

Ανοικτό Πανεπιστήμιο Κύπρου

Σχολή Θετικών και Εφαρμοσμένων Επιστημών

Μεταπτυχιακή Διατριβή στα Πληροφοριακά Συστήματα



Μελέτη Επιθέσεων (Denial Of Service) Σε Πρωτόκολλα Πολυμέσων

Ανδρέας Κοκκαλής

**Επιβλέπων Καθηγητής
Αναστάσιος Νταγιούκλας**

Μάιος 2012

Ανοικτό Πανεπιστήμιο Κύπρου

Σχολή Θετικών και Εφαρμοσμένων Επιστημών

Μηχανισμοί Ανίχνευσης Denial Of Service Attacks Σε Πρωτόκολλα Πολυμέσων

Ανδρέας Κοκκαλής

**Επιβλέπων Καθηγητής
Αναστάσιος Νταγιούκλας**

Η παρούσα μεταπτυχιακή διατριβή υποβλήθηκε
προς μερική εκπλήρωση των απαιτήσεων για απόκτηση

μεταπτυχιακού τίτλου σπουδών
στα Πληροφοριακά Συστήματα

από τη Σχολή Θετικών και Εφαρμοσμένων Επιστημών
του Ανοικτού Πανεπιστημίου Κύπρου

Μάιος 2012

Περίληψη

Η τεχνολογία του Streaming, έχει αποκτήσει μεγάλο ενδιαφέρον και έχει γίνει πολύ δημοφιλές στις μέρες μας. Streaming, είναι η τεχνολογία η οποία αναπτύχθηκε την τελευταία δεκαετία και έχει ως θεμέλια βάση της την τεχνολογία του world wide web. Κύρια ιδέα της τεχνολογίας αυτής είναι η προβολή και η ακοή ψηφιακών αρχείων όπως βίντεο, ήχου και animation από οποιονδήποτε χρήστη ο οποίος έχει πρόσβαση στο διαδίκτυο. Ήδη στην τεχνολογία την οποία περιγράφουμε, έχουμε ξεφύγει από την παλιά διαδικασία του stream η οποία είχε σαν πρώτη ιδέα, ένα ψηφιακό αρχείο (βίντεο, ήχου, animation) να κατεβαίνει στο pc του χρήστη ολόκληρο και μετά να αρχίζει το οποιοδήποτε media application την αναπαραγωγή του. Έχουμε παρέλθει στην τεχνολογία του streaming η οποία αφήνει το αρχείο να αναπαράγεται την ίδια στιγμή που το αρχείο ακόμη κατεβαίνει στην μηχανή του χρήστη. Με αυτή την τεχνολογία έχει μειωθεί στο ελάχιστο ο χρόνος οποίος χρειάζεται ένας χρήστης να περιμένει μέχρι το media application του να ήταν έτοιμο να αρχίζει την διαδικασία αναπαραγωγής του αρχείου.

Ο σκοπός της πτυχιακής είναι να μελετήσει τα προβλήματα ασφάλειας σε πρωτόκολλα πολυμέσων πραγματικού χρόνου. Στα πλαίσια της πτυχιακής θα μελετηθούν τα ζητήματα ασφάλειας και πιο συγκεκριμένα επιθέσεις τύπου Denial of Service (DoS) στο πρωτόκολλο RTSP (Real Time Streaming Protocol).

Το RTSP είναι ένα τυποποιημένο πρωτόκολλο για έναν πελάτη για να αλληλεπιδράσει με το διακομιστή προκειμένου να ζητηθεί και να ελεγχθεί η ροή πολυμέσων. Οι υπηρεσίες πολυμέσων είναι ευπαθείς σε επιθέσεις τύπου Denial Of Service. Συνήθως οι επιθέσεις τύπου Denial Of Service μπορεί να επιτευχθούν στέλνοντας προς το διακομιστή συνεχώς μηνύματα με σκοπό την κατανάλωση των πόρων του streaming server (π.χ. Memory utilization και CPU utilization). Ο στόχος της διπλωματικής είναι να καταγραφούν οι ευπάθειες του πρωτοκόλλου RTSP και να εξομοιωθούν οι επιθέσεις έτσι ώστε να καταγραφεί η επίδραση στους πόρους του διακομιστή.

Τα πειράματα που έχουμε κάνει την περίοδο της διπλωματικής βασίζονται σε κάποια σενάρια τα οποία έχουμε συντάξει εμείς, με απώτερο σκοπό να πετύχουμε άρνηση υπηρεσιών από τον διακομιστή μας και να μετρήσουμε τότε το Memory utilization και CPU utilization .

Κατά την διάρκεια των πειραμάτων, έχουμε δημιουργήσει ένα κώδικα ο οποίος χρησιμοποιεί το εργαλείο open rtsp για να πετύχουμε κάποιες επιθέσεις Denial of Service στον διακομιστή πολυμέσων Darwin Streaming Server.

Το εργαλείο open rtsp είναι ένα πρόγραμμα ανοικτού κώδικα όπου βοηθά στην αναπαραγωγή αρχείων όπου λαμβάνονται από τον χρήστη μέσω του πρωτοκόλλου RTSP. Ο Darwin Streaming Server είναι ένας διακομιστής αναπτυγμένος από την Apple το 1999 και έχει σαν πρωτόκολλο μεταφοράς των αρχείων του το RTSP/RTP.

Ακρώνυμα:

DSS = Darwin Streaming Server

RTSP = Real Time Streaming Protocol

DoS = Denail Of Service

OSI = Open System Interconnection

Summary

The technology of streaming has become very popular in our days. Streaming is a technology that has as a foundation the uses of World Wide Web. The main idea of this technology is the delivery of multimedia files through internet. As a development the name streaming it comes from an oldest technology names stream. The stream definition, is all about that a user has to wait in order the entire file has been downloaded and the starting playing the file. The new term named streaming because the user has the ability to play a file while is steel downloading.

The purpose of this thesis is to study and analyze security holes for RTSP protocol. The aim purpose is to study DoS attacks to RTSP. Multimedia services are vulnerable to DoS attacks. Usually Dos attacks are established while attackers are sending multiple requests to a server forcing those waste resources such as cpu and memory. Target of our study is to cause DoS to a streaming server and collect statistics of its behavior according to cpu and memory consumption.

We will use for our experiments Darwin Streaming Server that has been published from Apple in 1999, OpenRtsp tool that is an open source software tha help us to set the attacks.

Acronyms

DSS = Darwin Streaming Server

RTSP = Real Time Streaming Protocol

DoS = Denail Of Service

OSI = Open System Interconnection

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον Dr. Αναστάσιο Νταγιούκλα, για την υποστήριξη και την καθοδήγηση μέχρι την ολοκλήρωση της εργασίας.

Περιεχόμενα

1	Περιγραφή Πρωτοκόλλου Real Time Streaming Protocol	1
1.1	Εισαγωγή	1
1.2	Ορισμός Streaming και Downloading	3
1.3	Το Πρωτόκολλο RTSP	4
1.3.1	Χαρακτηριστικά Πρωτόκολλου RTSP.....	4
1.3.2	Λειτουργικότητα Πρωτόκολλου RTSP	5
1.3.3	Τρόποι Παράδοσης Υψηρεσιών Streaming	6
1.4	Οι Μέθοδοι του Πρωτοκόλλου RTSP	7
1.4.1	Options Method	9
1.4.2	Describe Method	9
1.4.3	Announce Method	10
1.4.4	Setup Method	11
1.4.5	Play Method	12
1.4.6	Pause Method	12
1.4.7	Teardown Method	13
1.4.8	Set_Parameter Method	13
1.4.9	Redirect Method	14
1.4.10	Record Method	14
1.4.11	Σχεδιάγραμμα Μίας Συνόδου Μεταξύ Πελάτη και Διακομιστή Με Την Ανταλλαγή Των Πακέτων Να Γίνεται Με Το RTSP	15
2	Ευπάθειες Του Πρωτοκόλλου Real Time Streaming Protocol	17
2.1	Γενική Περιγραφή	18
2.2	Επιθέσεις Τύπου Υπερχείλισης	18
2.2.1	Describe Message Attack.....	19
2.2.2	Options Message Attack.....	20
2.2.3	Get Parameters Message Attack.....	21
2.2.4	Set Parameters Message Attack.....	22
2.3	Επιθέσεις Τύπου Full Connections.....	23
2.3.1	Set_Up Message Attack.....	23
2.3.2	Play Attack	25

2.3.3	Record Attack	26
2.4	Επιθέσεις Τύπου Malformed Requests Attacks	27
2.4.1	Content_Len Attack	27
2.4.2	Teardown Attack	29
2.4.3	Hijack Attack	30
3	Πιθανά Σενάρια Επιθέσεων Τύπου Υπερχείλισης Στον Darwin Streaming Server	31
3.1	Εργαλείο OpenRtsp	33
3.1.1	Περιγραφή Εργαλείου OpenRtsp	33
3.1.1	Περιγραφή Εργαλείου OpenRtsp	33
3.2	Σενάρια Επιθέσεων Χρησιμοποιώντας Την Απλή Εντολή Του OpenRtsp	37
3.2.1	Περιγραφή Πειράματος Χρησιμοποιώντας Την Απλή Εντολή Του OpenRtsp Και Χρόνο Έκδοσης Των Αιτημάτων Μία Σταθερή Μεταβλητή	37
3.2.2	Περιγραφή Πειράματος Χρησιμοποιώντας Την Απλή Εντολή Του OpenRtsp Και Χρόνο Έκδοσης Των Αιτημάτων Μία Τυχαία Μεταβλητή	38
3.3	Σενάρια Επιθέσεων Χρησιμοποιώντας Την Απλή Εντολή Του OpenRtsp Ζητώντας Διαφορετικά Αρχεία Από Τον Streaming Server	39
3.3.1	Περιγραφή Πειράματος Χρησιμοποιώντας Την Απλή Εντολή Του OpenRtsp Ζητώντας Διαφορετικά Αρχεία Από Τον Streaming Server Και Χρόνο Έκδοσης Των Αιτημάτων Μία Σταθερή Μεταβλητή	39
3.3.2	Περιγραφή Πειράματος Χρησιμοποιώντας Την Απλή Εντολή Του OpenRtsp Ζητώντας Διαφορετικά Αρχεία Από Τον Streaming Server Και Χρόνο Έκδοσης Των Αιτημάτων Μία Σταθερή Μεταβλητή	40
3.4	Σενάρια Επιθέσεων Χρησιμοποιώντας Την Απλή Εντολή Του OpenRtsp Ζητώντας Κάποιο Αρχείο Που Δεν Βρίσκεται στον Streaming Server	41
3.4.1	Περιγραφή Πειράματος Χρησιμοποιώντας Την Απλή Εντολή Του OpenRtsp Ζητώντας Κάποιο Αρχείο Που Δεν Βρίσκεται στον Streaming Server Και Χρόνο Έκδοσης Των Αιτημάτων Μία Σταθερή Μεταβλητή	41
3.4.2	Περιγραφή Πειράματος Χρησιμοποιώντας Την Απλή Εντολή Του OpenRtsp Ζητώντας Κάποιο Αρχείο Που Δεν Βρίσκεται στον Streaming Server Και Χρόνο	

	Έκδοσης Των Αιτημάτων Μία Τυχαία Μεταβλητή.....	42
3.5	Σενάρια Επιθέσεων Χρησιμοποιώντας Την Απλή Εντολή Του OpenRtsp Ζητώντας Αρχεία Που Δεν Βρίσκονται στον Streaming Server	43
3.5.1	Περιγραφή Πειράματος Χρησιμοποιώντας Την Απλή Εντολή Του OpenRtsp Ζητώντας Αρχεία Που Δεν Βρίσκονται στον Streaming Server Και Χρόνο Έκδοσης Των Αιτημάτων Μία Σταθερή Μεταβλητή.....	43
3.5.2	Περιγραφή Πειράματος Χρησιμοποιώντας Την Απλή Εντολή Του OpenRtsp Ζητώντας Αρχεία Που Δεν Βρίσκονται στον Streaming Server Και Χρόνο Έκδοσης Των Αιτημάτων Μία Τυχαία Μεταβλητή.....	44
3.6	Σενάρια Επιθέσεων Χρησιμοποιώντας Την Εντολή Του OpenRtsp Για Quality Of Service.....	45
3.6.1	Περιγραφή Πειράματος Χρησιμοποιώντας Την Εντολή Του OpenRtsp Για Quality Of Service Και Χρόνο Έκδοσης Των Αιτημάτων Μία Σταθερή Μεταβλητή. ...	45
3.6.2	Περιγραφή Πειράματος Χρησιμοποιώντας Την Εντολή Του OpenRtsp Για Quality Of Service Και Χρόνο Έκδοσης Των Αιτημάτων Μία Τυχαία Μεταβλητή. ...	46
3.7	Σενάρια Επιθέσεων Χρησιμοποιώντας Την Εντολή Του OpenRtsp Για Quality Of Service Ζητώντας Διαφορετικά Αρχεία	47
3.7.1	Περιγραφή Πειράματος Χρησιμοποιώντας Την Εντολή Του OpenRtsp Για Quality Of Service Ζητώντας Διαφορετικά Αρχεία Και Χρόνο Έκδοσης Των Αιτημάτων Μία Σταθερή Μεταβλητή.....	47
3.7.2	Περιγραφή Πειράματος Χρησιμοποιώντας Την Εντολή Του OpenRtsp Για Quality Of Service Ζητώντας Διαφορετικά Αρχεία Και Χρόνο Έκδοσης Των Αιτημάτων Μία Τυχαία Μεταβλητή.....	48
3.8	Σενάρια Επιθέσεων Χρησιμοποιώντας Την Εντολή Του OpenRtsp Για Αλλαγή Στην Διάρκεια Του Αρχείου	48
3.8.1	Περιγραφή Πειράματος Χρησιμοποιώντας Την Εντολή Του OpenRtsp Για Αλλαγή Στην Διάρκεια Αναπαραγωγής Του Αρχείου Και Χρόνο Έκδοσης Των Αιτημάτων Μία Σταθερή Μεταβλητή.....	49
3.8.2	Περιγραφή Πειράματος Χρησιμοποιώντας Την Εντολή Του OpenRtsp Για Αλλαγή Στην Διάρκεια Αναπαραγωγής Του Αρχείου Και Χρόνο Έκδοσης Των Αιτημάτων Μία Τυχαία Μεταβλητή.....	50

3.9	Σενάρια Επιθέσεων Χρησιμοποιώντας Την Εντολή Του OpenRtsp Για Αλλαγή Στην Διάρκεια Αναπαραγωγής Του Αρχείου Και Quality Of Service.....	51
3.9.1	Περιγραφή Πειράματος Χρησιμοποιώντας Την Εντολή Του OpenRtsp Για Αλλαγή Στην Διάρκεια Αναπαραγωγής Του Αρχείου Και Quality Of Service Και Χρόνο Έκδοσης Των Αιτημάτων Μία Σταθερή Μεταβλητή	51
3.9.2	Περιγραφή Πειράματος Χρησιμοποιώντας Την Εντολή Του OpenRtsp Για Αλλαγή Στην Διάρκεια Αναπαραγωγής Του Αρχείου Και Quality Of Service Και Χρόνο Έκδοσης Των Αιτημάτων Μία Τυχαία Μεταβλητή	52
4	Προτεινόμενες Λύσεις Για Πιθανές Επιθέσεις Denial Of Service	54
4.1	Προτεινόμενες Λύσεις.....	55
4.1.1	Προτεινόμενη Λύση για αντιμετώπιση πιθανής επίθεσης Describe Message.....	55
4.1.2	Προτεινόμενη Λύση για αντιμετώπιση πιθανής επίθεσης Options Message.....	55
4.1.3	Προτεινόμενη Λύση για αντιμετώπιση πιθανής επίθεσης Get_Parameters Message.....	55
4.1.4	Προτεινόμενη Λύση για αντιμετώπιση πιθανής επίθεσης Teardown.....	56
4.1.5	Προτεινόμενη Λύση για αντιμετώπιση πιθανής επίθεσης Content_Len.....	56
4.1.6	Προτεινόμενη Λύση για αντιμετώπιση πιθανής επίθεσης Record.....	56
4.1.7	Προτεινόμενη Λύση για αντιμετώπιση πιθανής επίθεσης Play.....	57
4.1.8	Προτεινόμενη Λύση για αντιμετώπιση πιθανής επίθεσης Set_Up message.....	57
4.1.9	Προτεινόμενη Λύση για αντιμετώπιση πιθανής επίθεσης Set_Parameters.....	58
4.1.10	Προτεινόμενη Λύση για αντιμετώπιση πιθανής επίθεσης Hijack.....	58
5	Παρουσίαση Στατιστικών Μετρήσεων Ανά Σενάριο	59
5.1	Στατιστικά Επιθέσεων Χρησιμοποιώντας Την Απλή Εντολή Του OpenRtsp.....	60
5.1.1	Ανάλυση Αποτελεσμάτων Σεναρίου Χρησιμοποιώντας Την Απλή Εντολή Του OpenRtsp Και Χρόνο Έκδοσης Των Αιτημάτων Μία Σταθερή Μεταβλητή	60
5.1.2	Ανάλυση Αποτελεσμάτων Σεναρίου Χρησιμοποιώντας Την Απλή Εντολή Του	

	OpenRtsp Και Χρόνο Έκδοσης Των Αιτημάτων Μία Τυχαία Μεταβλητή.....	63
5.1.3	Ανάλυση Αποτελεσμάτων Σεναρίου Χρησιμοποιώντας Την Απλή Εντολή Του OpenRtsp Ζητώντας Διαφορετικά Αρχεία Από Τον Streaming Server Και Χρόνο Έκδοσης Των Αιτημάτων Μία Σταθερή Μεταβλητή.....	66
5.1.4	Ανάλυση Αποτελεσμάτων Σεναρίου Χρησιμοποιώντας Την Απλή Εντολή Του OpenRtsp Ζητώντας Διαφορετικά Αρχεία Από Τον Streaming Server Και Χρόνο Έκδοσης Των Αιτημάτων Μία Τυχαία Μεταβλητή.....	69
5.1.5	Ανάλυση Αποτελεσμάτων Σεναρίου Χρησιμοποιώντας Την Απλή Εντολή Του OpenRtsp Ζητώντας Κάποιο Αρχείο Που Δεν Βρίσκεται Στον Streaming Server Και Χρόνο Έκδοσης Των Αιτημάτων Μία Σταθερή Μεταβλητή.....	72
5.1.6	Ανάλυση Αποτελεσμάτων Σεναρίου Χρησιμοποιώντας Την Απλή Εντολή Του OpenRtsp Ζητώντας Κάποιο Αρχείο Που Δεν Βρίσκεται Στον Streaming Server Και Χρόνο Έκδοσης Των Αιτημάτων Μία Τυχαία Μεταβλητή.....	74
5.1.7	Ανάλυση Αποτελεσμάτων Σεναρίου Χρησιμοποιώντας Την Απλή Εντολή Του OpenRtsp Ζητώντας Αρχεία Που Δεν Βρίσκονται Στον Streaming Server Και Χρόνο Έκδοσης Των Αιτημάτων Μία Σταθερή Μεταβλητή.....	76
5.1.8	Ανάλυση Αποτελεσμάτων Σεναρίου Χρησιμοποιώντας Την Απλή Εντολή Του OpenRtsp Ζητώντας Αρχεία Που Δεν Βρίσκονται Στον Streaming Server Και Χρόνο Έκδοσης Των Αιτημάτων Μία Τυχαία Μεταβλητή.....	78
5.1.9	Ανάλυση Αποτελεσμάτων Σεναρίου Χρησιμοποιώντας Την Εντολή Του OpenRtsp Για Quality Of Service Και Χρόνο Έκδοσης Των Αιτημάτων Μία Σταθερή Μεταβλητή	80
5.1.10	Ανάλυση Αποτελεσμάτων Σεναρίου Χρησιμοποιώντας Την Εντολή Του OpenRtsp Για Quality Of Service Και Χρόνο Έκδοσης Των Αιτημάτων Μία Τυχαία Μεταβλητή...	83
5.1.11	Ανάλυση Αποτελεσμάτων Σεναρίου Χρησιμοποιώντας Την Εντολή Του OpenRtsp Για Quality Of Service Ζητώντας Διαφορετικά Αρχεία Και Χρόνο Έκδοσης Των Αιτημάτων Μία Σταθερή Μεταβλητή	86
5.1.12	Ανάλυση Αποτελεσμάτων Σεναρίου Χρησιμοποιώντας Την Εντολή Του OpenRtsp Για Quality Of Service Ζητώντας Διαφορετικά Αρχεία Και Χρόνο Έκδοσης Των Αιτημάτων Μία Τυχαία Μεταβλητή	88
5.1.13	Ανάλυση Αποτελεσμάτων Σεναρίου Χρησιμοποιώντας Την Εντολή Του OpenRtsp Για Αλλαγή στην Διάρκεια Αναπαραγωγής Του Αρχείου Και Χρόνο Έκδοσης Των Αιτημάτων Μία Σταθερή Μεταβλητή	90
5.1.14	Ανάλυση Αποτελεσμάτων Σεναρίου Χρησιμοποιώντας Την Εντολή Του OpenRtsp Για Αλλαγή στην Διάρκεια Αναπαραγωγής Του Αρχείου Και Χρόνο Έκδοσης	

Των Αιτημάτων Μία Τυχαία Μεταβλητή	93
5.1.15 Ανάλυση Αποτελεσμάτων Σεναρίου Χρησιμοποιώντας Την Εντολή Του OpenRtsp Για Αλλαγή στην Διάρκεια Αναπαραγωγής Του Αρχείου Και Quality Of Service Και Χρόνο Έκδοσης Των Αιτημάτων Μία Σταθερή Μεταβλητή	96
5.1.16 Ανάλυση Αποτελεσμάτων Σεναρίου Χρησιμοποιώντας Την Εντολή Του OpenRtsp Για Αλλαγή στην Διάρκεια Αναπαραγωγής Του Αρχείου Και Quality Of Service Και Χρόνο Έκδοσης Των Αιτημάτων Μία Τυχαία Μεταβλητή	99
Βιβλιογραφία	102

Κεφάλαιο 1

Περιγραφή Πρωτοκόλλου Real Time Streaming Protocol

Σε αυτό το κεφάλαιο παραθέτουμε κάποια σχετική δουλειά αναφορικά με το πρωτόκολλο RTSP. Ουσιαστικά θα περιγράψουμε το πρωτόκολλο αναφέροντας την χρησιμότητα του, τα πλεονεκτήματα του και την λειτουργία του.

1.1 Εισαγωγή

Σκοπός της πτυχιακής είναι να μελετήσει τα προβλήματα ασφάλειας σε πρωτόκολλα πολυμέσων πραγματικού χρόνου. Στα πλαίσια της πτυχιακής θα μελετηθούν τα ζητήματα ασφάλειας και πιο συγκεκριμένα επιθέσεις τύπου Denial of Service (DoS) στο πρωτόκολλο RTSP (Real Time Streaming Protocol).

Το RTSP είναι ένα πρωτόκολλο σχεδιασμένο για να χρησιμοποιείται σε εφαρμογές ψυχαγωγίας και σε συστήματα επικοινωνίας. Το πρωτόκολλο επίσης χρησιμοποιείται για τον έλεγχο της ροής δεδομένων από τον διακομιστή προς τους τελικούς χρήστες. Το πρωτόκολλο περιγράφεται εκτενώς πιο κάτω στο κεφάλαιο ένα.

Στο επόμενο κεφάλαιο δύο θα μελετήσουμε κάποια τρωτά σημεία του πρωτοκόλλου και πώς μπορεί κάποιο επιβλαβές λογισμικό να το εκμεταλλευτεί και να επιτύχει επιθέσεις στον διακομιστή μας τύπου Denial of Service (DoS).

Στο κεφάλαιο τρία αναφέρονται και αναλύονται αρκετά σενάρια επιθέσεων όπου με την χρήση κάποιου δικού μας προγράμματος το οποίο καλεί το εργαλείο OpenRtsp θα επιτύχουμε αυτές τις επιθέσεις και θα πάρουμε τις μετρήσεις από τον Darwin Streaming Server. Για κάθε σενάριο θα χρησιμοποιούμε μία συγκεκριμένη εντολή `top -b -d 5 > file_name.txt` στο λειτουργικό LINUX FEDORA 11 για να μπορέσουμε να καταγράψουμε την μεταβολή της κατάστασης του διακομιστή πριν την επίθεση και κατά την διάρκεια της επίθεσης ως προς το CPU processing και Memory usage του διακομιστή.

Στο κεφάλαιο τέσσερα παραθέτουμε κάποιες πιθανές λύσεις, για αντιμετώπιση επιθέσεων τύπου DoS στον διακομιστή μας.

Τέλος στο κεφάλαιο πέντε θα παρουσιάσουμε και θα αναλύσουμε όλες τις μετρήσεις που πήραμε από το εργαστήριο μας.

