	**Συνθήκες συμφόρησης**
Initial Bandwidth: 10Mbit	host, bbr, 40ms, 0.0, 110.0
Burst Buffer: 1600b	link, bw, 20mbit, 18.0
Buffer Latency: 100ms	link, bw, 10mbit, 18.0
Initial Link RTT: 0ms	link, RTT, 80ms, 20.0
Initial Link Loss: 10%	link, RTT, 0ms, 20.0

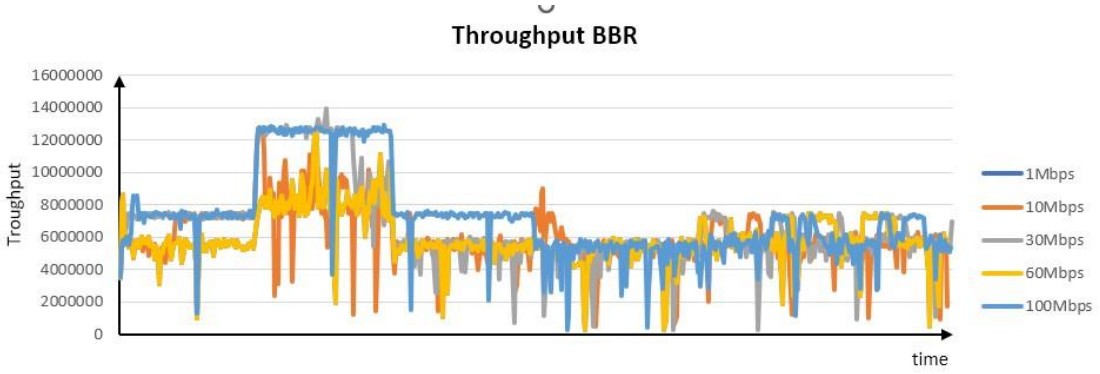
Στην Εικόνα 5-3 παρουσιάζεται η μέση καθυστέρηση (RTT) για την εφαρμογή των δύο αλγορίθμων. Αυτό που παρατηρείται είναι ότι στο CUBIC, το RTT αρχικά είναι σχετικά χαμηλό και όσο εξελίσσεται ο χρόνος, αυξάνεται απότομα. Αμέσως μετά εμφανίζεται δραστική μείωση. Ενώ στο BBR το RTT έχει αυξομειώσεις αλλά με μικρότερο πλάτος και γενικά εμφανίζει μία σταθερότητα.



Εικόνα 5-7 Σύγκριση RTT για BBR,CUBIC

5.2.2 Σενάριο 2ο Σύγκριση παραμέτρων μετατρέποντας τη μεταβλητή bandwidth

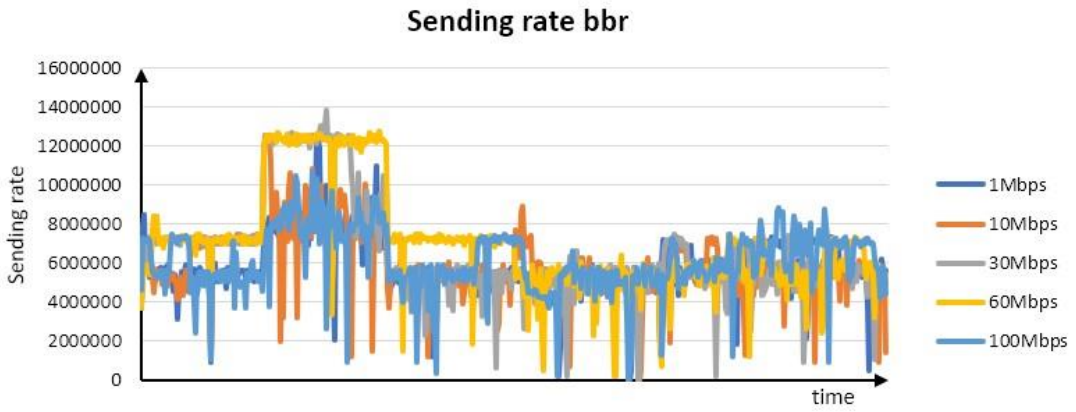
Στη συνέχεια παρουσιάζουμε τα υπόλοιπα σενάρια στα οποία οι συνθήκες συμφόρησης είναι οι ίδιες με αυτές στο 1ο σενάριο.



Εικόνα 5-8 Troughput BBR

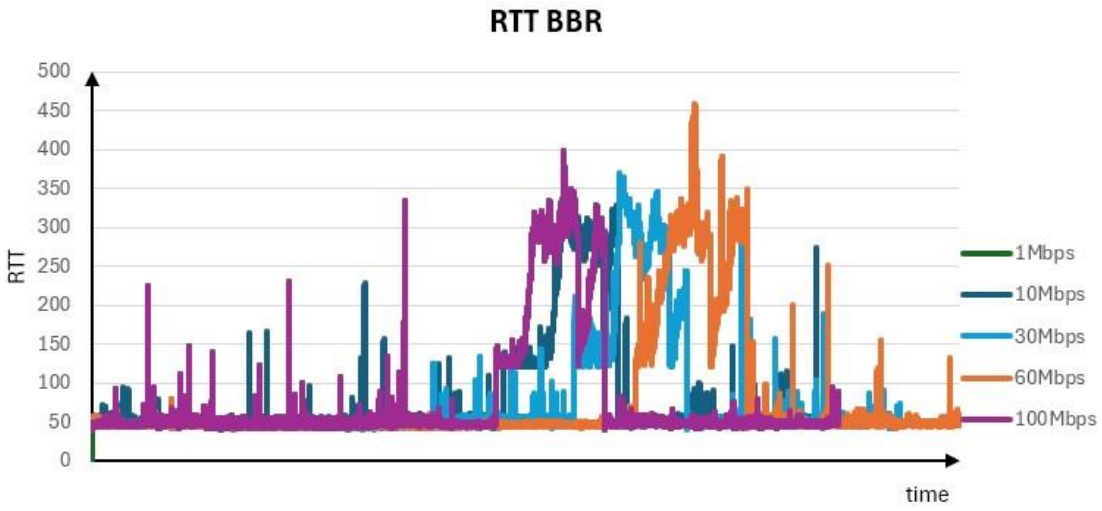
Η διαδικασία που ακολουθούμε σε αυτό το σενάριο είναι να τροποποιούμε το bandwidth και να κρατάμε σταθερές τις τιμές των υπόλοιπων παραμέτρων ώστε να κατανοήσουμε τον τρόπο συμπεριφοράς του BBR σχετικά με διάφορες μετρικές του συστήματος μας.

