Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης

Σχολή Τεχνολογικών Εφαρμογών
Τμήμα Εφαρμοσμένης Πληροφορικής και Πολυμέσων



Πτυχιακή Εργασία

Τίτλος: **Μελέτη μεθόδων ενεργών μετρήσεων για δίκτυα πολυμεσικών υπηρεσιών**

Κανδηλιεράκης Παύλος (ΑΜ:2154)

Επιβλέπων Καθηγητής: Πάλλης Ευάγγελος

Ευχαριστίες

Σε αυτό το σημείο θα ήθελα να ευχαριστήσω τους καθηγητές κ. Γαρδίκη Γεώργιο για την άψογη συνεργασία που είχα μαζί του μέχρι την ολοκλήρωσή της εργασίας μου ακόμα και μετά την αποχώρησή του από το ΤΕΙ καθώς και τον κ. Πάλλη Ευάγγελο που ανέλαβε την εργασία μου την τελευταία στιγμή. Επίσης θα ήθελα να ευχαριστήσω την οικογένεια μου που με στήριξε καθώς και την φίλη μου Ειρήνη για την πολύτιμη βοήθειά της στην εργασία.

Abstract

The aim of the thesis is to study, design and development tool for implementing active measurements for multimedia networks. Studied existing architectures and functions of specific tools. Through this study identified the need for measurements.

More specifically, code system OWAMP materialized from a university in Italy, which is based on java programming language and can be used on both Windows and Linux platforms, requiring only the installation of J2SE Java Runtime Environment (JRE), available for both Windows and Linux. To expand the system as a result of specific requirements, XML technology used to extract the results.

Thus, the system developed, the central issue of active measurements in multimedia networks, enables a user to perform a series of test measurements on a real network and obtain performance assessment results as delay, loss, duplication and performance.

Σύνοψη

Σκοπός της πτυχιακής αυτής είναι η μελέτη, σχεδίαση και ανάπτυξη εργαλείου για την υλοποίηση ενεργών μετρήσεων για δίκτυα πολυμεσικών υπηρεσιών. Μελετήθηκαν υπάρχουσες αρχιτεκτονικές και λειτουργίες συγκεκριμένων εργαλείων. Μέσα από τη μελέτη αυτή καθορίστηκαν οι ανάγκες για τις μετρήσεις.

Πιο συγκεκριμένα, χρησιμοποιήθηκε ο κώδικας του συστήματος OWAMP υλοποιημένος από πανεπιστήμιο της Ιταλίας, ο οποίος βασίζεται στη γλώσσα προγραμματισμού java και μπορεί να χρησιμοποιηθεί τόσο σε Windows όσο και Linux πλατφόρμες, απαιτώντας μόνο την εγκατάσταση του J2SE Java Runtime Environment (JRE), διαθέσιμο και για τα δύο Windows και Linux. Για την επέκταση του συστήματος, όπως προέκυψε από συγκεκριμένες απαιτήσεις, χρησιμοποιήθηκε XML τεχνολογία για την εξαγωγή των αποτελεσμάτων.

Έτσι το σύστημα που αναπτύχθηκε, με κεντρικό θέμα τις ενεργές μετρήσεις σε δίκτυα πολυμεσικών υπηρεσιών, δίνει τη δυνατότητα σε ένα χρήστη να εκτελέσει μια σειρά από δοκιμαστικές μετρήσεις πάνω σε πραγματικό δίκτυο και να λάβει αποτελέσματα απόδοσης εκτίμησης όπως καθυστέρηση, απώλειες, επικαλύψεις και απόδοση.

Περιεχόμενα

1. Εισαγωγή 9
1.1 Internet Protocol Suite 9
1.1.1 Επίπεδα της Σουίτας TCP/IP 9
1.1.2 Ανάλυση των επιπέδων 10
Εφαρμογής 10
1.1.3 Υλοποιήσεις 12
1.2 OSI Model 12
1.3 Πρωτόκολλο Internet & IPv4 vs. IPv6 14
1.3.1 Κύρια χαρακτηριστικά του Internet Protocol 15
1.3.2 Τι είναι το best effort delivery; 15
1.3.3 Γιατί να σχεδιαστεί ένα πρωτόκολλο αναξιόπιστο; 15
1.4 IPv4 vs. IPv6 15
1.5 Πρωτόκολλα TCP, UDP και RTP 16
1.5.1 Πρωτόκολλο TCP 16
1.5.2 Πρωτόκολλο UDP 17
1.5.3 TCP vs. UDP 18
1.5.4 Γιατί υπάρχει το UDP; Τι εξυπηρετεί; 19
1.5.5 Πρωτόκολλο RTP 19
1.6 Μέθοδοι δρομολόγησης (Routing Schemes) 20
1.6.1 Unicast 20
1.6.2 Broadcast 20
1.6.3 Multicast 21
1.6.4 Anycast 21
2. Μετρήσεις δικτύων 23
2.1 Διεύθυνση αποστολέα/ παραλήπτη 23
2.2 Ώρα αποστολής/λήψης 23
2.3 Αριθμός πακέτων που στάλθηκαν, αριθμός πακέτων που απορρίφθηκαν, αριθμός πακέτων που ελήφθησαν, αριθμός πακέτων που ελήφθησαν μετά τη λήξη του χρόνου, συνολικός αριθμός χαμένων πακέτων 23
2.4 Καθυστέρηση Δικτύου – delay 23
2.5 Παραμόρφωση χρονισμού (Jitter) 23
2.6 Throughput 24
2.6.1 Η μέγιστη απόδοση 24
2.6.2 Αξιοποίηση καναλιού - αποδοτικότητα καναλιού - Κανονικοποιημένη απόδοση 25
2.6.3 Goodput 26
3 Εργαλεία ενεργών μετρήσεων 27
3.1 perfSONAR 27
3.2 Bandwidth Test Controller (BWCTL) 27
3.2.1 Iperf 28
3.2.2 Thrulay 28
3.2.3 Nuttcp 29
3.3 Network Diagnostic Tool (NDT) 29
3.4 pS-Performance Toolkit 30
4. OWAMP 31
4.1 Χαρακτηριστικά 33
4.2 Απαιτήσεις 33
4.3 Σχέση της δοκιμής και τα πρωτόκολλα ελέγχου 33
4.4 Λογικό Μοντέλο 34
4.5 Επισκόπηση Πρωτοκόλλου 35
4.6 OWAMP-Control 35

5 Αναλυτική παρουσίαση εφαρμογής 39
Βιβλιογραφία 60
Παράρτημα 61
1 Εισαγωγή και απαιτήσεις 61
2 Αρχιτεκτονική συστήματος 62
3 Εκτέλεση J-OWAMP 64
3.1 OWAMP-Server: 64
3.2 OWAMP Session-Sender: 64
3.3 OWAMP Session-Receiver: 65
4 Διαμόρφωση J-OWAMP ενοτήτων 67
4.1 OWAMP Server, OWAMP Session-Sender and OWAMP Session-Receiver 67
4.2 OWAMP Control-Client 67

Κατάλογος εικόνων

Εικόνα 1: Πάνω:Δύο συσκευές συνδεδεμένες μεταξύ τους μέσω δρομολογητών. Κάτω:Η ροή των δεδομένων ανάμεσα στα διάφορα επίπεδα, της Σουίτας TCP/IP. 8
Εικόνα 2: Διαφορές στο header των πακέτων των δύο πρωτοκόλλων 15
Εικόνα 3: Έναρξη της σύνδεσης με three-way handshake 16
Εικόνα 4: Παράδειγμα ενθυλάκωσης δεδομένων σ΄ένα διάγραμμα UDP μέσα σε πακέτο IP 17
Εικόνα 5: Unicast μετάδοση 19
Εικόνα 6: Broadcast μετάδοση 20
Εικόνα 7: Multicast μετάδοση 20
Εικόνα 8: Anycast μετάδοση 21
Εικόνα 9: perfSONAR 26
Εικόνα 10: Bandwidth Test Controller (BWCTL) 26
Εικόνα 11: Iperf 27
Εικόνα 12: pS-Performance Toolkit 29
Εικόνα 13: Ένα πιθανό σενάριο των σχέσεων μεταξύ αυτών των ρόλων 33
Εικόνα 14: Απλουστευμένο σενάριο των σχέσεων μεταξύ αυτών των ρόλων 33
Εικόνα 15: Μήνυμα χαιρετισμού του διακομιστή 34
Εικόνα 16: απαντητικό μήνυμα πελάτη 35
Εικόνα 17: Μήνυμα εκκίνησης διακομιστή 36
Εικόνα 18: Κώδικας για το Control Client (κομμάτι 1ο) 38
Εικόνα 19: Κώδικας για το Control Client (κομμάτι 2ο) 39
Εικόνα 20: Κώδικας για το Control Client (κομμάτι 3ο) 40
Εικόνα 21: Κώδικας για το Control Client (κομμάτι 4ο) 41
Εικόνα 22: Κώδικας για το Control Client (κομμάτι 5ο) 42
Εικόνα 23: Κώδικας για το Control Client (κομμάτι 6ο) 43
Εικόνα 24: Κώδικας για το Control Client (κομμάτι 7ο) 44
Εικόνα 25: Κώδικας για το Control Client (κομμάτι 8ο) 45
Εικόνα 26: Κώδικας για το Control Client (κομμάτι 9ο) 46
Εικόνα 27: Κώδικας για το Control Client (κομμάτι 10ο) 47
Εικόνα 28: Κώδικας για το Control Client (κομμάτι 11ο) 48
Εικόνα 29: Κώδικας για το Control Client (κομμάτι 11ο) 49
Εικόνα 30: Κώδικας για το Control Client (κομμάτι 12ο) 50
Εικόνα 31: Κώδικας για το Control Client (κομμάτι 13ο) 51
Εικόνα 32: Κώδικας για το Fetch Client (κομμάτι 1ο) 52
Εικόνα 33: Κώδικας για το Fetch Client (κομμάτι 2ο) 53
Εικόνα 34: Κώδικας για το Fetch Client (κομμάτι 3ο) 54
Εικόνα 35: Κώδικας για το Fetch Client (κομμάτι 4ο) 55
Εικόνα 36: xml αρχείο με αποτελέσματα 56
Εικόνα 37: OWAMP αρχιτεκτονική 59
Εικόνα 38: OWAMP απλοποιημένη αρχιτεκτονική 59
Εικόνα 39: Window illustrating the execution of OWAMP_Server.exe 64
Εικόνα 40: Window illustrating the execution of OWAMP_SessionSender.exe 64
Εικόνα 41: Window illustrating the execution of OWAMP_SessionReceiver.exe 64
Εικόνα 42: Window illustrating the execution of OWAMP_ControlClient.exe 65

Ευρετήριο πινάκων

Πίνακας 1: Μοντέλο OSI.....9
Πίνακας 2: Μοντέλο TCP/IP 9
Πίνακας 3: Βασικές διαφορές των δύο πρωτοκόλλων 17
Πίνακας 4: Αριθμοί θυρών που χρησιμοποιούνται σε κάθε στοιχείο για TCP συνδέσεις 61

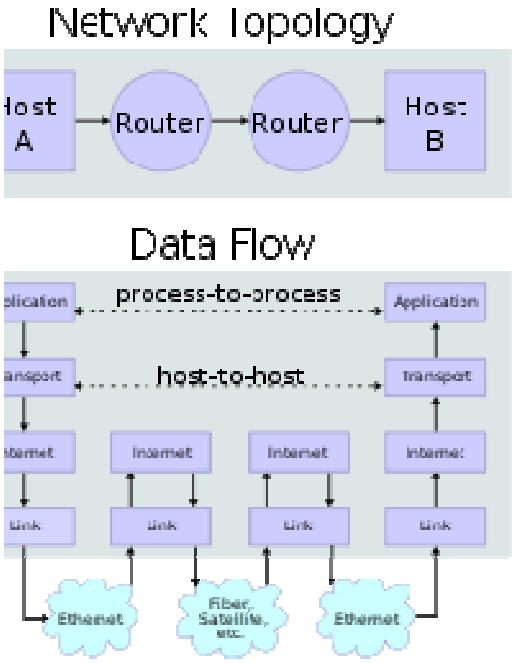
1. Εισαγωγή

Σε αυτήν την ενότητα θα παρουσιάσουμε κάποιες απαραίτητες έννοιες, σχετικές με τα δίκτυα υπολογιστών και τη μετάδοση δεδομένων σε αυτά, που θα βοηθήσουν περαιτέρω τόσο στην κατανόηση του πως οι υπολογιστές επικοινωνούν μεταξύ τους, όσο και στη λειτουργία της εφαρμογής μας.

Αρχικά γίνεται μια σύντομη παρουσίαση δύο βασικών εννοιών που χρησιμοποιούνται σήμερα στην επικοινωνία των υπολογιστών, της Internet Protocol Suite και του μοντέλου OSI.

1.1 Internet Protocol Suite

Το Internet Protocol Suite είναι ένα σύνολο πρωτοκόλλων επικοινωνίας που απαρτίζουν ουσιαστικά το Διαδίκτυο (Internet) και τα περισσότερα εμπορικά δίκτυα. Αποκαλείται πολλές φορές και TCP/IP Protocol Suite, παίρνοντας το όνομα αυτό από τα δύο σημαντικότερα πρωτόκολλα που περιέχει: το Transmission Control Protocol (TCP) και το Internet Protocol (IP), τα οποία ήταν τα πρώτα πρωτόκολλα που καθορίστηκαν.



1: Πάνω:Δύο συσκευές συνδεδεμένες τους μέσω δρομολογητών. Κάτω:Η ροή δομένων ανάμεσα στα διάφορα επίπεδα, ίτας TCP/IP.

Το Internet Protocol Suite μπορεί να θεωρηθεί ως ένα σύνολο τεσσάρων επιπέδων (layers) που το καθένα λύνει ένα σύνολο προβλημάτων που εμφανίζονται στη μετάδοση δεδομένων, και παρέχει μια, καθορισμένη με σαφήνεια, υπηρεσία στα πρωτόκολλα ανώτερου επιπέδου. Τα ανώτερα επίπεδα είναι πιο "κοντά" στον άνθρωπο-χρήστη, ασχολούνται με πιο αφηρημένα στοιχεία και στηρίζονται στα χαμηλότερα επίπεδα για να μετατρέψουν τα στοιχεία αυτά στις φυσικές μορφές που μπορούν τελικά να μεταδοθούν μέσα από τα δίκτυα.

1.1.1 Επίπεδα της Σουίτας TCP/IP

Τα πρωτόκολλα Διαδικτύου κάνουν χρήση της ενθυλάκωσης (encapsulation) για να παρέχουν γενικά πρωτόκολλα και υπηρεσίες. Ένα πρωτόκολλο υψηλού στρώματος χρησιμοποιεί τα πρωτόκολλα των κατώτερων για να λειτουργήσει.

Το παρακάτω σχεδιάγραμμα τοποθετεί τα διάφορα πρωτόκολλα του TCP/IP με βάση τα κριτήρια του μοντέλου OSI:

7	Εφαρμογής (Application)	π.χ. HTTP, SMTP, SNMP, FTP, Telnet, NFS
6	Παρουσίασης (Presentation)	π.χ. XDR, ASN.1, SMB, AFP
5	Συνεδρίας (Session)	π.χ. ISO 8327 / CCITT X.225, RPC, Netbios, ASP
4	Μεταφοράς (Transport)	π.χ. TCP, UDP, RTP, SPX, ATP
3	Δικτύου (Network)	π.χ.IP (Ipv4 ή Pv6), ICMP, IGMP, X.25, CLNP, ARP, OSPF, RIP, IPX, DDP
2	Συνδέσμου (Link)	π.χ. Ethernet, Token Ring, PPP, HDLC, Frame relay, ATM
1	Φυσικό (Physical)	π.χ. Ραδιοφωνικό σήμα, Λέϊζερ, Οπτική Ίνα

Πίνακας 1: Μοντέλο OSI

Τα τρία ανώτερα στρώματα του μοντέλου OSI (Εφαρμογής, Παρουσίασης και Συνεδρίας) αποτελούν ένα ενιαίο στρώμα στο TCP/IP, το επίπεδο Εφαρμογής. Τα χαρακτηριστικά του στρώματος Συνεδρίας αναλαμβάνονται από τις ίδιες εφαρμογές ή απλώς αγνοούνται. Ένα απλουστευμένο σχεδιάγραμμα της στοίβας του μοντέλου TCP/IP ακολουθεί:

4	Εφαρμογής	π.χ. HTTP, FTP, DNS *(Πρωτόκολλα δρομολόγησης, όπως το RIP, που βασίζονται στο πρωτόκολλο UDP μπορούν επίσης να καταχωρηθούν στο στρώμα Δικτύου)*
3	Μεταφοράς	π.χ. TCP, UDP, RTP *(Πρωτόκολλα δρομολόγησης, όπως το OSPF, που λειτουργούν πάνω από το IP, μπορούν επίσης να καταχωρηθούν στο στρώμα Δικτύου)*
2	Δικτύου	Για το TCP/IP, χρησιμοποιείται μόνο το IP *(Τα πρωτόκολλα ICMP και IGMP, παρόλο που βασίζονται πάνω στο IP για την λειτουργία τους, καταχωρούνται στο στρώμα Δικτύου. Το ARP αποτελεί μια από τις ολιγάριθμες εξαιρέσεις, εφόσον είναι ανεξάρτητο του IP)*
1	Συνδέσμου	π.χ. Ethernet, Token Ring, κλπ.

Πίνακας 2: Μοντέλο TCP/IP

Αυτά τα 4 επίπεδα, συναποτελούν το Μοντέλο Διαστρωμάτωσης του Internet ή αλλιώς, Μοντέλο αναφοράς του Internet.

1.1.2 Ανάλυση των επιπέδων

Εφαρμογής

Το στρώμα εφαρμογής χρησιμοποιείται από την πλειοψηφία των δικτυωμένων προγραμμάτων. Το πρόγραμμα παραδίδει τα δεδομένα σε μια μορφή που ορίζει το ίδιο. Εφόσον το TCP/IP δεν παρέχει στρώματα μεταξύ των στρωμάτων εφαρμογής και μεταφοράς, όλες οι λειτουργίες παρουσίασης και συνεδρίας πρέπει να υλοποιηθούν σε αυτό το επίπεδο. Αυτή η διαδικασία διευκολύνεται με τη χρήση βιβλιοθηκών.

Μεταφοράς

Το στρώμα μεταφοράς είναι υπεύθυνο για την μεταφορά μηνυμάτων, ανεξαρτήτως του υποκείμενου δικτύου, με έλεγχο σφαλμάτων (error control), κατάτμηση (fragmentation) και ρύθμιση ροής (flow control). Η μετάδοση μηνυμάτων μεταξύ δυο οντοτήτων μπορεί να κατηγοριοποιηθεί ως εξής:

1.connection-oriented, π.χ. TCP

2.connectionless, π.χ. UDP

Η λειτουργία του στρώματος αυτού μπορεί να συγκριθεί με αυτή οποιουδήποτε μηχανισμού/μέσου μεταφοράς, π.χ. ένα όχημα που πρέπει να εξασφαλίζει την πλήρη και ασφαλή διακίνηση του φορτίου του. Το στρώμα μεταφοράς παρέχει αυτή την υπηρεσία σύνδεσης εφαρμογών μεταξύ τους, κάνοντας χρήση θυρών (ports). Καθώς το IP προσφέρει μόνο παράδοση όσο το δυνατόν καλύτερα (best effort delivery), το στρώμα μεταφοράς είναι το πρώτο επίπεδο όπου λαμβάνεται υπόψιν το θέμα της αξιοπιστίας.

Παραδείγματος χάρη, σε μια προσπάθεια αξιόπιστης μετακίνησης δεδομένων, το TCP που είναι ένα connection-oriented πρωτόκολλο, έχει τα εξής χαρακτηριστικά:

- τα δεδομένα έρχονται στην ίδια σειρά με την οποία στάλθηκαν
- ελάχιστος έλεγχος σφαλμάτων
- ανεπιθύμητα αντίγραφα απορρίπτονται
- χαμένα/απορριφθέντα πακέτα ξαναστέλνονται
- έλεγχος κυκλοφοριακής συμφόρησης (congestion control)

Τα πρωτόκολλα δυναμικής δρομολόγησης (dynamic routing), που κανονικά θα έπρεπε να βρίσκονται σε αυτό το στρώμα του TCP/IP (αφού λειτουργούν πάνω από το IP) αντιμετωπίζονται συχνά ως τμήματα του επίπεδου δικτύου (π.χ. Το OSPF).

Το νέο SCTP είναι επίσης ένας "αξιόπιστος", connection-oriented μηχανισμός μεταφοράς. Είναι stream-oriented, όχι byte-oriented όπως το TCP, και προσφέρει την δυνατότητα multiplexing πολλών ρευμάτων (stream) σε μια μόνο σύνδεση. Προτείνει υποστήριξη multi-homing, την δυνατότητα δηλαδή για μια οντότητα να μπορέσει, στα πλαίσια μιας συγκεκριμένης σύνδεσης, να κάνει χρήση πολλαπλών (εφόσον υπάρχουν) διευθύνσεων IP, που αντιπροσωπεύουν πολλαπλές διεπαφές (διασυνδετικές διατάξεις), έτσι ώστε αν κάποια παρουσιάσει βλάβη, να μη χαθεί η σύνδεση.

Το UDP είναι ένα connectionless πρωτόκολλο διαγραμμάτων δεδομένων (datagrams). Όπως και το IP, είναι ένα best effort ή "αναξιόπιστο" πρωτόκολλο: ο έλεγχος σφαλμάτων είναι αδύναμος (απλό checksum). Χρησιμοποιείται κυρίως σε εφαρμογές streaming μέσων (ήχος, βίντεο, κλπ.) όπου η έγκαιρη άφιξη των δεδομένων είναι πιο σημαντική από την ακεραιότητα τους. Ο χρόνος που κερδίζεται σε σχέση με τα connection-oriented πρωτόκολλα, που πρέπει να καθιερώσουν μια αξιόπιστη σύνδεση, το καθιστά ιδανικό για απλές εφαρμογές ερωτημάτων/απαντήσεων (π.χ. DNS).

Το TCP και το UDP εκμεταλλεύονται από εφαρμογές που διακρίνονται (στο επίπεδο του δικτύου) από την θύρα TCP ή UDP τους. Ορισμένοι αριθμοί θυρών είναι κλειστοί και αναφέρονται σε πολύ συγκεκριμένες εφαρμογές.

Το RTP είναι ένα πρωτόκολλο διαγραμμάτων δεδομένων σχεδιασμένο για στοιχεία πραγματικού χρόνου (real-time) όπως τα streaming audio και video. Αν και παρουσιάζεται στο στρώμα μεταφοράς (αντί για το επίπεδο συνεδρίας), βασίζεται στο UDP για τη λειτουργία του.

Δικτύου

Ο σκοπός του στρώματος δικτύου είχε αρχικά καθοριστεί ως η μεταφορά πακέτων μέσω ενός ενιαίου δικτύου.

Με την εμφάνιση πιο σύνθετων μορφών δικτύων, προστέθηκαν επιπλέον χαρακτηριστικά στο στρώμα αυτό, έτσι ώστε ο ρόλος του να είναι πια η διακίνηση δεδομένων από το δίκτυο πηγής στο

δίκτυο προορισμού. Αυτό προϋποθέτει συνήθως τη δρομολόγηση πακέτων διαμέσου ενός δικτύου δικτύων (internetwork) ή διαδικτύου (με μικρά γράμματα).

Στην σουίτα πρωτοκόλλων Διαδικτύου, το IP μεταφέρει τα πακέτα δεδομένων από την πηγή, στον προορισμό. Το IP μπορεί να εξυπηρετήσει διάφορα πρωτόκολλα ανωτέρων επιπέδων (upper layer protocols) ˙ το καθένα τους προσδιορίζεται με έναν αποκλειστικό αριθμό πρωτοκόλλου: π.χ. Το ICMP και το IGMP έχουν τους αριθμούς 1 και 2 αντίστοιχα.

Μερικά πρωτόκολλα που στηρίζονται στο IP, π.χ. Το ICMP (χρησιμοποιείται για την διάδοση διαγνωστικών πληροφοριών σχετικά με την μεταφορά πακέτων μέσω IP) παρουσιάζονται πάνω από το IP αλλά παρέχουν υπηρεσίες επιπέδου διαδικτύου, απεικονίζοντας έτσι την ασυμβατότητα μεταξύ του Διαδικτύου, των πρωτοκόλλων Διαδικτύου και του μοντέλου OSI. Όλα τα πρωτόκολλα δρομολόγησης (π.χ. BGP, OSPF, RIP, κλπ.) ανήκουν επίσης στο στρώμα δικτύου, αν και θα μπορούσαν να τοποθετηθούν σε ανώτερα επίπεδα.

Συνδέσμου

Το στρώμα αυτό, ρόλος του οποίου είναι η διακίνηση πακέτων του επιπέδου δικτύου μεταξύ δυο οντοτήτων, δεν είναι στην ακρίβεια μέρος της σουίτας πρωτοκόλλων Διαδικτύου, διότι το IP λειτουργεί με διάφορα στρώματα συνδέσμου. Η διαδικασία διαβίβασης (αντ. λήψης) πακέτων σε ένα συγκεκριμένο επίπεδο συνδέσμου μπορεί να ελέγχεται είτε από τον οδηγό του interface, είτε το firmware ή σύνολο εξειδικευμένων κυκλωμάτων (chipsets), είτε τέλος από ένα συνδυασμό τον προ-αναφερθέντων. Αυτά θα εκτελέσουν τις λειτουργίες σύνδεσης δεδομένων (data link), όπως π.χ. την πρόσθεση επικεφαλίδας (packet header) πριν την αποστολή, την ίδια τη διαβίβαση του πλαισίου (frame) με τη χρήση ενός φυσικού μέσου.

Για συνδέσεις μέσω μόντεμ (σε γραμμή τηλεφώνου), τα πακέτα IP μεταφέρονται συνήθως χρησιμοποιώντας το PPP. Σε ευρυζωνικές συνδέσεις (π.χ. ADSL) συναντάμε το PPPoE. Σε τοπικά δίκτυα, τα πρωτόκολλα Ethernet ή IEEE 802.11 (για ενσύρματα ή ασύρματα δίκτυα αντίστοιχα) είναι πιο κοινά. Για δίκτυα ευρείας περιοχής (WAN) χρησιμοποιούνται συχνά το PPP πάνω σε γραμμές T-carrier ή E-carrier, το Frame relay, το ATM ή το Packet over SONET/SDH (POS).

Το στρώμα συνδέσμου είναι επίσης το επίπεδο όπου τα πακέτα μπορούν να αναχαιτιστούν για να σταλθούν σε ένα ιδεατό ιδιωτικό δίκτυο (Virtual Private Network, VPN). Σ' αυτήν την περίπτωση, τα δεδομένα του επιπέδου αυτού αντιμετωπίζονται ως δεδομένα εφαρμογής, και "ξανακατεβαίνουν" τη στοίβα πρωτοκόλλων Διαδικτύου για να σταλθούν. Στη λαμβάνουσα πλευρά, τα δεδομένα ανεβαίνουν δυο φορές τη στοίβα (μια για το VPN και μια δεύτερη για τη δρομολόγηση).

Το φυσικό επίπεδο, που αποτελείται από τα φυσικά στοιχεία του δικτύου (π.χ. hubs, repeaters, καλώδια δικτύου, οπτικές ίνες, ομοαξονικά καλώδια, κάρτες δικτύων) και τις προδιαγραφές χαμηλού επιπέδου των σημάτων (τάση, συχνότητα, κλπ.), θεωρείται συχνά ως μέρος του στρώματος συνδέσμου.

1.1.3 Υλοποιήσεις

Η πλειονότητα των λειτουργικών συστημάτων (Unix, Linux, οι διάφορες παραλλαγές BSD, Mac OS X, Microsoft Windows και οι εκδόσεις Windows Server) περιλαμβάνουν την στοίβα TCP/IP εξ ορισμού ˙ η αναζήτηση υλοποιήσεων αφορά μόνο ένα μικρό ποσοστό χρηστών. Αξίζει να σημειωθεί η ύπαρξη του Lightweight TCP/IP, μια υλοποίηση ανοιχτού κώδικα (open source) ειδικά σχεδιασμένη για ενσωματωμένα συστήματα (embedded systems) και του KA9Q NOS, διαμορφωμένο για ερασιτεχνικά ραδιοφωνικά συστήματα πακέτων (packet radio systems) ή υπολογιστών συνδεδεμένων μέσω σειριακών (serial) γραμμών.

1.2 OSI Model

Το OSI (Open Systems Interconnection) model δεν είναι παρά η εξέλιξη της TCP/IP suite. Αποτελείται από μια ιεραρχική δομή επτά επιπέδων που καθορίζει τις απαιτήσεις για την επικοινωνία μεταξύ δύο υπολογιστών. Το πρότυπο καθορίστηκε από το Διεθνή Οργανισμό για την Τυποποίηση (International Organization for Standardization), στα πρότυπα του ISO 7498-1. Δημιουργήθηκε για να επιτρέψει την επικοινωνία μεταξύ των διαφορετικών πλατφορμών που προέρχονται από διάφορους προμηθευτές. Το OSI επιτρέπει δηλαδή σε όλα τα στοιχεία ενός δικτύου να συνεργάζονται,

ανεξάρτητα από ποιος τα κατασκεύασε. Κάθε επίπεδο έχει την ιδιότητα ότι χρησιμοποιεί μόνο τις λειτουργίες του αμέσως κατώτερου επιπέδου και εξάγει πληροφορία και λειτουργία μόνο προς το αμέσως ανώτερό του επίπεδο.

Στο παρακάτω σχήμα φαίνεται η σχέση TCP/IP model και OSI model. Όπως είναι φανερό, τα 3 ανώτερα επίπεδα του OSI model, το Application layer, το Presentation layer και το Session layer, συνήθως υλοποιούνται σε ένα μόνο επίπεδο, πάνω από το Transport layer του TCP/IP model. Παρόλα αυτά δεν υπάρχει κάποιος συγκεκριμένος μηχανισμός που να απαγορεύει στο TCP/IP model να χρησιμοποιήσει περισσότερα του ενός επιπέδου, πάνω από το Transport layer.

Σήμερα χρησιμοποιείται μόνο ένα υποσύνολο ολόκληρου του προτύπου OSI. Θεωρείται ότι ένα μεγάλο μέρος των προδιαγραφών του είναι πάρα πολύ περίπλοκο και ότι η πλήρης ενσωμάτωση και λειτουργία του θα καθυστερήσει πολύ, αν και υπάρχουν πολλοί άνθρωποι που το υποστηρίζουν έντονα.

Επίπεδο 7: Εφαρμογών

Το επίπεδο εφαρμογών παρέχει στον χρήστη έναν τρόπο να προσπελάσει μέσω μιας εφαρμογής τις πληροφορίες ενός δικτύου. Αυτό το επίπεδο είναι η κύρια διασύνδεση του χρήστη με την εφαρμογή και, συνεπώς, με το δίκτυο. Στο επίπεδο αυτό γίνεται η διαχείριση των κατανεμημένων εφαρμογών, η αποστολή του ηλεκτρονικού ταχυδρομείου κλπ. Παραδείγματα πρωτοκόλλων επιπέδου εφαρμογών αποτελούν τα Telnet, FTP, SMTP και http.