1.2 Ορισμός Streaming και Downloading

Ο ορισμός του Streaming, αναφέρεται σε αρχεία τα οποία είναι τοποθετημένα, αποθηκευμένα σε κάποια μηχανή A και το αρχείο αρχίζει να παίζει πάνω στην μηχανή B από την οποία ζητήθηκε, χωρίς όμως να έχει κατεβεί ολόκληρο το αρχείο στην μηχανή B. Ένα αρχείο μπορεί να βρίσκεται χιλιόμετρα μακριά αποθηκευμένο σε μια μηχανή ή να βρίσκεται στο διπλανό δωμάτιο, αποθηκευμένο σε μια μηχανή. Το αρχείο δεν χρειάζεται να αποθηκευτεί στην μηχανή από την οποία ζητήθηκε για να αρχίσει την αναπαραγωγή. Απλά όταν στην μηχανή του χρήστη αρχίζουν να κατεβαίνουν τα πρώτα πακέτα του αρχείου, τότε ο media player αρχίζει την αναπαραγωγή. Συνήθως, σε κάθε streaming session, υπάρχει το λεγόμενο «buffer size», το οποίο καθορίζεται από τον διακομιστή, και στο οποίο κατεβαίνουν αρκετά δευτερόλεπτα από το αρχείο πριν να αρχίσει η αναπαραγωγή το. Αυτό γίνεται για να αποφύγουμε τις διακοπές στην αναπαραγωγή του αρχείου οι οποίες μπορεί να προκύπτουν από μια κακή σύνδεση. Έχουμε μεταβεί στην τεχνολογία του streaming η οποία αφήνει το αρχείο να αναπαράγεται την ίδια στιγμή που το αρχείο ακόμη κατεβαίνει στην μηχανή του χρήστη. Με αυτή την τεχνολογία έχει μειωθεί στο ελάχιστο ο χρόνος οποίος χρειάζονται ένας χρήστης να περιμένει μέχρι το media application του να ήταν έτοιμο να αρχίζει την διαδικασία αναπαραγωγής του αρχείου.

Ο ορισμός του Downloading, αναφέρεται σε κάποια αρχεία τα οποία βρίσκονται σε κάποια άλλη μηχανή, και τα οποία για να αρχίσει η αναπαραγωγή τους θα πρέπει να κατεβούν στην μηχανή την οποία ζητήθηκαν. Συνήθως για να αρχίσει η αναπαραγωγή του αρχείου πρέπει το αρχείο να κατεβεί ολόκληρο, ωστόσο υπάρχουν υπηρεσίες που αφήνουν τα αρχεία τους να αναπαράγονται όταν κατεβεί κάποιος αριθμός από megabytes στον υπολογιστή. Ο χρήστης μπορεί να αντιγράψει το αρχείο το οποίο κατέβασε, να το μεταφέρει και να αρχίσει την αναπαραγωγή του όποτε το θελήσει.

Σήμερα διαβάζουμε και βλέπουμε συνεχώς διαφημίσεις για live-streaming υπηρεσίες. Οι υπηρεσίες αυτές συνήθως έχουν να κάνουν με την μετάδοση ποδοσφαιρικών αγώνων και γενικά αθλητικών γεγονότων ή ακόμη και κάποιες ιστοσελίδες προσφέρουν υπηρεσίες για live streaming καναλιών τηλεόρασης. Όλα αυτά ασφαλώς δεν μπορούσαν να στηριχθούν και να γίνουν χρήσιμα εάν χρησιμοποιείτο η τεχνολογία του downloading, αφού θα έπρεπε να τελειώσει κάποιο γεγονός και μετά να αρχίσει η λήψη και η αναπαραγωγή του από τον χρήστη. Αντίθετα με την τεχνολογία του streaming, μπορούμε να μεταδίδουμε γεγονότα ασφαλώς με ελάχιστη καθυστέρηση, προσφέροντας έτσι καλύτερες υπηρεσίες προς τον τελικό χρήστη.

1.3 Το Πρωτόκολλο RTSP

Το RTSP είναι ένα πρωτόκολλο βασισμένο για να εξυπηρετεί το μοντέλο ροής δεδομένων client-διακομιστή ή πελάτη-διακομιστή. Με βάση τα επίπεδα του OSI model το RTSP είναι τοποθετημένο στο επίπεδο του application και είναι υπεύθυνο για την ροή δεδομένων μεταξύ χρήστη και διακομιστή σε πραγματικό χρόνο. Μπορεί να διαχειριστεί αρκετά αρχεία την ίδια χρονική στιγμή, στον ίδιο χρήστη και να προβάλλει στον χρήστη χωρίς παρεμβολές. Το πρωτόκολλο RTSP σε πιο κάτω network layers χρησιμοποιεί στο transport layer, πρωτόκολλα όπως το TCP, το UDP, και το RTP. Το πρωτόκολλο μπορεί να χρησιμοποιηθεί για την μετάδοση ζωντανών δεδομένων (αρχείων) ή και για την μετάδοση αποθηκευμένων αρχείων. Το πρωτόκολλο προσφέρει στον χρήστη υπηρεσίες όπως play, pause, stop, fast forward και rewind. Το πρωτόκολλο χρησιμοποιεί την θύρα 554.

1.3.1 Χαρακτηριστικά του πρωτόκολλου RTSP

Το κυριότερο χαρακτηριστικό αλλά και πλεονέκτημα που βρίσκουμε στο RTSP είναι ότι μπορεί αναλόγως του bandwidth το οποίο είναι διαθέσιμο μεταξύ διακομιστή και client, να σπάσει τα πακέτα της κάθε συνόδου ούτως ώστε να προσαρμόζονται στο αντίστοιχο bandwidth.

Επίσης το αρχείο μπορεί να αρχίσει να αναπαράγεται εφόσον ο player του user έχει κατεβάσει αρκετά πακέτα, μπορεί να αρχίσει να αναπαράγει το ένα πακέτο, να αποσυμπιέζει κάποιο άλλο, και να κατεβάσει κάποιο άλλο την ίδια χρονική στιγμή. Έτσι έχουμε σχεδόν χωρίς κάποια καθυστέρηση (playout delay)εκτός από την λογική καθυστέρηση του να ταξιδέψει ένα πακέτο από τον διακομιστή στον client, καμία άλλη καθυστέρηση. Playout Delay αναφέρουμε την καθυστέρηση η οποία παρουσιάζεται από το πρώτο αίτημα ανάκτησης κάποιου πολυμέσου μέχρι και την έναρξη της αναπαραγωγής του από τον media player.

Το RTSP χρησιμοποιεί δύο client/server socket, το κάθε ένα έχει την δική του θύρα, το ένα χρησιμοποιείται για την ανταλλαγή των πακέτων, του αρχείου και το άλλο για την ανταλλαγή πληροφοριών έλεγχου όπως pause, play.

Το RTSP παραμετροποιείται πολύ εύκολα αφού μπορούν να προστεθούν νέες μέθοδοι και παράμετροι εύκολα.

Το RTSP χρησιμοποιεί όλους τους μηχανισμούς ασφαλείας όπως το http authentication και digest authentication. Επίσης χρησιμοποιεί τους μηχανισμούς ασφαλείας που παρέχει το επίπεδο μεταφοράς.

Το RTSP μπορεί να χρησιμοποιεί για την μεταφορά του αρχείου κάποιο αναξιόπιστο πρωτόκολλο μεταφοράς όπως το UDP ή να χρησιμοποιεί κάποιο αξιόπιστο πρωτόκολλο όπως το TCP.

Κάθε αρχείο μίας συνεδρίας το οποίο προβάλλεται στον χρήστη μπορεί να βρίσκονται σε διαφορετικούς διακομηστές. Ο πελάτης αμέσως δημιουργεί ταυτόχρονες συνδέσεις με όλους τους εμπλεκόμενους διακομηστές για να ανακτήσει τα διάφορα αρχεία. Ο συγχρονισμός αυτός γίνεται στο επίπεδο μεταφοράς.

1.3.2 Λειτουργικότητα του πρωτόκολλου RTSP

Για να ανακτηθεί ένα αρχείο από ένα media διακομηστή, μπορεί ο client να ζητήσει μέσω http το αρχείο. Εάν το describe message είναι πολλαπλής αποστολής σε μία συγκεκριμένη ομάδα από χρήστες (multicast), τότε το μήνυμα περιέχει την multicast διεύθυνση και την θύρα που θα χρησιμοποιηθεί για ανάκτηση του αρχείου. Εάν το αρχείο θα αποσταλεί μόνο στον πελάτη τότε ο πελάτης αναγράφει στο μήνυμα τον προορισμό.

Ένας media server μπορεί να προσκληθεί για να συνδεθεί με ένα υπάρχον συνέδριο, είτε για να αναπαράγει ξανά σε κάποια άλλη ομάδα από χρήστες το αρχείο την ίδια χρονική στιγμή, είτε για να καταγράψει το γεγονός σε ένα αρχείο.

Επιτρέπει την μετάδοση real time events από τον διακομιστή σε πολλούς χρήστες την ίδια χρονική στιγμή.

Επιτρέπει την αλληλοεπίδραση μεταξύ client-server από διαφορετικά brand names μηχανές, αφού χρησιμοποιεί τα διεθνές πρότυπα που έχουν δοθεί από το IEFF.

Το RTSP , έχει σχεδιαστεί για να καλύπτει και να αντιμετωπίζει προβλήματα, τα οποία μπορεί να προκύψουν στην διάρκεια του session όπως την ποιότητα της υπηρεσίας, με την οποία μεταδίδεται ένα αρχείο. Επομένως, ο χρήστης ο οποίος θα χρησιμοποιήσει το RTSP για να παρακολουθήσει ένα γεγονός, αλλά και ο παροχέας ο οποίος θα προσφέρει το κάθε αρχείο δεν χρειάζεται να ανησυχούν για αυτά τα προβλήματα , αφού το RTSP, έχει τους κατάλληλους μηχανισμούς να αντιμετωπίζει τέτοια φύσεως προβλήματα.

1.3.3 Τρόποι Παράδοσης Υπηρεσιών Streaming

Σε αυτή την ενότητα θα περιγράψουμε τους τρόπους παράδοσης ενός αρχείου στον χρήστη μέσω υπηρεσιών Streaming.

Υπάρχουν δύο κύριοι τρόποι παράδοσης υπηρεσιών Streaming. Ο ένας τρόπος είναι ζωντανή μετάδοση κάποιου γεγονότος και ο δεύτερος τρόπος «κατά παραγγελία» On Demand.

On Demand παράδοση εννοούμε ότι κάποιο γεγονός, σε κάποια άγνωστη παρελθοντική χρονική στιγμή, έχει καταγραφεί από κάποιο διακομιστή πολυμέσων και μπορεί να προσφερθεί στον τελικό χρήστη οποτεδήποτε το ζητήσει. Σε αυτού του είδους παράδοσης ενός αρχείου ο τελικός χρήστης έχει την δυνατότητα να κάνει παύση στο αρχείο, fast forward, rewind.

Στην ζωντανή μετάδοση κάποιου γεγονότος γίνεται σε καθορισμένη χρονική στιγμή και μπορείς να παρακολουθήσεις γεγονότα που βρίσκονται χιλιόμετρα μακριά με ελάχιστη καθυστέρηση. Ασφαλώς λόγω του γεγονότος ότι το συμβάν το οποίο παρουσιάζεται είναι ζωντανό, ο χρήστης δεν έχει την δυνατότητα να κάνει παύση στο αρχείο, fast forward, rewind.

1.4 Οι Μέθοδοι του Πρωτόκολλου Real Time Streaming

Σε αυτή την ενότητα θα επεξηγήσουμε τις πιο σημαντικές μεθόδους του πρωτοκόλλου.

Οι μέθοδοι στο πρωτόκολλο RTSP, είναι τα μηνύματα τα οποία ανταλλάζει ένας χρήστης με τον επικείμενο διακομιστή για να μπορέσει να δημιουργηθεί και να λειτουργήσει η ζητούμενη σύνδεση. Οι μέθοδοι του πρωτοκόλλου είναι describe, options, setup, play, pause, record, announce, redirect, get-parameter, set-parameter και teardown. Ένας διακομιστής εμπερικλείει το αρχείο περιγραφής της παρουσιάσεις σε ένα HTTP μήνυμα, και το αποστέλλει στον φυλλομετρητή. Όταν ο φυλλομετρητής παραλάβει το μήνυμα, επικαλείται κάποιο media player ο οποίος είναι ένα βοηθητικό πρόγραμμα για την αναπαραγωγή του αρχείου.

Το αρχείο παρουσίασης περιέχει το ζητούμενο URL για το ανάλογο media αρχείο. Αμέσως μετά ο player, αποστέλλει ένα RTSP setup request, και ο διακομιστής απαντά με ένα RTSP OK μήνυμα. Ο player αποστέλλει ένα RTSP play request, ως υποθέσουμε για ένα αρχείο ήχου. Ο διακομιστής επομένως απαντά με ένα RTSP Ok. Σε αυτό το σημείο ο διακομιστής τοποθετεί το αρχείο στο δικό του εσωτερικό κανάλι σύνδεσης, για να μπορέσει να μεταδοθεί στον χρήστη. Ακολούθως ο χρήστης αποστέλλει στον διακομιστή ένα RTSP pause request, που είναι υπεύθυνο για να κάνει παύση την αναπαραγωγή του αρχείου. Ο διακομιστής του απαντάει με ένα RTSP Ok message. Όταν ο χρήστης τελειώνει με την αναπαραγωγή του αρχείου αποστέλλει στον διακομιστή ένα RTSP teardown request, και ο διακομιστής απάντα με ένα RTSP ok message, για να κλείνει η σύνδεση. Τα methods τα οποία έχουν τον πιο καθοριστικό ρόλο σε αυτή την επικοινωνία είναι το setup method, το play method, το pause method και το teardown.

Method Summary:

<u>Method</u>	<u>Direction</u>
DESCRIBE	C->S
ANNOUNCE	C->S, S->C

GET_PARAMETER	C->S, S->C
OPTIONS	C->S, S->C
PAUSE	C->S
PLAY	C->S
RECORD	C->S
REDIRECT	S->C
SETUP	C->S
SET_PARAMETER	C->S, S->C
TEARDOWN	C->S

Πίνακας 1.1: Οι μέθοδοι του πρωτοκόλλου σε συντομία και οι κατευθύνσεις από και προς μία κατεύθυνση. C= client, S = Server.

1.4.1 Options Method

Ένα option request μπορεί να σταλεί από ένα client προς τον διακομιστή οποιαδήποτε χρονική στιγμή. Συνήθως δεν επηρεάζει την κατάσταση του διακομιστή.

Παράδειγμα:

```
C->S: OPTIONS * RTSP/1.0
```

```
CSeq: 1
```

```
Require: implicit-play
```

```
Proxy-Require: gzipped-messages
```

```
S->C: RTSP/1.0 200 OK
```

```
CSeq: 1
```

```
Public: DESCRIBE, SETUP, TEARDOWN, PLAY, PAUSE
```

1.4.2 Describe Method

Η μέθοδος describe, ανακτά πληροφορίες σχετικά με την επερχόμενη παρουσίαση του αντικειμένου πολυμέσων που θα παρουσιαστεί στην σύνοδο (session) και αναγνωρίζεται από το URL στο οποίο ζητάτε από ένα client προς τον διακομιστή. Μπορεί επίσης να χρησιμοποιηθεί το header accept για να καθοριστεί κάποιος είδους περιγραφή η οποία είναι

κατανοητή και από τον client. Ο Διακομιστής απαντά με την περιγραφή του ζητούμενου πολυμέσου. Η ανταλλαγή αυτή των μηνυμάτων reply-response αρχικοποιούν την κατάσταση RTSP του πολυμέσου.

Παράδειγμα:

```
C->S: DESCRIBE rtsp://server.example.com/fizzle/foo RTSP/1.0
      CSeq: 312
      Accept: application/sdp, application/rtsl, application/mhég

S->C: RTSP/1.0 200 OK
      CSeq: 312
      Date: 23 Jan 1997 15:35:06 GMT
      Content-Type: application/sdp
      Content-Length: 376

      v=0
      o=mhandley 2890844526 2890842807 IN IP4 126.16.64.4
      s=SDP Seminar
```

1.4.3 Announce Method

Η μέθοδος announce εξυπηρετεί δύο προϋποθέσεις. Όταν στέλνεται από τον client στον διακομιστή, ανακοινώνει την περιγραφή της παρουσιάσεις ή των πολυμέσων το οποίο αρχικοποιείται από το URL που ζητείται από τον διακομιστή. Όταν αποστέλλεται από τον διακομιστή στον client, η μέθοδος ανανεώνει την περιγραφή της συνόδου σε πραγματικό χρόνο. Εάν ένα νέο πολυμέσο προστεθεί στην σύνοδο κατά την διάρκεια της παρουσιάσεις, θα σταλεί ολόκληρη περιγραφή της παρουσίασης αντί να σταλεί η περιγραφή μόνο του νέου περιεχομένου.

Παράδειγμα:

```
C->S: ANNOUNCE rtsp://server.example.com/fizzle/foo RTSP/1.0
      CSeq: 312
      Date: 23 Jan 1997 15:35:06 GMT
      Session: 47112344
      Content-Type: application/sdp
      Content-Length: 332

      v=0
      o=mhandley 2890844526 2890845468 IN IP4 126.16.64.4
```

1.4.4 Setup Method

Το αίτημα του setup που ζητείτε για ένα URI καθορίζει τον μηχανισμό μεταφοράς που θα χρησιμοποιηθεί για ένα συγκεκριμένο αρχείο. Ένας client μπορεί να στέλλει ένα setup request, για κάποιο αρχείο το οποίο ήδη παίζει στον συγκεκριμένο χρήστη, για να αλλάξει τις παραμέτρους μεταφοράς, της οποίες επιτρέπει ένας συγκεκριμένος διακομιστής. Εάν ο διακομιστής δεν επιτρέπει αυτές τις αλλαγές θα απαντήσει στον client με error message “Less method not valid in this state”. Από την στιγμή όπου το μήνυμα setup παρέχει όλες της πληροφορίες μεταφοράς η συσκευές firewalls και οποιεσδήποτε ενδιάμεσες συσκευές μεταξύ client και διακομιστή η οποίες χρειάζονται αυτές της πληροφορίες τότε μπορούν να περάσουν το describe response το οποίο έχει ήδη κρατηθεί για την αρχικοποίηση του αρχείου. Το transport reader, διευκρινίζει τις παραμέτρους μεταφοράς η οποίες είναι αποδεκτές από τον client για την μεταφορά των δεδομένων. Η απάντηση θα περιέχει της παραμέτρους μεταφοράς η οποίες επιλέχθηκαν από τον διακομιστή.

Παράδειγμα:

```
C->S: SETUP rtsp://example.com/foo/bar/baz.rm RTSP/1.0
      CSeq: 302
      Transport: RTP/AVP;unicast;client_port=4588-4589

S->C: RTSP/1.0 200 OK
      CSeq: 302
      Date: 23 Jan 1997 15:35:06 GMT
      Session: 47112344
      Transport: RTP/AVP;unicast;
               client_port=4588-4589;server_port=6256-6257
```

1.4.5 Play Method

Η μέθοδος play, επισημαίνει στον διακομιστή να αρχίσει να αποστέλλει δεδομένα μέσω των μηχανισμών που διευκρινίστηκαν στο setup message. Ένας client δεν μπορεί να εκδώσει ένα play requests μέχρι κάποιο ισχύουν setup request αναγνωριστεί ως επιτυχημένο. Ένα αίτημα play τοποθετεί το normal play time στην αρχή της ροής δεδομένων προς τον χρήστη μέχρι το τέλος της ροής. Ο διακομιστής πρέπει αυτά τα αιτήματα play να τα βάζει σε μια σειρά και να εξυπηρετούνται με σειρά. Όταν κάποιο play request καταφτάσει στον διακομιστή, έχει προτεραιότητα να εξυπηρετηθεί αυτό που ήδη εξυπηρετείται, μετά αυτά που ήδη περιμένουν και μετά αυτό που μόλις κατέφθασε. Με αυτό τον μηχανισμό μπορούμε να βλέπουμε κάποιο video από την αρχή μέχρι το τέλος χωρίς κανένα πρόβλημα. Αλλιώς θα βλέπαμε από την αρχή του βίντεο μέχρι τα 10δευτερα, μετά από τα 30δευτερα μέχρι τα 35δευτερα. Μετά από το 10' -20' και θα βλέπαμε ένα βίντεο όπου θα επικρατούσε μια αταξία και δεν θα βγάζαμε νόημα.

Παράδειγμα:

```
C->S: PLAY rtsp://audio.example.com/audio RTSP/1.0

CSeq: 835

Session: 12345678
Range: npt=10-15

C->S: PLAY rtsp://audio.example.com/audio RTSP/1.0
```

1.4.6 Pause Method

Το αίτημα pause όπως αντιλαμβανόμαστε και από την φράση προκαλεί την διακοπή ροής δεδομένων περιστασιακά. Εάν αναπαράγουμε ένα αρχείο ήχου είναι σαν να βάζουμε το αρχείο σε κατάσταση mute. Εάν αναπαράγουμε κάποια ομάδα από αρχεία, τότε σταματάει η ροή δεδομένων για όλα τα ενεργά αρχεία. Κατά την κατάσταση όπου το αρχείο είναι σε pause mode, ο διακομιστής κρατάει τα resources που ήδη κατανάλωνε για το συγκεκριμένο session. Ωστόσο όταν ο χρόνος που είναι σε κατάσταση pause υπερβεί το χρόνο που καταγράφεται στο timeout parameter της συνόδου το οποίο καταγράφεται στο setup message τα resources απελευθερώνονται.

Παράδειγμα:

```
C->S: PAUSE rtsp://example.com/fizzle/foo RTSP/1.0
CSeq: 834
Session: 12345678

S->C: RTSP/1.0 200 OK
```

1.4.7 Teardown Method

Το αίτημα teardown, σταματάει την ροή δεδομένων του αρχείου, και απελευθερώνει από τον διακομιστή τα resource που χρησιμοποιούνται για την συγκεκριμένη σύνοδο.

Παράδειγμα:

```
C->S: TEARDOWN rtsp://example.com/fizzle/foo RTSP/1.0
CSeq: 892
Session: 12345678
S->C: RTSP/1.0 200 OK
CSeq: 892
```

1.4.8 Set_Parameter Method

Αυτή η μέθοδος αναθέτει τις παραμέτρους για την παρουσίαση κάποιου αρχείου. Ένα αίτημα μπορεί να έχει μόνο μια παράμετρο για να μπορεί ο client να αντιληφθεί γιατί ένα αίτημα απέτυχε. Εάν κάποιο αίτημα περιέχει αρκετές παραμέτρους, τότε ο διακομιστής για να μπορέσει να δράσει πάνω στο αίτημα πρέπει να μπορεί να ικανοποιήσει όλες τις παραμέτρους. Ο διακομιστής είναι αυτός που επιτρέπει εάν μπορούν οι παράμετροι να αλλάζουν ή να παραμένουν στην ίδια τιμή. Οι παράμετροι είναι διαχωρισμένοι ομοιόμορφα ούτως ώστε όταν έχουμε κάποιο error να μπορεί να γίνεται αντιληπτό πιο εύκολα, αλλά και για να μπορεί πιο εύκολα μια παράμετρος να αλλάζει.

Παράδειγμα:

```
C->S: SET_PARAMETER rtsp://example.com/fizzle/foo RTSP/1.0  
  
CSeq: 421  
  
Content-length: 20  
  
Content-type: text/parameters  
  
barparam: barstuff  
  
S->C: RTSP/1.0 451 Invalid Parameter
```

1.4.9 Redirect Method

Ένα αίτημα Redirect, πληροφορεί τον χρήστη ότι πρέπει να ενωθεί σε κάποιο άλλο διακομιστή. Περιέχει το εναλλακτικό header location, που θα απευθυνθεί ο client. Μπορεί επίσης να περιέχει την παράμετρο range, που αποδεικνυε πότε αυτή η ανακατεύθυνση πρέπει να γίνει. Για να γίνει το redirection πρέπει ο client να εκδώσει ένα teardown αίτημα για να κλείσει το συγκεκριμένο session που του υπόδειξε το redirection, και μετά να εκδώσει είναι setup message για την νέα τοποθεσία του URI.