Σε αυτό το σενάριο παρουσιάζουμε το throughput με τη χρήση του πρωτοκόλλου BBR, μεταβάλλοντας το bandwidth από 1Mbps έως 100 Mbps. Στην Εικόνα (5-4) παρουσιάζονται τα αποτελέσματα για bandwidth ίσο με 1, 10, 30, και 100 Mbps. Αυτό που παρατηρείτε είναι ότι για όλες τις μεταβολές του bandwidth το throughput συμπεριφέρεται με τον ίδιο τρόπο. Δηλ. στην αρχή αυξάνεται μετά είναι σταθερό και μετά από λίγο μειώνει προσωρινά την ταχύτητα αποστολής.



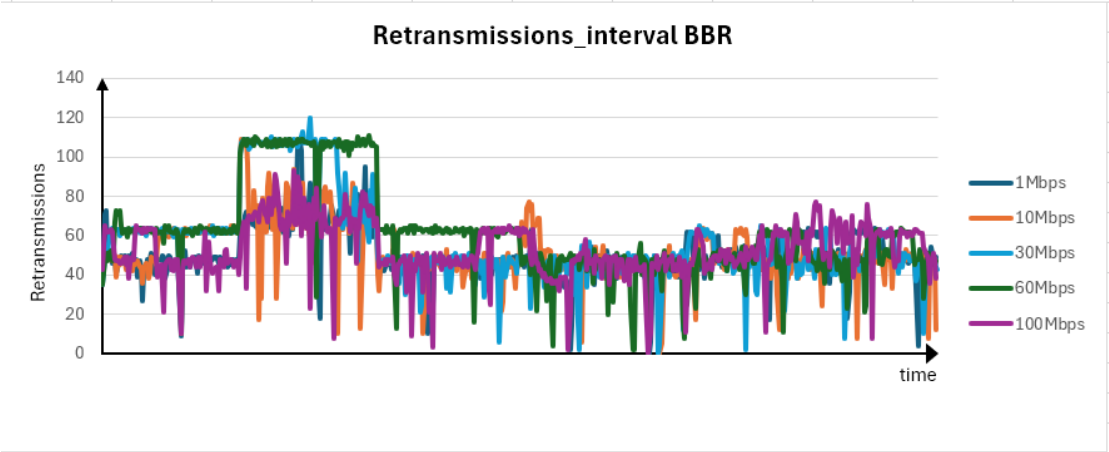
Εικόνα 5-9 Seding Rate(bandwidth)

Σε αυτή την περίπτωση παρουσιάζουμε το Sending rate για τοBBR , μεταβάλλοντας το bandwidth. Αυτό που παρατηρείτε είναι πώς ο ρυθμός αποστολής του BBR προσαρμόζεται με βάση το διαθέσιμο εύρος ζώνης κατά τη διάρκεια του χρόνου. Συγκεκριμένα το BBR προσαρμόζει το ρυθμό αποστολής του για να αξιοποιήσει την διαθέσιμη χωρητικότητα του δικτύου, χωρίς να προκαλεί συμφόρηση. Το BBR αρχικά αυξάνει τον ρυθμό αποστολής ώστε να βρει το μέγιστο διαθέσιμο εύρος ζώνης (BtlBw). Το BBR εναλλάσσει τον ρυθμό αποστολής σε μικρές αυξομειώσεις, ψάχνοντας να εντοπίσει αλλαγές στο διαθέσιμο εύρος ζώνης. Και στη συνέχεια Το BBR μειώνει για λίγο τον ρυθμό αποστολής του για να διασφαλίσει ότι λαμβάνει την ελάχιστη καθυστέρηση RTT, αποφεύγοντας την υπερβολική συμφόρηση.



Εικόνα 5-10 RTT BBR(bandwidth)

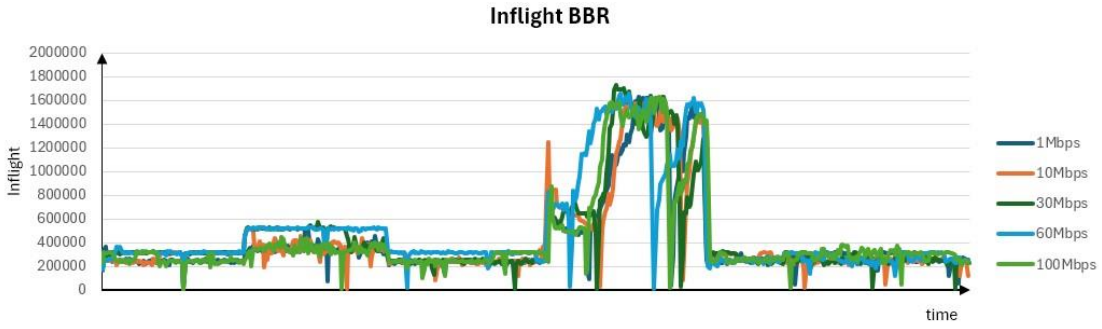
Σε αυτό το σενάριο παρουσιάζουμε το RTT για ροές BBR , μεταβάλλοντας το bandwidth. Το RTT αυξάνεται, καθώς ο αλγόριθμος αυξάνει γρήγορα τον ρυθμό αποστολής, προκαλώντας προσωρινή συμφόρηση. Στη συνέχεια Το RTT μειώνεται καθώς το BBR αποστραγγίζει την ουρά και σταθεροποιείται. Τέλος το BBR μειώνει τον ρυθμό αποστολής για να μετρήσει την ελάχιστη καθυστέρηση.



Εικόνα 5-11 Retransmissions BBR(bandwidth)

Στη συγκεκριμένη περίπτωση το BBR αυξάνει γρήγορα τον ρυθμό αποστολής για να εντοπίσει το μέγιστο διαθέσιμο εύρος ζώνης έτσι η γραφική παράσταση δείχνει αύξηση στις αναμεταδόσεις. Στη συνέχεια το BBR μειώνει τον ρυθμό αποστολής για να αδειάσει την

ουρά. Η μείωση αυτή έχει ως συνέπεια την πτώση των αναμεταδόσεων. Το BBR διατηρεί έναν σταθερό ρυθμό αποστολής, παρακολουθώντας ταυτόχρονα το εύρος ζώνης. Στη γραφική παράσταση, η γραμμή των αναμεταδόσεων παραμένει χαμηλή.