Επίπεδο 6: Παρουσίασης

Το επίπεδο παρουσίασης μετασχηματίζει τα δεδομένα σε τυπική μορφή που την αναμένει το επίπεδο εφαρμογών. Στο επίπεδο αυτό τα δεδομένα υφίστανται κρυπτογράφηση, συμπίεση, κωδικοποίηση MIME και όποια άλλη διαμόρφωση απαιτεί η μορφή δεδομένων ή ο σχεδιαστής του πρωτοκόλλου. Παραδείγματα αποτελούν η μετατροπή αρχείων από κώδικα EBCDIC σε κώδικα ASCII και η μετατροπή της δομής των δεδομένων σε μορφή XML ή αντίστροφα (π.χ. από XML σε έγγραφο τύπου DOC).

Επίπεδο 5: Συνόδου

Το επίπεδο συνόδου ελέγχει τις συνόδους (δηλαδή τις ανταλλαγές δεδομένων) μεταξύ δύο υπολογιστών, του Α και του Β. Ξεκινά, διαχειρίζεται και τερματίζει τη σύνδεση μεταξύ μιας τοπικής και μιας απομακρυσμένης εφαρμογής. Αντιμετωπίζει λειτουργίες FDX (full duplex, οι Α και Β μιλούν ταυτόχρονα από δύο κανάλια) ή HDX (half-duplex, μιλάει ο Α και μετά απαντάει ο Β από το ένα διαθέσιμο κανάλι), ενώ υποστηρίζει διαδικασίες αποθήκευσης κατάστασης, αναβολής, τερματισμού και επανεκκίνησης. Αυτό το επίπεδο είναι υπεύθυνο για το ομαλό κλείσιμο της συνόδου (που είναι ιδιότητα του TCP) και επίσης για την αποθήκευση και ανάκτηση κατάστασης, λειτουργίες οι οποίες δεν χρησιμοποιούνται στην στοίβα πρωτοκόλλων του Διαδικτύου.

Επίπεδο 4: Μεταφοράς

Το επίπεδο μεταφοράς διεκπεραιώνει τη μεταφορά των δεδομένων από χρήστη σε χρήστη, απαλλάσσοντας έτσι τα ανώτερα επίπεδα από κάθε ανάγκη να προσφέρουν αξιόπιστη μεταφορά δεδομένων από το ένα άκρο της επικοινωνίας στο άλλο. Το επίπεδο μεταφοράς ελέγχει την αξιοπιστία ενός χρησιμοποιούμενου καναλιού με έλεγχο ροής, κατάτμηση και αποτμηματοποίηση, καθώς και έλεγχο σφαλμάτων. Ορισμένα πρωτόκολλα καταγράφουν καταστάσεις και συνδέσεις, οπότε κρατούν λογαριασμό των πακέτων και επανεκπέμπουν αυτά που δεν παρελήφθησαν σωστά. Τα διάφορα πρωτόκολλα μορφοποιούν διαφορετικά τα εκπεμπόμενα πακέτα πληροφοριών, αλλά τα προς αποστολή δεδομένα παραλαμβάνονται αρχικά από τα ανώτερα επίπεδα.

Το συνηθέστερο παράδειγμα πρωτοκόλλου μεταφοράς είναι το TCP. Άλλα πρωτόκολλα μεταφοράς είναι τα UDP, πρωτόκολλο για ασυνδεσμική αποστολή δεδομένων, SCTP, κλπ.

Επίπεδο 3: Δικτύου

Το επίπεδο δικτύου παρέχει τα λειτουργικά και διαδικαστικά μέσα για τη μεταφορά στοιχειοσειρών δεδομένων μεταβλητού μήκους από μια προέλευση σε έναν προορισμό, μέσα από ένα

ή περισσότερα ενδιάμεσα δίκτυα, ενώ διατηρεί την ποιότητα εξυπηρέτησης που απαιτεί το επίπεδο μεταφοράς. Το επίπεδο δικτύου εκτελεί λειτουργίες δρομολόγησης, με πιθανές κατατμήσεις / αποτμηματοποιήσεις, και αναφέρει σφάλματα σχετικά με την παράδοση των πακέτων. Οι δρομολογητές λειτουργούν στο επίπεδο αυτό· διακινώντας δεδομένα σε διασυνδεδεμένα δίκτυα έκαναν το Διαδίκτυο πραγματικότητα. Υπάρχουν και δικτυακοί διακόπτες που σχετίζονται με τις διευθύνσεις (IP). Εδώ υπάρχει μια λογική οργάνωση και τις τιμές των διευθύνσεων τις καθορίζει ιεραρχικά ο τεχνικός των επικοινωνιών. Το πλέον αναγνωρίσιμο παράδειγμα πρωτοκόλλου δικτύου είναι το Πρωτόκολλο Διαδικτύου.

Επίπεδο 2: Ζεύξης Δεδομένων

Το επίπεδο ζεύξης δεδομένων παρέχει τα λειτουργικά και διαδικαστικά μέσα για τη μεταφορά δεδομένων από μια συσκευή ενός τοπικού δικτύου σε άλλη, αλλά και για την ανίχνευση και διόρθωση σφαλμάτων που συμβαίνουν στο φυσικό επίπεδο. Οι μη ιεραρχημένες διευθύνσεις των συσκευών εδώ είναι οι φυσικές (π.χ. MAC διευθύνσεις), δηλαδή είναι προκαθορισμένες και αποθηκευμένες στις κάρτες δικτύου των επικοινωνούντων κόμβων από το εργοστάσιο.

Το πιο γνωστό πρότυπο αυτού του επιπέδου είναι το Ethernet, για τοπικά δίκτυα. Άλλα παραδείγματα πρωτοκόλλων ζεύξης δεδομένων αποτελούν τα:

- HDLC και ADCCP, για συνδέσεις από-σημείο-σε-σημείο.
- 802.11, για ασύρματα τοπικά δίκτυα.
- Στα τοπικά δίκτυα της οικογένειας πρωτοκόλλων IEEE 802, και σε κάποια άλλα όπως το FDDI, αυτό το επίπεδο μπορεί να διαιρεθεί σε δύο μικρότερα:
    - Ένα επίπεδο ελέγχου πρόσβασης στο κοινό μέσο, το υποεπίπεδο MAC
    - Ένα ανώτερο επίπεδο ελέγχου λογικών συνδέσεων, το υποεπίπεδο LLC, όπου επικρατεί καθολικά το πρωτόκολλο IEEE 802.2 ανεξάρτητα από το υποκείμενο πρωτόκολλο MAC ή φυσικού επιπέδου.

Στο επίπεδο αυτό λειτουργούν οι δικτυακές γέφυρες και οι δικτυακοί διακόπτες. Η συνδεσιμότητα παρέχεται μόνο για κόμβους που συνδέονται στο ίδιο κοινό μέσο (τοπικό δίκτυο ή σύνδεση από-σημείο-σε-σημείο).

Επίπεδο 1: Φυσικό

Το φυσικό επίπεδο ορίζει όλες τις ηλεκτρικές και φυσικές προδιαγραφές της επικοινωνίας. Σ'αυτές περιλαμβάνονται οι σχηματισμοί των ακίδων, οι επιτρεπτές τάσεις, οι προδιαγραφές των καλωδίων κλπ. Συσκευές φυσικού επιπέδου είναι οι διανεμητές, οι επαναλήπτες, οι κάρτες δικτύου, οι προσαρμοστές διαύλου. Οι κυριότερες λειτουργίες και υπηρεσίες του φυσικού επιπέδου είναι:

- Έναρξη και τερματισμός της ηλεκτρικής σύνδεσης μιας επικοινωνιακής συσκευής.
- Συμμετοχή σε διαδικασίες όπου οι επικοινωνιακές συσκευές εξυπηρετούν αποτελεσματικά πολλούς χρήστες (πολυπλεξία). Επιλύονται προβλήματα προτεραιότητας πρόσβασης και ελέγχου ροής δεδομένων.
- Διαμόρφωση και αποδιαμόρφωση των ψηφιακών δεδομένων κατά τη μετάδοση από συσκευή σε συσκευή. Για παράδειγμα, τα ψηφιακά ηλεκτρικά σήματα μπορεί να ταξιδέψουν ως αναλογικά σε χάλκινο καλώδιο, μετά σε οπτική ίνα, μετά να μεταδοθούν από ραδιοζεύξη ή δορυφορικά, να φθάσουν πάλι αναλογικά σε χάλκινο καλώδιο και να γίνουν ψηφιακά στον παραλήπτη.

Οι παράλληλοι δίαυλοι SCSI λειτουργούν στο επίπεδο αυτό. Επίσης τα επίπεδα 1 και 2 αφορούν οι προδιαγραφές των πρωτοκόλλων Ethernet, Token Ring, FDDI και IEEE 802.11.

1.3 Πρωτόκολλο Internet & IPv4 vs. IPv6

Το Internet Protocol (IP) βρίσκεται στο Network layer του TCP/IP model και, όπως είπαμε, είναι ένα από τα σημαντικότερα πρωτόκολλα του. Ενσωματώνεται σε τεχνολογίες που βρίσκονται στο αμέσως από κάτω του επίπεδο, το Data Link layer, όπως για παράδειγμα το Ethernet. Είναι ένα data-

oriented πρωτόκολλο, το οποίο χρησιμοποιείται για να στέλνονται δεδομένα σε μορφή πακέτων μέσα από ένα δίκτυο υπολογιστών.

Τα πακέτα είναι μικρές ακολουθίες από bytes που αποτελούνται από την επικεφαλίδα (header) και το κυρίως μέρος. Η επικεφαλίδα περιγράφει τον προορισμό του πακέτου, τον οποίο χρησιμοποιούν τα routers στο Internet για να κατευθύνουν το πακέτο στον τελικό του προορισμό.

1.3.1 Κύρια χαρακτηριστικά του Internet Protocol

Το Internet Protocol έχει δύο κύρια χαρακτηριστικά:

- Είναι Connectionless πρωτόκολλο. Για να πάνε τα δεδομένα από έναν υπολογιστή σε έναν άλλο δεν χρειάζεται κάποια προηγούμενη επικοινωνία.

- Είναι αναξιόπιστο. Αυτό σημαίνει ότι υλοποιεί τα λεγόμενο best effort delivery. Δεν υπάρχει εγγύηση για τα πακέτα ότι δεν θα χαθούν, αλλοιωθούν ή ότι θα φτάσουν με τη σωστή σειρά. Από πλευράς αξιοπιστίας το μόνο που κάνει το IP είναι να ελέγχει την επικεφαλίδα του πακέτου που θα στείλει ότι είναι error free, με τη χρήση checksum.

Σε περίπτωση συμφόρησης δεδομένων το IP μπορεί να απορρίψει πακέτα ή ακόμα, για λόγους αποδοτικότητας, 2 συνεχόμενα πακέτα να τα στείλει από διαφορετικές διευθύνσεις.

1.3.2 Τι είναι το best effort delivery;

Ο όρος best effort delivery περιγράφει ένα network service, για το οποίο το δίκτυο δεν παρέχει εγγύηση ότι τα δεδομένα του μεταδίδονται σωστά ή ότι ο χρήστης έχει ένα εγγυημένο QoS ή κάποια συγκεκριμένη προτεραιότητα. Σε ένα best effort δίκτυο παρέχεται σε όλους τους χρήστες best effort service, με την έννοια ότι ο καθένας έχει ένα μη καθορισμένο μεταβλητό ρυθμό μετάδοσης και παράδοσης πακέτου, ανάλογα με την κίνηση στο δίκτυο την συγκεκριμένη χρονική στιγμή.

1.3.3 Γιατί να σχεδιαστεί ένα πρωτόκολλο αναξιόπιστο;

Ο κύριος λόγος για την έλλειψη αξιοπιστίας είναι για να μειωθεί η πολυπλοκότητα των routers. Έτσι οι routers μπορούν να διαχειριστούν όπως θέλουν αυτοί τα πακέτα που τους έρχονται, αν και οτιδήποτε λιγότερο από best effort delivery οδηγεί σε "poor experience" τον τελικό χρήστη. Έτσι, αν και δεν γίνονται εγγυήσεις για τα πακέτα, όσο καλύτερη είναι η προσπάθεια του δικτύου, τόσο καλύτερο το τελικό αποτέλεσμα για τον χρήστη. Τέλος να αναφέρουμε ότι τα περισσότερα πρωτόκολλα είναι βασισμένα στην ιδέα ότι ο έλεγχος σφαλμάτων είναι καλύτερο να γίνεται στην μεριά του τελικού χρήστη, το λεγόμενο end-to-end principle.

1.4 IPv4 vs. IPv6

Το IPv4 προέρχεται από τα αρχικά των λέξεων Internet Protocol version 4. Είναι η τεχνολογία υποδομής που μας επιτρέπει να συνδέουμε τις διάφορες συσκευές μας στο Internet. Όταν μια συσκευή ζητάει πρόσβαση στο Internet (ή πρόκειται για PC ή για Mac ή για έξυπνο τηλέφωνο ή για οποιαδήποτε άλλη συσκευή), παίρνει μια μοναδική αριθμητική διεύθυνση του τύπου 192.168.1.1, η επονομαζόμενη IP διεύθυνση η οποία και αποτελεί την ταυτότητα της συγκεκριμένης συσκευής στο Internet. Για να αποσταλούν δεδομένα μεταξύ δύο υπολογιστών που είναι συνδεδεμένοι στο Internet, τα πακέτα που θα διακινηθούν πρέπει να περιέχουν και τις δύο IP διευθύνσεις (αποστολέα και παραλήπτη). Χωρίς IP διευθύνσεις οι συσκευές δεν θα μπορούσαν να επικοινωνήσουν, όπως θα γινόταν και με το συμβατικό ταχυδρομείο (αν δεν γνωρίζουμε τη διεύθυνση κάποιου δεν μπορούμε να στείλουμε γράμμα).

Το IPv6 είναι η έκδοση 6 του πρωτοκόλλου επικοινωνίας που προαναφέραμε. Λειτουργεί παρόμοια με το IPv4 στο ότι παρέχει μοναδικές αριθμητικές διευθύνσεις στις διασυνδεδεμένες συσκευές. Υπάρχει όμως μια μεγάλη διαφορά: χρησιμοποιεί διευθύνσεις μήκους 128-bit. Θα εξηγήσουμε αμέσως τι σημαίνει αυτό και γιατί είναι τόσο κρίσιμο για τη λειτουργία του Internet.



λλων

Το IPv4 χρησιμοποιεί 32 bits για τις διευθύνσεις που μοιράζει. Αυτό σημαίνει ότι μπορεί να παράγει ένα μέγιστο αριθμό διευθύνσεων της τάξης του 2^32 (περίπου 4.29 δισεκατομμύρια). Μπορεί το νούμερο να μας φαίνεται υπερβολικά μεγάλο, αλλά όλες οι διευθύνσεις έχουν ήδη δοθεί σε διάφορους οργανισμούς, με αποτέλεσμα να φτάσουμε στη σημερινή κρίσιμη κατάσταση. Ας ξεκαθαρίσουμε κάτι όμως: ακόμη δεν έχουμε ξεμείνει οριστικά από διευθύνσεις. Αρκετές από αυτές παραμένουν αχρησιμοποίητες στα χέρια μεγάλων Ιδρυμάτων (όπως το MIT) ή εταιρειών (όπως Ford και IBM). Αρκετές διευθύνσεις είναι ακόμη διαθέσιμες για ανάθεση και αρκετές θα πωληθούν ή ανταλλαγούν καθόσον τείνουν να γίνουν απειλούμενο είδος.

Όπως αναφέρθηκε προηγούμενα το IPv6 χρησιμοποιεί διευθύνσεις 128-bit. Αρα έχει δυνατότητα παροχής 2^128 διευθύνσεων, για την ακρίβεια 340.282.366.920.938.000.000.000.000.000.000.000.000 από αυτές. Ο αριθμός αυτός είναι τόσο μεγάλος ώστε χρειάζεται το δεκαεξαδικό σύστημα αριθμών για να απεικονιστούν οι διευθύνσεις. Με άλλα λόγια, το νέο σύστημα παρέχει διευθύνσεις περισσότερο από αρκετές, ώστε το Internet να συνεχίσει να λειτουργεί απροβλημάτιστα για πολλά πολλά ακόμη χρόνια.

Η έλλειψη των IPv4 διευθύνσεων είχε προβλεφθεί εδώ και αρκετά χρόνια, οπότε το όλο σχέδιο της ανάπτυξης του καινούριου συστήματος και της μετάβασης σε αυτό ήταν σε εξέλιξη την τελευταία δεκαετία. Η πρόοδος όμως είναι αργή - μόνο ένα μικρό μέρος του παγκόσμιου ιστού υποστηρίζει το καινούριο πρωτόκολλο. Επιπρόσθετα, δουλεύοντας τα δύο πρωτόκολλα παράλληλα, απαιτείται ειδικός εξοπλισμός για την μεταξύ τους επικοινωνία. Για να κατασκευαστεί ο “διακόπτης”, λογισμικό και δρομολογητές (routers) πρέπει να μετατραπούν ώστε να υποστηρίζουν το αναβαθμισμένο δίκτυο. Αυτό απαιτεί χρόνο και χρήμα. Η πρώτη πραγματική δοκιμή προγραμματίστηκε στις 8 Ιουνίου 2011 (World IPv6 Day). Μεταξύ των πρωτοπόρων εταιρειών είναι η Google και το Facebook. Με τις δοκιμές τους θα βοηθήσουν ώστε η μετάβαση να γίνει όσο το δυνατόν πιο ανώδυνα για τον υπόλοιπο κόσμο.

Αρχικά δεν υπάρχει μεγάλη επίδραση στη ζωή του απλού χρήστη. Τα περισσότερα λειτουργικά υποστηρίζουν το IPv6 (Mac OS X 10.2 , Windows XP SP1). Όμως πολλοί δρομολογητές (routers) και εξυπηρετητές δικτύου (servers) δεν το υποστηρίζουν με αποτέλεσμα η συνεργασία των δύο πρωτοκόλλων να είναι αδύνατη. Το IPv6 βρίσκεται ακόμη σε βρεφικό στάδιο. Αναμένονται πολλά προβλήματα ασφαλείας αλλά και λάθη στον ίδιο τον κώδικα.

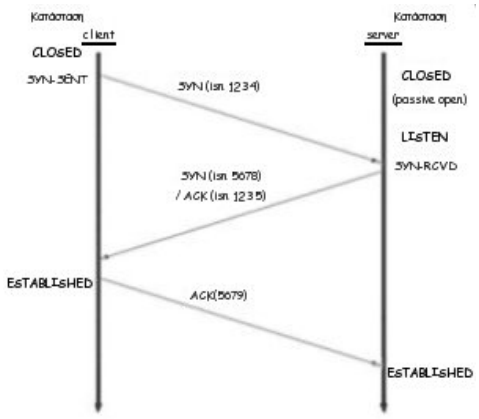
1.5 Πρωτόκολλα TCP, UDP και RTP

Ένα επίπεδο πιο πάνω από το IP, στο Transport layer, βρίσκονται το TCP & UDP. Οι software βιβλιοθήκες του TCP και του UDP, χρησιμοποιούν το Internet Protocol (IP) και παρέχουν ένα πιο απλοποιημένο περιβάλλον για να γραφούν δικτυακές εφαρμογές. Ας δούμε το TCP & UDP αναλυτικά παρακάτω.

1.5.1 Πρωτόκολλο TCP

Τα αρχικά του TCP προέρχονται από το Transmission Control Protocol. Είναι το πρωτόκολλο που χρησιμοποιείται κατεξοχήν στον Internet. Το TCP δημιουργεί μια ατελείωτη σειρά από μηνύματα (segments), τα οποία φαίνονται σαν μια συνεχής ροή δεδομένων. Αυτή η ροή είναι 2 κατευθύνσεων

(αποστολέας-παραλήπτης) και ταυτόχρονα πολύ αξιόπιστη. Ας δούμε τα κυριότερα χαρακτηριστικά του TCP:



Εικόνα 3: Έναρξη της σύνδεσης με three-way handshake

- Είναι connection protocol. Άρα χρησιμοποιείται μόνο μεταξύ 2 υπολογιστών. Πριν ξεκινήσει η μεταφορά δεδομένων πρέπει να γίνει ένα 3πλό handshaking (SYN, SYN-ACK, ACK) μεταξύ των 2 υπολογιστών (connection establishment) και το αντίστοιχο στον τερματισμό της αποστολής (connection termination).

- Είναι αξιόπιστο. Το TCP του παραλήπτη ενημερώνει συνεχώς το TCP του αποστολέα, για το πιο είναι το επόμενο πακέτο που περιμένει, σύμφωνα με τον αύξοντα αριθμό των πακέτων που έχει ήδη λάβει και αν αντιληφθεί ότι κάποιο πακέτο χάθηκε στην πορεία, τότε επιβάλλει retransmission. Αν το πακέτο δεν μπορεί να έρθει μετά από πολλαπλά retransmissions, τότε η σύνδεση διακόπτεται (timeout).

- Εγγυάται την σωστή σειρά άφιξης των δεδομένων στην εφαρμογή του παραλήπτη. Όταν τα δεδομένα έρθουν στην είσοδο του παραλήπτη με λάθος σειρά, τότε το TCP layer “κρατάει” αυτά τα δεδομένα μέχρι να έρθουν τα προηγούμενα τους. Αφού έρθουν τα διατάσσει στην σωστή σειρά και έπειτα τα παραδίδει στην εφαρμογή.

- Απόρριψη διπλών δεδομένων. Αποτρέπει την αποστολή διπλότυπων, δηλαδή δύο ακριβώς ίδιων δεδομένων.

- Προσφέρει αυτοματοποιημένο έλεγχο ροής δεδομένων (flow control). Όταν ο buffer του παραλήπτη γεμίσει, τότε σταματάει προσωρινά το transmission ή ελαττώνει τον ρυθμό μετάδοσης του, μέχρις ότου αδειάσει ο buffer. Μια απαραίτητη λειτουργία σε ένα κόσμο όπου επικοινωνούν μηχανές διαφορετικών ταχυτήτων κάτω από διαφορετικά δίκτυα.

- Προσφέρει αυτοματοποιημένο έλεγχο συμφόρησης (congestion control). Το TCP χρησιμοποιεί μια πληθώρα μηχανισμών για να επιτύχει την μέγιστη απόδοση μεταφοράς δεδομένων αποφεύγοντας την συμφόρηση δεδομένων στους routers του Internet, μια κατάσταση κατά την οποία πέφτει η απόδοση του δικτύου κατά μεγάλο βαθμό. Αυτοί οι μηχανισμοί ελέγχουν τον ρυθμό με τον οποίο τα δεδομένα μπαίνουν στο δίκτυο, κρατώντας αυτό το ρυθμό κάτω από ένα ασφαλές όριο. Συγκεκριμένα, ο βασικός αλγόριθμός που χρησιμοποιεί ονομάζεται Congestion Avoidance. Αν και υπάρχουν διάφορες παραλλαγές του, λειτουργεί ως εξής: Το TCP στέλνει δεδομένα στο δίκτυο αυξάνοντας σιγά σιγά ένα congestion window (χρησιμοποιείται για να ελέγχει τον ρυθμό με τον οποίο τα δεδομένα μπαίνουν στο δίκτυο). Όταν παρατηρηθεί συμφόρηση το μέγεθος του congestion window ελαττώνεται στο μισό, ώστε να αποφευχθεί η κατάρρευση του δικτύου και έπειτα από λίγο συνεχίζει να μεγαλώνει σιγά σιγά μέχρι να ξαναπαρατηρηθεί συμφόρηση, οπού και θα μειώσει πάλι το μέγεθος του congestion window στο μισό.

- Εγγυάται την ακεραιότητα του "μονοπατιού επικοινωνίας". Αυτό βέβαια δεν εμποδίζει την υποκλοπή των δεδομένων από τρίτους. Η τελευταία όμως είναι σχετικά δύσκολη, μιας και ο κακόβουλος χρήστης θα πρέπει να ακούσει όλη την ροή δεδομένων, καθώς δεν υπάρχουν συγκεκριμένου μεγέθους πακέτα. Εξαιτίας των παραπάνω χαρακτηριστικών του λοιπόν, το TCP χρησιμοποιείται και επιβάλλεται να χρησιμοποιείται, όπου η ακεραιότητα των δεδομένων είναι ύψιστης σημασίας. Δηλαδή σε web surfing, e-mails, file transfers και οποιαδήποτε άλλη μεταφορά data αρχείων ανάμεσα σε 2 υπολογιστές.

1.5.2 Πρωτόκολλο UDP

Τα αρχικά του UDP προέρχονται από το User Datagram Protocol. Είναι ένα σχεδόν μηδενικό πρωτόκολλο, με την έννοια ότι, οι μόνες υπηρεσίες που παρέχει είναι το checksum των προς μετάδοση δεδομένων και της πολυπλεξίας των ports επικοινωνίας του υπολογιστή. Για αυτό και

πολλές φορές αναφέρεται με το όνομα Unreliable Datagram Protocol! Χρησιμοποιώντας το UDP τα

γής

ράς

ου

μου

ια

προγράμματα μπορούν να στείλουν μικρά μηνύματα, γνωστά ως datagrams, το ένα στο άλλο.

- Είναι αναξιόπιστο (best effort delivery). Δε μπορεί να εγγυηθεί την ακεραιότητα ή τη σωστή σειρά άφιξης των δεδομένων, όπως το TCP. Τα πακέτα(datagrams) μπορούν να φτάσουν με διαφορετική σειρά, να εμφανίζονται διπλά ή να μην έρθουν και καθόλου χωρίς καμία ειδοποίηση.
- Είναι γρήγορο. Το παραπάνω χαρακτηριστικό του εξασφαλίζει μικρό delay και κάνει το UDP πιο γρήγορο από το TCP.
- Πολλαπλή χρηστικότητα. Μπορεί να χρησιμοποιηθεί τόσο σε Unicast όσο και σε Multicast δίκτυα, καθώς δεν είναι connection protocol. Θα αναφερθούμε σε αυτά παρακάτω.
- Είναι "ελαφρύ". Λιγότερο απαιτητικό σε πόρους, σε σχέση με το TCP. Δεν δημιουργεί μεγάλο overhead στο δίκτυο, καθώς δεν ελέγχει αν όντως κάποιο πακέτο έφτασε ή όχι.
- Έχει μικρότερο header. Το UDP έχει 8 bytes header, σε σχέση με το TCP που έχει 20 bytes header. Αυτό σημαίνει μικρότερο έξτρα overhead στο δίκτυο.

1.5.3 TCP vs. UDP

Στον πίνακα που ακολουθεί βλέπουμε συνοπτικά τις βασικές διαφορές των δύο πρωτοκόλλων που περιγράψαμε προηγουμένως:

UDP & TCP Comparison Table	UDP	TCP
Packet header size	8 Bytes	20 Bytes
Transport layer packet entity	Datagram	Segment
Port numbering	Yes	Yes
Error detection	Optional	Yes
Reliability: Error recovery by automatic repeat request (ARQ)	No	Yes
Virtual circuits: Sequence numbering and reordering	No	Yes
Flow control	No	Yes
Congestion avoidance: Variable congestion window, slow start, time outs	No	Yes
Multiple streams	No	No
ECN support	No	Yes

Πίνακας 3: Βασικές διαφορές των δύο πρωτοκόλλων

1.5.4 Γιατί υπάρχει το UDP; Τι εξυπηρετεί;

Το TCP λοιπόν, σύμφωνα με όσο αναφέραμε παραπάνω, φαντάζει σαν μια ιδανική λύση για την σωστή και αξιόπιστη κυρίως μεταφορά των δεδομένων μας. Γιατί χρησιμοποιούμε το UDP; Το UDP υπάρχει ακριβώς γιατί υπάρχουν εφαρμογές, όπου δεν μας ενδιαφέρει τόσο η ακεραιότητα των δεδομένων, όσο τα δεδομένα να φτάσουν όσο δυνατόν γρηγορότερα στον παραλήπτη, έστω και με κάποια απώλεια. Εκεί δηλαδή που το TCP είναι αργό και δεν μας εξυπηρετεί, έρχεται να πάρει τη θέση του το UDP. Μερικές εφαρμογές που χρησιμοποιούν το UDP είναι οι παρακάτω:

- Εφαρμογές οι οποίες μεταδίδουν real-time audio/video, όπως IPTV, VoIP. Εδώ μας ενδιαφέρει τα δεδομένα να φτάνουν την σωστή χρονική στιγμή. Οποιαδήποτε απώλεια τους μας επηρεάζει μόνο στην ποιότητα του αναπαραγόμενου σήματος.

- Servers, οι οποίοι απαντάνε σε μικρά αιτήματα ενός τεράστιου αριθμού από clients, όπως στα online παιχνίδια. Οι Servers δεν απασχολούνται με το να ελέγχουν την κατάσταση του κάθε connection και των παραμέτρων του χρησιμοποιώντας UDP, και έτσι μπορούν να εξυπηρετήσουν ένα πολύ μεγαλύτερο αριθμό χρηστών σε αντίθεση με το αν χρησιμοποιούσαν TCP.

- Όπως επίσης και κάποιες πολύ σημαντικές εφαρμογές όπως το Domain Name System(DNS), Simple Network Management Protocol(SNMP), Dynamic Host Configuration Protocol(DHCP), Routing Information Protocol(RIP).

Μετά από όλα αυτά, είναι φανερό ότι ένα πρόγραμμα που χρησιμοποιεί το πρωτόκολλο UDP πρέπει να ασχοληθεί το ίδιο με τα προβλήματα επικοινωνίας που μπορεί να προκύψουν: την αξιόπιστη παράδοση, το packetization και την επανασυναρμολόγηση, τον έλεγχο ροής, την αποφυγή συμφόρησης, κλπ. Επίσης, δεδομένου ότι το UDP στερείται μηχανισμών αποφυγής και ελέγχου δικτυακής συμφόρησης, απαιτούνται network-based μηχανισμοί για να ελαχιστοποιηθούν τα πιθανά προβλήματα κατάρρευσης δικτύου λόγω ανεξέλεγκτα υψηλών ρυθμών αποστολής πακέτων UDP. Με άλλα λόγια, δεδομένου ότι οι αποστολείς UDP δεν μπορούν να ανιχνεύσουν τη συμφόρηση, τα στοιχεία των δικτύων, όπως οι routers, πρέπει να χρησιμοποιούν τεχνικές packet queuing και απόρριψης πακέτων για να ελέγχουν την υπερβολική κίνηση πακέτων UDP στα δίκτυα.

Το Datagram Congestion Control Protocol (DCCP) σχεδιάζεται ως μια μερική λύση σε αυτό το πιθανό πρόβλημα προσθέτοντας TCP μηχανισμούς ελέγχου συμφόρησης σε ροές πακέτων UDP υψηλής ταχύτητας (π.χ. media streaming).

1.5.5 Πρωτόκολλο RTP

Το Real-time Transport Protocol (RTP) καθορίζει ένα τυποποιημένο format πακέτου για αποστολή ήχου/βίντεο μέσω του internet. Μπορεί να δουλέψει παράλληλα με το Real-time Streaming Protocol (RTSP) το οποίο επιτρέπει τον απομακρυσμένο έλεγχο ενός media server με εντολές παρόμοιες ενός βίντεο. Μερικές από τις εντολές που παρέχει το RTSP είναι η περιγραφή, εγκατάσταση, αναπαραγωγή, παύση, εγγραφή και τερματισμός. Οι εντολές αυτές αφορούν την περιγραφή του αρχείου προς μετάδοση, τον έλεγχο της αναπαραγωγής και τον τερματισμό της συνόδου με τον server. Όλες αυτές οι εντολές στέλνονται με το πρωτόκολλο HTTP.