Παράδειγμα:

```
S->C: REDIRECT rtsp://example.com/fizzle/foo RTSP/1.0  
CSeq: 732  
Location: rtsp://bigserver.com:8001  
Range: clock=19960213T143205Z-
```

1.4.10 Record Method

Αυτή η μέθοδος αρχικοποιεί την καταγραφή κάποιας συγκεκριμένης διάρκειας από το αρχείο. Το πεδίο Timestamp που περιέχεται στο record αίτημα μας υποδειχνει το start και end time (UTC) που θα γίνει το recording. Εάν δεν δίνεται κάποιο UTC τότε χρησιμοποιείται ο χρόνος που δίνεται στο presentation, άρα αρχίζει η καταγραφή από τις αρχές το αρχείου. Εάν η σύνοδος αρχίσει την αναπαραγωγή τότε το recording αρχίζει από την συγκεκριμένη στιγμή που

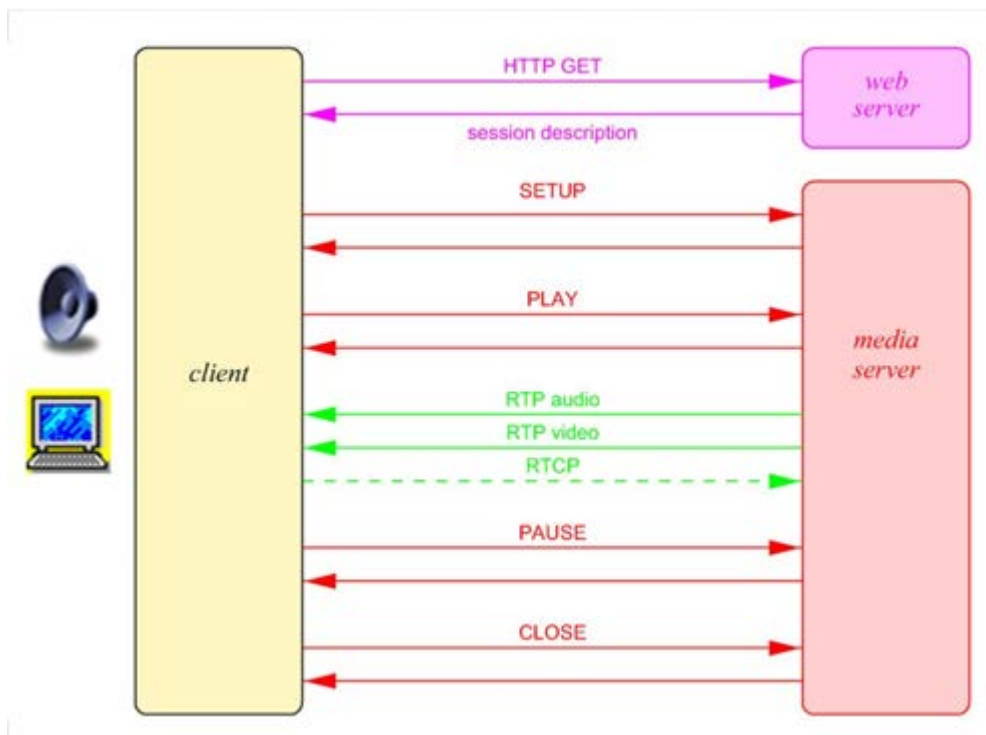
αποστέλλεται το αίτημα record . Ο διακομιστής αποφασίζει εάν θα αποθηκεύσει το συγκεκριμένο αρχείο που έγινε record κάτω από το URI που ζητήθηκε ή σε κάποιο άλλο URI. Η απάντηση του διακομιστή θα είναι το νούμερο 201 που σημαίνει ότι το record δημιουργήθηκε και θα περιέχει ένα header location που θα δίνει την νέα τοποθεσία του αρχείου, εάν υπάρχει οι διακομιστή πολυμέσων που υποστηρίζουν το παράγοντα του recording, για live παρουσιάσεις, πρέπει να υποστηρίζουν το clock range format γιατί το SMPTE format δεν κάνει αίσθηση.

Παράδειγμα:

```
C->S: RECORD rtsp://example.com/meeting/audio.en
954
Session: 12345678
Conference: 128.16.64.19/32492374
```

1.4.11 Σχεδιάγραμμα μίας συνόδου μεταξύ πελάτη και διακομιστή με την ανταλλαγή των πακέτων να γίνεται με βάση το RTSP.

Ένας διακομιστής εμπερικλείει το αρχείο περιγραφής της παρουσιάσεις σε ένα HTTP μήνυμα, και το αποστέλλει στον φυλλομετρητή. Όταν ο φυλλομετρητής παραλάβει το μήνυμα, επικαλείται κάποιο media player ο οποίος είναι ένα βοηθητικό πρόγραμμα για την αναπαραγωγή του αρχείου. Το αρχείο παρουσιάσεις περιέχει το ζητούμενο URL για το ανάλογο media αρχείο. Αμέσως μετά ο player, αποστέλλει ένα RTSP setup request, και ο διακομιστής απαντά με ένα RTSP OK μήνυμα. Ο player αποστέλλει ένα RTSP play request, ως υποθέσουμε για ένα αρχείο ήχου. Ο διακομιστής επομένως απαντά με ένα RTSP Ok. Σε αυτό το σημείο ο διακομιστής τοποθετεί το αρχείο στο δικό του εσωτερικό κανάλι σύνδεσης, για να μπορέσει να μεταδοθεί στον χρήστη. Ακολούθως ο χρήστης αποστέλλει στον διακομιστή ένα RTSP pause request, που είναι υπεύθυνο για να κάνει παύση την αναπαραγωγή του αρχείου. Ο διακομιστής του απαντάει με ένα RTSP Ok message. Όταν ο χρήστης τελειώνει με την αναπαραγωγή του αρχείου αποστέλλει στον διακομιστή ένα RTSP teardown request, και ο διακομιστής απάντα με ένα RTSP ok message, για να κλείνει η σύνοδος.



Πίνακας 1.2: Σχεδιάγραμμα μίας συνόδου μεταξύ πελάτη και διακομιστή με την ανταλλαγή των πακέτων να γίνεται με βάση το RTSP.

Κεφάλαιο 2

Ευπάθειες του Πρωτοκόλλου Real Time Streaming Protocol

Σε αυτό το κεφάλαιο έχουμε καταγράψει όλες τις πιθανές ευπάθειες του πρωτοκόλλου. Επίσης για κάθε ευπάθεια του πρωτοκόλλου που εντοπίσαμε αναγράφουμε και πιθανές λύσεις για αντιμετώπιση των επιθέσεων.

2.1 Γενική περιγραφή

Οι ευπάθειες του πρωτοκόλλου που εντοπίσαμε χωρίζονται σε 3 κατηγορίες.

- Επιθέσεις Υπερχείλισης
- Full-Connections Attacks
- Mallformed Requist Attack

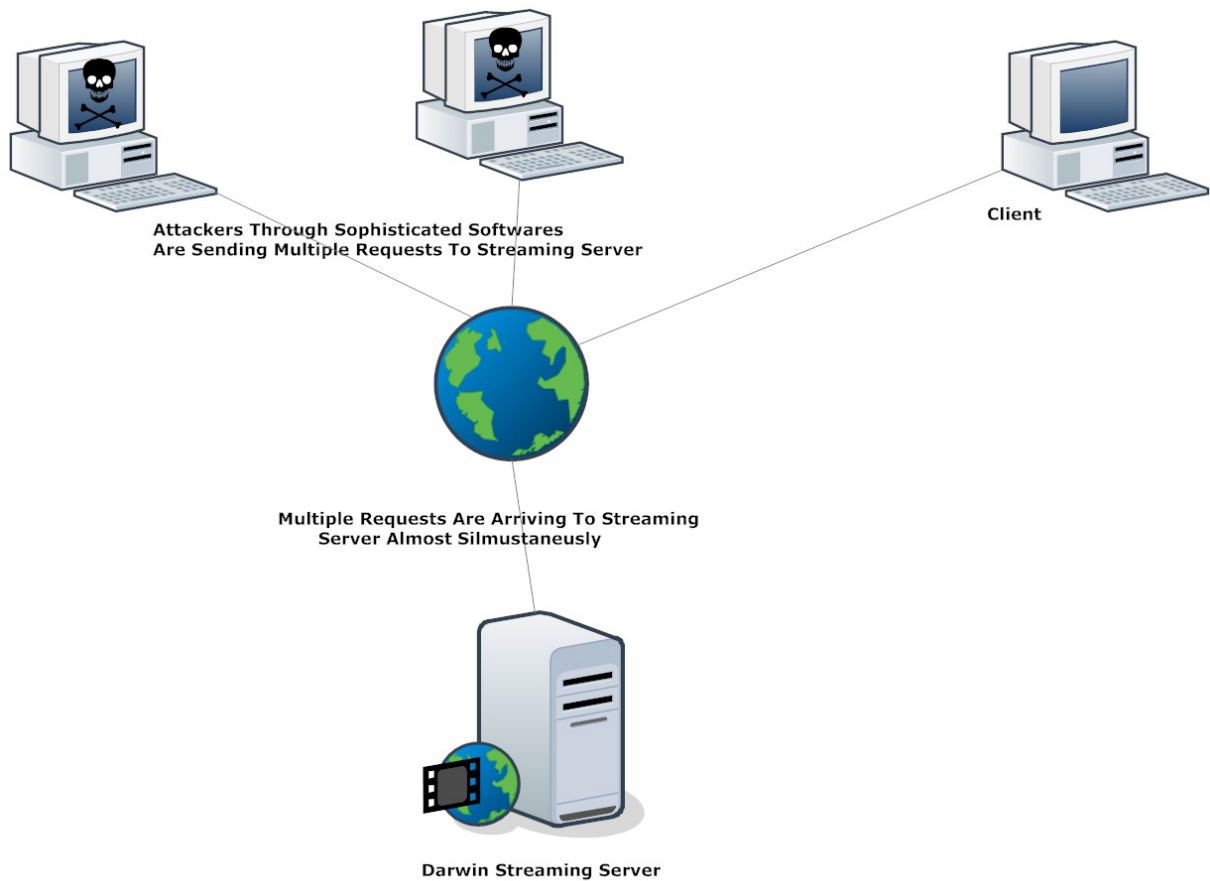
Πιο κάτω αναλύονται όλες οι επιθέσεις και σε ποιες κατηγορίες ανήκουν.

2.2 Επιθέσεις τύπου Υπερχείλισης

Επιθέσεις υπερχειλίσης ονομάζονται οι επιθέσεις όπου ο streaming server κατακλύζεται από πακέτα αιτημάτων ούτως ώστε να προκληθεί συμφόρηση στο socket του διακομιστή με αποτέλεσμα να ασχολείται ο διακομιστής με όλα τα αχρείαστα αιτήματα, και να παραβλέπει τα αιτήματα από τους κανονικούς χρήστες, και έτσι να επιτύχουμε DOS.

2.2.1 Describe Message Attack

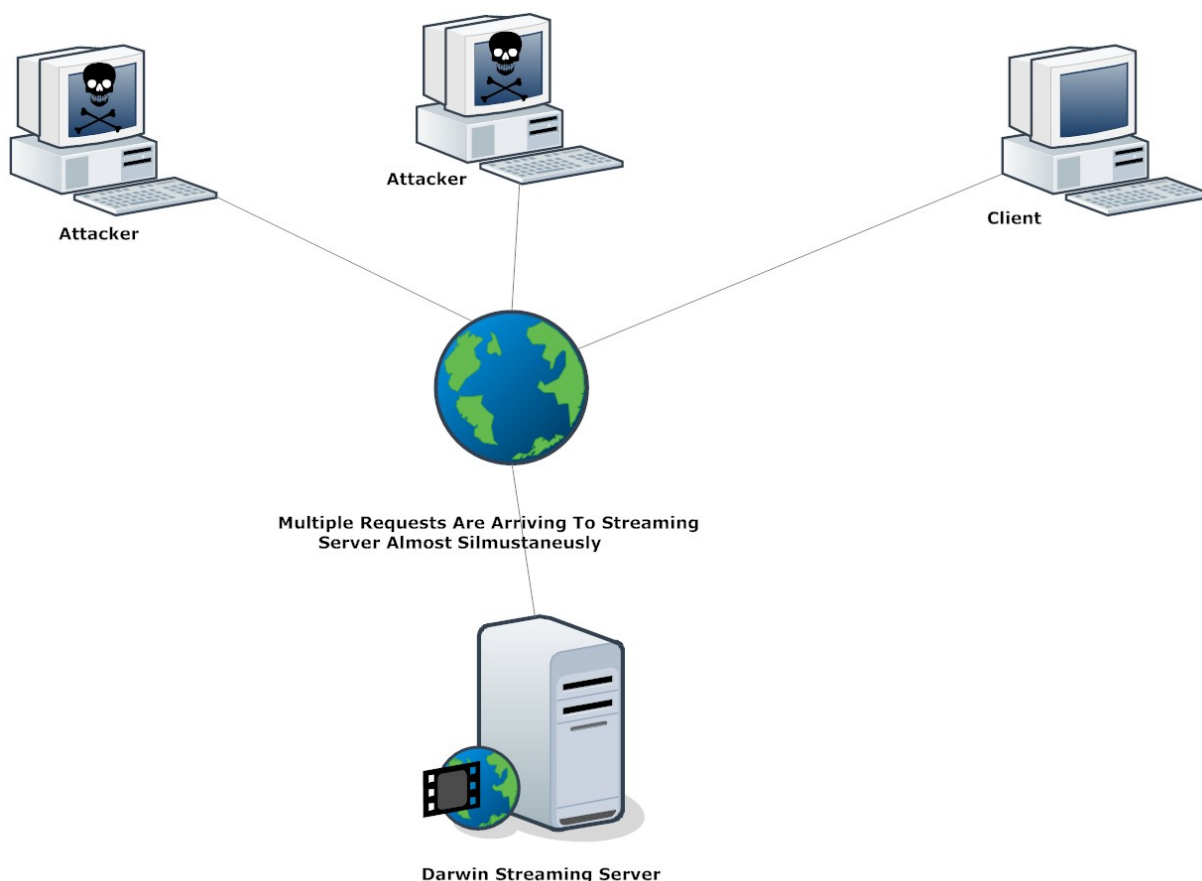
Ένας επιτιθέμενος μπορεί να επιτύχει μία επίθεση τύπου flooding εάν αρχίσει να αποστέλλει describe μηνύματα προς τον διακομιστή. Θα του στείλει τόσα πολλά μηνύματα που ο διακομιστής δεν θα μπορεί να τα διαχειριστεί όλα μαζί και έτσι θα προκληθεί άρνηση υπηρεσιών (Denial Of Service), όταν κάποιος ανεξάρτητος πελάτης ζητήσει κάποιο αρχείο από τον διακομιστή. Στα μηνύματα describe περιλαμβάνεται το url που ζητάει ο χρήστης και σε ποια μορφή δεδομένων μπορεί να τα διαχειριστεί.



Εικόνα 2.1: Σε αυτή την εικόνα βλέπουμε τρεις χρήστες οι οποίοι ζητούν κάποιο/α αρχεία από τον διακομιστή μας. Οι δύο από τους τρεις χρήστες μέσω κάποιων έξυπνων προγραμμάτων στέλνουν πολλαπλά αιτήματα προς τον διακομιστή θέλοντας να προκαλέσουν Denial Of Service στον διακομιστή.

2.2.2 Options Message Attack

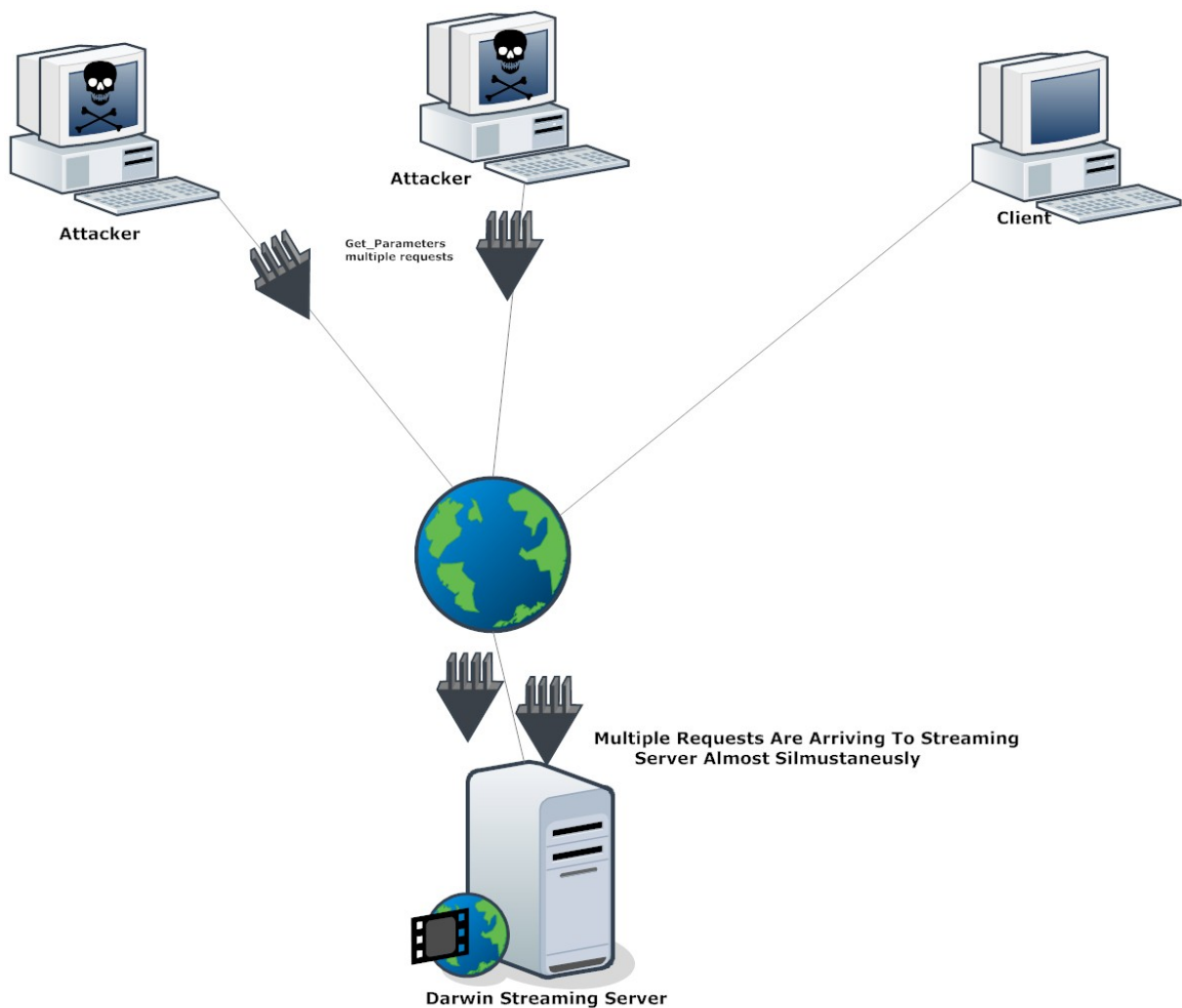
Ένας επιτιθέμενος μπορεί να επιτύχει μία επίθεση τύπου flooding εάν αρχίσει να αποστέλλει options μηνύματα προς τον διακομιστή. Αυτού του είδους τα μηνύματα αποστέλλονται από τον χρήστη στον διακομιστή για να μπορέσει να μάθει τι πόροι είναι ελεύθεροι από τον διακομιστή για να καταφέρει να γίνει η σύνδεση. Όταν ο διακομιστής κατακλιθεί από τέτοιου είδους μηνύματα δεν θα μπορεί να τα διαχειριστή όλα με αποτέλεσμα να προκληθεί άρνηση υπηρεσιών σε κάποιους χρήστες που θα στείλουν ένα Option μήνυμα.



Εικόνα 2.2: Σε αυτή την εικόνα βλέπουμε τους επιτιθέμενους να κατακλύζουν με options messages τον διακομιστή πολυμέσων, για να τον φέρουν σε τέτοια κατάσταση που να μην μπορεί να ανταποκριθεί σε άλλα μηνύματα option τα οποία έρχονται από κανονικούς χρήστες.

2.2.3 Get_Parameters Message Attack

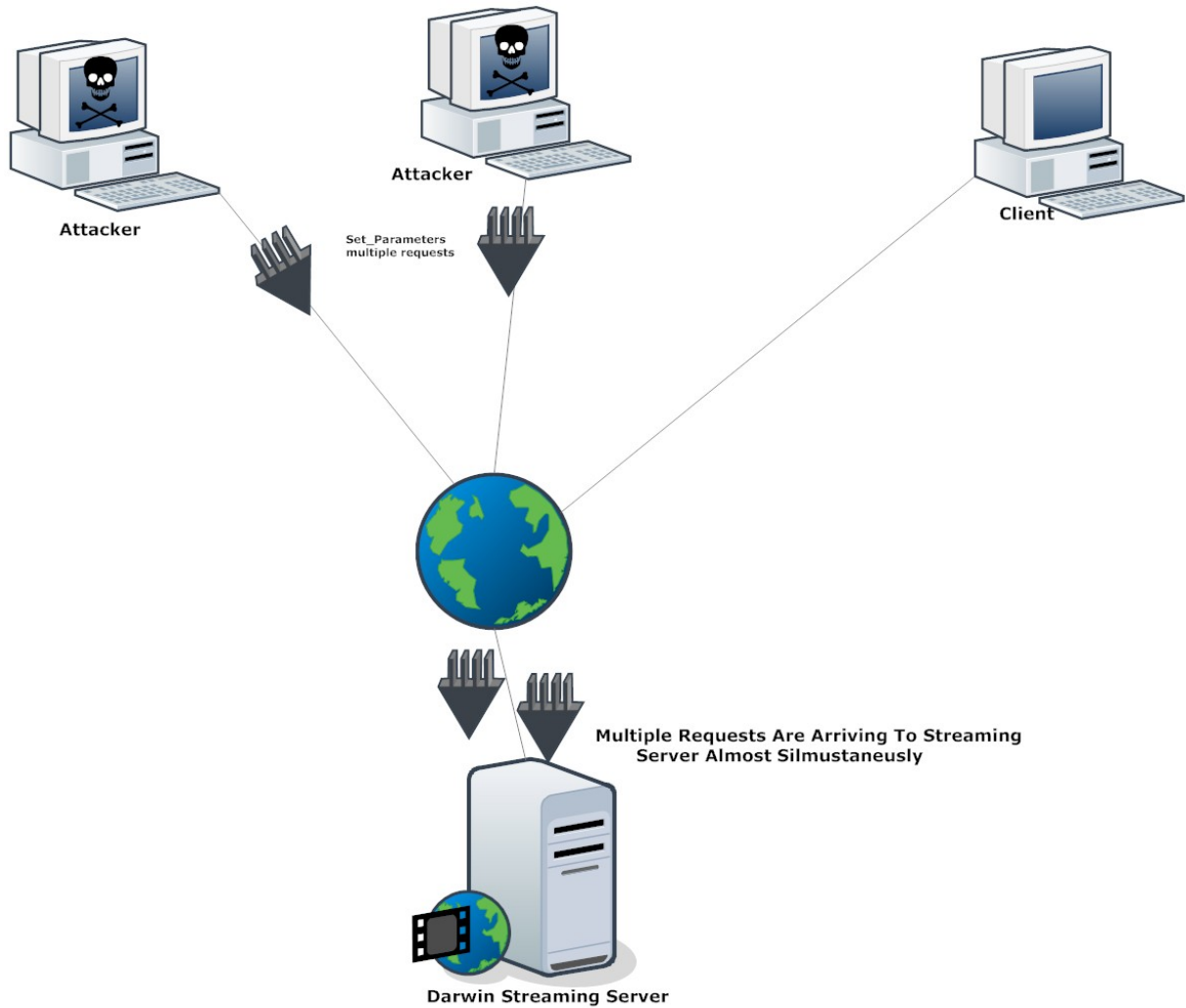
Ένας επιτιθέμενος μπορεί να επιτύχει μία επίθεση τύπου flooding εάν αρχίσει να αποστέλλει Get_Parameters μηνύματα προς τον διακομιστή. Αυτού του είδους τα μηνύματα αποστέλλονται προς τον διακομιστή ρωτώντας τον για κάποιο αρχείο. Μπορεί όμως να αποστέλλονται επίτηδες μηνύματα ζητώντας αρχεία τα οποία δεν υπάρχουν στον διακομιστή, με αποτέλεσμα αυτού του είδους τα μηνύματα να είναι πιο επιβλαβής αφού ο διακομιστής πρέπει να εξετάσει τις αιτήσεις χωρίς κανένα ουσιαστικό αποτέλεσμα.



Εικόνα 2.3: Σε αυτή την εικόνα βλέπουμε τους επιτιθέμενους να κατακλύζουν με Get_Parameters messages τον διακομιστή πολυμέσων. Τα μηνύματα που παραλαμβάνει κάνουν τον διακομιστή να σπαταλάει πόρους για να εξυπηρετήσει ανώφελα αιτήματα, αφού τα περισσότερα μπορεί να είναι και για αρχεία τα οποία δεν περιέχονται στον διακομιστή.