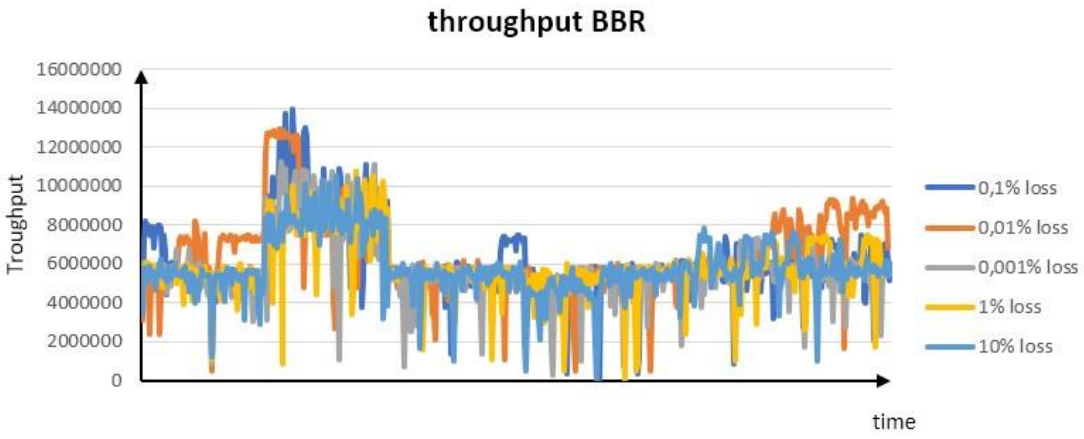


Εικόνα 5-12 Inflight BBR(bandwidth)

Στη γραφική αυτή παρατηρούμε τη συμπεριφορά των δεδομένων σε πτήση μεταβάλλοντας το bandwidth. Αυτό που παρατηρούμε είναι ότι στην αρχή αυξάνετε ο αριθμό των πακέτων σε πτήση καθώς το BBR προσπαθεί να στείλει όσο το δυνατόν περισσότερα δεδομένα για να ανακαλύψει την πραγματική χωρητικότητα του δικτύου. Έπειτα το BBR μειώνει τον αριθμό των πακέτων σε πτήση για να αδειάσει την ουρά που έχει δημιουργηθεί πιο πριν και στη συνέχεια σταθεροποιεί τον αριθμός των πακέτων. Τέλος ο BBR μειώνει τον αριθμό των πακέτων σε πτήση, για να μετρήσει τον ελάχιστο χρόνο καθυστέρησης και να αποφύγει τη συσσώρευση δεδομένων στην ουρά.

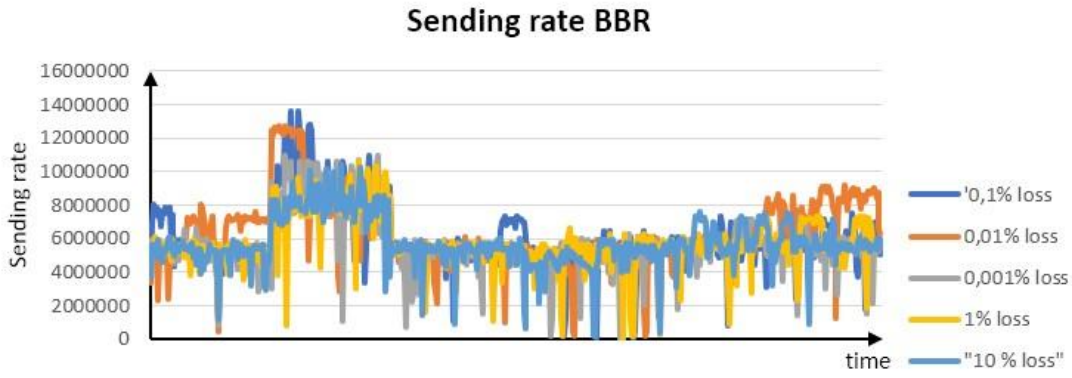
5.2.3 Σενάριο 3ο Σύγκριση παραμέτρων μετατρέποντας τη μεταβλητή loss

Στα επόμενα σενάρια που ακολουθούν χρησιμοποιούμε τα ίδια χαρακτηριστικά με τα παραπάνω με τη διαφορά ότι η μεταβλητή που αλλάζουμε τώρα είναι το loss και κρατάμε όλα τα υπόλοιπα σταθερά



Εικόνα 5-13 Throughput BBR (loss)

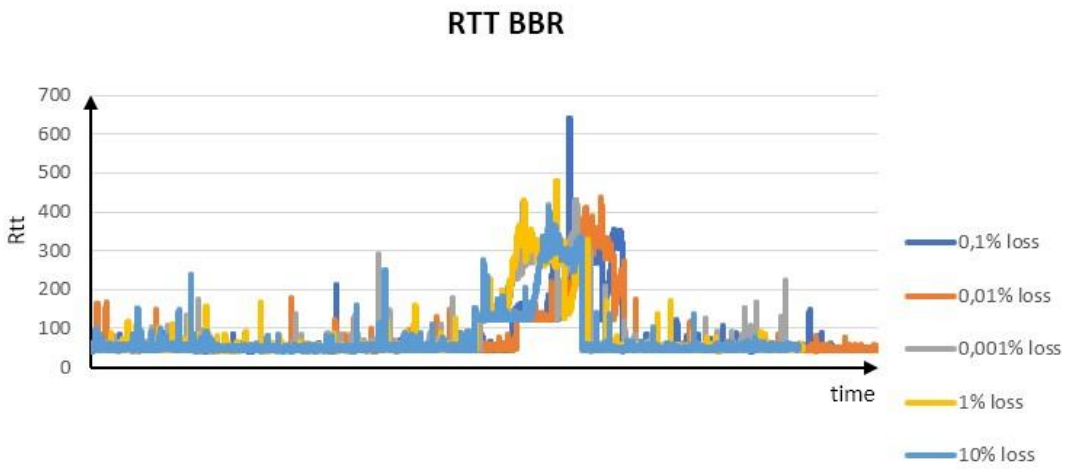
Στο σενάριο αυτό παρουσιάζουμε τη συμπεριφορά του throughput στις μεταβολές της απώλειας. Αυτό που παρατηρούμε είναι ότι το throughput αρχικά αυξάνεται με κάποιες αυξομειώσεις και στη συνέχεια μειώνεται και μένει σταθερό. Επίσης παρατηρούμε ότι όσο μεγαλύτερη είναι η απώλεια η πακέτων τόσο μικρότερη είναι η τιμή του throughput. Το BBR προσπαθεί να προσαρμοστεί στις συνθήκες του δικτύου, προσδιορίζοντας το εύρος ζώνης συμφόρησης για να ελαχιστοποιήσει την απώλεια δεδομένων.



