



ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΘΕΜΑ:

**ΑΞΙΟΛΟΓΗΣΗ ΑΠΟΔΟΣΗΣ ΤΩΝ TCP VARIANTS ΣΕ ΕΝΣΥΡΜΑΤΑ
ΔΙΚΤΥΑ ΜΕ ΤΗΝ ΧΡΗΣΗ OMNeT++**



Θωμάς Κιτσάκης

Επιβλέπων: Μαργαρίτη Σπυριδούλα



ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΘΕΜΑ:

**ΑΞΙΟΛΟΓΗΣΗ ΑΠΟΔΟΣΗΣ ΤΩΝ TCP VARIANTS ΣΕ ΕΝΣΥΡΜΑΤΑ
ΔΙΚΤΥΑ ΜΕ ΤΗΝ ΧΡΗΣΗ OMNeT++**



Θωμάς Κιτσάκης

Επιβλέπων: Μαργαρίτη Σπυριδούλα

**PERFORMANCE VALIDATION OF TCP VARIANTS IN WIRED
NETWORKS USING OMNeT++**

Εγκρίθηκε από τριμελή εξεταστική επιτροπή

Τόπος, Ημερομηνία

ΕΠΙΤΡΟΠΗ ΑΞΙΟΛΟΓΗΣΗΣ

1. Επιβλέπων καθηγητής Όνομα Επίθετο,

τίτλος, βαθμίδα

2. Μέλος επιτροπής Όνομα Επίθετο,

τίτλος, βαθμίδα

3. Μέλος επιτροπής Όνομα Επίθετο,

τίτλος, βαθμίδα

© ΘΩΜΑΣ ΚΙΤΣΑΚΗΣ, 2021.

Με επιφύλαξη παντός δικαιώματος. All rights reserved

Δήλωση μη λογοκλοπής

Δηλώνω υπεύθυνα και γνωρίζοντας τις κυρώσεις του Ν. 2121/1993 περί Πνευματικής Ιδιοκτησίας, ότι η παρούσα πτυχιακή εργασία είναι εξ 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.

Θωμάς Κιτσάκης

Άρτα 2021

Υπογραφή

ΠΕΡΙΛΗΨΗ

Από τα μέσα της δεκαετίας του '90 και ειδικότερα από τις αρχές του 2000, η επικοινωνία μεταξύ των ανθρώπων που είναι βασισμένη στα δίκτυα των υπολογιστών έχει γνωρίσει μια πολύ μεγάλη αλματώδη ανάπτυξη. Η πλειοψηφία πλέον σε πολλούς κλάδους όπως στην ιατρική, γενικές επιστήμες, βιομηχανία, κ.α. είναι βασισμένη στα δίκτυα υπολογιστών. Κυρίο ρόλο σε όλα αυτά παίζει η υποδομή των δικτύων που είναι τα ενσύρματα δίκτυα αλλά και τα ασύρματα τα τελευταία χρόνια έτσι ώστε να είναι εφικτό όλο αυτή η δυνατότητα να επικοινωνούμε. Όπου είναι ενσύρματη δικτύωση είναι βασισμένη στα ομοαξονικά καλώδια που χρησιμοποιούνται κυρίως για το τηλεοπτικό σήμα στις τηλεοράσεις, τα χάλκινα καλώδια που χρησιμοποιούνται ευρέως έως και σήμερα, και τα τελευταία χρονιά μεγάλα ανάπτυξη έχει η οπτική είναι χάρη στις πολύ μεγάλες ταχύτητες των δεδομένων που υποστηρίζει. Η οπτική ίνα τα επόμενα χρονιά θα έχει καταλυτικό ρολό στις επικοινωνίες με τις δυνατότητες που υποστηρίζει σαν υλικό υποδομής, αξιοπιστίας κ.α.

Στην παρούσα πτυχιακή γίνεται μία παρουσίαση του πρωτοκόλλου TCP, των Αλγορίθμων ελέγχου συμφόρησης, και η αξιολόγηση απόδοσης των TCP Variants πάνω στο μοντέλο του προσομοιωτή το OMNeT++ με χρήση των ενσύρματων δικτύων. Οι στόχοι του πρωτοκόλλου είναι η σωστή λήψη και παραλαβή των μηνυμάτων που στέλνονται μεταξύ των 2 ακρών όπως η καθυστέρηση, ο ρυθμός μεταφοράς των πακέτων ή διακύμανση του παραθύρου συμφόρησης, ενός δικτύου βασισμένο στα ενσύρματα δίκτυα.

Στο τέλος της μελέτης, θα έχουμε μία κάπως διαφορετική αντίληψη όταν για παράδειγμα με ένα “κλικ” θα αποφασίζουμε να έχουμε πρόσβαση σε κάποια ιστοσελίδα ή θα ζητάμε να αποθηκεύσουμε κάποιο αρχείο από έναν εξυπηρετητή στην ηλεκτρονική συσκευή μας, θα αναλογισθούμε πόσο πολύπλοκοι μηχανισμοί αναλαμβάνουν δράση στο TCP, για την σωστή και γρήγορη εξυπηρέτηση μας.

Λέξεις – κλειδιά: TCP , Έλεγχος συμφόρησης, Variants , Πρωτόκολλα, OMNeT++

Abstract

Since the mid-1990s and especially since the early 2000s, communication between people using computer networks has experienced a very rapid growth. The vast of many fields such as medicine, general sciences, industry, etc. is based on computer networks. The main reason for this growth is due to the infrastructure of the networks, which are wired networks but also in the last years wireless ones, so that all this communication is possible. Wired networking is based on coaxial cables used primarily for the TV signal on televisions, copper cables which are widely used to this day, and in recent years optical fiber growth has grown thanks to the very high data speeds it supports. The optical fiber in the coming years will have the main role in communications with the capabilities it supports as infrastructure material, reliability etc.

In this dissertation there is a presentation of the TCP protocol, the Congestion Control Algorithms and the evaluation of the performance of the TCP Variants on the OMNeT ++ simulator model using wired networks. The goals of the protocol are the correct reception and receiving of messages sent between the 2 ends such as the delay, the packet transfer rate, the congestion window fluctuation, of a network based on wired networks.

At the end of the study, we will have a slightly different perception of what is happening in the background when, for example, with a "click" we decide to access a website or ask to save a file from a server on our electronic device. We will then consider how complex mechanisms work in TCP, for our reliable and fast service.

Keywords: TCP, Congestion Control, Variants, Protocols, OMNeT ++

ΠΕΡΙΕΧΟΜΕΝΑ

1.ΕΙΣΑΓΩΓΗ	- 5 -
1.2 Το Διαδίκτυο.....	- 5 -
1.3 Δίκτυα Υπολογιστών	- 6 -
1.3.1 Δομικά Στοιχεία.....	- 7 -
1.4 Μοντέλο Επικοινωνίας Δεδομένων	- 8 -
1.4.1 Μοντέλο Shannon και Weaver	- 9 -
2. ΜΟΝΤΕΛΑ ΑΝΑΦΟΡΑΣ	- 11 -
2.1 Μοντέλο αναφοράς OSI.....	- 11 -
2.1.1 Στρωματοποίηση κατά OSI.....	- 12 -
2.2 Το Μοντέλο TCP/IP.....	- 13 -
2.2.1 Τα επίπεδα του TCP/IP	- 16 -
2.3 Πρωτόκολλα Μεταφοράς	- 20 -
2.3.1 UDP: User Datagram Protocol	- 20 -
2.3.2 SCTP - Streaming Control Transmission Protocol.....	- 20 -
2.3.3 DCCP Datagram Congestion Control Protocol	- 21 -
2.3.4 TCP Transmission Control Protocol	- 20 -
3. TCP: ΠΡΩΤΟΚΟΛΛΟ ΑΞΙΟΠΙΣΤΗΣ ΜΕΤΑΦΟΡΑΣ.....	- 22 -
3.1 Χαρακτηριστικά TCP.....	- 22 -
3.2 Αρχιτεκτονική TCP	- 23 -
3.3 Επικεφαλίδα TCP.....	- 25 -
3.4 Θύρες (Ports)	- 26 -
3.5 Συνδέσεις TCP.....	- 27 -
3.6 Λειτουργία του TCP	- 28 -
3.7 Μηχανισμός Αξιόπιστης Μεταφοράς Δεδομένων TCP	- 31 -
3.8 Μηχανισμός επιβεβαίωσης κυλιόμενου παραθύρου του TCP.....	- 32 -
3.9 Μηχανισμοί Ελέγχου Κίνησης του TCP	- 34 -
3.10 Χρονιστές Ελέγχου Κίνησης του TCP.....	- 35 -
4. TCP ΑΛΓΟΡΙΘΜΟΙ ΕΛΕΓΧΟΥ ΣΥΜΦΟΡΗΣΗΣ.....	- 37 -
4.1 Έλεγχος συμφόρησης TCP με αλγόριθμο σταδιακής αύξησης και πολλαπλής μείωσης (AIMD Additive increase/multiplicative decrease).....	- 38 -
5 TCP VARIANTS.....	- 45 -
5.1 Εισαγωγή	- 45 -

5.3 Ο Αλγόριθμος TCP – New Reno	- 48 -
5.4 Ο Αλγόριθμος TCP Vegas	- 50 -
5.5 Ο Αλγόριθμος TCP Sack	- 51 -
6. ΕΙΣΑΓΩΓΗ ΣΤΗΝ ΠΡΟΣΟΜΟΙΩΣΗ ΤΩΝ ΔΙΚΤΥΩΝ	- 53 -
6.1 Τα Πλεονεκτήματα Και Μειονεκτήματα της Προσομοίωσης	- 53 -
6.1.1 Πλεονεκτήματα.....	- 53 -
6.1.2 Μειονεκτήματα.....	- 54 -
6.2 Το OMNeT++.....	- 55 -
7. ΠΡΟΣΟΜΟΙΩΣΕΙΣ ΤΩΝ TCP VARIANTS ΣΤΟ OMNeT++.....	- 59 -
7.1 Εισαγωγή	- 59 -
7.1.1 Μετρικές απόδοσης.....	- 60 -
7.2 Simulation TCP Variants	- 60 -
7.2.2 Περίπτωση Μελέτης Throughput	- 65 -
7.2.3 Περίπτωση Μελέτης Cwnd (Παράθυρο Συμφόρησης).....	- 68 -
7.2.4 Περίπτωση Μελέτης RTT (Round Trip Time)	- 74 -
7.2.6 Αριθμός Πακέτων.....	- 77 -
7.2.7 Ποσοστά χρήσεως γραμμής	- 79 -
9. ΣΥΜΠΕΡΑΣΜΑΤΑ ΜΕΛΕΤΗΣ	- 82 -
ΠΑΡΑΡΤΗΜΑ Α: Κώδικας Client - Server	- 85 -

1.ΕΙΣΑΓΩΓΗ

1.2 Το Διαδίκτυο

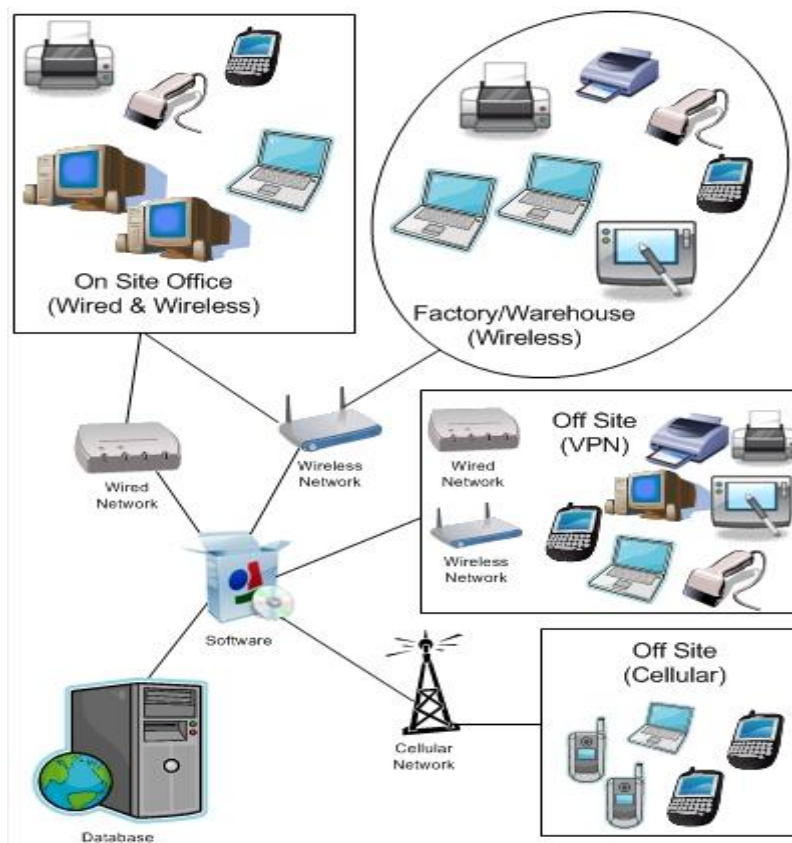
Το Διαδίκτυο ή στην αγγλική ορολογία το γνωστό μας «internet» είναι ένα παγκόσμιο σύστημα διασυνδεδεμένων δικτυακών συσκευών, όπως υπολογιστών, κινητών τηλεφώνων, tablets, servers, κ.ά. Το 1969 το Υπουργείο Αμύνης των Η.Π.Α ανέπτυξε για τις επιχειρησιακές του ανάγκες το δίκτυο APRANET. στην ουσία είναι το πρώτο δίκτυο παγκοσμίως μεταγωγής πακέτων. Το τότε δίκτυο ήταν πολύ μικρό για τα σημερινά δεδομένα κάλυπτε μόνο 4 περιοχές των Αμερικάνικων Πολιτειών. Με τον καιρό όμως το δίκτυο και οι ανάγκες άρχισαν να μεγαλώνουν, περίπου το 1974 παρουσιάστηκε το πρωτόκολλο μεταφοράς (Transmission Control Protocol -TCP), και τα αμέσως επόμενα χρόνια βελτιώθηκε από την κοινότητα APRANET, αλλά και από διάφορους δημόσιους οργανισμούς όπως για παράδειγμα από το Γαλλικό δίκτυο Cyclades. Και έτσι στην συνέχεια, καθιερώθηκε το πρωτόκολλο μεταφοράς (TCP - Transmission Control Protocol), όπως επίσης και το γνωστό IP(Internet Protocol). Όπου μέχρι και σήμερα είναι η βάση του διαδικτύου, το οποίο αποκαλείται μοντέλο αναφοράς TCP/IP . Το TCP/IP στηρίζεται για την διασύνδεση και την επικοινωνία των συσκευών σε διάφορα πρωτόκολλα επικοινωνίας και επίπεδα υλικού και λογισμικού τα οποία όλα μεταξύ θεωρούνται σαν «αλυσίδα» για την εύρυθμη επικοινωνία του Διαδικτύου, όπου και θα το μελετήσουμε σε άλλο κεφάλαιο παρακάτω.[1.]



Εικόνα 1.1: Networks and Devices.

1.3 Δίκτυα Υπολογιστών

Όπως αναφέραμε και παραπάνω στην εισαγωγή, τα πρώτα δίκτυα ηλεκτρονικών υπολογιστών παρουσιάστηκαν στα τέλη της δεκαετίας του '60 αρχές της δεκαετίας του '70 και τις τελευταίες δεκαετίες, ιδιαίτερα από το '90 και μετά, γνωρίζουν μεγάλη ανάπτυξη, καθώς η τεχνολογία μετάδοσης ψηφιακών δεδομένων εξελίσσεται με εξαιρετικά γοργούς ρυθμούς και οι τηλεπικοινωνιακές εφαρμογές συνεχώς αυξάνουν. Σήμερα, έχουν δημιουργηθεί κατάλληλες δικτυακές υποδομές, οι οποίες είναι σε θέση να φιλοξενήσουν εξελιγμένα δίκτυα υπολογιστών και καινοτόμες δικτυακές υπηρεσίες. Πλέον σχεδόν όλες οι εταιρίες μικρές-μεγάλες στην πλειοψηφία τους έχουν τους Η/Υ τους συνδεδεμένους σε δίκτυα. Όπως διάφορα ιδρύματα, οργανισμοί, βιομηχανίες κ.ά. Ειδικά στον τομέα της ιατρικής, της εκπαίδευσης, τα δίκτυα έχουν βελτιώσει κατά πολύ τις προσφερόμενες υπηρεσίες τους.. Ποιο συγκεκριμένα τα δίκτυα Η/Υ απαρτίζονται από διάφορες συσκευές γενικού σκοπού όπως routers, switch, εκτυπωτές κ.ά. που έχουν τη δυνατότητα να επεξεργάζονται πληροφορίες που λαμβάνουν με διάφορους τρόπους και να τις αποδίδουν με μορφές του τύπου όπως φωνή, ήχος εικόνα, μέσω διαδικτυακών εφαρμογών που αυξάνουν την αποτελεσματικότητα σε πολλές καθημερινές ανάγκες. Σε γενικές γραμμές ένα δίκτυο Η/Υ είναι ένα σύνολο διασυνδεδεμένων υπολογιστών όπως για παράδειγμα στην εικόνα 1.2, που μπορούν να μέσω διαφόρων πρωτοκόλλων (που θα αναλύσουμε σε άλλο κεφάλαιο στην συνέχεια), λογισμικού (Software) ,υλικού (Hardware) να επικοινωνήσουν μεταξύ τους και να ανταλλάξουν πληροφορίες.. [1.]



Εικόνα 1.2: Δίκτυο Η/Υ και διαφόρων μέσων υλικού

1.3.1 Δομικά Στοιχεία

Η υλοποίηση ενός δικτύου στηρίζεται σε διάφορες τεχνολογικές επιλογές και λαμβάνονται υπόψη διάφοροι παράγοντες, όπως για παράδειγμα η έκταση που θα καλύπτει το δίκτυο, το είδος και τα τεχνικά χαρακτηριστικά των δικτυακών συσκευών επικοινωνίας, τις δικτυακές υποδομές, τα υπολογιστικά συστήματα, το βαθμός ασφάλειας του δικτύου.

Συνοπτικά τα δομικά στοιχεία που απαρτίζουν ένα δίκτυο υπολογιστών είναι:

1. Τα μέσα μετάδοσης (Ενσύρματα & Ασύρματα):

Όπως για παράδειγμα αν μιλάμε για ενσύρματη σύνδεση του δικτύου, τότε είναι τα καλώδια συνεστραμμένου ζεύγους (UTP), οπτική ίνα, ομοαξονικό καλώδιο, ή συνδυασμός δύο ή και παραπάνω τύπων καλωδίων.

2. Τα υπολογιστικά συστήματα:

Συνήθως όταν αναφερόμαστε σε κάποιο υπολογιστικό σύστημα, εννοούμε μια πλήρης υπολογιστική συσκευή του τύπου ένας Ηλεκτρονικός Υπολογιστής

3. Διακομιστές (Server):

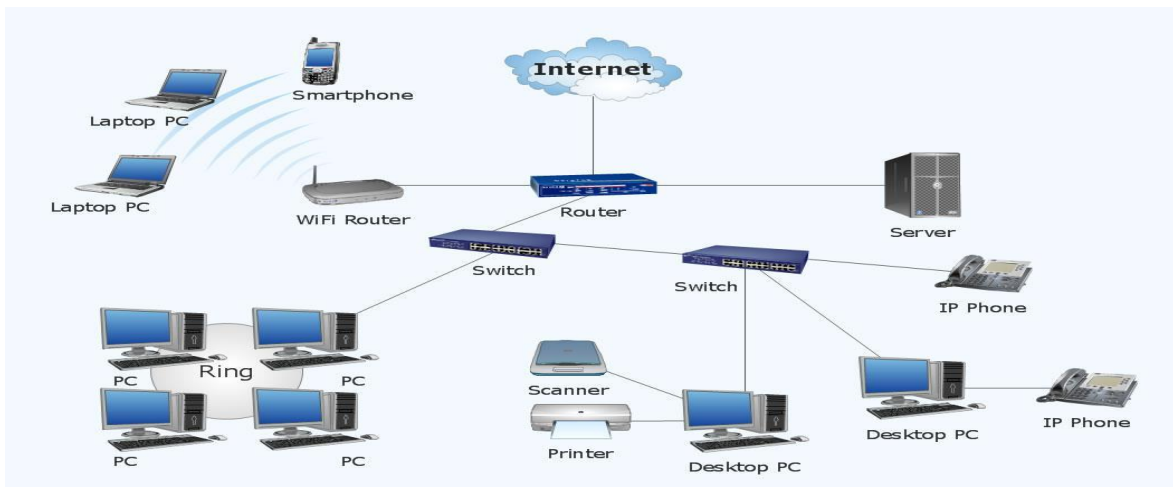
Θεωρούνται οι Η/Υ, οι οποίοι σε σχέση με ένα απλό Η/Υ, έχουν παραπάνω δυνατότητες σε θέμα υλικού, και είναι υπεύθυνοι για την ομαλή και απρόσκοπτη λειτουργία ενός μεγάλου δικτύου.

4. Λογισμικό (Software):

Αναφέρεται στον τύπο του λειτουργικού συστήματος όπου θα είναι εγκατεστημένο σε κάποιο server ή έναν απλό Η/Υ, όπως και στα διάφορα προγράμματα-εφαρμογές που είναι εγκατεστημένα πάνω στον λειτουργικό σύστημα για την υποστήριξη του δικτύου και όχι μόνο.

5. Οι διάφορες συσκευές επικοινωνίας (Router, Switch, κ.τ.λ.): (Εικόνα 1.3).

Θεωρούνται οι συσκευές οι οποίες είναι υπεύθυνες για την προώθηση των δεδομένων του δικτύου αλλά και τον διαμοιρασμό σε άλλες συσκευές επικοινωνίας..[1.], [2.]



Εικόνα 1.3: Διάφορα Δομικά στοιχεία και συσκευές επικοινωνίας ενός Δικτύου

1.4 Μοντέλο Επικοινωνίας Δεδομένων

Όταν το 1854 ο Samuel Morse εφηύρε τον τηλεγράφο και περίπου 20 χρόνια αργότερα με την εφεύρεση του τηλεφώνου από τον Graham Bell, από τότε ξεκίνησε μια νέα εποχή στον χώρο των επικοινωνιών. Πλέον στην εποχή μας η ψηφιακή επικοινωνία είναι κυριαρχεί στην καθημερινότητα μας.. Στόχος είναι μεταξύ δύο η και περισσότερων σταθμών επικοινωνίας είναι η σωστή αποστολή και λήψη των δεδομένων, πακέτων η μηνυμάτων, από τον δέκτη προς τον παραλήπτη

Στις αρχές του 1970, ολοένα και αυξάνεται η χρήση των Η/Υ κυρίως σε επαγγελματικό επίπεδο και διάφορους άλλους οργανισμούς, όπου αρχικά η μετάδοση των πληροφοριών και δεδομένων γινόταν από το υπάρχον τηλεφωνικό δίκτυο

Στην επόμενη δεκαετία του 80, ξεκίνησαν την λειτουργία τα πρώτα δημοσιά δίκτυα μετάδοσης δεδομένων, όπου μπορούσαν να τα χρησιμοποιήσουν και ιδιώτες παράλληλα, εκτός από επιχειρήσεις και δημοσίους οργανισμούς

Στην Δεκαετία του 90 τότε που ξεκίνησε να επεκτείνεται ο τομέας της πληροφορικής οδήγησε τους τότε ειδικούς σε νέες μορφές επικοινωνιών. Όπου ο στόχος τότε ήταν όλα τα δίκτυα να ενοποιηθούν σε ένα ενιαίο δίκτυο. Και συγκεκριμένα τα Δίκτυα ISDN άρχισαν να κερδίζουν έδαφος. Γιατί έχουν τη δυνατότητα να παρέχουν στους χρήστες την δυνατότητα της ενοποιημένης και ομοιόμορφης διαχείρισης διαφόρων είδους δεδομένων όπως π.χ. φωνή, εικόνα, κείμενο σε μια μόνο σύνδεση. [1].[2].

1.4.1 Μοντέλο Shannon και Weaver

Κατά την διάρκεια του Β΄ Παγκοσμίου πολέμου δημιουργήθηκε και αναπτύχθηκε από τους Shannon και Weaver στα εργαστήρια της Bell Telephone με έδρα την Βοστώνη της Μασαχουσέτης, το μοντέλο γραμμικής επικοινωνίας για την μεταφορά μηνυμάτων. στο οποίο μετέχουν πέντε λειτουργικές μονάδες (, πομπός, πηγή, κανάλι, δέκτη, προορισμός).

Πηγή: Αρχή διαδρομής της πληροφορίας.

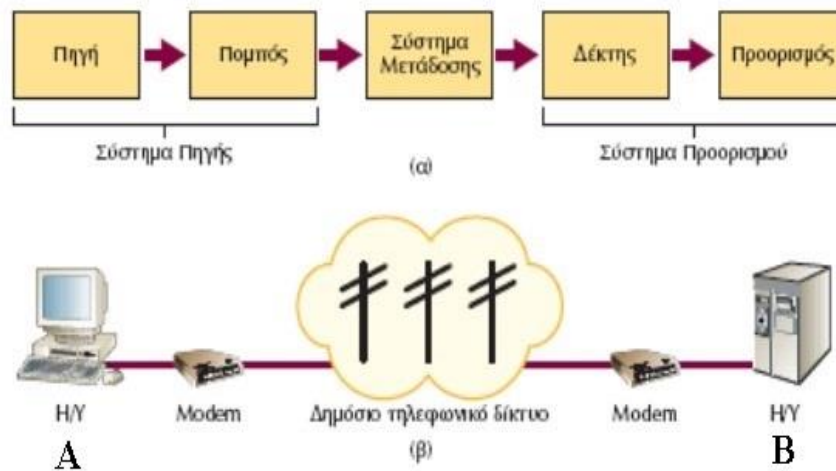
Πομπός: Συσκευή η οποία έχει την δυνατότητα να τροποποιεί τα σήματα δηλαδή να κωδικοποιεί με τέτοιο τρόπο την πληροφορία ,όπου παράγονται ηλεκτρομαγνητικά σήματα και μεταδίδονται στην συνέχεια με κάποιο μέσο μετάδοσης.

Κανάλι: Είναι τα μέσα από την από την οποία μεταδίδεται η κωδικοποιημένη πληροφορία, από τον πομπό όπως π.χ. οι τηλεφωνικές γραμμές, τα ηλεκτρομαγνητικά κύματα του ραδιοφώνου και της τηλεόρασης, οι οπτικές ίνες κτλ.

Δέκτης: Συσκευή η οποία στην συνέχεια του καναλιού δέχεται το σήμα όπου παραμετροποιείται καταλληλά σε τέτοια μορφή έτσι ώστε από την συσκευή προορισμού να είναι κατανοητή η πληροφορία.

Προορισμός: Τελικός προορισμός της πληροφορίας.

Το επικοινωνιακό μοντέλο: Χρησιμοποιείται για την περιγραφή και την ανάλυση των διαφόρων σταδίων κατά τη μετάδοση της πληροφορίας από την πηγή και τον δέκτη. Ένα μοντέλο επικοινωνίας δεδομένων περιγράφει τον εξοπλισμό, τις διεπαφές (interfaces) και τα μέσα μετάδοσης τα οποία ορίζουν το μονοπάτι από την πηγή στο δέκτη. [2.]



Εικόνα 1.4: Η Επικοινωνία δύο σταθμών Ηλεκτρονικών Υπολογιστών

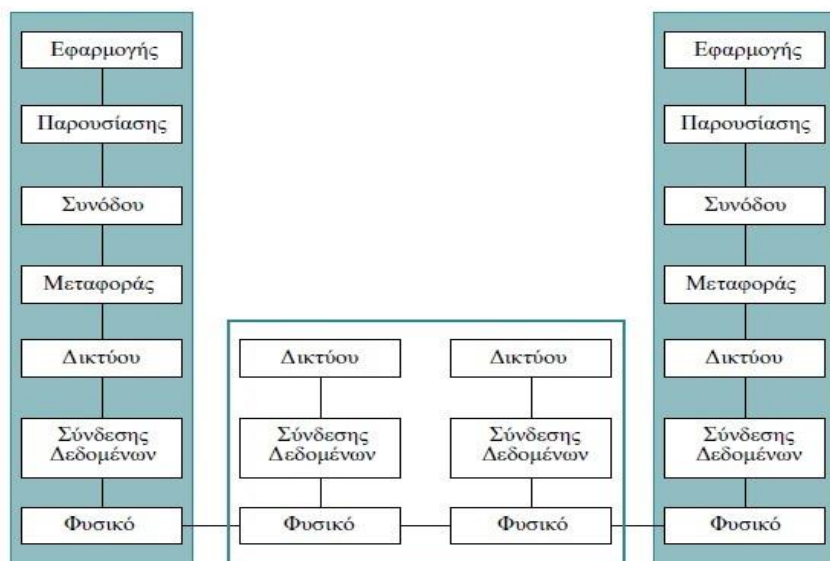
Στην εικόνα 1.4. βλέπουμε την επικοινωνία δύο Ηλεκτρονικών Υπολογιστών με βάση το τηλεφωνικό δίκτυο. Ένα μοντέλο επικοινωνίας δεδομένων περιγράφει τον εξοπλισμό, τις διεπαφές (interfaces) και τα μέσα μετάδοσης τα οποία ορίζουν το μονοπάτι από την πηγή στο δέκτη.

2. ΜΟΝΤΕΛΑ ΑΝΑΦΟΡΑΣ

2.1 Μοντέλο αναφοράς OSI

Το Μοντέλο OSI (Open Systems Interconnection) ή μοντέλο επτά επιπέδων (εικόνα 2.1), αναπτύχθηκε από το διεθνή Οργανισμό τυποποίησης ISO, και χρησιμοποιήθηκε για να ταξινομήσει και να δομήσει τη σουίτα πρωτοκόλλων. Το Μοντέλο OSI αποτελείται όπως αναφέραμε από επτά επίπεδα , τα οποία επικοινωνούν μεταξύ τους, αλλά οι λειτουργίες αυτών των επιπέδων είναι ανεξάρτητες, έτσι ώστε οι αλλαγές σε ένα επίπεδο να μην έχουν επίδραση στα άλλα επίπεδα. Το μοντέλο OSI, κατά κάποιο τρόπο υποδιαιρεί τις λειτουργίες ενός υπολογιστικού συστήματος ή ενός τηλεπικοινωνιακού δικτύου.

Το κάθε επίπεδο στο OSI, θα αξιοποιήσει τις λειτουργίες μέσω των πληροφοριών που θα λάβει από το αμέσως χαμηλότερο επίπεδο από αυτό. Γενικά τα επίπεδα στο OSI, έχουν την δυνατότητα να επικοινωνούν με άλλα επίπεδα από την στοίβα ασχέτως αν τα επίπεδά αυτά έχουν υλοποιηθεί με διαφορετικά πρωτόκολλα. [3.].



Εικόνα 2.1: Τα επίπεδα του OSI

2.1.1 Στρωματοποίηση κατά OSI

1. Φυσικό Επίπεδο

Το φυσικό επίπεδο ορίζει όλες τις ηλεκτρικές και φυσικές προδιαγραφές της επικοινωνίας. Δηλαδή Είναι υπεύθυνο για την μετάδοση των δεδομένων πάνω σε ένα μέσο μεταφοράς όπως, π.χ. η οπτική ίνα, το ομοαξονικό καλώδιο, Όπως επίσης καθορίζεται πως θα γίνει ο συγχρονισμός των συσκευών μεταξύ τους. Ποια στάθμη τάσης αντιστοιχεί στο bit 1 και ποια στο bit 0; Όπως ο ρυθμός μετάδοσης των bit, χρονική διάρκεια κ.ά.

2. Ζεύξης Δεδομένων

Το επίπεδο ζεύξης είναι το αμέσως επόμενο επίπεδο από το φυσικό. Παρέχει τα λειτουργικά και διαδικαστικά μέσα για τη μεταφορά δεδομένων από μια συσκευή ενός υπολογιστικού συστήματος ή ενός δικτύου σε μία άλλη. Είναι υπεύθυνο να μεταφέρει τα δεδομένα από το φυσικό επίπεδο και να τα προωθεί στο επίπεδο δικτύου . Αφού πρώτα κάνει τον απαραίτητο έλεγχο όπως ανίχνευση και διόρθωση σφαλμάτων που συμβαίνουν στο φυσικό επίπεδο. Βέβαια το επίπεδο ζεύξης μπορεί να εκτελέσει και την αντιστροφή λειτουργία δηλαδή να δέχεται πληροφορίες και δεδομένα από το Network Layer (Επίπεδο δικτύου) και να τις μεταβιβάζει στο Φυσικό επίπεδο.

3. Επίπεδο Δικτύου

Το επίπεδο αυτό είναι κατά κύριο λόγο υπεύθυνο για την σωστή μεταφορά των δεδομένων στους τελικούς προορισμούς. Εκτελεί την λειτουργία της δρομολόγησης των πακέτων, που είναι η καθοδήγηση των πακέτων στον τελικό αποδεκτή. όπως και για την διευθυνσιοδότηση. Δηλαδή την αντιστοίχιση μιας μοναδικής διεύθυνσης σε ένα υπολογιστή. Στο επίπεδο αυτό, τα routers (δρομολογητές) εκτελούν τις παραπάνω λειτουργίες.