Το RTP δεν έχει κάποια συγκεκριμένη TCP/UDP πόρτα από την οποία επικοινωνεί. Το μόνο στο οποίο υπακούει είναι ότι οι UDP επικοινωνίες γίνονται σε ζυγές πόρτες και η αμέσως επόμενη μονή πόρτα χρησιμοποιείται για το RTP Control Protocol (RTCP). Το τελευταίο δίνει περιοδικά feedback για το QOS που παρέχει το RTP. Μαζεύει στατιστικά, όπως τα πακέτα που στάλθηκαν, τα χαμένα πακέτα, το jitter και το round trip delay. Μια εφαρμογή μπορεί να τα χρησιμοποιήσει για να αυξήσει το QOS, ίσως μειώνοντας την ροή πληροφορίας ή χρησιμοποιώντας διαφορετικό codec. Το RTP λοιπόν είναι κατάλληλο να χρησιμοποιηθεί για δεδομένα με real-time χαρακτηριστικά, όπως το interactive audio και video. Επίσης το χρησιμοποιούν υπηρεσίες VoIP στις οποίες η εγκατάσταση κλήσης όπως και ο τερματισμός της, γίνεται χρησιμοποιώντας άλλα 2 πρωτόκολλα είτε το SIP είτε το H.323. Οι υπηρεσίες που παρέχει το RTP είναι οι παρακάτω:

- Ένδειξη του τι είδους περιεχόμενο μεταφέρεται
- Αρίθμηση πακέτων

- Δυνατότητα υπολογισμού του jitter
- Παρακολούθηση της διαδικασίας αποστολής.
- ΔΕΝ παρέχει flow & congestion control από μόνο του.

Επίσης το RTCP προσφέρει πληροφορία σχετικά με την λαμβανόμενη ποιότητα, την οποία μπορεί να χρησιμοποιήσει η εφαρμογή για να κάνει τοπικές αλλαγές, όπως να αποφασίσει σε περίπτωση συμφόρησης να μειώσει τον ρυθμό αποστολής.

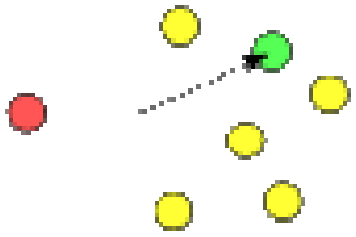
1.6 Μέθοδοι δρομολόγησης (Routing Schemes)

Υπάρχουν 4 μέθοδοι δρομολόγησης/αποστολής ή αλλιώς Routing Schemes, το καθένα με τα δικά του μοναδικά χαρακτηριστικά. Είναι, με τη σειρά που παρουσιάζονται, το Unicast, το Broadcast, το Multicast και το Anycast. Οι ορολογίες αυτές αναφέρονται και στον τύπο των διευθύνσεων IP που χρησιμοποιούνται.

1.6.1 Unicast

Ο πιο συνηθισμένος τύπος για μια IP διεύθυνση είναι μια unicast διεύθυνση και είναι ο πιο διαδεδομένος τρόπος μετάδοσης πληροφορίας στο σημερινό Internet. Συνήθως αναφέρεται σε έναν μεμονωμένο αποστολέα ή παραλήπτη και μπορεί να χρησιμοποιηθεί τόσο για αποστολή όσο και παραλαβή. Συνήθως μια unicast διεύθυνση αντιστοιχίζεται με μια μόνο συσκευή, αλλά αυτό δεν σημαίνει ότι υπάρχει αντιστοιχία 1-1. Ορισμένοι υπολογιστές έχουν πολλές διαφορετικές unicast διευθύνσεις, η κάθε μια για την δικιά της ξεχωριστή χρήση. Στέλνοντας την ίδια πληροφορία σε διαφορετικές unicast διευθύνσεις, απαιτεί από τον αποστολέα να στείλει την ίδια πληροφορία τόσες φορές όσες και οι παραλήπτες του.

Στην unicast δρομολόγηση κάθε router εξετάζει το διεύθυνση παραλήπτη του λαμβανόμενου πακέτου και ψάχνει αυτήν την διεύθυνση σε έναν πίνακα, ώστε να προσδιορίσει ποια διασύνδεση να χρησιμοποιήσει ώστε το πακέτο να φτάσει πιο κοντά στον προορισμό του.



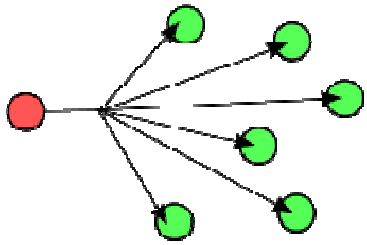
Εικόνα 5: Unicast μετάδοση

Η διεύθυνση αποστολέα του πακέτου δεν παίζει κανένα ρόλο.

1.6.2 Broadcast

Το broadcast, στέλνοντας την πληροφορία σε όλους τις πιθανούς προορισμούς, δίνει τη δυνατότητα στον αποστολέα, να στείλει την πληροφορία μόνο μια φορά και όλοι οι παραλήπτες να την πάρουν. Στο IP πρωτόκολλο, η διεύθυνση 255.255.255.255 παριστάνει ένα περιορισμένο τοπικό broadcast. Για παράδειγμα, το να στείλεις σε όλες τις διευθύνσεις σε ένα τοπικό δίκτυο που αρχίζουν με 192.0.2, η directed broadcast διεύθυνση είναι 192.0.2.255 (υποθέτοντας ότι το netmask είναι 255.255.255.0).

Δυστυχώς δεν υπάρχει καμία μορφή internet-wide broadcast και είναι περιορισμένο σαν τεχνολογία μόνο στα τοπικά Ethernet δίκτυα. Στο IPv6 το broadcast δεν υφίσταται σε καμία μορφή και έχει δώσει την θέση του στο multicasting.

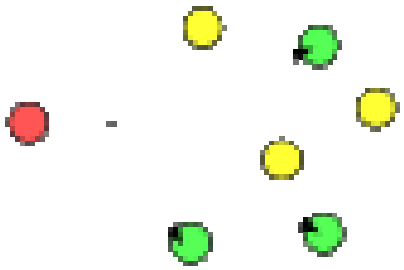


Εικόνα 6: Broadcast μετάδοση

1.6.3 Multicast

Μια multicast διεύθυνση αντιστοιχίζεται με ένα group από ενδιαφερόμενους χρήστες. Σύμφωνα με το RFC3171 της IANA (Internet Assigned Numbers Authority), οι διευθύνσεις 224.0.0.0 μέχρι 239.255.255.255 είναι ορισμένες ως multicast διευθύνσεις. Αυτές οι διευθύνσεις ήταν γνωστές παλαιότερα και με την ονομασία Class D. Ο αποστολέας στέλνει ένα datagram UDP πακέτο από την unicast διεύθυνση του, στην multicast διεύθυνση και οι routers αναλαμβάνουν να κάνουν αντίγραφα του πακέτου μόνο όταν χρειάζεται και να το στείλουν σε όσους παραλήπτες δήλωσαν ενδιαφέρον για το συγκεκριμένο πακέτο από τον συγκεκριμένο αποστολέα. Δεν απαιτείται προηγούμενη γνώση για το ποιοι ή πόσοι παραλήπτες υπάρχουν στο δίκτυο.

Κάθε υπολογιστής (στην ουσία κάθε εφαρμογή του υπολογιστή) που θέλει να λάβει πληροφορία από ένα multicast group, πρέπει να χρησιμοποιήσει το IGMP για να πάρει μέρος. Τα ενδιάμεσα routers πρέπει κι αυτά να χρησιμοποιούν το IGMP για να επικοινωνήσουν. Σε αντίθεση με τη unicast δρομολόγηση που αναφέραμε παραπάνω, η διεύθυνση αποστολέα (που είναι μια απλή unicast διεύθυνση) χρησιμοποιείται για να προσδιορίσει την κατεύθυνση του data stream. Η πηγή του multicast traffic θεωρείται σαν upstream. Το router προσδιορίζει ποιες downstream διασυνδέσεις είναι προορισμοί για αυτό το multicast group και στέλνει το πακέτο μέσω των κατάλληλων διασυνδέσεων. Ο όρος reverse path forwarding χρησιμοποιείται για να περιγράψει αυτό το είδος της δρομολόγησης. Δηλαδή, το να δρομολογείς πακέτα μακριά από την πηγή, παρά κοντά προς τον προορισμό. Τέλος, όπως αναφέρει και ο Internet architect Dave Clark για το multicast: "Βάζεις πακέτα στην μια άκρη, και το δίκτυο συνωμοτεί να τα μεταφέρει σε όποιον τα ζητήσει".

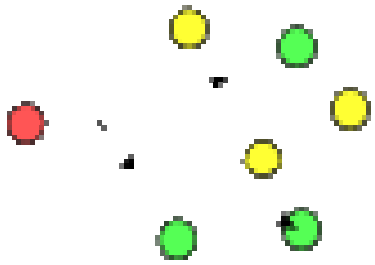


Εικόνα 7: Multicast μετάδοση

1.6.4 Anycast

Όπως το broadcast και το multicast, έτσι και το anycast είναι μια-προς-πολλούς τοπολογία. Η διαφορά του είναι ότι η πληροφορία δεν μεταδίδεται σε όλους τους παραλήπτες, αλλά μόνο σε αυτόν που ο router θα αποφασίσει ότι είναι πιο κοντά στον αποστολέα. Δηλαδή στο anycast κάθε διεύθυνση αντιστοιχίζεται με ένα σύνολο παραληπτών, όπως στο multicast και το broadcast, αλλά μόνο ένας από αυτούς επιλέγεται σε μια δεδομένη στιγμή για να πάρει πληροφορία από τον εκάστοτε αποστολέα. Το

anycast είναι πολύ χρήσιμο για να εξισορροπεί μεταβολές στον φόρτο του δικτύου. Χρησιμοποιείται κυρίως στο DNS και στο UDP.



Εικόνα 8: Anycast μετάδοση

2. Μετρήσεις δικτύων

Σε αυτή την ενότητα θα εξετάσουμε τα στοιχεία που λαμβάνουμε από μία ενεργή μέτρηση. Τα στοιχεία αυτά είναι τα εξής: διεύθυνση αποστολέα, διεύθυνση παραλήπτη, ώρα αποστολής, ώρα λήψης, αριθμός πακέτων που στάλθηκαν, αριθμός πακέτων που απορρίφθηκαν, αριθμός πακέτων που ελήφθησαν, αριθμός πακέτων που ελήφθησαν μετά τη λήξη του χρόνου, συνολικός αριθμός χαμένων πακέτων, διπλότυπα πακέτα, μέσος χρόνος καθυστέρησης, ελάχιστος χρόνος καθυστέρησης, μέγιστος χρόνος καθυστέρησης, παραμόρφωση χρονισμού (jitter),διακύμανση, throughput.

2.1 Διεύθυνση αποστολέα/ παραλήπτη

Η διεύθυνση αποστολέα/παραλήπτη είναι η τοπική διεύθυνση IP για να συνδεθεί ο αποστολέας/παραλήπτης. Αυτή η διεύθυνση μπορεί να χρησιμοποιηθεί σε ένα multi-homed εξυπηρετητή. Η διεύθυνση του αποστολέα/παραλήπτη πρέπει να έχει την ακόλουθη μορφή:

```
IPv4_address    // for IPv4 address
[IPv6_address]  // for IPv6 address
```

2.2 Ώρα αποστολής/λήψης

Η ώρα αποστολής/ λήψης μας δείχνει την χρονική στιγμή κατά την οποία ξεκίνησε η αποστολή/λήψη του πρώτου πακέτου.

2.3 Αριθμός πακέτων που στάλθηκαν, αριθμός πακέτων που απορρίφθηκαν, αριθμός πακέτων που ελήφθησαν, αριθμός πακέτων που ελήφθησαν μετά τη λήξη του χρόνου, συνολικός αριθμός χαμένων πακέτων

Από αυτά τα στοιχεία βλέπουμε τον αριθμό τον πακέτων που στάλθηκαν συνολικά, πόσα από αυτά απορρίφθηκαν για οποιονδήποτε λόγο, πόσα από αυτά έφτασαν τελικά στον αποστολέα, πόσα από αυτά έφτασαν μετά τη λήξη του χρόνου καθώς και έναν συνολικό αριθμό των χαμένων πακέτων.

2.4 Καθυστέρηση Δικτύου – delay

Η καθυστέρηση δικτύου είναι ένα σημαντικό χαρακτηριστικό του σχεδιασμού και της απόδοσης ενός δικτύου υπολογιστών ή τηλεπικοινωνιακού δικτύου . Η καθυστέρηση ενός δικτύου καθορίζει πόσο καιρό παίρνει για ένα κομμάτι δεδομένων να ταξιδέψει σε όλο το δίκτυο από έναν κόμβο ή τελικό σημείο στο άλλο. Συνήθως μετριέται σε πολλαπλάσια ή κλάσματα δευτερολέπτων. Η καθυστέρηση μπορεί να διαφέρει, ανάλογα με τη διαδρομή που θα ακολουθηθεί. Παρά το γεγονός ότι μόνο οι χρήστες ενδιαφέρονται για τη συνολική καθυστέρηση του δικτύου, οι μηχανικοί πρέπει να εκτελούν ακριβείς μετρήσεις. Έτσι, οι μηχανικοί παρουσιάζουν συνήθως τόσο τη μέγιστη και τη μέση καθυστέρηση, και διανέμουν τη καθυστέρηση σε διάφορα μέρη:

- Καθυστέρηση επεξεργασίας - χρόνος για να διευθετηθεί η επικεφαλίδα του πακέτου,
- Καθυστέρηση ουράς- χρόνος που το πακέτο περνά στην ουρά δρομολόγησης,
- Καθυστέρηση μετάδοσης - χρόνος που χρειάζεται για να μεταδοθούν τα bits του πακέτου επάνω στο σύνδεσμο
- Καθυστέρηση διάδοσης - χρόνος για ένα σήμα για να φτάσει στον προορισμό του

Υπάρχει ένα ορισμένο ελάχιστο επίπεδο καθυστέρησης που θα σημειωθεί λόγω του χρόνου που χρειάζεται για να μεταδοθεί ένα πακέτο σειριακά μέσω ενός συνδέσμου . Σε αυτό προστίθενται διαφορετικά επίπεδα καθυστέρησης λόγω συμφόρησης του δικτύου . Οι IP καθυστερήσεις του δικτύου μπορεί να κυμαίνονται από λίγα χιλιοστά του δευτερολέπτου σε μερικές εκατοντάδες χιλιοστά του δευτερολέπτου.

2.5 Παραμόρφωση χρονισμού (Jitter)

Για εφαρμογές όπως η ροή ήχου και βίντεο, δεν έχει μεγάλη σημασία αν τα πακέτα θα χρειαστούν 20 msec ή 30 msec για να παραδοθούν, στο βαθμό που ο χρόνος διέλευσης είναι σταθερός. Η διαταραχή (δηλαδή η απόκλιση) στους χρόνους άφιξης των πακέτων ονομάζεται παραμόρφωση χρονισμού (jitter). Η υψηλή παραμόρφωση χρονισμού οδηγεί σε ακανόνιστη ποιότητα στον ήχο ή το βίντεο. Σε αντιδιαστολή, μια συμφωνία ότι το 99% των πακέτων θα παραδίδεται με καθυστέρηση στην περιοχή απο 24,5 msec έως 25,5 msec μπορεί να είναι αποδεκτή.

Η περιοχή που επιλέγεται θα πρέπει, βέβαια, να είναι εφικτή. Θα πρέπει να λαμβάνει υπόψιν το χρόνο διέλευσης λόγω της ταχύτητας του φωτός καθώς και την ελάχιστη καθυστέρηση μέσω των

δρομολογητών, αφήνοντας πιθανόν και κάποιο περιθώριο για μερικές άλλες αναπόφευκτες καθυστερήσεις.

Μπορούμε να εκτιμήσουμε την παραμόρφωση χρονισμού υπολογίζοντας τον αναμενόμενο χρόνο διέλευσης για κάθε κόμβο της διαδρομής. Όταν ένα πακέτο φτάνει σε ένα δρομολογητή, ο δρομολογητής ελέγχει για να δει αν το πακέτο βρίσκεται “πίσω” ή “μπροστά” σε σχέση με το πρόγραμμά του. Η πληροφορία αυτή αποθηκεύεται στο πακέτο και ενημερώνεται σε κάθε άλμα. Αν το πακέτο βρίσκεται μπροστά σε σχέση με το πρόγραμμα, καθυστερεί όσο χρειάζεται για να επανέλθει στο πρόγραμμα. Αν βρίσκεται πίσω σε σχέση με το πρόγραμμα, ο δρομολογητής προσπαθεί να το διώξει όσο πιο γρήγορα γίνεται.

Στην πραγματικότητα, ο αλγόριθμος που καθορίζει ποιο από τα πολλά πακέτα που ανταγωνίζονται για μια γραμμή εξόδου θα σταλεί στην συνέχεια, μπορεί να επιλέγει πάντοτε το πακέτο που είναι πιο καθυστερημένο σε σχέση με το πρόγραμμα του. Με αυτόν τον τρόπο τα πακέτα που βρίσκονται μπροστά σε σχέση με το πρόγραμμα θα επιταχύνονται – άρα θα μειώνεται και στις δύο περιπτώσεις η παραμόρφωση χρονισμού.

Σε μερικές εφαρμογές, όπως το βίντεο κατόπιν αιτήσεως, μπορούμε να εξαλείψουμε την παραμόρφωση χρονισμού αποθηκεύοντας προσωρινά τα δεδομένα στον παραλήπτη και προσκομίζοντας στη συνέχεια τα δεδομένα από το χώρο αποθήκευσης, αντί να προσκομίζονται τα δεδομένα σε πραγματικό χρόνο από το δίκτυο. Παρόλα αυτά, για κάποιες άλλες εφαρμογές, και ειδικά αυτές που απαιτούν αλληλεπίδραση μεταξύ ανθρώπων σε πραγματικό χρόνο (όπως η τηλεφωνία μέσω internet και η βιντεοδιάσκεψη) η εγγενής καθυστέρηση της προσωρινής αποθήκευσης δεν είναι αποδοτική.

2.6 Throughput

Σε δίκτυα επικοινωνίας, όπως το Ethernet ή packet radio, στη διακίνηση ή το δίκτυο διεκπεραίωσης είναι ο μέσος όρος της επιτυχούς παράδοσης μηνυμάτων σε ένα κανάλι επικοινωνίας. Αυτά τα δεδομένα μπορεί να παραδοθούν σε φυσική ή λογική σύνδεση, ή να περάσει μέσα από ένα συγκεκριμένο κόμβο του δικτύου . Η απόδοση συνήθως μετριέται σε bits ανά δευτερόλεπτο (bit/sec ή bps), και μερικές φορές σε πακέτα δεδομένων ανά δευτερόλεπτο ή πακέτα δεδομένων ανά χρονοθυρίδα .

Η απόδοση του συστήματος ή η συνολική απόδοση είναι το άθροισμα των ποσοστών των δεδομένων που παραδίδονται σε όλους τους τερματικούς σταθμούς σε ένα δίκτυο. Η απόδοση μπορεί να αναλυθεί μαθηματικά μέσω της θεωρίας των ουρών , όπου το φορτίο σε πακέτα ανά μονάδα χρόνου συμβολίζεται ως ρυθμός άφιξης λ, και η απόδοση σε πακέτα ανά μονάδα χρόνου συμβολίζεται ως ρυθμός αναχώρησης μ.

Throughput είναι ουσιαστικά συνώνυμο με την ψηφιακή κατανάλωση εύρους ζώνης .

2.6.1 Η μέγιστη απόδοση

Οι χρήστες των συσκευών τηλεπικοινωνιών, σχεδιαστές συστημάτων, και οι ερευνητές στη θεωρία της επικοινωνίας, συχνά ενδιαφέρονται να γνωρίζουν την αναμενόμενη απόδοση ενός συστήματος. Από τη σκοπιά του χρήστη, αυτό συχνά διατυπώνεται είτε ως “ποια συσκευή θα πάρει τα στοιχεία που υπάρχουν πιο αποτελεσματικά για τις ανάγκες μου;” , ή "ποια συσκευή θα παραδώσει τα περισσότερα δεδομένα ανά μονάδα;". Οι σχεδιαστές συστημάτων συνήθως ενδιαφέρονται και για την επιλογή των πλέον αποτελεσματικών αρχιτεκτονικών ή τους περιορισμούς σχεδιασμού για ένα σύστημα, το οποίο το οδηγεί στην τελική απόδοσή του. Στις περισσότερες περιπτώσεις, το σημείο αναφοράς για το τι σύστημα είναι σε θέση, ή πιο προσφέρει τη μέγιστη απόδοση, είναι αυτό που ενδιαφέρει το χρήστη ή σχεδιαστή. Κατά την εξέταση της απόδοσης, ο όρος της «μέγιστης απόδοσης», χρησιμοποιείται συχνά σε περιπτώσεις όπου οι δοκιμές της μέγιστης απόδοσης συζητούνται λεπτομερώς.

Τέσσερις διαφορετικές τιμές έχουν νόημα στο πλαίσιο της «μέγιστης απόδοσης», που χρησιμοποιούνται. Πρόκειται για τη «μέγιστη θεωρητική απόδοση», «μέγιστη δυνατή απόδοση», «μέγιστη στιγμιαία απόδοση» και «μέγιστη βιώσιμη απόδοση». Αυτά αντιπροσωπεύουν διαφορετικές ποσότητες και πρέπει να λαμβάνεται μέριμνα όταν οι ίδιοι οι ορισμοί χρησιμοποιούνται κατά τη σύγκριση διαφορετικών τιμών της μέγιστης απόδοσης. Συγκρίνοντας τις τιμές απόδοσης, αυτή

εξαρτάται επίσης από κάθε κομμάτι που μεταφέρει την ίδια ποσότητα πληροφοριών. Η συμπίεση δεδομένων μπορεί σημαντικά να παραποιήσει υπολογισμούς απόδοσης, συμπεριλαμβανομένης της δημιουργίας τιμών μεγαλύτερων από το 100%. Εάν η επικοινωνία μεσολαβείται από διάφορες συνδέσεις σε σειρά με διαφορετικούς ρυθμούς bit, η μέγιστη απόδοση του συνολικού συνδέσμου είναι μικρότερη ή ίση με το χαμηλότερο ρυθμό μετάδοσης bit. Η χαμηλότερη τιμή συνδέσμου στη σειρά αναφέρεται ως το σημείο συμφόρησης .

- **Η μέγιστη θεωρητική απόδοση**

Ο αριθμός αυτός είναι στενά συνδεδεμένος με την ικανότητα του καναλιού του συστήματος, και είναι η μέγιστη δυνατή ποσότητα δεδομένων που μπορούν να μεταδοθούν υπό ιδανικές συνθήκες. Σε ορισμένες περιπτώσεις ο αριθμός αυτός αναφέρεται ως ίσος με την χωρητικότητα καναλιού, αν και αυτό μπορεί να είναι παραπλανητικό, δεδομένου ότι μόνο μη packetized συστήματα (ασύγχρονα) μπορούν να επιτύχουν αυτό χωρίς συμπίεση δεδομένων. Ο αριθμός αυτός, όπως και η «μέγιστη δυνατή απόδοση», χρησιμοποιείται κυρίως για να υπολογίζεται κατά προσέγγιση, όπως για τον καθορισμό ορίων για τις πιθανές επιδόσεις, νωρίς, σε μια φάση σχεδιασμού του συστήματος.

- **Η μέγιστη στιγμιαία απόδοση**

Οι παραπάνω τιμές είναι θεωρητικές ή υπολογιζόμενες τιμές. Η μέγιστη στιγμιαία απόδοση είναι πραγματική, υλοποιείται από το σύστημα, ή μια προσομοίωση του συστήματος. Η τιμή είναι η απόδοση που μετριέται σε σύντομο χρονικό διάστημα. Μαθηματικά, αυτό είναι το όριο που λαμβάνεται σε σχέση με την απόδοση καθώς ο χρόνος πλησιάζει το μηδέν. Αυτός ο όρος είναι συνώνυμος με το "στιγμιαίο throughput". Αυτός ο αριθμός είναι χρήσιμος για συστήματα που βασίζονται σε δεδομένα ριπής εκπομπής, ωστόσο, για τα συστήματα με υψηλό κύκλο αυτό είναι λιγότερο πιθανό να είναι ένα χρήσιμο μέτρο της απόδοσης του συστήματος.

- **Η μέγιστη βιώσιμη απόδοση**

Αυτή η τιμή είναι ο μέσος όρος της απόδοση ή ενσωματωμένος σε μεγάλα χρονικά διάστημα (μερικές φορές θεωρείται άπειρο). Για δίκτυα υψηλού κύκλου αυτό είναι πιθανό να είναι ο πιο ακριβής δείκτης της απόδοσης του συστήματος. Η μέγιστη απόδοση ορίζεται ως η ασυμπτωτική απόδοση όταν το φορτίο (το ποσό των εισερχόμενων δεδομένων) είναι πολύ μεγάλο. Σε συστήματα μεταγωγής πακέτων όπου το φορτίο και η διακίνηση πάντα είναι ίσα (όπου η απώλεια πακέτων δεν συμβαίνει), η μέγιστη απόδοση μπορεί να οριστεί ως το ελάχιστο φορτίο σε bit/sec που προκαλεί ο χρόνος παράδοσης (το latency) για να γίνει ασταθής και να αυξηθεί προς το άπειρο.

2.6.2 Αξιοποίηση καναλιού - αποδοτικότητα καναλιού - Κανονικοποιημένη απόδοση

Το throughput είναι μερικές φορές σε εξομάλυνση και υπολογίζεται σε ποσοστό, αλλά η εξομάλυνση μπορεί να προκαλέσει σύγχυση σχετικά με το τι ποσοστό έχει σχέση με τι. Η αξιοποίηση καναλιού, η αποδοτικότητα καναλιού και η κανονικοποιημένη απόδοση σε ποσοστό είναι λιγότερο ασαφής όροι.

Η αποδοτικότητα καναλιού**,** γνωστό και ως αποδοτικότητα αξιοποίησης του εύρους ζώνης**,** σε ποσοστό είναι η επίτευξη απόδοσης που σχετίζεται με το καθαρό bitrate σε bit/sec από ένα ψηφιακό κανάλι επικοινωνίας. Για παράδειγμα, αν η διακίνηση είναι 70 Mbit/sec σε 100 Mbit/sec σύνδεση Ethernet, η αποδοτικότητα του καναλιού είναι το 70%. Σε αυτό το παράδειγμα, είναι αποτελεσματικά 70Mbits των δεδομένων που μεταδίδονται κάθε δευτερόλεπτο.

Η αξιοποίηση καναλιού είναι αντ ' αυτού ένας όρος που σχετίζεται με τη χρήση του καναλιού χωρίς να λαμβάνεται υπόψη η διακίνηση. Δεν μετράει μόνο τα bits δεδομένων, αλλά και τα overheads που διακινούνται στο δίαυλο. Η μεταφορά overheads αποτελείται από προοίμιες ακολουθίες, κεφαλίδες, πλαίσια και πακέτα αναγνώρισης. Οι ορισμοί υποθέτουν ένα αθόρυβο κανάλι. Σε αντίθετη περίπτωση, η απόδοση δεν θα είναι το μόνο που σχετίζεται με τη φύση (απόδοσης) του πρωτοκόλλου, αλλά και για την αναμετάδοση από την συνισταμένη της ποιότητας του καναλιού. Σε μια απλούστερη προσέγγιση, η αποδοτικότητα καναλιού μπορεί να είναι ίση με τη αξιοποίηση καναλιού θεωρώντας ότι αναγνωρίζουν τα πακέτα μηδενικού μήκους και ότι ο πάροχος επικοινωνίας δεν θα δει οποιοδήποτε εύρος ζώνης σε σχέση με την αναμετάδοση ή τις κεφαλίδες.

Σε μια point-to-point ή point-to-multipoint επικοινωνία, όπου μόνο ένα τερματικό μεταδίδει, η μέγιστη απόδοση είναι συχνά ισοδύναμη ή πολύ κοντά στο φυσικό ρυθμό μετάδοσης δεδομένων (τη χωρητικότητα καναλιού), δεδομένου ότι η αξιοποίηση καναλιού μπορεί να είναι σχεδόν 100% σε ένα τέτοιο δίκτυο.

Για παράδειγμα, στο Ethernet το μέγιστο μέγεθος του πλαισίου ειναι 1526 bytes (μέγιστο ωφέλιμο φορτίο 1500 byte + 8 προοίμιο + 14 byte κεφαλίδα + 4 byte τρέιλερ Byte). Ένα πρόσθετο κενό που αντιστοιχεί σε 12 byte προστίθεται μετά το κάθε καρέ. Αυτό αντιστοιχεί σε μέγιστη αξιοποίηση καναλιού του είναι 1526 / (1526 +12) • 100% = 99,22%. Η μέγιστη απόδοση ή η αποτελεσματικότητα του καναλιού τότε είναι 1500 / (1526 +12) = 97,5 Mbit/sec εκτός από τα γενικά Ethernet overheads.

2.6.3 Goodput

Η μέγιστη απόδοση είναι συχνά μια αναξιόπιστη μέτρηση του εύρους ζώνης, για παράδειγμα, η μετάδοση δεδομένων σε bits ανά δευτερόλεπτο. Όπως επισημάνθηκε παραπάνω, η επίτευξη απόδοσης είναι συχνά μικρότερη από την μέγιστη απόδοση. Επίσης, τα overheads επηρεάζουν το εύρος ζώνης.

Η απόδοση δεν είναι μια καλά καθορισμένη μέτρηση όταν πρόκειται για το πώς να ασχοληθεί με τα overheads. Συνήθως μετριέται σε ένα σημείο αναφοράς, κάτω από το στρώμα δικτύου και πάνω από το φυσικό επίπεδο.

Ο πιο απλός ορισμός είναι ο αριθμός των bits ανά δευτερόλεπτο που είναι να παραδοθεί στο φυσικό. Ένα τυπικό παράδειγμα, όπου ο ορισμός αυτός ασκείται είναι ένα δίκτυο Ethernet. Σε αυτή την περίπτωση η μέγιστη απόδοση είναι το ακαθάριστο bitrate ή πρωτεύων bitrate .

Ωστόσο, στο πλαίσιο σχεδίων που περιλαμβάνουν κώδικα διόρθωσης σφαλμάτων (κωδικοποίηση του διαύλου), ο πλεονάζων κώδικας σφάλματος συνήθως αποκλείεται από την διακίνηση. Ένα παράδειγμα στο modem επικοινωνίας, όπου η απόδοση συνήθως μετριέται στη διεπαφή μεταξύ του πρωτοκόλλου Point-to-Point (PPP) και το κύκλωμα μεταγωγής συνδέεται μέσω μόντεμ. Σε αυτή την περίπτωση η μέγιστη απόδοση συχνά ονομάζεται καθαρό bitrate ή χρήσιμο bitrate .