Εικόνα 5-14 Sending Rate BBR(loss)

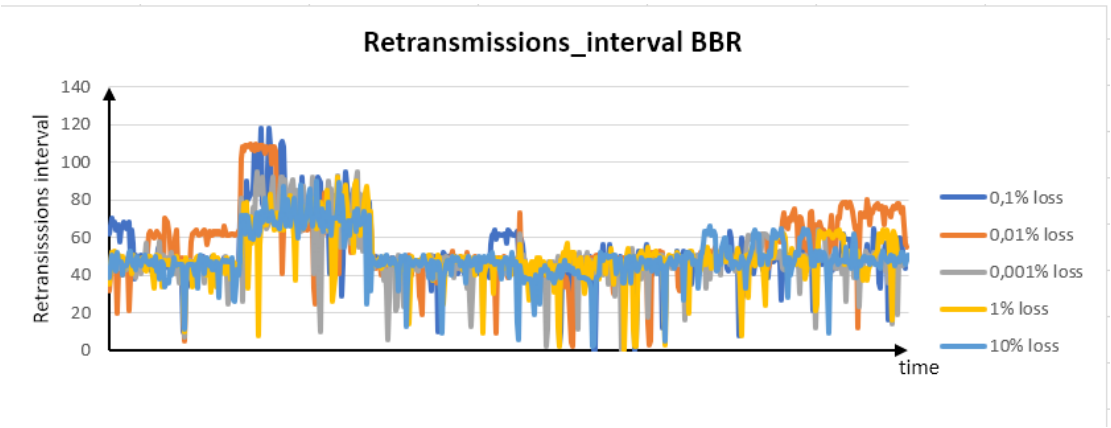
Εδώ παρουσιάζεται η συμπεριφορά του sending rate. Παρατηρούμε ότι το sending rate έχει διάφορες αυξομειώσεις και αυτό οφείλεται στη συμπεριφορά του BBR το οποίο προσπαθεί

να διατηρήσει υψηλό ρυθμό αποστολής, ακόμα και όταν συμβαίνουν απώλειες. Επίσης βλέπουμε ότι ο ρυθμός αποστολής (sending rate) μεταβάλετε ανάλογα με τις απώλειες του συστήματος. Συγκεκριμένα στο γράφημα μας όσο μεγαλύτερη είναι η απώλεια τόσο μικρότερος είναι ο ρυθμός αποστολής.



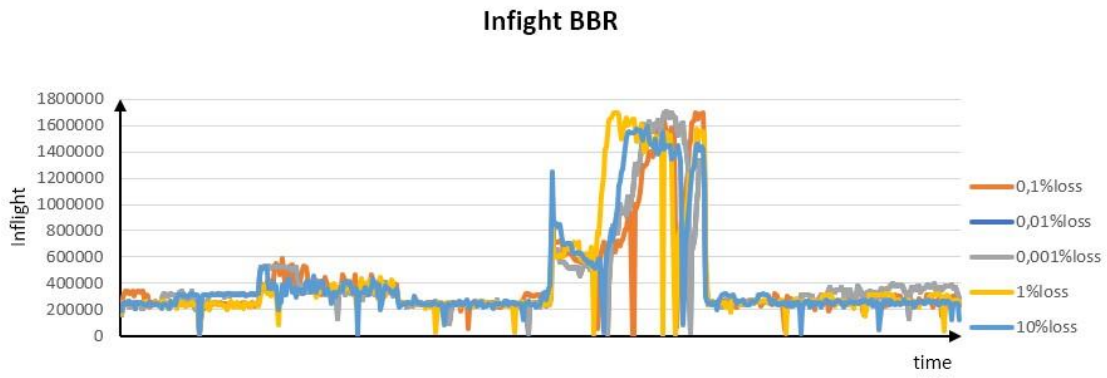
Εικόνα 5-15 RTT BBR(loss)

Η γραφική αυτή μας δείχνει τη συμπεριφορά του RTT όταν συμβαίνουν απώλειες στο σύστημα μας. Αυτό που παρατηρείτε είναι ότι όσο μεγαλύτερη είναι η απώλεια τόσο μικρότερος ο χρόνος RTT. Οι κορυφές στο RTT δείχνουν τις στιγμές που το δίκτυο είναι υπό έντονη συμφόρηση και ο αλγόριθμος προσπαθεί να βρει τη βέλτιστη ροή δεδομένων.



Παρατηρούμε ότι όσο μεγαλύτερες είναι οι απώλειες που συμβαίνουν τόσο λιγότερες αναμεταδώσεις κάνει το BBR.Αυτό συμβαίνει διότι ο αλγόριθμος αυτός έχει ως στόχο να

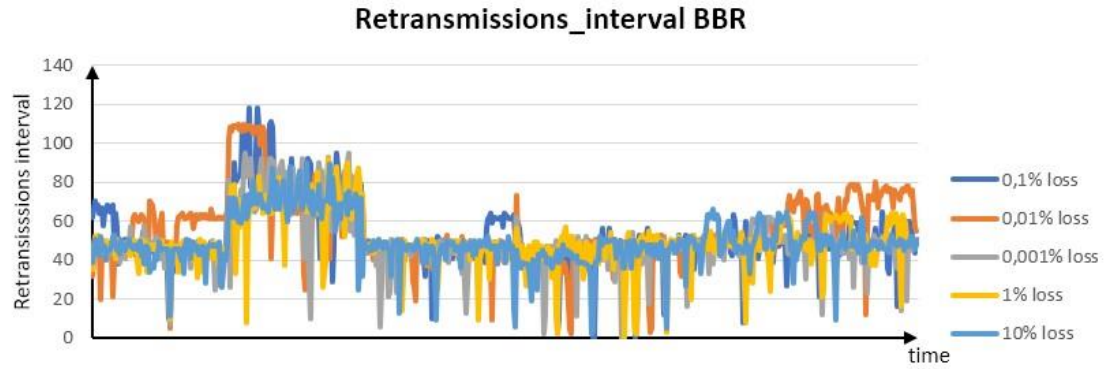
προσαρμόζει τις αναμεταδόσεις ανάλογα με το επίπεδο απώλειας και τις συνθήκες συμφόρησης του δικτύου έτσι το BBR δεν επιλέγει άμεση αναμετάδοση με κάθε απώλεια.