4. Επίπεδο Μεταφοράς

Το επίπεδο μεταφοράς είναι υπεύθυνο για την αξιόπιστη μεταφορά και παράδοση των δεδομένων, από χρήστη σε χρήστη. Το επίπεδο μεταφοράς ελέγχει την αξιοπιστία ενός χρησιμοποιούμενου καναλιού με έλεγχο ροής κατάκτηση και αποτμηματοποίηση καθώς και έλεγχο σφαλμάτων . Το επίπεδο αυτό στην ουσία θεωρείτε ένα Software Interface μεταξύ των τριών χαμηλότερων επιπέδων και των υψηλότερων. Το συνηθέστερο παράδειγμα πρωτοκόλλου μεταφοράς είναι το TCP όπου θα δούμε και αναλυτικότερα σε άλλη ενότητα παρακάτω.

5. Επίπεδο Συνόδου

Το επίπεδο Συνοδού εκτελεί την λειτουργία της απρόσκοπτης επικοινωνίας των εφαρμογών παράλληλα σε διαφορετικά υπολογιστικά συστήματα, όπως π.χ. δύο ή τρεις Η/Υ επικοινωνούν μεταξύ με κάποιες εφαρμογές επικοινωνίας η οτιδήποτε άλλο. Δηλαδή θα διαχειριστεί την έναρξη και τον τερματισμό της επικοινωνίας, της ομαλής λειτουργίας κ.α.

6. Επίπεδο Παρουσίασης

Το επίπεδο παρουσίασης μετασχηματίζει τα δεδομένα σε τυπική μορφή που την αναμένει το επίπεδο εφαρμογών. Αυτό το επίπεδο επίσης χρησιμοποιεί και παρέχει βελτιωμένες εμπειρίες για την διασύνδεση χρηστών με κάποιον η κάποιους κεντρικούς σταθμούς, Κάποιες από τις υπηρεσίες που προσφέρει είναι: τα δεδομένα υφίστανται κρυπτογράφηση, συμπίεση, κωδικοποίηση και όποια άλλη διαμόρφωση απαιτεί η μορφή δεδομένων ή ο σχεδιαστής του πρωτοκόλλου.

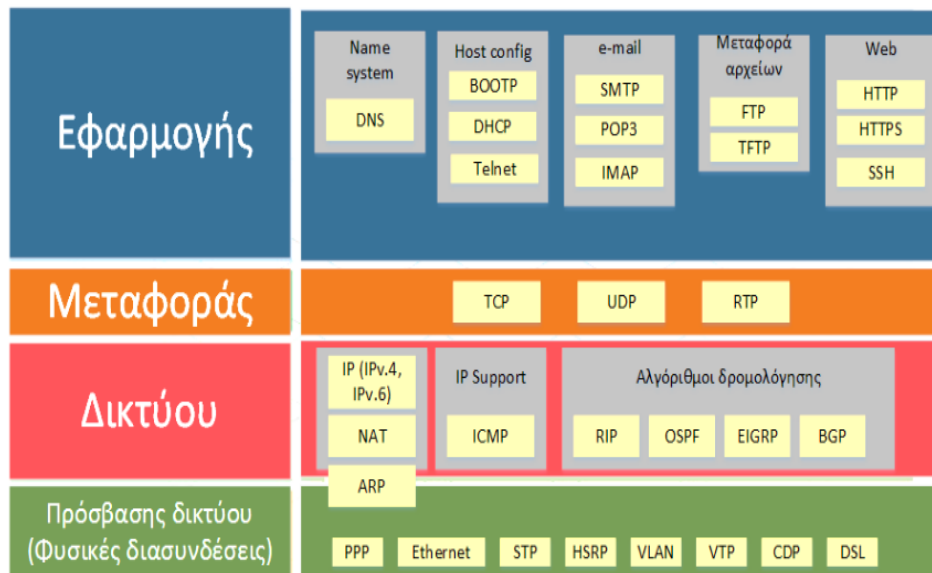
7. Επίπεδο Εφαρμογών

Το επίπεδο εφαρμογών είναι στην κορυφή τις λίστες όλων των επιπέδων. Όπως αναφέρει και το όνομα του παρέχει τις υπηρεσίες και τις εφαρμογές στον χρήστη για την επικοινωνία μεταξύ των Η/Υ.παρέχει επίσης και διαφορά πρωτοκολλά όπως το email, το FTP για μεταφορά αρχείων , τις remote control δηλαδή της απομακρυσμένη εργασία κ.α.[4.].

2.2 Το Μοντέλο TCP/IP

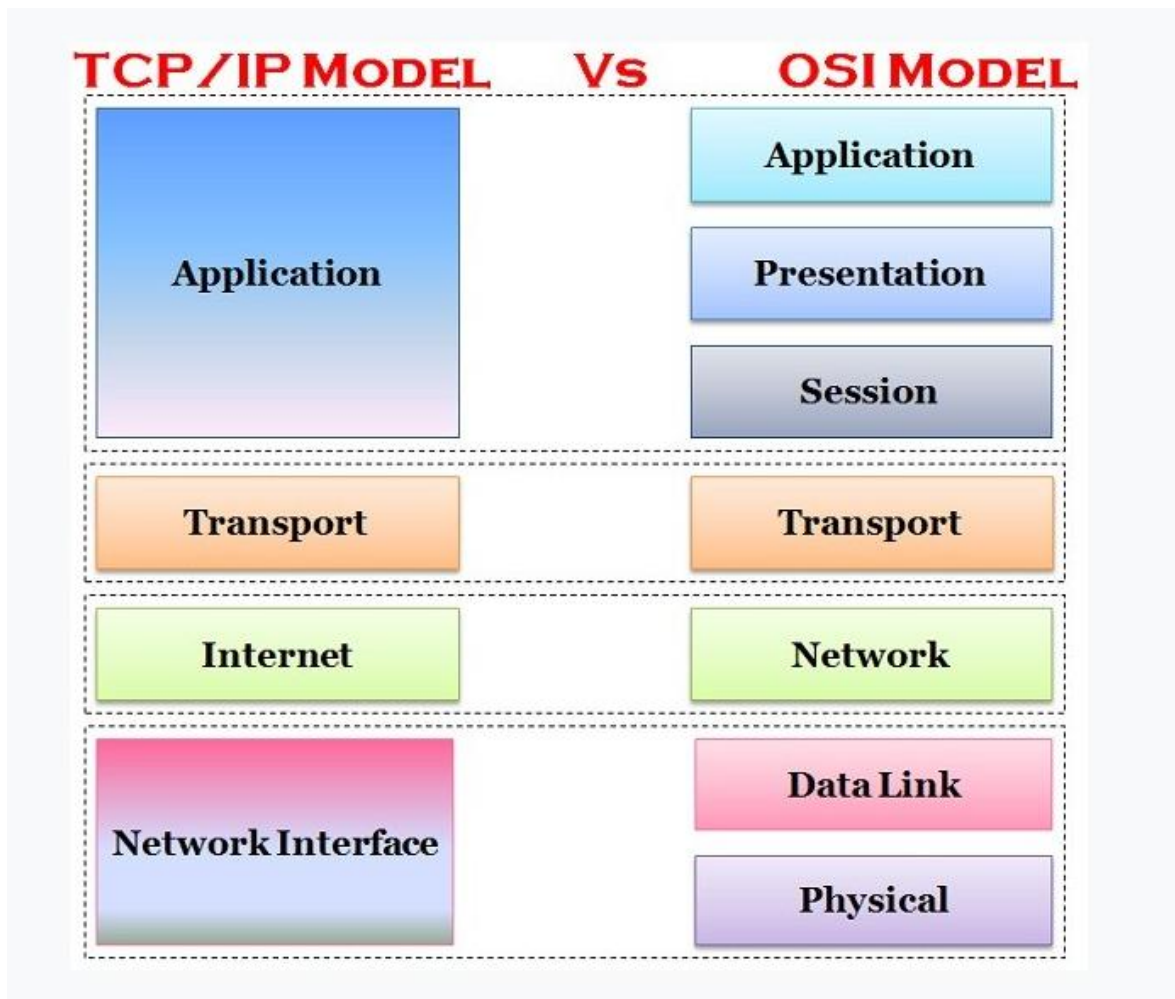
Όπως είχαμε επισημάνει και στην εισαγωγή το TCP/IP παρουσιάστηκε στις αρχές τις δεκαετίας του '70. Σήμερα αποτελεί το κυριότερο πρότυπο για τις ανάγκες επικοινωνίας στο Διαδίκτυο Συγκεκριμένα στηρίζει την επικοινωνία και την διασύνδεση των συσκευών στο Διαδίκτυο και βασίζεται σε διαφορά πρωτόκολλα επικοινωνίας (εικόνα 2.2.) για την σωστή

και ομαλή λειτουργία του διαδικτύου και των συσκευών. Αποτελείται και βασίζεται και αυτό όπως το Μοντέλο Αναφοράς OSI σε επίπεδα. Ποιο συγκεκριμένα το TCP/IP διαμορφώνει και καθορίζει την επικοινωνία των δεδομένων μεταξύ των συσκευών που είναι συνδεδεμένες στο Διαδίκτυο δηλαδή να φτάσουν τα πακέτα σωστά στον τελικό προορισμό όπως ακριβώς είχαν δηλαδή αποσταλεί καθορίζει με λίγα λόγια την σωστή αποστολή και δρομολόγηση των πακέτων στον προορισμό τους. Για να επιτευχθεί σωστά όλο αυτό το κάθε επίπεδο αποτελείτε από διαφορά άλλα πρωτόκολλα επικοινωνίας, όπως για παράδειγμα για την περιήγηση σε μια ιστοσελίδα του διαδικτύου χρησιμοποιούνται τα HTTP, και HTTPS πρωτόκολλα, ενώ για τη μεταφορά κάποιου αρχείου το FTP



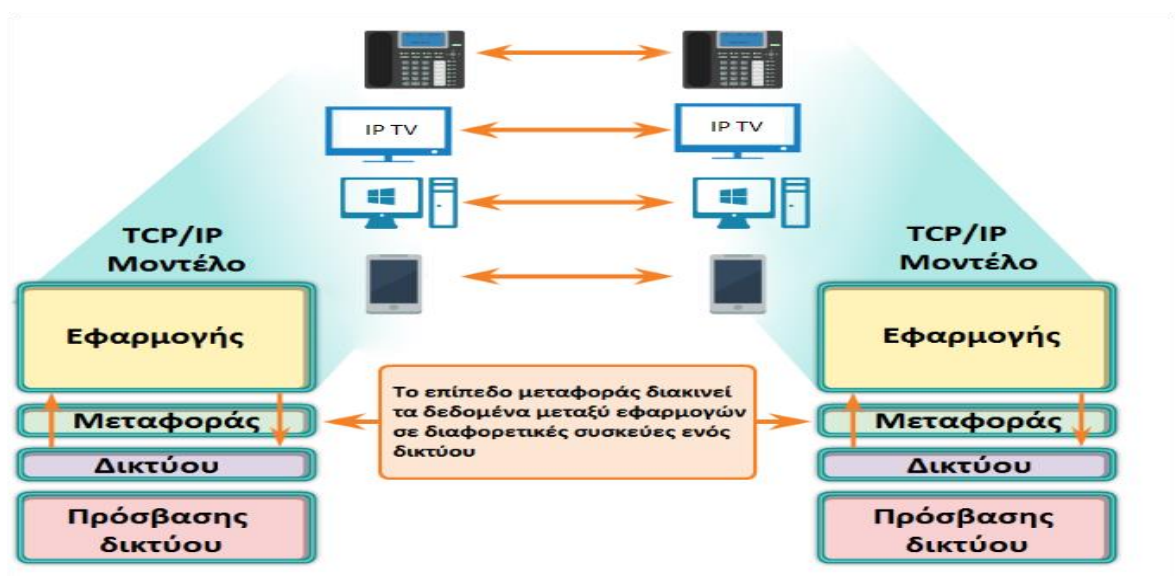
Εικόνα 2.2: Βασικά πρωτόκολλα του TCP/IP ανά επίπεδο

Το TCP/IP αποτελείτε από τα 4 επίπεδα (layers) από τα οποία κάποια είναι κοινά με αυτά του OSI, όμως αυτό δεν σημαίνει ότι αντιστοιχούν απολύτως, απλώς ταιριάζουν σε μεγάλο βαθμό. π.χ. Εικόνα 2.3



Εικόνα 2.3 Σύγκριση TCP/IP και OSI

Στην εικόνα 2.4 βλέπουμε ένα παράδειγμα σύνδεσης διαφόρων δικτυακών συσκευών μέσω του πρωτόκολλου μεταφοράς



Εικόνα 2.4: Σύνδεση TCP/IP

2.2.1 Τα επίπεδα του TCP/IP

Το μοντέλο TCP/IP όπως αναφέραμε παραπάνω αποτελείται από τέσσερα επίπεδα ή στρώματα (layers). Κάθε επίπεδο όπως και στην περίπτωση του OSI, υλοποιεί κάποια πρωτόκολλα. Αυτά τα τέσσερα επίπεδα είναι:

- 1) *Επίπεδο Φυσικό ή επίπεδο από Host προς Host.*
- 2) *Επίπεδο Διαδικτύου.*
- 3) *Επίπεδο Μεταφοράς.*
- 4) *Επίπεδο Εφαρμογής.*

1. Φυσικό Επίπεδο

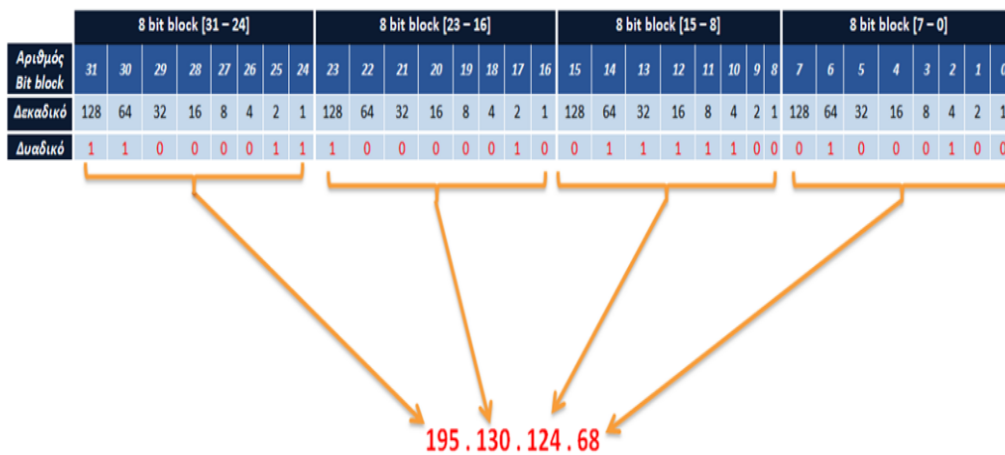
Το φυσικό επίπεδο, όπου αποτελείται από την ζεύξη δεδομένων και το φυσικό επίπεδο, είναι υπεύθυνο για την επικοινωνία του σταθμού με το δίκτυο. Πρόκειται για το κατώτερο επίπεδο του μοντέλου, Είναι υπεύθυνο για την παροχή μιας διεπαφής που θα του επιτρέψει την επικοινωνία με το διαδίκτυο. Αυτό το επίπεδο καθορίζει το φυσικό μέσο που χρησιμοποιείται για την επικοινωνία των συσκευών μετάδοσης δηλαδή για τα δομικά μέσα δικτύωσης σε ότι έχει να κάνει με καλώδια π.χ. UTP ή καλώδιο οπτικής ίνας ή κάρτα δικτύου.[5.]

2. Το Επίπεδο Διαδικτύου

Το επίπεδο Διαδικτύου (Internet Layer) βρίσκεται στο 2^ο επίπεδο του TCP/IP πάνω από το φυσικό επίπεδο. Είναι υπεύθυνο για την μεταφορά των πακέτων μέσα σε ένα δίκτυο, στην ουσία καθορίζει την δρομολόγηση των πακέτων στο διαδίκτυο δεν υπάρχει εγγυημένη παράδοση. Ενδεχόμενος τα πακέτα μπορεί να μην φτάσουν καν στον προορισμό τους η και με λάθος σειρά. Σε αυτό το επίπεδο το βασικό πρωτόκολλο είναι το IP, το οποίο καθορίζει την διευθυνσιοδότηση σε ένα υπολογιστικό σύστημα δηλαδή καταχωρεί την Διεύθυνση IP σε μία συσκευή. [5.]

➤ Το πρωτόκολλο IP (IP – Internet Protocol)

Το πρωτόκολλο IP βρίσκεται στο επίπεδο Διαδικτύου του TCP/IP. Η τρέχουσα έκδοση είναι η τέσσερα όπου αναφέρατε ως IPv4. Το πρωτόκολλο IP είναι υπεύθυνο για την μεταφορά ή το πέρασμα των πακέτων από έναν υπολογιστή σε κάποιον άλλον ή σε περισσότερους προορισμούς, οτιδήποτε είναι συνδεδεμένο στο Διαδίκτυο (Internet) αντιστοιχίζεται σε μία μοναδική διεύθυνση προορισμού που ονομάζεται διεύθυνση IP. Όπως είχαμε αναφέρει σε προηγούμενη ενότητα, κάποια από τα δομικά στοιχεία των δικτύων των Η/Υ είναι και τα routers (δρομολογητές) κι έτσι μέσω αυτών μπορούν ανταλλάσσουν τα δεδομένα με βάση της IP σε κάθε προορισμό.

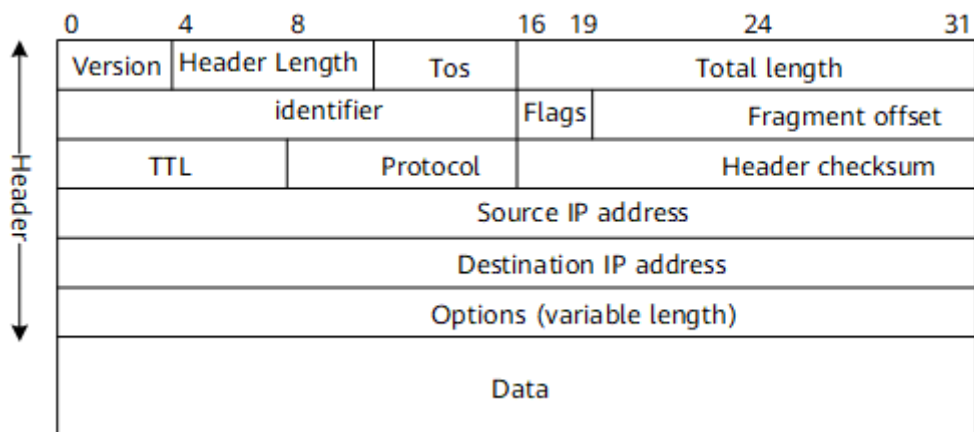


Εικόνα 2.5: Αναλυτική απεικόνιση δομής μιας IP διεύθυνσης

Στην εικόνα 2.5 βλέπουμε την αναλυτική δομή μιας IP διεύθυνσης. Αποτελείται από τέσσερις οκτάδες bits με βάση το δυαδικό σύστημα.

Μορφή Πακέτου IPv4

Ένα πακέτο αποτελείται από το τμήμα της επικεφαλίδας και το τμήμα δεδομένων. Κάθε πακέτο IPv4 έχει την μορφή της Εικόνας 2.6. όπου αποτελείται από 14 πεδία. [5.]



Εικόνα 2.6: Datagram IPv4

Τα υποχρεωτικά πεδία στην κεφαλίδα του IPv4 περιγράφονται στην συνέχεια. [5.]

1. Το Πεδίο “**Version**” είναι μεγέθους 4 bit με τιμή 4 .Για το λόγο είναι της έκδοσης: IPv4.
2. Το Πεδίο “**Header Length**” το πεδίο αυτό είναι μεγέθους 4 bit περιέχοντας το συνολικό μήκος της κεφαλίδας περιγράφεται όμως σε λέξεις των 32 bit (4 byte)..Συνήθως οι λέξεις του είναι 5, άρα και η τιμή του είναι 5.
3. Το πεδίο ‘**Type of Service**’- Τύπος (ή Ποιότητα) Υπηρεσίας. Το μέγεθος του είναι 8 bits (1 byte). Ενημερώνει τον δρομολογητή με ποιον τρόπο θα δρομολογηθεί ένα πακέτο.
4. Το πεδίο ‘**Total Length**’ Περιγράφει το συνολικό μήκος του πακέτου IP, το οποίο είναι 16 bit. εκφρασμένου σε byte, το οποίο έχει μήκος, δηλαδή 65.535 byte.
5. Το πεδίο ‘**Identification**’ ή ταυτότητα. Όταν τα διάφορα κομμάτια του πακέτου κατακερματιστούν πρέπει να έχουν τον ίδιο αναγνωριστικό αριθμό του πακέτου, έτσι ώστε να υποδηλώνουν μέρη του ίδιου πακέτου. Είναι μεγέθους 16 Bit.
6. Το πεδίο ‘**Flags**’. Εδώ υπάρχουν τρία συνολικά bit που χρησιμοποιούνται ως σημαίες.
7. Το πεδίο ‘**Fragmentation Offset**’. Σε αυτό το πεδίο μας αναφέρεται για το αρχικό πακέτο δεδομένων η απόσταση σε byte των δεδομένων και είναι εκφρασμένη ως δύναμη του 8. Για παράδειγμα εάν αυτή η τιμή είναι x, τότε το πρώτο byte στο παρόν τεμάχιο αντιστοιχεί στην θέση με αύξοντα αριθμό $8*x$ στα δεδομένα του αρχικού πακέτου.

8. Το πεδίο ‘TTL’ σημαίνει “Time To Live”. Κάθε φορά που το πακέτο περνάει από έναν δρομολογητή, μειώνεται κατά ένα, είναι ουσιαστικά ένας μετρητής εάν μηδενισθεί, τότε το πακέτο αυτό απορρίπτεται και ο δρομολογητής αποστέλλει κατάλληλο μήνυμα (ICMP) στον αποστολέα του πακέτου .
9. Το πεδίο ‘Protocol’. Σε αυτό το πεδίο γίνεται ενημέρωση για τον τύπο του πακέτου που φθάνει στο επίπεδο μεταφοράς . Στην Ουσία είναι κάτι σαν κωδικός του πακέτου, για παράδειγμα: 22 για το TCP, ή 34 για το UDP.
10. Το ‘Header Checksum’. Σε αυτό το πεδίο γίνεται ενημέρωση για τον ορθό έλεγχο της μετάδοσης της επικεφαλίδας. Το μέγεθος της είναι 16 Bit.
11. Το πεδίο ‘Source Address’. Είναι η διεύθυνση IP του αποστολέα.
12. Το πεδίο ‘Destination Address’. Η διεύθυνση IP του προορισμού.
13. **Επιλογές IP (IP Options):** Αφορά σε διαγνωστικούς και διαχειριστικούς ελέγχους. Συνήθως δεν χρησιμοποιείται.
14. **Data:** Το τμήμα δεδομένων του πακέτου.

3. Το Επίπεδο Μεταφοράς

Σε αυτό το επίπεδο ορίζεται μία αξιόπιστη σύνδεση για μεταφορά πακέτων από μια συσκευή που είναι συνδεδεμένη στο διαδίκτυο σε μία άλλη. Διασφαλίζει ότι τα πακέτα θα φτάσουν στον προορισμό με ταχύτητα και με ακρίβεια. Χρησιμοποιείτε το πρωτόκολλο μεταφοράς TCP όπου και θα το αναλύσουμε στην συνέχεια που με βάση διάφορους αλγορίθμους παρέχει αξιοπιστία και συνέπεια στην μεταφορά των πακέτων χρησιμοποιώντας όπως έλεγχο ροής, διόρθωση σφαλμάτων. Εκτός από την αξιόπιστη μεταφορά με σύνδεση το TCP, πάνω στο επίπεδο μεταφοράς βρίσκεται και μεταφορά πακέτων χωρίς σύνδεση του πρωτοκόλλου UDP. Οπού δεν παρέχεται αξιοπιστία στην παράδοση των πακέτων. [5.]

4. Το Επίπεδο Εφαρμογής

Το Επίπεδο Εφαρμογής (Application Layer) είναι το ανώτερο επίπεδο από τα τέσσερα επίπεδα του μοντέλου TCP/IP και βρίσκεται πάνω από το επίπεδο Μεταφοράς (Transport Layer). Είναι υπεύθυνο για τις εφαρμογές που χρησιμοποιούνται για την μεταφορά των δεδομένων και πακέτων δηλαδή για μια μεταφορά αρχείου, η αποστολή ενός email. Αυτό

που κάνει δηλαδή είναι να εφαρμόζει τα πρωτοκολλά εφαρμογής, και την λειτουργία των προγραμμάτων σε σχέση με την επικοινωνία των χρηστών

Το επίπεδο εφαρμογής περιλαμβάνει τα πρωτόκολλα όπως τα:

- HTTP (Hypertext Transfer Protocol), για την μεταφορά υπερκείμενου στο Διαδίκτυο.
- FTP (Πρωτόκολλο Μεταφοράς Αρχείων).
- POP, IMAP. Για την αποστολή ενός email.

2.3 Πρωτόκολλα Μεταφοράς

2.3.4 TCP Transmission Control Protocol

Είναι το κυριότερο και σημαντικότερο πρωτόκολλο του διαδικτύου. Είναι υπεύθυνο για τον μεγαλύτερο όγκο διακίνησης δεδομένων στο internet, το οποίο παρέχει την αξιοπιστία αποστολής και λήψης δεδομένων. Και θα το αναλύσουμε διεξοδικά στο επόμενο κεφάλαιο.

2.3.2 SCTP - Streaming Control Transmission Protocol

Το Streaming Control Transmission Protocol (SCTP) είναι ένα πρωτόκολλο επικοινωνίας δικτύων υπολογιστών στο επίπεδο μεταφοράς του Internet Protocol Suite (IP). αναπτύχθηκε με στόχο την μεταφορά τηλεπικοινωνιακών σημάτων, είναι ένα πρωτόκολλο unicast και υποστηρίζεται η παράδοση δεδομένων από άκρο σε άκρο εντός ακριβώς δύο τελικών σημείων. Ωστόσο, τα τελικά σημεία ενδέχεται να έχουν περισσότερες από μία διευθύνσεις IP. Όπως και το TCP, το SCTP είναι πρωτόκολλο με σύνδεση (Connection oriented) και προσφέρει αξιόπιστες υπηρεσίες μεταφοράς, εξασφαλίζοντας ότι τα δεδομένα στο δίκτυο θα μεταδοθούν χωρίς λάθη και με τη σωστή σειρά. Ωστόσο, σε αντίθεση με το TCP το SCTP μεταφέρει μηνύματα, όχι απλά bytes. Γι' αυτόν τον λόγο, το SCTP χαρακτηρίζεται και ως message-oriented πρωτόκολλο. Το SCTP μπορεί να χαρακτηριστεί ως προσανατολισμένο στα μηνύματα, που σημαίνει ότι μεταφέρει μια ακολουθία μηνυμάτων (το καθένα είναι μια ομάδα byte), αντί να μεταφέρει ένα αδιάκοπο ρεύμα byte όπως το TCP.

Τα χαρακτηριστικά του SCTP περιλαμβάνουν:

- Αξιόπιστη μετάδοση τόσο διατεταγμένων όσο και μη ταξινομημένων ροών δεδομένων.

- Πολλαπλή υποστήριξη στην οποία ένα ή και τα δύο τελικά σημεία μιας σύνδεσης μπορούν να αποτελούνται από περισσότερες από μία διευθύνσεις IP, επιτρέποντας διαφανή αποτυχία μεταξύ περιττών διαδρομών δικτύου.
- Η παράδοση κομματιών σε ανεξάρτητα ρεύματα εξαλείφει τον άσκοπο αποκλεισμό της γραμμής, σε αντίθεση με την παράδοση byte-stream TCP. [7.]

2.3.1 UDP: User Datagram Protocol

Το UDP είναι ένα πρωτόκολλο του επιπέδου μεταφοράς. Το οποίο είναι προσανατολισμένο χωρίς σύνδεση, σε σχέση με το TCP. Θεωρείται ένα πρωτόκολλο αναξιόπιστο. Για την μεταφορά πακέτων στο διαδίκτυο. Δεν υπάρχει καμία εγγύηση για ορθή παραλαβή. Επί πλέον, καμία σχετική ενημέρωση δεν παρέχεται στον αποστολέα και παραλήπτη για την παράδοση των πακέτων. Δεν υποστηρίζει μηχανισμούς για την αξιόπιστη αποστολή και παραλαβή των πακέτων δηλαδή μηχανισμούς ελέγχου ροής και σφαλμάτων. Για παράδειγμα χρησιμοποιείτε στις εφαρμογές video streaming ή την προβολή κάποιου video στον Web, η στις εφαρμογές ήχου όπου εκεί σημασία έχει κατά κύριο λόγο η ταχύτητα να φτάσει στον τελικό προορισμό. Και αν εάν απαιτείται επαλήθευση για την επιβεβαίωση αποστολής του πακέτου μπορεί να πραγματοποιηθεί στο επίπεδο εφαρμογής του TCP. [6.]

2.3.3 DCCP Datagram Congestion Control Protocol

Το Datagram Congestion Control Protocol (DCCP) είναι ένα πρωτόκολλο επιπέδου μεταφοράς προσανατολισμένο προς τα μηνύματα. Η διαφορά του από το διαδεδομένο UDP, είναι ότι παρέχει έλεγχο συμφόρησης στις unreliable ροές.. ο DCCP είναι χρήσιμο για εφαρμογές με χρονικούς περιορισμούς στην παράδοση δεδομένων. Τέτοιες εφαρμογές περιλαμβάνουν streaming media, online παιχνίδια για πολλούς παίκτες και τηλεφωνία μέσω Διαδικτύου. Σε τέτοιες εφαρμογές, τα παλιά μηνύματα γίνονται γρήγορα άχρηστα, έτσι ώστε η λήψη νέων μηνυμάτων να προτιμάται από την εκ νέου αποστολή χαμένων μηνυμάτων.

Κάποια από τα χαρακτηριστικά του DCCP είναι:

- Επιλογή μηχανισμού ελέγχου συμφόρησης. Ο κύριος στόχος του DCCP είναι το να προσφέρει μηχανισμούς ελέγχου συμφόρησης για ροές δεδομένων. Το DCCP

μπορεί να υποστηρίξει διαφορετικούς μηχανισμούς οι οποίοι αναγνωρίζονται από έναν αριθμό CCID (Congestion Control Identifier).

- Unreliable μεταφορά datagrams με επιβεβαίωση. Όπως και στο UDP δεν υπάρχει αναμετάδοση των χαμένων πακέτων. Οι επιβεβαιώσεις πακέτων, που είναι αναγκαίες για τους μηχανισμούς ελέγχου συμφόρησης, μπορεί να γίνουν με διαφορετικούς τρόπους όπως με διανύσματα επιβεβαίωσης ή με επιβεβαίωση για κάθε πακέτο. [8.]

3. TCP: Transmission Control Protocol

Το Transmission Control Protocol (TCP) είναι το κύριο πρωτόκολλο μεταφοράς που χρησιμοποιείται στο Internet. Είναι αρμόδιο για τον μεγαλύτερο όγκο του γραπτού υλικού που έχει γραφεί για οποιοδήποτε πρωτόκολλο δικτύωσης.. Αυτό απεικονίζει βεβαίως τη σημασία του TCP στο Διαδίκτυο, όπου είναι το πιο χρησιμοποιημένο πρωτόκολλο μεταφοράς. Γενικά στο Internet το πρωτόκολλο Ελέγχου Μετάδοσης TCP είναι ένα πρωτόκολλο του Επιπέδου Μεταφοράς το οποίο παρέχει αξιοπιστία. Ο σκοπός του είναι να παραλαμβάνει και να μεταδίδει αξιόπιστα τις πληροφορίες μεταξύ εφαρμογών που συνδέονται. Η λειτουργία του γίνεται «πάνω» από το πρωτόκολλο IP, το οποίο είναι μη αξιόπιστο για μεταφορά.

3.1 Χαρακτηριστικά TCP

Ένα από τα κύρια χαρακτηριστικά του TCP είναι όταν μια σύνδεση εγκαθιδρυθεί με επιτυχία μεταξύ αποστολέα και παραλήπτη, τότε όλα τα δεδομένα μεταξύ δύο υπολογιστών αποστέλλονται με την μορφή πακέτων. Τα οποία κάποια από τα κύρια χαρακτηριστικά είναι τα εξής :

- **Παρέχει Αξιοπιστία:** Σε μια σύνδεση στο TCP ανάμεσα στον αποστολέα και τον παραλήπτη, το TCP χρησιμοποιεί κάποιους μηχανισμούς για να διασφαλίσει ότι τα πακέτα που αποστέλλονται θα φτάσουν επιτυχώς στον παραλήπτη. Δηλαδή αν το πακέτο που στάλθηκε θα επιβεβαιωθεί, την επαναποστολή πακέτων που χάθηκαν, αν κάποιο πακέτο έχει χαθεί τότε πρέπει να ξανασταλθεί, ή επίσης αν ένα πακέτο δεν έχει φθάσει στον προορισμό του, τότε να ζητηθεί από τον παραλήπτη η επαναποστολή του.

- **Η Σειρά πακέτων:** Όταν σε μία σύνδεση αποσταλούν δύο πακέτα διαδοχικά στην σειρά, πρέπει να φθάσουν στον τελικό προορισμό και με τη σειρά που στάλθηκαν, έτσι το TCP παρέχει και στον αποστολέα και στον παραλήπτη, την ασφάλεια ότι τα πακέτα θα φθάσουν στον προορισμό τους με την σωστή σειρά.
- **Πλήρης αμφίδρομη επικοινωνία(Full duplex).** Μια σύνδεση TCP επιτρέπει την ταυτόχρονη μεταφορά των δεδομένων και προς τις δυο κατευθύνσεις, και παρέχει τη δυνατότητα και στα δύο προγράμματα εφαρμογών να στέλνουν δεδομένα οποιαδήποτε στιγμή. Επιτρέπει την ταυτόχρονη χρήση του από πολλές εφαρμογές ταυτόχρονα σε ένα μόνο υπολογιστή. Είναι για παράδειγμα δυνατόν, ταυτόχρονα σε ένα υπολογιστή να βλέπουμε ιστοσελίδες (περισσότερες της μίας ταυτόχρονα, και μάλιστα αν θέλουμε χρησιμοποιώντας και διαφορετικά προγράμματα φυλλομετρητή), να ελέγχουμε το ηλεκτρονικό μας ταχυδρομείο και να μεταφέρουμε αρχεία μέσω της υπηρεσίας μεταφοράς αρχείων (το γνωστό FTP). [9.]