2.2.4 Set_Parameters Message Attack

Ένας επιτιθέμενος μπορεί να επιτύχει μία επίθεση τύπου flooding εάν αρχίσει να αποστέλλει Set_Parameters μηνύματα προς τον διακομιστή. Αυτού του είδους τα μηνύματα είναι υπεύθυνα για την αρχικοποίηση της τιμής που είναι υπεύθυνη για την παρουσίαση του αρχείου.



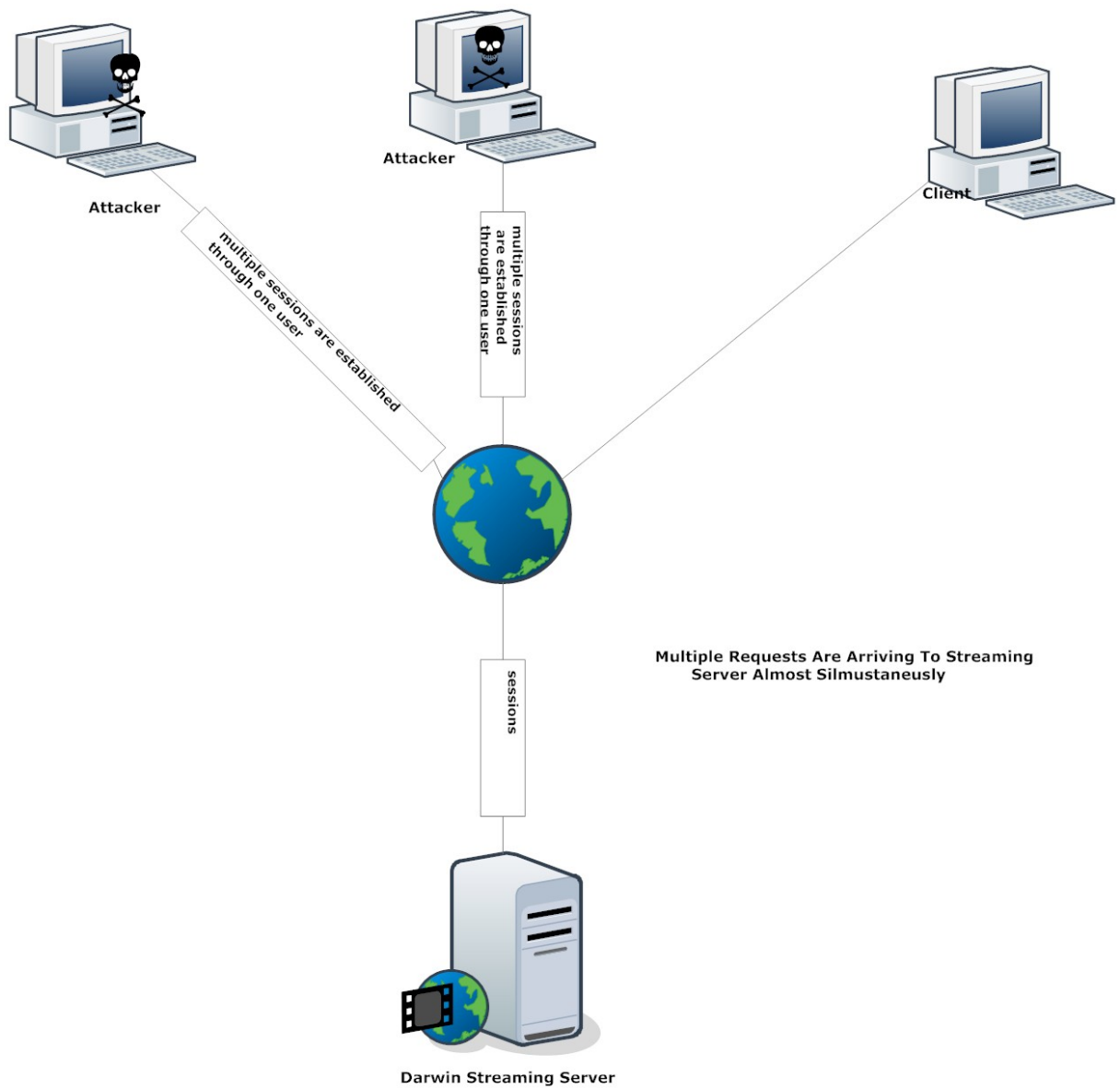
Εικόνα 2.4: Σε αυτή την εικόνα βλέπουμε τους επιτιθέμενους μέσω εξειδικευμένων προγραμμάτων να κατακλύζουν με Set_Parameters messages τον διακομιστή πολυμέσων.

2.3 Επιθέσεις τύπου Full Connections

Full-Connections attacks είναι οι επιθέσεις οι οποίες γίνονται μετά που η σύνοδος αρχίσει, με κάποια μηνύματα που αποστέλλονται από τον χρήστη στον διακομιστή να είναι επιβλαβής. Αυτού του είδους οι επιθέσεις προκαλούν στον διακομιστή να καταναλώνει τους επεξεργαστικούς πόρους με αποτέλεσμα να είναι πιο δύσκολο να εντοπιστούν και το αποτέλεσμα των επιθέσεων να είναι συνεχές. Για παράδειγμα κάποια επίθεση προκαλεί τον επεξεργαστή του διακομιστή να καταναλώνει το 100% της επεξεργαστικής του ικανότητας.

2.3.1 Set_Up Message Attack

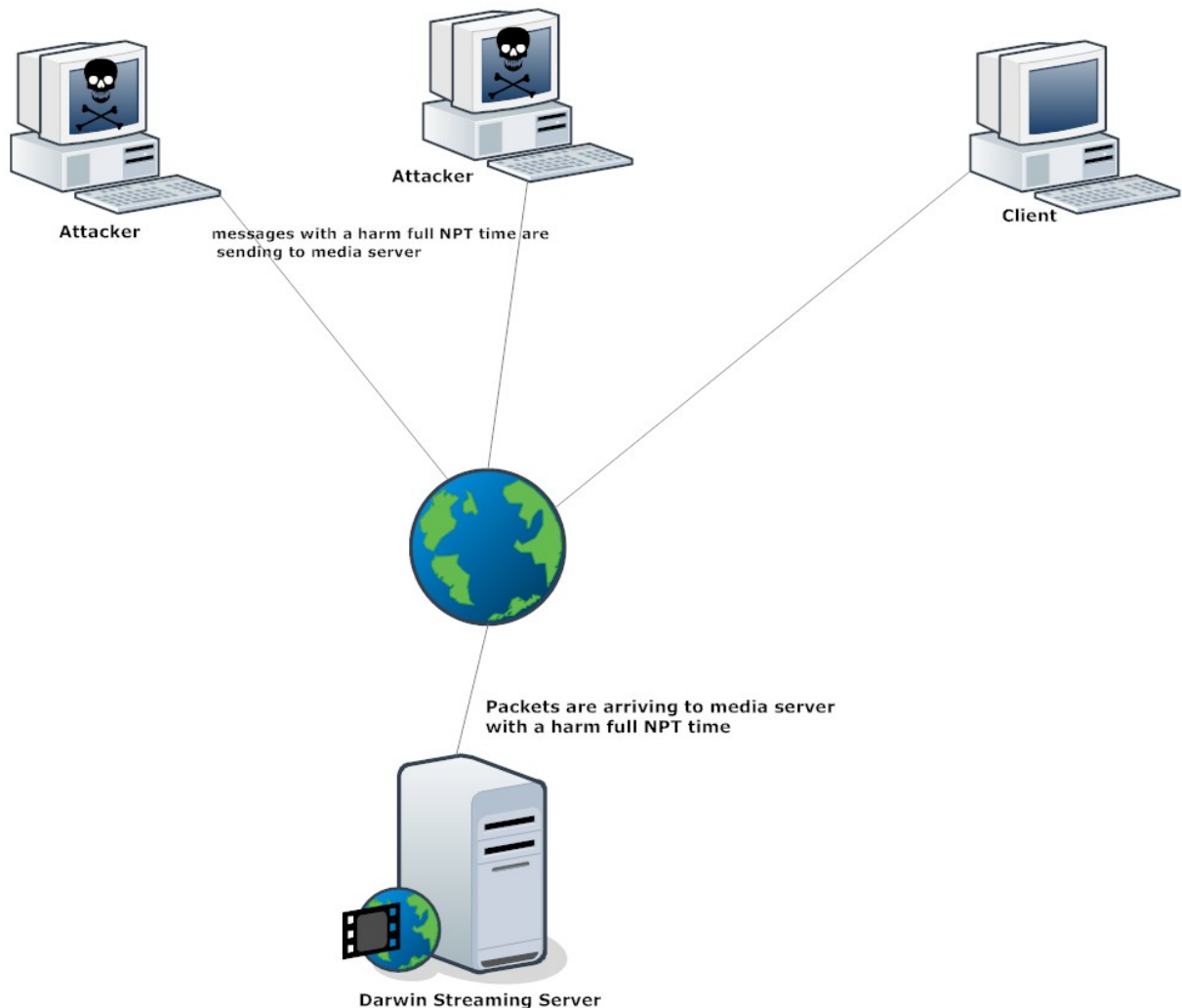
Ένας επιτιθέμενος μπορεί να επιτύχει μία επίθεση τύπου Full-Connections εάν αρχίσει να αποστέλλει προς τον διακομιστή SETUP μηνύματα. Τα μηνύματα αυτά είναι υπεύθυνα για την αρχικοποίηση της συνόδου. Ένας επιτιθέμενος μπορεί να το εκμεταλλευτεί αυτό και να εκδώσει πολλά μηνύματα SETUP για να αρχίσει πολλές συνόδους με τον διακομιστή, με αποτέλεσμα να σπαταλούνται οι επεξεργαστικοί πόροι του διακομιστή σε αχρείαστες συνόδους, και να μην μπορεί να εξυπηρετήσει τους κανονικούς χρήστες.



Εικόνα 2.5: Σε αυτή την εικόνα βλέπουμε τους επιτιθέμενους μέσω να κατακλύζουν το διακομιστή με Set_Up μηνύματα για να μπορέσουν να αρχικοποιήσουν πολλές ταυτόχρονες συνόδους.

2.3.2 Play Attack

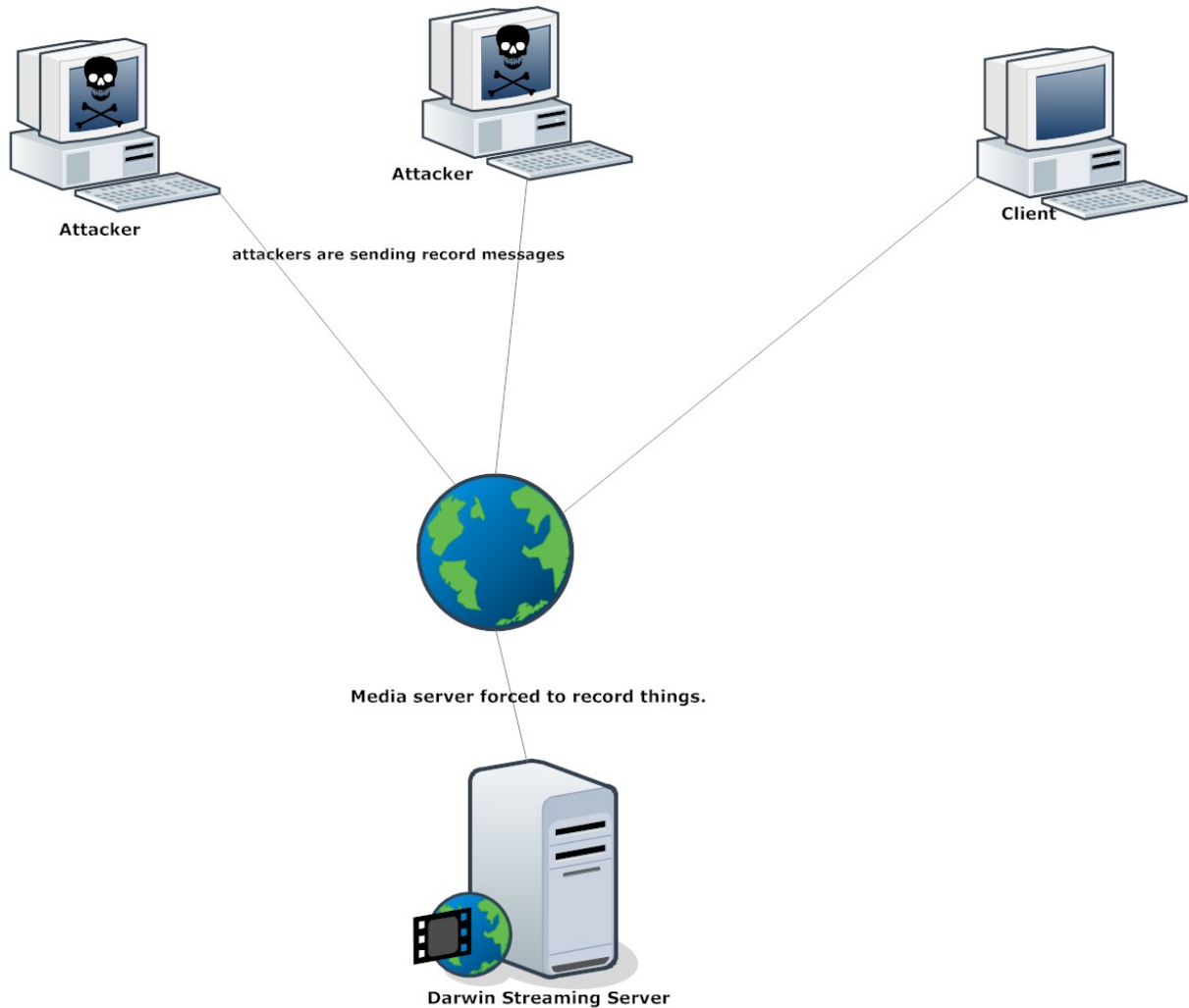
Ένας επιτιθέμενος μπορεί να επιτύχει μία επίθεση τύπου Full-Connections εάν αρχίσει να αποστέλλει προς τον διακομιστή PLAY μηνύματα με τυχαίο NPT time. Το normal play time είναι το πεδίο στο οποίο αναφέρεται πόση θα πρέπει να είναι η διάρκεια παρουσίασης του media. Ο επιτιθέμενος μεταβάλλει την τιμή του normal play time τυχαία. Αυτού του είδους η επίθεση σε ένα streaming server μπορεί να προκαλέσει σοβαρές επιπτώσεις. Μπορεί να καταναλώνει το εύρος ζώνης του διακομιστή, να αυξάνει disk read and write και να καταναλώνει την μνήμη του διακομιστή.



Εικόνα 2.6: Σε αυτή την εικόνα βλέπουμε τους επιτιθέμενους να αποστέλλουν στον διακομιστή μηνύματα play διαφοροποιώντας όμως το normal play time .

2.3.3 Record Attack

Ένας επιτιθέμενος μπορεί να επιτύχει μία επίθεση τύπου Full-Connections εάν αρχίσει να αποστέλλει προς τον διακομιστή μηνύματα Record. Τα μηνύματα αυτά είναι υπεύθυνα για εγγραφή κάποιου media. Αυτή η επίθεση προκαλεί στον διακομιστή επιβάρυνση του όλου συστήματος αφού σπαταλείται ώρα, επεξεργαστική πόροι αλλά και αποθηκευτική μνήμη για να εγγραφεί κάποιο media που είναι αχρείαστο.



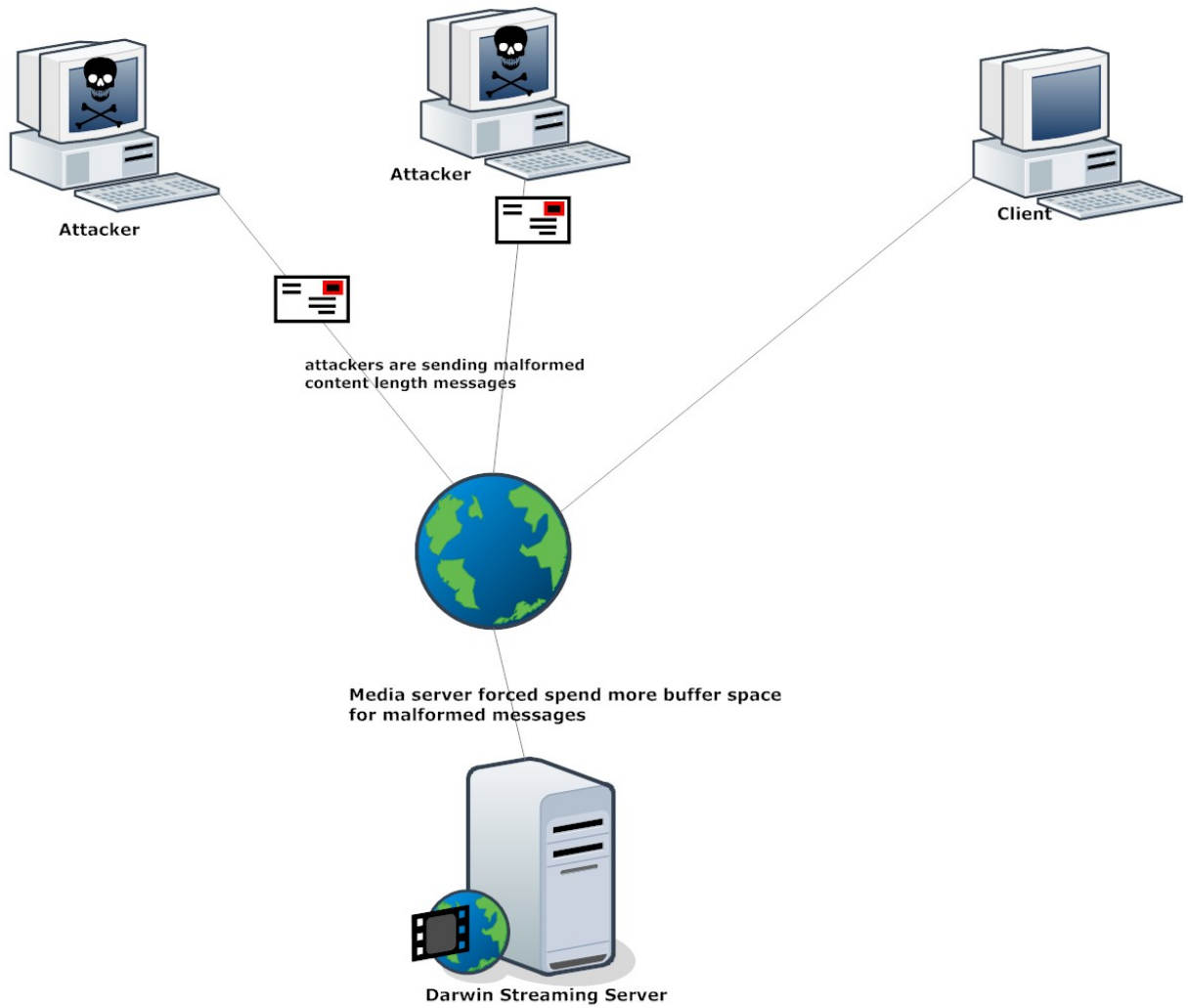
Εικόνα 2.7: Σε αυτή την εικόνα βλέπουμε τους επιτιθέμενους να αποστέλλουν στον διακομιστή μηνύματα record, για να τον αναγκάσουν να σπαταλήσει πόρους εγγράφοντας άσκοπα πράγματα.

2.4 Επιθέσεις τύπου Malformed Requests Attacks

Malformed request attack, είναι οι επιθέσεις οι οποίες εκμεταλλεύονται κάποιες ευπάθειες του πρωτοκόλλου. Σε αυτές του είδους τις επιθέσεις ο επιτιθέμενος επεμβαίνει στο συντακτικό των μηνυμάτων, το οποίο είναι προκαθορισμένο από την διεθνή επιτροπή IEEF, όσον αφορά το πρωτόκολλο RTSP. Ας υποθέσουμε ότι το header field του πρωτοκόλλου μπορεί να είναι μέχρι το 20 kb/ps και ο επιτιθέμενος επέμβει στο μήνυμα και από το kb το κάνει 30. Αυτό θα προκαλέσει συμφόρηση στον διακομιστή μας αφού θα περιμένει 20 kb και θα παραλάβει 30 θα χρειαστεί περισσότερη επεξεργαστική μνήμη αλλά και χρόνο για να διαχειριστή αυτό το αίτημα.

2.4.1 Content_Length Attack

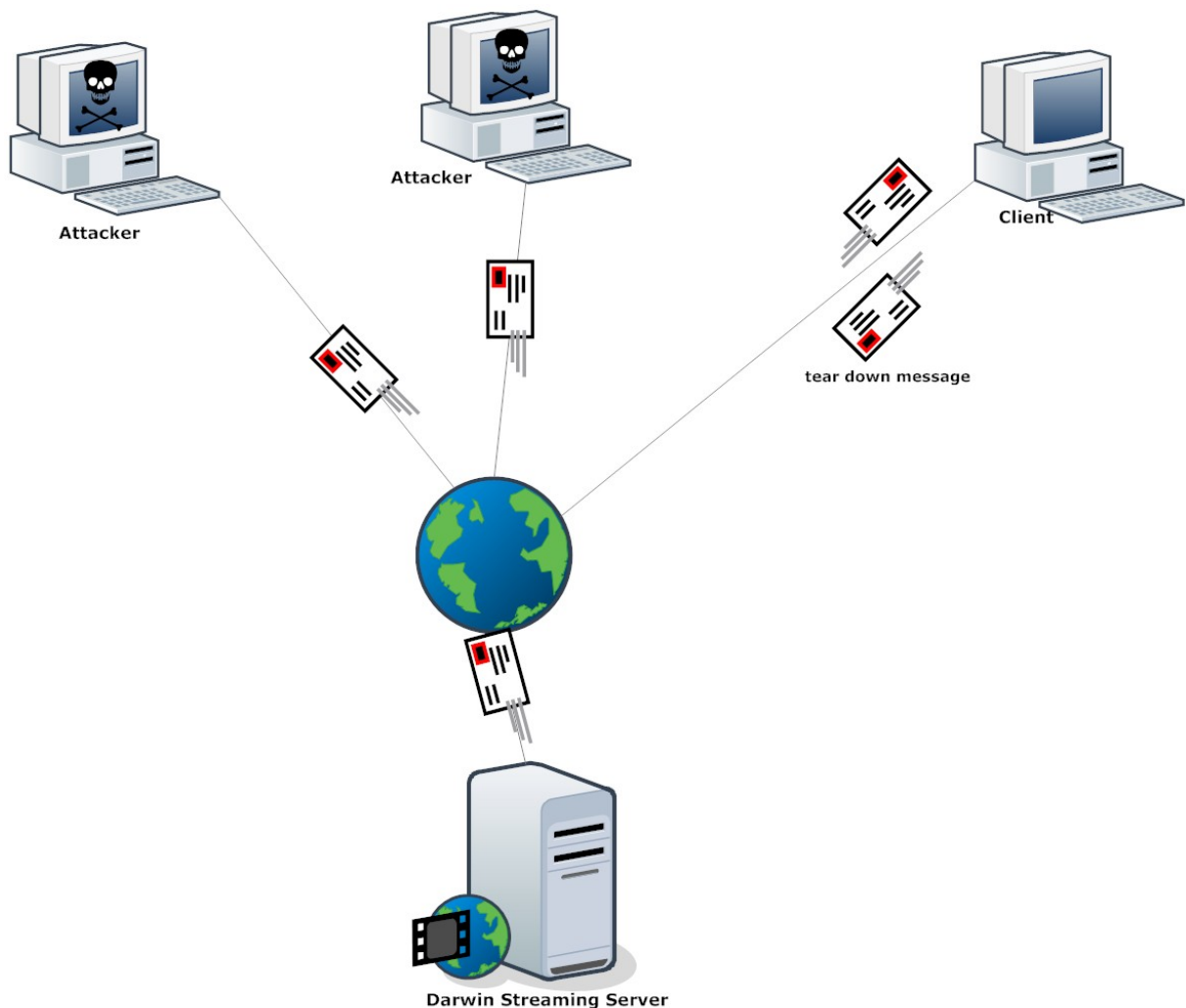
Ένας επιτιθέμενος μπορεί να επιτύχει μία επίθεση τύπου Malformed Request Attack εάν αρχίσει να μεταβάλλει το Content Length Header Field του αιτήματος. Το πεδίο αυτό είναι υπεύθυνο για να καθορίζει το μέγεθος του μηνύματος. Με την μεταβολή της τιμής του πεδίου ο διακομιστής αναγκάζεται να κρατήσει και να σπαταλήσει περισσότερη μνήμη για να κάνει buffer το μήνυμα από ότι θα έπρεπε. Εάν ο επιτιθέμενος δεν τερματίσει το message body θα προκαλέσει τον διακομιστή να κρατάει το μήνυμα που έγινε buffer στην μνήμη, με αποτέλεσμα πολλή επεξεργαστική μνήμη να σπαταλείται από τον διακομιστή.