Για να προσδιοριστεί το πραγματικό ποσοστό των δεδομένων ενός δικτύου ή σύνδεσης, το "goodput" μπορεί να χρησιμοποιηθεί ως ορισμός μέτρησης. Για παράδειγμα, στον τομέα της μετάδοσης αρχείου, το "goodput" αντιστοιχεί στο μέγεθος αρχείου (σε bits) διαιρούμενο με το χρόνο μετάδοσης του αρχείου.

Η " goodput "είναι το ποσό των χρήσιμων πληροφοριών που παραδίδεται ανά δευτερόλεπτο για το επίπεδο εφαρμογών. Εξαιρούνται τα πακέτα που απορρίφθηκαν ή των αναμεταδόσεων πακέτων καθώς και γενικά το πρωτόκολλο. Εξαιτίας αυτού, το "goodput" είναι χαμηλότερο από την απόδοση.

3 Εργαλεία ενεργών μετρήσεων

Στις μέρες μας υπάρχουν πολλά εργαλεία για την υλοποίηση ενεργών μετρήσεων. Μερικά από αυτά είναι το perfSONAR, Bandwidth Test Controller (BWCTL) , NTD, pS-NPTToolkit. Στην ενότητα αυτή θα κάνουμε μια μικρή αναφορά στα εργαλεία αυτά.

3.1 perfSONAR

Το perfSONAR είναι μια εφαρμογή για την παρακολούθηση της απόδοσης του δικτύου, πράγμα που καθιστά ευκολότερο να λύσουμε end-to-end προβλήματα απόδοσης σε μονοπάτια που διασχίζουν πολλά δίκτυα. Περιέχει μια σειρά από υπηρεσίες που παρέχει μετρήσεις απόδοσης σε ένα περιβάλλον. Αυτές οι υπηρεσίες ενεργούν ως ένα ενδιάμεσο στρώμα, μεταξύ των εργαλείων μέτρησης της απόδοσης και τις διαγνωστικές ή εφαρμογές οπτικοποίησης. Αυτό το στρώμα έχει ως στόχο να καταστήσει την ανταλλαγή και τη μέτρηση της απόδοσης μεταξύ των δικτύων, με σαφώς καθορισμένα πρωτόκολλα.



Εικόνα 9: perfSONAR

Το perfSONAR είναι προσανατολισμένο στις υπηρεσίες. Αυτό σημαίνει ότι το σύνολο των στοιχειωδών λειτουργιών έχουν απομονωθεί και μπορούν να παρέχονται από διαφορετικές οντότητες που ονομάζονται υπηρεσίες. Όλες αυτές οι υπηρεσίες επικοινωνούν μεταξύ τους με τη χρήση σαφώς καθορισμένων πρωτοκόλλων. Το perfSONAR έχει τρία πλαίσια:

- Μια κοινοπραξία των οργανισμών που επιδιώκουν να χτίσουν ένα δίκτυο middleware απόδοσης που είναι διαλειτουργικό σε πολλαπλά δίκτυα και χρήσιμο για την ενδοδικτυακή και διαδίκτυακή ανάλυση. Ένας από τους κύριους στόχους είναι να καταστήσει ευκολότερη την επίλυση των end-to-end προβλήματων απόδοσης σε μονοπάτια που διασχίζουν πολλά δίκτυα.

- Ένα πρωτόκολλο. Προϋποθέτει μια σειρά από ρόλους (οι διάφοροι τύποι υπηρεσιών), και ορίζει το πρωτόκολλο πρότυπο (σύνταξη και σημασιολογία), με τον οποίο επικοινωνούν, και επιτρέπει στον καθένα να γράψει μια υπηρεσία που παίζει έναν από αυτούς τους ρόλους. Το πρωτόκολλο βασίζεται στο SOAP μηνύμα XML, και μετά το Open Grid Forum (OGF) και Μέτρηση εργασίας Δίκτυο Ομίλου (NM-WG).

- Αρκετά, πακέτα διαλειτουργικού λογισμικού (εφαρμογές των διαφόρων υπηρεσιών) που προσπαθούν να εφαρμόσουν ένα διαλειτουργικό πλαίσιο middleware peformance. Αυτές οι σειρές του κώδικα αναπτύχθηκαν από διάφορους εταίρους. Μερικά κομμάτια του κώδικα είναι «πιο σημαντικά» από άλλα, γιατί ο στόχος τους είναι να διασφαλίζουν τη διαλειτουργικότητα μεταξύ των τομέων (π.χ. Lookup Υπηρεσία και την Υπηρεσία ελέγχου ταυτότητας). Διαφορετικά υποσύνολα του κώδικα είναι σημαντικά για κάθε εταίρο, με μια μεγάλη επικάλυψη. Οι υπηρεσίες που αναπτύσσονται δρούν ως ένα ενδιάμεσο στρώμα, μεταξύ των εργαλείων μέτρησης της απόδοσης και τις διαγνωστικές ή εφαρμογές οπτικοποίησης.

3.2 Bandwidth Test Controller (BWCTL)

Το bwctl είναι μια εφαρμογή της γραμμής εντολών του πελάτη που περικλείει τα δοκιμαστικά εργαλεία Iperf, Thrulay, και Nuttcp. Αυτές οι δοκιμές μπορούν να μετρήσουν το μέγιστο εύρος ζώνης TCP, με διάφορες διαθέσιμες επιλογές ρύθμισης, ή, κάνοντας μια δοκιμή UDP, την καθυστέρηση, το jitter, και την απώλεια ενός δικτύου.



Εικόνα 10: Bandwidth Test Controller (BWCTL)

Η bwctl εφαρμογή πελάτη λειτουργεί με την επαφή μιας bwctld διεργασίας για τα δύο σημεία της δοκιμής. Το bwctl θα λειτουργήσει σε τρία μέρη. Ο πελάτης μπορεί να οργανώσει μια δοκιμή μεταξύ δύο εξυπηρετητών σε δύο διαφορετικά συστήματα. Εάν το τοπικό σύστημα προορίζεται να είναι ένα από τα άκρα της δοκιμής, το bwctl θα ανιχνεύσει κατά πόσον ένα τοπικό bwctld εκτελείται και θα χειριστεί την απαιτούμενη λειτουργικότητα

διακομιστή εάν χρειάζεται. Το bwctld διαχειρίζεται και προγραμματίζει τους πόρους του τερματικού στο οποίο λειτουργεί.

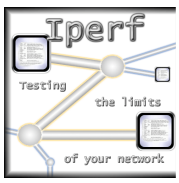
Το bwctl από τη μεριά του πελάτη χρησιμοποιείται για να ζητήσει το είδος της δοκιμής απόδοσης που θέλει. Επιπλέον, το ζητά, όταν η δοκιμή θα πρέπει να εκτελεστεί. Το bwctl απαντά είτε με μια δοκιμαστική συνεδρία είτε με ένα μήνυμα απόρριψης. Μόλις το bwctl είναι σε θέση να πάρει μια αντίστοιχη συνεδρία από τις δύο διαδικασίες bwctld (μία για κάθε υποδοχής που συμμετέχουν στη δοκιμή), επιβεβαιώνει την σύνδεση. Στη συνέχεια, οι μέθοδοι bwctld εκτελούν το τεστ και επιστρέφουν τα αποτελέσματα. Τα αποτελέσματα επιστρέφονται στον πελάτη από τις δύο πλευρές της δοκιμής. Επιπρόσθετα, οι διεργασίες bwctl μοιράζονται τα αποτελέσματα από τις αντίστοιχες πλευρές της δοκιμής με τον άλλον.

Το bwctl χρησιμοποιείται για να επιτρέψει στις μη ειδικές δοκιμές απόδοσης να συνδεθούν χωρίς να έχουν δώσει πλήρη λογαριασμό χρηστών στο συγκεκριμένο σύστημα. Οι χρήστες θέλουν να έχουν τη δυνατότητα να τρέξουν δοκιμες απόδοσης για να διαπιστωθεί το εφικτό ή το διαθέσιμο εύρος ζώνης μεταξύ ενός ζεύγους. Είναι συχνά χρήσιμο να δοκιμαστεί σε πολλαπλά σημεία κατά μήκος μίας διαδρομής δικτύου για τον προσδιορισμό των χαρακτηριστικών του δικτύου κατά μήκος αυτής της διαδρομής. Συνήθως, οι χρήστες που θέλουν να κάνουν αυτή την αποτύπωση αυτού του μονοπατιού πρέπει να επικοινωνήσουν με τους διαχειριστές του δικτύου / συστήματος που ελέγχουν τους πελάτες κατά μήκος της διαδρομής άμεσα. Ο διαχειριστής θα πρέπει να εκτελέσει είτε το μισό του τεστ για τον χρήστη ή να παρέχει ένα λογαριασμό χρήστη στον κεντρικό υπολογιστή. Επίσης, διαδρομές δικτύου με ενδιαφέρον συνήθως ελέγχονται από πολλούς διαχειριστές. Αυτά τα εμπόδια έχουν κάνει αυτό το είδος της δοκιμής να είναι δύσκολο στην πράξη.

Το bwctl σχεδιάστηκε για να βοηθήσει με αυτό το πρόβλημα. Επιτρέπει σε ένα διαχειριστή να διαμορφώσει μια δεδομένη υποδοχή ως τελικό σημείο Iperf, Thrulay ή Nuttcp που μπορεί να χρησιμοποιηθεί από πολλούς χρήστες, χωρίς την ανησυχία ότι οι χρήστες θα αλληλεπιδρούν μεταξύ τους. Ειδικά όρια μπορεί να εφαρμοστούν σε συγκεκριμένους χρήστες, έτσι οι μεμονωμένες δοκιμές δεν θα αλληλεπιδρούν μεταξύ τους. Το bwctl επιτρέπει στον διαχειριστή να ταξινομήσει τις εισερχόμενες συνδέσεις με βάση ένα όνομα χρήστη και AES κλειδί (που παράγεται από μια φράση πρόσβασης) ή συνδυασμός και των δύο, εναλλακτικά, βασίζεται σε IP/netmask. Μόλις η σύνδεση είναι διαβαθμισμένη, το bwctl μπορεί να προσδιορίσει τον ακριβή τύπο και την ένταση της απόδοσης των δοκιμών που θα επιτρέπεται.

3.2.1 Iperf

Το Iperf είναι ένα εργαλείο για τη μέτρηση του εύρους ζώνης και την ποιότητα της σύνδεσης δικτύου. Το Jperf μπορεί να συσχετιστεί με το Iperf και να παρέχει ένα γραφικό περιβάλλον γραμμένο σε Java. Η σύνδεση του δικτύου οριοθετείται από δύο τερματικα που εκτελούν το Iperf.



Εικόνα 11: Iperf

Η ποιότητα ενός συνδέσμου μπορεί να δοκιμαστεί ως εξής:

- Latency (χρόνος απόκρισης ή RTT): μπορεί να μετρηθεί με την εντολή Ping.
- Jitter (παραλλαγή του latency): μπορεί να μετρηθεί με μια δοκιμή Iperf UDP.
- Απώλεια δεδομένων: μπορεί να μετρηθεί με μια δοκιμή Iperf UDP.

Το εύρος ζώνης μετριέται μέσω TCP δοκιμών. Το Iperf χρησιμοποιεί τις διαφορετικές ικανότητες των TCP και UDP για την παροχή στατιστικών στοιχείων για τις συνδέσεις δικτύου. Τέλος, το Iperf μπορεί να εγκατασταθεί πολύ εύκολα σε οποιοδήποτε UNIX / Linux ή σύστημα Microsoft Windows. Ένα τερματικό πρέπει να ορίζεται ως πελάτης, και ένα άλλο ως διακομιστή.

3.2.2 Thrulay

Το thrulay πρόγραμμα χρησιμοποιείται για να μετρήσει την ικανότητα, καθυστέρηση, και άλλες μετρήσεις απόδοσης του δικτύου με την αποστολή ογκώδες ροών TCP ή UDP.

Ειδικά χαρακτηριστικά της thrulay περιλαμβάνουν:

- Για το πρωτόκολλο TCP, την ικανότητα να μετρήσει μετ 'επιστροφής την καθυστέρηση απόδοσης

- Για το πρωτόκολλο UDP, την ικανότητα να μετρήσει μονόδρομη καθυστέρηση, με ποσοστιαία απώλεια πακέτων και διπλώτυπων πακέτων.

3.2.3 Nuttcp

Το nuttcp είναι ένα εργαλείο μέτρησης της απόδοση του δικτύου που προορίζεται για χρήση από το δίκτυο και το σύστημα διαχείρισης. Πιο βασική χρήση του είναι να προσδιοριστεί το TCP (ή UDP) δίκτυο διακίνησης στιβάδας με μεταφορά buffers μνήμης από ένα σύστημα πηγής κατά μήκος ενός δικτύου διασύνδεσης σε ένα σύστημα προορισμού, είτε τη μεταφορά δεδομένων για ένα καθορισμένο χρονικό διάστημα, ή εναλλακτικά μεταφορά ενός συγκεκριμένου αριθμού bytes. Εκτός από την αναφορά την επιτευχθείσας απόδοσης του δικτύου σε Mbps, το nuttcp παρέχει επίσης πρόσθετες χρήσιμες πληροφορίες που σχετίζονται με τη μεταφορά δεδομένων, όπως το χρήστη, το σύστημα, το χρόνο, τη χρήση της CPU του αποστολέα και του παραλήπτη, και το ποσοστό απώλειας (για τις UDP μεταφορές).

Το nuttcp βασίζεται σε nttcp, το οποίο με τη σειρά του ήταν μια ενίσχυση από κάποιον στη Silicon Graphics (SGI), σχετικά με το αρχικό ttcp, το οποίο γράφτηκε από τον Mike Muuss στο BRL πριν τον Δεκέμβριο 1984, ώστε να συγκρίνουν την απόδοση των TCP στοιβών από UC Berkeley και BBN για να βοηθήσει την DARPA να αποφασίσει ποια έκδοση θα τοποθετήσει στην πρώτη έκδοση BSD Unix. Το nuttcp έχει πολλά χρήσιμα χαρακτηριστικά πέραν εκείνων του βασικού ttcp / nttcp, όπως μια κατάσταση λειτουργίας διακομιστή, που περιορίζει την ταχύτητα, παρέχει πολλαπλά παράλληλα ρεύματα, και το χρονόμετρο που βασίζεται στη χρήση. Οι πιο πρόσφατες αλλαγές περιλαμβάνουν υποστήριξη IPv6, IPv4 multicast, και τη δυνατότητα να οριστεί το μέγιστο μέγεθος τμήματος ή TOS / DSCP bits. Το nuttcp συνεχίζει να εξελίσσεται για να ανταποκριθούν στις νέες απαιτήσεις που θα προκύψουν και να προσθέσει νέα επιθυμητά χαρακτηριστικά. Το nuttcp έχει κατασκευαστεί με επιτυχία και λειτουργεί σε μια ποικιλία από Solaris, SGI, και PPC/X86 συστήματα Linux, και ίσως θα πρέπει να δουλεύουν μια χαρά για τις περισσότερες εκδόσεις του Unix. Επίσης, έχει χρησιμοποιηθεί με επιτυχία σε διάφορες εκδόσεις του λειτουργικού συστήματος Windows.

3.3 Network Diagnostic Tool (NDT)

Το Network Diagnostic Tool (NDT) είναι ένα client/server πρόγραμμα που παρέχει τη διαμόρφωση του δικτύου και τον έλεγχο της απόδοσης σε σταθερούς ή φορητούς υπολογιστές. Το σύστημα αποτελείται από ένα πρόγραμμα-πελάτη (γραμμή εντολών ή java applet) και ένα ζευγάρι των προγραμμάτων διακομιστή (ένα webserver και μια μηχανή δοκιμής/ανάλυσης). Και οι δύο γραμμές εντολών και web-based πελάτες επικοινωνούν με Web100-ενισχυμένο server για να εκτελέσει αυτές τις διαγνωστικές λειτουργίες. Σε πολλαπλό επίπεδο τα αποτελέσματα επιτρέπουν σε αρχάριους και έμπειρους χρήστες να βλέπουν και να κατανοούν τα αποτελέσματα της δοκιμής.

Αρκετές μελέτες έχουν δείξει ότι η πλειοψηφία των προβλημάτων απόδοσης του δικτύου εμφανίζεται σε ή κοντά σε σταθερό - φορητό υπολογιστή των χρηστών. Τα προβλήματα αυτά περιλαμβάνουν, αλλά δεν περιορίζονται σε αυτά, στην αμφίδρομη κατάσταση των Ethernet/FastEthernet συνδέσεων, μη σωστής ρύθμησης των TCP buffers στον υπολογιστή του χρήστη, ή προβλήματα με την τοπική υποδομή δικτύου. Η NDT έχει σχεδιαστεί για να εντοπίσει γρήγορα και εύκολα ένα συγκεκριμένο σύνολο συνθηκών που είναι γνωστό ότι επηρεάζουν την απόδοση του δικτύου. Μια πολυεπίπεδη σειρά από μηνύματα απλής γλώσσας, κατάλληλα για αρχάριους χρήστες, καθώς και λεπτομερή αποτελέσματα των εξετάσεων, κατάλληλα για μηχανικό δίκτυο, που παράγονται και διατίθενται στο χρήστη. Τέλος, τα αποτελέσματα της δοκιμής μπορούν εύκολα να αποσταλούν σε κατάλληλο διαχειριστή για να βοηθήσει στη φάση επίλυσης προβλήματος.

Η NDT αποτελείται από διάφορα στοιχεία. Και οι δύο διαδικασίες client και server που χρησιμοποιούνται για να εκτελεστεί μια συγκεκριμένη σειρά δοκιμών. Οι διαδικασίες του διακομιστή περιλαμβάνουν ένα βασικό πρόγραμμα περιήγησης στο Web (fakewww) για να χειριστεί τα εισερχόμενα αιτήματα web-based client. Ο διακομιστής τρέχει επίσης μια δεύτερη διαδικασία (web100srv) που εκτελεί τις συγκεκριμένες δοκιμές προκειμένου να προσδιοριστούν τα προβλήματα που ενδεχομένως θα υπάρξουν. Η διαδικασία web100srv αναλύει στη συνέχεια τα αποτελέσματα των δοκιμών και επιστρέφει τα αποτελέσματα στον πελάτη.

Τόσο η γραμμή εντολών (web100clt) και Web based (java applet) βασίζονται σε client που περιλαμβάνουν το πακέτο NDT. Ο client (web100clt) μπορεί να μεταγλωτιστεί και να εγκατασταθεί

χειροκίνητα σε πολυάριθμους υπολογιστές-πελάτες. Το web-based client χρησιμοποιεί μια βοηθητική εφαρμογή Java για την αυτοματοποίηση της διαδικασίας ελέγχου. Αυτό το applet θα κατέβει, όταν ο web server έχει πρόσβαση και άρση του περιορισμού για την χειροκίνητη φόρτωση λογισμικού στον υπολογιστή-πελάτη. Αυτή η προσέγγιση επιτρέπει στους διαχειριστές του συστήματος να προ-φορτώνουν το πρόγραμμα της γραμμής εντολών στην κύρια κατηγορία υπολογιστών server τους, επιτρέποντας ταυτόχρονα στη συντριπτική πλειοψηφία των χρηστών να φορτώσει αυτόματα το web-based client ανάλογα με τις ανάγκες τους.

3.4 pS-Performance Toolkit

Το PS-Performance Toolkit είναι ένα προϊόν του pS-PS αναπτυξιακού έργου. Παρέχεται ο ευκολότερος τρόπος για να αναπτυχθεί ένα δίκτυο perfSONAR που μετράει την απόδοση στους κόμβους. Περιλαμβάνει μια σειρά από διαγνωστικά εργαλεία δικτύου για τον εντοπισμό σφαλμάτων, έλεγχο της απόδοσης του δικτύου και μπορεί να χρησιμοποιηθεί χωρίς την εργασία διαμόρφωσης. Με ένα μικρό ποσό της διαμόρφωσης χρησιμοποιώντας ένα web-based γραφικό περιβάλλον χρήστη, το PS-Performance Toolkit τερματικό μπορεί να τρέξει τακτικές εξετάσεις και να παρέχει μια συνεχή καταγραφή της απόδοσης του δικτύου.

Το PS-Performance Toolkit χρησιμοποιείται αυτή τη στιγμή από το έργο USATLAS για την επαλήθευση της απόδοσης του δικτύου μεταξύ LHC Tier 1 και Tier2 ιστοσελίδων, και έχει αρχίσει να χρησιμοποιείται από ορισμένους Tier3 ιστοσελίδες για να ελέγχονται οι επιδόσεις του δικτύου σε Tier2 ιστοσελίδες. Το PS-Performance Toolkit είναι μια ιδιαίτερα ελκυστική επιλογή για εικονικές κοινότητες όπως LHC, διότι ελαχιστοποιεί την κατανόηση, από τις ερευνητικές ομάδες, της τεχνολογίας του δικτύου επιτρέποντάς τους να επικεντρωθούν στην πραγματική επιστήμη που προσπαθούν να επιτύχουν.

Η αρχική ιδέα για το PS-Performance Toolkit ήρθε από την ομάδα διαχείρησης δικτύων Internet2 Workshop. Η αυθεντική μορφή των εργαστηρίων αυτών αποτελούνταν από την ημέρα της εγκατάστασης του εργαλείου (π.χ. εγκατάσταση του Web100 ενισχυμένου πυρήνα και άλλα εργαλεία απόδοσης) ακολουθούμενο μερικές φορές από τη δοκιμή και τη χρησιμοποίηση τους. Αυτή η συγκεκριμένη μορφή δεν αφήνει αρκετό χρόνο για να φτάσουμε στην κύρια αιτία ώστε πολλοί από τους συμμετέχοντες να επιλέξουν να εγγραφούν, την επίλυση των προβλημάτων απόδοσης του δικτύου. Για να γίνουν αυτά τα εργαστήρια πιο χρήσιμα για τους συμμετέχοντες, το προσωπικό του Internet2, εξέτασαν τις μεθόδους που θα τους επιτρέψει την πρόσβασης στα εργαλεία που χρησιμοποιούνται στην αντιμετώπιση προβλημάτων του δικτύου.

4. OWAMP

Η ομάδας IETF IP Performance Metrics (IPPM) έχει ορίσει τις μετρήσεις για την καθυστέρηση μονόδρομων πακέτων [RFC2679] και την απώλεια [RFC2680] σε διαδρομές Διαδικτύου. Παρά το γεγονός ότι υπάρχουν πλέον πολλές πλατφόρμες μέτρησης που συλλέγουν αυτές τις μετρήσεις, δεν υπάρχει σήμερα ένα πρότυπο που θα επιτρέπει την έναρξη των ρευμάτων δοκιμής ή την ανταλλαγή των πακέτων για τη συλλογή μετρήσεων με λειτουργικό τρόπο.

Με την όλο και πιο ευρεία διαθεσιμότητα των οικονομικά προσιτών παγκοσμίως σε συστήματα εντοπισμού θέσης (GPS) και CDMA με βάση τις πηγές του χρόνου, οι οικοδεσπότες έχουν όλο και περισσότερο στη διάθεσή τους ακριβή χρόνο, είτε άμεσα είτε μέσω της εγγύτητάς τους προς το Πρωτόκολλο Δικτυακού Χρόνου (NTP). Με την τυποποίηση της τεχνικής για τη συλλογή IPPM μονόδρομων ενεργών μετρήσεων, ελπίζουμε να δημιουργήσουμε ένα περιβάλλον όπου οι IPPM μετρήσεις μπορούν να συλλεχθούν σε ένα ευρύτερο πλέγμα των διαδρομών στο Διαδίκτυο από ό,τι είναι σήμερα δυνατό. Ένα ιδιαίτερα συναρπαστικό όραμα είναι η ευρεία διάδοση των ανοικτών servers OWAMP που θα κάνουν τη μέτρηση της μονόδρομης καθυστέρησης τόσο απλή όσο η μέτρηση του χρόνου μετ 'επιστροφής με ένα ICMP-based εργαλείο όπως το ping.

Πρόσθετοι στόχοι του σχεδιασμού του OWAMP περιλαμβάνουν τη δυσκολία να ανιχνεύσουν και να χειραγωγήσουν, την ασφάλεια, το λογικό διαχωρισμό του ελέγχου και τη λειτουργικότητα των δοκιμών, καθώς και τη στήριξη των μικρών πακέτων δοκιμής. (Όντας δύσκολο να ανιχνεύσει παρεμβολές με μετρήσεις εξίσου δύσκολο είναι σε ενδιάμεσους κόμβους του δικτύου) Η OWAMP δοκιμαστική κυκλοφορία είναι δύσκολο να ανιχνευθεί, επειδή είναι απλώς ένα ρεύμα UDP πακέτων από και προς διαπραγματεύσιμους αριθμούς θυρών, με μη σταθερά πακέτα (μέγεθος).

Το OWAMP επίσης υποστηρίζει ένα κρυπτογραφημένο τρόπο που επισκιάζει περισσότερο την κυκλοφορία και καθιστά αδύνατο να αλλάξει timestamps χωρίς ανίχνευση. Τα χαρακτηριστικά ασφαλείας περιλαμβάνουν προαιρετική ταυτότητα και/ή κρυπτογράφηση και έλεγχο μηνυμάτων δοκιμής. Αυτά τα χαρακτηριστικά μπορεί να είναι χρήσιμα ώστε να αποτραπεί η μη εξουσιοδοτημένη πρόσβαση στα αποτελέσματα ή εσκεμμένες επιθέσεις όπου προσπαθούν να παρέχουν ειδική μεταχείριση στα OWAMP ρεύματα δοκιμής ή που επιχειρούν να τροποποιήσουν τον αποστολέα που δημιούργησε τα timestamps για να παραποιήσουν τα αποτελέσματα της δοκιμής.

Το OWAMP είναι μια εφαρμογή της γραμμής εντολών του πελάτη και χρησιμοποιείται για να προσδιορίσει έναν τρόπο latencies μεταξύ των κεντρικών υπολογιστών. Πρόκειται για μια εφαρμογή του πρωτοκόλλου OWAMP όπως ορίζεται από το http://www.rfc-editor.org/rfc/rfc4656.txt.

Με τις αμφίδρομες μετρήσεις, είναι δύσκολο να απομονωθεί η κατεύθυνση στην οποία βιώνεται η συμφόρηση. Οι μονόδρομες μετρήσεις λύνουν αυτό το πρόβλημα και κάνουν την κατεύθυνση της συμφόρησης αμέσως εμφανή. Δεδομένου ότι η κίνηση μπορεί να είναι ασύμμετρη σε πολλές περιοχές που είναι κυρίως παραγωγοί ή καταναλωτές των δεδομένων, αυτό παρέχει πιο χρήσιμες μετρήσεις. Οι μονόδρομες μετρήσεις επιτρέπουν στο χρήστη να απομονώσει καλύτερα τα αποτελέσματα των συγκεκριμένων τμημάτων ενός δικτύου για την αντιμετώπιση της κυκλοφορίας.

Με το One-Way Active Measurements Protocol (OWAMP), οι πάροχοι δικτύου θα είναι σε θέση να γνωρίζουν καλύτερα την ακριβή συμπεριφορά των δικτύων τους και να ζητήσουν πόρους, όπου η βελτίωση είναι πιο πιθανή. (Σημείωση: Η παθητική παρατήρηση ενός συνδέσμου χάνει τις παροδικές ουρές. Η ενεργή μέτρηση θα μπορούσε να τους δει.) Οι χρήστες θα είναι περισσότερο ενημερωμένοι σχετικά με την απόδοση του δικτύου. Αυτό θα οδηγήσει σε μια καλύτερη κατανομή των πόρων από τους παρόχους δικτύου, μείωση της κυκλοφοριακής συμφόρησης σε περιοχές όπου είναι δυνατόν.

Η αυξανόμενη διαθεσιμότητα των πηγών για ακριβή χρόνο επιτρέπει στα δίκτυα να φιλοξενούν πακέτα timestamp με τα τυπικά σφάλματα που είναι μικρότερα από τις καθυστερήσεις που παρατηρούνται σε πραγματικά μη-LAN δίκτυα. Αυτό καθίσταται δυνατό για μονόδρομες μετρήσεις που πρέπει να συλλέγονται σε ένα ευρύ πλέγμα των διαδρομών του δικτύου. Επιπλέον, ο ανοικτός κώδικας του OWAMP καθιστά δυνατό η μέτρηση να γίνει τόσο κοινή όσο μία μετ' επιστροφής μέτρηση (εργαλεία όπως το ping).

Το πρωτόκολλο OWAMP απλοποιεί επίσης την ανάλυση των αποτελεσμάτων των μετρήσεων. Η αποστολή και λήψη timestamps για κάθε πακέτο μέτρησης κάνουν την ανάλυση πιο απλή, γιατί δεν χρειάζεται να αναλάβει την επιστροφή και τη διατήρηση των μεταξύ πακέτων απόστασης. Για παράδειγμα, η αναδιάταξη πακέτων, η οποία μπορεί να έχει συνέπειες στην απόδοση του TCP, μπορεί να μετρηθεί υπό ποικίλα σενάρια εισόδου, με διαχωρισμό της αναδιάταξης για τα προς τα εμπρός και επιστροφής μονοπάτια.

Ο OWAMP έλεγχος συνεδρίας χρησιμοποιεί παραδοσιακή client-server επικοινωνία μεταξύ ελέγχου-πελάτη και ενός εξυπηρετητή, χρησιμοποιώντας το OWAMP-Control πρωτόκολλο. Ο διακομιστής owampd υλοποιείται χρησιμοποιώντας το συμβατικό μοντέλο αποδοχής. Οι δύο πλευρές διαπραγματεύονται με ένα τρόπο τις παραμέτρους της δοκιμής χρησιμοποιώντας την OWAMP-Control πρωτόκολλο. Αυτή η εφαρμογή OWAMP τότε ωθεί πρόσθετες διαδικασίες να στέλνουν και να λαμβάνουν τα OWAMP-Test διαμορφωμένα πακέτα που εφαρμόζουν τα τελικά σημεία συνεδρίασης για τον Αποστολέα και Δέκτη.

Χρησιμοποιώντας OWAMP, είναι δυνατή η συλλογή δεδομένων ενεργής μέτρησης επαρκή για να καθορίσει μια ευρεία κατηγορία των μοναδικών χαρακτηριστικών (π.χ., πιθανότητα απώλειας, η μέση καθυστέρηση, το jitter). Μη-μοναδικά χαρακτηριστικά, όπως το αναμενόμενο κενό μεταξύ αφίξεως των πακέτων που εστάλησαν το ένα πίσω από το άλλο, μπορεί να μετρηθεί επίσης. Σημείωση: Όλες οι μετρήσεις γίνονται με συνθετικό κυκλοφορίας. Η προσομοίωση εφαρμογής είναι εκτός του πεδίου εφαρμογής του OWAMP. Το πρωτόκολλο δεν έχει σχεδιαστεί για να είναι σε θέση να στείλει ένα πακέτο, μόλις η απάντηση του προηγούμενου πακέτου φτάσει, αλλά μπορεί να στείλει σε κάθε προκαθορισμένο χρονικό διάστημα (συμπεριλαμβανομένου του πακέτου που στάλθηκε αμέσως μετά το τελευταίο).