3.2 Αρχιτεκτονική TCP

Το TCP στην ουσία είναι ένα σύστημα από περίπλοκους μηχανισμούς που εξασφαλίζουν την σωστή λειτουργία του πρωτοκόλλου. Στην συνέχεια θα δούμε την αρχιτεκτονική του TCP δηλαδή από τι αποτελείται και το καταντά λειτουργικό σε μια σύνδεση αναμεσα από υπολογιστικά συστήματα όπως είναι οι θύρες (Ports) επικοινωνίας, η μορφή των δεδομένων, ή επικεφαλίδα του TCP κ.α. Αναλύοντας όλα τα παραπάνω θα διαπιστώσουμε πόσο περίπλοκο είναι πραγματικά το TCP, είναι στην ουσία ένα ολόκληρο σύστημα από διαδικασίες που αλληλοεπιδρούν μεταξύ τους, και διαδικασίες που έχουν σχεδιασθεί να εκτελούν ένα σύνολο από καλά ορισμένους στόχους.

Υπολογιστής A



Υπολογιστής B



Εικόνα 3.1: Σύστημα επικοινωνίας του TCP

Αυτό που βλέπουμε και στην παραπάνω εικόνα 3.1 είναι ότι το λογισμικό TCP του A Υπολογιστή επικοινωνεί με το λογισμικό του B Υπολογιστή. Όταν «ο υπολογιστής A κάνει μία σύνδεση με τον υπολογιστή B» αυτό που συμβαίνει είναι ότι το λογισμικό TCP του υπολογιστή A έχει κάνει μία σύνδεση με το λογισμικό TCP του υπολογιστή B, όπου και οι δύο ενεργούν για μία τοπική εφαρμογή.

Οι ακραίοι κόμβοι είναι υπεύθυνοι για την επαλήθευση των επικοινωνιών σε ένα TCP/IP δίκτυο. Οι ακραίοι κόμβοι είναι αυτοί που προσπαθούν να επικοινωνήσουν – αντίθετα με τους ενδιάμεσους κόμβους που προωθούν το μήνυμα. Σε ένα τοπικό δίκτυο όπως παραπάνω, τα δεδομένα περνούν από το υποδίκτυο προέλευσης στο υποδίκτυο προορισμού με δρομολογητές. Αυτοί οι δρομολογητές γενικά λειτουργούν στο επίπεδο δικτύου ή internet. Το σημαντικό είναι ότι οι δρομολογητές δεν ενδιαφέρονται για τις πληροφορίες του επιπέδου μεταφοράς, απλώς περνούν τα δεδομένα στο επίπεδο μεταφοράς ως μεταφορείς των IP Datagram.

3.3 Επικεφαλίδα TCP

Τα πακέτα του πρωτοκόλλου TCP διατυπώνονται ως τμήματα (segments). Η TCP επικεφαλίδα (TCP header) είναι από τα κυριότερα μέρη ενός segment που δίνει συγκεκριμένες πληροφορίες για το πρωτόκολλο TCP (εικόνα 3.2).

TCP επικεφαλίδα				
+	Bits 0 - 3	4 - 9	10 - 15	16 - 31
0	Source Port Θύρα Προέλευσης		Destination Port Θύρα Προορισμού	
32	Sequence Number Αριθμός ακολουθίας			
64	Acknowledgment Number Αριθμός επιβεβαίωσης			
96	Data Offset	Reserved	Flags Σημαίες	Window Παράθυρο
128	Checksum Άθροισμα ελέγχου		Urgent Pointer Επείγοντα δεδομένα	
160	Options Επιλογές (προαιρετικές)			
160/192+	Data Δεδομένα			

Εικόνα 3.2: TCP Επικεφαλίδα

Η επικεφαλίδα του TCP περιλαμβάνει τα ακόλουθα πεδία: [9.]

- **Θύρα Προέλευσης (Source Port) (16-bit):** Προσδιορίζει την port (θύρα) του αποστολέα.
- **Θύρα Προορισμού (Destination Port) (16-bit):** Προσδιορίζει την port (θύρα) του παραλήπτη.
- **Αριθμός Σειράς (Sequence Number) (32-bit):** Ο sequence number (αριθμός ακολουθίας) όπου:

α) Για το πρώτο byte, αντιστοιχίζεται ο αρχικός αριθμός της σειράς, έκτος αν υπάρχει SYN flag, όπου τότε η σημαία πάρει τον αριθμό ένα, τότε ο αρχικός αριθμός ακολουθίας είναι ISN+1

β) Αλλιώς, εάν δεν υπάρχει η SYN flag, τότε η πρώτη octet δεδομένων είναι ο αριθμός ακολουθίας.

- **Αριθμός Αναγνώρισης (Acknowledgment number) (32-bit):** Όταν ληφθεί ένα πακέτο γνωστοποιείτε ότι λήφθηκε. Και είναι ο αριθμός του τελευταίου byte που έχει ληφθεί, όπου τονίζεται ως +1
- **Μετατόπιση Δεδομένων (Data Offset) (4-bit):** Ενημερώνει το μέγεθος της επικεφαλίδας, και την αρχή των δεδομένων, και είναι μεγέθους 32 bit.
- **Δεσμευμένο (Reserved) (6-bit):** Πεδίο 6 bit για μελλοντική χρήση. Η τιμή των bit πρέπει να είναι 0.
- **Σημαίες Ελέγχου (Flags) (1 bit η κάθε μία):** (Εικόνα 3.3)

Σημαία	Σημασία	Προέλευση ονομασίας
URG	Το πεδίο urgent pointer είναι σημαντικό	UR Gent
ACK	Το πεδίο επιβεβαίωσης είναι σημαντικό	ACK nowledgment
PSH	Λειτουργία ώθησης	Pu SH
RST	Επαναρύθμιση σύνδεσης	Re SeT
SYN	Συγχρονισμός αριθμών ακολουθίας	SYN chronize
FIN	Ο αποστολέας δεν στέλνει άλλα δεδομένα	FIN (=τέλος)

Εικόνα 3.3: Σημαίες Ελέγχου (Flags)

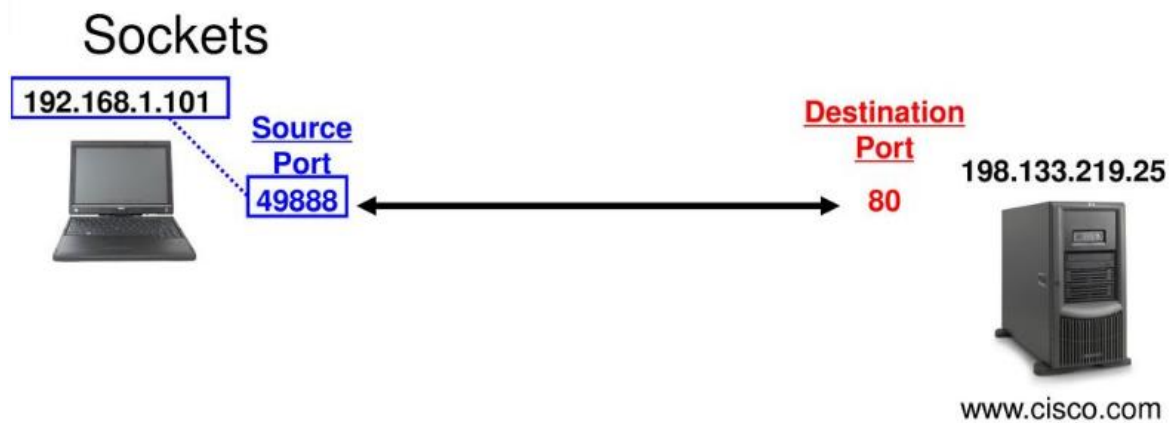
- **Παράθυρο (Window) (16-bit):** Ο αριθμός των επιτρεπόμενων δεδομένων που μπορεί να δεχθεί ο αποστολέας.
- **Checksum (16-bit):** χρησιμοποιείται για έλεγχο λαθών στην επικεφαλίδα και στα δεδομένα
- **Επιλογές (Option):** Το πεδίο αυτό καθορίζει κάποιες ρυθμίσεις της επικεφαλίδας
- **Δείκτης Επείγοντος (Urgent pointer) (16-bit):** Πεδίο που ελέγχει την ακεραιότητα του αποσπάσματος.
- **Δεδομένα (Data):** Τα Δεδομένα που μεταδίδονται με το απόσπασμα.

3.4 Θύρες (Ports)

Στο σύστημα TCP, οι διάφορες εφαρμογές μπορούν να αποστείλουν ή να παραλάβουν τα δεδομένα μέσω του επιπέδου μεταφοράς TCP. Όμως για να είναι δυνατή η διάκριση μεταξύ των εφαρμογών χρησιμοποιούνται τα λεγόμενα internet ports.

Θύρα ή port είναι μία προκαθορισμένη εσωτερική διεύθυνση η οποία ορίζει μία διαδρομή από μια συγκεκριμένη εφαρμογή προς το επίπεδο μεταφοράς. Οι θύρες των διαφόρων

εφαρμογών ορίζονται από τον διεθνή οργανισμό ICANN. Κάθε θύρα έχει ένα συγκεκριμένο ακέραιο αριθμό (1 έως 65535) όπου οι εφαρμογές χρησιμοποιούν διαφορετικές θύρες για τις ανάγκες τις επικοινωνία τους. π.χ. εικόνα 3.4. [10.]



Εικόνα 3.4: Η επικοινωνία δύο Η/Υ, με βάση τις θύρες (Ports)

Συνήθη Ports

Port	TCP ή UDP	Protoco l	Σχόλια
20,21	TCP	FTP	File Transfer Protocol
22	TCP	SSH	Secure Shell
23	TCP	Telnet	
25	TCP	SMTP	Simple Mail Transfer Protocol
110	TCP	POP3	Post Office Protocol 3. Για λήψη emails.
80	TCP	HTTP	Hypertext Transfer Protocol
443	TCP	HTTPS	Secure HTTP
53	TCP/UDP	DNS	Domain Name Service
161,162	UDP	SNMP	Simple Network Management Protocol

Εικόνα 3.5: Διάφορα Συνήθη Ports

Στην εικόνα 3.5 παρουσιάζονται κάποια από τα πιο συνηθισμένα ports που χρησιμοποιούνται στο διαδίκτυο.

3.5 Συνδέσεις TCP

Το TCP όπως αναφέραμε και ξέρουμε είναι ένα πρωτόκολλο προσανατολισμένο σε συνδέσεις., προκειμένου δύο σημεία να επικοινωνήσουν πρέπει πρώτα να εγκατασταθεί σύνδεση και μέσω αυτής, στη συνέχεια να γίνεται επικοινωνία (εκπομπή και λήψη δεδομένων). Ένας από τους λόγους ύπαρξης του TCP είναι να παρέχει μια διασύνδεση, έτσι

ώστε οι εφαρμογές να μπορούν να έχουν πρόσβαση στο Διαδίκτυο. Αυτή η σύνδεση παρέχεται μέσω των TCP θυρών που αναφέραμε στην προηγούμενη ενότητα, Στο TCP πρέπει να είναι ανοικτή διασύνδεση με την εφαρμογή. Το TCP υποστηρίζει 2 καταστάσεις ανοίγματος:

1. **Παθητική Ανοικτή:** Οι servers του διαδικτύου βρίσκονται πάντα σε μία κατάσταση αναμονής, αυτό σημαίνει ότι είναι σε ετοιμότητα να λάβουν και να στείλουν τα δεδομένα. Είναι μια διαδικασία που ενημερώνει το TCP ότι μπορεί να πραγματοποιήσει συνδέσεις.
 2. **Ενεργητική Ανοικτή:** Αφού έχει ενημερωθεί ο TCP ότι μπορεί να πραγματοποιηθεί μια σύνδεση, τότε με την σειρά τους οι εφαρμογές μπορούν να αιτηθούν μια σύνδεση για εγκατάσταση.
- **Διακομιστής (Server):** Είναι το λογισμικό το οποίο προσφέρει υπηρεσίες.
 - **Πελάτης (Client):** 'Ο οποίος ζητά και λαμβάνει υπηρεσίες.

3.6 Λειτουργία του TCP

Η μετάδοση των πληροφοριών μεταξύ των δύο άκρων με τη χρήση του πρωτοκόλλου TCP γίνεται σε τρεις φάσεις:

1. Εγκατάσταση Σύνδεσης
2. Μεταφορά δεδομένων
3. Τερματισμός Σύνδεσης

Είναι γνωστή σαν τριμερής χειραγία / 3-way handshake. Όταν ένας server είναι σε κατάσταση αναμονής και είναι έτοιμος να δεχθεί κλήση από κάποιον client, βρίσκεται στην κατάσταση παθητικής σύνδεσης.. Στην επομένη φάση οι εφαρμογές μπορούν να αιτηθούν την σύνδεση για εγκατάσταση.

πρέπει να έχει και μια δεσμευμένη πόρτα – port, η λεγομένη παθητική σύνδεση (passive open). Αμέσως μετά περνά στην φάση active open, και ξεκινά η σύνδεση.

1. Εγκατάσταση Σύνδεσης – Έναρξη Σύνδεσης

Βήμα 1: Το 1^ο βήμα είναι ο υπολογιστής 1 ή πελάτης να στείλει ένα πακέτο segment με αρχικό αριθμό ακολουθίας ISN (Initial Sequence Number). (εικόνα 3.6).

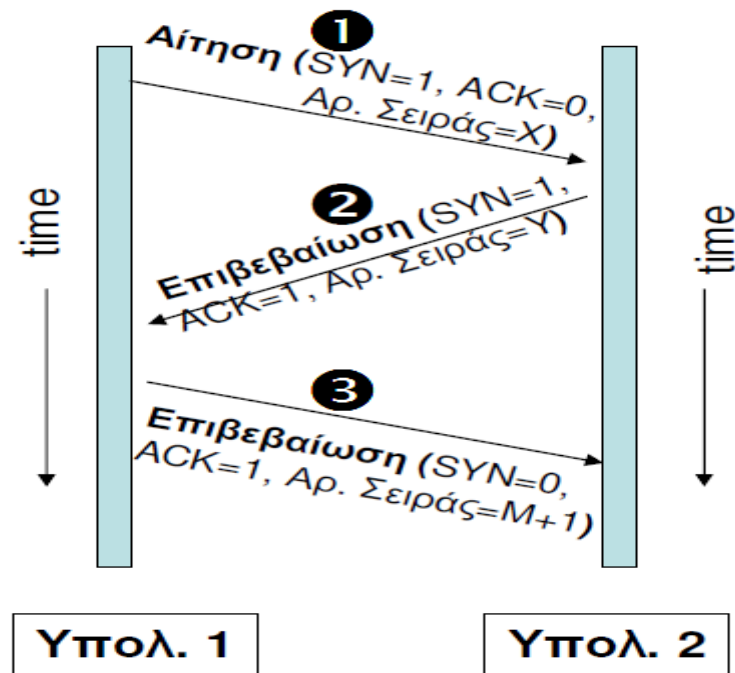
Βήμα 2: Στην Συνέχεια ο υπολογιστής 2 ή server θα ενημερώσει είτε θετικά για την έγκριση της σύνδεσης είτε αρνητικά δηλαδή:

- Θα στείλει πίσω segment τον δικό του αριθμό ακολουθίας το ISN όπου ISN+1, που αυτό σημαίνει θετική απάντηση.
- Ή θα έχει αρνητική απάντηση όπου δεν αποδέχεται την σύνδεση.

Βήμα 3: Αφού λοιπόν ο υπολογιστής 2 έχει αποδεχθεί την σύνδεση, ο υπολογιστής 1 θα λάβει τότε segment επιβεβαίωσης , όπου θα στείλει πίσω με την σειρά άλλο segment επιβεβαίωσης όπου είναι M+1 ο αριθμός της σειράς.

Σημ.1: Ο αριθμός που δηλώνει την αρχή που θα χρησιμοποιηθεί για την σειρά των Bytes ονομάζεται ISN.

Σημ.2: Όπου M είναι ο τελευταίος αριθμός της σειράς που λήφθηκε από τον υπολογιστή 2.



Εικόνα 3.6: Εγκατάσταση Σύνδεσης

2. Μεταφορά Δεδομένων

Αφού λοιπόν έχει γίνει η εγκατάσταση της σύνδεσης και από τις δύο πλευρές, και έχουν ανταλλαχθεί οι αριθμοί ISNs, το επόμενο βήμα είναι η ανταλλαγή πακέτων ανάμεσα στον πελάτη και τον server από τις εφαρμογές. Υπάρχουν 2 τεχνικές για την ομαλή μεταφορά των δεδομένων:

1. Ο Έλεγχος Ροής.
2. Και οι τεχνικές ελέγχου συμφόρησης.

Όπου στον Έλεγχο ροής χρησιμοποιείτε η μεταβλητή που ονομάζεται παράθυρο λήψης που δεσμεύει τον διαθέσιμο χώρο που υπάρχει για τα δεδομένα με βάση τις απαιτήσεις του παραλήπτη που μπορεί να παραλάβει, στην συνέχεια αν ελαττωθεί ο διαθέσιμος χώρος μειώνεται και ο ρυθμός μετάδοσης από τον αποστολέα, αποφεύγοντας έτσι την υπερχειλίση στον παραλήπτη.

Τις Τεχνικές ελέγχου συμφόρησης θα τις αναλύσουμε σε επομένη ενότητα.

3. Τερματισμός Σύνδεσης

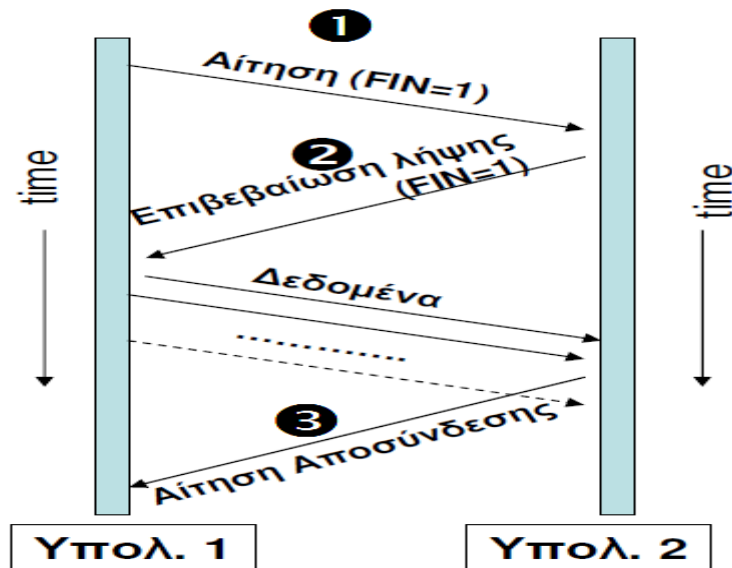
Για τον τερματισμό μίας σύνδεσης υπάρχουν δύο εκδοχές,

1. Τερματισμός από την μία πλευρά του άκρου.

2. Τερματισμός και από τις δύο πλευρές.

Όταν θα τερματισθεί από τη μία πλευρά η κατάσταση λέγεται Half-closed, Δηλαδή ο ένας υπολογιστής θα σταματήσει να στέλνει πακέτα, αλλά θα μπορεί να δέχεται ακόμα. Για να γίνει αυτό πρέπει ο υπολογιστής που θέλει να τερματίσει τη σύνδεση να στείλει πίσω μία σημαία FIN, η οποία υποδηλώνει στο άλλο άκρο ότι θα σταματήσει να στέλνει πακέτα. Στην συνέχεια η άλλη πλευρά θα το επιβεβαιώσει με ACK, και τέλος η πλευρά που ζήτησε τον τερματισμό της σύνδεσης το επιβεβαιώνει και αυτή με την σειρά της στέλνοντας ένα πακέτο ACK.

Και η 2^η εκδοχή να ζητήσουν και οι δύο πλευρές τον τερματισμό της σύνδεσης, στέλνοντας ταυτόχρονα και οι δύο πλευρές 2 σημαίες FIN, όπου το επιβεβαιώνουν γυρίζοντας πίσω με πακέτα ACK. Και τέλος τερματίζουν την σύνδεση (εικόνα 3.7), όπου ούτε μπορούν πλέον να στείλουν δεδομένα ούτε να λάβουν.[11.]



Εικόνα 3.7: Τερματισμός Σύνδεσης

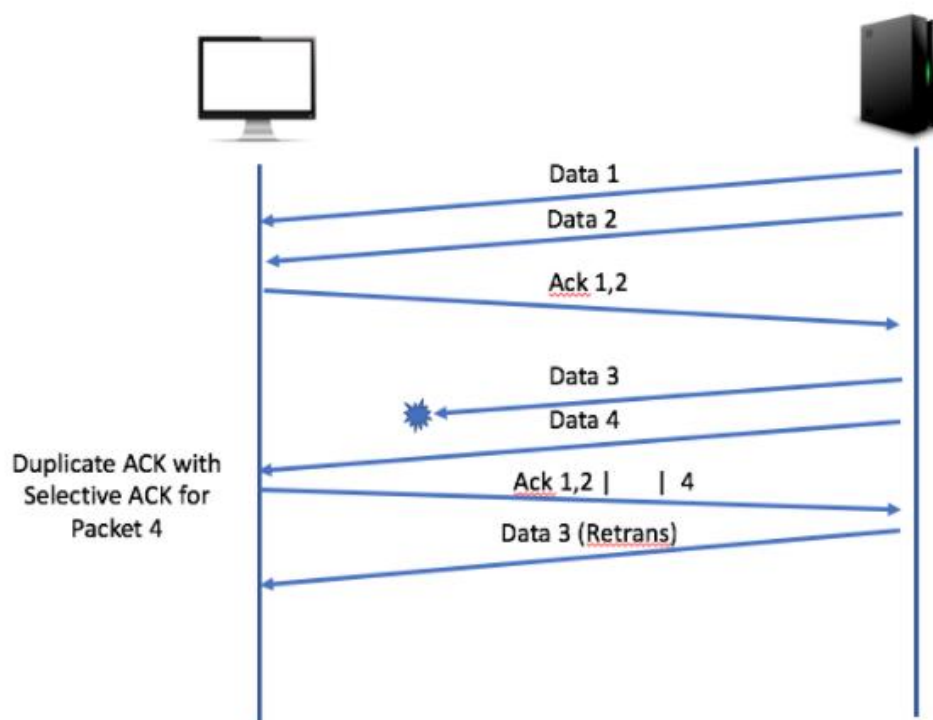
3.7 Μηχανισμός Αξιοπίστης Μεταφοράς Δεδομένων TCP

Όπως έχουμε αναφέρει το TCP στοχεύει στην αξιοπίστη μεταφορά των δεδομένων, και στην συνεχή και ομαλή ροή των δεδομένων από ένα τερματικό σε ένα άλλο. Στο TCP για να επιτευχθεί όλο αυτό χρησιμοποιεί τον μηχανισμό ή την λειτουργία επιβεβαίωση λήψης με αναμετάδοση (acknowledgement). Αυτή η τεχνική έχει την ιδιότητα καθώς ο δέκτης λαμβάνει τα δεδομένα από τον αποστολέα να στέλνει πίσω ένα μήνυμα επιβεβαίωσης λήψης. Και σε περίπτωση που δεν λάβει επιβεβαίωση ο αποστολέας από τον δεκτή θα θέσει

σε εφαρμογή έναν χρονιστή ή χρονόμετρο αντίστροφης μέτρησης (timer) έτσι ώστε να ξαναστείλει το πακέτο.

Επίσης υπάρχει το πρόβλημα με τα διπλά μηνύματα επιβεβαίωσης (duplicate ACKs).

Για κάποιο ενδεχόμενο κατά της διάρκειας της μεταφοράς μπορεί κάποιο η περισσότερα πακέτα να χαθούν και έτσι να υπάρχει κάποια απώλεια. Τα duplicate ACKs αποτελούν μέρος του μηχανισμού του TCP, και χρησιμοποιούνται ως διαδικασία της γρήγορης αναμετάδοσης και της ανάκτησης πακέτων. Όπου στην εικόνα 3.8 βλέπουμε και ένα παράδειγμα της διαδικασίας., όπου συγκεκριμένα έφτασε διπλή επιβεβαίωση για το πακέτο 4 ενώ για το πακέτο 3 καμία επιβεβαίωση, και έτσι αναμεταδίδεται το πακέτο 3. [11.]



Εικόνα 3.8: TCP Duplicate / Selective Acknowledgments

3.8 Μηχανισμός επιβεβαίωσης κυλιόμενου παραθύρου του TCP

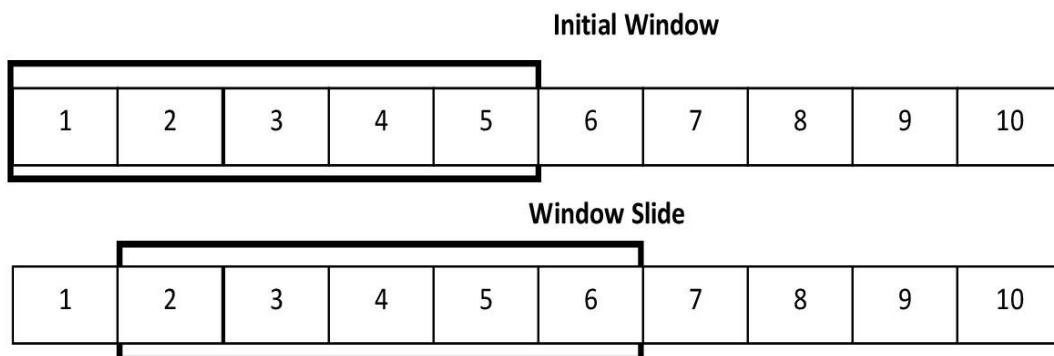
Το παράθυρο του TCP ή μηχανισμός κυλιόμενου παραθύρου, είναι ένας αξιόπιστος μηχανισμός για την μεταφορά των δεδομένων. Η επικοινωνία ανάμεσα σε ένα client-server είναι full-duplex. Χρησιμοποιείται για τον έλεγχο ροής σε ένα δίκτυο από άκρη σε άκρη .

Με τον μηχανισμό του παραθύρου με μία μόνο επιβεβαίωση είναι εφικτό να επιβεβαιωθούν πολλά πακέτα.

Όταν ένας αποστολέας είναι σε θέση να στείλει πακέτα προτού λάβει το μήνυμα επιβεβαίωσης από τον παραλήπτη που έχει λάβει αυτά τα πακέτα, τα πακέτα αυτά που είναι στην ουρά είναι ο αριθμός των μη επιβεβαιωμένων πακέτων. Όπου αν η επιβεβαίωση από τον παραλήπτη δεν φθάσει μέσα σε ένα προκαθορισμένο πλαίσιο τότε αυτά ξαναστέλλονται.

Η Διαδικασία ολίσθησης του παραθύρου είναι ως εξής:

- Η επικεφαλίδα του TCP καθορίζει στον αποστολέα κατά μήκος του παραθύρου μπορεί να στείλει όλα τα πακέτα, και χωρίς να πάρει πίσω επιβεβαίωση ACK, στην συνέχεια και κάθε πακέτο που έχει στείλει μπαίνει σε λειτουργία ένας χρονομετρητής (timeout timers). Με βάση τον αριθμό ακολουθίας ο παραλήπτης πρέπει να επιβεβαιώσει τα πακέτα που έλαβε. Με τον αριθμό όμως του τελευταίου πακέτου. Όταν ο παραλήπτης στέλνει πίσω στον αποστολέα την επιβεβαίωση ACK, το παράθυρο μετακινείται δεξιά.



Εικόνα 3.10: Παράδειγμα κυλιόμενου παραθύρου

- Για παράδειγμα αν μπορεί ο αποστολέας να στείλει πέντε και πάνω πακέτα του TCP. Και χωρίς να λάβει από τον παραλήπτη επιβεβαίωση, στην συνέχεια αφού λάβει για το 1^ο πακέτο επιβεβαίωση θα μετακινηθεί το παράθυρο κατά μία θέση δεξιά, για να σταλθεί και το 6^ο πακέτο στην συνέχεια της μετάδοσης..
- Όμως αν τύχει και χαθεί πακέτο ή και περισσότερα πακέτα κατά την μετάδοση παραλήπτης δεν θα είναι σε θέση να στείλει επιβεβαίωση στον αποστολέα γι' αυτό το πακέτο ή τα πακέτα, τότε θα μπει σε λειτουργία ο χρονομετρητής και θα μπει σε διαδικασία η επαναμετάδοση του πακέτου που χάθηκε/

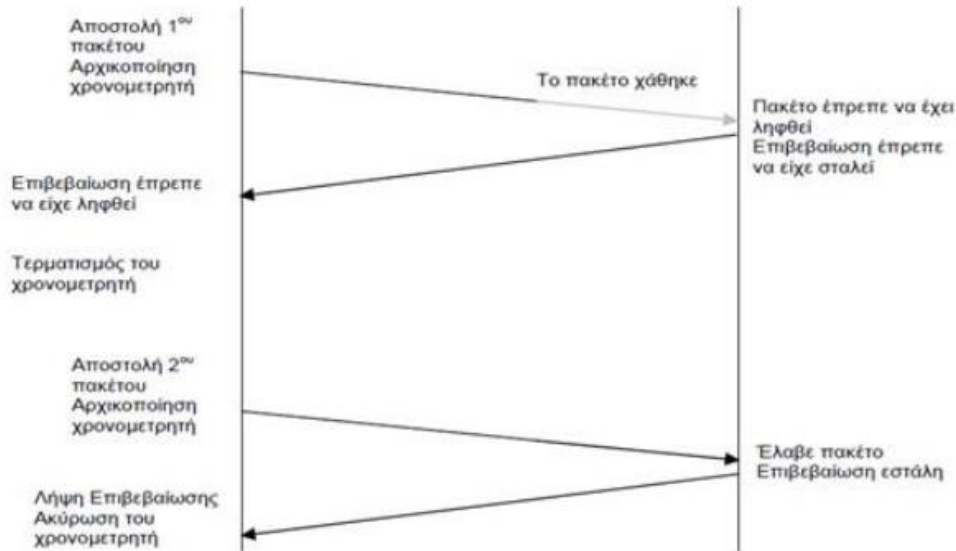
- Όταν ο παραλήπτης λάβει και τα πέντε πακέτα τότε στέλνει στον αποστολέα την επιβεβαίωση ACK, ότι τα έχει λάβει όλα τα πακέτα **π.χ. Εικόνα 3.10**.
- Σε ένα παράθυρο το μήκος του δηλώνεται σε bytes. Και ο παραλήπτης είναι αυτός που θα ορίσει το μέγεθος του.

Πολλές φορές όταν στέλνουμε ένα αρχείο από έναν υπολογιστή σ έναν άλλο ή κατεβάζουμε ένα αρχείο από το Internet, παρατηρούμε κάποιες φορές ότι ο χρόνος αποστολής η λήψης έχει διακυμάνσεις συνήθως μεγάλο χρόνο, οπού στην συνέχεια μπορεί να μειωθεί αισθητά. [12.]. [13.]

3.9 Μηχανισμοί Ελέγχου Κίνησης του TCP

Για την αξιόπιστη λειτουργία του TCP και την επιτυχής μεταφορά των πακέτων, χρησιμοποιεί διάφορους χρονομετρητές κίνησης. Όπως αναφέραμε σε προηγούμενη ενότητα όταν φύγει ένα πακέτο από τον αποστολέα για να φθάσει στον παραλήπτη τίθεται σε εφαρμογή ένας χρονομετρητής και περιμένει την επιβεβαίωση του πακέτου (ACK). Αν ο χρόνος λήξει πριν την επιβεβαίωση του πακέτου ο αποστολέας αναμεταδίδει το πακέτο. Υπάρχουν παράγοντες όμως που παίζουν ρόλο έτσι ώστε π.χ. να μην φθάσει μια επιβεβαίωση πίσω στον αποστολέα η και να καθυστερήσουν να φθάσουν στον τελικό προορισμό, όπως την κυκλοφορία στο διαδίκτυο, απόσταση αποστολέα - παραλήπτη κ.ά.. Και επειδή στο TCP παρέχει την δυνατότητα επικοινωνούν ταυτόχρονα πολλές εφαρμογές με πολλούς προορισμούς, δεν είναι εφικτό να γνωρίζει σε όλο το φάσμα του διαδικτύου, τις καθυστερήσεις με ακρίβεια χρόνου. Αυτό που δίνεται να κάνει είναι να εκτιμά την καθυστέρηση μετάβασης και επιστροφής (RTT Round – Trip Delay), για την ενεργές συνδέσεις

Από την στιγμή που θα σταλθεί ένα πακέτο και πάρει επιβεβαίωση χρησιμοποιείτε μια μαθηματική στατιστική συνάρτηση που εκτιμά τον χρόνο μετάβασης και επιστροφής και προκύπτει ένας σταθμισμένος μέσος όρος αυτό βοηθάει αν η καθυστέρηση είναι μικρή επανέρχεται το χρονόμετρο σε κανονική τιμή.