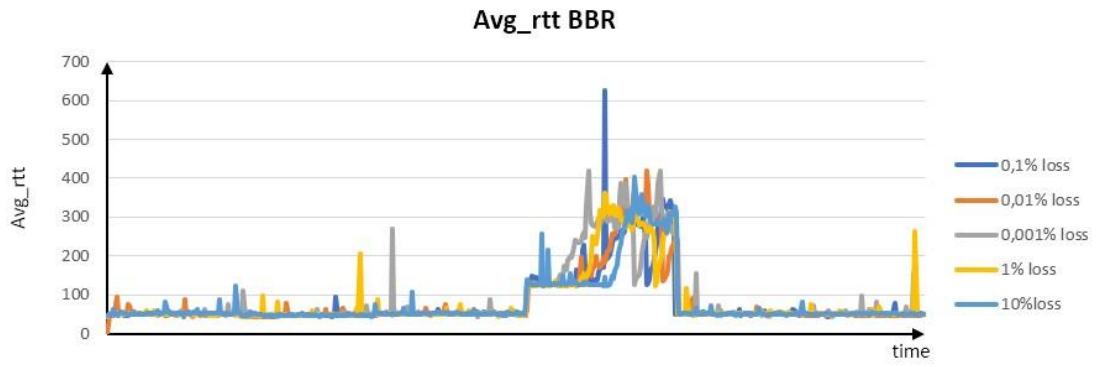
Εικόνα 5-16 Inflight BBR(loss)

Η γραφική παράσταση του inflight άμε δείχνει ότι όταν συναντάμε χαμηλή απώλεια η καμπύλη μπορεί να είναι πιο σταθερή ή να αυξάνεται, καθώς το BBR προσπαθεί να διατηρήσει υψηλό inflight για μέγιστη απόδοση.

Αντίθετα όταν συναντάμε **υψηλή απώλεια**, παρατηρούμε απότομες πτώσεις, καθώς το BBR μειώνει το inflight για να αποφύγει τη συνεχή συμφόρηση.



Εικόνα 5-17 Retransmissions BBR(loss)

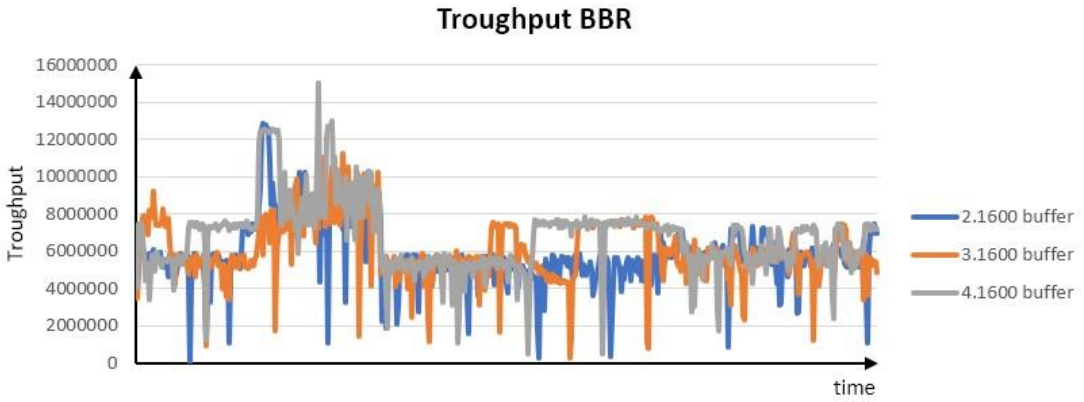


Εικόνα 5-18 Avg_RTT BBR(loss)

Παρατηρούμε ότι σε συνθήκες ήπιας απώλειας, το RTT μπορεί να παραμένει σχετικά σταθερό. Όμως σε περιόδους έντονης συμφόρησης, το RTT αυξάνεται απότομα. Τώρα σχετικά με το σχήμα μας βλέπουμε ότι το RTT αυξάνεται όσο μικρότερες απώλειες συμβαίνουν στο σύστημα μας.

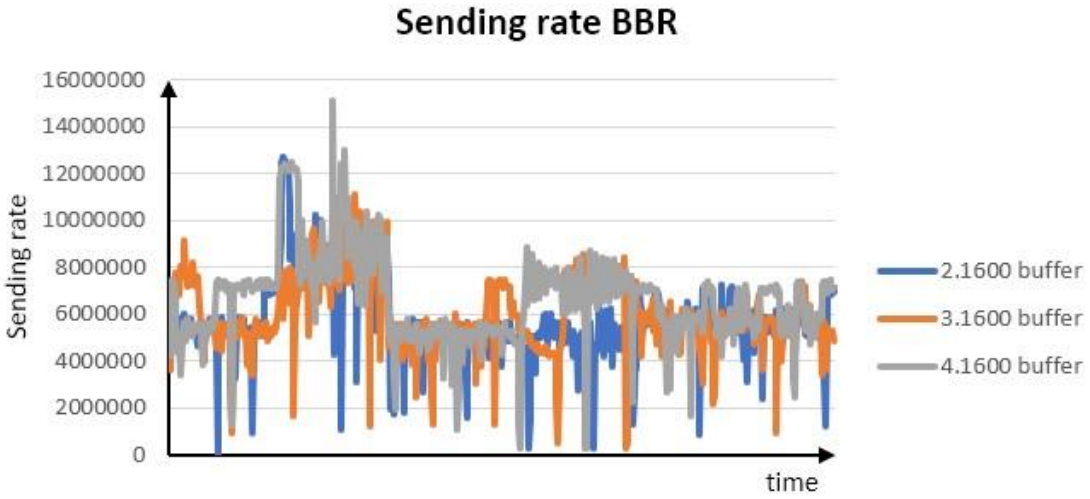
5.2.4 Σενάριο 4 Σύγκριση παραμέτρων μετατρέποντας τη μεταβλητή buffer_size

Στα επόμενα σενάρια που ακολουθούν χρησιμοποιούμε τα ίδια χαρακτηριστικά με τα παραπάνω με τη διαφορά ότι η μεταβλητή που αλλάζουμε τώρα είναι το buffer size και κρατάμε όλα τα υπόλοιπα σταθερά