Το OWAMP έχει σχεδιαστεί για να είναι ικανό να αναπτυχθεί σε όσο το δυνατόν περισσότερα συστήματα. Ακριβώς όπως είναι δυνατό να κάνετε ping από τους περισσότερους υπολογιστές του δικτύου σήμερα, η ευρεία διάδοση του OWAMP θα καταστήσει δυνατή τη διεξαγωγή πιο ακριβή μετρήσεων.

Η ομάδα ανάπτυξης του OWAMP αναγνωρίζει ότι τα συστήματα μέτρησης του δικτύου έχουν γίνει πιο δυσκίνητα όσο το μέγεθός τους αυξάνεται. Όταν μία πλήρες αρχιτεκτονική μέτρησης χρησιμοποιείται, η ποσότητα του χώρου στο δίσκο και η χωρητικότητα του δικτύου που χρησιμοποιείται από το σύστημα θα αυξηθεί ως το τετράγωνο του αριθμού των κόμβων μέτρησης. Αν και τίποτα δεν μπορεί να γίνει για να αντιμετωπιστεί αυτό το πρόβλημα, το OWAMP είχε σχεδιαστεί ώστε να μην εισάγονται νέα προβλήματα κλιμάκωσης. Επιτρέπει στο χρήστη να διεξάγει μόνο τις επιθυμητές συνεδρίες μέτρησης, και να διατηρήσει όσο (ή λίγο) θέλει τα δεδομένα. Το OWAMP επίσης δεν απαγορεύει την επιλογή της τοποθεσίας, όπου αποθηκεύονται τα αποτελέσματα της μέτρησης: είναι δυνατόν να είναι όλα τα δεδομένα αποθηκευμένα σε μία κεντρική τοποθεσία ή να αποθηκεύει δεδομένα σε κάθε δέκτη και fetch όπως χρειάζεται.

Η owping client χρησιμοποιείται για να ζητήσει μια συνεδρία μέτρησης. Οι παράμετροι owping επιτρέπουν στο χρήστη να επιλέξει το πρόγραμμα αποστολής, την κατεύθυνση της δοκιμής, καθώς και το μέγεθος του πακέτου. Η owping εφαρμογή επικοινωνεί με μια διαδικασία owampd ώστε να ζητήσει τη συγκεκριμένη μονόδρομη δέσμη μετρήσεων. Το owampd είναι υπεύθυνο για την εφαρμογή των περιορισμών που επιβάλλονται από την πολιτική του διαχειριστή του συστήματος σε αυτό το σύστημα.

Το owampd επιτρέπει σε ένα διαχειριστή συστήματος να διαμορφώσει οποιαδήποτε δεδομένη υποδοχή ως τελικό σημείο της δοκιμής OWAMP. Ειδικά όρια πολιτικής μπορεί να εφαρμοστούν σε συγκεκριμένους χρήστες, και οι μεμονωμένες δοκιμές συντονίζονται. Έτσι δεν θα παρεμβαίνουν μεταξύ τους. Το OWAMP επιτρέπει στον διαχειριστή να ταξινομήσει τις εισερχόμενες συνδέσεις με βάση ένα όνομα χρήστη και το AES κλειδί (που παράγεται από μια φράση πρόσβασης) ή συνδυασμό αυτών των δύο, εναλλακτικά, βασίζεται σε IP/μάσκα. Μόλις η σύνδεση είναι διαβαθμισμένη, το owampd μπορεί να καθορίσει τις ακριβείς παραμέτρους της δοκιμής που θα πρέπει να επιτρέπεται.

4.1 Χαρακτηριστικά

- Πλήρης υποστήριξη IPv6. Δεν υπάρχουν επιλογές που απαιτούνται. Αν ο στόχος της δοκιμής προσδιορίζεται από ένα όνομα κεντρικού υπολογιστή DNS, και το όνομα ορίζεται και με IPv4 και με IPv6 διεύθυνση, η owping εφαρμογή της γραμμής εντολών προτιμά τη διεύθυνση IPv6.

- Το χρονοδιάγραμμα και το μέγεθος των πακέτων, διαμορφώνεται από τον πελάτη.

- Προστασία των πόρων, όπως ορίζεται από την πολιτική ελέγχων που εφαρμόζονται στο owampd.

- Ο καθορισμός των θυρών για τα πακέτα που αποστέλλονται ώστε να είναι αποδεκτό από το τείχος προστασίας.

- Ευρύ φάσμα των στατιστικών.

4.2 Απαιτήσεις

Το OWAMP προτιμά ένα συγχρονισμένο ρολόι ώστε οι μετρήσεις να έχουν νόημα. Όμως, το πιο σημαντικό, το ρολόι πρέπει να είναι σταθερό. Αν το ρολόι του συστήματος έχει αποκλίσεις κατά τη διάρκεια μιας συνεδρίας OWAMP, τα αποτελέσματα μπορεί να είναι παραπλανητικά.

Το OWAMP προτιμά το NTP (ntpd) για να συγχρονίσει το ρολόι του συστήματος. Το NTP πρέπει να ρυθμιστεί σωστά στο σύστημα ώστε να υπολογίσει μια λογική εκτίμηση του σφάλματος χρόνου και να σταθεροποιήσει το ρολόι. Το NTP πρέπει να ρυθμιστεί με όχι λιγότερα από 4 ρολόγια για το σκοπό αυτό. Ορισμένες μετρήσεις θα εξακολουθούν να έχουν νόημα αν δεν είναι συγχρονισμένα τα ρολόγια. Για παράδειγμα, οι μετρήσεις jitter εξακολουθούν να ισχύουν.

Αξίζει να σημειωθεί ότι το OWAMP θα πρέπει να τρέχει σε πραγματικό υλικό. Εικονικά πακέτα όπως το VMware, Xen, linux kvm, παράλληλα θα παρουσιάζουν αστάθεια στο ρολόι που θα κάνει τις OWAMP μετρήσεις πιο χαοτικές.

Επιπλέον, η διαχείριση ενέργειας των περισσότερων υλικών του υπολογιστή θα πρέπει να απενεργοποιηθεί. (Επιτάχυνση και επιβράδυνση του επεξεργαστή προκαλεί αστάθεια στο ρολόι.)

4.3 Σχέση της δοκιμής και τα πρωτόκολλα ελέγχου

Το OWAMP αποτελείται στην πραγματικότητα από δύο αλληλένδετα πρωτόκολλα: το OWAMP-Control και το OWAMP-Test. Το OWAMP-Control χρησιμοποιείται για την εισαγωγή, εκκίνηση, και τη διακοπή της συνεδρίας δοκιμής και για να φέρει τα αποτελέσματά τους, ενώ το OWAMP-Test χρησιμοποιείται για την ανταλλαγή των πακέτων ελέγχου μεταξύ δύο κόμβων μέτρησης. Μολονότι το OWAMP-Test μπορεί να χρησιμοποιηθεί σε συνδυασμό με ένα στοιχείο ελέγχου πρωτοκόλλου, εκτός από το OWAMP-Control, οι συγγραφείς έχουν σκόπιμα επιλέξει να περιλάβει και τα δύο πρωτόκολλα με τον ίδιο RFC για να ενθαρρυνθεί η εφαρμογή και η ανάπτυξη του OWAMP-Control με κοινό παρονομαστή τον έλεγχο του πρωτοκόλλου για τις μονόδρομες μετρήσεις. Έχοντας μια πλήρη και ανοικτή λύση στη μονόδρομη μέτρηση που είναι απλή στην εφαρμογή και την ανάπτυξη ζωτικής σημασίας για την εξασφάλιση ενός μέλλοντος στο οποίο οι μονόδρομες ενεργές μετρήσεις θα μπορούσαν να γίνουν ως ένα κοινό ping.

Το OWAMP-Control έχει σχεδιαστεί για να υποστηρίξει τη διαπραγμάτευση μονόδρομων ενεργών περιόδων μέτρησης και των αποτελεσμάτων ανάκτησης με ένα άμεσο τρόπο. Κατά την έναρξη της συνεδρίας, υπάρχει διαπραγμάτευση των διευθύνσεων του αποστολέα και του παραλήπτη καθώς και στους αριθμούς θυρών, την ώρα έναρξης της συνεδρίας, τη διάρκεια συνεδρίας, το μέγεθος του πακέτου δοκιμής, το μέσο διάστημα δειγματοληψίας για το ρεύμα δοκιμής, καθώς και ορισμένα χαρακτηριστικά της.

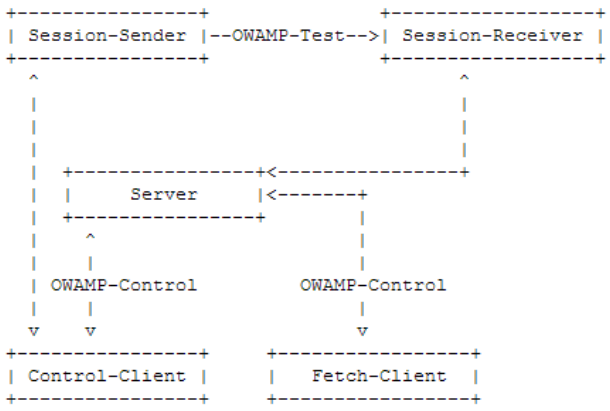
Η πολύ γενική έννοια του τύπου πακέτου [RFC 2330], συμπεριλαμβανομένου το μέγεθος των πακέτων και τη συμπεριφορά (PHB) [RFC2474], η οποία θα μπορούσε να χρησιμοποιηθεί για την υποστήριξη της μονόδρομης μέτρησης των χαρακτηριστικών του δικτύου. Επιπλέον, το OWAMP-Control υποστηρίζει την ανά σύνοδο κρυπτογράφηση και έλεγχο ταυτότητας τόσο για δοκιμή όσο και για έλεγχο της κυκλοφορίας, διακομιστές μέτρησης που μπορούν να λειτουργήσουν ως διαμεσολαβητές για τη δοκιμή των τελικών ρευμάτων, και η ανταλλαγή τιμών για την ψευδο-τυχαία διαδικασία Poisson που περιγράφει το ρεύμα που παράγεται από δοκιμή του αποστολέα.

Πιστεύουμε ότι το OWAMP-Control μπορεί αποτελεσματικά να υποστηρίξει ένα ενεργό τρόπο μέτρησης σε μία ποικιλία από περιβάλλοντα, από δημοσίως προσβάσιμους σταθμούς μέτρησης που λειτουργούν για την παρακολούθηση του δικτύου ανάπτυξης σε ιδιωτικά εταιρικά δίκτυα. Αν η ολοκλήρωση με Απλό Πρωτόκολλο Διαχείρισης Δικτύου (SNMP) ή πρωτόκολλο διαχείρισης ιδιόκτητων δικτύων απαιτείται, πύλες μπορούν να δημιουργηθούν.

4.4 Λογικό Μοντέλο

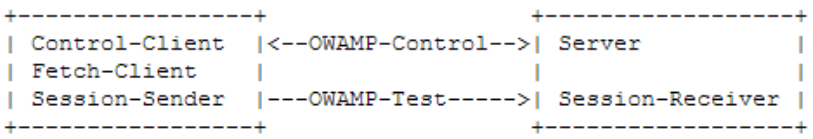
Αρκετοί ρόλοι είναι λογικά διαχωρισμένοι ώστε να καταστεί δυνατή η ευρεία ευελιξία σε χρήση. Συγκεκριμένα, ορίζουμε τα ακόλουθα:

- Session-Sender: Η αποστολή στο τελικό σημείο μιας συνεδρίας OWAMP-Test,
- Session-Receiver: Η λήψη στο τελικό σημείο μιας συνεδρίας OWAMP-Test,
- Server: Ένα σύστημα διακομιστή που διαχειρίζεται μία ή περισσότερες OWAMP-Test συνεδρίες, είναι ικανή να διαμορφώνει την ανά περίοδο κατάσταση σε τελικές συνεδρίες, και είναι σε θέση να επιστρέφει τα αποτελέσματα μιας συνόδου δοκιμής,
- Control-client: Ένα σύστημα το οποίο ξεκινά για τις αιτήσεις OWAMP-Test συνεδρίες, πυροδοτεί την εκκίνηση ενός συνόλου των συνεδριών, και μπορεί να προκαλέσει τη λήξη τους,
- Fetch-client: Ένα σύστημα το οποίο ξεκινά τα αιτήματα που φέρουν τα αποτελέσματα των ολοκληρωμένων OWAMP-Test συνεδριών.



Εικόνα 13: Ένα πιθανό σενάριο των σχέσεων μεταξύ αυτών των ρόλων

Διαφορετικοί λογικοί ρόλοι μπορεί να παίξουν με την ίδια σειρά. Για παράδειγμα, στο παραπάνω σχήμα, θα μπορούσαν να υπάρχουν στην πραγματικότητα μόνο δύο host: ο ένας παίζει το ρόλο του control-client, Fetch-client, και Session-Sender, και ο άλλος παίζει το ρόλο του Server και Session-Receiver. Αυτό φαίνεται παρακάτω.



Εικόνα 14: Απλουστευμένο σενάριο των σχέσεων μεταξύ αυτών των ρόλων

Τέλος, επειδή πολλά μονοπάτια στο Διαδίκτυο περιλαμβάνουν τμήματα που οι μεταφορές IP είναι πάνω από ATM, καθυστέρηση και απώλεια στις μετρήσεις μπορεί να περιλαμβάνουν τα αποτελέσματα του ATM τεμαχισμού και της επανασυναρμολόγησης (SAR). Κατά συνέπεια, το

OWAMP έχει σχεδιαστεί για να επιτρέπει σε μικρά πακέτα δοκιμής που θα ταιριάζουν στο εσωτερικό του ωφέλιμο φορτίο ενός ATM κελιού (αυτό επιτυγχάνεται μόνο σε χωρίς έλεγχο λειτουργία).

4.5 Επισκόπηση Πρωτοκόλλου

Όπως περιγράφεται παραπάνω, το OWAMP αποτελείται από δύο αλληλένδετα πρωτόκολλα: το OWAMP-Control και OWAMP-Test. Το πρώτο είναι σε επίπεδα πάνω από το TCP και χρησιμοποιείται για την κίνηση και τον έλεγχο συνεδρίας μέτρησης και τη λήψη των αποτελεσμάτων. Το τελευταίο πρωτόκολλο είναι σε επίπεδα πάνω από το UDP και χρησιμοποιείται για την αποστολή μονήρων μετρήσεων πακέτων κατά μήκος της διαδρομής Διαδίκτυο υπό δοκιμή.

Ο εμπνευστής της δέσμης μετρήσεων δημιουργεί μια σύνδεση TCP σε μία γνωστή θύρα, την 861, σχετικά με το σημείο στόχο και αυτό το πλαίσιο παραμένει ανοικτό για τη διάρκεια των OWAMP-Test συνεδριών. Ο OWAMP-Server πρέπει να παρακολουθεί αυτή τη θύρα. Τα OWAMP-Control μηνύματα μεταδίδονται μόνο πριν το OWAMP-Test. Οι συνεδρίες πραγματικά ξεκινάνε μέχρι να ολοκληρωθούν (με την πιθανή εξαίρεση του πρόωρου Stop-Session μήνυμα).

Τα OWAMP-Control και OWAMP-Test πρωτόκολλα υποστηρίζουν τρεις τρόπους λειτουργίας: χωρίς έλεγχο ταυτότητας, επικυρωμένα, και κρυπτογραφημένα. Οι επικυρωμένες ή κρυπτογραφημένες λειτουργίες απαιτούν ότι τα τελικά σημεία έχουν ένα κοινό μυστικό.

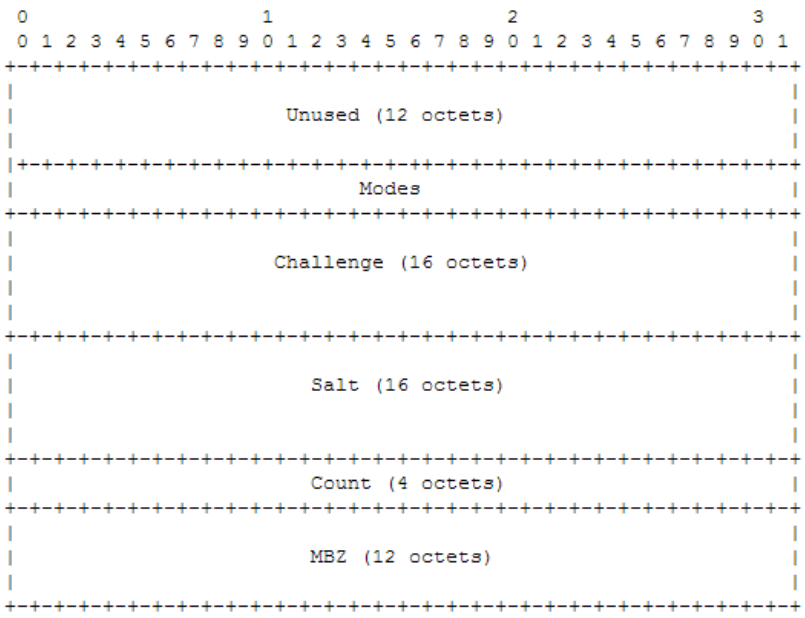
4.6 OWAMP-Control

Ο τύπος του κάθε μηνύματος του OWAMP-Control μπορεί να βρεθεί μετά από την ανάγνωση των πρώτων 16 οκτάδων. Το μήκος του κάθε μηνύματος του OWAMP-Control μπορεί να υπολογιστεί κατά την ανάγνωση σταθερού μεγέθους μέρους του. Δεν είναι μικρότερο μήνυμα από 16 οκτάδες.

Η εφαρμογή θα πρέπει να εξαλείψει αχρησιμοποίητες καταστάσεις για να προβλέψει εκτός υπηρεσιών επιθέσεις, ή απεριόριστη χρήση της μνήμης, στον server. Για παράδειγμα, αν το πλήρες μήνυμα ελέγχου δεν παραληφθεί εντός ορισμένου χρονικού διαστήματος μετά το αναμενόμενο, η σύνδεση TCP που σχετίζεται με την OWAMP-Control συνεδρία θα πρέπει να εγκαταλειφθεί.

Πριν το OWAMP-Control ή το Fetch-Client εκδώσει εντολές σε ένα διακομιστή, πρέπει να δημιουργηθεί μια σύνδεση με το διακομιστή.

Πρώτον, ένας πελάτης ανοίγει μια σύνδεση TCP με το διακομιστή σε μια γνωστή θύρα την 861. Ο διακομιστής απαντά με ένα χαιρετισμό διακομιστή:



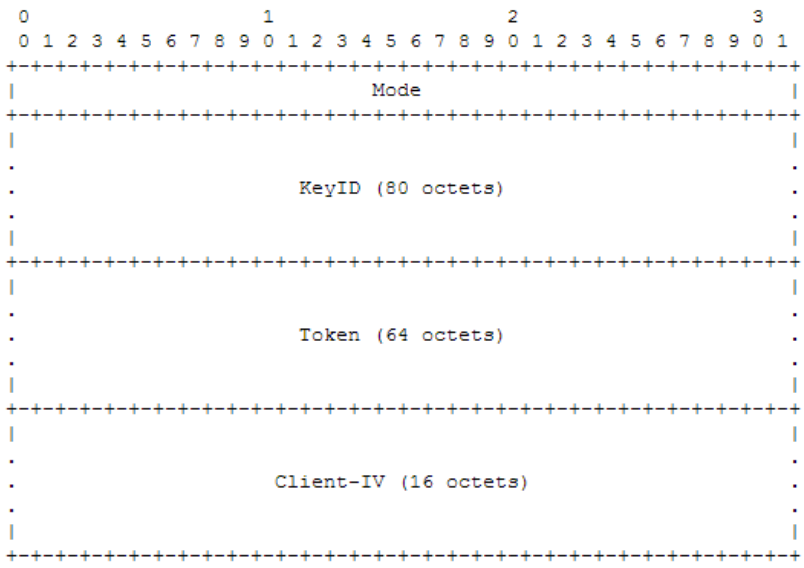
Εικόνα 15: Μήνυμα χαιρετισμού του διακομιστή

Οι ακόλουθες τιμές για το πεδίο Mode είναι ουσιαστικές, για έλεγχο χωρίς ταυτότητα με την τιμή 1, για έλεγχο ταυτότητας με την τιμή 2, και για κρυπτογραφημένα μηνύματα με την τιμή 4. Η τιμή του πεδίου Mode αποστέλλεται από το διακομιστή ή με ενα bit-wise ή από την τιμή του πεδίου που είναι πρόθυμη να υποστηρίξει κατά τη διάρκεια αυτής της συνόδου. Έτσι, τα τελευταία τρία ψηφία του πεδίου Mode χρησιμοποιούνται από τα 32-bit του πεδίου αυτού. Τα πρώτα 29 bits ΠΡΕΠΕΙ να είναι μηδέν. Ένας πελάτης πρέπει να αγνοήσει τις τιμές στα πρώτα 29 bits από αυτές τις καταστάσεις. (Με αυτό τον τρόπο, τα δυαδικά ψηφία είναι διαθέσιμα για μελλοντική επέκταση του πρωτοκόλλου. Αυτός είναι ο μόνος μηχανισμός προβλεπόμενης επέκτασης.)

Το πεδίο Challenge είναι μια τυχαία ακολουθία οκτάδων που παράγονται από το διακομιστή. Στη συνέχεια, χρησιμοποιείται από τον πελάτη για να αποδείξει την κατοχή ενός κοινού μυστικού με ένα τρόπο που προβλέπεται παρακάτω.

Τα πεδία Salt και Count χρησιμοποιούνται κατά τον υπολογισμό ενός κλειδιού από ένα κοινό μυστικό όπως περιγράφεται παρακάτω. Το πεδίο Salt πρέπει να δημιουργείται ψευδο-τυχαία.

Εάν η τιμή του πεδίου mode είναι μηδέν, ο διακομιστής δεν επιθυμεί να επικοινωνήσει με τον πελάτη και μπορεί να κλείσει τη σύνδεση αμέσως. Ο πελάτης πρέπει να κλείσει τη σύνδεση εάν λάβει ένα χαιρετισμό με την τιμή του πεδίου mode ίση με το μηδέν. Ο πελάτης μπορεί να κλείσει τη σύνδεση, αν η επιθυμητή λειτουργία δεν είναι διαθέσιμη. Σε αντίθετη περίπτωση, ο πελάτης πρέπει να απαντήσει με το εξής απαντητικό μήνυμα.



Εικόνα 16: απαντητικό μήνυμα πελάτη

Εδώ το πεδίο Mode είναι η επιλογή όπου ο πελάτης επιλέγει να χρησιμοποιήσει κατά τη διάρκεια αυτής της OWAMP συνεδρίας Ελέγχου. Θα μπορούσε επίσης να χρησιμοποιηθεί για όλα τα OWAMP τεστ που ξεκίνησαν υπό τον έλεγχο αυτής της συνόδου ελέγχου OWAMP. Σε Mode, με τιμή 0 ή 1 bit πρέπει να οριστούν τα τελευταία τρία κομμάτια. Αν είναι 1 bit που βρίσκεται μέσα στα τελευταία τρία bits, αυτό το κομμάτι πρέπει να δείχνει ότι η λειτουργία του διακομιστή συμφώνησε (δηλαδή, το ίδιο κομμάτι πρέπει να έχει τεθεί από τον server στον χαιρετισμό του διακομιστή). Τα πρώτα 29 bits του Mode πρέπει να είναι μηδέν. Ένας διακομιστής πρέπει να αγνοεί τις τιμές των πρώτων 29 bits. Αν είναι μηδέν bits η λειτουργία καθορίζεται από τον πελάτη, ο πελάτης δηλώνει ότι δεν θα συνεχίσει με τη σύνοδο. Στην περίπτωση αυτή, ο πελάτης και ο διακομιστής πρέπει να κλείσουν τη σύνδεση TCP που σχετίζονται με τη σύνοδο του OWAMP ελέγχου.

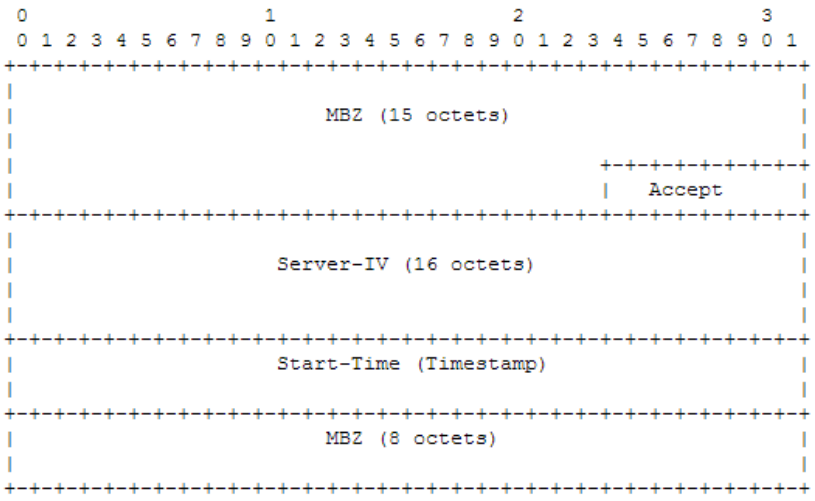
Σε μη έλεγχο ταυτότητας, τα KeyID, Token, και ο Clilent-IV δεν χρησιμοποιούνται. Σε αντίθετη περίπτωση, το KeyID είναι UTF-8 string, από 80 οκτάδες και πάνω σε μήκος (αν το string είναι μικρότερο, προστίθεται με μηδέν octets), όπου ορίζει το διακομιστή που μοιράζει το κοινό μυστικό που ο πελάτης επιθυμεί να χρησιμοποιήσει για τον έλεγχο ταυτότητας ή την κρυπτογράφηση, ενώ το Token είναι η συνένωση των 16-οκτάδων, μια 16-οκτάδα με το AES κλειδί συνόδου που χρησιμοποιείται για την κρυπτογράφηση, και μια 32-οκτάδα με το HMAC-SHA1 κλειδί συνόδου που χρησιμοποιείται για έλεγχο ταυτότητας. Το Token είναι από μόνο του κρυπτογραφημένο χρησιμοποιώντας το AES (Advanced Encryption Standard) [AES] σε Cipher Block Chaining (CBC). Η κρυπτογράφηση θα πρέπει να πραγματοποιείται με τη χρήση σε Διάνυσμα αρχικοποίησης (IV) του μηδέν και ένα κλειδί που προέρχεται από το κοινόχρηστο μυστικό που σχετίζεται με το KeyID. (Τόσο ο διακομιστής όσο και ο πελάτης χρησιμοποιούν την ίδια αντιστοιχία από το ίδιο keyid σε κοινά μυστικά. Ο διακομιστής έχει προετοιμαστεί για να διεξαγάγει συνεδρίες με περισσότερους από έναν πελάτες και χρησιμοποιεί keyid για να επιλέξει το μυστικό κλειδί. Συνήθως ο πελάτης έχει διαφορετικά μυστικά κλειδιά για διαφορετικούς servers. Η κατάσταση είναι Ανάλογη με εκείνη με τους κωδικούς πρόσβασης).

Το κοινό μυστικό είναι ένα συνθηματικό και δεν πρέπει να περιέχει νέες γραμμές. Το μυστικό κλειδί προέρχεται από τη συνθηματική φράση χρησιμοποιώντας ένα κωδικό πρόσβασης με βάση το κλειδί παραγωγής λειτουργίας PBKDF2 (PKCS # 5) [RFC2898]. Η λειτουργία PBKDF2 απαιτεί αρκετές παραμέτρους: η PRF είναι HMAC-SHA1 [RFC2104], το solt και το count είναι όπως μεταδίδονται από τον διακομιστή.

Το AES κλειδί συνόδου, HMAC Session-key και Client-IV είναι που δημιουργούνται τυχαία από τον πελάτη. Το AES κλειδί συνόδου και το HMAC-κλειδί πρέπει να παράγεται με αρκετή

εντροπία ώστε να μην μειωθεί η ασφάλεια του υποκείμενου cipher [RFC4086]. Το Client-IV πρέπει απλώς να είναι μοναδικό (Δηλαδή, δεν πρέπει ποτέ να επαναληφθεί για διάφορες συνεδρίες με τη χρήση του ίδιου μυστικού κλειδιού. Ένας απλός τρόπος για να επιτευχθεί αυτό χωρίς τη χρήση περίπλοκων καταστάσεων είναι να δημιουργήσει ο Client-IV τιμές χρησιμοποιώντας ένα ασφαλές κρυπτογραφημένο ψευδο-τυχαίο κώδικα. Εάν αυτό γίνει, η πρώτη επανάληψη είναι απίθανο να συμβεί πριν από 2 ^ 64 συνεδρίες με το ίδιο μυστικό κλειδί).

Ο διακομιστής πρέπει να απαντήσει με το ακόλουθο μήνυμα εκκίνησης διακομιστή:



Εικόνα 17: Μήνυμα εκκίνησης διακομιστή

Το πεδίο MBZ πρέπει να είναι μηδέν. Ο πελάτης πρέπει να αγνοήσει την τιμή του. Το πεδίο MBZ (πρέπει να είναι μηδέν) έχει την ίδια σημασιολογία: το κομμάτι που στέλνει το μήνυμα πρέπει να ρυθμιστεί ώστε να είναι όλα τα bits ίσα με το μηδέν, το κομμάτι που ερμηνεύει το μήνυμα πρέπει να αγνοεί την τιμή. (Με αυτό τον τρόπο, το πεδίο θα μπορούσε να χρησιμοποιηθεί για μελλοντικές επεκτάσεις.)

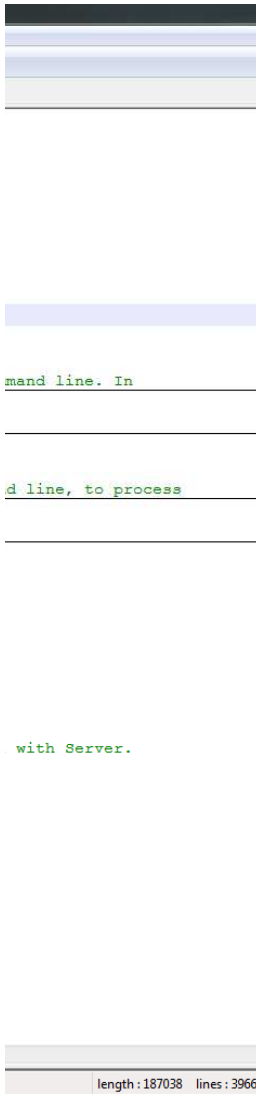
Η τιμή του Server-IV είναι τυχαία και δημιουργείται από το διακομιστή. Σε έλεγχο ταυτότητας λειτουργίας, το server-IV είναι αχρησιμοποίητο.

Το πεδίο Accept σημαίνει προθυμία του server για να συνεχιστεί η επικοινωνία. Μια μηδενική τιμή στο πεδίο σημαίνει ότι ο server δέχεται την πιστοποίηση και είναι πρόθυμος να προβεί σε περαιτέρω συναλλαγές. Μη μηδενικές τιμές δείχνουν ότι ο διακομιστής δεν αποδέχεται την πιστοποίηση ή για κάποιο άλλο λόγο, δεν είναι διατεθειμένος να προβεί σε περαιτέρω συναλλαγές σε αυτή τη σύνοδο ελέγχου OWAMP.