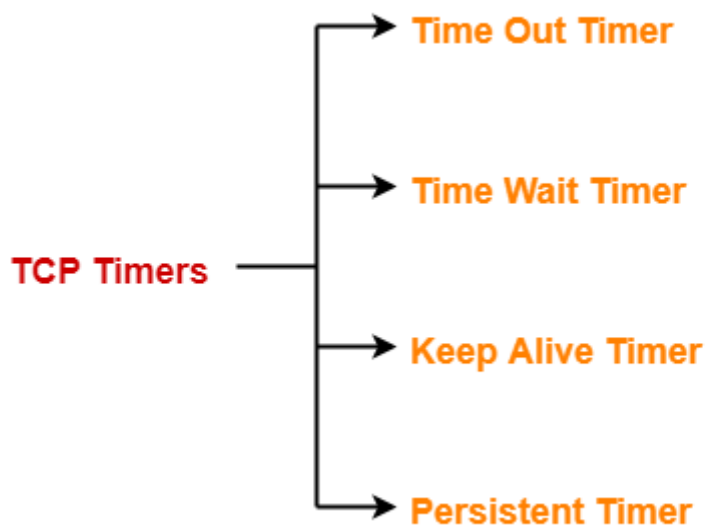


Εικόνα 3.11: Λήξη χρονομετρητή και Επαναμετάδοση

- Στην εικόνα 3.11, πριν την επιβεβαίωση του πακέτου ο χρόνος έχει λήξει, και έτσι επαναμεταδίδεται.

3.10 Χρονιστές Ελέγχου Κίνησης του TCP

Το TCP χρησιμοποιεί συγκεκριμένους μηχανισμούς για τον έλεγχο της κίνησης των συνδέσεων στο διαδίκτυο ή για να αποτρέπουν κάποιες αδιέξοδες καταστάσεις π.χ. μεγάλη κίνηση στο Διαδίκτυο έτσι ώστε να αποτρέπουν την κατάρρευση του. Αυτοί οι μηχανισμοί ονομάζονται χρονόμετρα ελέγχου ή χρονιστές.. Ποιο συγκεκριμένα είναι τέσσερις (εικόνα 3.12) . Ο κυριότερος χρονιστής είναι ο χρονιστής επαναποστολής πακέτων (retransmission timer)



Εικόνα 3.12: Χρονιστές - Timers

Αναλυτικότερα:

1) Χρονιστής επαναποστολής (retransmission timer)

Ίσως ο σημαντικότερος χρονιστής στο TCP. Όταν αποστολέας καθυστερήσει να λάβει μια επιβεβαίωση ACK ότι έφτασε το πακέτο επιτυχώς στον δέκτη τότε μπαίνει σε λειτουργία ο χρονιστής επαναποστολή (το χρονόμετρο είναι ήδη δε λειτουργία αυτομάτως με την αποστολή του πακέτου), έτσι στην συνέχεια ο αποστολέας να ξαναστείλει το πακέτο.

Επίσης αν έχει λάβει απάντηση ότι έχει φτάσει το πακέτο επιτυχώς, το χρονόμετρο που είναι σε λειτουργία από την αποστολή σταματά την μέτρηση.

Ο Χρονιστής επαναποστολής χρησιμοποιεί δυναμικές μετρήσεις χρόνου για κάθε σύνδεση, επειδή για τον λόγο όλες οι συνδέσεις δεν είναι σταθερού χρόνου προφανώς και παίζουν ρόλο και άλλο παράγοντες, όπως το μήκος των διάδρομων και ο χρόνος επιστροφής (RTT).

Οι μηχανισμοί ελέγχου βασίζονται σε μετρήσεις χρόνων και ιδιαίτερα της μετ' επιστροφής διάρκειας ταξιδιού των πακέτων (RTT-Round trip times), προσμετράτε ως εκτίμηση διακύμανσης. Και υπάρχει στο Standard πρωτόκολλο του TCP.

Ο τύπος για την διακύμανσή του RTT είναι:

$$\text{SRTT}(K+1) = \alpha * \text{SRTT}(K) + (1-\alpha) * \text{RTT}(K+1)$$

Η παραπάνω εξίσωση, σε συνδυασμό με τις προηγούμενες τιμές που έχουν υπολογιστεί από τον αποστολέα, καθορίζει τον εκτιμώμενο χρόνο επιστροφής για το επόμενο πακέτο που θα αποσταλεί

Χρονικό όριο επαναποστολής: (Retransmission Timeout RTO): Ο χρόνος του διαμορφώνεται από τον χρόνο επιστροφής και την τυχόν απόκλιση του. Ο Συνηθέστερος τύπος για τον υπολογισμό του είναι : $RTO = RTTs + 4 * RTTd$.

2) Χρονιστής Εμμονής (persistence timer)

Σ αυτόν τον χρονιστή, όταν ένας παραλήπτης έχει παραλάβει ένα παράθυρο μεγέθους μηδέν, μπαίνει σε λειτουργία τότε το χρονόμετρο επίμονης. Και θα είναι σε λειτουργία μέχρι ο αποστολέας λάβει μία επιβεβαίωση ACK από τον παραλήπτη, ότι το παράθυρο δεν είναι πλέον μηδενικό.

3) Χρονιστής αναμονής χρόνου (Time Wait Timer): Αυτός ο Χρονιστής μπαίνει σε λειτουργία κατά το κλείσιμο μιας TCP σύνδεσης.

Σ αυτήν την περίπτωση όταν ο αποστολέας έχει λάβει επιβεβαίωση για το τελευταίο ACK, και γίνεται αποστολή της σημαίας FIN για το κλείσιμο της σύνδεσης, μπαίνει σε λειτουργία

ο Χρονιστής αναμονής ο λόγος ότι ακόμα και στο κλείσιμο μιας TCP σύνδεσης, υπάρχει πιθανότητα να υπάρχουν ακόμα υπολείμματα από πακέτα τα οποία επιχειρούν να έχουν πρόσβαση στις κλειστές θύρες των άκρων, έτσι ο Χρονιστής δεν επιτρέπει αυτή την διαρροή των πακέτων

4) Χρονιστής «επιβίωσης» και Χρονιστής αδράνειας (keep-alive and idle timer):

Η δουλειά αυτού του χρονιστή είναι να διατηρεί τις μεγάλες συνδέσεις του TCP σε αδράνεια όταν για παράδειγμα δεν γίνεται κάποια ενέργεια από κάποιο άκρο.π.χ.) αν έχουμε ανοιχτή σε κάποιον browser μία σελίδα,και αποφασίζουμε να την αφήσουμε ανοιχτή χωρίς καμία ενέργεια, τότε ο Χρονιστής ελέγχου επιβίωσης όταν εκπνέει ο χρόνος του, αποστέλλεται ένα άδειο πακέτο, και μπαίνει σε λειτουργία ο Χρονιστής αδράνειας, και έχει πάρει κάποια απάντηση μέχρι να τελειώσει ο χρόνος του, η σύνδεση θα παραμείνει ανοιχτή. Αν όχι τότε τερματίζεται, όπου πολλές φορές έχουμε παρατηρήσει όπως αναφέραμε αν δεν κάνουμε κάποια ενέργεια, στην σελίδα και μετρά από πολύ ώρα αποφασίσουμε να πατούμε καπό πάνω στην σελίδα θα διαπιστώσουμε ότι αυτή θα ανανεωθεί πάλι. [13.]

4. TCP ΑΛΓΟΡΙΘΜΟΙ ΕΛΕΓΧΟΥ ΣΥΜΦΟΡΗΣΗΣ

Όπως έχουμε αναφέρει και παραπάνω, το TCP πρωτόκολλο ελέγχου μετάδοσης παρέχει αξιοπιστία και παράδοση από άκρο σε άκρο των πακέτων που μεταφέρονται στο δίκτυο. Επίσης το TCP έχει σχεδιαστεί για να χειριστεί και το πρόβλημα κατάρρευσης του δικτύου. Ήδη από τα μέσα της δεκαετίας του '90 και μετέπειτα, διακινείτε μεγάλος όγκος δεδομένων στο Διαδίκτυο, , με αποτέλεσμα να υπόκεινται σε προβλήματα συμφόρησης. Τον Οκτώβριο του '86, το Διαδίκτυο είχε το πρώτο από αυτά έγινε μια σειρά από κάποιες συμφορήσεις στα δικτυα

Συμφόρηση (Congestion) ονομάζεται η κατάσταση ενός δικτύου ή τμήματος δικτύου όταν δεν μπορεί να διαβιβάσει όλη την κίνηση που δέχεται, δηλ. η εισερχόμενη σε κάποιο υπομήμα του δικτύου κίνηση είναι λιγότερη από αυτή που εξέρχεται συν την αύξηση της αποθήκευσης στο τμήμα αυτό σε κάποιο χρονικό διάστημα παρατήρησης. Ο έλεγχος της συμφόρησης του Δικτύου, είναι ένα από τα κύρια χαρακτηριστικά του TCP.

Ο τρόπος που πραγματοποιεί έλεγχο συμφόρησης (congestion control) το TCP είναι από τα πιο χαρακτηριστικά και επιτυχή στοιχεία του πρωτοκόλλου. Είναι ένας μηχανισμός προσαρμοσμένος πλήρως στη φιλοσοφία σχεδίασης του διαδικτύου όπου τα υποκείμενα φυσικά δίκτυα είναι ανομοιογενή και αντιμετωπίζονται σαν αναξιόπιστα μέσω του πρωτοκόλλου IP του στρώματος δικτύου. Η κεντρική ιδέα είναι να χρησιμοποιεί ο αλγόριθμος του TCP την άφιξη κάθε επιβεβαίωσης (ACKs) σαν ένδειξη ότι το πακέτο διαβιβάστηκε επιτυχώς και άρα μπορεί να στείλει ένα νέο με τη βεβαιότητα ότι δεν προκαλεί συμφόρηση. Το TCP επιχειρεί περαιτέρω να εκτιμήσει από το ρυθμό των αφίξεων επιβεβαιώσεων τη διαπερατότητα (throughput) που μπορεί να υποστηρίξει το δίκτυο, ούτως ώστε να εισάγει τα πακέτα μέχρι το ρυθμό που δεν προκαλεί συμφόρηση. Η προσπάθεια των αλγορίθμων ελέγχου συμφόρησης του TCP είναι να φέρουν την κάθε ροή και χρήστη να μοιράζεται δίκαια, δηλαδή ίσα τη διαθέσιμη χωρητικότητα του δικτύου. Η παράμετρος η οποία αποτελεί το εργαλείο ποσοτικοποίησης των ενεργειών των αλγορίθμων αποφυγής συμφόρησης δεν είναι άλλη από το παράθυρο ολίσθησης. Προς αποφυγή συγχύσεως πρέπει να τονισθεί ότι ενώ για τον έλεγχο ροής ο δέκτης στέλνει μια αναγγελία παραθύρου, ταυτόχρονα οι αλγόριθμοι υπολογίζουν ένα συνιστάμενο παράθυρο αποφυγής συμφόρησης.

4.1 Έλεγχος συμφόρησης TCP με αλγόριθμο σταδιακής αύξησης και πολλαπλής μείωσης (AIMD Additive increase/multiplicative decrease)

Στον αλγόριθμο AIMD, περιλαμβάνονται οι τρεις στάνταρ αλγόριθμοι του TCP για τον έλεγχο της συμφόρησης

1. Αλγόριθμος ‘αργής εκκίνησης’ (Slow start)
2. Αλγόριθμος ‘ αποφυγής συμφόρησης’ (Congestion avoidance)
3. Αλγόριθμος ‘ταχείας επανεκπομπής’ (Fast retransmit)

Ο μηχανισμός ελέγχου συμφόρησης TCP παρακολουθεί σε κάθε πλευρά της σύνδεσης δύο πρόσθετες μεταβλητές:

1. Τα παράθυρα συμφόρησης (*CongWin*)
2. Και το *threshold*

Το παράθυρο συμφόρησης CongWin όπου συνήθως θα το δούμε γραμμένο (*cwnd*) είναι μια μεταβλητή του TCP, που έχει την δυνατότητα να περιορίζει τον όγκο των δεδομένων που μπορεί να στείλει ο TCP στο δίκτυο πριν λάβει ένα μήνυμα ACK. Γενικά ο TCP έχει και από την πλευρά του σταθμού, αλλά και από την πλευρά του δέκτη ή παραλήπτη δύο ζώνες που ονομάζονται Buffer Zone, Όταν τα πακέτα δεδομένων φτάσουν στον παραλήπτη πολλές φορές λόγω του όγκου των δεδομένων ο προορισμός “γεμίζει», έτσι τα πακέτα τότε αποθηκεύονται προσωρινά στις buffer zone, τόσο από την μεριά του δεκτή όσο και του παραλήπτη, καθώς μένουν σε αναμονή, ώστε να προωθηθούν στα επόμενα επίπεδα. Όλο αυτό μπορεί να οδηγήσει σε συμφόρηση του Δικτύου και να προκαλέσει απώλεια πακέτων, α, μειωμένη απόδοση δεδομένων, υποβάθμιση της απόδοσης, ακόμη και κατάρρευση του δικτύου σε κάποιες ακραίες περιπτώσεις.

Ενώ το παράθυρο RcvWin είναι το παράθυρο που δημιουργοποιεί ο δέκτης ή ο παραλήπτης.

Σε κάθε περίπτωση το πλήθος των ανεπιβεβαίωτων πακέτων δεν μπορεί να υπερβαίνει το ελάχιστο όριο των παραμέτρων CongWin και RcvWin.

Δηλαδή: $\text{LastByteSent} - \text{LastByteAcked} \leq \min\{\text{CongWin}, \text{RcvWin}\}$.

Παρακάτω θα δούμε ένα παράδειγμα πακέτων στο TCP, θα δούμε πώς το παράθυρο συμφόρησης (CongWin) συμπεριφέρεται καθ' όλη την διάρκεια μια TCP σύνδεσης, με τους τρεις στάνταρ αλγορίθμους.

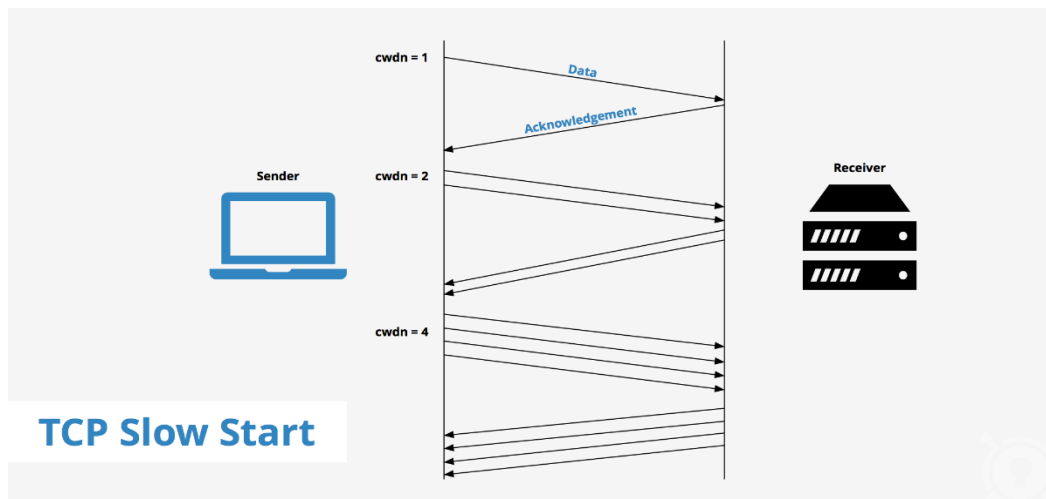
Φάση 1: Ο Αλγόριθμος Slow Start (εικόνα 4.1), είναι ένας αλγόριθμος που μπορεί και εξισορροπεί την ταχύτητα μιας σύνδεσης δικτύου. Ο TCP Slow Start αυξάνει σταδιακά τον όγκο των δεδομένων που μεταδίδονται μέχρι να βρεθεί η να υπάρξει η μέγιστη χωρητικότητα του δικτύου.

Ο TCP Slow Start εμποδίζει τη συμφόρηση ενός δικτύου ρυθμίζοντας τον όγκο των δεδομένων που αποστέλλονται μέσω αυτού. Κατά κάποιο τρόπο ελέγχει τη σύνδεση μεταξύ ενός αποστολέα και παραλήπτη καθορίζοντας την ποσότητα δεδομένων που μπορεί να μεταδοθεί με κάθε πακέτο, και αυξάνει αργά την ποσότητα δεδομένων έως ότου επιτευχθεί η χωρητικότητα του δικτύου. Αυτό εξασφαλίζει όσο το δυνατόν περισσότερα δεδομένα που να μπορούν να μεταδοθούν χωρίς να μπλοκάρει ή να φράξει το δίκτυο.

Η αργή εκκίνηση -Slow Start, σε αντίθεση με αυτό που υπονοεί το όνομά της, επιβάλλει κατά την αρχική εκκίνηση της σύνδεσης να αυξάνει εκθετικά και όχι γραμμικά (δηλ. προσθετικά).

Δηλαδή το παράθυρο συμφόρησης όταν το πρώτο πακέτο σταλθεί και αναγνωριστεί το πρώτο τμήμα του πακέτου πριν εκπνεύσει ο χρόνος του χρονομέτρου, τότε ο αποστολέας αυξάνει το παράθυρο συμφόρησης στο διπλάσιο από ότι ήταν αρχικά. Και αυτό συνεχίζεται κάθε φορά που φτάνει μια επιβεβαίωση για τα προηγούμενα τμήματα δηλαδή θα αυξάνεται το παράθυρο στο διπλάσιο σε σχέση με αυτό που ήταν την τελευταία φορά. Αυτή η διαδικασία συνεχίζεται για όσο διάστημα το παράθυρο συμφόρησης είναι κάτω από το threshold και οι επιβεβαιώσεις φθάνουν πριν από την εκπνοή των χρονικών ορίων που έχουν οι timers.

Μπορούμε να δούμε ότι κατά τη διάρκεια αυτής της φάσης το μέγεθος του παραθύρου συμφόρησης αυξάνεται εκθετικά, δηλαδή το παράθυρο συμφόρησης αρχικοποιείται σε ένα πακέτο. Μετά από έναν χρόνο μετ' επιστροφής (RTT) το μέγεθος του παραθύρου αυξάνεται σε δύο πακέτα. Μετά από δύο RTT το παράθυρο αυξάνεται σε τέσσερα πακέτα. Μετά από τρεις RTT το παράθυρο αυξάνεται σε οκτώ πακέτα. Και ούτω καθεξής. Αυτός ο μηχανισμός αναφέρεται ως αργή εκκίνηση. Η αργή εκκίνηση πήρε αυτό το όνομα για να δηλώσει αυτή τη σταδιακή αύξηση έστω και εκθετική Αρχίζει με 1MSS ξεκινάει αργά, αλλά στην συνέχεια επιταχύνει γρήγορα



Εικόνα 4.1: Αλγόριθμος «Αργής Εκκίνησης – Slow Start TCP»

Φάση 2: Αλγόριθμος ‘αποφυγής συμφόρησης’ Congestion avoidance

Όταν το μέγεθος του παραθύρου θα είναι μεγαλύτερο από την τιμή του threshold, τότε τελειώνει και η φάση της “αργής κίνησης”. Και μεταβαίνουμε στον “έλεγχο αποφυγής συμφόρησης”. Όμως δεν μπορούμε να προβλέψουμε ακριβώς την έναρξη της συμφόρησης ώστε να αποτρέψουμε την απώλεια των πακέτων.

Η ανίχνευση απώλειας πακέτου είναι το γεγονός το οποίο προσδιορίζει την συμφόρηση. Ο αλγόριθμος Congestion Avoidance είναι ένας τρόπος να εξεταστούν τα χαμένα πακέτα. Τυπικά απώλεια πακέτων που προκαλείται από τη βλάβη είναι πολύ μικρή (πολύ λιγότερο από 1 %), επομένως η απώλεια ενός πακέτου γίνεται από συμφόρηση κάπου στο δίκτυο μεταξύ της πηγής και του προορισμού.

Υπάρχουν δύο ενδεικτικές απώλειες πακέτων:

1. Η λήψη των διπλότυπων αναγνωρίσεων, ACKs.
2. Η λήξη χρόνου

Όταν κάπου στο δίκτυο έχουμε μία ένδειξη συμφόρησης οι αλγόριθμοι Congestion avoidance και slow start, υλοποιούνται μαζί. Παρόλο που είναι ανεξάρτητοι μεταξύ τους, για κάθε σύνδεση οι αλγόριθμοι διατηρούν 2 μεταβλητές.

- 1) Το παράθυρο συμφόρησης (cwnd)
- 2) Και το κατώφλι αργής εκκίνησης (ssthresh – slow start threshold)

Κατά την αποφυγή συμφόρησης, το cwnd αυξάνεται κατά περίπου 1 πλήρες μεγέθους τμήματος ανά χρόνο μετ’ επιστροφής (RTT). Σε αυτή τη φάση μετά από κάθε RTT το μέγεθος του παραθύρου συμφόρησης αυξάνεται εκθετικά, κάθε φορά που δέχεται μια επιβεβαίωση (ACK). Αυτή ή αύξηση αντιστοιχεί σε εκθετική μεταβολή, και ποιο συγκεκριμένα:

- Αρχικά το παράθυρο συμφόρησης (cwnd) είναι = 1
- Μετά από 1 RTT, $cwnd = 2^{(1)} = 2$
- 2 RTT, $cwnd = 2^{(2)} = 4$
- 3 RTT, $cwnd = 2^{(3)} = 8$

Στην Συνέχεια έχουμε την φάση: Γραμμική μεταβολή ή Προσθετική προσαύξηση. Το μέγεθος του cwnd (παράθυρο συμφόρησης) αυξάνει το πρόσθετο. Μετά από κάθε RTT $cwnd = cwnd + 1$. Η αύξηση θα πρέπει να είναι το πολύ ένα πακέτο κάθε φορά

- Αρχικά $cwnd = i$
- Μετά από 1 RTT, $cwnd = i+1$
- 2 RTT, $cwnd = i+2$

➤ 3 RTT, cwnd = i+3

Φάση 3: Αλγόριθμος ‘ταχείας επανεκπομπής’ (Fast retransmit)/ Fast Recovery

Το TCP μπορεί να στείλει ένα διπλό αντίγραφο ACK όταν φτάσει ένα πακέτο εκτός σειράς. Ο σκοπός αυτού του ACK είναι να ενημερώσει τον αποστολέα ότι κάποιο πακέτο έχει ληφθεί σε λάθος σειρά και να του υποδείξει τον σειριακό αριθμό του πακέτου που περιμένει. Από την πλευρά του αποστολέα, τα διπλά ACK μπορεί να προκληθούν από μια σειρά προβλημάτων του δικτύου.

Όπως:

1. μπορεί να προκληθούν από μειωμένα πακέτα
2. Τα διπλά ACK μπορεί να προκληθούν από την αναδιάταξη των πακέτων δεδομένων του Δικτύου

Ο αποστολέας TCP πρέπει να χρησιμοποιήσει τον αλγόριθμο «γρήγορης αναμετάδοσης» για τον εντοπισμό και την επιδιόρθωση των απωλειών, με βάση τα εισερχόμενα διπλότυπα ACK. Μετά από τη λήψη τριών συνεχόμενων DupAcks θεωρούμε πιο πιθανό το πακέτο να έχει χαθεί.

Ο TCP πραγματοποιεί αναμετάδοση αυτού που φαίνεται να λείπει χωρίς να περιμένει να λήξει ο χρονοδιακόπτης επανεκπομπής.

Στον Αλγόριθμο γρήγορης ανάκαμψης, μετά την επαναμετάδοση του πακέτου, μπαίνει σε εφαρμογή ο μηχανισμός αποφυγής συμφόρησης, και όχι ο αλγόριθμος slow start. Όλο αυτό επιτρέπει ένα υψηλό βαθμό στην ρυθμοαπόδοση του TCP.

Ο λόγος για τον οποίο δεν εκτελείται αργή εκκίνηση σε αυτήν την περίπτωση είναι ότι η λήψη των διπλών ACK ενημερώνει τον TCP επιπλέον από ένα πακέτο που χάθηκε ο αλγόριθμος «γρήγορης ανάκτησης» διέπει τη μετάδοση νέων δεδομένων έως ότου φτάσει ένα μη διπλό ACK. Ο λόγος για την απρόβλεπτη εκκίνηση είναι ότι η λήψη των διπλών ACK όχι μόνο υποδηλώνει ότι έχει χαθεί ένα τμήμα, αλλά επίσης ότι τα τμήματα πιθανότατα εγκαταλείπουν το δίκτυο.

Οι αλγόριθμοι γρήγορης επανεκπομπής και γρήγορης ανάκτησης εφαρμόζονται μαζί ως εξής.

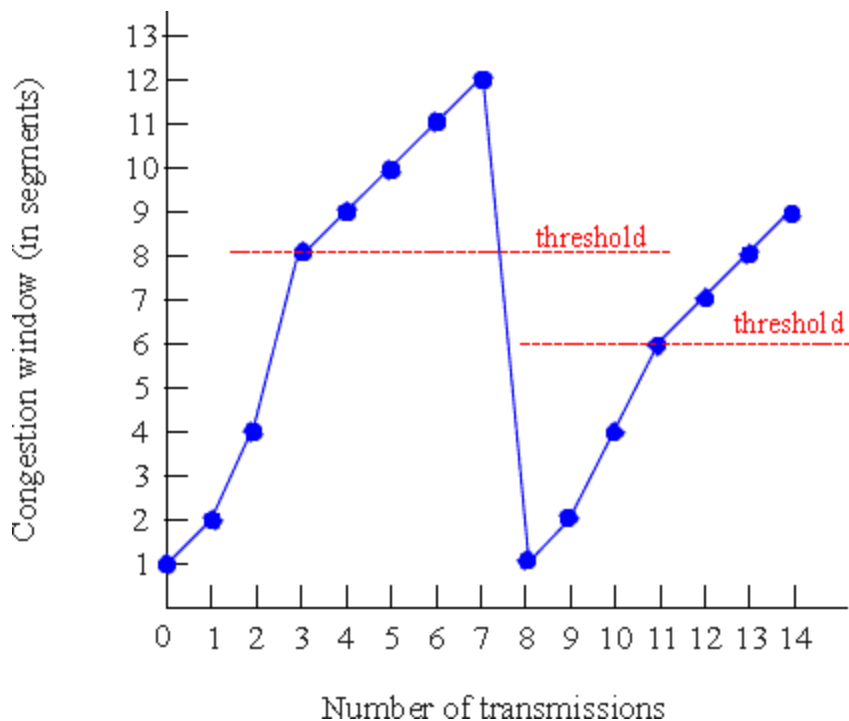
1. Στο πρώτο και δεύτερο διπλό ACK που ελήφθησαν σε έναν αποστολέα,

Το TCP πρέπει I να στείλει ένα τμήμα των προηγουμένως μη αποσταλμένων δεδομένων.

2. Όταν ληφθεί το τρίτο ACK θέτουμε το κατώτατο κατώφλι $ssthresh$, στο μισό της ελάχιστης τιμής του τρέχοντος παραθύρου συμφόρησης ($cwnd$) και του παραθύρου ($advertised\ window$).
3. Το χαμένο τμήμα που ξεκινά πρέπει να αναμεταδοθεί και το $cwnd$ ορίστηκε σε $ssthresh$ συν $3 * SMSS$. Αυτό τεχνητά «φουσκώνει» το παράθυρο συμφόρησης κατά τον αριθμό των τμημάτων (τρία) που έχουν αποχωρήσει από το δίκτυο και στο οποίο ο δέκτης έχει αποθηκευτεί.

Παράδειγμα.

Στην εικόνα 4.2 απεικονίζονται οι μεταβολές του παραθύρου συμφόρησης (*CongWin*) του TCP. Σε αυτό το σχήμα, η τιμή του *threshold* τίθεται αρχικά ίση με 8 (MSS).



Εικόνα 4.2: Η μεταβολή του TCP παραθύρου συμφόρησης

- Το παράθυρο συμφόρησης ανεβαίνει με εκθετικό τρόπο με την διαδικασία της 'Slow start' και φτάνει στο όριο στην Τρίτη μετάδοση.

- Το παράθυρο συμφόρησης ανεβαίνει συνέχεια γραμμικά μέχρι να εμφανισθεί απώλεια segment, η οποία γίνεται αμέσως μετά την 7^η εκπομπή.
- Παρατηρείστε ότι όταν παρουσιάζεται απώλεια πακέτου, στην 7^η εκπομπή τότε το παράθυρο συμφόρησης (*CongWin*) είναι 12 (MSS).
- Αμέσως μετά (στην 8^η εκπομπή) το *threshold* παίρνει την τιμή $\text{threshold} = \text{CongWin} / 2 = 6$ (MSS) και το παράθυρο συμφόρησης, ορίζεται ίσο με $\text{CongWin} = 1$. [14.].[15.].[16.]

5 TCP VARIANTS

5.1 Εισαγωγή

Όπως αναφέραμε σε προηγούμενο κεφάλαιο το μεγαλύτερο μέρος της κυκλοφορίας σήμερα στο Διαδίκτυο ελέγχεται από το πρωτόκολλο μεταφοράς (TCP-Transmission Control Protocol). Ως εκ τούτου, η απόδοση του TCP ασκεί σημαντική επίδραση στην συνολική απόδοση του Διαδικτύου. Έτσι η κυκλοφορία στο Web Internet αποτελεί την πλειοψηφία των TCP υλοποιήσεων στους σημερινούς Web εξυπηρετητές., το TCP είναι ένα σύνθετο πρωτόκολλο με πολλές διαμορφώσιμες από τον χρήστη παραμέτρους και μια σειρά από διαφορετικές εφαρμογές.. [17.]

Σαν τελική θεώρηση, η σταθερότητα και η δικαιοσύνη του τρέχοντος διαδικτύου στηρίζονται στην χρήση των μηχανισμών ελέγχου συμφόρησης. Επομένως είναι σημαντικό να εξεταστούν οι εφαρμογές TCP για την συμμόρφωση τους, στον έλεγχο συμφόρησης από το ένα άκρο στο άλλο.

Το TCP είναι το αξιόπιστο πρωτόκολλο προσανατολισμένο στη σύνδεση που παρέχει αξιόπιστη μεταφορά δεδομένων μεταξύ των κόμβων. Διασφαλίζει ότι τα δεδομένα φθάνουν σωστά στον προορισμό χωρίς απώλεια ή ζημιά. Τα δεδομένα μεταδίδονται με τη μορφή συνεχούς ροής οκτάδων. Η αξιόπιστη μεταφορά οκτάδων (bit) επιτυγχάνεται με τη χρήση ενός αριθμού ακολουθίας σε κάθε οκτάδα. Μια άλλη πτυχή του TCP είναι ο μηχανισμός τριπλής χειραψίας για τη δημιουργία σύνδεσης μεταξύ των κόμβων [8]. Επιπλέον, το TCP χρησιμοποιεί την εκχώρηση θύρας (Port) ως μηχανισμό διευθύνσεων για τη διαφοροποίηση κάθε σύνδεσης για τις περιπτώσεις που απαιτούνται περισσότερες συνδέσεις TCP μεταξύ κόμβων. Μετά την εισαγωγή της πρώτης έκδοσης του TCP υπάρχουν πολλές διαφορετικές παραλλαγές TCP.

Οι σύγχρονες υλοποιήσεις TCP περιέχουν έναν αριθμό αλγορίθμων που στοχεύουν στον έλεγχο της συμφόρησης του δικτύου, διατηρώντας ταυτόχρονα καλή απόδοση του χρήστη. Οι πρώτες υλοποιήσεις TCP ακολούθησαν ένα μοντέλο back-back χρησιμοποιώντας αθροιστική θετική αναγνώριση και απαιτούσε τη λήξη του χρονομετρητή εκ νέου αποστολής για την εκ νέου αποστολή δεδομένων που χάθηκαν κατά τη μεταφορά. Έτσι, οι σύγχρονες εφαρμογές TCP οδηγούν στην ελαχιστοποίηση της συμφόρησης δικτύου. Παρακάτω θα αναλύσουμε και θα εξετάσουμε τέσσερις παραλλαγές του TCP. Συγκεκριμένα τους αλγορίθμους: 1) TCP Reno, 2) TCP New Reno, 3) TCP Vegas και 4) TCP Sack.

5.2 Αλγόριθμος TCP Reno

Η παραλλαγή TCP Reno, Παρουσιάστηκε το 1990 από τον Van Jacobson.

Στον TCP Reno εφαρμόζονται οι αλγόριθμοι Slow Start, Congestion Avoidance, Fast Retransmit, Fast recovery. Ο Μηχανισμός TCP Reno, υλοποιείται σήμερα από τα περισσότερα λειτουργικά συστήματα.

Ο TCP Reno, αυξάνει το congestion window (cwnd), για κάθε ACK που δέχεται από ένα segment ανά RTT, και μειώνει στο μισό το παράθυρο συμφόρησης για κάθε απώλεια ανά RTT

Δηλαδή: Increase: $W=W + (1 / W)$

Decrease: $W=W - ((1 / 2)W)$

Ο TCP Reno, σε γενικές γραμμές επεκτείνει την διαδικασία γρήγορης επαναμετάδοσης χρησιμοποιώντας την γρήγορη ανάκαμψη (fast recovery). Στο TCP Reno όταν ληφθούν τρία διπλά πακέτα ACKs, είναι σημάδι συμφόρησης, και χρησιμοποιείται η γρήγορη αναμετάδοση. Τότε η πηγή ξαναστέλνει αμέσως το τμήμα χωρίς να περιμένει την εκπνοή του χρονιστή. Θέτει το Conthresh στο μισό του cwnd Σε επόμενη φάση η πηγή μπαίνει σε fast recovery δηλαδή ρίχνει το cwnd στο Conthresh+3 φορές το μέγεθος τμήματος, αλλά δεν θα εκτελέσει την αργή κίνηση, εκτελεί την αποφυγή συμφόρησης, δηλαδή θα συνεχίζει να αυξάνει το cwnd από την τιμή που βρίσκεται κατά $1/cwnd$ σε κάθε ACK.