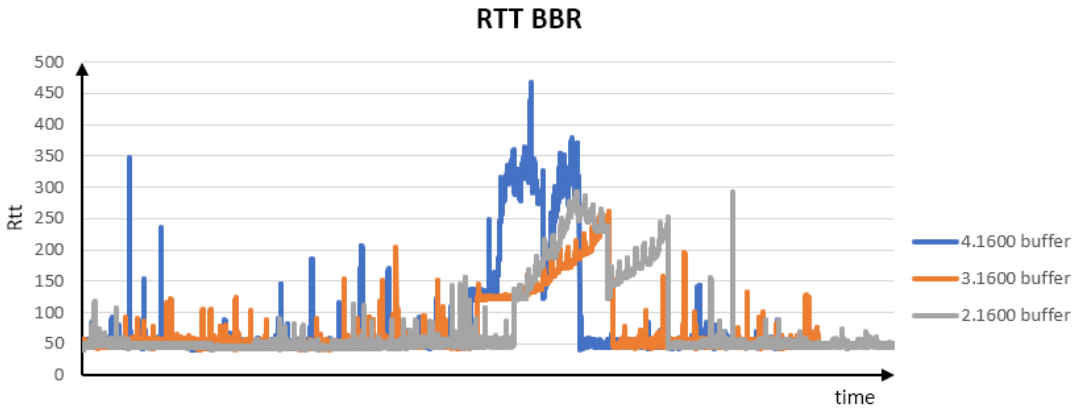
Εικόνα 5-19 Throughput BBR(buffer_size)

Η γραφική απεικόνιση δείχνει ότι, με τη χρήση μικρότερου buffer, το throughput είναι χαμηλότερο λόγω της αυξημένης πιθανότητας συμφόρησης και απώλειας πακέτων. Ενώ αντίθετα με την αύξηση του buffer έχουμε και αύξηση του throughput .



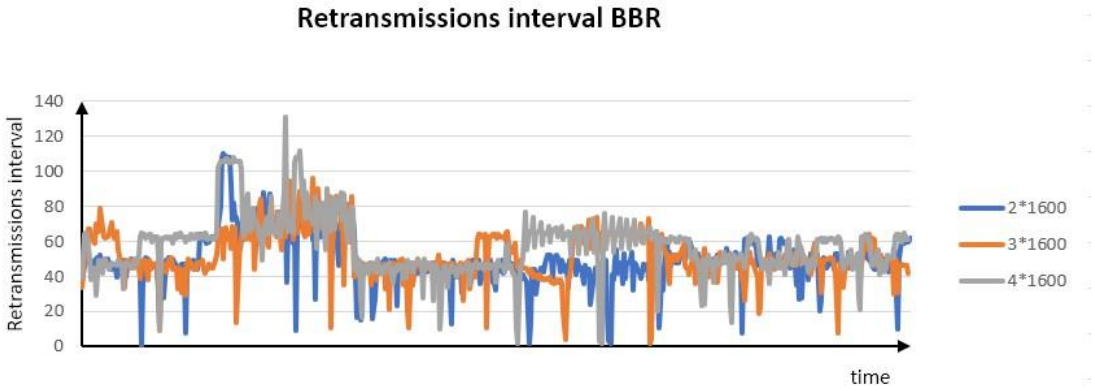
Εικόνα 5-20 Sending rate BBR (buffer_size)

Όταν το μέγεθος του buffer είναι μικρό έχουμε μείωση στον ρυθμό αποστολής ,ενώ όταν το μέγεθος του buffer αυξάνεται, αυξάνεται και ο ρυθμός αποστολής. Καθώς ο BBR έχει τη δυνατότητα να στέλνει περισσότερα δεδομένα με υψηλότερο ρυθμό χωρίς να προκαλεί άμεσα συμφόρηση.



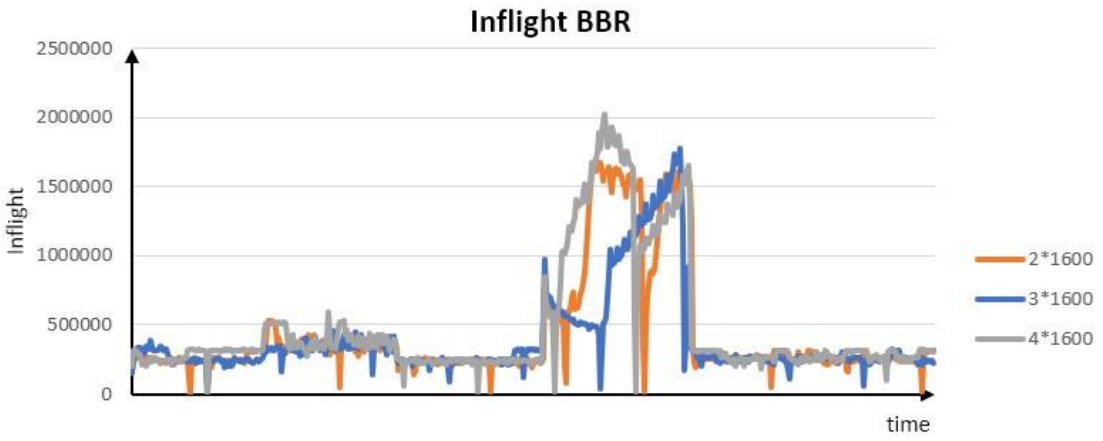
Εικόνα 5-21 RTT BBR(buffer_size)

Η γραφική παράσταση δείχνει χαμηλές τιμές RTT, όταν ο buffer είναι μικρός καθώς ο buffer γεμίζει γρήγορα και η υπερφόρτωση οδηγεί σε απώλειες πακέτων ενώ όταν ο το μέγεθος του buffer αυξάνεται, το RTT αυξάνεται επίσης, διότι τα πακέτα περιμένουν περισσότερο στον buffer πριν αποσταλούν.



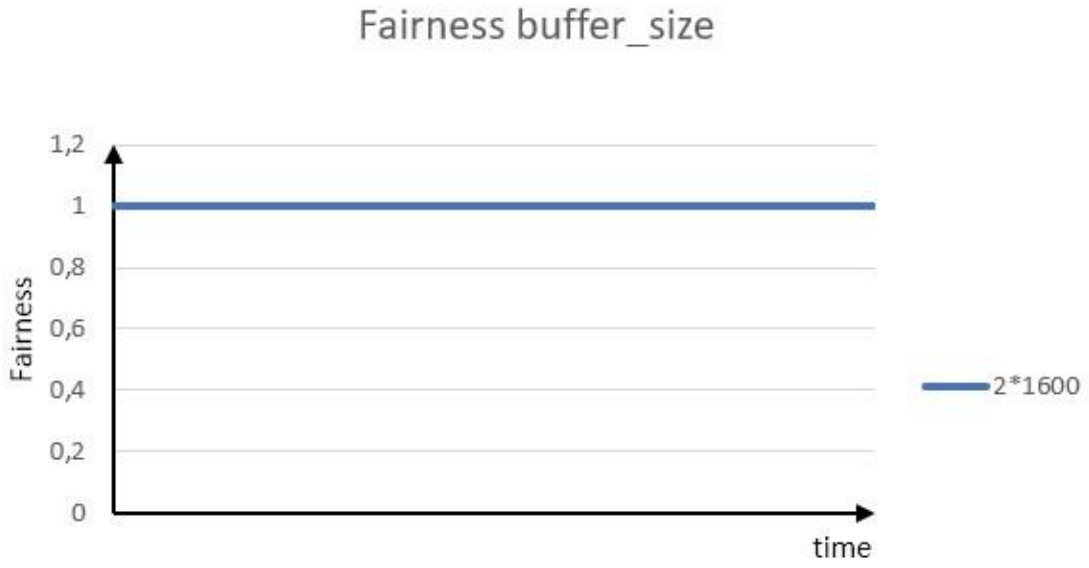
Εικόνα 5-22 Retransmissions BBR(buffer_size)