Αν μία αρνητική (μη μηδενική) απάντηση αποστέλλεται, ο διακομιστής μπορεί (και πρέπει ο πελάτης) να κλείσει τη σύνδεση μετά από αυτό το μήνυμα. Η ώρα έναρξης είναι ένα timestamp που αντιπροσωπεύει το χρόνο, όταν η τρέχουσα συγκεκριμενοποίηση του διακομιστή άρχισε να λειτουργεί. (Για παράδειγμα, σε ένα πολλών χρηστών γενικής χρήσεως λειτουργικό σύστημα, θα μπορούσε να είναι ο χρόνος όταν η διεργασία του διακομιστή ξεκίνησε.). Αν το πεδίο Accept είναι μη μηδενικό, θα πρέπει να καθοριστεί ότι όλα τα κομμάτια του είναι μηδενικά. Σε έλεγχο ταυτότητας και κρυπτογράφηση, η ώρα έναρξης κρυπτογραφείται, εκτός και αν το πεδίο Accept είναι μη-μηδενικό.

5 Αναλυτική παρουσίαση εφαρμογής

Σε αυτό το κεφάλαιο θα παρουσιαστεί αναλυτικά το κομμάτι του κώδικα στο οποίο έγινε τροποποίηση της εφαρμογής.



Σε αυτό το κομμάτι του κώδικα όπως φαίνεται από το screen-shot γίνονται οι αρχικοποιήσεις των απαραίτητων μεταβλητών. Αρχικά ορίζουμε ένα stream reader στο οποίο θα εισάγονται τα ορίσματα από το χρήστη κατά την εκτέλεση του προγράμματος. Στη συνέχεια ορίζεται και αρχικοποιείται η μεταβλητή στην οποία θα αποθηκευτεί η διεύθυνση του ntp Server και το χρονικό διάστημα ανανέωσης της ώρας από τον ntp server (updateinterval). Έπειτα ορίζονται οι επόμενες μεταβλητές απαραίτητες για τη λειτουργία του προγράμματος. Ορίζονται και αρχικοποιούνται μεταβλητές για τον έλεγχο του προγράμματος (testError τύπου boolean), για την επίτευξη της επικοινωνίας μεταξύ του client και του Server (serverAddress τύπου inetAddress, ServerTCPportNumber τύπου integer, address και addressAux τύπου string, Ipv6 τύπου boolean) καθώς και οι μεταβλητές που θα αποθηκευτούν τα ορίσματα που θα δώσει ο χρήστης.



Με τη χρήση της συνάρτησης do-while ελέγχουμε από εδώ και πέρα τη ροή του προγράμματος. Αρχικά δίνουμε την τιμή false στην μεταβλητή testError. Στη συνέχεια κομμάτι κομμάτι ξεχωρίζουμε τα ορίσματα που έδωσε ο χρήστης. Ελέγχοντας το κάθε όρισμα καταλαβαίνουμε το επόμενο στοιχείο σε ποια μεταβλητή πρέπει να αποθηκευτεί (αν είναι -s ακολουθεί η ip διεύθυνση του server, αν είναι -c ακολουθεί η ip διεύθυνση του client κτλ). Αν ακολουθήσει αμέσως το επόμενο όρισμα χωρίς κάποια τιμή τότε ενεργοποιείται το testError και ξανά ζητούνται τα ορίσματα από το χρήστη.



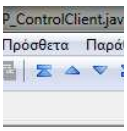
ickets");

ickets");

');

it");

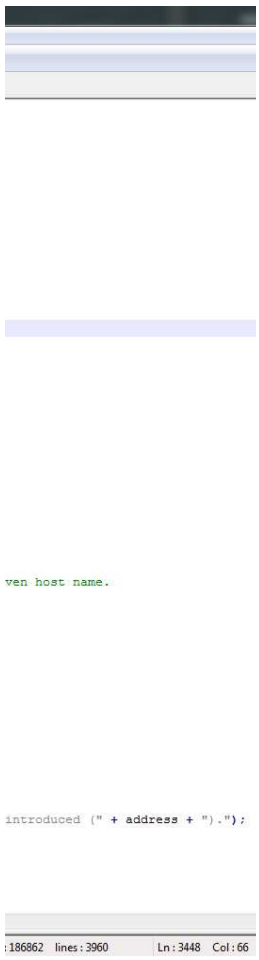
out");



;

p time");

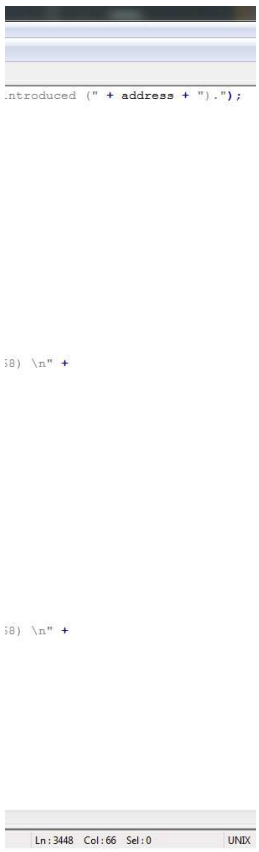
Σε αυτά τα δύο screen shot συνεχίζεται ο διαχωρισμός των ορισμάτων που έδωσε ο χρήστης με τον ίδιο τρόπο που περιγράφτηκε παραπάνω μέχρι το μέγεθος της σειμβολοσειράς να είναι μικρότερο ή ίσο από 4 χαρακτήρες.



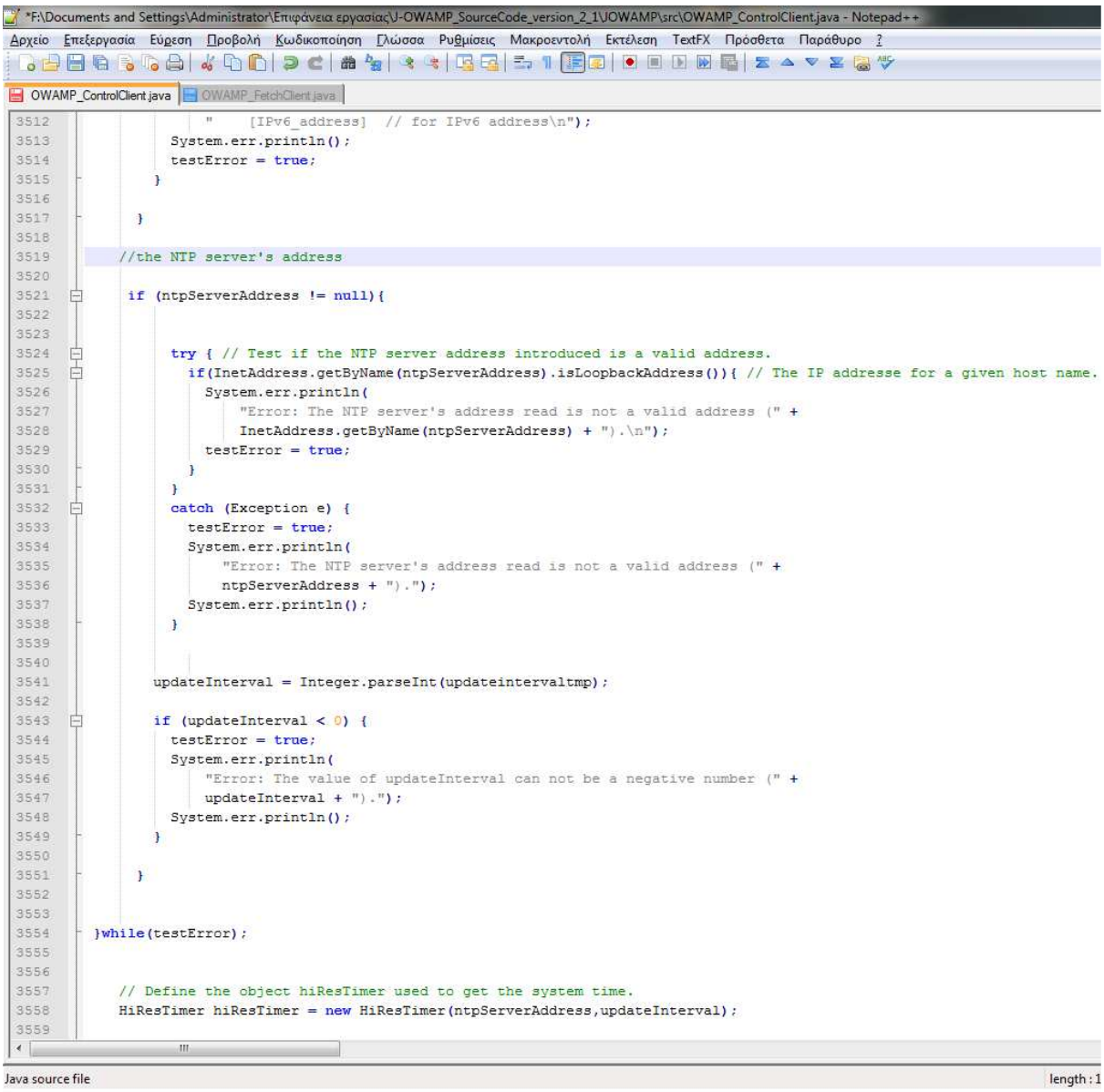
Αφού έχει γίνει η αποθήκευση των ορισμάτων του χρήστη σε μεταβλητές συνεχίζεται η εκτέλεση του προγράμματος. Εάν η τιμή της μεταβλητής testError δεν είναι αληθής (true) τότε αρχικοποιούνται κάποιες ακόμα μεταβλητές που είναι απαραίτητες για την επίτευξη της επικοινωνίας server και client και ελέγχεται η διεύθυνση του server για την έκδοση IP. Εάν η διεύθυνση είναι τύπου Ipv6 τότε ελέγχεται αν ακολουθεί και η θύρα επικοινωνίας αλλιώς ορίζεται η προεπιλεγμένη. Το ίδιο γίνεται και για τη διεύθυνση τύπου Ipv4.

Στη συνέχεια ελέγχεται αν η θύρα επικοινωνίας είναι στα πλαίσια που ορίζει το πρωτόκολλο δηλαδή μεταξύ 0-65536.

Τέλος μετατρέπει τη μεταβλητή address τύπου string σε μορφή InetAddress δηλαδή σε μορφή διεύθυνσης και ελέγχει αν σύμφωνα με το μέγεθος της συμβολοσειράς αν έχει την επιτρεπτή μορφή ip είτε τύπου ipv4 είτε ipv6.



Αν εντοπιστεί κάποιο Exception από την try-catch τότε τυπώνονται τα σχετικά μηνύματα που ενημερώνουν το χρήστη για την εγκυρότητα της τιμής που έδωσε και αλλάζει την boolean τιμή της μεταβλητής testError σε αληθή ώστε να διακοπή η συνέχεια του προγράμματος στον αμέσως επόμενο έλεγχο.

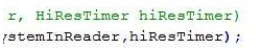


Εικόνα 24: Κώδικας για το Control Client (κομμάτι 7ο)

Το επόμενο κομμάτι που θα ελεγχθεί είναι η διεύθυνση του NTP Server. Αν η τιμή της μεταβλητής δεν είναι κενή όπως ορίστηκε αρχικά (δηλαδή έχει δοθεί στο αρχικό στάδιο του κώδικα από το χρήστη) τότε ελέγχεται αν είναι έγκυρη διεύθυνση διαφορετικά τυπώνεται το ανάλογο μήνυμα σφάλματος και αλλάζει η τιμή του testError σε αληθή.

Αν η διεύθυνση είναι έγκυρη τότε ορίζεται το χρονικό διάστημα ανανέωσης του χρόνου από τον NTP Server. Η αρχική τιμή στην μεταβλητή αυτή είναι 2 (λεπτά) εκτός και αν έχει οριστεί από το χρήστη διαφορετικό χρονικό διάστημα. Αν αυτό το χρονικό διάστημα είναι μικρότερο του 0 τότε αλλάζει η τιμή του testError σε αληθή και τυπώνεται το ανάλογο μήνυμα σφάλματος.

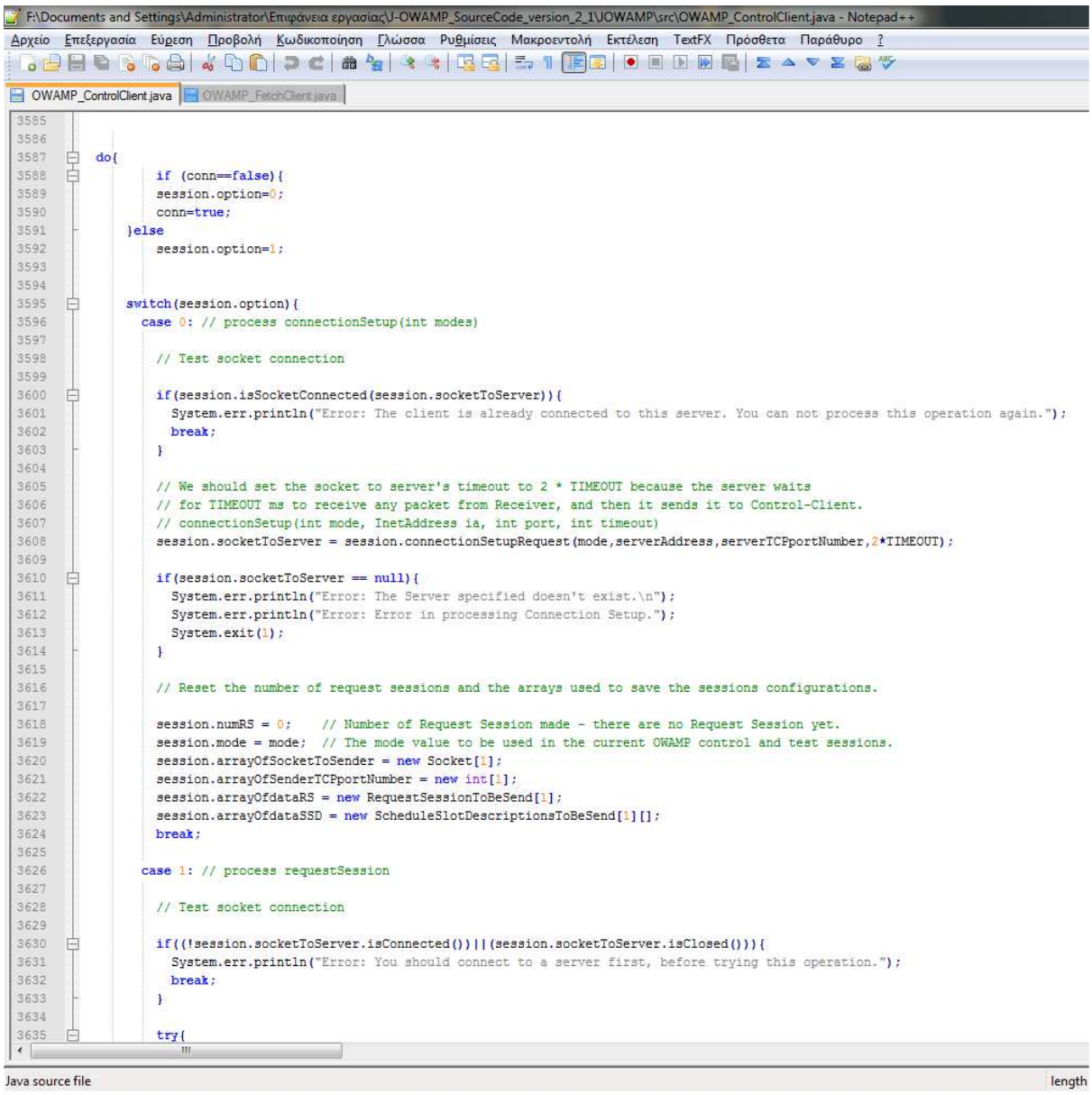






Αφού έχουν γίνει οι απαραίτητοι έλεγχοι μέχρι τώρα ορίζουμε την τιμή του αντικειμένου hiResTimer το οποίο χρησιμοποιείται για να συγχρονιστεί το ρολόι του συστήματος με την ώρα του NTP Server. Αν δεν έχει οριστεί NTP Server τότε ορίζεται η ώρα του συστήματος.

Στη συνέχεια προετοιμάζεται η διασύνδεση της συνεδρίας μεταξύ πελάτη και εξυπηρετητή με βάση τα ορίσματα που έχουν οριστεί (δηλαδή τη διεύθυνση του server, την θύρα επικοινωνίας και το χρόνο). Έπειτα ορίζονται και αρχικοποιούνται οι απαραίτητες μεταβλητές για την συνέχεια του προγράμματος.

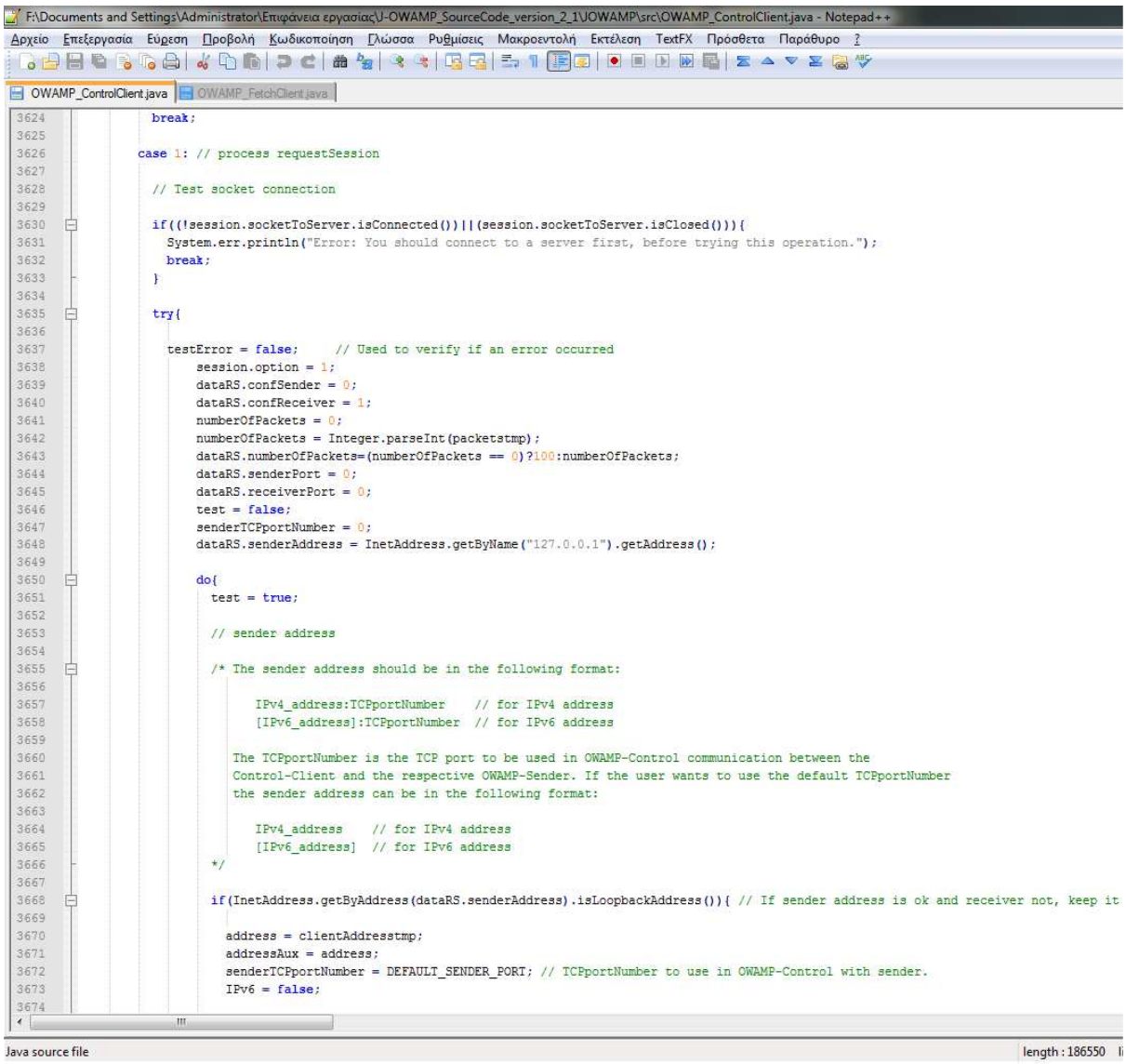


Εικόνα 26: Κώδικας για το Control Client (κομμάτι 9ο)

Αφού αρχικοποιηθούν οι απαραίτητες μεταβλητές ελέγχεται εάν υπάρχει κάποια άλλη ενεργή σύνδεση με τον Server εκείνη τη στιγμή (η ενεργή σύνδεση αφορά αιτήματα από τον ίδιο client τα οποία είναι σε εξέλιξη αφού ο Server μπορεί να απαντήσει σε περισσότερους από έναν client) ώστε να δοθεί η κατάλληλη τιμή στην μεταβλητή session.

Με βάση την τιμή του Session γίνονται οι ακόλουθες ενέργειες. Αν η τιμή είναι 0 τότε γίνονται οι ακόλουθες ενέργειες. Εάν υπάρχει ενεργή σύνδεση τυπώνεται το ανάλογο μήνυμα σφάλματος. Δημιουργείται η σύνδεση με τον εξυπηρετητή. Ελέγχεται αν υπάρχει επικοινωνία με τον server. Εάν δεν υπάρχει τυπώνεται το ανάλογο μήνυμα σφάλματος και διακόπτεται η εκτέλεση του προγράμματος. Τέλος γίνεται επαναφορά των μεταβλητών που αφορούν τη σύνδεση και ενδεχομένως να έχουν μείνει με κάποια τιμή από προηγούμενη σύνδεση.

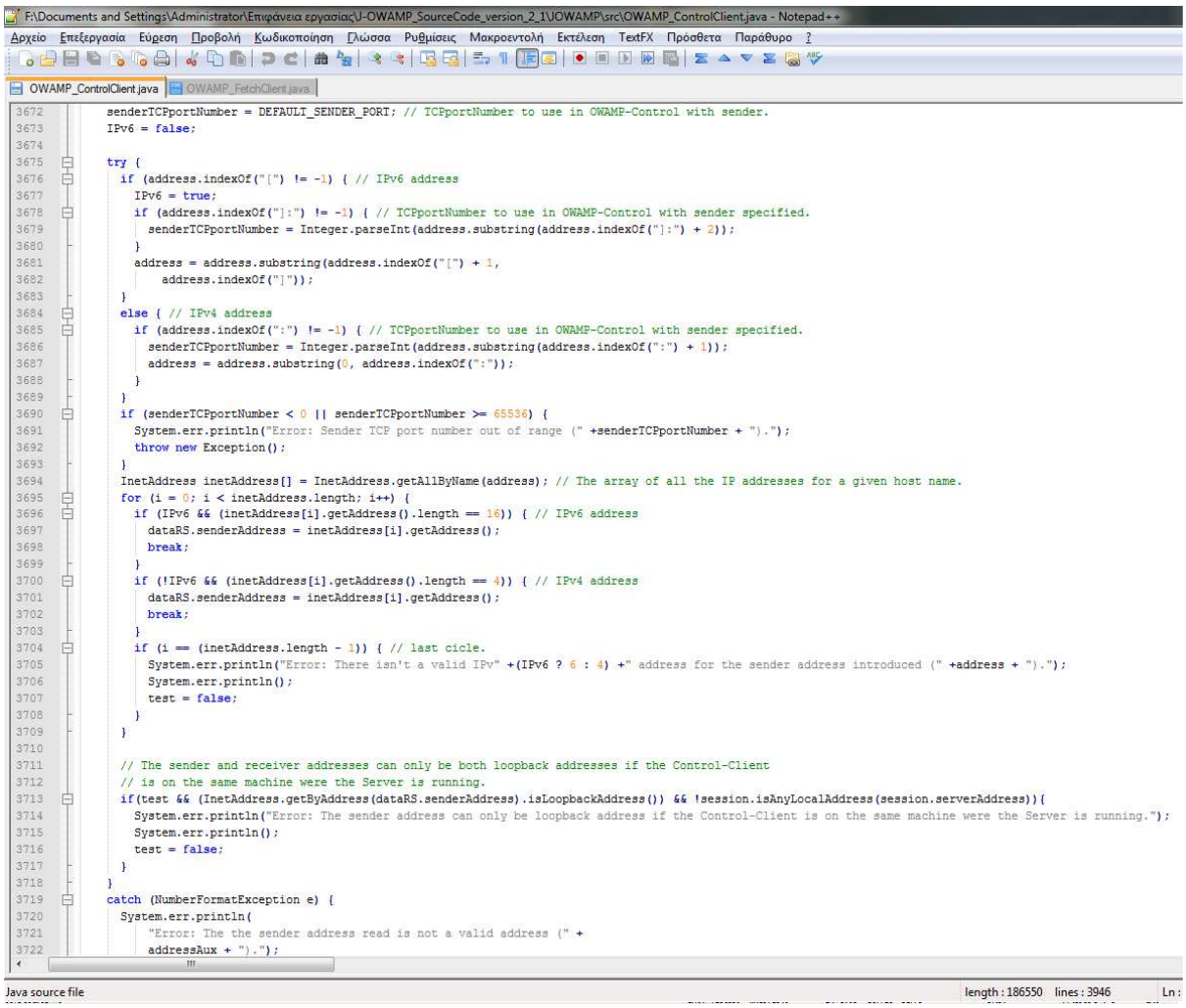
Στην περίπτωση που η τιμή του Session είναι 1 τότε ελέγχεται η σύνδεση με τον Server αν είναι ανοικτή. Αν δεν είναι τυπώνεται το ανάλογο μήνυμα σφάλματος.




```
3624            break;
3625
3626          case 1: // process requestSession
3627
3628            // Test socket connection
3629
3630            if((!session.socketToServer.isConnected())||(session.socketToServer.isClosed())){
3631              System.err.println("Error: You should connect to a server first, before trying this operation.");
3632              break;
3633            }
3634
3635            try{
3636
3637              testError = false;     // Used to verify if an error occurred
3638                  session.option = 1;
3639                  dataRS.confSender = 0;
3640                  dataRS.confReceiver = 1;
3641                  numberOfPackets = 0;
3642                  numberOfPackets = Integer.parseInt(packetstmp);
3643                  dataRS.numberOfPackets=(numberOfPackets == 0)?100:numberOfPackets;
3644                  dataRS.senderPort = 0;
3645                  dataRS.receiverPort = 0;
3646                  test = false;
3647                  senderTCPportNumber = 0;
3648                  dataRS.senderAddress = InetAddress.getByName("127.0.0.1").getAddress();
3649
3650                  do{
3651                    test = true;
3652
3653                    // sender address
3654
3655                    /* The sender address should be in the following format:
3656
3657                          IPv4_address:TCPportNumber    // for IPv4 address
3658                          [IPv6_address]:TCPportNumber  // for IPv6 address
3659
3660                       The TCPportNumber is the TCP port to be used in OWAMP-Control communication between the
3661                       Control-Client and the respective OWAMP-Sender. If the user wants to use the default TCPportNumber
3662                       the sender address can be in the following format:
3663
3664                          IPv4_address    // for IPv4 address
3665                          [IPv6_address]  // for IPv6 address
3666                    */
3667
3668                    if(InetAddress.getByAddress(dataRS.senderAddress).isLoopbackAddress()){ // If sender address is ok and receiver not, keep it
3669
3670                      address = clientAddresstmp;
3671                      addressAux = address;
3672                      senderTCPportNumber = DEFAULT_SENDER_PORT; // TCPportNumber to use in OWAMP-Control with sender.
3673                      IPv6 = false;
3674
```

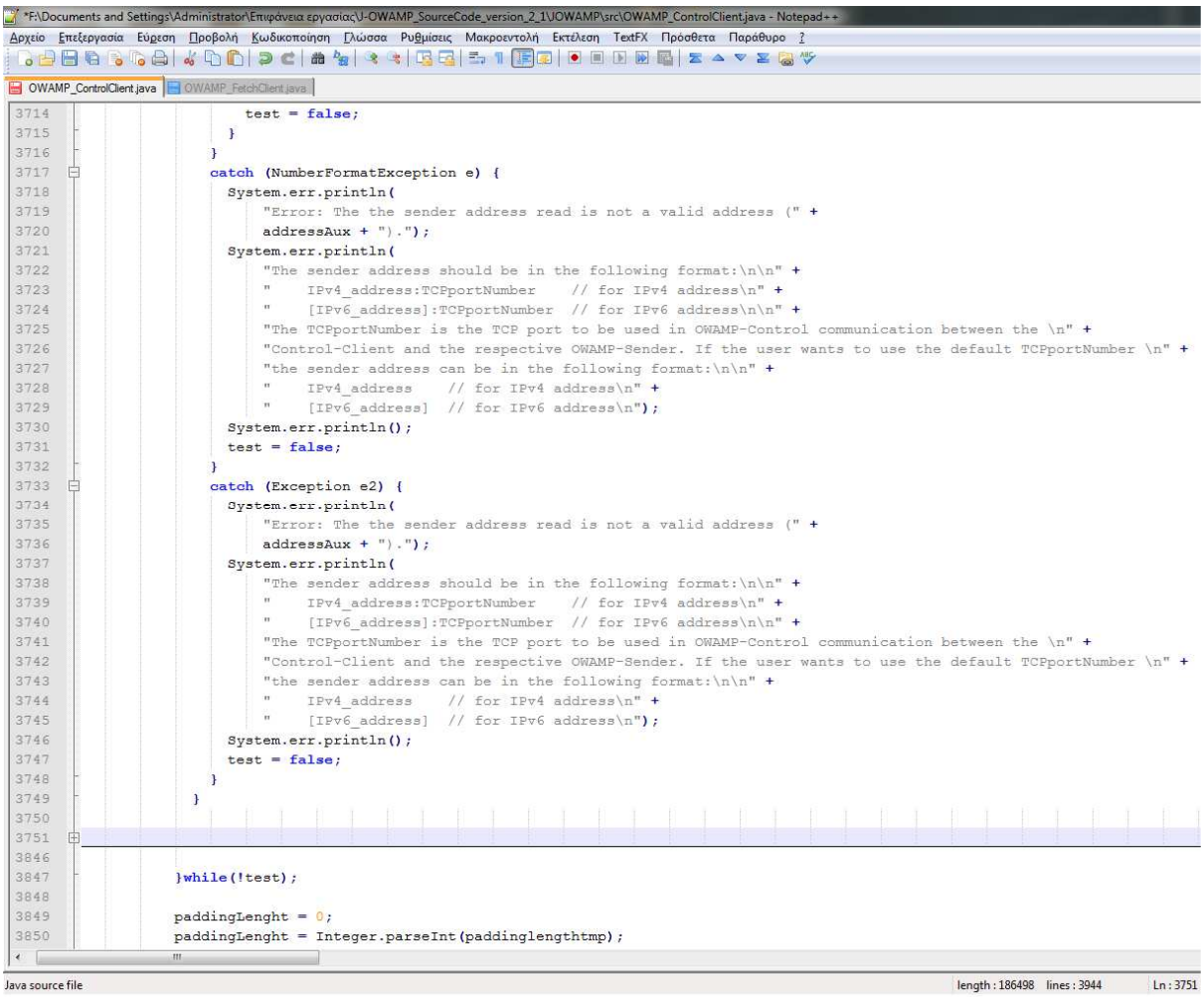
Εικόνα 27: Κώδικας για το Control Client (κομμάτι 10ο)

Συνεχίζοντας στην περίπτωση του Session που είναι 1 και έχει πραγματοποιηθεί η σύνδεση, ελέγχεται αν για κάποιο λόγο δεν έχει πραγματοποιηθεί η σύνδεση, αρχικοποιούνται οι απαραίτητες μεταβλητές και ελέγχεται εάν η διεύθυνση του αποστολέα είναι η loopback διεύθυνση δηλαδή η διεύθυνση που δηλώνει τον ίδιο τον υπολογιστή.