Παρακάτω βλέπουμε κάποιες περιπτώσεις συμφόρησης του Reno:

1. Όπου το παράθυρο συμφόρησης είναι μικρότερο από το όριο αργής εκκίνησης ssthresh, το μέγεθος του παραθύρου αυξάνεται κατά ένα, όπως
if (cwnd < ssthresh) (1)
{
cwnd = cwnd + 1;
}
2. Στην φάση της αποφυγής της σύμφορης ενημερώνει ο αποστολέας το παράθυρο συμφόρησης cwnd το οποίο το νέο μέγεθος του θα είναι με βάση την εξίσωση : $cwnd = cwnd + (f / cwnd)$
3. Από την άλλη πλευρά, όταν το TCP RENO ανιχνεύει την απώλεια πακέτων με τον γρήγορο αλγόριθμο αναμετάδοσης τότε το ssthresh θα είναι ως:
 $Ssthresh = cwnd/2$

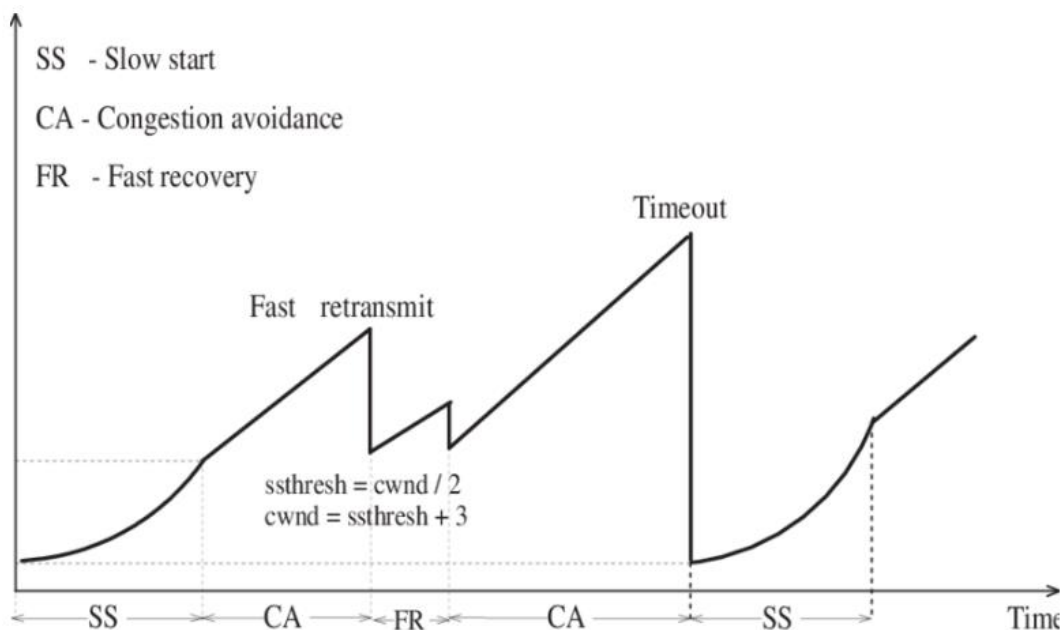
Στο TCP Reno υπάρχουν και κάποια μειονεκτήματα όπως έχει τον περιορισμό που μπορεί να ανιχνεύσει μόνο μία απώλεια ενός πακέτου, αφού έχει δημιουργηθεί η σύνδεση της αργής εκκίνησης, όταν ο δέκτης θα στείλει ένα αναγνωρισμένο τμήμα ACK, τότε ο αποστολέας θα ορίζει το παράθυρο cwnd σε 2 τμήματα, και στέλνει 2 πακέτα, Όταν ο δέκτης στέλνει δύο αναγνωρισμένα ACK στον αποστολέα, ο αποστολέας ορίζει το παράθυρο cwnd σε τέσσερα και στέλνει τέσσερα πακέτα. Και όλο αυτό καθιστά στο TCP, “διπλή συμφόρηση”, σε κάθε πακέτο που σταλθεί ανά RTT.

Όταν το εύρος ζώνης του δικτύου φτάσει στο μέγιστο, τότε το μέγεθος του παραθύρου συμφόρησης προσεγγίζει την χωρητικότητα του δικτύου.

Η αργή εκκίνηση αποσκοπεί στην αποφυγή της συμφόρησής, που προκαλείται στον μεγάλο όγκο δεδομένων κατά την διάρκεια της σύνδεσης. Όταν δηλαδή ο αποστολέας θα στείλει ένα μεγάλο αριθμό πακέτων, με το παράθυρο cwnd να αυξάνεται εκθετικά, Και τότε ο αριθμός των χαμένων πακέτων θα είναι πολύ μεγάλος, όταν θα συμβεί η συμφόρηση.

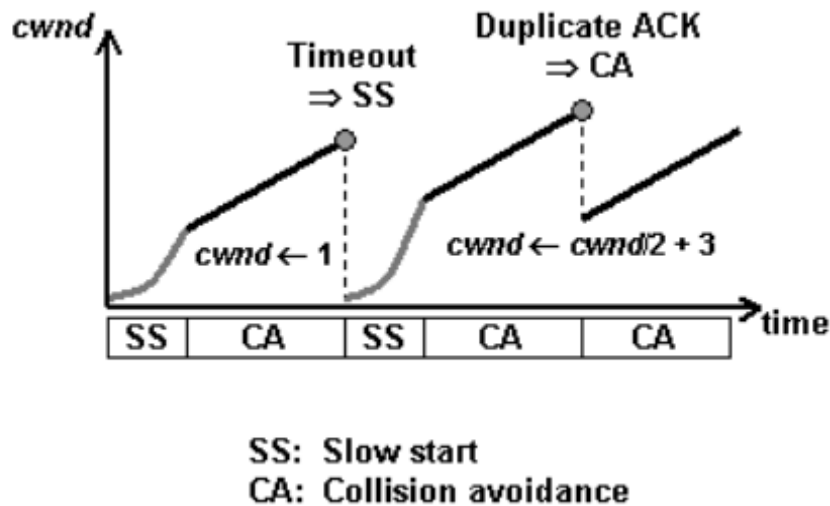
Γι’ αυτό όπως αναφέραμε και παραπάνω, στην ταχεία επανεκπομπής και ταχείας ανάκαμψης, στα τριπλά ACK, μας δίνουν για το δίκτυο περισσότερες πληροφορίες, για την κατάσταση του ενδιάμεσου δικτύου, δηλαδή κάποιες απώλειες μπορεί και να ήταν ένα τυχαίο γεγονός.

Με την ταχεία ανάκαμψη αποφεύγονται περίπου οι μισές εκπνοές χρονιστών επιτυγχάνοντας αύξηση περίπου 30% στην διοχέτευση (throughput). [13.], [18.]



Εικόνα 5.2: Εξέλιξη του μεγέθους παραθύρου (Cwnd) στο TCP Reno

Στις εικόνες 5.2 και 5.3 βλέπουμε την εξέλιξη του παραθύρου συμφόρησης κατά την διάρκεια της γρήγορης αναμετάδοσης (Fast retransmit), όπου σ αυτό το σημείο το cwnd μειώνετε στο μισό.



Εικόνα 5.3: Συμπεριφορά του TCP Reno

5.3 Ο Αλγόριθμος TCP – New Reno

Ο TCP New – Reno, θεωρείτε η αναβάθμιση ή επέκταση του Reno. Έχει κάποια παραπάνω πλεονεκτήματα έναντι του Reno.

Συγκεκριμένα εστιάζει στα χαμένα πακέτα του Reno, από το παράθυρο μετάδοσης, Δηλαδή υπάρχει μια τροποποίηση στον αλγόριθμο Fast Recovery του TCP Reno . είναι μια βέλτιστη τροποποίηση της γρήγορης αναμετάδοσης (Fast Recovery). Δηλαδή έχει την αποτελεσματικότητα να εντοπίζει περισσότερες απώλειές πακέτων στην φάση την γρήγορης αναμετάδοσης του TCP reno, όπου το κυριότερο πρόβλημα η μειονέκτημα που αντιμετωπίζει ο Reno, εστιάζεται στις απώλειες πακέτων στο Δίκτυο. Γενικά τα βήματα ελέγχου συμφόρησης στο TCP Reno και New-Reno είναι ίδια. Η σημαντικότερη αλλαγή είναι στις φάσεις Fast Retransmit Και Fast Recovery. Στον TCP Reno, ο αποστολέας όταν θα στείλει ένα πακέτο, και ο δέκτης θα το παραλάβει σαν μερική επιβεβαίωση, τότε βγάζει τον αποστολέα από την Γρήγορη ανάκαμψη. Στον TCP New - Reno, δεν συμβαίνει αυτό διότι το θεωρεί το πακέτο ότι έχει χαθεί, και πρέπει να γίνει ξανά επαναμετάδοση, γενικά ο New Reno μπορεί και αναγνωρίζει όλα τα πακέτα, σε αντίθεση με τον Reno, που τον χαρακτηρίζει μερική αναγνώριση πακέτων. Έτσι ο TCP New Reno, θεωρείτε ποιο αποτελεσματικός σε πολλαπλές απώλειες πακέτων, γιατί μπορεί να ανακάμψει, χωρίς να λήξει ο χρονομετρητής

Παρακάτω θα δούμε πως συμπεριφέρονται οι αλγόριθμοι γρήγορης επανεκκπομπής, ανάκαμψης

1) Στο 3^ο διπλό ACK όταν ληφθεί, τότε επαναμεταδίδεται το χαμένο πακέτο και τίθεται στο μισό το ssthresh στο τρέχων παράθυρο συμφόρησης cwnd, όχι όμως λιγότερο από 2/3 πακέτα. Το παράθυρο συμφόρησης τότε μεγαλώνει ίσο με το ssthresh δηλαδή 3 φορές το μέγεθος του πακέτου

$$\text{Τύπος: ssthresh} = \max(\text{FlightSize} / 2, 2 * \text{MSS})$$

2) Κατά 1 πακέτο αυξάνεται το cwnd κάθε φορά που φθάνει ένα διπλό ACK.

3) Όταν λαμβάνεται το ACK που επιβεβαιώνει τα δεδομένα, τότε το cwnd θα είναι ίσο με το + στο πρώτο βήμα, Αυτό σημαίνει ότι είναι η επιβεβαίωση της επαναμετάδοσης. Μετά από χρόνο RTT.

4) Το χαμένο πακέτο μπορεί και μεταδίδεται ξανά, όταν το cwnd είναι ίσο με το ssthresh 3*MSS.

5) Όπως και αν η νέα τιμή του cwnd το επιτρέπει μπορεί να μεταδώσει ένα τμήμα του πακέτου.

Και ποιο αναλυτικά θα δούμε και σε κώδικα C++ τα βήματα ελέγχου συμφόρησης.

Step 1: Initially

```
0 < cwnd <= min (4 * mss, max (2 * mss)
```

```
SS_threshold = max (cwnd / 2, 2 * MSS)
```

Step 2: Slow Start Algorithm (Exponential Increases)

```
If (receive acks && cwnd < ss_threshold) cwnd = cwnd + 1;
```

Step 3: Congestion Avoidance Algorithm (Additive Increase)

```
If (receive ACKs) {
```

```
If (cwnd > ss_threshold)
```

```
cwnd = cwnd + segsize * segsize / cwnd;
```

```
Else
```

```
cwnd = cwnd + 1; }
```

Step 4: Congestion Detection Algorithm (Multiplicative Decrease): Fast Retransmission and Fast Recovery

```
If (congestion) {
```

```
If (Receive same Acks 3 time or retransmission time out) {
```

```
SS_threshold = cwnd / 2;
```



```

If (Retransmission time out) {
  cwnd = initial;
  Exit and call Slow Start step;
Else /* Receive same Acks 3 time*/
  cwnd = SS_threshold;
Exit and call congestion avoidance step; } } }

```

To TCP New -Reno, αντιμετωπίζει και αυτό παθογένειες, όπως χρειάζεται χρόνο ίσο με 1 RTT, για να ανιχνεύσει μια απώλεια πακέτου, όταν θα ληφθεί 1^ο ACK, τότε μπορεί να συμπεράνει ποιο άλλο πακέτο χάθηκε.[19.], [20.]

5.4 Ο Αλγόριθμος TCP Vegas

Σε παραπάνω ενότητα αναλύσαμε τον αλγόριθμο του TCP Reno, Τώρα όσον αφορά τον TCP Vegas θεωρείτε ότι είναι μία παραλλαγή ή τροποποίηση του. Όπου Το 1994 παρουσιάστηκε από τον Brakmo (Jamal and Sultan, 2008). Η αυξομείωση του παραθύρου, στηρίζεται στο (RTT Round trip time). Δηλαδή στην εκτίμηση της καθυστέρησης του πακέτου. Θεωρείτε νέος μηχανισμός για τον έλεγχο της συμφόρησης του Δικτύου για την επαναποστολή των χαμένων πακέτων κατά την μεταφορά.. Επίσης διαθέτει νέο μηχανισμό ανίχνευσης συμφόρησης για τις φάσεις της αργής εκκίνησης(slow start) όπως και για την φάση της αποφυγής συμφόρησης (congestion avoidance). Κατά την διάρκεια της συμφόρησης τα πακέτα που υπολογίζονται συνεχώς από τον αποστολέα, οπού σταθεροποιούνται σε έναν αριθμό σχετικά μικρό με τον όγκο που πρόκειται να διοχετευτούν προς τον τελικό αποδεκτή. Ακόμα όπως αναφέραμε υπολογίζει τον χρόνο μετ. επιστροφής του πακέτου (RTT) αν για κάποιο πακέτο δεν έρθει η επιβεβαίωση τότε αναμεταδίδεται . Γίνεται ταχύτερος ελεγχος των χαμένων πακέτων. Σχετικά με τον νέο μηχανισμό της αργής εκκίνησης καθορίζει έτσι με τέτοιο τρόπο το παράθυρο συμφόρησης (cwnd), χωρίς να γίνει κάποια απώλεια πακέτου, υπάρχει μια δίκαιη κατανομή στο μέγεθος που αποθηκεύονται τα πακέτα. Όταν θα λάβει ο αποστολέας διπλό ACK , θα γίνει έλεγχος από τον Vegas σχετικά με το RTT, και αν ο χρόνος αποστολής του τμήματος είναι μεγαλύτερος από το RTT, τότε θα ξαναστείλει το τμήμα αυτό, χωρίς να περιμένει το 3^ο διπλό ACK, όπως συμβαίνει στον Reno.

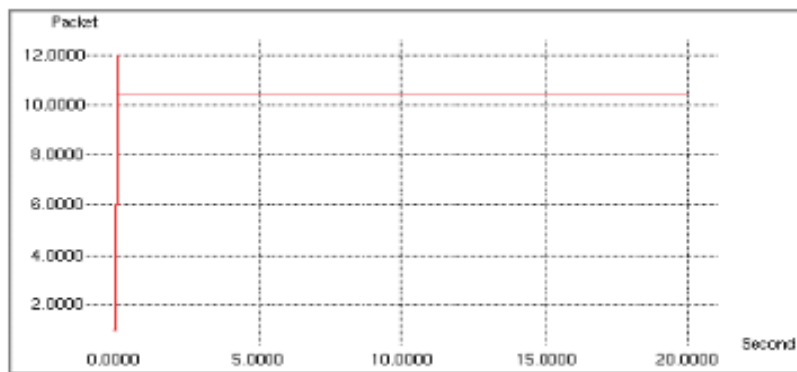
Το Vegas συγκρίνει την πραγματική εκτίμηση RTT με την πιθανή επιστροφή RTT και υπολογίζει τη διαφορά ως εξής:

- $Diff = (Αναμενόμενο - Πραγματικό) BaseRTT$

Η ιδιαιτερότητα του Vegas είναι να προσαρμόζει τον ρυθμό ροής των πακέτων, στην ουσία κρατάει έναν συνεχόμενο ρυθμό ροής καθ' όλη την διάρκεια της σύνδεσης (Εικόνα 5.4) . Παρακάτω περιγράφονται τα κατώτατα όρια σχετικά με την συμμόρφωση του παραθύρου

- Το παράθυρο (cwnd) αυξάνεται αν: $\text{If Diff} < a$,
- Το παράθυρο (cwnd) μειώνεται : $\text{If Diff} > b$,

Όπου τα a & b , θεωρούνται τιμές thresholds, Και diff, η διαφορά όπως αναφέραμε και παραπάνω. [21.]



Εικόνα 5.4: TCP Vegas Ρυθμός Ροής

5.5 Ο Αλγόριθμος TCP Sack

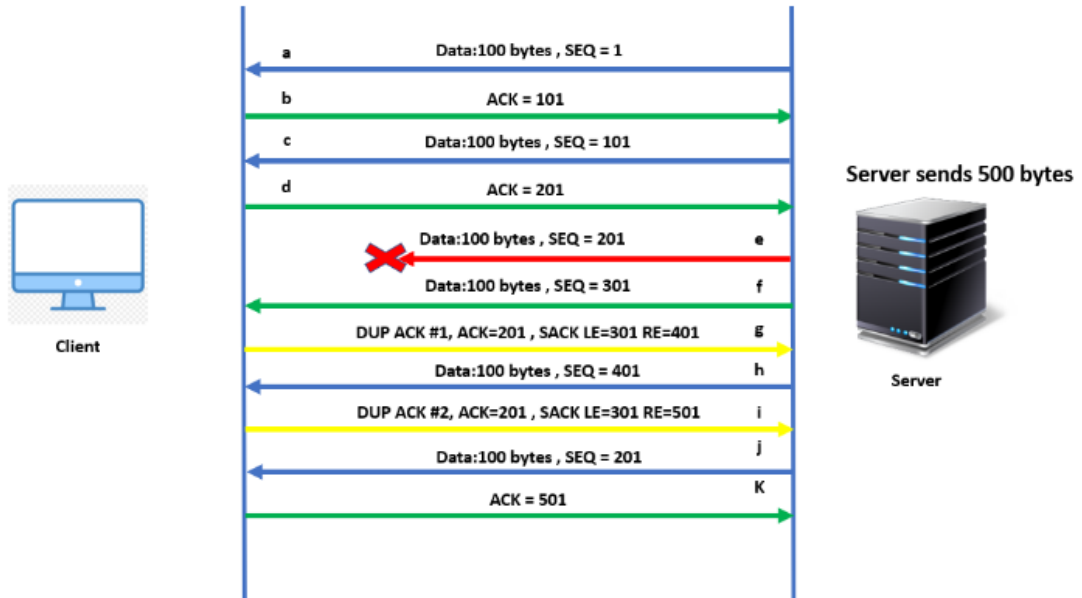
Ο TCP Sack θεωρείτε πως είναι και μία επέκταση του αλγορίθμου TCP New – Reno, και με την σειρά του TCP Reno. Στον New - Reno, όπως αναφέραμε και παραπάνω έχει μειονέκτημα πως για κάθε απώλεια πακέτου διαρκεί ένα RTT. Το πλεονέκτημα που έχει ο TCP Sack είναι ότι παρέχει επιλεκτική επιβεβαίωση πακέτων στον αποστολέα (εικόνα 5.5), δηλαδή με λίγα λόγια, όταν φθάσει στον αποστολέα η επιβεβαίωση, παρέχει πληροφορίες για το πόσα πακέτα έχουν παραληφθεί και πόσα έχουν αναγνωρισθεί. Και έτσι με αυτόν τον τρόπο υπολογίζονται πόσα πακέτα έχουν χαθεί, και έτσι δεν ξαναστέλνονται τα πακέτα.

Ο TCP Sack διατηρεί επίσης την Slow Start και retransmit

Λειτουργία: Στον TCP Sack ο αποστολέας μπαίνει στην διαδικασία γρήγορης ανάκαμψης τότε ορίζεται μια μεταβλητή τύπου σωλήνα ή αγωγού ελέγχει το πλήθος των πακέτων που λείπουν κατά μήκος της πορείας, στην συνέχεια το παράθυρο cwnd πέφτει στο μισό του τρέχοντος μεγέθους και όποτε λαμβάνει μία επιβεβαίωση ACK τότε μειώνει μειώνεται αυτομάτως και το μέγεθος του αγωγού κατά ένα, όπως επίσης όταν αποστέλλεται ένα πακέτο αυξάνεται κατά ένα. Όταν ο αγωγός μειώνεται σε μέγεθος και είναι μικρότερος από το παράθυρο συμμόρφωσης, τότε ανιχνεύει τα τμήματα από τα πακέτα που δεν έχουν ληφθεί

και τα αποστέλλει πάλι, και έτσι μπορεί να τα στείλει σε λιγότερο από ένα RTT, και έτσι λοιπόν ο TCP Sack, στέλνει περισσότερα πακέτα σε ένα μόνο RTT.

Το μειονέκτημα του Sack είναι η περιπλοκή διαδικασία για την επιλογή πακέτων, όπου αυτό αποσκοπεί σε καθυστέρηση στην επιβεβαίωση των πακέτων προς τον αποστολέα [21.], [22.]



Εικόνα 5.5: Επιλεκτική αναγνώριση πακέτων

6. ΕΙΣΑΓΩΓΗ ΣΤΗΝ ΠΡΟΣΟΜΟΙΩΣΗ ΤΩΝ ΔΙΚΤΥΩΝ

Στα Δίκτυα Η προσομοίωση σε υπολογιστή μπορεί να χρησιμοποιηθεί για να βοηθήσει την μοντελοποίηση και την ανάλυση πολλών συστημάτων.

Στην προσομοίωση, το υπό μελέτη δίκτυο μοντελοποιείται και μέσα από μαθηματικούς τύπους υπολογίζεται η αλληλεπίδραση των συστατικών του δικτύου. Η μοντελοποίηση είναι ικανή να αναπαράγει έναν πραγματικό φόρτο και μια πραγματική συμπεριφορά μέσα σε μια τοπολογία ενός πραγματικού δικτύου. Όλες οι μεταβλητές μπορούν να τροποποιηθούν έτσι ώστε να αναδειχθεί η συμπεριφορά του δικτύου κάτω από διαφορετικές συνθήκες. Στην εξομοίωση, πραγματοποιείται μια μελέτη του δικτύου ως προς την απόδοση του έτσι ώστε να προβλεφθεί η επίδραση που θα έχει η όποια αλλαγή σε κάποια παράμετρο λειτουργίας του.

Στα δίκτυα υπολογιστών, η προσομοίωση είναι μια τεχνική όπου ένα πρόγραμμα μοντελοποιεί την συμπεριφορά ενός δικτύου, χρησιμοποιώντας μαθηματικό μοντέλο είτε υπολογίζοντας κάποια (host/πακέτα), είτε συλλαμβάνοντας και αναπαριστώντας τις παρατηρήσεις από ένα δίκτυο. Η συμπεριφορά ενός δικτύου και οι διαφορετικές εφαρμογές και υπηρεσίες που υποστηρίζει μπορούν να παρατηρηθούν μέσω οπτικής απεικόνισης των αποτελεσμάτων της προσομοίωσης. Διαφορετικά χαρακτηριστικά του συστήματος μπορούν να τροποποιηθούν έτσι ώστε να προσδιοριστεί πως θα συμπεριφερόταν το δίκτυο κάτω από διαφορετικές συνθήκες.

6.1 Τα Πλεονεκτήματα Και Μειονεκτήματα της Προσομοίωσης

6.1.1 Πλεονεκτήματα

Τα τελευταία χρόνια με την τεχνολογική ανάπτυξη των μοντέλων προσομοιώσεων έχουμε την δυνατότητα οι χρήστες να μπορούν να αναπτύξουν, να μελετήσουν και να βελτιώσουν για παράδειγμα τα συστατικά στοιχεία ενός φυσικού δικτύου με μεγάλη ακρίβεια και λεπτομέρεια και να προσομοιώσουν τις τυχόν προβλήματα των πολλαπλών επιλογών σχεδιασμού του, προτού χρειαστεί να εφαρμόσουν την εφαρμογή των αποτελεσμάτων σε πραγματικό χρόνο. Σε γενικές γραμμές μπορούμε να ορίσουμε ότι παραμέτρους θέλουμε σε μια προσομοίωση, λαμβάνοντας πολύ λεπτομερή αποτελέσματα τα οποία δεν μπορούμε να τα μελετήσουμε πειραματικά με άλλη μέθοδο με την μέχρι τώρα.

Σε γενικές γραμμές κάποια από τα πλεονεκτήματα είναι:

- Σε πολλές περιπτώσεις μπορεί να κοστίζει λιγότερο σε χρήμα και χρόνο.
- Κάποιες φορές αποτελεί την μόνη λύση για την αντιμετώπιση κάποιων προβλημάτων.
- Αποτελεί μια ασφαλής μέθοδος.
- Δίνεται η δυνατότητα πολλαπλών προσομοιώσεων.

6.1.2 Μειονεκτήματα

Σε μία προσομοίωση, υπάρχουν και κάποιοι παράμετροι που μπορούν να δυσκολέψουν την διεξαγωγή των πειραμάτων.

Για την ανάπτυξη ενός προσομοιωτή και την ακριβή και αξιόπιστη λειτουργία του απαιτούνται σημαντικοί πόροι, όπως χρήματα και εργατοώρες.

Επίσης υπάρχουν και άλλοι παράμετροι που επηρεάζουν ένα φυσικό δίκτυο, που όπως είναι αντιληπτό ένα μοντέλο προσομοιώσεις είναι αντικειμενικά δύσκολο να τους συμπεριλάβει όλους τους παράγοντες πράγμα που καθιστά αδύνατο. Ένα μοντέλο προσομοίωσης για να πάρει τα μέγιστα αποτελέσματα και να φτάσει σε ένα σημείο βέλτιστο, όσο επιτυχημένη και αν είναι η προσομοίωση, θα παραμένουν μόνο εκτιμήσεις/προβλέψεις.

Κάποια από τα μειονεκτήματα είναι:

- Κόστος ανάπτυξης σε χρόνο και χρήμα
- Μπορεί να μας οδηγήσει σε λάθος αποτελέσματα και εκτιμήσεις
- Δεν αντανakλά με ακρίβεια το φυσικό δίκτυο με το εικονικό

6.2 Το OMNeT++

Το OMNeT++, είναι λογισμικό προσομοίωσης ενσύρματων και ασυρμάτων δικτύων το οποίο είναι γραμμένο πάνω σε γλώσσα προγραμματισμού C++. Μπορεί να χρησιμοποιηθεί για κατασκευή και προσομοιώσεις δικτύων δηλαδή εικονικά δίκτυα, όπου μπορούμε εμείς οι ίδιοι να κατασκευάσουμε κάποιο πριν ακόμα την φυσική του κατασκευή έτσι ώστε να πάρουμε διάφορες μετρήσεις σχετικά με την απόδοση του δικτύου χωρίς να χάσουμε έτσι κάποιο ανεπιθύμητο κόστος αλλά και χρόνο πριν την κατασκευή του, αλλά και όχι μόνο όπως επίσης για διάφορες αλλά μελέτες, όπως να πάρουμε κάποιο ήδη κατασκευασμένο δίκτυο, και να λάβουμε κάποια αποτελέσματα που θέλουμε.

Το OMNeT++ παρέχει την ανοιχτή βιβλιοθήκη λογισμικού INET, στην οποία υπάρχουν έτοιμα διαφορά ενσύρματα και ασύρματα δίκτυα όπως αυτό το οποίο θα δούμε και παρακάτω για τις προσομοιώσεις .

Τα κυρία χαρακτηριστικά του είναι οι κατασκευές των δικτύων οι οποίες είναι με κατάληξη αρχείου (εικόνα 3.1) στο τέλος “.ned”. Όπως βλέπουμε παρακάτω στην εικόνα. Με λίγα λόγια βρίσκεται το κατασκευασμένο δίκτυο .



Εικόνα 6.1: ClientServer.ned

Άλλος σημαντικός παράγοντας είναι τα αρχεία με κατάληξη “.ini”, όπου βρίσκεται το γραφικό περιβάλλον αλλά και ο κώδικας παραμετροποίησης των δικτύων όπως π.χ. μπορούμε να ρυθμίσουμε τον ρυθμό αποστολής των δεδομένων ή το μέγεθος του αρχείου. Σχετικό παράδειγμα βλέπουμε στην εικόνα 6.2

- Configuration
- General
- Scenarios
- Random Numbers
- Result Recording
- Debugging
- Event Log
- Cmdenv
- Qtenv
- Tkenv
- Extensions
- Parallel Simulation
- Advanced
- Sections
- Parameters

Parameter Assignments

Configuration: Config inet-reno

Network: ClientServer
Section fallback chain: Config inet-reno > General

HINT: Drag the icons to change the order of entries.

Section/Key	Value	Comment	
General			
• **server.hasPcapRecorder	⚠ true		
• **server.pcapRecorder.pcapFile	⚠ "results/server.pcap"		
• **client1.hasPcapRecorder	⚠ true		
• **client1.pcapRecorder.pcapFile	⚠ "results/client1.pcap"		
• **crcMode	"computed"		
• **pcapRecorder.pcapLinkType	⚠ 204	PPP	
• **numApps	1		
• **client*.app[*].typename	"TcpSessionApp"		
0_1 **client*.app[0].active	true		
0_1 **client*.app[0].localPort	-1		
• **client*.app[0].connectAddress	"server"		
0_1 **client*.app[0].connectPort	1000		
• **client*.app[0].tOpen	0.2s		
• **client*.app[0].tSend	0.4s		
• **client*.app[0].sendBytes	1000000B		
0_1 **client*.app[0].sendScript	""		
• **client*.app[0].tClose	25s		
• **server*.app[*].typename	"TcpEchoApp"		
0_1 **server*.app[0].localPort	1000		
• **server*.app[0].echoFactor	2.0		
0_1 **server*.app[0].echoDelay	0s		

New
Remove
Add...

17 unassigned parameters

Εικόνα 6.2: Parameter Assignments

Τέλος για να πάρουμε τα αποτελέσματα από την προσομοίωση κάποιου δίκτυο όπως τις γραφικές παραστάσεις που θα σούμες και στην συνέχεια υπεύθυνα για αυτήν την δουλειά είναι η καρτέλα “results. (Εικόνα 6.3).



Εικόνα 6.3: Αρχεία Results

Όπου με την επιλογή πάνω στα αρχεία με κατάληξη “.sca”, μεταφερόμαστε στο γραφικό περιβάλλον με τα στατιστικά και τις γραφικές παραστάσεις μετρά το τέλος της προσομοίωσης. Παρακάτω στην εικόνα 6.4 βλέπουμε και ένα δείγμα με κάποιες από τις επιλογές του περιβάλλοντος

Browse Data

Here you can see all data that come from the files specified in the Inputs page.

All (178 / 178) Vectors (52 / 52) Scalars (120 / 120) Histograms (6 / 6)

runID filter module filter statistic name filter [↓] [↕]

Name	Value	
> ClientServer.client1.lo[0].lo		
> ClientServer.client1.ppp[0].ppp		
> ClientServer.client1.ppp[0].ppp.queue		
> ClientServer.client1.pppg\$o[0].channel		
> ClientServer.client1.tcp.conn-4		
> ClientServer.client1.udp		
> ClientServer.server.app[0]		
> ClientServer.server.ipv4.arp		
> ClientServer.server.ipv4.ip		
> ClientServer.server.lo[0].lo		
> ClientServer.server.ppp[0].ppp		
> ClientServer.server.ppp[0].ppp.queue		
> ClientServer.server.pppg\$o[0].channel		
> ClientServer.server.tcp.conn-1		
▼ ClientServer.server.tcp.conn-5		
> dupAcksvector (vector)	0.0 (1890)	
> rcvAcksvector (vector)	50002.0 (1891)	
> rcvAdvsvector (vector)	557069.9534883721 (1892)	
> rcvOooSegsvector (vector)	138.0 (275)	
> rcvSeqsvector (vector)	549776.0888418827 (1891)	
> rcvWndsvector (vector)	7502.299312533051 (1891)	
> sndAcksvector (vector)	549831.8350079323 (1891)	
> sndWndsvector (vector)	7504.0 (2)	
> tcpRcvQueueBytesvector (vector)	473.19968262364455 (3781)	
> unackedsvector (vector)	0.0 (1)	
> ClientServer.server.udp		

Εικόνα 6.4: Browse Data

7. ΠΡΟΣΟΜΕΙΩΣΕΙΣ ΤΩΝ TCP VARIANTS ΣΤΟ OMNeT++

7.1 Εισαγωγή

Στην ενότητα αυτή παρουσιάζεται η πειραματική αξιολόγηση των διαφόρων εκδοχών του TCP που έγινε για να μελετηθεί η συμπεριφορά τους και η αποτελεσματικότητα τους. Για την αξιολόγηση χρησιμοποιήθηκε το OMNeT++ 5.6.2 και η βιβλιοθήκη INET 4.6.2. Το λογισμικό εγκαταστάθηκε σε επιτραπέζιο υπολογιστή με τα ακόλουθα χαρακτηριστικά:

1. Έκδοση Λογισμικού Συστήματος: Windows 10 Pro, 64bit
 2. 8 GB μνήμη RAM
 3. Intel(R) Core(TM) i5- 10400F 2.9GHz CPU,
 4. Storage 128 GB- SSD
- Η σύγκριση έγινε μεταξύ των:
1. Του τυπικού πρωτοκόλλου TCP (Sack)
 2. Του πρωτοκόλλου TCP Reno
 3. Του πρωτοκόλλου TCP New – Reno
 4. Του πρωτοκόλλου TCP Vegas
- Όπου τα σενάρια που υλοποιήθηκαν:
1. Ενσύρματη επικοινωνία client -server: Το συγκεκριμένο μοντέλο αποτελείται από τον αποστολέα και τον παραλήπτη, που επικοινωνούν μεταξύ τους με μία ενσύρματη γραμμή των 100Mbps με τύπο καλωδίου UTP. Ο προεπιλεγμένος ρυθμός μετάδοσης της σύνδεση μεταξύ client -server ήταν των 10Mbps. Εμείς επιλέγουμε να την αναβαθμίσουμε γιατί λόγω ότι πλέον και στην συνέχεια η πλειοψηφία των γραμμών ανεβαίνει σε ρυθμό ταχύτητας.
 2. Και η χρονική διάρκεια της προσομοίωσης σε όλα τα πρωτόκολλα είναι στα δύο (2) sec.