Στην γραφική αυτή παρατηρούμε ότι καθώς το μέγεθος του buffer αυξάνεται αυξάνονται και οι αναμεταδόσεις. Αυτό είναι λογικό να συμβαίνει καθώς όσο μεγαλύτερος είναι ο buffer τόσα περισσότερα πακέτα μπορούν να μεταδοθούν. Έτσι είναι πιο εύκολο να χχάνονται με αποτέλεσμα και να απαιτείται αναμετάδοση. Από την άλλη όταν το buffer μικραίνει, υπάρχει μικρότερος χώρος για τα δεδομένα, μειώνοντας την πιθανότητα απώλειας πακέτων αρά και τις αναμεταδόσεις



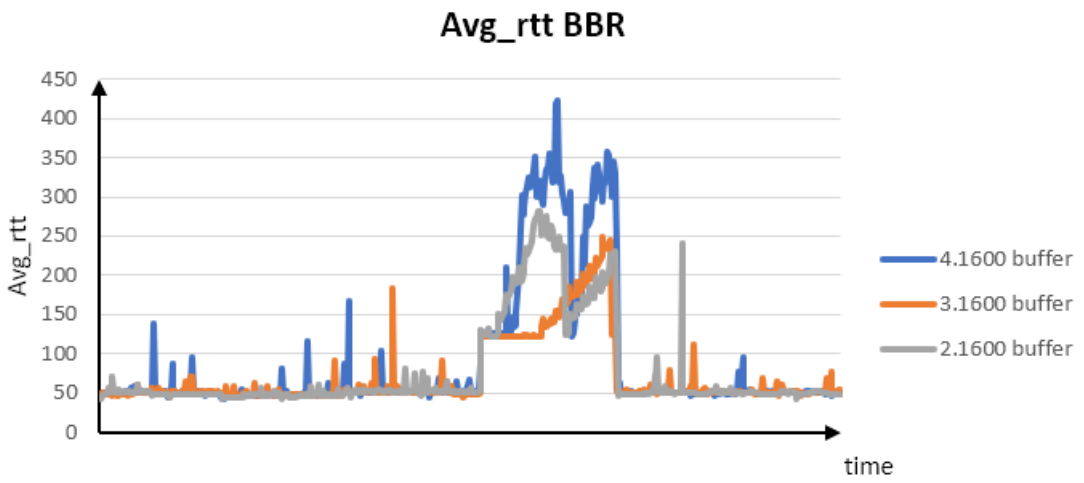
Εικόνα 5-23 Inflight BBR (buffer_size)

Η γραφική παράσταση του inflight μας δείχνει ότι όταν συναντάμε μικρο buffer έχουμε μεγάλο inflight ενώ όταν το μέγεθος του buffer αυξάνεται αυξάνεται και η αποστολή δεδομένων.



Εικόνα 5-24 Fairness BBR (buffer_size)

Αυτό που παρατηρείται είναι ότι η δικαιοσύνη μένει σταθερή στο 1 ανεξάρτητα από το μέγεθος του buffer αυτό σημαίνει ότι οι ροές μοιράζονται το εύρος ζώνης του δικτύου ισότιμα και δίκαια. Η συγκεκριμένη γραφική είναι ίδια για όλα τα σενάρια γι' αυτό και επιλέξαμε να την παρουσιάσουμε μόνο σε μια περίπτωση.



Εικόνα 5-25 Avg_RTT BBR (buffer_size)

Η γραφική παράσταση του avg RTT μας δείχνει ότι όταν έχουμε μικρότερο buffer έχουμε μικρότερο μέσο χρόνο επιστροφής ενώ αντίθετα καθώς το buffer μεγαλώνει, αυξάνεται η δυνατότητα του δικτύου να συγκρατεί πακέτα χωρίς να χρειάζεται να τα απορρίψει. Αυτό έχει ως αποτέλεσμα να αυξάνεται ο μέσος χρόνος επιστροφής(Avg_RTT).

6 Συμπεράσματα

Η συμφόρησή δικτύων είναι ένα πολύ συχνό φαινόμενο το οποίο επηρεάζει την απόδοση του TCP προκαλώντας καθυστερήσεις και απώλειες δεδομένων.

Η παρούσα διπλωματική εργασία έχει ως στόχο να εξετάσει την έννοια της συμφόρησης, τους λόγους για τους οποίους εμφανίζεται αλλά και τις τεχνικές που εφαρμόζονται ώστε να ελεγχθεί και να μειωθεί όσο το δυνατόν περισσότερο. Παράλληλα αναλύουμε το πρωτόκολλο TCP και τους μηχανισμούς ελέγχου συμφόρησης. Συγκρίνουμε διάφορους αλγόριθμους ώστε να εντοπίσουμε αυτούς που λειτουργούν πιο αποτελεσματικά στην αποφυγή και την ελαχιστοποίηση του συγκεκριμένου φαινομένου. Τέλος αξιολογούμε διάφορες μετρικές ώστε να κατανοήσουμε ποιος αλγόριθμος λειτουργεί καλύτερα σε

συνθήκες συμφόρησης . Για την αντιμετώπιση αυτού του φαινομένου κρίνεται απαραίτητή η χρήση διάφορων τεχνικών για την εξασφάλιση μιας σταθερής και αξιόπιστης σύνδεσης. Στα πλαίσια της διπλωματικής εργασίας, εξετάσαμε μέσω προσομοιώσεων τους αλγορίθμους BBR και CUBIC για διάφορες μετρικές ώστε να κατανοήσουμε πλήρως τον τρόπο με τον οποίο συμπεριφέρονται σε συνθήκες συμφόρησης. Συγκεκριμένα εργαστήκαμε σε τέσσερα διαφορετικά σενάρια: α) εξετάσαμε συγκριτικά τη συμπεριφορά των δύο πρωτοκόλλων β) μεταβάλλουμε το εύρος ζώνης γ) μεταβάλουμε το ρυθμό απώλειας πακέτων δ) μεταβάλουμε το μέγεθος του καταχώρηση (buffer size). Στη συνέχεια παρατηρούμε τα αποτελέσματα και τα συγκρίνουμε . Αυτό που παρατηρήσαμε είναι ότι η επιλογή του κατάλληλου αλγορίθμου εξαρτάται από το σενάριο χρήσης και τις συνθήκες που επικρατούν στο δίκτυο μας. Επίσης υπάρχουν ακόμη περιθώρια βελτίωσης (δηλ. μπορούν να γίνουν περαιτέρω ερευνητικές εργασίες για την αντιμετώπιση ορισμένων προκλήσεων, όπως για παράδειγμα η επιλογή του μεγέθους για το buffer). Τέλος μπορεί να γίνει υπερεκτίμηση του BDP και αυτό οδηγεί σε προβλήματα δικαιοσύνης.