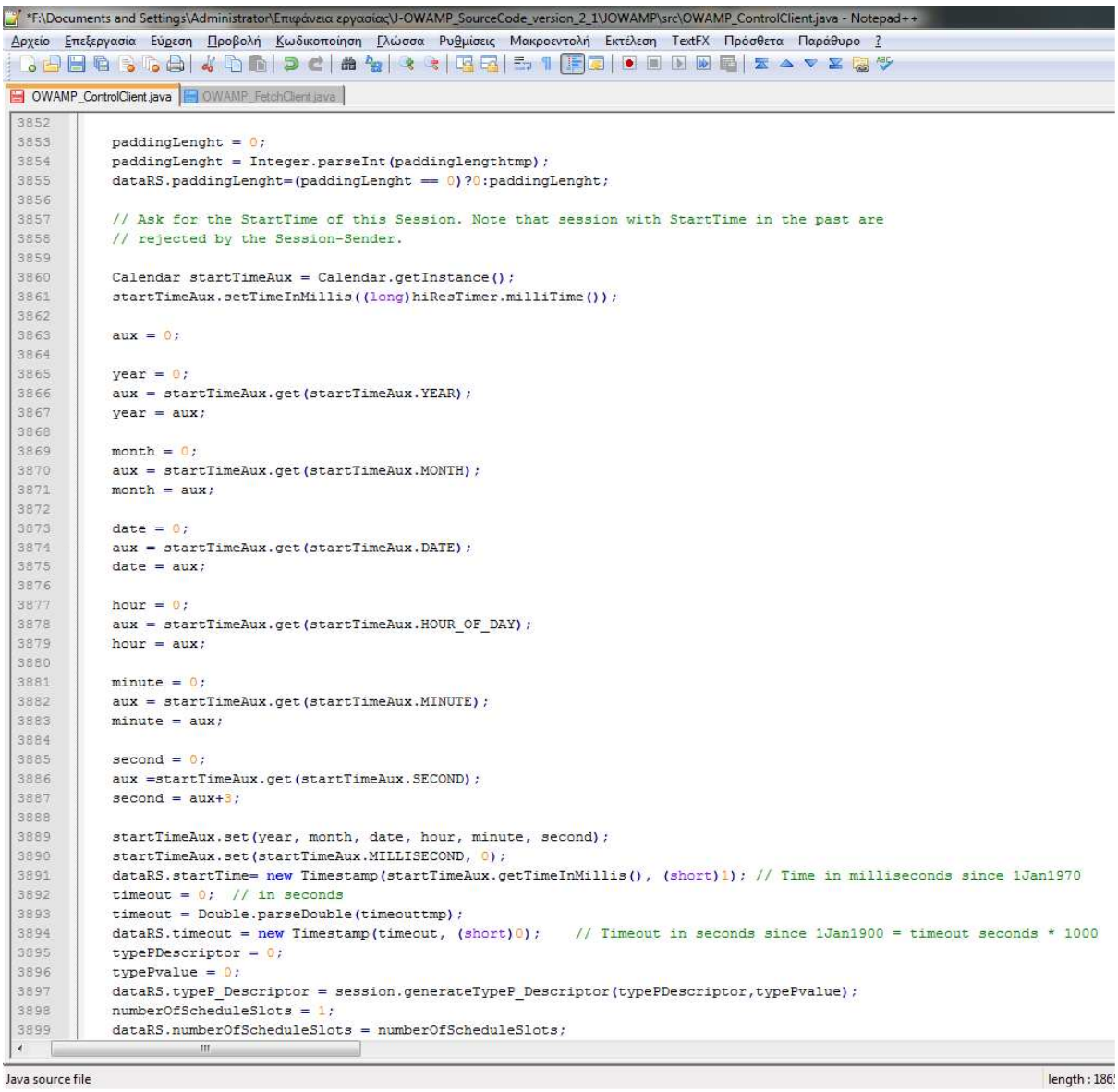
Εικόνα 28: Κώδικας για το Control Client (κομμάτι 11ο)

Εάν η διεύθυνση είναι η loopback τότε ελέγχεται ο τύπος της ip έκδοσης (ipv4 ή ipv6) καθώς και η θύρα επικοινωνίας και αποθηκεύεται η διεύθυνση στην κατάλληλη μεταβλητή. Τέλος ελέγχεται εάν η διεύθυνση του παραλήπτη είναι και αυτή η loopback. Αυτό γίνεται γιατί επιτρέπεται να είναι και οι δύο διευθύνσεις loopback μόνο εάν ο control-client και ο server είναι στο ίδιο μηχάνημα.



Εικόνα 29: Κώδικας για το Control Client (κομμάτι 11ο)

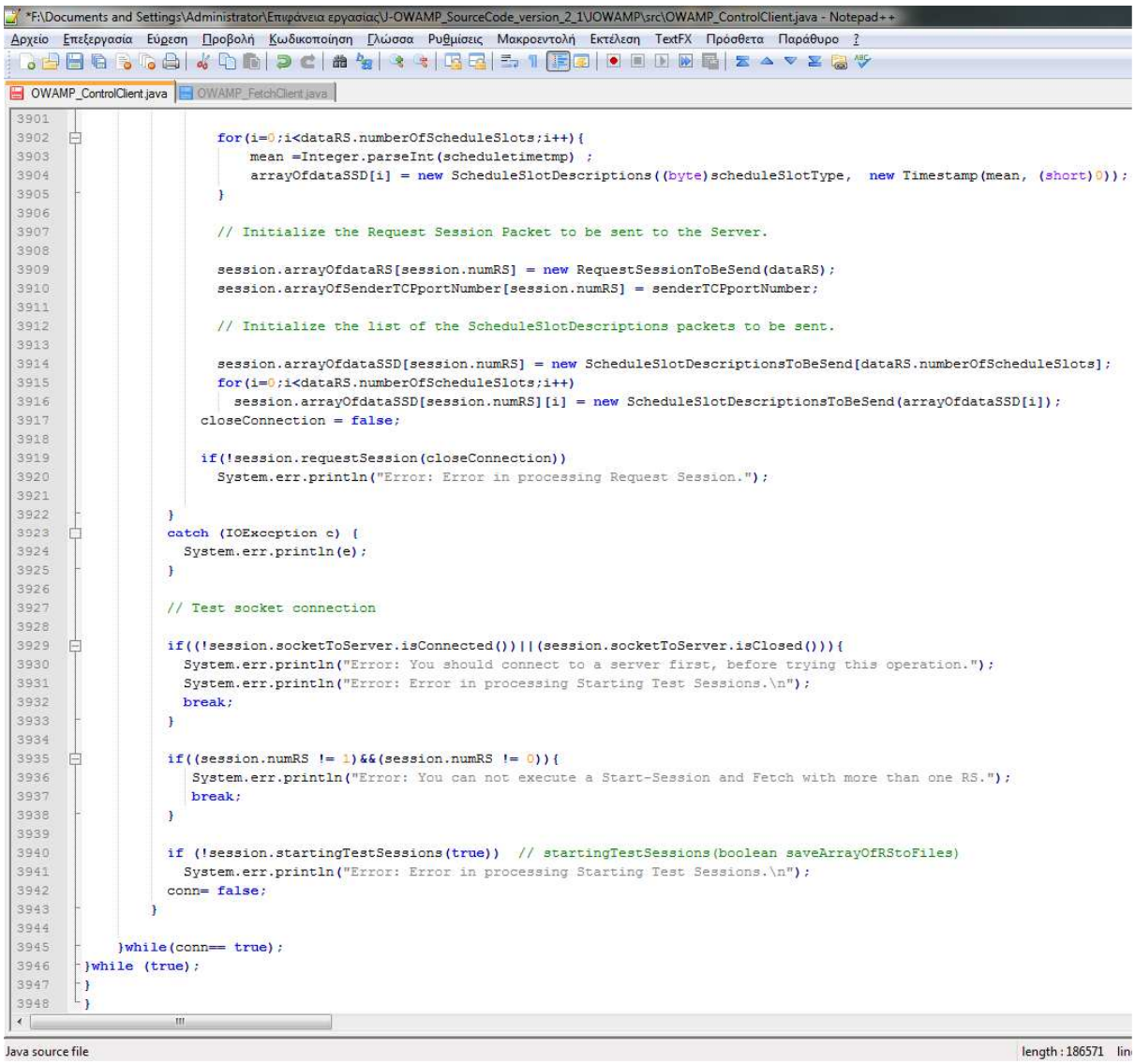
Εάν η μορφή της διεύθυνσης δεν είναι σωστή τότε τυπώνεται το ανάλογο μήνυμα σφάλματος. Τέλος βλέπουμε ότι αυτή η διαδικασία επαναλαμβάνεται όσο δεν υπάρχει κάποιο σφάλμα στους ελέγχους των δεδομένων (η μεταβλητή test δεν είναι false).



Εικόνα 30: Κώδικας για το Control Client (κομμάτι 12ο)

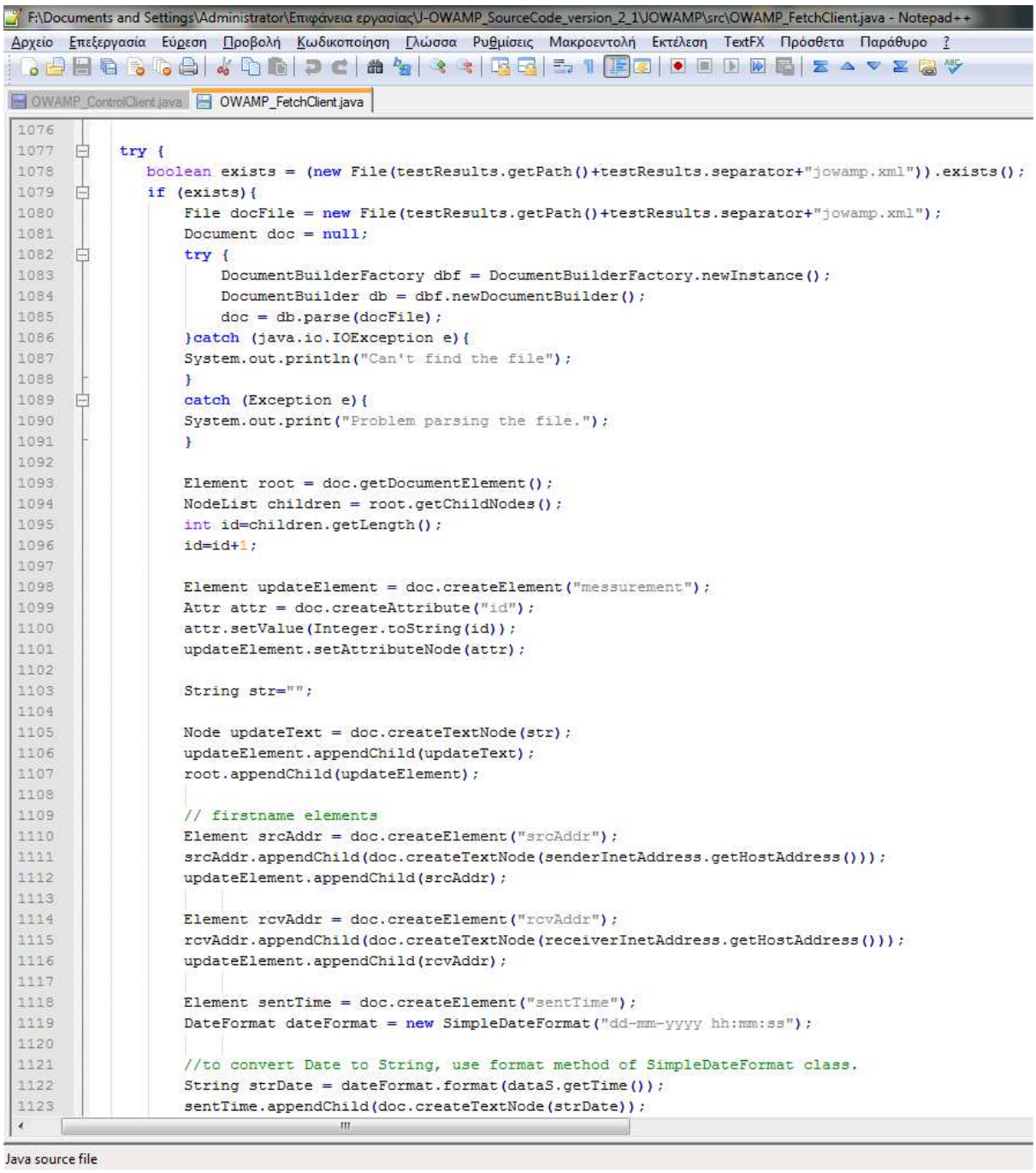
Σε αυτό το screen shot ορίζονται οι τελευταίες μεταβλητές πριν την τελική φάση του προγράμματος που είναι η έναρξη των δοκιμών. Αρχικά ορίζεται (σε bytes) το επιπλέον μέγεθος που μπορεί να εισαχθεί στα δοκιμαστικά πακέτα. Στη συνέχεια ορίζεται ο χρόνος έναρξης της δοκιμής. Παρατηρώντας τον κώδικα βλέπουμε ότι όλες οι δοκιμές θα ξεκινάνε 3 δευτερόλεπτα μετά την εκκίνηση του προγράμματος.

Τέλος ορίζεται ο αριθμός των schedule slots. Αυτή η παράμετρος καθορίζει μια ομάδα πακέτων που θα αποσταλεί με το αίτημα. Κάθε πακέτο αντιπροσωπεύει ένα slot. Έτσι έχουμε ένα schedule με ένα συγκεκριμένο αριθμό από slot. Κάθε slot έχει έναν τύπο και μια παράμετρο. Εμείς στο πρόγραμμά μας ακολουθούμε μια καθορισμένη ακολουθία. Η παράμετρος αυτή εκφράζεται ως timestamp και καθορίζει ένα χρονικό διάστημα το οποίο είναι η καθυστέρηση η ίδια. Ο αποστολέας ξεκινά την έναρξη του schedule και εκτελεί τις οδηγίες του slot. Περιμένει για το συγκεκριμένο χρονικό διάστημα και στέλνει ένα πακέτο δοκιμών. Το schedule είναι κυκλικό, όταν δεν υπάρχουν άλλα slot ο αποστολέας επιστρέφει στο πρώτο slot.



Εικόνα 31: Κώδικας για το Control Client (κομμάτι 13ο)

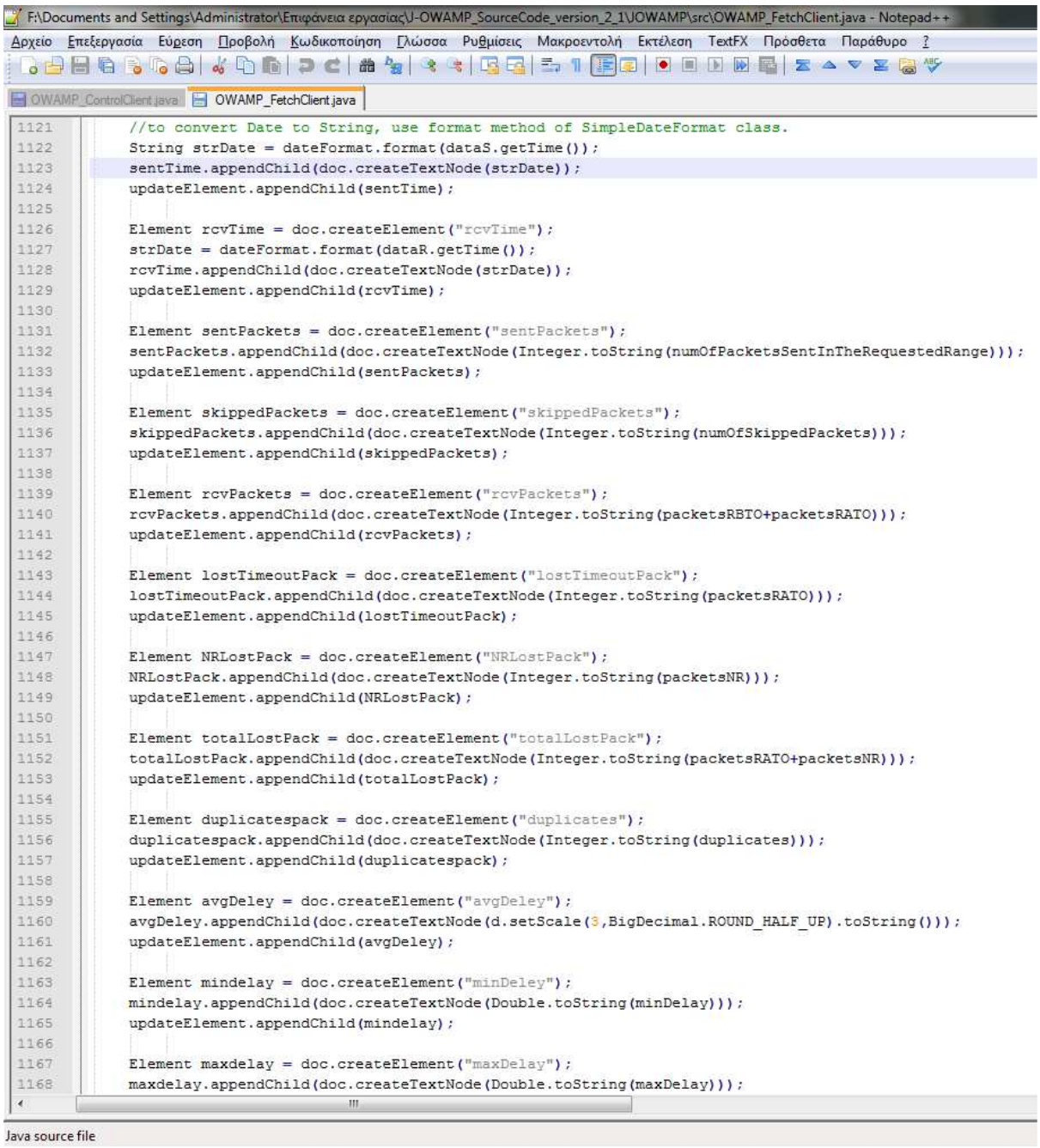
Στο τελευταίο κομμάτι του κώδικα προετοιμάζεται η σύνοδος πακέτων αίτησης που θα αποσταλεί στον server. Στη συνέχεια αρχικοποιείται η λίστα των schedule slot που θα σταλεί. Και τέλος ξεκινάει η αποστολή των δοκιμαστικών πακέτων.




```
1076
1077    try {
1078       boolean exists = (new File(testResults.getPath()+testResults.separator+"jowamp.xml")).exists();
1079       if (exists){
1080           File docFile = new File(testResults.getPath()+testResults.separator+"jowamp.xml");
1081           Document doc = null;
1082           try {
1083               DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();
1084               DocumentBuilder db = dbf.newDocumentBuilder();
1085               doc = db.parse(docFile);
1086           }catch (java.io.IOException e){
1087           System.out.println("Can't find the file");
1088           }
1089           catch (Exception e){
1090           System.out.print("Problem parsing the file.");
1091           }
1092
1093           Element root = doc.getDocumentElement();
1094           NodeList children = root.getChildNodes();
1095           int id=children.getLength();
1096           id=id+1;
1097
1098           Element updateElement = doc.createElement("messurement");
1099           Attr attr = doc.createAttribute("id");
1100           attr.setValue(Integer.toString(id));
1101           updateElement.setAttributeNode(attr);
1102
1103           String str="";
1104
1105           Node updateText = doc.createTextNode(str);
1106           updateElement.appendChild(updateText);
1107           root.appendChild(updateElement);
1108
1109           // firstname elements
1110           Element srcAddr = doc.createElement("srcAddr");
1111           srcAddr.appendChild(doc.createTextNode(senderInetAddress.getHostAddress()));
1112           updateElement.appendChild(srcAddr);
1113
1114           Element rcvAddr = doc.createElement("rcvAddr");
1115           rcvAddr.appendChild(doc.createTextNode(receiverInetAddress.getHostAddress()));
1116           updateElement.appendChild(rcvAddr);
1117
1118           Element sentTime = doc.createElement("sentTime");
1119           DateFormat dateFormat = new SimpleDateFormat("dd-mm-yyyy hh:mm:ss");
1120
1121           //to convert Date to String, use format method of SimpleDateFormat class.
1122           String strDate = dateFormat.format(dataS.getTime());
1123           sentTime.appendChild(doc.createTextNode(strDate));
```

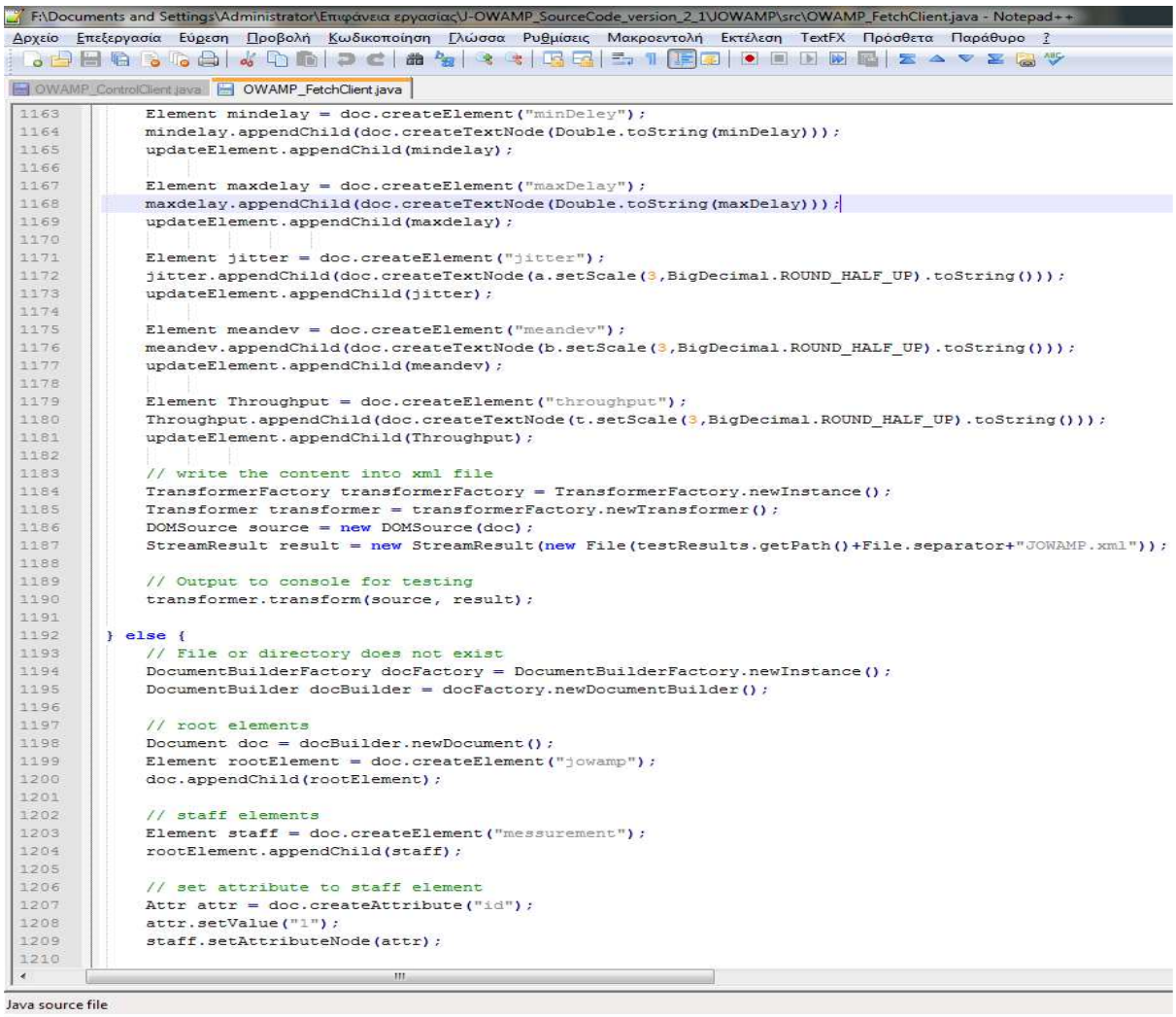
Εικόνα 32: Κώδικας για το Fetch Client (κομμάτι 1ο)

Σε αυτό τον κώδικα γίνεται η αποθήκευση των αποτελεσμάτων των μετρήσεων σε ένα αρχείο xml. Αρχικά ελέγχεται εάν το αρχείο υπάρχει. Εάν υπάρχει τότε δημιουργούνται οι απαραίτητες μεταβλητές ώστε να προστεθούν στο ήδη υπάρχον αρχείο τα τρέχον αποτελέσματα. Αρχικά μετριέται ο αριθμός των μέχρι εκείνη τη στιγμή roots. Δημιουργείται το επόμενο root μαζί με μια τιμή id που είναι ο αριθμός των υπάρχον συν ένα. Στη συνέχεια δημιουργούνται τα στοιχεία που είναι απαραίτητα για τα αποτελέσματα. Σε αυτό το screen shot δημιουργούνται τα στοιχεία για την διεύθυνση του αποστολέα και του παραλήπτη καθώς και ο χρόνος εκκίνησης της δοκιμής.



Εικόνα 33: Κώδικας για το Fetch Client (κομμάτι 2ο)

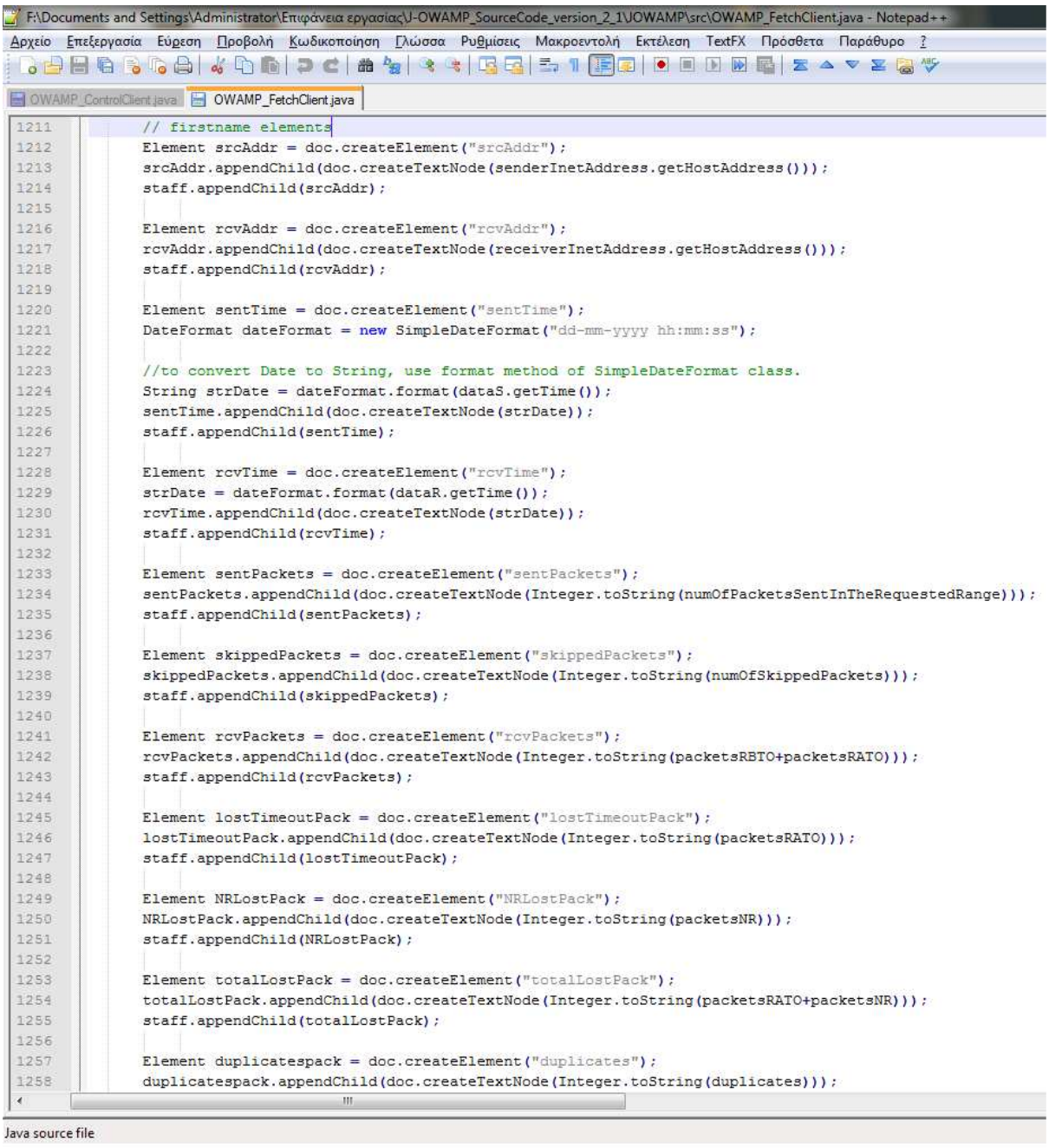
Στη συνέχεια δημιουργούνται τα στοιχεία για το χρόνο λήψης των απαντητικών πακέτων (οι χρόνοι και στις δύο περιπτώσεις αφορούν τον client και όχι τον server), τον αριθμό των πακέτων που στάλθηκαν συνολικά, τον αριθμό των πακέτων που απορρίφθηκαν, τον αριθμό των πακέτων που ελήφθησαν, τον αριθμό των πακέτων που απορρίφθηκαν λόγω χρόνου, το συνολικό αριθμό των πακέτων που χάθηκαν, τον αριθμό των διπλότυπων πακέτων, τη μέση, ελάχιστη και μέγιστη καθυστέρηση.