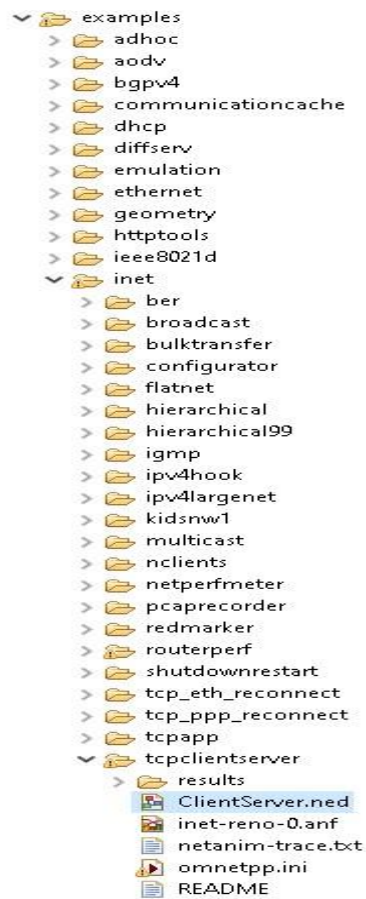
7.1.1 Μετρικές απόδοσης

Η αξιολόγηση των πρωτοκόλλων βασίστηκε στις παρακάτω μετρικές:

- **Throughput:** Η μέγιστη δυνατότητα ροής δεδομένων ανάμεσα στα δύο άκρα της σύνδεσης, με συνυπολογισμό της επιβάρυνσης των επιπλέον πληροφοριών του δικτύου (data transferred/time). Μετράτε σε bits ή Bytes per second (bps/Bps)
- **End-to-end delay:** Η καθυστέρηση από άκρο σε άκρο αναφέρεται στο χρόνο που χρειάζεται για να μεταδοθεί ένα πακέτο σε ένα δίκτυο από την πηγή στον προορισμό.
- **Παράθυρο Συμφόρησης (Cwnd):** τις διακυμάνσεις που είχε ως το μέγεθος κατά την συμφόρηση.
- **RTT(Round Trip Time):** Η χρονική διάρκεια επιβεβαίωσης ενός πακέτου
- **Αριθμός μεταφοράς πακέτων κατά την διάρκεια της προσομοίωσης**
- Και τέλος τα **ποσοστά χρήσης** των γραμμών ανά πρωτόκολλο

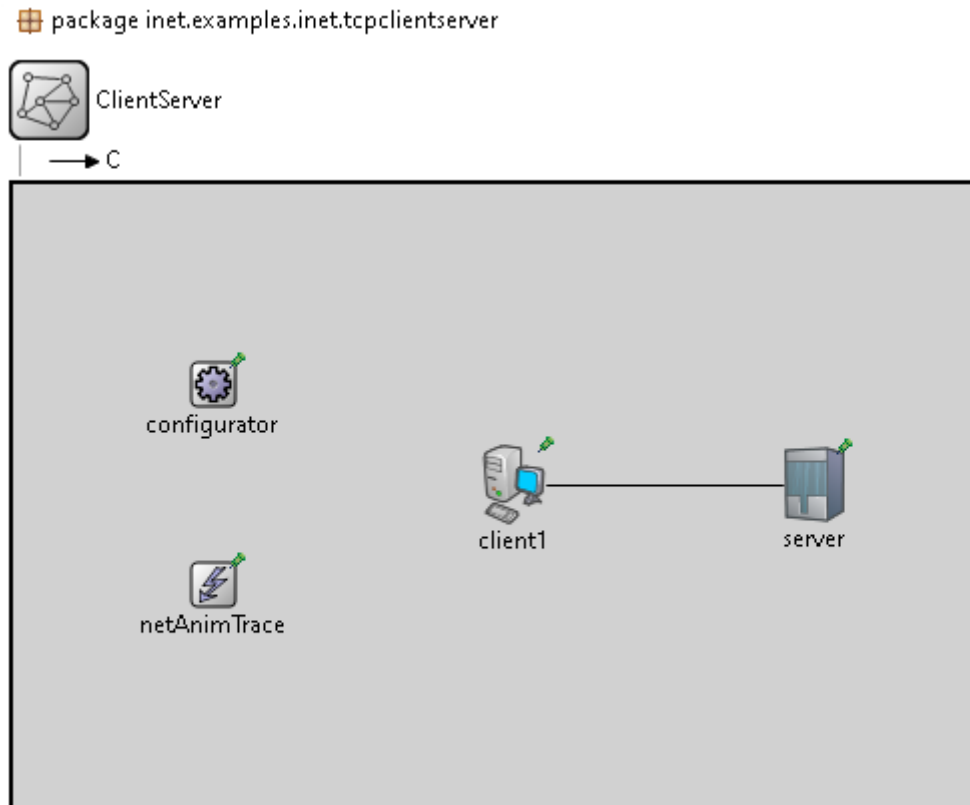
7.2 Simulation TCP Variants

Όπως αναφέραμε και παραπάνω, η βιβλιοθήκη inet προσφέρει μία πληθώρα έτοιμων παραδειγμάτων δικτύων. Για το παρακάτω χρησιμοποιήσαμε το μοντέλο “tcpclientserver” και εφαρμόσαμε τους αλγόριθμους (Tcp Reno, Tcp New – Reno, Tcp Sack, & Tcp Vegas) το οποίο είναι εγκατεστημένο στον φάκελο examples όπως φαίνεται παρακάτω στην εικόνα 7.1, το οποίο αρχείο είναι κατάληξης .NED, το οποίο αυτό του τύπου κατάληξη είναι η ολοκληρωμένη τοπολογία ενός δικτύου όπως και όλα τα αρχεία.



Εικόνα 7.1: Παράδειγμα - tcpclientserver

Και η γραφική απεικόνιση του Δικτύου:



Εικόνα 7.2: Γραφική απεικόνιση μοντέλου Client - Server

Στην παραπάνω (εικόνα 7.2), αυτό που βλέπουμε στην γραφική απεικόνιση του δικτύου που αποτελείτε:

- από ένα client και server που επικοινωνούν μεταξύ τους ενσύρματα, τις ρυθμίσεις του δικτύου (configuration), και την τροφοδοσία του δικτύου (net AminTrace)
- Το συγκεκριμένο μοντέλο αποτελείτε από αποστολέα και έναν παραλήπτη, όπου επικοινωνούν μεταξύ τους από μία ενσύρματη γραμμή των 100Mbps με τύπο καλωδίου UTP. (η αρχική γραμμή ήταν των 10Mbps, όμως την αναβαθμίσαμε σε 100Mbps λόγω ότι πλέον και στην συνέχεια η πλειοψηφία των γραμμών ανεβαίνει σε ρυθμό ταχύτητας).
- Και το μέγεθος του πακέτου που αποστείλαμε είναι μεγέθους 7 Mb.

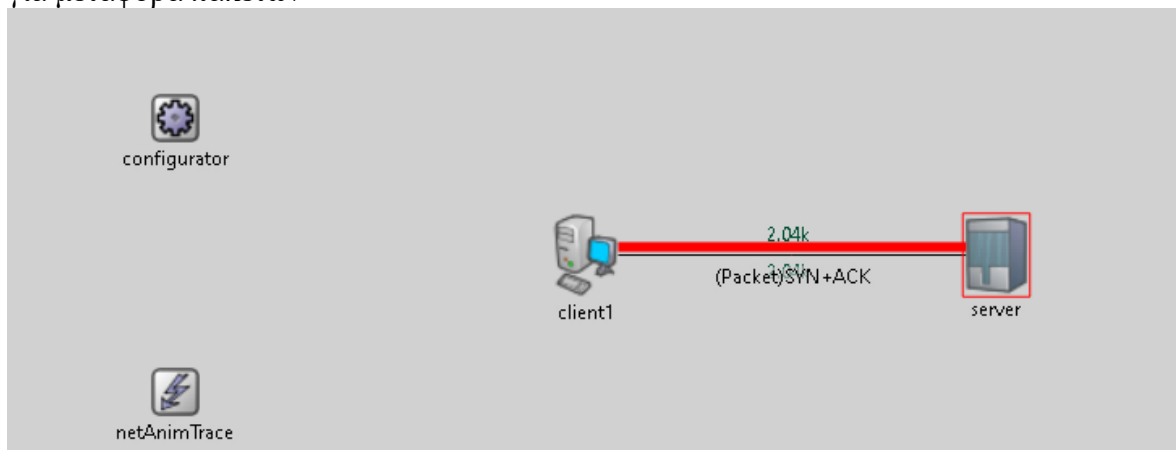
Υπάρχει και το σχετικό παράρτημα στο τέλος της εργασίας με τον κώδικα του μοντέλου client – server.

Αρχική Διαδικασία προσομοίωσης

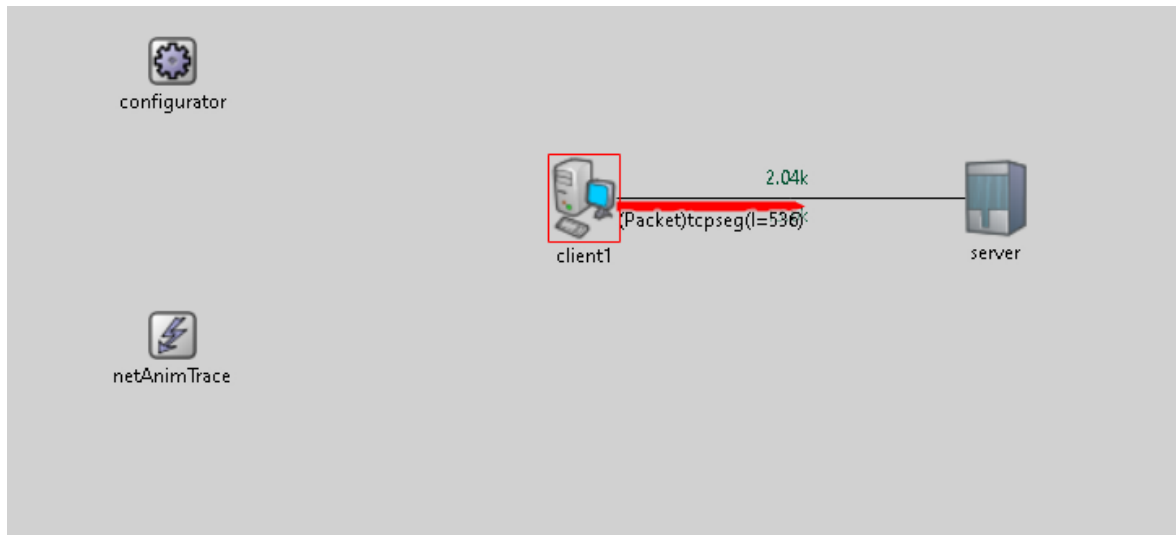
```
** Event #1 t=0 ClientServer.server.tcp (Tcp, id=51) on PassiveOPEN (inet::Request, id=90)
Initializing module ClientServer.server.tcp.conn-1, stage 0
INFO:Tcp connection created for (inet::Request)PassiveOPEN
DETAIL (TcpConnection)ClientServer.server.tcp.conn-1:Connection <none>:-1 to <none>:-1 on socketId=1 in INIT
INFO (TcpConnection)ClientServer.server.tcp.conn-1:App command: OPEN_PASSIVE
DETAIL (TcpConnection)ClientServer.server.tcp.conn-1:Starting to listen on: <none>:1000
INFO (TcpConnection)ClientServer.server.tcp.conn-1:Transition: INIT --> LISTEN (event was: OPEN_PASSIVE)
** Event #2 t=0.2 ClientServer.client1.app[0] (TcpSessionApp, id=18) on selfmsg timer (omnetpp::cMessage, id=72)
INFO:Connecting to server(192.168.1.2) port=1000
** Event #3 t=0.2 ClientServer.client1.tcp (Tcp, id=16) on ActiveOPEN (inet::Request, id=99)
Initializing module ClientServer.client1.tcp.conn-4, stage 0
INFO:Tcp connection created for (inet::Request)ActiveOPEN
DETAIL (TcpConnection)ClientServer.client1.tcp.conn-4:Connection <none>:-1 to <none>:-1 on socketId=4 in INIT
INFO (TcpConnection)ClientServer.client1.tcp.conn-4:App command: OPEN_ACTIVE
DETAIL (TcpConnection)ClientServer.client1.tcp.conn-4:Assigned ephemeral port 1025
DETAIL (TcpConnection)ClientServer.client1.tcp.conn-4:OPEN: <none>:1025 --> 192.168.1.2:1000
INFO (TcpConnection)ClientServer.client1.tcp.conn-4:Tcp Header Option MSS(=536) sent
INFO (TcpConnection)ClientServer.client1.tcp.conn-4:Sending: .1025 > .1000: SYN [49999..49975] (l=-24) win 7504 options MSS
INFO (TcpConnection)ClientServer.client1.tcp.conn-4:Transition: INIT --> SYN_SENT (event was: OPEN_ACTIVE)
```

Εικόνα 7.3: Έναρξη Client-Server

Στην εικόνα 7.3 βλέπουμε ότι η σύνδεση είναι αρχικά σε κατάσταση OPEN_PASSIVE που όπως αναλύσαμε σε προηγούμενο κεφάλαιο αυτό σημαίνει ότι βρίσκονται πάντα σε μία κατάσταση αναμονής, αυτό δείχνει ότι είναι σε ετοιμότητα να λάβουν και να στείλουν τα πακέτα. Και τέλος μεταβαίνει σε κατάσταση OPEN_ACTIVE, και η σύνδεση είναι έτοιμη για μεταφορά πακέτων



Εικόνα 7.4: Γραφική Απεικόνιση SYN+ACK



Εικόνα 7.5: Αποστολή Πακέτων

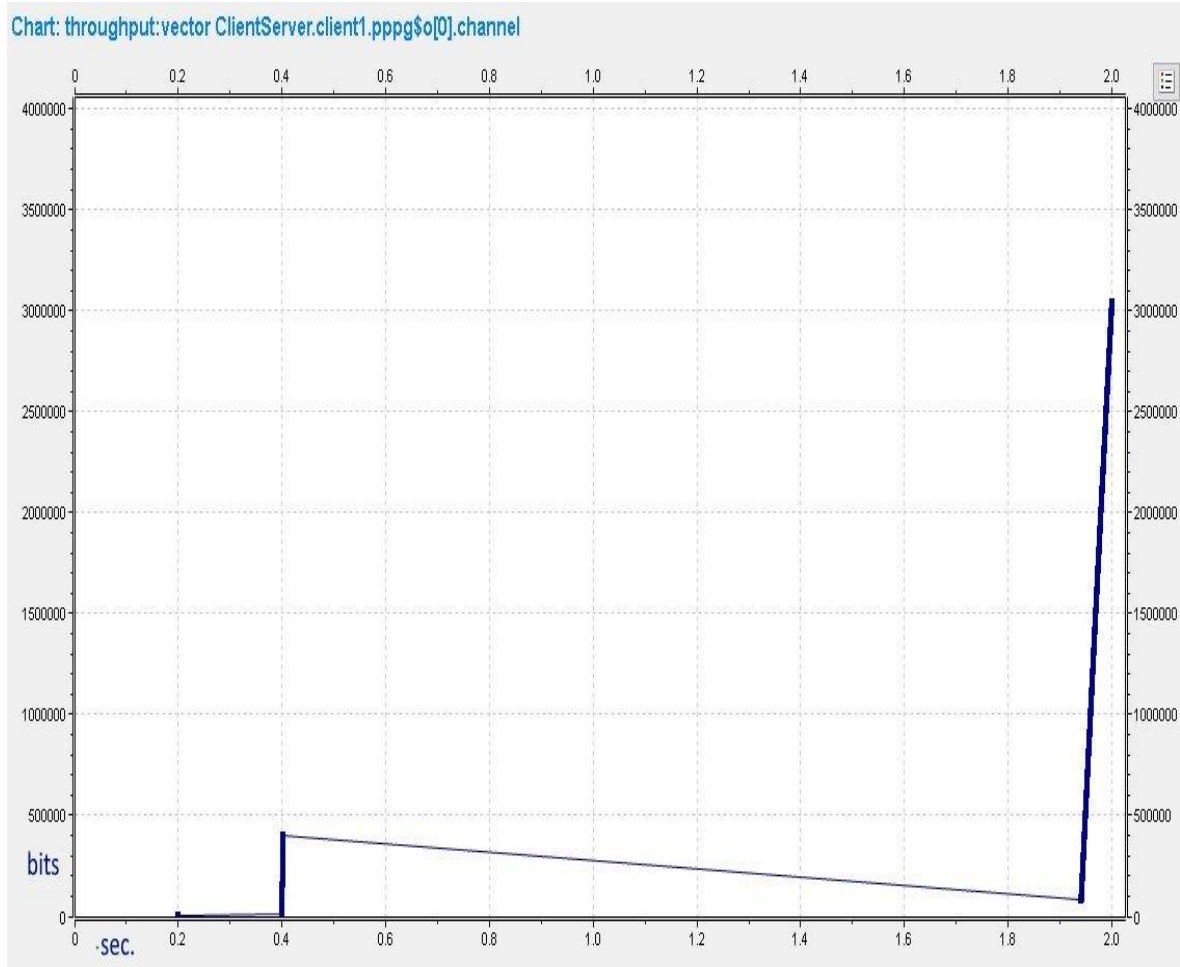
Για να ξεκινήσει η διαδικασία αποστολής των δεδομένων σε αρχική φάση όπως παρατηρούμε και από τις παραπάνω εικόνες πρέπει:

- Ο client να στείλει ένα μήνμα στον server έτσι ώστε να γίνει η εκκίνηση του πρωτοκόλλου σύνδεσης ,(εικόνα 7.3).
- Στην συνέχεια να ανοιχτούν οι θύρες (ports) προορισμού , (εικόνα 7.3)
- Να σταλθεί έπειτα από τον client το αίτημα SYN συγχρονισμού (εικόνα 7.3)
- Και τέλος ο server να στείλει στον client την επιβεβαίωση SYN+ ACK, ότι από την πλευρά του είναι εγκατεστημένη σύνδεση και έτοιμη για αποστολή πακέτων (Εικόνα 7.4)
- Αποστολή πακέτων (Εικόνα 7.5)

7.2.2 Περίπτωση Μελέτης Throughput

Το throughput όπως αναφέραμε μετράτε η μέγιστη ροή των δεδομένων ανά δευτερόλεπτο, δηλαδή bits/sec ή bytes/sec, αναμεσα σε μία σύνδεση. Εμείς πήραμε ως μέτρο σύγκρισης την μονάδα bits/sec.

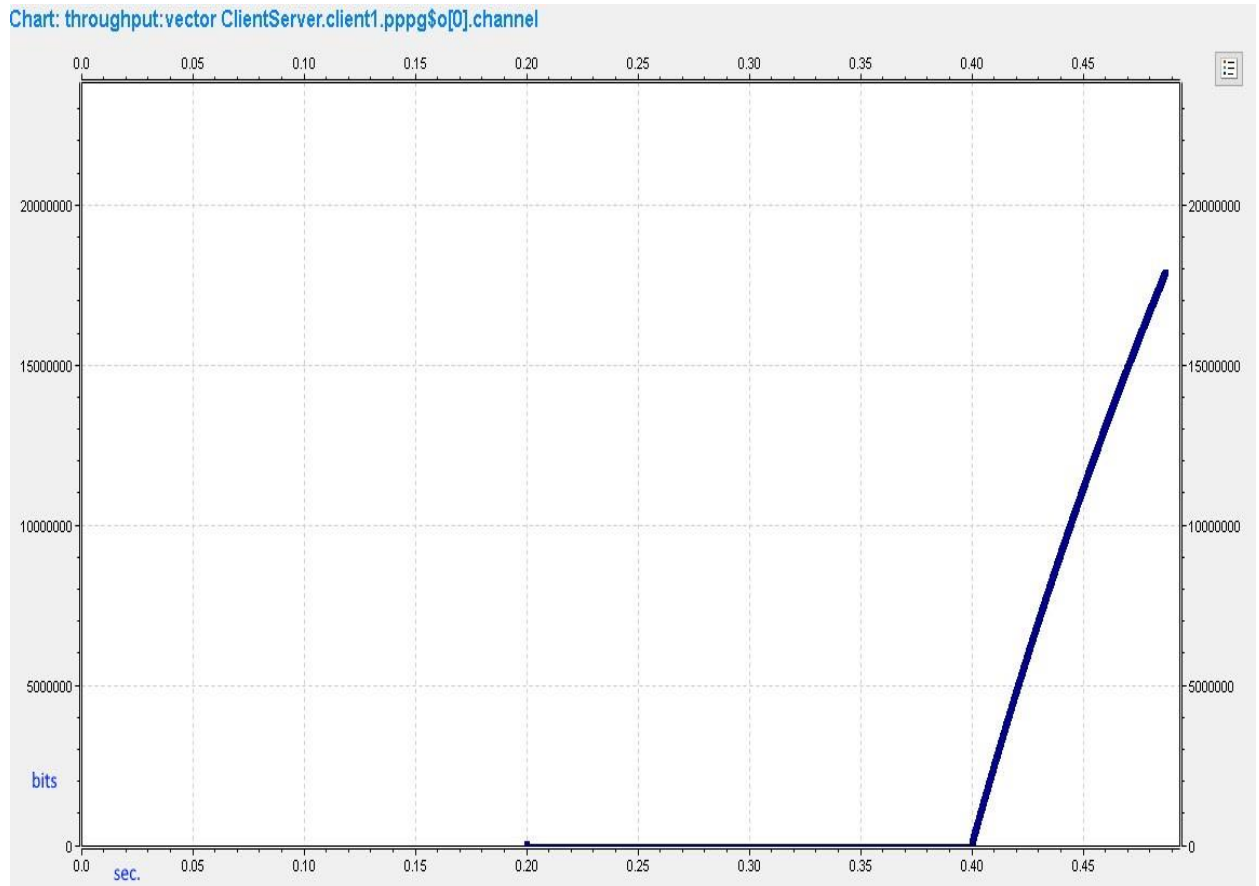
TCP Reno:



Εικόνα 7.6: Throughput TCP Reno

Στην εικόνα 7.6 παρατηρούμε στο χρονικό όριο μεταξύ 0.2 sec & 0.4 sec, ότι δεν έγινε καμία ροή των δεδομένων προς το client. Στο σημείο των 0.4 sec παρατηρήθηκε μία μικρή αύξηση των bits περίπου 500000 αλλά αμέσως μειώθηκε η ροή γραμμικά μέχρι την χρονική περίοδο περίπου των 1.94 sec. Όπου πήρε μία απότομη κλήση η καμπύλη και μέχρι το πέρας της προσομοίωσης έφτασε στο μέγιστο ρυθμό μετάδοσης λίγο πάνω από 3.000.000 bits.

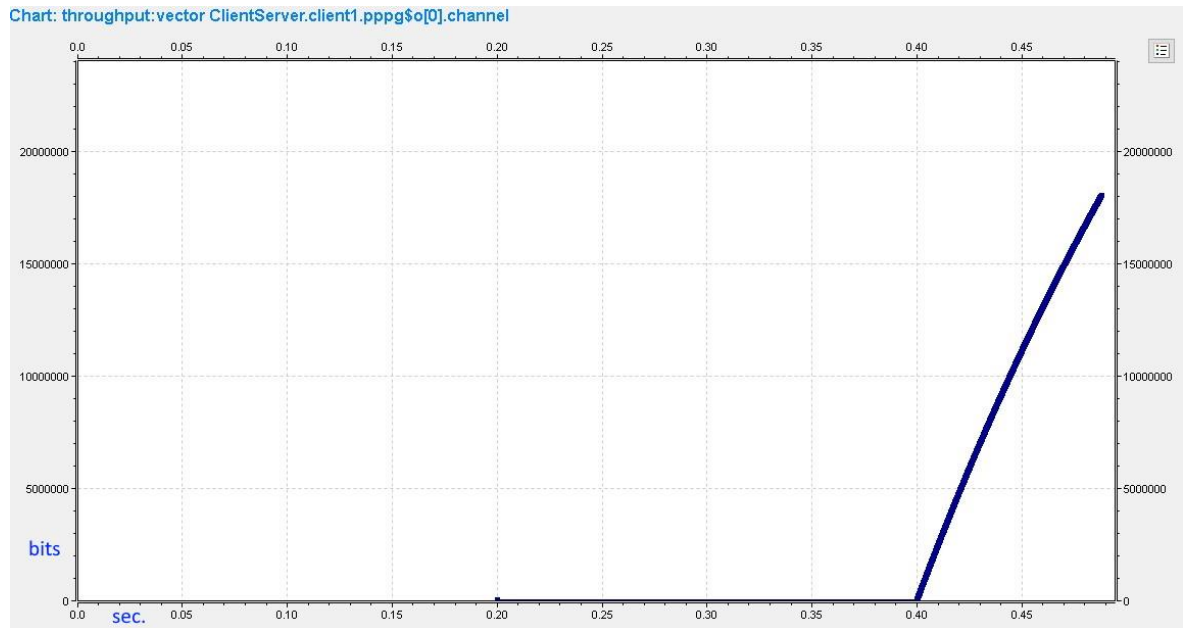
TCP New – Reno:



Εικόνα 7.7: Throughput TCP New – Reno

Στο γράφημα 7.7, για τον αλγόριθμο του New – Reno την χρονική στιγμή 0.4 sec και έπειτα αρχίζει και παρατηρείτε μια μεγάλη απότομη αύξηση της ροής, και σε σχέση με τον Reno ο μέγιστος όγκος ροής των bits είναι πολύ μεγαλύτερος, φτάνει μέχρι 18.000.000 bits

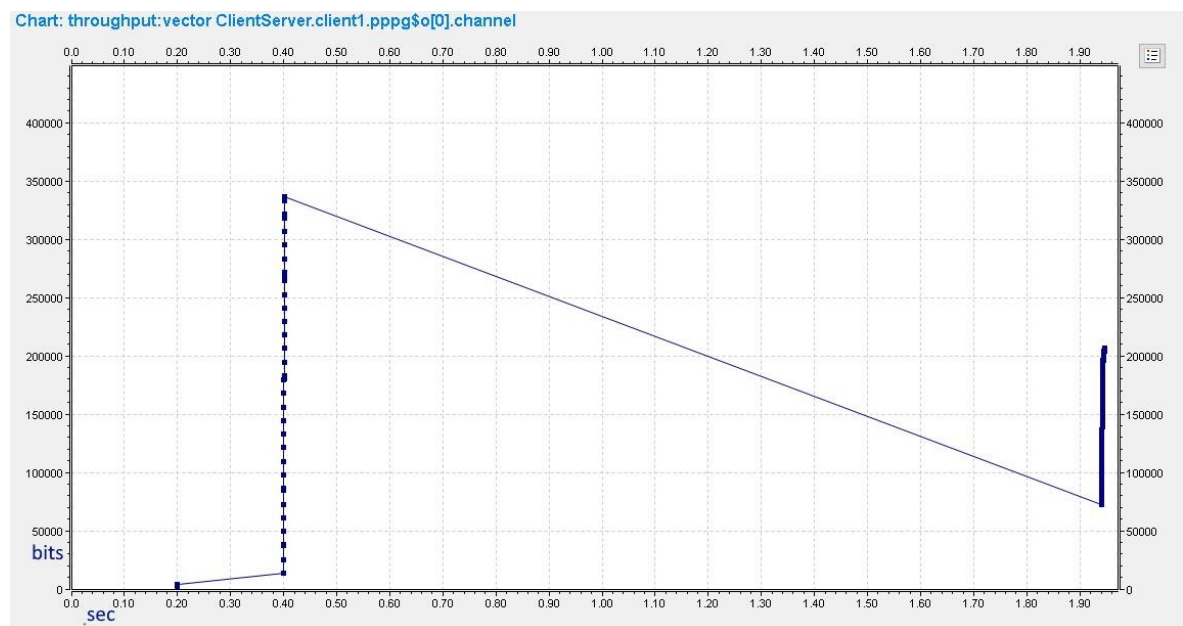
TCP Vegas



Εικόνα 7.8: Throughput TCP Vegas

Παραπάνω στο γράφημα 7.8, η συμπεριφορά του TCP Vegas δεν παρουσιάζει καμία σχεδόν σημαντική αλλαγή ως προς την ροή των δεδομένων.

TCP Sack



Εικόνα 7.9: Throughput TCP Sack

Τέλος στον TCP Sack, όπως παρατηρούμε και από το γράφημα 7.9, στο χρονικό διάστημα 0.2 sec και 0.4 sec υπάρχει μία πολύ μικρή ροή δεδομένων, όπου όμως την χρονική στιγμή $t = 0.4$ παρουσιάζει μία μεγάλη απότομη αύξηση, φτάνοντας στο μέγιστο των bits κοντά στα 350000 bits περίπου, αριθμός αισθητά μειωμένος σε σχέση με τους προηγούμενους αλγόριθμους

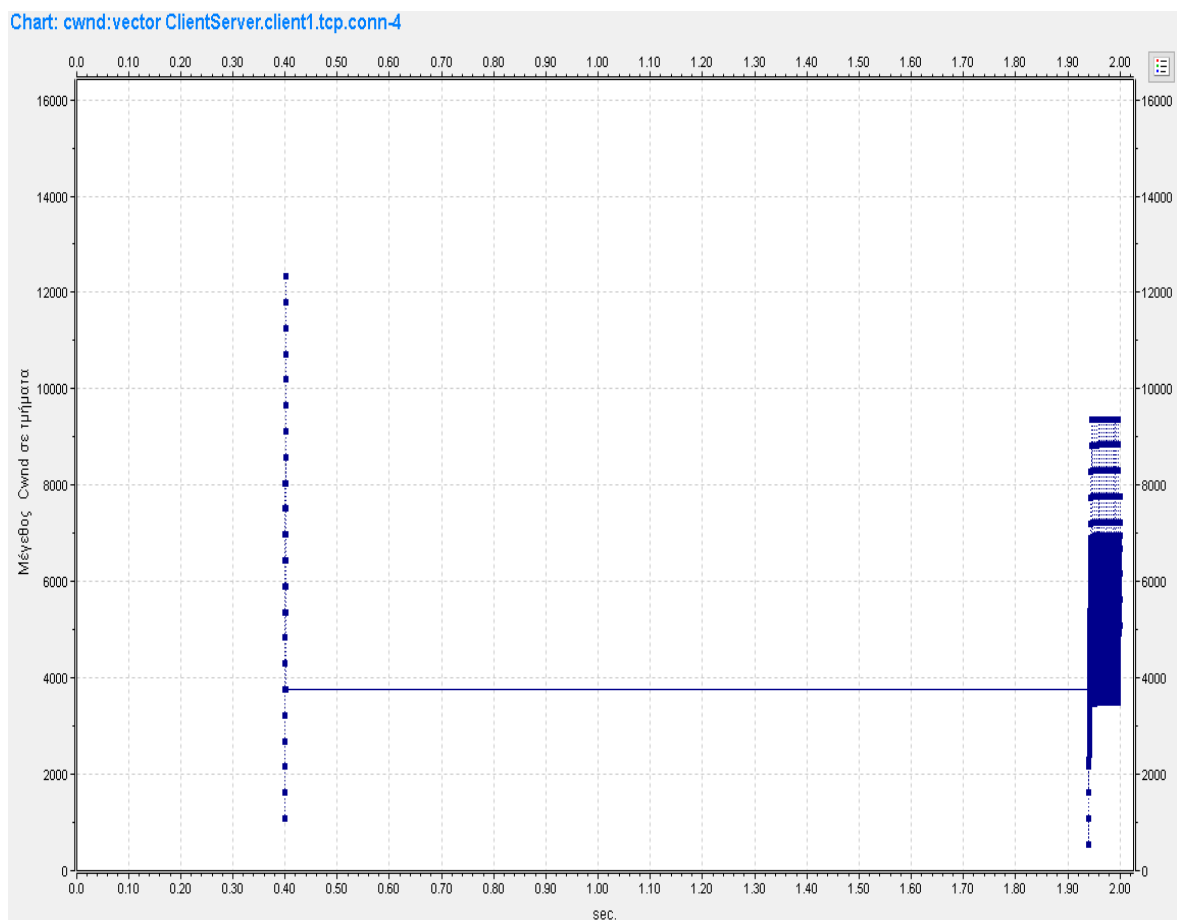
Αν παρατηρήσουμε καλύτερα στην κάθετη γραμμή στο σημείο των 0.4 sec. Υπάρχουν κάποιες διακυμάνσεις στα δεδομένα, αυτό συμβαίνει γιατί όπως εξηγήσαμε και στην ανάλυση της ενότητας του Tcp Sack , λόγω τις επιλεκτικής αναγνώρισης των πακέτων που εφαρμόζει

7.2.3 Περίπτωση Μελέτης Cwnd (Παράθυρο Συμφόρησης)

Το Cwnd (Παράθυρο Συμφόρησης), το μέγεθος του προσμετράτε σε τμήματα. Οπου κάθε τμήμα αποτελείτε από bytes. Χρησιμοποιείτε από το TCP κατά την περίοδο που παρατηρείτε συμφόρηση στο δίκτυο, έτσι ώστε να μειώνει την ροή των δεδομένων και να αποτρέπει κάποια πιθανή κατάρρευση του δικτύου.