Εικόνα 2.8: Σε αυτή την εικόνα βλέπουμε τους επιτιθέμενους να αποστέλλουν στον διακομιστή αλλαγμένα content length μηνύματα αναγκάζοντας τον διακομιστή να σπαταλήσει περισσότερο χώρο στο buffer του από το αναμενόμενο.

2.4.2 TearDown Attack

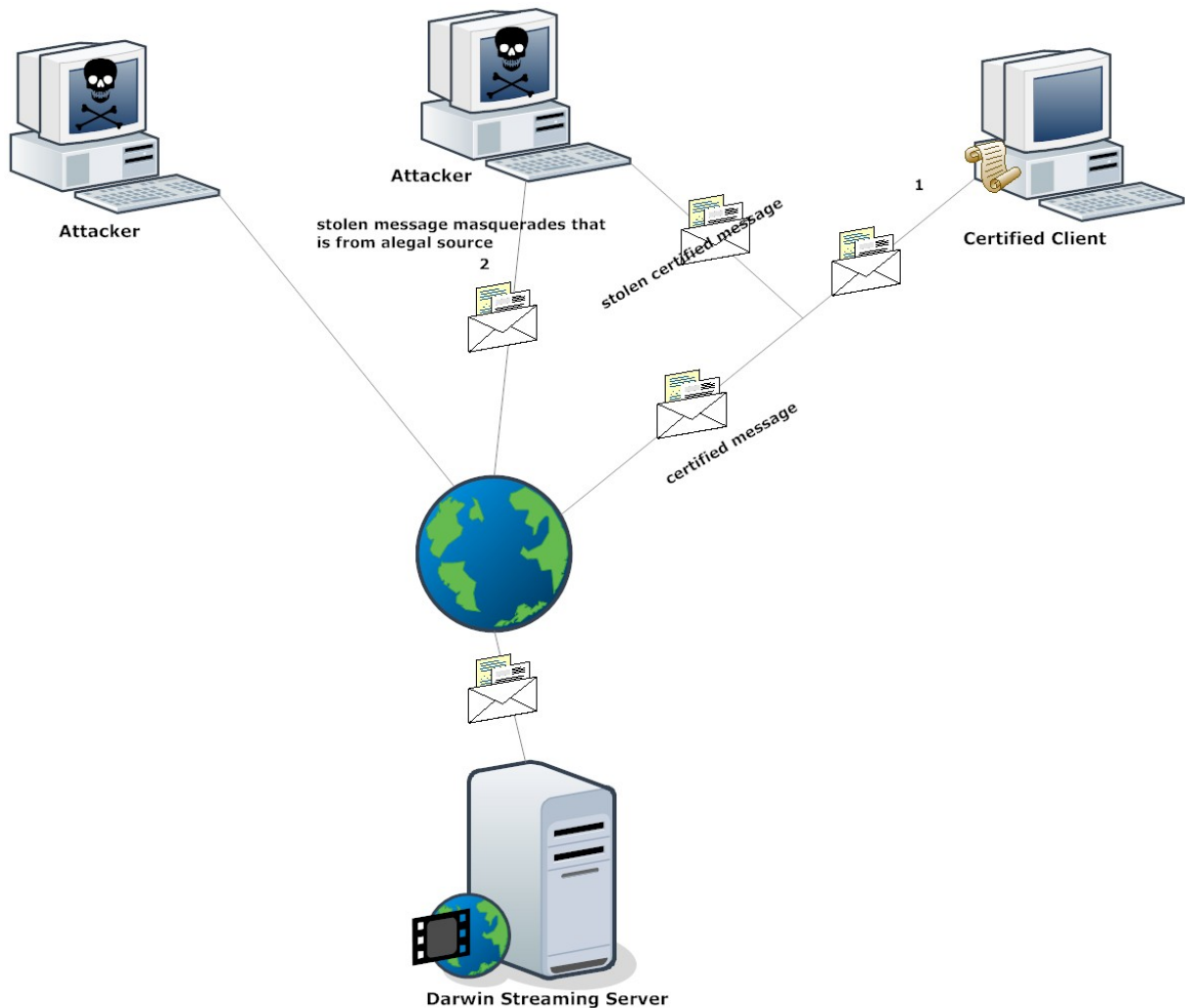
Ένας επιτιθέμενος μπορεί να επιτύχει μία επίθεση τύπου Teardown Attack εάν αρχίσει να αποστέλλει μηνύματα τύπου teardown στον διακομιστή για κάποιες συνόδους οι οποίες δεν είναι αυτός ο τελικός χρήστης. Το teardown μήνυμα τερματίζει την σύνοδο στην οποία απευθύνετε. Ακόμη μπορεί να αρχίσουν κάποια sessions και ο επιτιθέμενος να μην αποστείλει ποτέ το μήνυμα teardown με αποτέλεσμα ο διακομιστής να σπαταλάει επεξεργαστικούς πόρους σε συνόδους οι οποίες έπρεπε να έχουν ειδη κλείσει.



Εικόνα 2.9: Σε αυτή την εικόνα βλέπουμε τους επιτιθέμενους να μην αποστέλλουν πίσω teardown μήνυμα το οποίο είναι υπεύθυνο για τον τερματισμό της συνόδου και απελευθέρωσης των πόρων.

2.4.3 Hijack Attack

Ένας επιτιθέμενος μπορεί να επιτύχει μία επίθεση τύπου Hijack Attack εάν αρχίσει να αποστέλλει μηνύματα προς τον διακομιστή προσποιούμενος ότι το μήνυμα είναι από κάποιο έμπιστο χρήστη. Με αυτή την манούβρα ο διακομιστής δίνει στον χρήστη πρόσβαση στην διαχείριση των πόρων του διακομιστή.



Εικόνα 2.9: Σε αυτή την εικόνα βλέπουμε τον επιτιθέμενο να υποκλέπτει ένα certified μήνυμα από μία 'νόμιμη' πηγή και μετά να το αποστέλει στον διακομιστή ότι είναι και αυτός 'νόμιμη πηγή.'

Κεφάλαιο 3

Πιθανά Σενάρια Επιθέσεων

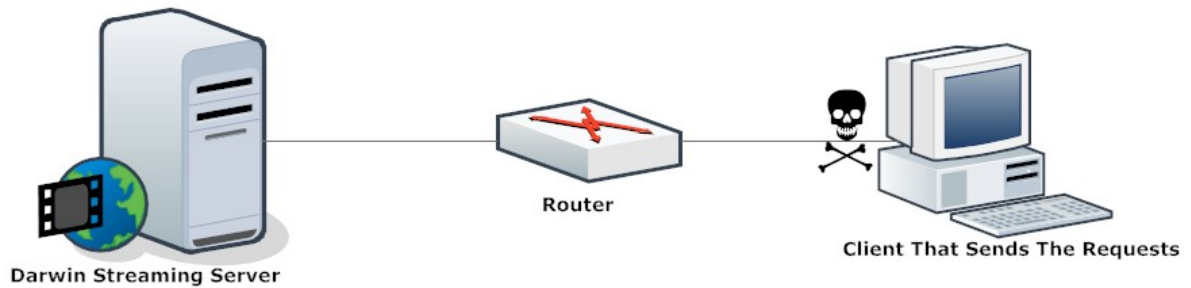
Τύπου Υπερχείλισης Στον Darwin

Streaming Server

Σε αυτό το κεφάλαιο θα καταγράψουμε πιθανά σενάρια τύπου Flooding τα οποία μπορούμε να πετύχουμε με κάποιες εντολές χρησιμοποιώντας το πρόγραμμα OpenRtsp.

- Χρησιμοποίηση Πρωτοκόλλου : RTSP,RTP
- Εργαλείο για εξομίωση επιθέσεων : OpenRTSP
- Default simultaneously connections on Darwin Stre. Server : 1000 requests

Σε ένα προσωπικό εργαστήριο, έχουμε δημιουργήσει ένα δικό μας ανεξάρτητο δίκτυο στο οποίο είναι εγκατεστημένος ο Streaming Server. Στο ίδιο δίκτυο είναι τοποθετημένος και ένας προσωπικός υπολογιστής πάνω στον οποίο είναι εγκατεστημένο το εργαλείο OpenRtsp αλλά και το κωδικοποιημένο πρόγραμμα σε γλώσσα java το οποίο απλά με ένα βρόχο καλεί το OpenRtsp αυτόματα. Οι δύο μηχανές επικοινωνούν μεταξύ τους με ένα Router.



Ο χρήστης ανάλογα με το σενάριο εξομοιώνει το πρόγραμμα χρησιμοποιώντας την κατάλληλη εντολή. Χρησιμοποιώντας το πρόγραμμα (java) απλά τοποθετεί την σωστή εντολή του OpenRtsp και αρχίζει τις επιθέσεις. Στον διακομηστή παράλληλα μόλις αρχίσει το πρόγραμμα να στέλνει τα requests τρέχουμε την εντολή `top -b -d 5 > file_name.txt` η οποία ανά πέντε δευτερόλεπτα καταγράφει σε ξεχωριστό αρχείο cpu usage και memory usage για κάθε διεργασία του διακομηστή. Για να αρχίσει μία καινούρια δοκιμή, συλλογής στατιστικών πρέπει να τελειώσει η προηγούμενη αλλά συνάμα ο διακομηστής να απελευθερωθεί από όλες τις συνόδους.

3.1 Εργαλείο OpenRtsp

Σε αυτή την ενότητα δίνετε μία γενική περιγραφή του εργαλείου OpenRtsp και των εντολών του.

3.1.1 Περιγραφή Εργαλείου OpenRtsp

Το OpenRtsp είναι ένα πρόγραμμα εντολών το οποίο μπορεί να χρησιμοποιηθεί για να ανοίγει, να μεταδίδει, να λαμβάνει και να καταγράφει κάποιο media αρχείο το οποίο καθορίζεται από κάποιο RTSP URL.

Ο πιο απλός τρόπος για να τρέξει το πρόγραμμα είναι δίνοντας του την εντολή `openRTSP <url>` όπου το URL είναι το αντίστοιχο url για το αρχείο το οποίο ζητάμε. Το πρόγραμμα θα ανοίξει το ζητούμενο αρχείο χρησιμοποιώντας το RTSP Describe message.

Εντολές που μπορούν να χρησιμοποιηθούν με την χρήση του OPEN_RTSP :

<u>Εντολή</u>	<u>Περιγραφή Εντολής</u>
-4	output a '.mp4'-format file (to 'stdout')
-a	play only the audio stream
-A <codec-number>	specify the static RTP payload format number of the audio codec to request from the server (" <i>playSIP</i> " only)
-b <buffer-size>	change the output file buffer size
-B <buffer-size>	change the input network socket buffer size

-c	play continuously
-M <MIME-subtype>	specify the MIME subtype of a dynamic RTP payload format for the audio codec to request from the server <i>("playSIP" only)</i>
-d <duration>	specify an explicit duration
-D <maximum-inter-packet-gap>	specify a maximum period of inactivity to wait before exiting
-f <frame-rate>	specify the video frame rate (used only with "-q", "-4", or "-i")
-F <fileName-prefix>	specify a prefix for each output file name
-h <height>	specify the video image height (used only with "-q", "-4", or "-i")
-H	output a QuickTime 'hint track' for each audio/video track (used only with "-q" or "-4")
-i	output a '.avi'-format file (to 'stdout')
-l	try to compensate for packet losses (used only with "-q", "-4", or "-i")
-m	output each incoming frame into a separate file

-n	be notified when RTP data packets start arriving
-o	request the server's command options, without sending "DESCRIBE" (<i>"openRTSP" only</i>)
-O	don't request the server's command options; just send "DESCRIBE" (<i>"openRTSP" only</i>)
-p <startingPortNumber>	specify the client port number(s)
-Q	output 'QOS' statistics about the data stream (when the program exits)
-q	output a QuickTime '.mov'-format file (to 'stdout')
-r	play the RTP streams, but don't receive them ourself
-s <initial-seek-time>	request that the server seek to the specified time (in seconds) before streaming
-S <byte-offset>	assume a simple RTP payload format (skipping over a special header of the specified size)
-t	stream RTP/RTCP data over TCP, rather than (the usual) UDP. (<i>"openRTSP" only</i>)
-T <http-port-number>	like "-t", except using RTSP-over-HTTP tunneling.

	<i>("openRTSP" only)</i>
<code>-u <username> <password></code>	specify a user name and password for digest authentication
<code>-V</code>	print less verbose diagnostic output
<code>-v</code>	play only the video stream
<code>-w <width></code>	specify the video image width (used only with "-q", "-4", or "-i")
<code>-y</code>	try to synchronize the audio and video tracks (used only with "-q" or "-4")
<code>-3</code>	output a '.mp4'-format file (to 'stdout')

Πίνακας 3.1: Πιο πάνω ο πίνακας με όλες τις εντολές του OpenRtsp και η διπλανή στήλη με την περιγραφή κάθε εντολής.

3.2 Σενάρια Επιθέσεων Χρησιμοποιώντας Την Απλή Εντολή Του OpenRtsp

Σε αυτή την ενότητα θα χρησιμοποιήσουμε την απλή εντολή του OpenRtsp η οποία εντολή είναι `Openrtsp rtsp://192.168.10.6/sample_300kbit.mov`. Για κάθε παραλλαγή του σεναρίου δηλαδή πεντακόσια, επτακόσια, οκτακόσια, και χίλια requests θα κάνουμε δεκαπέντε επαναλήψεις. Ακολούθως θα ενώσουμε όλα τα αρχεία όπου αφορούν την κάθε μία παραλλαγή του σεναρίου ξεχωριστά για να έχουμε μία ξεκάθαρη εικόνα των αποτελεσμάτων.

3.2.1 Περιγραφή Πειράματος Χρησιμοποιώντας Την Απλή Εντολή Του OpenRtsp Και Χρόνο Έκδοσης Των Αιτημάτων Μία Σταθερή Μεταβλητή

Με την χρήση του εργαλείου OpenRTSP θα εξομοιώσουμε επιθέσεις με πεντακόσια, επτακόσια, οκτακόσια και χίλια requests. Όλα τα requests θα ζητούν από τον διακομιστή το ίδιο αρχείο. Κάθε σύνοδος θα αρχίζει με καθυστέρηση (x) ml second. Όπου (x) να είναι μία σταθερή μεταβλητή. Στόχος του πειράματος είναι η δημιουργία DoS στον διακομιστή με απλά κανονικά αιτήματα. Θα θέλαμε να δούμε πώς μεταβάλλεται η χρήση του Cpu και του Memory στην μηχανή μας, από την διεργασία που χρησιμοποιεί ο DSS όσον αυξάνεται ο αριθμός των αιτημάτων, αφού η χρονική άφιξης των αιτημάτων θα είναι μία σταθερή μεταβλητή.

Open RTSP Command :

```
Openrtsp rtsp://192.168.10.6/sample_300kbit.mov
```

3.2.2 Περιγραφή Πειράματος Χρησιμοποιώντας Την Απλή Εντολή Του OpenRtsp Και Χρόνο Εκδοσης Των Αιτημάτων Μία Τυχαία Μεταβλητή

Με την χρήση του εργαλείου OpenRTSP θα εξομοιώσουμε επιθέσεις με πεντακόσια, επτακόσια, οκτακόσια, χίλια, χίλια εκατό requests. Όλα τα requests θα ζητούν από τον διακομιστή το ίδιο αρχείο. Κάθε session θα αρχίζει με καθυστέρηση (x) ml second. Όπου (x) να είναι μία τυχαία μεταβλητή. Στόχος του πειράματος είναι η δημιουργία DoS στον διακομιστή με απλά κανονικά αιτήματα. Θα θέλαμε να δούμε πώς μεταβάλλεται η χρήση του Cpu και του Memory στην μηχανή μας, από την διεργασία που χρησιμοποιεί ο DSS όσον αυξάνεται ο αριθμός των αιτημάτων. Η χρονική άφιξης των αιτημάτων θα είναι μία τυχαία μεταβλητή άρα μπορεί να αργήσει να σταλεί προς τον διακομιστή και έτσι ο διακομιστής να καταφέρει να κλείσει κάποια σύνοδο πριν να δεχτεί κάποια άλλη. Ασφαλώς θα μπορούσε να γίνει το αντίθετο και τα αιτήματα να καταφθάνουν με τέτοια χρονική απόκλιση με αποτέλεσμα διακομιστής να μην μπορεί να απελευθερώσει κάποιους πόρους για να εξυπηρετήσει όλα τα αιτήματα.

Open RTSP Command:

```
Openrtsp rtsp://192.168.10.6/sample_300kbit.mov
```

3.3 Σενάρια Επιθέσεων Χρησιμοποιώντας Την Απλή Εντολή Του OpenRtsp Ζητώντας Διαφορετικά Αρχεία Από Τον Streaming Server

Σε αυτή την ενότητα θα χρησιμοποιήσουμε την απλή εντολή του OpenRtsp η οποία εντολή είναι `Openrtsp rtsp://192.168.10.6/sample_300kbit.mov`. Η διαφορά όμως τώρα είναι ότι στις παραλλαγές του σεναρίου ζητούνται διαφορετικά αρχεία, όπως `anι,mov`. Για παράδειγμα στα πεντακόσια requests τα εκατό θα είναι για `anι`, τα άλλα εκατό για `mpeg4` και ούτω καθεξής. Για κάθε παραλλαγή του σεναρίου δηλαδή πεντακόσια, επτακόσια, οκτακόσια, χίλια θα κάνουμε δεκαπέντε επαναλήψεις. Ακολούθως θα ενώσουμε όλα τα αρχεία όπου αφορούν την κάθε μία παραλλαγή του σεναρίου ξεχωριστά για να έχουμε μία ξεκάθαρη εικόνα των αποτελεσμάτων.

3.3.1 Περιγραφή Πειράματος Χρησιμοποιώντας Την Απλή Εντολή Του OpenRtsp Ζητώντας Διαφορετικά Αρχεία Από Τον Streaming Server Και Χρόνο Εκδοσης Των Αιτημάτων Μία Σταθερή Μεταβλητή

Με την χρήση του εργαλείου OpenRTSP θα εξομοιώσουμε μία επίθεση με πεντακόσια, επτακόσια, οκτακόσια και χίλια requests. Όλα τα requests θα ζητούν από τον διακομιστή 4-5 διαφορετικά αρχεία. Τα αρχεία που θα ζητούνται από τον διακομιστή θα είναι και διαφορετικής μορφής, όπως `mpeg4`, `anι,mov`. Κάθε session θα αρχίζει με καθυστέρηση (x) ml second. Όπου (x) να είναι μία σταθερή μεταβλητή. Στόχος του πειράματος είναι η δημιουργία DoS στον διακομιστή με απλά κανονικά αιτήματα. Θα θέλαμε να δούμε πώς μεταβάλλεται η χρήση του Cpu και του Memory στην μηχανή μας, από την διεργασία που χρησιμοποιεί ο DSS όσον αυξάνεται ο αριθμός των αιτημάτων και η διαφορετικότητα των αιτημάτων, αφού η χρονική άφιξη των αιτημάτων θα είναι μία σταθερή μεταβλητή.

Open RTSP Command :

```
Openrtsp rtsp://192.168.10.6/sample_300kbit.mov
```

3.3.2 Περιγραφή Πειράματος Χρησιμοποιώντας Την Απλή Εντολή Του OpenRtsp Ζητώντας Διαφορετικά Αρχεία Από Τον Streaming Server Και Χρόνο Εκδοσης Των Αιτημάτων Μία Τυχαία Μεταβλητή

Με την χρήση του εργαλείου OpenRTSP θα εξομοιώσουμε μία επίθεση με πεντακόσια, επτακόσια, οκτακόσια, χίλια, χίλια εκατό requests. Όλα τα requests θα ζητούν από τον διακομιστή 4-5 διαφορετικά αρχεία. Τα αρχεία που θα ζητούνται από τον διακομιστή θα είναι και διαφορετικής μορφής και διάρκειας, όπως mpeg4, avi, mov. Κάθε session θα αρχίζει με καθυστέρηση (x) ml second. Όπου (x) να είναι μία random μεταβλητή. Στόχος του πειράματος είναι η δημιουργία DoS στον διακομιστή με απλά κανονικά αιτήματα. Θα θέλαμε να δούμε πώς μεταβάλλεται η χρήση του Cpu και του Memory στην μηχανή μας, από την διεργασία που χρησιμοποιεί ο DSS όσον αυξάνεται ο αριθμός των αιτημάτων και η διαφορετικότητα των αιτημάτων. Η χρονική άφιξης των αιτημάτων θα είναι μία τυχαία μεταβλητή άρα μπορεί να αργήσει να σταλεί προς τον διακομιστή και έτσι ο διακομιστής να καταφέρει να κλείσει κάποια σύνοδο πριν να δεχτεί κάποια άλλη. Ασφαλώς θα μπορούσε να γίνει το αντίθετο και τα αιτήματα να καταφθάνουν με τέτοια χρονική απόκλιση με αποτέλεσμα διακομιστής να μην μπορεί να απελευθερώσει κάποιους πόρους για να εξυπηρετήσει όλα τα αιτήματα.

Open RTSP Command :

```
Openrtsp rtsp://192.168.10.6/sample_300kbit.mov
```

3.4 Σενάρια Επιθέσεων Χρησιμοποιώντας Την Απλή Εντολή Του OpenRtsp Ζητώντας Κάποιο Αρχείο Που Δεν Βρίσκεται Στον Streaming Server

Σε αυτή την ενότητα θα χρησιμοποιήσουμε την απλή εντολή του OpenRtsp η οποία εντολή είναι `Openrtsp rtsp://192.168.10.6/sample_ThatIsNotIncluded_300kbit.mov`. Το αρχείο που ζητείτε από τον διακομιστή δεν υπάρχει πάνω στον διακομιστή. Για κάθε παραλλαγή του σεναρίου δηλαδή πεντακόσια, επτακόσια, οκτακόσια, χίλια requests θα κάνουμε δεκαπέντε επαναλήψεις. Ακολούθως θα ενώσουμε όλα τα αρχεία όπου αφορούν την κάθε μία παραλλαγή του σεναρίου ξεχωριστά για να έχουμε μία ξεκάθαρη εικόνα των αποτελεσμάτων.

3.4.1 Περιγραφή Πειράματος Χρησιμοποιώντας Την Απλή Εντολή Του OpenRtsp Ζητώντας Κάποιο Αρχείο Που Δεν Βρίσκεται Στον Streaming Server Και Χρόνο Εκδοσης Των Αιτημάτων Μία Σταθερή Μεταβλητή

Με την χρήση του εργαλείου OpenRTSP θα εξομοιώσουμε επιθέσεις με πεντακόσια, επτακόσια, οκτακόσια, χίλια, χίλια εκατό requests. Όλα τα requests θα ζητούν από τον διακομιστή το ίδιο αρχείο, το οποίο αρχείο όμως δεν θα υπάρχει στον διακομιστή. Κάθε session θα αρχίζει με καθυστέρηση (x) ml second. Όπου (x) να είναι μία σταθερή μεταβλητή. Στόχος του πειράματος είναι η δημιουργία DoS στον διακομιστή με απλά κανονικά αιτήματα. Θα θέλαμε να δούμε πώς μεταβάλλεται η χρήση του Cpu και του Memory στην μηχανή μας, από την διεργασία που χρησιμοποιεί ο DSS όσον αυξάνεται ο αριθμός των αιτημάτων σε αρχεία τα οποία δεν βρίσκονται στον διακομιστή. Ο διακομιστής θα πρέπει συνεχώς να σπαταλάει άσκοπα από τους πόρους του για κάποιο αρχείο το οποίο δεν μπορεί να προσφέρει. Κάνει πιο δύσκολο το έργο του διακομιστή αφού τα άσκοπα μηνύματα αποστέλλονται με σταθερή χρονική μεταβλητή.

Open RTSP Command :

```
Openrtsp rtsp://192.168.10.6/sample_ThatIsNotIncluded_300kbit.mov
```

3.4.2 Περιγραφή Πειράματος Χρησιμοποιώντας Την Απλή Εντολή Του OpenRtsp Ζητώντας Κάποιο Αρχείο Που Δεν Βρίσκεται Στον Streaming Server Και Χρόνο Εκδοσης Των Αιτημάτων Μία Τυχαία Μεταβλητή

Με την χρήση του εργαλείου OpenRTSP θα εξομοιώσουμε επιθέσεις με πεντακόσια, επτακόσια, οκτακόσια, χίλια, χίλια εκατό requests. Όλα τα requests θα ζητούν από τον διακομιστή το ίδιο αρχείο, το οποίο αρχείο όμως δεν θα υπάρχει στον διακομιστή. Κάθε session θα αρχίζει με καθυστέρηση (x) ml second. Όπου (x) να είναι μία random μεταβλητή. Στόχος του πειράματος είναι η δημιουργία DoS στον διακομιστή με απλά κανονικά αιτήματα. Θα θέλαμε να δούμε πώς μεταβάλλεται η χρήση του Cpu και του Memory στην μηχανή μας, από την διεργασία που χρησιμοποιεί ο DSS όσον αυξάνεται ο αριθμός των αιτημάτων σε αρχεία τα οποία δεν βρίσκονται στον διακομιστή, και πώς μεταβάλλεται σε σχέση με το προηγούμενο πείραμα αφού η χρονική άφιξης των αιτημάτων είναι μία τυχαία μεταβλητή. Ο διακομιστής θα πρέπει συνεχώς να σπαταλάει άσκοπα από τους πόρους του για κάποιο αρχείο το οποίο δεν μπορεί να προσφέρει.