Αναφορές

[1] Cardwell, N., Cheng, Y., Gunn, C.S., Yeganeh, S.H. and Jacobson, V., 2017. BBR: congestion-based congestion control. *Communications of the ACM*, *60*(2), pp.58-66.

[2] Martin, J. and Nilsson, A.A., 1997. *The evolution of congestion control in TCP/IP: from reactive windows to preventive rate control*. North Carolina State University. Center for Advanced Computing and Communication.

[3] Albuquerque, C., Vickers, B.J. and Suda, T., 2004. Network border patrol: Preventing congestion collapse and promoting fairness in the internet. IEEE/ACM Transactions on Networking, 12(1), pp.173-186.

[4] Gettys J, Nichols K. Bufferbloat: Dark Buffers in the Internet: Networks without effective AQM may again be vulnerable to congestion collapse. Queue. 2011 Nov 29;9(11):40-54.

[5] Siveraman A. (2017) Lecture 6 , Congection collapse, https://anirudhsk.github.io/teaching/lectures/lec6.pdf

[6] Chongnan, C. and Zhongda, T., 2024, May. Survey on TCP Congestion Control in Ad Hoc Wireless Networks. In *2024 36th Chinese Control and Decision Conference (CCDC)* (pp. 3230-3235). IEEE.

[7] Jacobson, V. (1988). Congestion avoidance and control. *ACM SIGCOMM computer communication review*, *18*(4), 314-329.

[8] Featherly, Kevin. "ARPANET - a Packet of Data." *Encyclopædia Britannica*, 2019, www.britannica.com/topic/ARPANET/A-packet-of-data.

[9] Cerf, V. and Kahn, R., 1974. A protocol for packet network intercommunication. *IEEE Transactions on communications*, *22*(5), pp.637-648.

[10] https://courses.cs.washington.edu/courses/cse461/17au/lectures/04-2-congestionControl.pdf

[11] Gibbens, R. J., & Kelly, F. P. (1999). Resource pricing and the evolution of congestion control. *Automatica*, *35*(12), 1969-1985.

[12] scs.stanford.edu/ :l6.pdf

[13] Peterson, L. L., & Davie, B. S. (2007). *Computer networks: a systems approach*. Morgan Kaufmann.

[14] Zhang, S. (2019). An evaluation of BBR and its variants. *arXiv preprint arXiv:1909.03673*.

[15] Qureshi, B., Othman, M., & Hamid, N. A. W. (2009). Progress in Various TCP Variants: Issues, Enhancements and Solutions. *Mausaum Journal of Computing*, *1*(14), 493-499.

[16] Rejaie, R., Handley, M., & Estrin, D. (1999, March). RAP: An end-to-end rate-based congestion control mechanism for realtime streams in the Internet. In *IEEE INFOCOM'99. Conference on Computer Communications. Proceedings. Eighteenth*

Annual Joint Conference of the IEEE Computer and Communications Societies. The Future is Now (Cat. No. 99CH36320) (Vol. 3, pp. 1337-1345). IEEE.

[17] Welzl, M. (2005). *Network congestion control: managing internet traffic*. John Wiley & Sons.

[18] Saedi, T., & El-Ocla, H. (2021). TCP CERL+: Revisiting TCP congestion control in wireless networks with random loss. *Wireless Networks*, *27*, 423-440.

[19] Lorincz, J., Klarin, Z., & Ožegović, J. (2021). A comprehensive overview of TCP congestion control in 5G networks: Research challenges and future perspectives. *Sensors*, *21*(13), 4510.

[20] Scholz, D., Jaeger, B., Schwaighofer, L., Raumer, D., Geyer, F., & Carle, G. (2018, May). Towards a deeper understanding of TCP BBR congestion control. In *2018 IFIP networking conference (IFIP networking) and workshops* (pp. 1-9). IEEE.

[21] https://datatracker.ietf.org/doc/html/rfc2018

[22] Ha, S., Rhee, I., & Xu, L. (2008). CUBIC: a new TCP-friendly high-speed TCP variant. *ACM SIGOPS operating systems review*, *42*(5), 64-74.

[23] https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4594375[16]

[24] Philip, A. A., Ware, R., Athapathu, R., Sherry, J., & Sekar, V. (2021, November). Revisiting TCP congestion control throughput models & fairness properties at scale. In *Proceedings of the 21st ACM Internet Measurement Conference* (pp. 96-103).

[25] Scholz, D., Jaeger, B., Schwaighofer, L., Raumer, D., Geyer, F., & Carle, G. (2018, May). Towards a deeper understanding of TCP BBR congestion control. In *2018 IFIP networking conference (IFIP networking) and workshops* (pp. 1-9). IEEE.

[26] https://courses.cs.washington.edu/courses/cse461/20sp/slides/3-transport-apr8.pdf (σελίδα 26)

[27] Rejaie, R., Handley, M., & Estrin, D. (1999, March). RAP: An end-to-end rate-based congestion control mechanism for realtime streams in the Internet. In *IEEE INFOCOM'99. Conference on Computer Communications. Proceedings. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. The Future is Now (Cat. No. 99CH36320)* (Vol. 3, pp. 1337-1345). IEEE.

[28] Bauer, S., Clark, D.D. and Lehr, W., 2009, August. The evolution of internet congestion. TPRC.

[29] Prvan, M. and Ožegović, J., 2020. Methods in teaching computer networks: a literature review. *ACM Transactions on Computing Education (TOCE)*, *20*(3), pp.1-35.

[30] Martinez, C. H., Kim, V., Chen, Y., Kazerooni, E. A., Murray, S., Criner, G. J., ... & COPDGene Investigators. (2014). The clinical impact of non-obstructive chronic bronchitis in current and former smokers. *Respiratory medicine*, *108*(3), 491-499..