TCP Reno

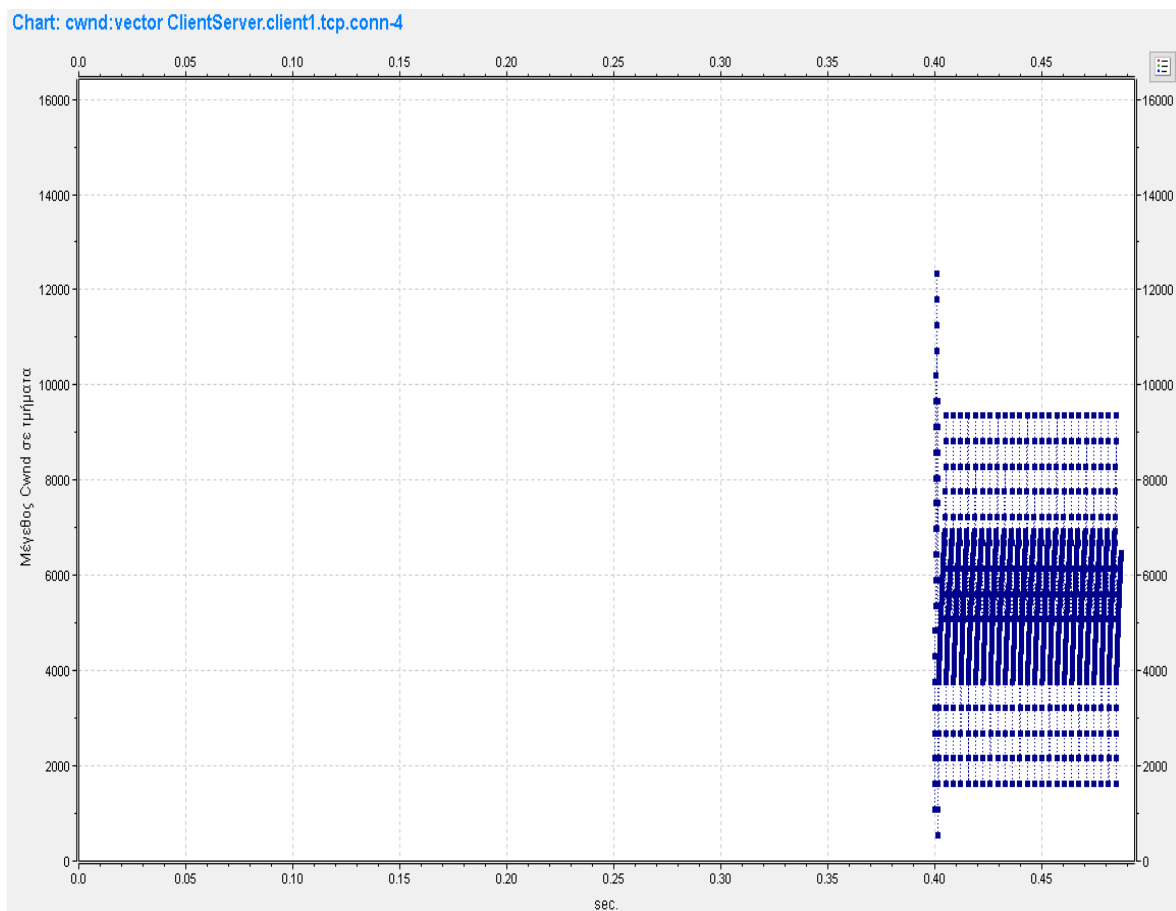
Στο παρακάτω γράφημα (Εικόνα 7.10) βλέπουμε διακύμανση του παραθύρου από την συμφόρηση που συνέβη στο δίκτυο μεταξύ χρονικού διαστήματος $t=0.4\text{sec.}$ και $t=2\text{sec.}$



Εικόνα 7.10: Cwnd TCP Reno

Συμπερασματικά αυτό που αντιλαμβανόμαστε, είναι ότι το παράθυρο συμφόρησης αυξήθηκε 2 φορές έτσι ώστε να αποφευχθεί η κατάρρευση του δικτύου από όταν έφτασε στο μέγιστο όγκο δεδομένων που μεταφέρθηκαν και βλέπουμε στον TCP Reno, με την διαδικασία γρήγορης ανάκαμψης (fast recovery), αποφεύγεται σχετικά γρηγορά η κατάρρευση του δικτύου με το που διαπιστώνει κάποια είδους συμφόρηση.

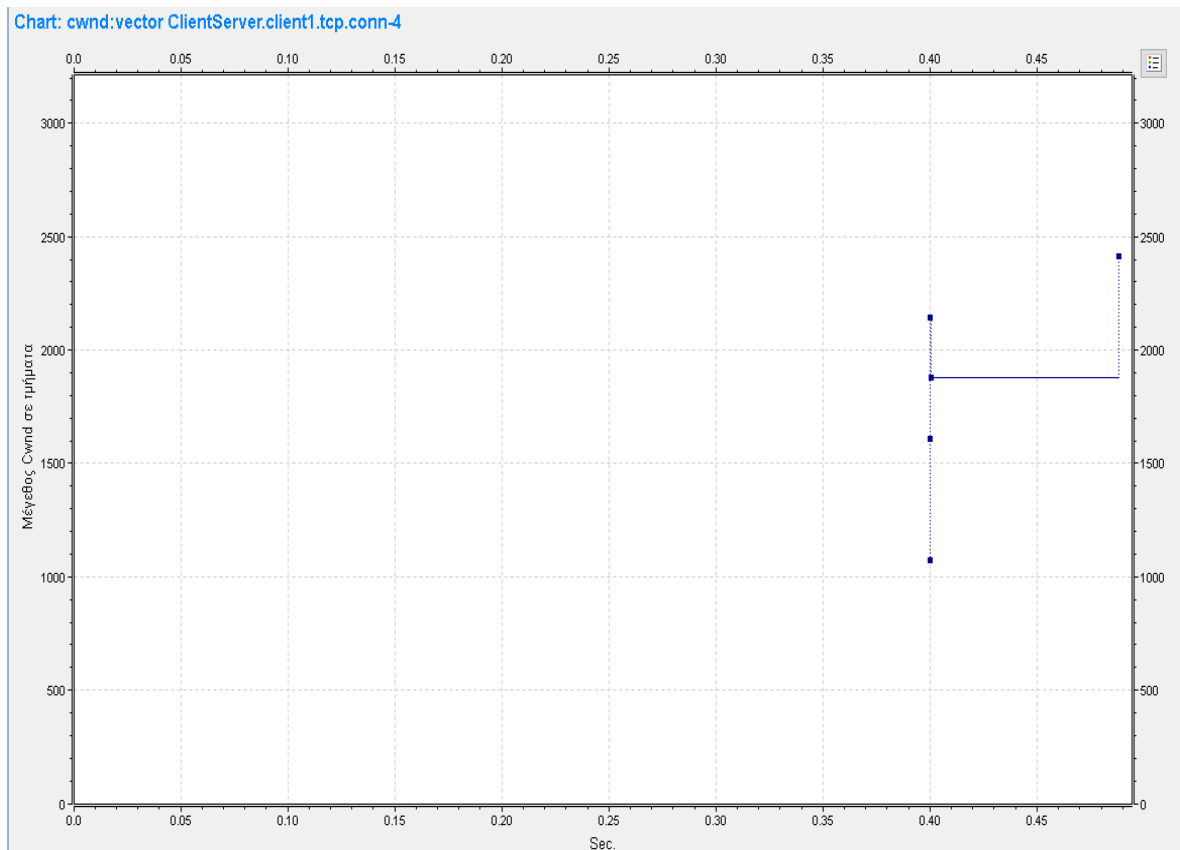
TCP New Reno



Εικόνα 7.11: Cwnd TCP New Reno

Αντιθέτως με τον Reno, στον New- Reno βλέπουμε ότι το ειδικότερα στο $t = 0.4s$ και έπειτα το παραθυρο και είχε πολλές διακυμάνσεις μήκος. Και επίσης το παραθυρο εφτασε να έχει την ίδια τιμη με το ssthresh εκεινη την χρονική στιγμή.

TCP Vegas

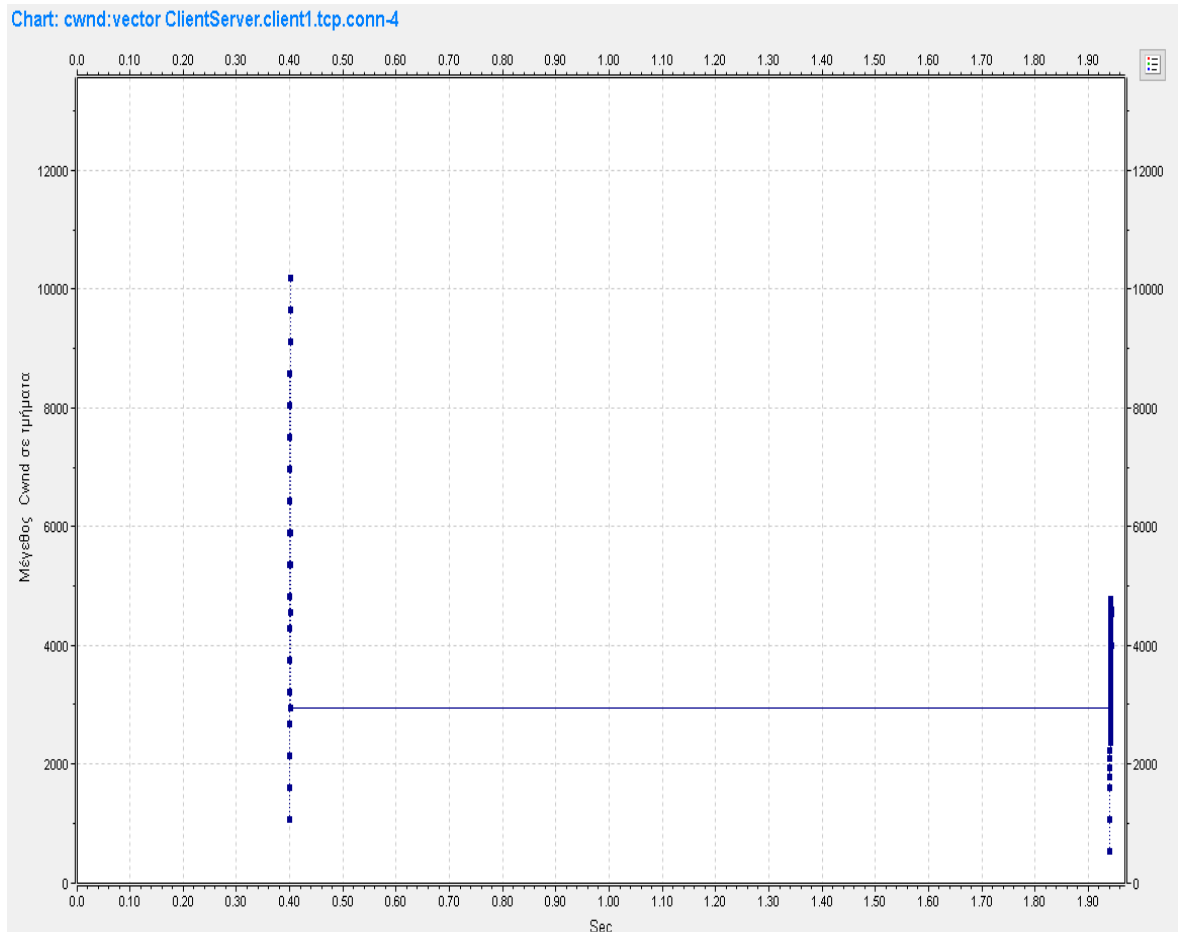


Εικόνα 7.12: Cwnd TCP Vegas

Με μία πρώτη ματιά αντιλαμβανόμαστε ότι στον Vegas το παράθυρο συμφόρησης δεν έχει μεγάλες αυξομειώσεις λόγο όπως αναφεραμε και στην αναλυση του σε προηγούμενη ενότητα, έχει έναν ελεγχόμενο ρυθμό ροής δεδομένων .

TCP Sack

Στον TCP Sack στα χρονικά διαστήματα $t = 0.4$ & $t = 1.94$ αυξάνεται δυο φορές, και ειδικότερα σε $t = 0.4$ φτάνει στο μέγιστο όριο

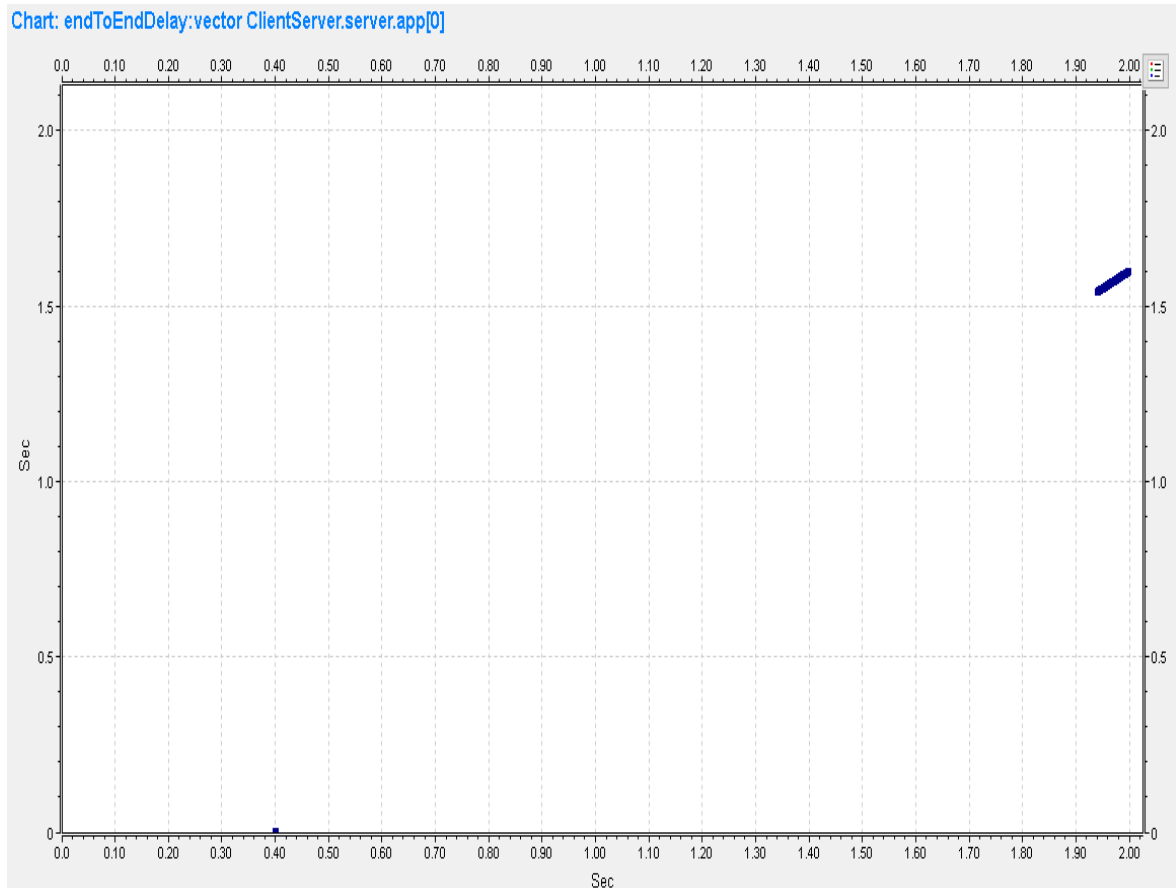


Εικόνα 7.13: Cwnd TCP Sack

7.2.4 Περίπτωση Μελέτης End to End Delay

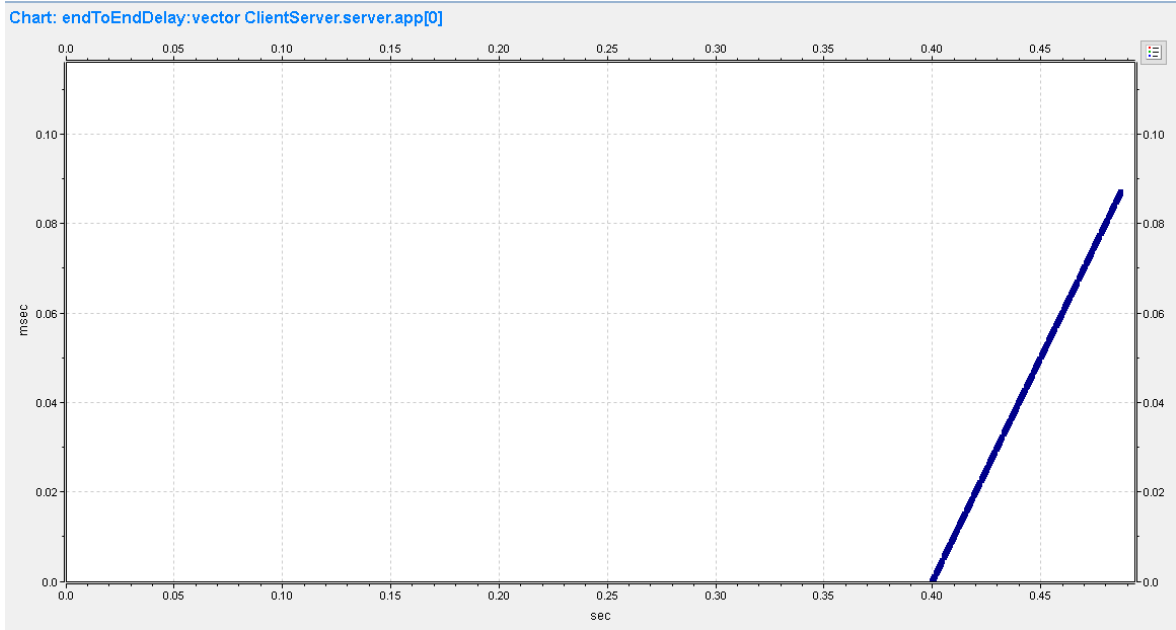
Από αυτή την προσομοίωση θα συγκρινουμε την καθυστέρηση που παρουσιάζουν οι μηχανισμοί από άκρο σε άκρο στην μετάδοση των πακέτων.

TCP Reno



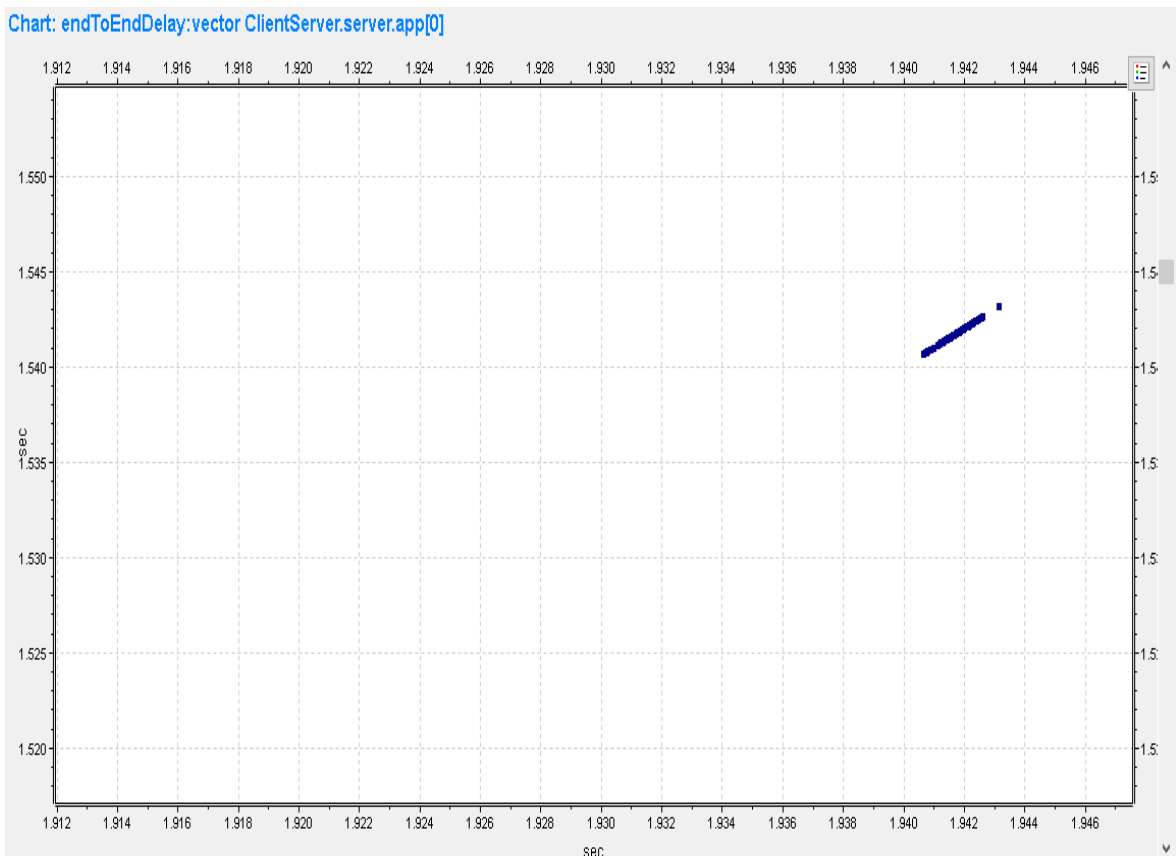
Εικόνα 7.14: End to End Delay TCP Reno

TCP New Reno



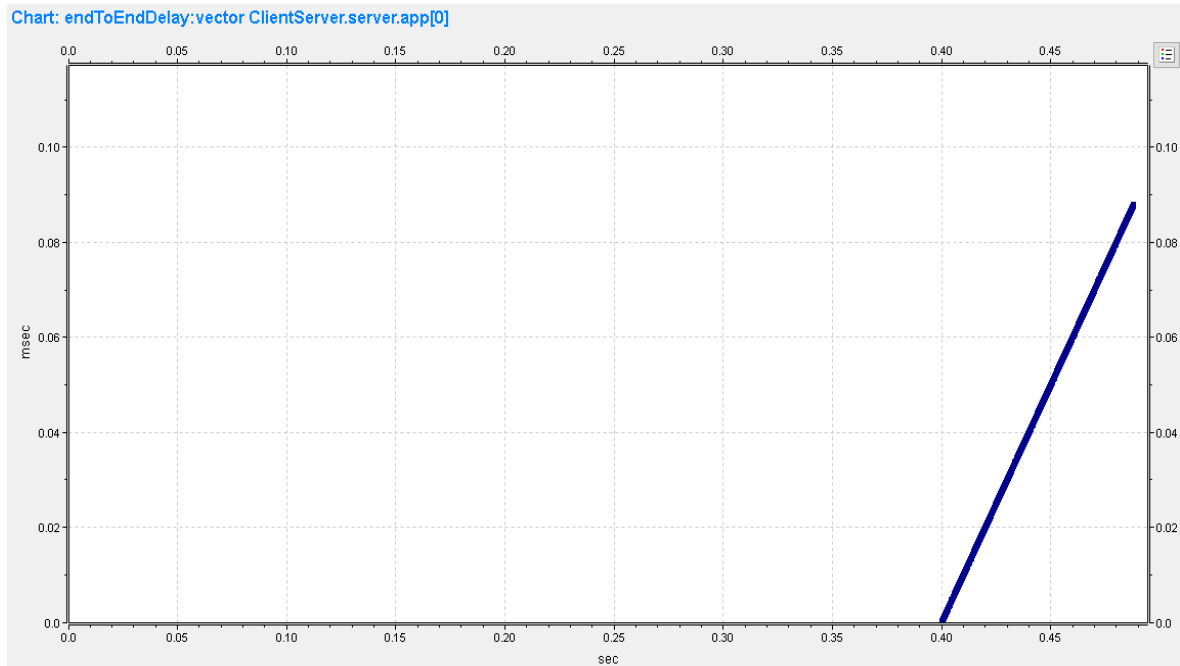
Εικόνα 7.15: End to End Delay TCP New Reno

TCP Sack



Εικόνα 7.16: End to End Delay TCP Sack

TCP Vegas



Εικόνα 7.17: End to End Delay TCP Vegas

Συγκριτικά από τους τέσσερις μηχανισμούς:

Μηχανισμοί	End to End Delay
TCP RENO	0.06 sec.
TCP NEW RENO	0.09 Sec
TCP SACK	0.002 Sec.
TCP VEGAS	0.08 Sec

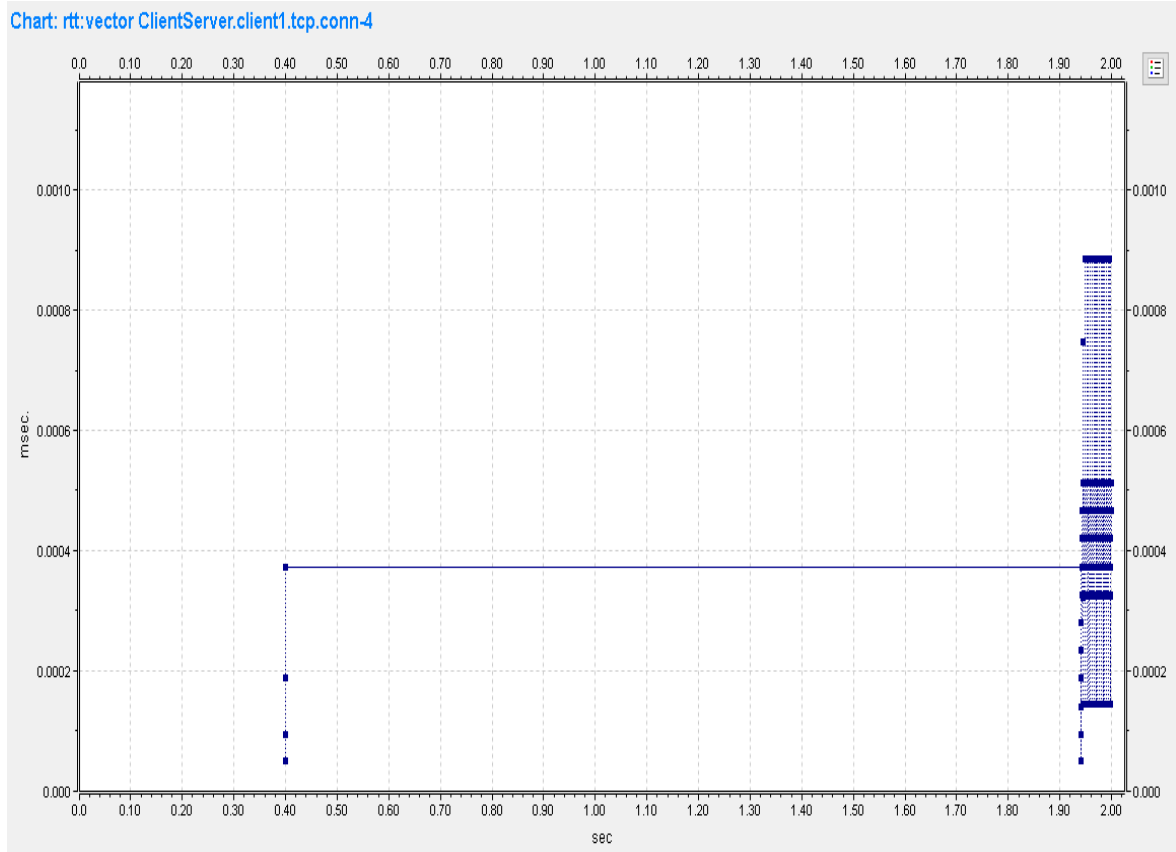
Πίνακας 1: Συγκριτικός των 4 μηχανισμών για End to End Delay

Αυτός που έχει την μικρότερη αναμονή από άκρο σε άκρο είναι ο TCP Sack, και την μεγαλύτερη ο TCP New Reno.

7.2.5 Περίπτωση Μελέτης RTT (Round Trip Time)

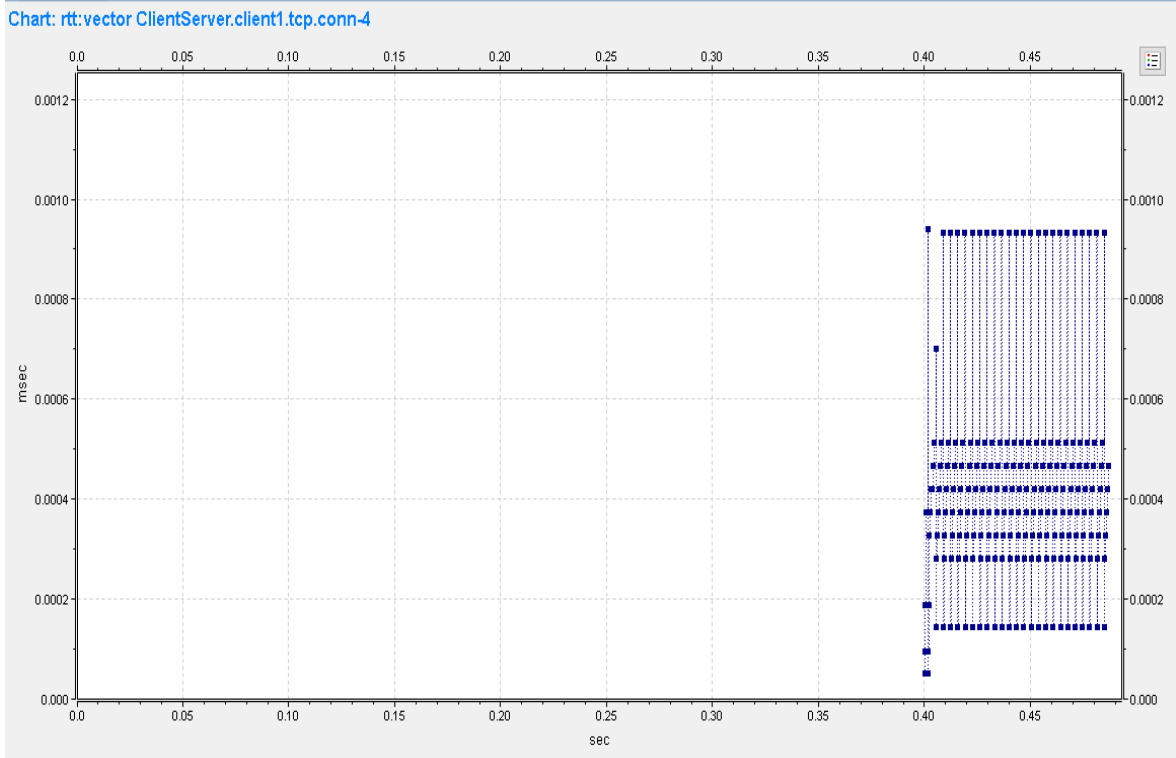
Το RTT, όταν κατά την διάρκεια μίας σύνδεσης είναι ο χρόνος που απαιτείτε όταν στέλνετε ένα πακέτο από τον αποστολέα στον παραλήπτη και η διάρκεια που χρειάζεται για να επιβεβαιωθεί από τον παραλήπτη προς το αποστολέα ότι έλαβε επιτυχώς το πακέτο, και πόσες επιβεβαιώσεις πακέτων πραγματοποιήθηκαν κατά την διάρκεια της σύνδεσης.