Open RTSP Command :

3.5 Σενάρια Επιθέσεων Χρησιμοποιώντας Την Απλή Εντολή Του OpenRtsp Ζητώντας Αρχεία Που Δεν Βρίσκονται Στον Streaming Server

Σε αυτή την ενότητα θα χρησιμοποιήσουμε την απλή εντολή του OpenRtsp η οποία εντολή είναι `Openrtsp rtsp://192.168.10.6/sample_ThatIsNotIncluded_300kbit.mov`. Τα αρχεία που ζητούνται από τον διακομιστή δεν υπάρχουν πάνω στον διακομιστή. Για κάθε παραλλαγή του σεναρίου δηλαδή πεντακόσια, επτακόσια, οκτακόσια, χίλια requests θα κάνουμε δεκαπέντε επαναλήψεις. Ακολούθως θα ενώσουμε όλα τα αρχεία όπου αφορούν την κάθε μία παραλλαγή του σεναρίου ξεχωριστά για να έχουμε μία ξεκάθαρη εικόνα των αποτελεσμάτων.

3.5.1 Περιγραφή Πειράματος Χρησιμοποιώντας Την Απλή Εντολή Του OpenRtsp Ζητώντας Αρχεία Που Δεν Βρίσκονται Στον Streaming Server Και Χρόνο Εκδοσης Των Αιτημάτων Μία Σταθερή Μεταβλητή

Με την χρήση του εργαλείου OpenRTSP θα εξομοιώσουμε μία επίθεση με πεντακόσια, επτακόσια, οκτακόσια, χίλια, χίλια εκατό requests. Όλα τα requests θα ζητούν από τον διακομιστή 4-5 διαφορετικά αρχεία, τα οποία αρχεία όμως δεν θα υπάρχουν στον διακομιστή. Τα αρχεία που θα ζητούνται από τον διακομιστή θα είναι και διαφορετικής μορφής, όπως `mpeg4,mp3, avi,mov`. Κάθε session θα αρχίζει με καθυστέρηση (x) ml second. Όπου (x) να είναι μία σταθερή μεταβλητή. Στόχος του πειράματος είναι η δημιουργία DoS στον διακομιστή με απλά κανονικά αιτήματα. Θα θέλαμε να δούμε πώς μεταβάλλεται η χρήση του Cpu και του Memory στην μηχανή μας, από την διεργασία που χρησιμοποιεί ο DSS όσον αυξάνεται ο αριθμός των αιτημάτων σε αρχεία τα οποία δεν βρίσκονται στον διακομιστή. Ο διακομιστής θα πρέπει συνεχώς να σπαταλάει άσκοπα από τους πόρους του για κάποια αρχεία το οποία δεν μπορεί να προσφέρει. Κάνει πιο δύσκολο το έργο του διακομιστή αφού τα άσκοπα μηνύματα αποστέλλονται με σταθερή χρονική μεταβλητή και θα πρέπει την σχεδόν την ίδια χρονική στιγμή να ψάχνει για διαφορετικά αρχεία.

Open RTSP Command :

```
Openrtsp rtsp://192.168.10.6/sample_ThatIsNotIncluded_300kbit.mov
```

3.5.2 Περιγραφή Πειράματος Χρησιμοποιώντας Την Απλή Εντολή Του OpenRtsp Ζητώντας Αρχεία Που Δεν Βρίσκονται Στον Streaming Server Και Χρόνο Εκδοσης Των Αιτημάτων Μία Τυχαία Μεταβλητή

Με την χρήση του εργαλείου OpenRTSP θα εξομοιώσουμε επιθέσεις με πεντακόσια, επτακόσια, οκτακόσια, χίλια, χίλια εκατό requests. Όλα τα requests θα ζητούν από τον server 4-5 διαφορετικά αρχεία. Τα αρχεία που θα ζητούνται από τον server θα είναι και διαφορετικής μορφής, όπως mpeg4, mp3, avi, mov. Κάθε session θα αρχίζει με καθυστέρηση (x) ml second. Όπου (x) να είναι μία random μεταβλητή. Στόχος του πειράματος είναι η δημιουργία DoS στον διακομιστή με απλά κανονικά αιτήματα. Θα θέλαμε να δούμε πώς μεταβάλλεται η χρήση του Cpu και του Memory στην μηχανή μας, από την διεργασία που χρησιμοποιεί ο DSS όσον αυξάνεται ο αριθμός των αιτημάτων σε αρχεία τα οποία δεν βρίσκονται στον διακομιστή, και πώς μεταβάλλεται σε σχέση με το προηγούμενο πείραμα αφού η χρονική άφιξης των αιτημάτων είναι μία τυχαία μεταβλητή. Ο διακομιστής θα πρέπει συνεχώς να σπαταλάει άσκοπα από τους πόρους του για κάποια αρχεία τα οποία δεν μπορεί να προσφέρει.

Open RTSP Command :

```
Openrtsp rtsp://192.168.10.6/sample_ThatIsNotIncluded_300kbit.mov
```


3.6 Σενάρια Επιθέσεων Χρησιμοποιώντας Την Εντολή Του OpenRtsp Για Quality Of Service

Σε αυτή την ενότητα θα χρησιμοποιήσουμε την εντολή του OpenRtsp η οποία είναι υπεύθυνη να παρουσιάζει στατιστικά για την ποιότητα μετάδοσης δεδομένων. Η εντολή είναι `Openrtsp -Q rtsp://192.168.10.6/sample_300kbit.mov`. Για κάθε παραλλαγή του σεναρίου δηλαδή πεντακόσια, επτακόσια, οκτακόσια, χίλια requests, θα κάνουμε δεκαπέντε επαναλήψεις. Ακολούθως θα ενώσουμε όλα τα αρχεία όπου αφορούν την κάθε μία παραλλαγή του σεναρίου ξεχωριστά για να έχουμε μία ξεκάθαρη εικόνα των αποτελεσμάτων.

3.6.1 Περιγραφή Πειράματος Χρησιμοποιώντας Την Εντολή Του OpenRtsp Για Quality Of Service Και Χρόνο Εκδοσης Των Αιτημάτων Μία Σταθερή Μεταβλητή

Με την χρήση του εργαλείου OpenRTSP θα εξομοιώσουμε επιθέσεις με 500 πεντακόσια, επτακόσια, οκτακόσια, χίλια, χίλια εκατό requests. Όλα τα requests θα ζητούν από τον διακομιστή το ίδιο αρχείο. Στα request θα ζητούμε από τον διακομιστή και Quality Of Service. Κάθε session θα αρχίζει με καθυστέρηση (x) ml second. Όπου (x) να είναι μία σταθερή μεταβλητή. Στόχος του πειράματος είναι η δημιουργία DoS στον διακομιστή με αιτήματα τα οποία αποστέλλονται μέσω παραλλαγής του εργαλείου OpenRtsp, και είναι υπεύθυνο να ζητήσει από τον διακομιστή στατιστικά για την ποιότητα μετάδοσης δεδομένων. Θα θέλαμε να δούμε πώς μεταβάλλεται η χρήση του Cpu και του Memory στην μηχανή μας, από την διεργασία που χρησιμοποιεί ο DSS όσον αυξάνεται ο αριθμός των αιτημάτων σε αρχεία τα οποία βρίσκονται στον διακομιστή. Με αυτό τον τρόπο θα αναγκάσουμε τον διακομιστή μας να καταναλώσει αρκετούς από τους πόρους του αφού θα πρέπει να προσφέρει στον χρήστη και στατιστικά για την ποιότητα μετάδοσης του αρχείου. Και όλο αυτό με χρονική άφιξη των αιτημάτων μία σταθερή μεταβλητή.

Open RTSP Command :

```
Openrtsp -Q rtsp://192.168.10.6/sample_300kbit.mov
```

3.6.2 Περιγραφή Πειράματος Χρησιμοποιώντας Την Εντολή Του OpenRtsp Για Quality Of Service Και Χρόνο Εκδοσης Των Αιτημάτων Μία Τυχαία Μεταβλητή

Με την χρήση του εργαλείου OpenRTSP θα εξομοιώσουμε επιθέσεις με πεντακόσια, επτακόσια, οκτακόσια, χίλια, χίλια εκατό requests. Όλα τα requests θα ζητούν από τον διακομηστή το ίδιο αρχείο. Στα request θα ζητούμε από τον διακομηστή και Quality Of Service. Κάθε session θα αρχίζει με καθυστέρηση (x) ml second. Όπου (x) να είναι μία random μεταβλητή. Στόχος του πειράματος είναι η δημιουργία DoS στον διακομηστή με αιτήματα τα οποία αποστέλλονται μέσω παραλλαγής του εργαλείου OpenRtsp, και είναι υπεύθυνο να ζητήσει από τον διακομηστή στατιστικά για την ποιότητα μετάδοσης δεδομένων . Θα θέλαμε να δούμε πώς μεταβάλλεται η χρήση του Cpu και του Memory στην μηχανή μας, από την διεργασία που χρησιμοποιεί ο DSS όσον αυξάνεται ο αριθμός των αιτημάτων σε αρχεία τα οποία βρίσκονται στον διακομηστή, και πώς μεταβάλλεται σε σχέση με το προηγούμενο πείραμα αφού η χρονική άφιξης των αιτημάτων είναι μία τυχαία μεταβλητή. Με αυτό τον τρόπο θα αναγκάσουμε τον διακομηστή μας να καταναλώσει αρκετούς από τους πόρους του αφού θα πρέπει να προσφέρει στον χρήστη και στατιστικά για την ποιότητα μετάδοσης του αρχείου.

Open RTSP Command :

```
Openrtsp -Q rtsp://192.168.10.6/sample_300kbit.mov
```

3.7 Σενάρια Επιθέσεων Χρησιμοποιώντας Την Εντολή Του OpenRtsp Για Quality Of Service Ζητώντας Διαφορετικά Αρχεία

Σε αυτή την ενότητα θα χρησιμοποιήσουμε την απλή εντολή του OpenRtsp η οποία εντολή είναι `Openrtsp -Q rtsp://192.168.10.6/sample_300kbit.mov`. Η εντολή είναι υπεύθυνη να παρουσιάζει στατιστικά για την ποιότητα μετάδοσης δεδομένων. Η διαφορά όμως τώρα είναι ότι στις παραλλαγές του σεναρίου ζητούνται διαφορετικά αρχεία, όπως `aniv.mov`. Για παράδειγμα στα πεντακόσια requests τα εκατό θα είναι για `aniv`, τα άλλα εκατό για `mpeg4` και ούτω καθεξής. Για κάθε παραλλαγή του σεναρίου δηλαδή πεντακόσια, επτακόσια, οκτακόσια και χίλια requests θα κάνουμε δεκαπέντε επαναλήψεις. Ακολουθώντας θα ενώσουμε όλα τα αρχεία όπου αφορούν την κάθε μία παραλλαγή του σεναρίου ξεχωριστά για να έχουμε μία ξεκάθαρη εικόνα των αποτελεσμάτων.

3.7.1 Περιγραφή Πειράματος Χρησιμοποιώντας Την Εντολή Του OpenRtsp Για Quality Of Service Ζητώντας Διαφορετικά Αρχεία Και Χρόνο Εκδοσης Των Αιτημάτων Μία Σταθερή Μεταβλητή

Με την χρήση του εργαλείου OpenRTSP θα εξομοιώσουμε επιθέσεις με πεντακόσια, επτακόσια, οκτακόσια, χίλια, χίλια εκατό requests. Όλα τα requests θα ζητούν από τον διακομιστή 4-5 διαφορετικά αρχεία. Στα request θα ζητούμε από τον διακομιστή και Quality Of Statistics. Τα αρχεία που θα ζητούνται από τον διακομιστή θα είναι και διαφορετικής μορφής, όπως `mpeg4`, `mp3`, `aniv.mov`. Κάθε session θα αρχίζει με καθυστέρηση (x) ml second. Όπου (x) να είναι μία σταθερή μεταβλητή. Στόχος του πειράματος είναι η δημιουργία DoS στον διακομιστή μέσω παραλλαγής του εργαλείου OpenRtsp, και είναι υπεύθυνο να ζητήσει από τον διακομιστή στατιστικά για την ποιότητα μετάδοσης δεδομένων. Θα θέλαμε να δούμε πώς μεταβάλλεται η χρήση του Cpu και του Memory στην μηχανή μας, από την διεργασία που χρησιμοποιεί ο DSS όσον αυξάνεται ο αριθμός των αιτημάτων αφού η χρονική άφιξη των αιτημάτων θα είναι μία σταθερή μεταβλητή. Θα αναγκάσουμε τον διακομιστή να σπαταλάει πόρους ψάχνοντας για κάποια αρχεία τα οποία δεν μπορεί να προσφέρει.

3.7.2 Περιγραφή Πειράματος Χρησιμοποιώντας Την Εντολή Του OpenRtsp Για Quality Of Service Ζητώντας Διαφορετικά Αρχεία Και Χρόνο Εκδοσης Των Αιτημάτων Μία Τυχαία Μεταβλητή

Με την χρήση του εργαλείου OpenRTSP θα εξομοιώσουμε επιθέσεις με πεντακόσια, επτακόσια, οκτακόσια, χίλια, χίλια εκατό requests. Όλα τα requests θα ζητούν από τον διακομιστή 4-5 διαφορετικά αρχεία. Στα request θα ζητούμε από τον διακομιστή και Quality Of Statistics. Τα αρχεία που θα ζητούνται από τον διακομιστή θα είναι και διαφορετικής μορφής, όπως mpeg4, mp3, avi, mov. Κάθε session θα αρχίζει με καθυστέρηση (x) ml second. Όπου (x) να είναι μία random μεταβλητή. Στόχος του πειράματος είναι η δημιουργία DoS στον διακομιστή μέσω παραλλαγής του εργαλείου OpenRtsp, και είναι υπεύθυνο να ζητήσει από τον διακομιστή στατιστικά για την ποιότητα μετάδοσης δεδομένων. Θα θέλαμε να δούμε πώς μεταβάλλεται η χρήση του Cpu και του Memory στην μηχανή μας, από την διεργασία που χρησιμοποιεί ο DSS όσον αυξάνεται ο αριθμός των αιτημάτων. Η χρονική άφιξη των αιτημάτων θα είναι μία τυχαία μεταβλητή άρα μπορεί να αργήσει να σταλεί προς τον διακομιστή και έτσι ο διακομιστής να καταφέρει να κλείσει κάποια σύνοδο πριν να δεχτεί κάποια άλλη. Ασφαλώς θα μπορούσε να γίνει το αντίθετο και τα αιτήματα να καταφθάνουν με τέτοια χρονική απόκλιση με αποτέλεσμα διακομιστής να μην μπορεί να απελευθερώσει κάποιους πόρους για να εξυπηρετήσει όλα τα αιτήματα.

Open RTSP Command:

```
Openrtsp -Q rtsp://192.168.10.6/sample_300kbit.mov
```

3.8 Σενάρια Επιθέσεων Χρησιμοποιώντας Την Εντολή Του OpenRtsp Για Αλλαγή στην Διάρκεια Αναπαραγωγής Του Αρχείου

Σε αυτή την ενότητα θα χρησιμοποιήσουμε την εντολή του OpenRtsp η οποία είναι υπεύθυνη για την αλλαγή στην διάρκεια αναπαραγωγής του αρχείου. Η εντολή είναι

`Openrtsp -d -10 rtsp://192.168.10.6/sample_300kbit.mov`. με αυτό τον τρόπο στρεσάρουμε τον διακομιστή μας να καταναλώσει περισσότερους πόρους για να κρατήσει ανοικτές τις συνόδους για περισσότερο από το αναμενόμενο χρόνο. Για κάθε παραλλαγή του σεναρίου δηλαδή πεντακόσια, επτακόσια, οκτακόσια, και χίλια requests, θα κάνουμε δεκαπέντε επαναλήψεις. Ακολούθως θα ενώσουμε όλα τα αρχεία όπου αφορούν την κάθε μία παραλλαγή του σεναρίου ξεχωριστά για να έχουμε μία ξεκάθαρη εικόνα των αποτελεσμάτων.

3.8.1 Περιγραφή Πειράματος Χρησιμοποιώντας Την Εντολή Του OpenRtsp Για Αλλαγή στην Διάρκεια Αναπαραγωγής Του Αρχείου Και Χρόνο Εκδοσης Των Αιτημάτων Μία Σταθερή Μεταβλητή

Με την χρήση του εργαλείου OpenRTSP θα εξομοιώσουμε μία επίθεση με πεντακόσια, επτακόσια, οκτακόσια, χίλια, χίλια εκατό requests. Όλα τα requests θα ζητούν από τον διακομιστή το ίδιο αρχείο. Κάθε session θα αρχίζει με καθυστέρηση (x) ml second. Όπου (x) να είναι μία σταθερή μεταβλητή. Όλα τα session θα έχουν διαφορετικό duration time από το κανονικό duration που θα είχε το αρχείο. Η διαφορά στο duration θα είναι μία σταθερή μεταβλητή δέκα δευτερολέπτων. Στόχος του πειράματος είναι η δημιουργία DoS στον διακομιστή με αιτήματα τα οποία αποστέλλονται μεσω παραλλαγής του εργαλείου OpenRtsp, και είναι υπεύθυνο να κρατάει για περισσότερη ώρα ανοικτό η σύνοδος από την κανονική ώρα που ορίζει ο DSS . Θα θέλαμε να δούμε πώς μεταβάλλεται η χρήση του Cpu και του Memory στην μηχανή μας, από την διεργασία που χρησιμοποιεί ο DSS όσον αυξάνεται ο αριθμός των αιτημάτων σε αρχεία τα οποία βρίσκονται στον διακομιστή. Με αυτό τον τρόπο θα αναγκάσουμε τον διακομιστή μας να καταναλώσει αρκετούς από τους πόρους του αφού θα χρειάζεται να σπαταλάει πόρους σε συνόδους οι οποίοι θα έπρεπε να τερματιστούν. Και όλο αυτό με χρονική άφιξη των αιτημάτων μία σταθερή μεταβλητή.

Open RTSP Command:

Σχόλια : αύξηση στο duration time κατά 10 sec

```
Openrtsp -d -10 rtsp://192.168.10.6/sample_300kbit.mov
```

3.8.2 Περιγραφή Πειράματος Χρησιμοποιώντας Την Εντολή Του OpenRtsp Για Αλλαγή στην Διάρκεια Αναπαραγωγής Του Αρχείου Και Χρόνο Εκδοσης Των Αιτημάτων Μία Τυχαία Μεταβλητή

Με την χρήση του εργαλείου OpenRTSP θα εξομοιώσουμε επιθέσεις με πεντακόσια, επτακόσια, οκτακόσια, χίλια, χίλια εκατό requests. Όλα τα requests θα ζητούν από τον διακομιστή το ίδιο αρχείο. Κάθε session θα αρχίζει με καθυστέρηση (x) ml second. Όπου (x) να είναι μία σταθερή μεταβλητή. Όλα τα session θα έχουν διαφορετικό duration time από το κανονικό duration που θα είχε το αρχείο. Η διαφορά στο duration θα είναι μία random μεταβλητή. . Στόχος του πειράματος είναι η δημιουργία DoS στον διακομιστή με αιτήματα τα οποία αποστέλλονται μεσω παραλλαγής του εργαλείου OpenRtsp, και είναι υπεύθυνο να κρατάει για περισσότερη ώρα ανοικτό η σύνοδος από την κανονική ώρα που ορίζει ο DSS . Θα θέλαμε να δούμε πώς μεταβάλλεται η χρήση του Cpu και του Memory στην μηχανή μας, από την διεργασία που χρησιμοποιεί ο DSS όσον αυξάνεται ο αριθμός των αιτημάτων σε αρχεία τα οποία βρίσκονται στον διακομιστή, με χρονική άφιξη των αιτημάτων μία τυχαία μεταβλητή. Με αυτό τον τρόπο θα αναγκάσουμε τον διακομιστή μας να καταναλώσει αρκετούς από τους πόρους του αφού θα χρειάζεται να σπαταλάει πόρους σε συνόδους οι οποίοι θα έπρεπε να τερματιστούν.

Open RTSP Command :

Σχόλια : αύξηση στο duration time κατά 10 sec

```
Openrtsp -d -y rtsp://192.168.10.6/sample_300kbit.mov
```

3.9 Σενάρια Επιθέσεων Χρησιμοποιώντας Την Εντολή Του OpenRtsp Για Αλλαγή στην Διάρκεια Αναπαραγωγής Του Αρχείου Και Quality Of Service

Σε αυτή την ενότητα θα χρησιμοποιήσουμε την εντολή του OpenRtsp η οποία είναι υπεύθυνη για την αλλαγή στην διάρκεια αναπαραγωγής του αρχείου αλλά και την παρουσίαση στατιστικών της ποιότητας παράδοσης των δεδομένων. Η εντολή είναι

Openrtsp -Q -d -10 rtsp://192.168.10.6/sample_300kbit.mov. με αυτό τον τρόπο στρεσάρουμε τον διακομιστή μας να καταναλώσει περισσότερους πόρους για να κρατήσει ανοικτές τις συνόδους για περισσότερο από το αναμενόμενο χρόνο. Για κάθε παραλλαγή του σεναρίου δηλαδή πεντακόσια, επτακόσια, οκτακόσια, χίλια requests, θα κάνουμε δεκαπέντε επαναλήψεις. Ακολούθως θα ενώσουμε όλα τα αρχεία όπου αφορούν την κάθε μία παραλλαγή του σεναρίου ξεχωριστά για να έχουμε μία ξεκάθαρη εικόνα των αποτελεσμάτων.

3.9.1 Περιγραφή Πειράματος Χρησιμοποιώντας Την Εντολή Του OpenRtsp Για Αλλαγή στην Διάρκεια Αναπαραγωγής Του Αρχείου Και Quality Of Service Για Χρόνο Έκδοσης Των Αιτημάτων Μία Σταθερή Μεταβλητή

Με την χρήση του εργαλείου OpenRTSP θα εξομοιώσουμε επιθέσεις με πεντακόσια, επτακόσια, οκτακόσια, χίλια, χίλια εκατό requests. Όλα τα requests θα ζητούν από τον διακομιστή 4-5 διαφορετικά αρχεία. Στα request θα ζητούμε από τον διακομιστή και Quality Of Statistics. Τα αρχεία που θα ζητούνται από τον διακομιστή θα είναι και διαφορετικής μορφής, όπως mpeg4, mp3, avi, mov.. Κάθε session θα αρχίζει με καθυστέρηση (x) ml second. Όπου (x) να είναι μία σταθερή μεταβλητή. Όλα τα session θα έχουν διαφορετικό duration time από το κανονικό duration που θα είχε το αρχείο. Η διαφορά στο duration θα είναι μία σταθερή μεταβλητή. Στόχος του πειράματος είναι η δημιουργία DoS στον διακομιστή με αιτήματα τα οποία αποστέλλονται μέσω παραλλαγής του εργαλείου OpenRtsp, και είναι υπεύθυνο να κρατάει για περισσότερη ώρα ανοικτό η σύνοδος από την κανονική ώρα που ορίζει ο DSS αλλά και να αποστέλλει στον χρήστη στατιστικά της ποιότητας μετάδοσης των δεδομένων. Θα θέλαμε να δούμε πώς μεταβάλλεται η χρήση του Cpu και του Memory στην μηχανή μας, από την διεργασία που χρησιμοποιεί ο DSS όσον αυξάνεται ο αριθμός των αιτημάτων. Με αυτό τον τρόπο θα αναγκάσουμε τον διακομιστή

μας να καταναλώσει αρκετούς από τους πόρους του αφού θα χρειάζεται να σπαταλάει πόρους σε συνόδους οι οποίοι θα έπρεπε να τερματιστούν αλλά και να τοποθετεί στην αποστολή των δεδομένων και στατιστικά . Και όλο αυτό με χρονική άφιξη των αιτημάτων μία σταθερή μεταβλητή.