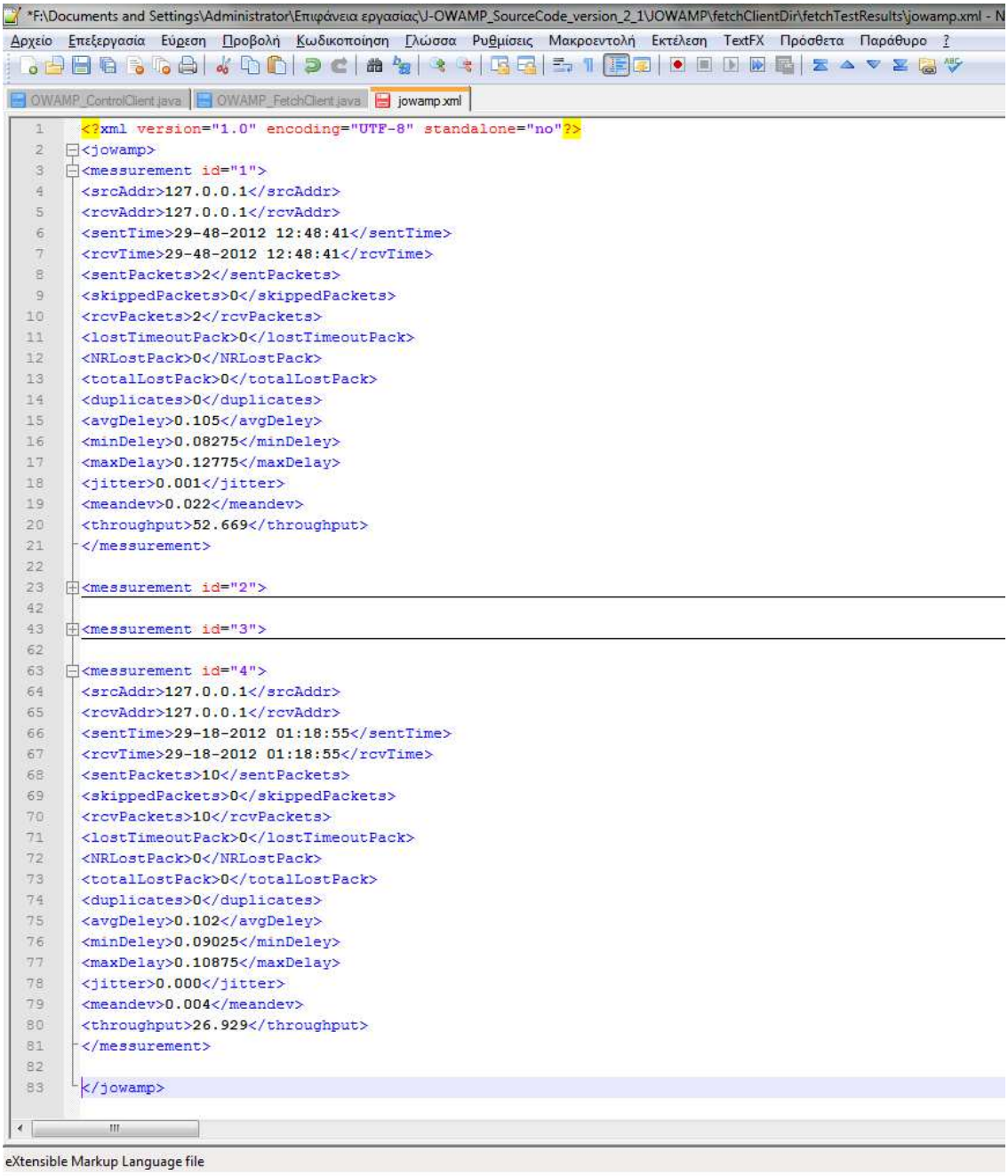
Εικόνα 34: Κώδικας για το Fetch Client (κομμάτι 3ο)

Σε αυτό το screen shot συνεχίζεται η δημιουργία των στοιχείων που αποθηκεύονται τα αποτελέσματα. Τα τελευταία τρία στοιχειά είναι το jitter, η διακύμανση και το throughput. Τέλος όλα τα στοιχεία γράφονται στο αρχείο.

Διαφορετικά (δηλαδή αν δεν υπάρχει το αρχείο) δημιουργούνται οι μεταβλητές ώστε να αποθηκευτεί το αρχείο από την αρχή. Όπως θα παρατηρήσουμε και στα επόμενα δύο screen shot τα ορίσματα είναι ακριβώς τα ίδια με την περίπτωση που υπάρχει το αρχείο.



Εικόνα 35: Κώδικας για το Fetch Client (κομμάτι 4ο)



Εικόνα 36: xml αρχείο με αποτελέσματα

Σε αυτό το screen shot βλέπουμε τη μορφή που έχει το αρχείο xml με κάποια δοκιμαστικά αποτελέσματα.

Βιβλιογραφία

[1] Δίκτυα υπολογιστών – Adrew S. Tanenbaum τέταρτη αμερικάνικη έκδοση
[2] http://www.av.it.pt/jowamp/
[3] http://www.wikipedia.org/
[4] http://www.perfsonar.net/
[5] http://www.internet2.edu/performance/bwctl/
[6] http://openmaniak.com/iperf.php
[7] http://www.nuttcp.net/nuttcp/Welcome%20Page.html
[8] http://www.rfc-editor.org/rfc/rfc4656.txt.
[9] http://docs.oracle.com/javase/1.5.0/docs/api/

Παράρτημα

1 Εισαγωγή και απαιτήσεις

Το J-OWAMP σύστημα μπορεί να χρησιμοποιηθεί τόσο σε Windows όσο και Linux πλατφόρμες, απαιτώντας μόνο την εγκατάσταση του J2SE Java Runtime Environment (JRE), διαθέσιμο και για τα δύο Windows και Linux. Η εγκατάσταση της παρούσας ενότητας είναι υποχρεωτική, προκειμένου να επιτραπεί η εκτέλεση εφαρμογών Java. Το J2SE Java Runtime Environment (JRE) 5,0 απαιτείται για την IPv6 υποστήριξη για τα Microsoft Windows. Με την J2SDK/JRE 1,4 έκδοση, υπάρχει υποστήριξη IPv6 σε Linux.

Το OWAMP απαιτεί ένα συγχρονισμένο ρολόι, προκειμένου να παρέχει ουσιαστικές μετρήσεις. Όμως, το πιο σημαντικό, το ρολόι πρέπει να είναι σταθερό. Εάν το ρολόι συστήματος είναι κλιμακωτό κατά τη διάρκεια μιας συνεδρίας OWAMP, τα αποτελέσματα μπορεί να είναι παραπλανητικά. Το OWAMP απαιτεί τη χρήση του GPS ή NTP για να συγχρονίσει το ρολόι του συστήματος.

Ο συγχρονισμός του χρόνου, η ακρίβεια και η ανάλυση είναι μερικές από τις πιο σημαντικές απαιτήσεις του OWAMP. Καλύτερη ακρίβεια και ανάλυση μπορεί να ληφθεί με χρήση μετρητών επιδόσεων σε συνδυασμό με το χρόνο του συστήματος, προκειμένου να υπολογιστεί ακριβές και μικρότερο χρόνο προσαυξήσης. Για να γίνει αυτό, μια αναφορά πρέπει να καθοριστεί. Σε κάθε χρονική στιγμή η τρέχουσα ώρα μπορεί να προσδιοριστεί ως εξής: currentTime = + timeReference (currentCounterValue - referenceCounterValue) / counterFrequency. Η χρονική αναφορά μπορεί να ορισθεί με τη χρήση του χρόνου του συστήματος, στην οποία περίπτωση το σύστημα θα πρέπει να είναι συγχρονισμένο, ή απευθείας από ένα διακομιστή NTP. Το J-OWAMP επιτρέπει αυτές τις δύο προσεγγίσεις.

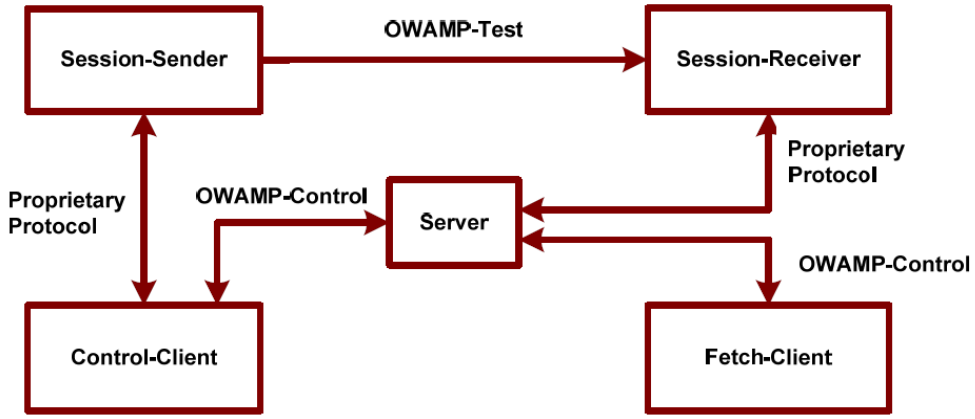
Για τη δεύτερη προσέγγιση, το J-OWAMP περιλαμβάνει έναν πελάτη NTP που μπορεί να χρησιμοποιηθεί για να πάρει το χρόνο αναφοράς απευθείας από ένα διακομιστή NTP, χωρίς την απαίτηση ενός συγχρονισμένου ρολογιού στο μηχάνημα που φιλοξενεί το J-OWAMP. Αυτή η προσέγγιση μπορεί να οδηγήσει σε καλύτερα αποτελέσματα από την πρώτη. Και στις δύο προσεγγίσεις, η αναφορά του χρόνου θα πρέπει να συγχρονίζεται συχνά. Ως εκ τούτου, ένα χρονικό διάστημα ενημέρωσης θα πρέπει να οριστεί.

Με το J-OWAMP είναι δυνατόν να επιτευχθούν δύο είδη συνεδριών δοκιμής: ενιαία δοκιμή συνεδρίας και κατά διαστήματα δοκιμή συνεδρίας. Η ενιαία περίοδο δοκιμών αντιστοιχεί στη διαμόρφωση μιας ενιαίας δοκιμής μεταξύ δύο μηχανών. Ο χρήστης μπορεί να καθορίσει τον αριθμό των πακέτων, τα interarrivals πακέτα, το μήκος του πακέτου και η ώρα έναρξης της συνεδρίασης. Τα αποτελέσματα που λαμβάνονται με αυτό το είδος της συνόδου είναι οι μετρήσεις απόδοσης εκτίμησης όπως καθυστέρηση, απώλειες, επικαλύψεις και απόδοση. Το διάστημα εμπιστοσύνης μιας περιόδου δοκιμών επιτρέπει το διαχωρισμό μιας πλήρους περιόδου σε έναν αριθμό μικρότερων διαστημάτων δοκιμών, και την εκτέλεση ενός συνόλου δοκιμών σε κάθε διάστημα για να επιτρέψει την κατασκευή του διαστήματος εμπιστοσύνης. Για παράδειγμα, αν η περίοδος κύκλου ρυθμίζεται σε 24 ώρες, ο χρήστης μπορεί να ορίσει 24 διαστήματα (διάρκειας μιας ώρας), και να ορίζουν επίσης ότι σε κάθε διάστημα συνολικά 10 δοκιμές θα πρέπει να πραγματοποιηθούν με σκοπό να υπολογιστεί το διάστημα εμπιστοσύνης. Στην περίπτωση αυτή, το χρονικό διάστημα μεταξύ της έναρξης των διαδοχικών συνεδριών μπορεί επίσης να οριστεί (δηλαδή 2 λεπτά).

Σε αυτό το εγχειρίδιο, θα αναφερθούμε στην έκδοση των Windows του J-OWAMP για να εξηγήσουν πώς να χρησιμοποιήσετε αυτήν την εφαρμογή για την εκτέλεση ενεργών μετρήσεις. Το εγχειρίδιο είναι δομημένο στην ακόλουθο τρόπο: ενότητα 2 παρουσιάζει την αρχιτεκτονική του συστήματος, ενότητα 3 δείχνει πώς να τρέξει J- OWAMP, ενότητα 4 περιγράφει λεπτομερώς όλα τα αναγκαία μέτρα για να ρυθμίσετε εύκολα οι διάφορες ενότητες του συστήματος και ενότητα 5 περιγράφει αναλυτικά όλα τα αναγκαία μέτρα για να εύκολα να ρυθμίσετε τη μονάδα CalculateCI.

2 Αρχιτεκτονική συστήματος

Το J-OWAMP σύστημα εφαρμόζει την OWAMP αρχιτεκτονική που φαίνεται στο Σχήμα 1:



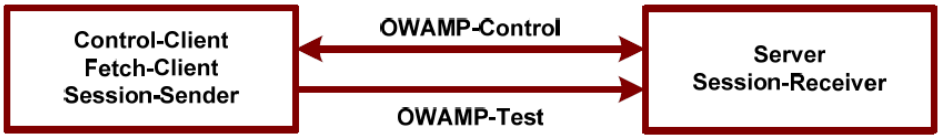
Εικόνα 37: OWAMP αρχιτεκτονική

Η αρχιτεκτονική OWAMP βασίζεται σε δύο αλληλεξαρτώμενα πρωτόκολλα, η-OWAMP Control και η OWAMP-Test, το οποίο μπορεί να εγγυηθεί την πλήρη απομόνωση μεταξύ του πελάτη και διακομιστή. Το OWAMP-Control χρησιμοποιείται για να αρχίζει και να τελειώνει συνεδρίες δοκιμών καθώς και να λαμβάνει τα αποτελέσματα των δοκιμών αυτών, ενώ το OWAMP-Test πρωτόκολλο χρησιμοποιείται για να επιτρέπει την ανταλλαγή των πακέτων ελέγχου μεταξύ δύο σημείων που ανήκουν στην παρακολούθηση δικτύου.

Η προτεινόμενη αρχιτεκτονική περιλαμβάνει τα ακόλουθα στοιχεία (Σχήμα 1):

- Session-Sender: ο αποστολέας των πακέτων δοκιμής. Το έργο υλοποιείται από το JOWAMP_SessionSender.exe.
- Session-Receiver: ο δέκτης των πακέτων δοκιμής. Το έργο υλοποιείται από το JOWAMP_SessionReceiver.exe.
- Server: η οντότητα που είναι υπεύθυνη για τη συνολική διαχείριση του συστήματος. Υλοποιείται από τις JOWAMP_Server.exe
- Control-Client: ένα τερματικό σύστημα που προγραμματίζει τα αιτήματα για δοκιμές, πυροδοτεί την έναρξη ενός συνόλου συνεδρίας και μπορεί επίσης να τελειώσει μία ή όλες τις τρέχουσες συνεδρίες. Το έργο υλοποιείται από την εφαρμογή JOWAMP_ControlClient.exe.
- Fetch-Client: ένα τερματικό σύστημα που ενεργοποιεί τα αιτήματα για τα αποτελέσματα των δοκιμών για συνεδρίες που έχουν ήδη ολοκληρωθεί ή βρίσκονται σε εξέλιξη. Το έργο υλοποιείται από την εφαρμογή JOWAMP_FetchClient.exe.

Ένα στοιχείο του δικτύου μπορεί να διεξάγει διάφορες λογικές λειτουργίες ταυτόχρονα (απλοποιημένο σενάριο). Για παράδειγμα, μπορούμε να έχουμε μόνο δύο στοιχεία του δικτύου: το ένα είναι η διενέργεια λειτουργίας που αντιστοιχεί σε Control-Client, ένα Fetch-Client και μια Session-Sender και το άλλο είναι η εκτέλεση των καθηκόντων που αντιστοιχούν σε ένα Server και μια Session-Receiver (Σχήμα 2).



Εικόνα 38: OWAMP απλοποιημένη αρχιτεκτονική

Η αρχιτεκτονική του σχήματος 1 επιτρέπει τον ορισμό ενός μόνο πελάτη και ένα διακομιστή στο δίκτυο (ενδεχομένως να εγκατασταθεί σε μηχανήματα με τη μεγαλύτερη ικανότητα επεξεργασίας) και επιτρέπει την εγκατάσταση των αποστολέων και των δεκτών σε κάθε μηχάνημα του δικτύου, πράγμα που οδηγεί σε μείωση επεξεργαστικού φόρτου. Με τον τρόπο αυτό, ο διαχειριστής του δικτύου μπορεί να πραγματοποιήσει δοκιμές σε όλο το δικτύο από μία μόνο μηχανή, το οποίο όμως δεν είναι δυνατόν στο απλοποιημένο σενάριο.

Σημείωση: Όλες οι μηχανές που είναι υπεύθυνες για την εκτέλεση των διαφόρων εφαρμογών που συνθέτουν αυτό το σύστημα μέτρησης πρέπει να συγχρονιστούν με ακρίβεια, ώστε να διασφαλιστεί καλά αποτελέσματα των μετρήσεων.

3 Εκτέλεση J-OWAMP

Για να εκτελεστεί το J-OWAMP ο χρήστης πρέπει να εκτελέσει κάθε ένα από τα στοιχεία (εφαρμογές) που συνθέτουν την αρχιτεκτονική του: μπορεί να κάνει διπλό κλικ με το κουμπί του ποντικιού στην αντίστοιχο εικονίδιο ή να χρησιμοποιήσετε τη γραμμή εντολών (DOS παράθυρο στο λειτουργικό σύστημα Windows). Με διπλό κλικ οι πόρτες που θα χρησιμοποιηθούν στις συνδέσεις TCP μεταξύ των διαφορετικων στοιχείων του J-OWAMP είναι οι εξής:

J-OWAMP element	Port number
JOWAMP-Server	22368
JOWAMP Session-Receiver	28181
JOWAMP Session-Sender	4181

Πίνακας 4: Αριθμοί θυρών που χρησιμοποιούνται σε κάθε στοιχείο για TCP συνδέσεις

Για να χρησιμοποιήσετε άλλους αριθμούς θυρών, κάθε μία από αυτές τις ενότητες θα πρέπει να εκτελείται στη γραμμή εντολών με την ακόλουθη μορφή:

3.1 OWAMP-Server:

Usage: OWAMP_Server [-options]

όπου τα options περιπλαμβάνουν:

-a | -address : Η τοπική διεύθυνση IP για να συνδεθεί. Αυτή η διεύθυνση μπορεί να χρησιμοποιηθεί σε ένα multi- homed εξυπηρετητή για τον Server που θα δέχονται μόνο αιτήματα σύνδεσης . Εάν δεν ορίζεται, από προεπιλογή θα δεχτεί συνδέσεις για κάθε/όλες τις τοπικές διευθύνσεις.

-p | -port port : Η TCP πόρτα που θα χρησιμοποιηθεί στο OWAMP-Control.

-r | -receiverTCPport port : Η TCP πόρτα που θα χρησιμοποιηθεί στο OWAMP- Control με όλα τα OWAMP Session-Receivers.

-n | -ntpServer : Η διεύθυνση του διακομιστή NTP χρησιμοποιείται για να πάρει τη χρονική αναφορά. Εάν δεν ορίζεται, ο χρόνος του σύστημα θα χρησιμοποιηθεί για να οριστεί η αναφορά του χρόνου.

-u | -updateInterval : Η τιμή (σε λεπτά) που θα χρησιμοποιηθεί για το χρονικό διάστημα μεταξύ δύο διαδοχικών ενημερώσεις από την αναφορά του χρόνου.

-? -help :Τυπώνει αυτό το μήνυμα βοήθειας.

Η receiverTCPport είναι ο αριθμός θύρας TCP που θα χρησιμοποιηθεί στο OWAMP-Control μεταξύ του Server και σε όλα τα OWAMP Session-Receivers. Είναι η θύρα TCP, όπου η OWAMP Session-Receivers θα σας περιμένει για αιτήματα σύνδεσης. Εάν δεν ορίζεται, η προεπιλεγμένη θύρα.

Σημείωση: Η παραμετροποίηση των κάθε συνεδρίων δοκιμής αποθηκεύονται σε ένα αρχείο του οποίου το όνομα αποτελείται από συνεδρίαση SID σειρά με την επέκταση ". rs». Αυτό το αρχείο αποθηκεύεται στον κατάλογο serverRSdir που βρίσκεται στον κατάλογο από όπου το OWAMP-Server εκτελείται. Ο serverRSdir καταλόγος γίνεται κάθε φορά που η OWAMP-Server έχει ξεκινήσει. Αν υπάρχει, θα διαγράφεται και δημιούργησε και πάλι.

3.2 OWAMP Session-Sender:

Usage: OWAMP_SessionSender [-options]

όπου τα options περιπλαμβάνουν:

-a | -address : Η τοπική διεύθυνση IP για να συνδεθεί. Αυτή η διεύθυνση μπορεί να χρησιμοποιηθεί σε ένα multi- homed εξυπηρετητή για ένα Session-Sender που θα δέχεται μόνο αιτήσεις σύνδεσης σε μία από τις διευθύνσεις του. Αν δεν ορίζεται, θα δεχθεί από προεπιλογή συνδέσεις σε οποιεσδήποτε/όλες τις τοπικές διευθύνσεις.

-p | -port port : Η TCP πόρτα που θα χρησιμοποιηθεί στο OWAMP-Control.

-r | -portRange [lowport,highport] : Η UDP πόρτα που θα χρησιμοποιηθεί στο OWAMP-Test sessions.

-n | -ntpServer : Η διεύθυνση του διακομιστή NTPπου θα χρησιμοποιηθεί για
να πάρει την αναφορά του χρόνου. Εάν δεν ορίζεται, η ώρα του συστήματος θα πρέπει να χρησιμοποιηθεί για να πάρει το χρόνο αναφοράς.

-u | -updateInterval Η τιμή (σε λεπτά) που θα χρησιμοποιηθεί για το χρονικό διάστημα μεταξύ δύο διαδοχικών ενημερώσεις από την αναφορά του χρόνου.

-? -help :Τυπώνει αυτό το μήνυμα βοήθειας.

3.3 OWAMP Session-Receiver:

Usage: OWAMP_SessionReceiver [-options]

όπου τα options περιπλαμβάνουν:

-a | -address : Η τοπική διεύθυνση IP για να συνδεθεί. Αυτή η διεύθυνση μπορεί να χρησιμοποιηθεί σε ένα multi- homed εξυπηρετητή για ένα Session- Receiver που θα δέχεται μόνο αιτήσεις σύνδεσης σε μία από τις διευθύνσεις του. Αν δεν ορίζεται, θα δεχθεί από προεπιλογή συνδέσεις σε οποιεσδήποτε/όλες τις τοπικές διευθύνσεις.

-p | -port port : Η TCP πόρτα που θα χρησιμοποιηθεί στο OWAMP-Control.

-r | -portRange [lowport,highport] : Η UDP πόρτα που θα χρησιμοποιηθεί στο OWAMP-Test sessions.

-s | -sleepVal sleepVal : Ο αριθμός των milliseconds που χρειάζεται να περιμένει μεταξύ της αποστολής των δύο διαδοχικών πακέτων. Αυτή η τιμή χρησιμοποιείται για να αποφευχθεί η υπερφόρτωση της CPU κατά τη διάρκεια της Fetch Sessions.

-k | -keepResults : Αν καθοριστεί, η διαμόρφωση και τα αποτελέσματα της συνεδρίας δοκιμής θα πρέπει να διατηρείται μετά το επεξεργασία ενός Fetch- Session ζητά όλα τα αποτελέσματα της δοκιμής αυτής της συνόδου. Διαφορετικά, τα αποτελέσματα διαγράφονται (που χρησιμοποιείται από προεπιλογή).

-n | -ntpServer : Η διεύθυνση του διακομιστή NTPπου θα χρησιμοποιηθεί για να πάρει την αναφορά του χρόνου. Εάν δεν ορίζεται, η ώρα του συστήματος θα πρέπει να χρησιμοποιηθεί για να πάρει το χρόνο αναφοράς.

-u | -updateInterval Η τιμή (σε λεπτά) που θα χρησιμοποιηθεί για το χρονικό διάστημα μεταξύ δύο διαδοχικών ενημερώσεις από την αναφορά του χρόνου.

-? -help :Τυπώνει αυτό το μήνυμα βοήθειας.

Η sleepVal τιμή χρησιμοποιείται για να ρυθμίσετε το χρονικό διάστημα μεταξύ της αποστολής των δύο διαδοχικών πακέτων εγγραφής, όταν το σύστημα είναι σε λειτουργία fetch. Αυτή η τιμή χρησιμοποιείται για να αποτρέψει την υπερφόρτωση του επεξεργαστή του μηχανήματος (όπου η Session-Receiver είναι εγκαταστημένη).

Εάν η keepResults τιμή ορίζεται, η διαμόρφωση και τα αποτελέσματα της δοκιμής συνόδου θα κρατηθούν μετά την επεξεργασία ενός Fetch- Session. Η διαμόρφωση της συνεδρίας δοκιμής αποθηκεύεται σε ένα αρχείο του οποίου το όνομα αποτελείται από συνεδρίαση SID σειρά με επέκταση ». rs». Αυτό το αρχείο αποθηκεύεται στο receiverDir \ receiverRSdir κατάλογο που βρίσκεται ο κατάλογος από όπου η Session-Receiver βρίσκεται σε λειτουργία. Τα αποτελέσματα της δοκιμής (τα αποτελέσματα που λαμβάνονται για κάθε πακέτο) αποθηκεύονται σε ένα αρχείο του οποίου το όνομα αποτελείται από τη σύνοδο SID σειρά με την επέκταση ". OWP». Αυτό το αρχείο αποθηκεύεται στον
receiverDir \ κατάλογο receiverTestResults που βρίσκονται επίσης στον κατάλογο από όπου ο Session-Receiver βρίσκεται σε λειτουργία. Ο κατάλογος receiverDir δημιουργείται κάθε φορά που η OWAMP Session-Receiver έχει ξεκινήσει. Αν υπάρχει, θα πρέπει να διαγραφεί και να δημιουργηθεί ξανά.

Εάν η keepResults δεν ορίζεται, η διαμόρφωση και τα αποτελέσματα της δοκιμής συνόδου διαγράφονται μετά την επεξεργασία ενός Fetch-Session που ζήτησαν όλα τα αποτελέσματα αυτής της δοκιμής συνεδρίας (επιλογή που χρησιμοποιείται από προεπιλογή).

Αυτές οι γραμμές εντολών πρέπει να εκτελεστούν στον κατάλογο που περιέχει τα εκτελέσιμα αρχεία.

4 Διαμόρφωση J-OWAMP ενοτήτων

Οι ακόλουθες υπο-ενότητες δείχνουν πώς ο χρήστης μπορεί να ρυθμίσει κάθε μονάδα του J-OWAMP σύστηματος μέτρησης.

4.1 OWAMP Server, OWAMP Session-Sender and OWAMP Session-Receiver

Η εκτέλεση κάθε μία από αυτές τις ενότητες μπορεί να γίνει όπως περιγράφεται στην παράγραφο 3.

Η εκτέλεση του εκτελέσιμου OWAMP Server παρουσιάζεται στο παρακάτω παράθυρο:



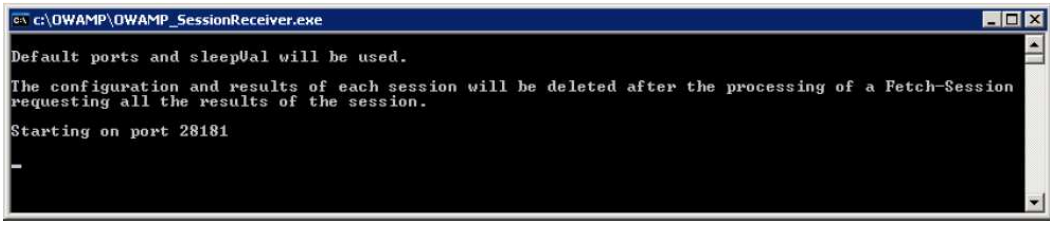
Εικόνα 39: Window illustrating the execution of OWAMP_Server.exe

The execution of OWAMP Session-Sender executable presents the following window:



Εικόνα 40: Window illustrating the execution of OWAMP_SessionSender.exe

The execution of OWAMP Session-Receiver executable presents the following window:



Εικόνα 41: Window illustrating the execution of OWAMP_SessionReceiver.exe

4.2 OWAMP Control-Client

Διαμορφώνει συνεδρίες δοκιμών, και αποθηκεύει τα αποτελέσματα σε αρχείο xml.

Σε αυτόν τον τρόπο λειτουργίας του OWAMP Control-Client εκτελείται με τη χρήση των Windows απο το εκτελέσιμο αρχείο ή από τη γραμμή εντολών χωρίς ορίσματα. Η εκτέλεση αυτής της ενότητας παρουσιάζεται παρακάτω:



Εικόνα 42: Window illustrating the execution of OWAMP_ControlClient.exe

Πριν δωθούν τα ορίσματα για την οποιαδήποτε περίοδο δοκιμών, είναι απαραίτητο να δημιουργήσει μια σύνδεση με διακομιστή OWAMP. Η σύνδεση αυτή γίνεται αυτόματα κατά την εκτέλεση της εφαρμογής JOWAMP_ControlClient.exe

Τα αποδεκτά ορίσματα είναι τα εξής:

-s για την ip του server (default 127.0.0.1),

Η διεύθυνση του server πρέπει να έχει την ακόλουθη μορφή:

IPv4_address:TCPportNumber // for IPv4 address
[IPv6_address]:TCPportNumber // for IPv6 address

Το TCPportNumber είναι η TCP πόρτα που θα χρησιμοποιηθεί στο OWAMP-Control communication μεταξύ του Control-Client και OWAMP-Server. Εάν ο χρήστης θέλει να χρησιμοποιήσει την προεπιλεγμένη πόρτα αυτή είναι η TCPportNumber (22368) και η διεύθυνση του server πρέπει να έχει την ακόλουθη μορφή:

IPv4_address // for IPv4 address
[IPv6_address] // for IPv6 address

-c για την ip του client (default 127.0.0.1),

Η διεύθυνση του client πρέπει να έχει την ακόλουθη μορφή:

IPv4_address:TCPportNumber // for IPv4 address
[IPv6_address]:TCPportNumber // for IPv6 address

Το TCPportNumber είναι η TCP πόρτα που θα χρησιμοποιηθεί στο OWAMP-Control communication μεταξύ του Control-Client και OWAMP-Sender. Εάν ο χρήστης θέλει να χρησιμοποιήσει την προεπιλεγμένη πόρτα αυτή είναι η TCPportNumber (28181) και η διεύθυνση του server πρέπει να έχει την ακόλουθη μορφή:

IPv4_address // for IPv4 address
[IPv6_address] // for IPv6 address

-p για τον αριθμό των πακέτων που θα σταλούν (default 100),

ο αριθμό των δοκιμαστικών πακέτων που θα σταλούν.

-o για το χρονικό διάστημα μεταξύ της αποστολής των πακέτων (default 1 sec),

το χρονικό διάστημα που καθορίζεται για κάθε schedule packet.

-t για το timeout (default 10 sec),

το μέγιστο χρονικό διάστημα κατά το οποίο ένα πακέτο θα πρέπει να ληφθεί. Τα πακέτα που φθάνουν στον δέκτη μετά το χρόνο αυτό θεωρούνται χαμένα,

-l για το padding length (default 0),

επιπλέον μέγεθος που μπορεί να εισαχθεί στα δοκιμαστικά πακέτα,

-n για το ntp sever (default 127.0.0.1:22368),

Η προεπιλεγμένη διεύθυνση είναι 127.0.0.1:22368 (διεύθυνση loopback).

IPv4_address:TCPportNumber // for IPv4 address
[IPv6_address]:TCPportNumber // for IPv6 address

Αν δωθεί διαφορετική διεύθυνση, αυτή θα είναι και η διεύθυνση του διακομιστή NTP. Εάν όχι, ο χρόνος του συστήματος χρησιμοποιείται για να καθορίσει την χρονική αναφορά.

-i για το interval time των updates του χρόνου (default 2, χρειάζεται μόνο αν δώσουμε ntp server),

Το χρονικό διάστημα μεταξύ δύο διαδοχικών ενημερώσεων από την αναφορά του χρόνου είναι η επόμενη παράμετρος για τη διαμόρφωση. Αν δεν δοθεί διεύθυνση απο το προηγούμενο όρισμα τότε η τιμή του interval time γίνεται -1.