TCP Reno



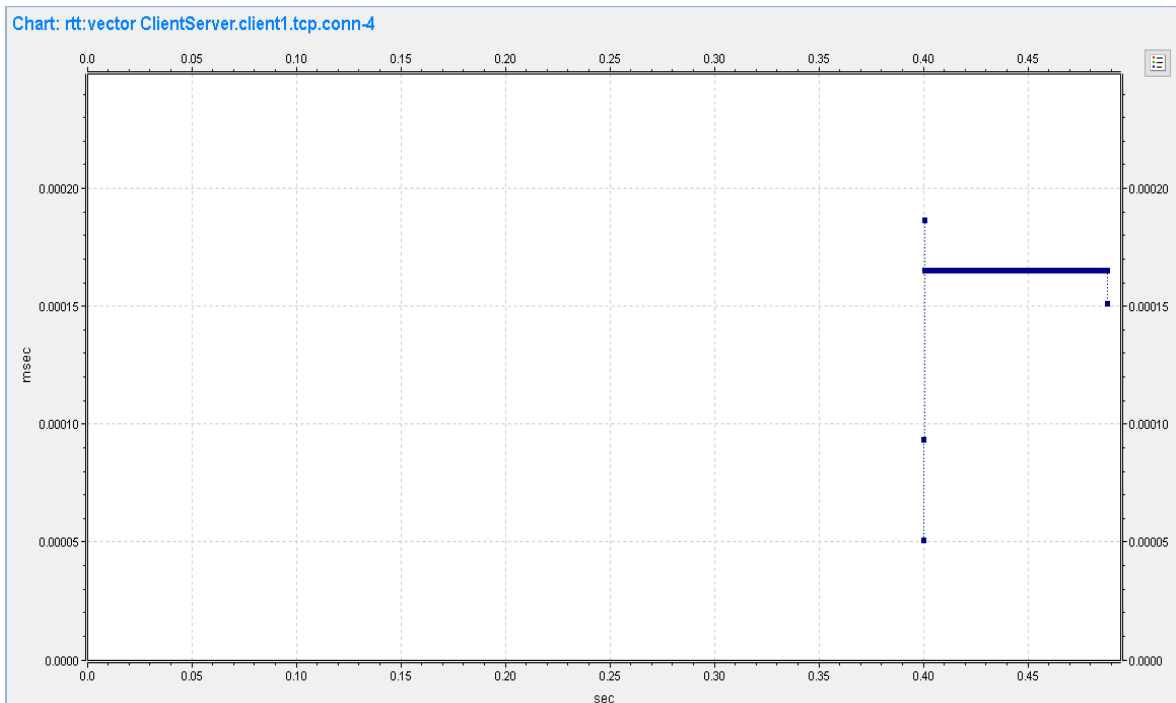
Εικόνα 7.18: RTT TCP Reno

TCP New Reno



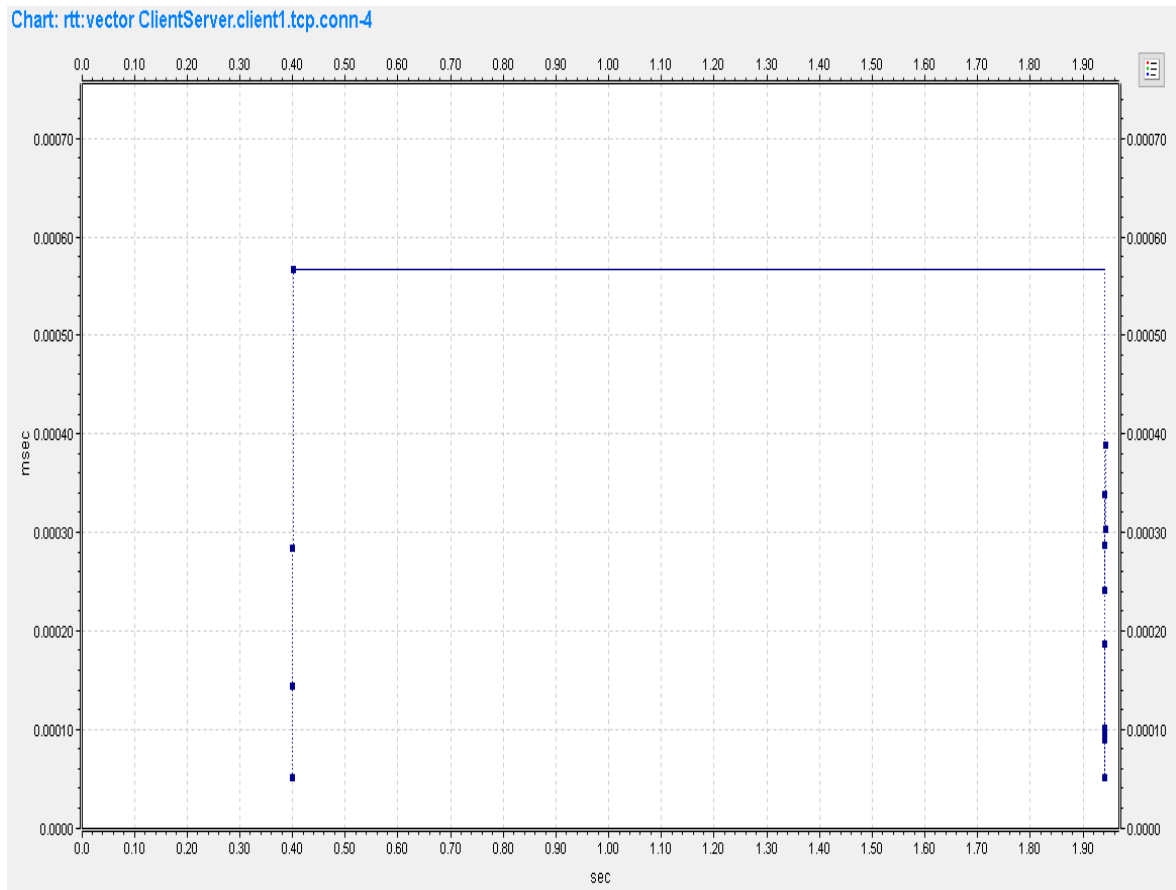
Εικόνα 7.19: RTT TCP New Reno

TCP Vegas



Εικόνα 7.19: RTT TCP Vegas

TCP Sack



Εικόνα 7.20: RTT TCP Sack

Συγκριτικά από τους 4 μηχανισμούς ο TCP Sack παρατηρούμε ότι έχει πολύ καλύτερη διαχείριση στον χρόνο που απαιτείτε για επιβεβαιωθεί το πακέτο στον αποστολέα από τον παραλήπτη λόγω όπως αναφέραμε και στην ανάλυση του ο κύριος μηχανισμός του λειτουργίας του είναι να αναγνωρίζει επιλεκτικά τα πακέτα, έτσι ώστε να μην στέλνονται διπλά η τριπλά

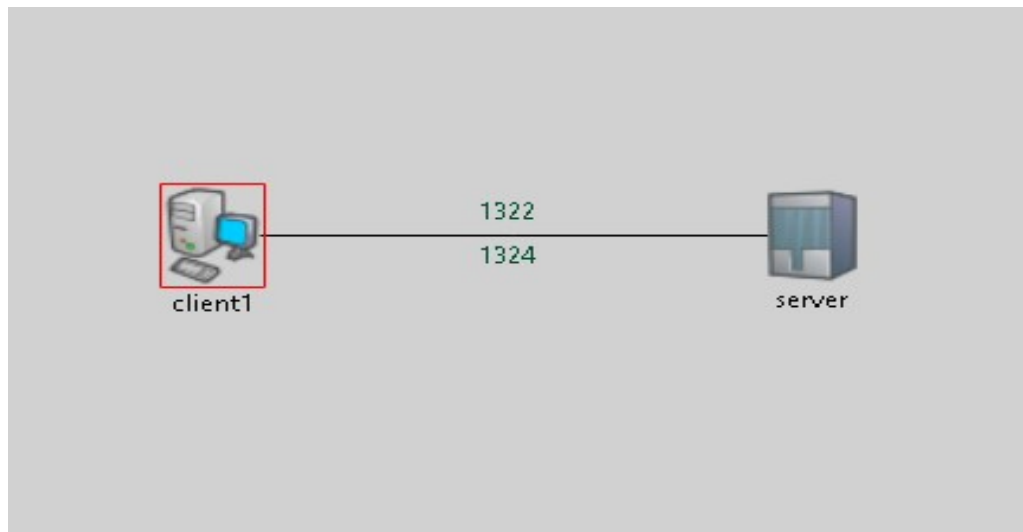
7.2.6 Αριθμός Πακέτων

Με τροποποίηση του κώδικα, και ποιο συγκεκριμένα προσθήκη της γραμμής: **throughputDisplayFormat = " "**; (Ο Κώδικας ολόκληρος υπάρχει στο τέλος της εργασίας (ΠΑΡΑΡΤΗΜΑ Α), μπορούμε να λαμβάνουμε κατά την διάρκεια της προσομοίωσης διαφορά στατιστικά του δικτύου όπως:

- “N”: Αριθμός πακέτων.
- “V”: Όγκος (σε bytes).
- “p”: Τρέχοντα πακέτα/δευτερόλεπτο.
- “b”: Τρέχον εύρος ζώνης.
- “u”: Τρέχον ποσοστό χρήσης καναλιού (%).
- “P”: Μέσος όρος πακέτων/δευτερόλεπτο
- “B”: Μέσος όρος εύρους ζώνης

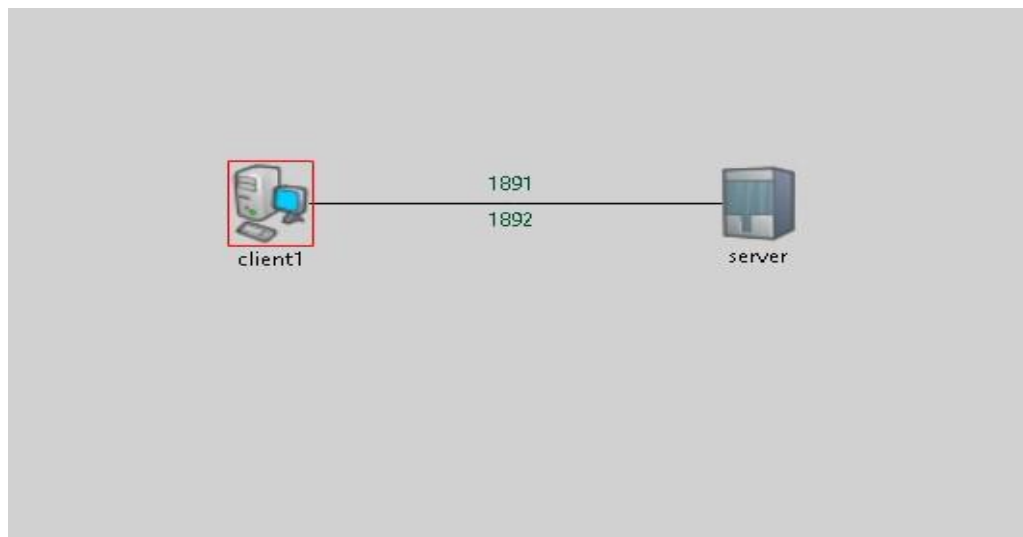
➤ “U”: Μέσος όρος ποσοστού χρήσης καναλιού (%).

- **TCP Reno**



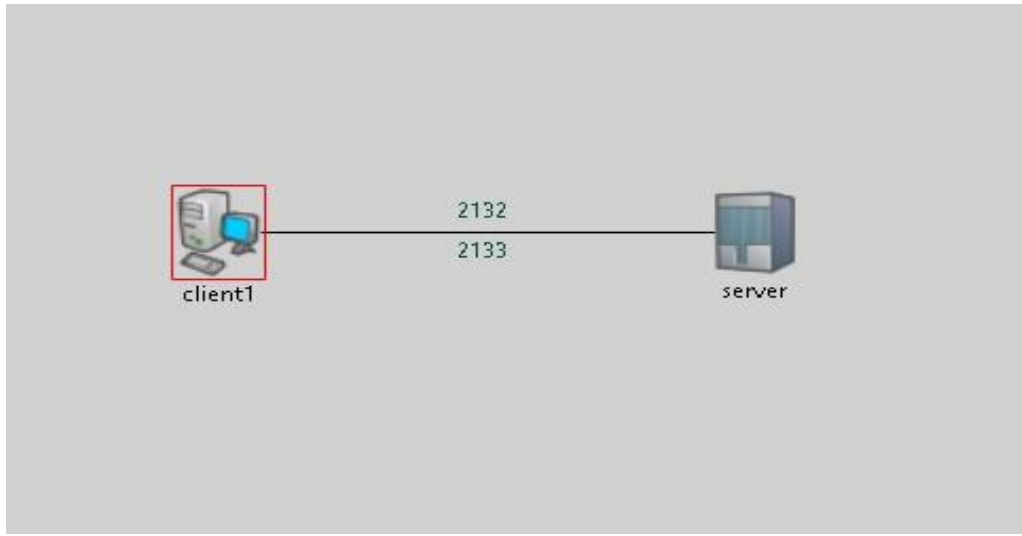
Εικόνα 7.21: TCP Reno

- **TCP New Reno**



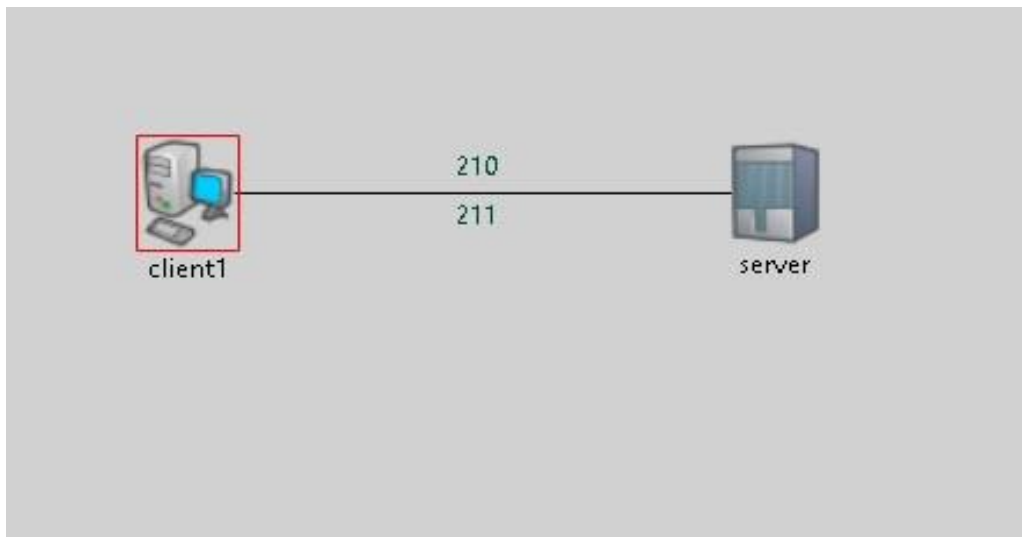
Εικόνα 7.22: TCP New Reno

- **TCP Vegas**



Εικόνα 7.23: TCP Vegas

- **TCP Sack**



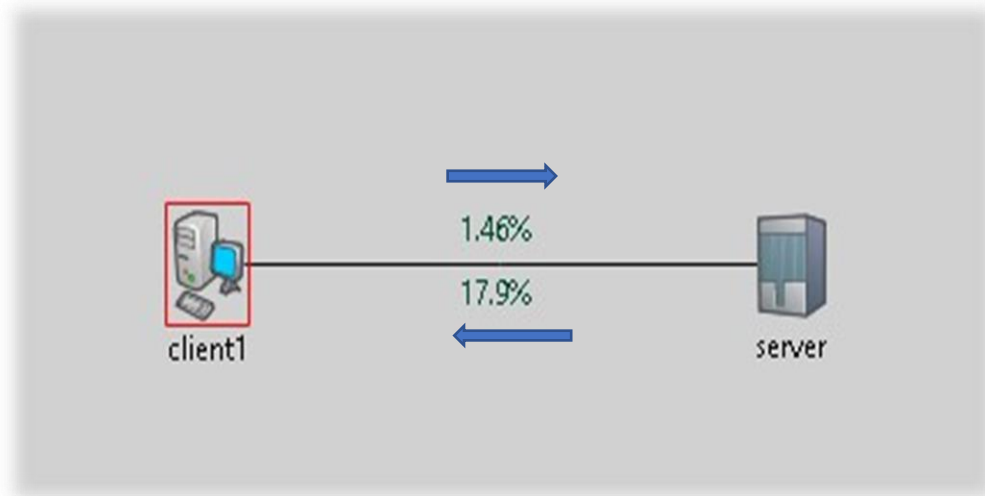
Εικόνα 7.24: TCP Sack

Όπως διαπιστώνεται τα λιγότερα πακέτα που διακινήθηκαν ήταν από το TCP Sack

7.2.7 Ποσοστά χρήσεως γραμμής

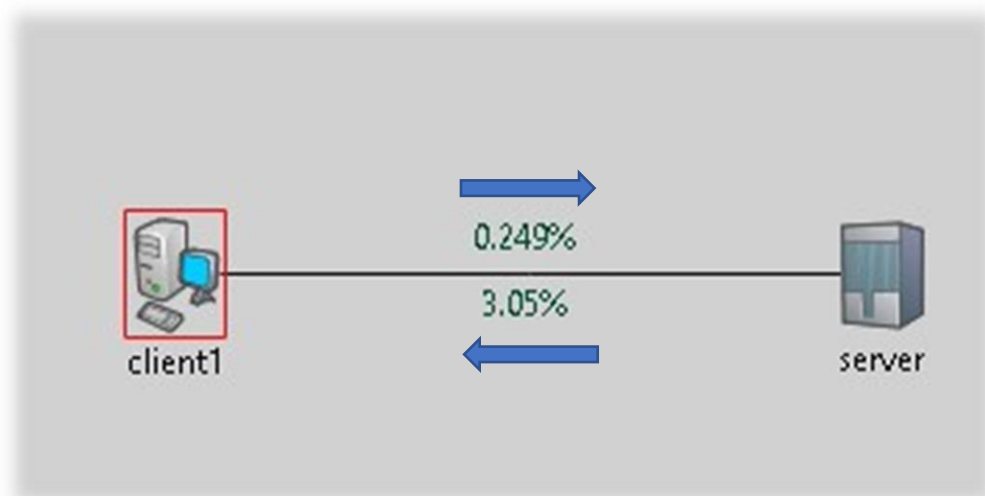
Στα επόμενα γραφήματα απεικονίζονται, ο Μέσος ορός χρήσης των καναλιών (“*U*”), του Δικτύου μεταξύ client- Server.

- **TCP Reno**



Εικόνα 7.25: TCP RENO

- **TCP New Reno**



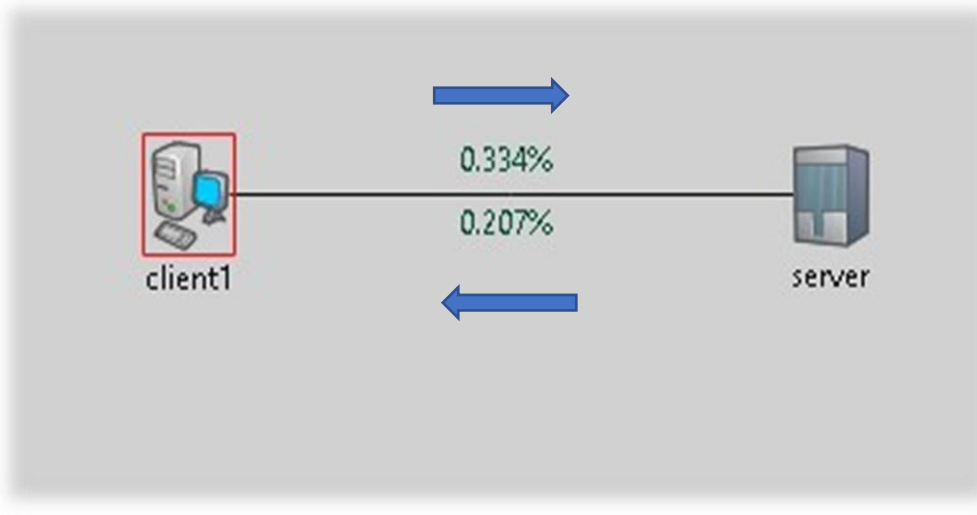
Εικόνα 7.26: TCP NEW RENO

- **TCP Vegas**



Εικόνα 7.28: VEGAS

- **TCP Sack**



Εικόνα 7.27: TCP SACK

Με μία πρώτη ματιά διαπιστώνουμε πως ο TCP Vegas είχε τον μεγαλύτερο μέσο όρο χρήσης της γραμμής,, και ο TCP Sack τον μικρότερο μέσο όρο.

➤ **Από τους τέσσερις μηχανισμούς :**

- Είχε την καλύτερη διαχείριση μεταξύ client – server ήταν ο TCP Sack, & ο Vegas
- Καλύτερο Throughput ο TCP Sack
- Όπως πολύ μικρή χρήση της γραμμής ο TCP Sack
- Πολύ λιγότερη αναμονή από άκρο σε άκρο ο TCP Sack

- Μικρότερο Παράθυρο Συμφόρησης ο TCP Vegas
- Μικρότερο RTT ο TCP Vegas.

9. ΣΥΜΠΕΡΑΣΜΑΤΑ ΜΕΛΕΤΗΣ

Το Πρωτόκολλο TCP όπως και οι παραλλαγές (reno, new reno κ.α.) όπως περιγράψαμε και παραπάνω είναι πρωτοκολλά αξιόπιστης μεταφοράς δεδομένων στο διαδίκτυο, όπως και το βασικό πρωτόκολλο που χρησιμοποιείτε στην πλειοψηφία (TCP) στο διαδίκτυο. Είναι ένα πολυσύνθετο και πολυεπίπεδο πρωτόκολλο. Διέπτετε από πολλές εφαρμογές και κανόνες για στην σωστή λειτουργία του και παραμέτρους επίσης. Για να αποφευχθεί η κατάρρευση η συμφόρηση στο Δίκτυο πρέπει όλα τα παραπάνω να είναι σε πλήρη σκοπιά για την υποστήριξη του. Όπου με τις παραλλαγές του που μελετήσαμε και είδαμε διάφορες καταστάσεις σε περίπτωση που υπάρξει μια συμφόρηση στο δίκτυο όπως τις τεχνικές που χρησιμοποιεί ένας μηχανισμός ελέγχου, για την αποφυγή της συμφόρησης , την αναμονή που έχει από άκρο σε άκρο στην μεταφορά των δεδομένων, όπως επίσης τους χρόνους που χρειάζεται να επιβεβαιώσει ο παραλήπτης στον αποστολέα για την επιτυχή αποστολής των πακέτων.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1.] William Stallings, Επικοινωνίες Υπολογιστών και Δεδομένων, Έτος 2012, 8^η Έκδοση, Εκδόσεις Τζιόλα.
- [2.] Σπυριδούλα Μαργαρίτη – Ελευθέριος Στεργίου, Τοπικά και Αστικά Δίκτυα (Lan-Man), Έτος 2007, Εκδόσεις Νέων Τεχνολογιών.
- [3.] Larry L. Peterson & Bruce S. Davie, Δίκτυα Υπολογιστών, Έτος 2007, 4η Αμερικάνικη Έκδοση, Εκδόσεις Κλειδάριθμος.
- [4.] Douglas E. Comer, Δίκτυα και Διαδίκτυα Υπολογιστών, Έτος 2004, 6η Αμερικάνικη Έκδοση, Εκδόσεις Κλειδάριθμος.
- [5.] Casad Joe, Teach Yourself TCP/IP in 24 Hours, Έτος 2004, Εκδόσεις Γκιούρδας Μ.
- [6.] J. Postel, Internet Standard, RFC 768, Διαθέσιμο: <https://datatracker.ietf.org/doc/html/rfc768>.
- [7.] R. Stewart, Proposed Standard, RFC 4960, Διαθέσιμο: <https://datatracker.ietf.org/doc/html/rfc4960>.
- [8.] E. Kohler, M. Handley, S. Floyd, Proposed Standard, RFC 4340, Διαθέσιμο: <https://datatracker.ietf.org/doc/html/rfc4340>.
- [9.] Jon Postel, Internet Standard, RFC 793, Διαθέσιμο: <https://datatracker.ietf.org/doc/html/rfc793>.
- [10.] M. Larsen, Tieto, F. Gont, Best Current Practice, RFC 6056, Διαθέσιμο: <https://datatracker.ietf.org/doc/html/rfc6056>.
- [11.] Ελευθέριος Στεργίου, ΠΡΩΤΟΚΟΛΛΑ & ΑΡΧΙΤΕΚΤΟΝΙΚΕΣ ΔΙΚΤΥΩΝ, Διαθέσιμο: <https://www.dit.uoi.gr/e->

class/modules/document/file.php/120/%CE%98%CE%B5%CF%89%CF%81%CE%AF%CE%B1%20%28Presentations%29/Chapter_8-TCP.pd.

- [12.] TCP Sliding Window, Διαθέσιμο:
http://www.tcpipguide.com/free/t_TCPSlidingWindowAcknowledgmentSystemForDataTranspo-8.htm,
- [13.] Γιάννης Αγγελόπουλος, ΤΟ ΠΡΩΤΟΚΟΛΛΟ TCP, (ΤΕΙ Πειραιά - Τμήμα Αυτοματισμού), Διαθέσιμο: <https://docplayer.gr/2033218-To-protokollo-tcp-transmission-control-protocol.html>.
- [14.] Ελευθέριος Στεργίου, Έλεγχος συμφόρησης με το Πρωτόκολλο TCP, (Σημειώσεις Μαθήματος Πρωτόκολλα και Αρχιτεκτονικές Δικτύων Αρτα 2020).
- [15.] Χαράλαμπος Ζ. Πατρικάκης, Δίκτυα Υπολογιστών- Ενότητα 4: Μετάδοση δεδομένων από το στρώμα μεταφοράς, Διαθέσιμο:
<http://eclass.teipir.gr/openeclass/modules/document/file.php/ENGI100/01%20%CE%94%CE%B9%CE%B1%CF%86%CE%AC%CE%BD%CE%B5%CE%B9%CE%B5%CF%82%20%CE%9C%CE%B1%CE%B8%CE%AE%CE%BC%CE%B1%CF%84%CE%BF%CF%82/%CE%95%CE%BD%CF%8C%CF%84%CE%B7%CF%84%CE%B1%204.pdf>.
- [16.] W. Stevens, Proposed Standard,RFC 2001, Διαθέσιμο:
<https://datatracker.ietf.org/doc/html/rfc2001>.
- [17.] Σταύρος Βαλσαμίδης, Σωτήριος Κοντογιάννης Προσδιορισμός της TCP συμπεριφοράς των web εξυπηρετητών, Διαθέσιμο:
<https://spooky.math.uoi.gr/~skontog/impress/tbit-presentation.pdf>.
- [18.] Yueqiu JIANG, Junkun ZHANG, Qixue GUAN, Improvement of TCP Reno Congestion Control Protocol, 2014 by IFSA Publishing, S. L., Διαθέσιμο:
https://www.researchgate.net/publication/288250636_Improvement_of_TCP_Reno_Congestion_Control_Protocol.
- [19.] Ghassan A. Abed, Mahamod Ismail and Kasmiran Jumari, A Survey on Performance of Congestion Control Mechanisms for Standard TCP Versions, Διαθέσιμο:
https://www.researchgate.net/publication/256868797_A_Survey_on_Performance_of_Congestion_Control_Mechanisms_for_Standard_TCP_Versions.
- [20.] Dhananjay Bisen1 and Sanjeev Sharma, IMPROVE PERFORMANCE OF TCP NEW RENO OVER MOBILE AD-HOC NETWORK USING ABRA. Διαθέσιμο: <https://arxiv.org/ftp/arxiv/papers/1105/1105.0251.pdf> .

- [21.] Harjinder Kaur and Dr. Gurpreet Singh, TCP Congestion Control and Its Variants, Διαθέσιμο: https://www.ripublication.com/acst17/acstv10n6_17.pdf
- [22.] E. Blanton, M. Allman, L. Wang, I. Jarvinen, M. Kojo, Y. Nishida, Conservative Loss Recovery Algorithm Based on, Selective Acknowledgment (SACK) for TCP, RFC 3517, Διαθέσιμο: <https://datatracker.ietf.org/doc/html/rfc6675>

Αναφορές Εικόνων

1. Εικόνα 1.1: Networks and Devices. Διαθέσιμη: <https://baictchs09.wordpress.com/2018/10/15/network/>
2. Εικόνα 1.2: Δίκτυο Η/Υ και διαφόρων μέσων υλικού, Διαθέσιμη: https://commons.wikimedia.org/wiki/File:Barcode_System_Diagram.jpeg
3. Εικόνα 1.3: Διάφορα Δομικά στοιχεία και συσκευές επικοινωνίας ενός Δικτύου, Διαθέσιμη: https://www.flashmemorysummit.com/English/Collaterals/Proceedings/2016/20160808_PreConfD_Davis.pdf
4. Εικόνα 1.4: Η Επικοινωνία δύο σταθμών Ηλεκτρονικών Υπολογιστών, Διαθέσιμη: <https://kokkalidis.files.wordpress.com/2015/11/6-data-communication-model.pdf>
5. Εικόνα 2.1: Τα επίπεδα του OSI, Διαθέσιμη: Larry L. Peterson & Bruce S. Davie, Δικτυα Υπολογιστών, Έτος 2007, 4η Αμερικάνικη Έκδοση, Εκδόσεις Κλειδάριθμος.
6. Εικόνα 2.2: Βασικά πρωτόκολλα του TCP/IP ανά επίπεδο, Διαθέσιμη: https://codebrakes.gr/tutorials/net_tutorial_3.html
7. Εικόνα 2.3 Σύγκριση TCP/IP και OSI, Διαθέσιμη: <https://alldifferences.net/difference-between-tcp-ip-and-osi-model/>
8. Εικόνα 2.4: Σύνδεση TCP/IP, Διαθέσιμη: https://codebrakes.gr/tutorials/net_tutorial_13.html
9. Εικόνα 2.5: Αναλυτική απεικόνιση δομής μιας IP διεύθυνσης, Διαθέσιμη: https://codebrakes.gr/tutorials/net_tutorial_3.html
10. Εικόνα 2.6: Datagram IPv4, Διαθέσιμη: <https://support.huawei.com/enterprise/en/doc/EDOC1000178017/dd76ea1f/ipv4-packet-format>
11. Εικόνα 3.2: TCP Επικεφαλίδα, Διαθέσιμη: Ελευθέριος Στεργίου, TCP header, (Σημειώσεις Μαθήματος Πρωτόκολλα και Αρχιτεκτονικές Δικτύων Άρτα 2020).
12. Εικόνα 3.3: Σημαίες Ελέγχου (Flags), Διαθέσιμη: Ελευθέριος Στεργίου, TCP header, (Σημειώσεις Μαθήματος Πρωτόκολλα και Αρχιτεκτονικές Δικτύων Άρτα 2020).
13. Εικόνα 3.4: Η επικοινωνία δύο Η/Υ, με βάση τις θύρες(Ports), Διαθέσιμη: <https://slideplayer.com/slide/12713895/>

14. Εικόνα 3.5: Διάφορα Συνήθη Ports, Διαθέσιμη: Ελευθέριος Στεργίου, TCP, UDP Πρωτόκολλα, (Σημειώσεις Μαθήματος Πρωτόκολλα και Αρχιτεκτονικές Δικτύων Άρτα 2020
15. Εικόνα 3.6: Εγκατάσταση Σύνδεσης, Διαθέσιμη: Ελευθέριος Στεργίου, TCP, UDP Πρωτόκολλα, (Σημειώσεις Μαθήματος Πρωτόκολλα και Αρχιτεκτονικές Δικτύων Άρτα 2020
16. Εικόνα 3.7: Τερματισμός Σύνδεσης, Διαθέσιμη: Ελευθέριος Στεργίου, TCP, UDP Πρωτόκολλα, (Σημειώσεις Μαθήματος Πρωτόκολλα και Αρχιτεκτονικές Δικτύων Άρτα 2020
17. Εικόνα 3.8: TCP Duplicate / Selective Acknowledgments, Διαθέσιμη: <https://accedian.com/blog/network-packet-loss-retransmissions-and-duplicate-acknowledgements/>
18. Εικόνα 3.11: Λήξη χρονομετρητή και Επαναμετάδοση, Διαθέσιμη: Ιωάννης Κ. Πλαστήρας, ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ Προσομοιώσεις Δικτύων με Χρήση του OMNeT++, <https://pergamos.lib.uoa.gr/uoa/dl/frontend/file/lib/default/data/1324338/theFile>
19. Εικόνα 3.12: Χρονιστές – Timers, Διαθέσιμη: <https://www.gatevidyalay.com/tcp-timers-transmission-control-protocol/>
20. Εικόνα 4.1: Αλγόριθμος "Αργής Εκκίνησης – Slow Start TCP", Διαθέσιμη: <https://www.keycdn.com/support/tcp-slow-start>
21. Εικόνα 4.2: Η μεταβολή του TCP παραθύρου συμφόρησης, Διαθέσιμη: Ελευθέριος Στεργίου, TCP, UDP Πρωτόκολλα, (Σημειώσεις Μαθήματος Πρωτόκολλα και Αρχιτεκτονικές Δικτύων Άρτα 2020
22. Εικόνα 5.2: Εξέλιξη του μεγέθους παραθύρου στο TCP Reno, Διαθέσιμη: https://www.researchgate.net/figure/Evolution-of-the-window-size-in-TCP-Reno-It-consists-of-slow-start-congestion_fig8_313751318
23. Εικόνα 5.3: Συμπεριφορά του TCP Reno, Διαθέσιμη: Harjinder Kaur and Dr. Gurpreet Singh, TCP Congestion Control and Its Variants, Διαθέσιμο: https://www.ripublication.com/acst17/acstv10n6_17.pdf
24. Εικόνα 5.4: TCP Vegas Ρυθμός Ροής, Διαθέσιμη: https://www.researchgate.net/publication/256868797_A_Survey_on_Performance_of_Congestion_Control_Mechanisms_for_Standard_TCP_Versions
25. Εικόνα 5.5: Επιλεκτική αναγνώριση πακέτων, Διαθέσιμη: <https://iponwire.com/tcp-sack-in-depth-review>

ΠΑΡΑΡΤΗΜΑ Α: Κώδικας Client – Server

package inet.examples.inet.tcpclientserver;

```

import inet.common.misc.NetAnimTrace;
import inet.networklayer.configurator.ipv4.Ipv4NetworkConfigurator;
import inet.node.inet.StandardHost;
import ned.DatarateChannel;
import inet.common.misc.ThruputMeteringChannel; // Αυτή η γραμμή προστέθηκε για τον
υπολογισμό του Throughput, τον υπολογισμό δηλαδή της διακίνησης των δεδομένων στο
δίκτυο

```

```

network ClientServer

```

```

{
  parameters:
  double per = default(0);
  types:
  channel C extends ThruputMeteringChannel
  {
    delay = 0.1us;
    datarate = 100Mbps; // Η αρχική γραμμή ήταν στα 10 Mbps.
    per = per;
    thruputDisplayFormat = "U"; // Μέσος όρος ποσοστού χρήσης καναλιού (%).
  }
  submodules:
  client1: StandardHost {
    parameters:
    @display("p=250,150;i=device/pc3");
  }
  server: StandardHost {
    parameters:
    @display("p=400,150;i=device/pc2");
  }
  configurator: Ipv4NetworkConfigurator {
    parameters:
    @display("p=100,100;is=s");
  }
}

```

```

}
netAnimTrace: NetAnimTrace {
  @display("p=100,200;is=s");
}
connections:
client1.pppg++ <--> C <--> server.pppg++;
}

```