Open RTSP Command :

Σχόλια : αύξηση στο duration time κατά 10 sec

```
Openrtsp -Q -d -10 rtsp://192.168.10.6/sample_300kbit.mov
```

3.9.2 Περιγραφή Πειράματος Χρησιμοποιώντας Την Εντολή Του OpenRtsp Για Αλλαγή στην Διάρκεια Αναπαραγωγής Του Αρχείου Και Quality Of Service Και Χρόνο Εκδοσης Των Αιτημάτων Μία Τυχαία Μεταβλητή

Με την χρήση του εργαλείου OpenRTSP θα εξομοιώσουμε επιθέσεις με πεντακόσια, επτακόσια, οκτακόσια, χίλια, χίλια εκατό requests. Όλα τα requests θα ζητούν από τον διακομιστή 4-5 διαφορετικά αρχεία. Στα request θα ζητούμε από τον διακομιστή και Quality Of Statistics .Τα αρχεία που θα ζητούνται από τον διακομιστή θα είναι και διαφορετικής μορφής, όπως mpeg4,mp3, avi,mov.. Κάθε session θα αρχίζει με καθυστέρηση (x) ml second. Όπου (x) να είναι μία σταθερή μεταβλητή. Όλα τα session θα έχουν διαφορετικό duration time από το κανονικό duration που θα είχε το αρχείο. Η διαφορά στο duration θα είναι μία random μεταβλητή. Στόχος του πειράματος είναι η δημιουργία DoS στον διακομιστή με αιτήματα τα οποία αποστέλλονται μεσω παραλλαγής του εργαλείου OpenRtsp, και είναι υπεύθυνο να κρατάει για περισσότερη ώρα ανοικτό η σύνοδος από την κανονική ώρα που ορίζει ο DSS αλλά και να αποστέλλει στον χρήστη στατιστικά της ποιότητας μετάδοσης τν δεδομένων. Θα θέλαμε να δούμε πώς μεταβάλλεται η χρήση του Cpu και του Memory στην μηχανή μας, από την διεργασία που χρησιμοποιεί ο DSS όσον αυξάνεται ο αριθμός των αιτημάτων Με αυτό τον τρόπο θα αναγκάσουμε τον διακομιστή να σπαταλάει πόρους σε συνόδους οι οποίοι θα έπρεπε να τερματιστούν αλλά και να τοποθετεί

στην αποστολή των δεδομένων και στατιστικά . Και όλο αυτό με χρονική άφιξη των αιτημάτων μία τυχαία μεταβλητή.

Open RTSP Command :

Σχόλια : αύξηση στο duration time κατά 10 sec

```
Openrtsp -Q -d -10 rtsp://192.168.10.6/sample_300kbit.mov
```

Κεφάλαιο 4

Προτεινόμενες Λύσεις Για Πιθανές Επιθέσεις Denial Of Service

Σε αυτό το κεφάλαιο θα αναλύσουμε πιθανές λύσεις στις οποίες θα μπορούσαμε να εφαρμόσουμε για να αντιμετωπίσουμε πιθανές επιθέσεις Denial Of Service στον διακομιστή μας μέσω του πρωτοκόλλου RTSP.

4.1 Προτεινόμενες Λύσεις

Σε αυτή την ενότητα θα παραθέσουμε κάποιες λύσεις για αντιμετώπιση επιθέσεων στο πρωτόκολλο RTSP.

4.1.1 Προτεινόμενη Λύση για αντιμετώπιση πιθανής επίθεση Describe Message

Μπορούμε με την τοποθέτηση κάποιου Firewall στην περιφέρεια του δικτύου μας όπου βρίσκεται ο Streaming Server, να ελέγχουμε την κίνηση από και προς το δίκτυο μας. Θα έχουμε κάποιο κανόνα στο Firewall μας όπου θα σταματάει τα πακέτα τα οποία έρχονται από μία διεύθυνση ή πλήθος διευθύνσεων σχεδόν ταυτόχρονα και θα έχουν ως προορισμό τον Streaming Server. Για παράδειγμα μπορούμε να βάλουμε κανόνα ο οποίος θα επιτρέπει σε μία διεύθυνση να δημιουργεί μέχρι δέκα συνόδους ανά λεπτό.

4.1.2 Προτεινόμενη Λύση για αντιμετώπιση πιθανής επίθεση Options Message

Μπορούμε με την τοποθέτηση κάποιου Firewall στην περιφέρεια του δικτύου μας όπου βρίσκεται ο Streaming Server, να ελέγχουμε την κίνηση από και προς το δίκτυο μας. Θα έχουμε κάποιο κανόνα στο Firewall μας όπου θα σταματάει τα πακέτα τα οποία έρχονται από μία διεύθυνση ή πλήθος διευθύνσεων σχεδόν ταυτόχρονα και θα έχουν ως προορισμό τον Streaming Server. Για παράδειγμα μπορούμε να βάλουμε κανόνα ο οποίος θα επιτρέπει σε μία διεύθυνση να δημιουργεί μέχρι δέκα συνόδους ανά λεπτό.

4.1.3 Προτεινόμενη Λύση για αντιμετώπιση πιθανής επίθεση Get_Parameters Message

Μπορούμε με την τοποθέτηση κάποιου Firewall στην περιφέρεια του δικτύου μας όπου βρίσκεται ο Streaming Server, να ελέγχουμε την κίνηση από και προς το δίκτυο μας. Θα έχουμε κάποιο κανόνα στο Firewall μας όπου θα σταματάει τα πακέτα τα οποία έρχονται από μία διεύθυνση ή πλήθος διευθύνσεων σχεδόν ταυτόχρονα και θα έχουν ως προορισμό τον Streaming Server. Για παράδειγμα μπορούμε να βάλουμε κανόνα ο οποίος θα επιτρέπει σε μία διεύθυνση να δημιουργεί μέχρι δέκα συνόδους ανά λεπτό.

Ακολούθως θα μπορούσαμε να έχουμε ένα ξεχωριστό Streaming Server στο δίκτυο μας, ακριβώς ο ίδιος όπως τον αρχικό. Δηλαδή ίδιο λειτουργικό αλλά κυρίως ίδιο περιεχόμενο αρχείων. Αυτός ο διακομιστής δεν θα προσφέρεται για δημόσια χρήση παρά μόνο για ιδιωτική. Θα βάλουμε κανόνα στο firewall μας ο οποίος θα αναγκάζει όλα τα μηνύματα που καταφθάνουν από εξωτερικές διευθύνσεις να τα προωθεί στον διακομιστή ιδιωτικής χρήσης. Εάν η αίτηση η οποία καταφθάνει είναι εντάξει και δεν προορίζεται για κάποιο αρχείο το οποίο δεν βρίσκεται στον διακομιστή θα αποστέλλει το αίτημα μέσω του Router πίσω στον διακομιστή δημόσιας χρήσης για να το διαχειριστή αυτός. Εάν όμως το αρχείο δεν βρίσκεται στον διακομιστή τότε το πακέτο θα απορρίπτετε.

4.1.4 Προτεινόμενη Λύση για αντιμετώπιση πιθανής επίθεση Teardown

Μπορούμε να χρησιμοποιήσουμε μία λύση την οποία χρησιμοποιεί είδη το πρωτόκολλο όταν είναι σε κατάσταση παύσης ένα media. Όταν το timeout parameter της συνόδου το οποίο καταγράφεται στο setup message τερματιστεί τότε πρέπει να τερματίζεται και η σύνοδος.

4.1.5 Προτεινόμενη Λύση για αντιμετώπιση πιθανής επίθεση Content_Len

Μία καλή λύση που θα μπορούσαμε να προτείνουμε για αυτού του τύπου επίθεσης, θα μπορούσαμε να χρησιμοποιήσουμε Intrusion Detection Systems (IDS) στο δίκτυο μας και να δώσουμε κανόνες στο σύστημα για τα πακέτα τα οποία προορίζονται για τον Streaming Server. Αφού περάσουν από το σύστημα τότε θα αναλύονται για τυχόν ανωμαλίες. Εάν κάποιο μήνυμα διαπιστωθεί ότι υπάρχει ανωμαλία απορρίπτετε και καταστρέφεται από το σύστημα προτού παραληφθεί από τον Streaming Server αφού το σύστημα είναι ενδιάμεσα στον διακομιστή και τον router.

4.1.6 Προτεινόμενη Λύση για αντιμετώπιση πιθανής επίθεση Record

Μπορούμε με την τοποθέτηση κάποιου Firewall στην περιφέρεια του δικτύου μας όπου βρίσκεται ο Streaming Server, να ελέγχουμε την κίνηση από και προς το δίκτυο μας. Θα έχουμε κάποιο κανόνα στο Firewall μας όπου θα σταματάει τα πακέτα τα οποία έρχονται από μία διεύθυνση ή πλήθος διευθύνσεων σχεδόν ταυτόχρονα και θα έχουν ως προορισμό τον Streaming Server. Για παράδειγμα μπορούμε να βάλουμε κανόνα ο οποίος θα επιτρέπει σε μία διεύθυνση να δημιουργεί μέχρι δέκα συνόδους ανά λεπτό.

4.1.7 Προτεινόμενη Λύση για αντιμετώπιση πιθανής επίθεσης Play

Μπορούμε με την τοποθέτηση κάποιου Firewall στην περιφέρεια του δικτύου μας όπου βρίσκεται ο Streaming Server, να ελέγχουμε την κίνηση από και προς το δίκτυο μας. Θα έχουμε κάποιο κανόνα στο Firewall μας όπου θα σταματάει τα πακέτα τα οποία έρχονται από μία διεύθυνση ή πλήθος διευθύνσεων σχεδόν ταυτόχρονα και θα έχουν ως προορισμό τον Streaming Server. Για παράδειγμα μπορούμε να βάλουμε κανόνα ο οποίος θα επιτρέπει σε μία διεύθυνση να δημιουργεί μέχρι δέκα συνόδους ανά λεπτό.

Επίσης θα μπορούσαμε να προτείνουμε για αυτού του τύπου επίθεσης, θα μπορούσαμε να χρησιμοποιήσουμε Intrusion Detection Systems (IDS) στο δίκτυο μας και να δώσουμε κανόνες στο σύστημα για τα πακέτα τα οποία προορίζονται για τον Streaming Server. Αφού περάσουν από το σύστημα τότε θα αναλύονται για τυχόν ανωμαλίες. Εάν κάποιο μήνυμα διαπιστωθεί ότι υπάρχει ανωμαλία απορρίπτετε και καταστρέφεται από το σύστημα προτού παραληφθεί από τον Streaming Server αφού το σύστημα είναι ενδιάμεσα στον διακομιστή και τον router.

4.1.8 Προτεινόμενη Λύση για αντιμετώπιση πιθανής επίθεσης Set_Up message

Μπορούμε με την τοποθέτηση κάποιου Firewall στην περιφέρεια του δικτύου μας όπου βρίσκεται ο Streaming Server, να ελέγχουμε την κίνηση από και προς το δίκτυο μας. Θα έχουμε κάποιο κανόνα στο Firewall μας όπου θα σταματάει τα πακέτα τα οποία έρχονται από μία διεύθυνση ή πλήθος διευθύνσεων σχεδόν ταυτόχρονα και θα έχουν ως προορισμό τον Streaming Server. Για παράδειγμα μπορούμε να βάλουμε κανόνα ο οποίος θα επιτρέπει σε μία διεύθυνση να δημιουργεί μέχρι δέκα συνόδους ανά λεπτό.

4.1.9 Προτεινόμενη Λύση για αντιμετώπιση πιθανής επίθεσης Set_Parameters

Μπορούμε με την τοποθέτηση κάποιου Firewall στην περιφέρεια του δικτύου μας όπου βρίσκεται ο Streaming Server, να ελέγχουμε την κίνηση από και προς το δίκτυο μας. Θα έχουμε κάποιο κανόνα στο Firewall μας όπου θα σταματάει τα πακέτα τα οποία έρχονται από μία διεύθυνση ή πλήθος διευθύνσεων σχεδόν ταυτόχρονα και θα έχουν ως προορισμό τον Streaming Server. Για παράδειγμα μπορούμε να βάλουμε κανόνα ο οποίος θα επιτρέπει σε μία διεύθυνση να δημιουργεί μέχρι δέκα συνόδους ανά λεπτό.

Επίσης θα μπορούσαμε να προτείνουμε για αυτού του τύπου επίθεσης, θα μπορούσαμε να χρησιμοποιήσουμε Intrusion Detection Systems (IDS) στο δίκτυο μας και να δώσουμε κανόνες στο σύστημα για τα πακέτα τα οποία προορίζονται για τον Streaming Server. Αφού περάσουν από το σύστημα τότε θα αναλύονται για τυχόν ανωμαλίες. Εάν κάποιο μήνυμα διαπιστωθεί ότι υπάρχει ανωμαλία απορρίπτετε και καταστρέφεται από το σύστημα προτού παραληφθεί από τον Streaming Server αφού το σύστημα είναι ενδιάμεσα στον διακομιστή και τον router.

4.1.10 Προτεινόμενη Λύση για αντιμετώπιση πιθανής επίθεσης Hijack

Μπορούμε με την τοποθέτηση κάποιου Firewall στην περιφέρεια του δικτύου μας όπου βρίσκεται ο Streaming Server, να ελέγχουμε την κίνηση από και προς το δίκτυο μας. Θα έχουμε κάποιο κανόνα στο Firewall μας όπου θα σταματάει τα πακέτα τα οποία έρχονται από μία διεύθυνση ή πλήθος διευθύνσεων σχεδόν ταυτόχρονα και θα έχουν ως προορισμό τον Streaming Server. Για παράδειγμα μπορούμε να βάλουμε κανόνα ο οποίος θα επιτρέπει σε μία διεύθυνση να δημιουργεί μέχρι δέκα συνόδους ανά λεπτό.

Επίσης θα μπορούσαμε να προτείνουμε για αυτού του τύπου επίθεσης, θα μπορούσαμε να χρησιμοποιήσουμε Intrusion Detection Systems (IDS) στο δίκτυο μας και να δώσουμε κανόνες στο σύστημα για τα πακέτα τα οποία προορίζονται για τον Streaming Server. Αφού περάσουν από το σύστημα τότε θα αναλύονται για τυχόν ανωμαλίες. Εάν κάποιο μήνυμα διαπιστωθεί ότι υπάρχει ανωμαλία απορρίπτετε και καταστρέφεται από το σύστημα προτού παραληφθεί από τον Streaming Server αφού το σύστημα είναι ενδιάμεσα στον διακομιστή και τον router.

Κεφάλαιο 5

Παρουσίαση Στατιστικών Μετρήσεων Ανά Σενάριο

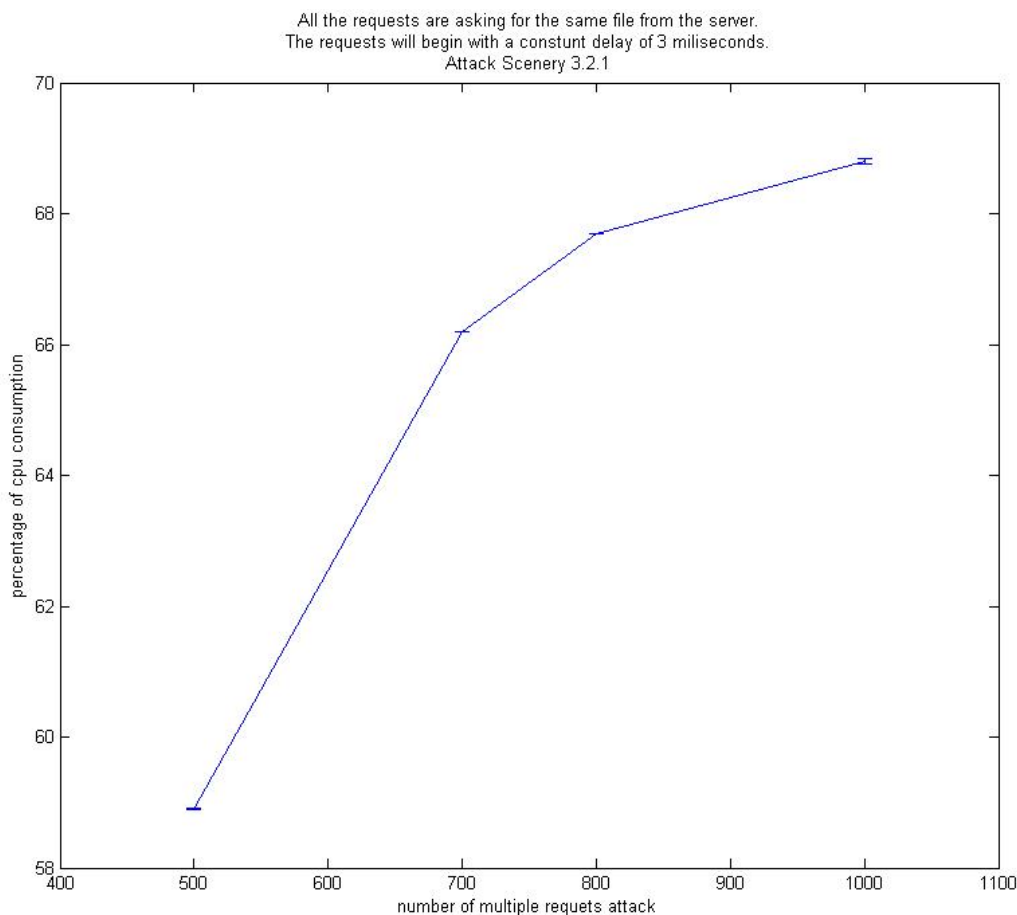
Σε αυτό το κεφάλαιο θα παρουσιάσουμε με γραφικές παραστάσεις τις μετρήσεις που πήραμε από τα πειράματα. Σε κάθε κεφάλαιο παρουσιάζουμε τον μέσο όρο κατανάλωσης cpu και memory στον Darwin Streaming Server, από τις δεκαπέντε επαναληπτικές δοκιμές ανά σενάριο. Όπου δεν χρειάστηκε πρόπον αφού τα αποτελέσματα των μετρήσεων ήταν αμελητέα, οι γραφικές παραστάσεις δεν παρουσιάστηκαν.

5.1 Στατιστικά Επιθέσεων Χρησιμοποιώντας Την Απλή Εντολή Του OpenRtsp

Σε αυτή την ενότητα θα αναλύσουμε τα αποτελέσματα που συλλέχτηκαν κατά την εκτέλεση του σεναρίου χρησιμοποιώντας την απλή εντολή του OpenRtsp.

5.1.1 Ανάλυση Αποτελεσμάτων Σεναρίου Χρησιμοποιώντας Την Απλή Εντολή Του OpenRtsp Και Χρόνο Έκδοσης Των Αιτημάτων Μία Σταθερή Μεταβλητή

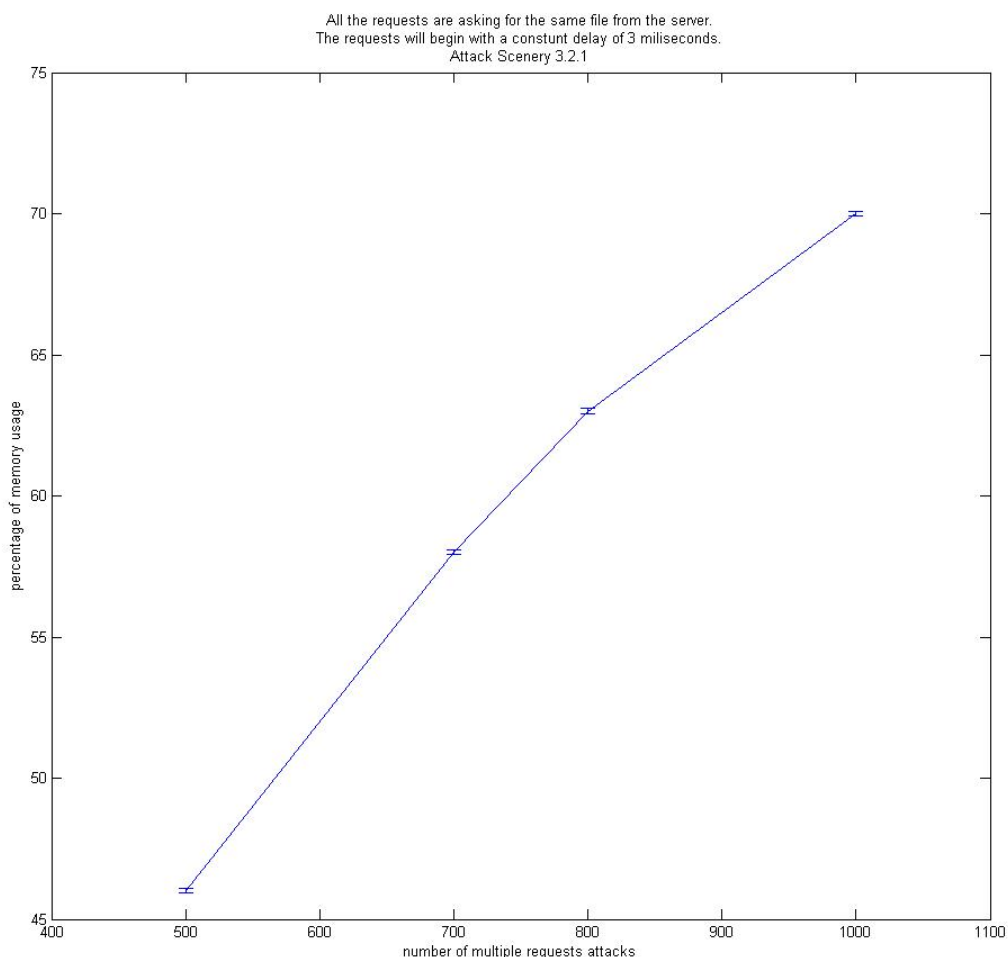
Πιο κάτω παρουσιάζουμε τα γραφικά αποτελέσματα του σεναρίου 3.2.1. Στην γραφική παράσταση αναγράφεται η μέση τιμή για την κάθε παραλλαγή του σεναρίου. Οι παραλλαγές του σεναρίου υπενθυμίζουμε είναι για πεντακόσια, επτακόσια, οκτακόσια και χίλια αιτήματα. Στο πιο κάτω διάγραμμα παρατηρούμε την αύξηση της χρήσης του CPU όσον αυξάνονται και τα αιτήματα.



Γραφική 5.1: Μέση τιμή κατανάλωσης CPU χρησιμοποιώντας την απλή εντολή του OpenRtsp και χρόνο έκδοσης των αιτημάτων μία σταθερή μεταβλητή

Παρατηρούμε ότι για πεντακόσια αιτήματα η μέση τιμή κατανάλωσης CPU από την διεργασία του DSS είναι 59%. Για επτακόσια αιτήματα έχουμε μία αύξηση της τάξεως του 7% και η μέση τιμή ανεβαίνει στο 66%. Για οκτακόσια αιτήματα έχουμε μία αύξηση 2% και η μέση τιμή ανεβαίνει στο 68%. Για χίλια αιτήματα η μέση τιμή ανεβαίνει στο 69%. Παρατηρούμε μία λογική αύξηση της μέσης τιμής κατανάλωσης του CPU αφού ο διακομηστής μας αναγκάζεται να εξυπηρετήσει όλο και περισσότερα αιτήματα, σχεδόν ταυτόχρονα.

Στην επόμενη γραφική παράσταση θα παρατηρήσουμε την συμπεριφορά της διεργασίας του DSS ως προς την κατανάλωση Memory. Αναγράφεται για κάθε παραλλαγή του σεναρίου η μέση τιμή κατανάλωσης μνήμης. Παρατηρούμε μία σημαντική αύξηση στο ποσοστό κατανάλωσης της μνήμης όσον αυξάνονται τα



Γραφική 5.2: Μέση τιμή κατανάλωσης Memory χρησιμοποιώντας την απλή εντολή του OpenRtsp και χρόνο έκδοσης των αιτημάτων μία σταθερή μεταβλητή

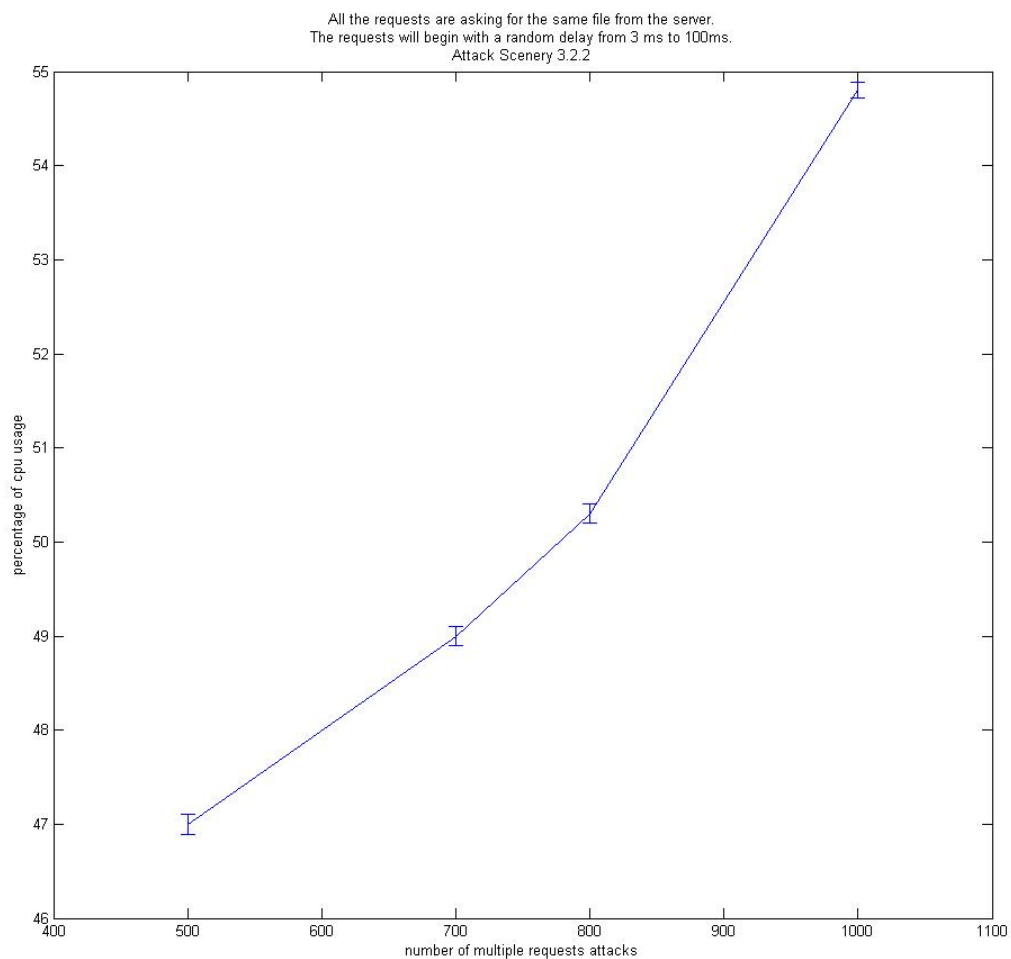
αιτήματα προς τον διακομηστή. Και αυτό είναι λογικό αφού όσον πιο πολλά ταυτόχρονα αιτήματα λάβει ο διακομηστής, τόσο πιο πολύ θα χρειαστεί να καταναλώσει πόρους για να εξυπηρετήσει όλα τα αιτήματα. Διαπιστώνουμε ότι για πεντακόσια αιτήματα η μέση τιμή κατανάλωσης Memory από την διεργασία του DSS είναι 46%. Για επτακόσια αιτήματα έχουμε μία σημαντική αύξηση της τάξεως

του 11% και η μέση τιμή ανεβαίνει στο 57%. Για οκτακόσια αιτήματα έχουμε μία αύξηση 6% και η μέση τιμή ανεβαίνει στο 63%. Τέλος για χίλια αιτήματα η μέση τιμή ανεβαίνει στο 70%.

Το γενικό συμπέρασμα που αποκομίζουμε από τις πιο πάνω γραφικές παραστάσεις είναι ότι η κατανάλωση cpu και memory από την διεργασία του DSS ανεβαίνει όσο ανεβαίνουν και τα ταυτόχρονα αιτήματα. Ο διακομιστής θέλοντας να εξυπηρετήσει όλα τα αιτήματα αφού αυτός είναι και ο σκοπός του, καταναλώνει όλο και περισσότερους πόρους όσο τα αιτήματα στο socket του συσσωρεύονται.

5.1.2 Ανάλυση Αποτελεσμάτων Σεναρίου Περιγραφή Πειράματος Χρησιμοποιώντας Την Απλή Εντολή Του OpenRtsp Και Χρόνο Έκδοσης Των Αιτημάτων Μία Τυχαία Μεταβλητή

Πιο κάτω παρουσιάζουμε τα γραφικά αποτελέσματα του σεναρίου 3.2.2. Στην γραφική παράσταση αναγράφεται η μέση τιμή για την κάθε παραλλαγή του σεναρίου. Οι παραλλαγές του σεναρίου υπενθυμίζουμε είναι για πεντακόσια, επτακόσια, οκτακόσια και χίλια αιτήματα. Στο πιο κάτω διάγραμμα παρατηρούμε την αύξηση της χρήσης του CPU όσον αυξάνονται και τα αιτήματα.

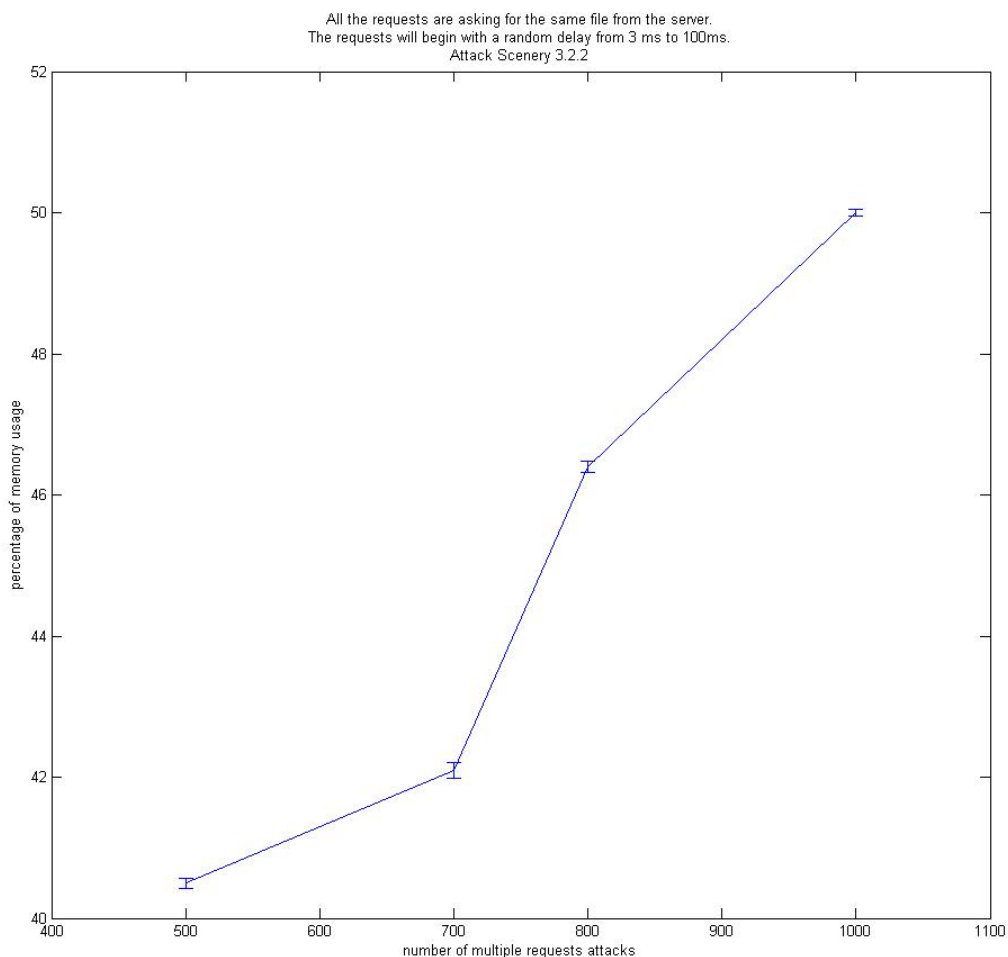


Γραφική 5.3: Μέση τιμή κατανάλωσης CPU Χρησιμοποιώντας Την Απλή Εντολή Του OpenRtsp Και Χρόνο Έκδοσης Των Αιτημάτων Μία Τυχαία Μεταβλητή.

Παρατηρούμε ότι για πεντακόσια αιτήματα η μέση τιμή κατανάλωσης CPU είναι 47% για επτακόσια αιτήματα 49%, για οκτακόσια αιτήματα 50% και φτάνει σχεδόν το 55% στα χίλια

αιτήματα. Είναι μία λογική συνέπεια η αύξηση στο ποσοστό κατανάλωσης CPU αφού τα αιτήματα που παραλαμβάνει ο διακομιστής είναι αρκετά περισσότερα.

Στην επόμενη γραφική παράσταση θα παρατηρήσουμε την συμπεριφορά της διεργασίας του DSS ως προς την κατανάλωση Memory. Αναγράφεται για κάθε παραλλαγή του σεναρίου η μέση τιμή κατανάλωσης μνήμης. Παρατηρούμε μία σημαντική αύξηση στο ποσοστό κατανάλωσης της μνήμης όσον αυξάνονται τα αιτήματα.



Γραφική 5.4: Μέση τιμή κατανάλωσης Memory Χρησιμοποιώντας Την Απλή Εντολή Του OpenRtsp Και Χρόνο Έκδοσης Των Αιτημάτων Μία Τυχαία Μεταβλητή.

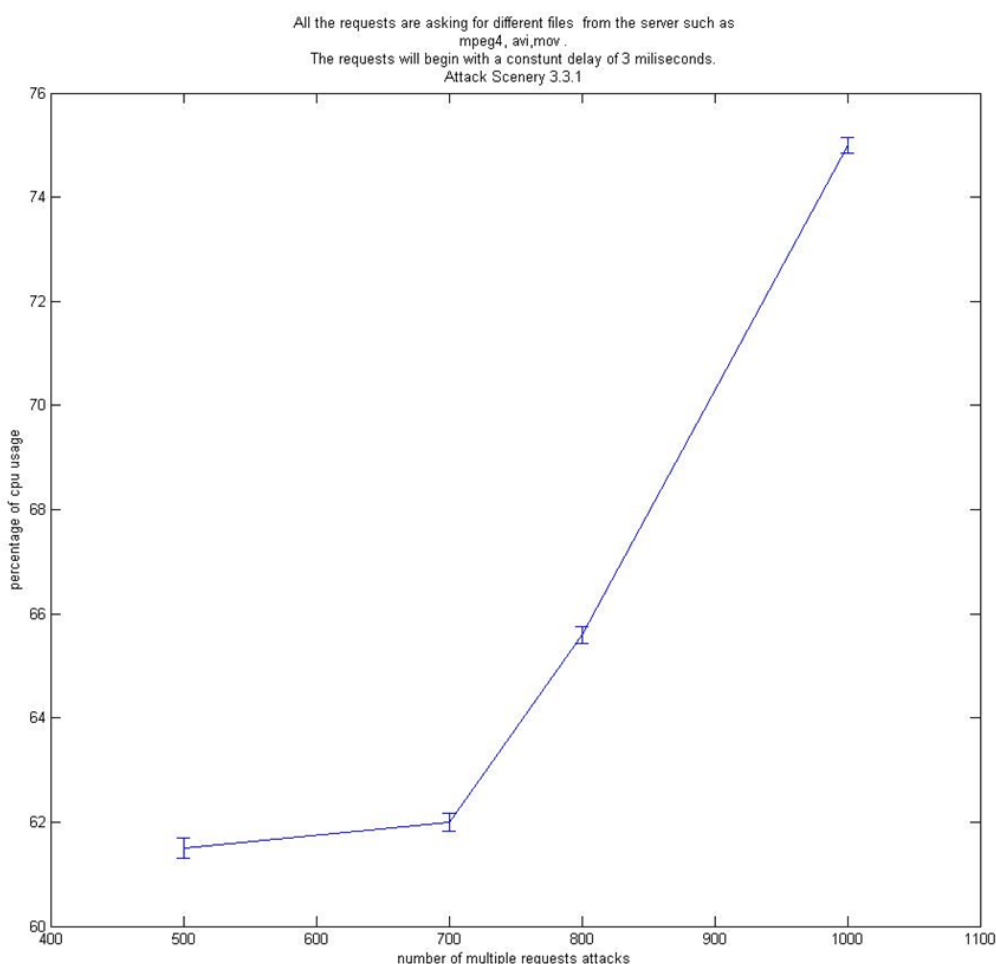
Παρατηρούμε ότι για πεντακόσια αιτήματα η μέση τιμή κατανάλωσης CPU είναι 41% για επτακόσια αιτήματα 42,5%, για οκτακόσια αιτήματα 47% και φτάνει σχεδόν το 50% στα χίλια αιτήματα. Είναι μία λογική συνέπεια η αύξηση στο ποσοστό κατανάλωσης Memory αφού τα αιτήματα που παραλαμβάνει ο διακομιστής είναι αρκετά περισσότερα.

Το γενικό συμπέρασμα που αποκομίζουμε από τις πιο πάνω γραφικές παραστάσεις είναι ότι η κατανάλωση cpu και memory από την διεργασία του DSS ανεβαίνει όσο ανεβαίνουν και τα ταυτόχρονα αιτήματα. Ο διακομιστής θέλοντας να εξυπηρετήσει όλα τα αιτήματα αφού αυτός είναι και ο σκοπός του, καταναλώνει όλο και περισσότερους πόρους όσο τα αιτήματα στο socket του συσσωρεύονται.

Επιπλέον εδώ μπορούμε να αναφέρουμε ότι σε σύγκριση των δύο πειραμάτων τα οποία είναι αρκετά όμοια, και διαφέρουν μόνο στην ώρα άφιξης των αιτημάτων στον διακομιστή, διαπιστώνουμε ότι περισσότερη ζημιά δημιουργούν τα αιτήματα τα οποία καταφθάνουν στον διακομιστή με ελάχιστη διαφορά χρόνου. Ο διακομιστής προσπαθεί να τα εξυπηρετήσει όλα σχεδόν ταυτόχρονα και έτσι σπαταλάει αρκετή από την επεξεργαστική του δύναμη. Θυμίζουμε ότι όταν είχαμε χίλια αιτήματα με μία σταθερή μεταβλητή ανέβηκε στο 69% και η κατανάλωση μνήμης στο 70%. Αντίθετα όταν ο διακομιστής έχει κάποιο χρονικό περιθώριο για να εξυπηρετήσει κάποια αιτήματα, δεν δυσκολεύεται και ιδιαίτερα αφού για χίλια αιτήματα φτάνουμε για κατανάλωση CPU και Memory 55% και 50% αντίστοιχα.

5.1.3 Ανάλυση Αποτελεσμάτων Σεναρίου Χρησιμοποιώντας Την Απλή Εντολή Του OpenRtsp Ζητώντας Διαφορετικά Αρχεία Από Τον Streaming Server Και Χρόνο Έκδοσης Των Αιτημάτων Μία Σταθερή Μεταβλητή

Πιο κάτω παρουσιάζουμε τα γραφικά αποτελέσματα του σεναρίου 3.3.1. Στην γραφική παράσταση αναγράφεται η μέση τιμή για την κάθε παραλλαγή του σεναρίου. Οι παραλλαγές του σεναρίου υπενθυμίζουμε είναι για πεντακόσια, επτακόσια, οκτακόσια και χίλια αιτήματα. Στο πιο κάτω διάγραμμα παρατηρούμε την αύξηση της χρήσης του CPU όσον αυξάνονται και τα αιτήματα.

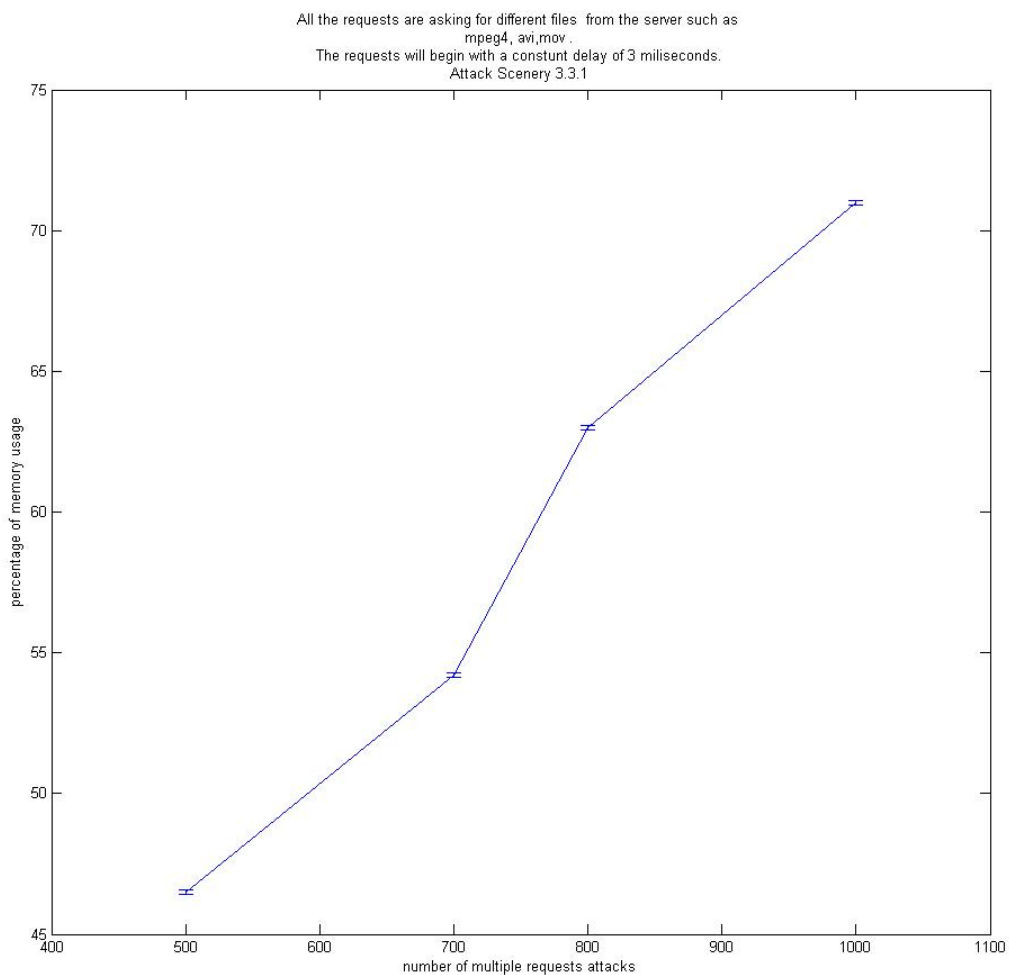


Γραφική 5.5: Μέση τιμή κατανάλωσης CPU Χρησιμοποιώντας Την Απλή Εντολή Του OpenRtsp Ζητώντας Διαφορετικά Αρχεία Από Τον Streaming Server Και Χρόνο Έκδοσης Των Αιτημάτων Μία Σταθερή Μεταβλητή

Παρατηρούμε ότι για πεντακόσια αιτήματα η μέση τιμή κατανάλωσης CPU από την διεργασία του DSS είναι 61,5%. Για επτακόσια αιτήματα έχουμε μία αύξηση της τάξεως του 0,5% και η

μέση τιμή ανεβαίνει στο 62%. Για οκτακόσια αιτήματα έχουμε μία αύξηση 4% και η μέση τιμή ανεβαίνει στο 66%. Για χίλια αιτήματα η μέση τιμή εκτοξεύεται στο 75%. Παρατηρούμε μία λογική αύξηση της μέσης τιμής κατανάλωσης του CPU αφού ο διακομιστής μας αναγκάζεται να εξυπηρετήσει όλο και περισσότερα αιτήματα, σχεδόν ταυτόχρονα επιπλέον είναι και για διαφορετικά αρχεία.

Στην επόμενη γραφική παράσταση θα παρατηρήσουμε την συμπεριφορά της διεργασίας του DSS ως προς την κατανάλωση Memory. Αναγράφεται για κάθε παραλλαγή του σεναρίου η μέση τιμή κατανάλωσης μνήμης. Παρατηρούμε μία σημαντική αύξηση στο ποσοστό κατανάλωσης της μνήμης όσον αυξάνονται τα αιτήματα.



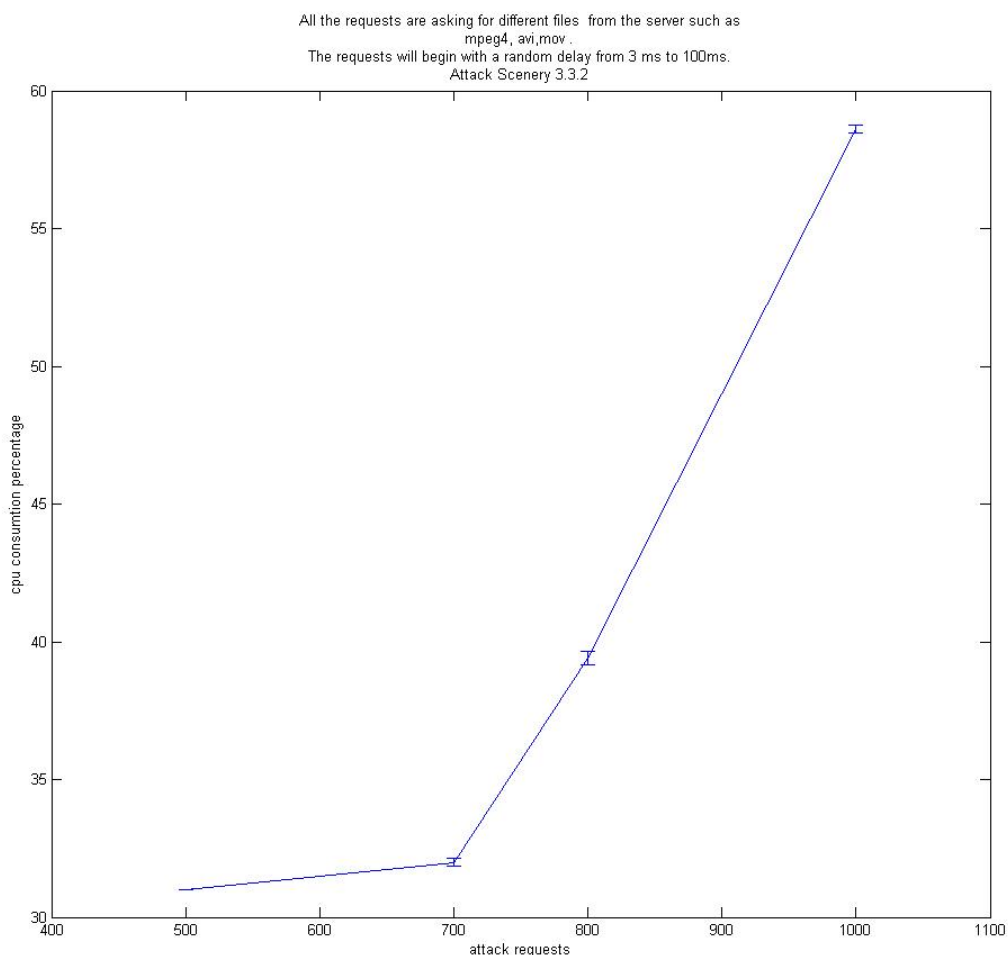
Γραφική 5.6: Μέση τιμή κατανάλωσης Memory Χρησιμοποιώντας Την Απλή Εντολή Του OpenRtsp Ζητώντας Διαφορετικά Αρχεία Από Τον Streaming Server Και Χρόνο Έκδοσης Των Αιτημάτων Μία Σταθερή Μεταβλητή

Διαπιστώνουμε ότι για άλλη μία φορά όσον αυξάνονται τα αιτήματα προς τον διακομιστή να αυξάνεται και η κατανάλωση μνήμης για να μπορέσει ο διακομιστής να ανταποκριθεί στα αιτήματα που λαμβάνει. Για πεντακόσια αιτήματα έχουμε 46% χρησιμοποίηση μνήμης από την διεργασία του DSS, για επτακόσια αιτήματα ανεβαίνει στο 56%, για οκτακόσια αιτήματα ανεβαίνει στο 64% και τέλος για χίλια αιτήματα φτάνει το 71%.

Το γενικό συμπέρασμα που παίρνουμε από τις πιο πάνω γραφικές παραστάσεις είναι ότι και πάλι όσον αυξάνονται τα αιτήματα προς τον διακομιστή αυξάνεται και η κατανάλωση επεξεργαστικής δύναμης αλλά και μνήμης από τον για να μπορέσει να ανταποκριθεί σε όλα τα αιτήματα. Μπορούμε να συγκρίνουμε αυτό το πείραμα με το 3.2.1, όπου η μοναδική διαφορά είναι ότι ζητούνται διαφορετικά αρχεία κάθε φορά, και να παρατηρήσουμε ότι σε αυτό το πείραμα καταφέραμε να καταναλώσουμε από την επεξεργαστική δύναμη του DSS, στα χίλια αιτήματα 75% ενώ στο προηγούμενο καταναλώσαμε 70% στα χίλια αιτήματα. Άρα εάν ο DSS δέχεται επίθεση από μία ομάδα χρηστών με σκοπό την πρόκληση DoS και ζητούν διαφορετικά αρχεία θα καταφέρουν να στρεσάρουν περισσότερο τον DSS παρά να ζητούν όλοι το ίδιο αρχείο. Όσο αφορά την κατανάλωση μνήμης κυμάνθηκε περίπου στα ίδια επίπεδα, αλλά και πάλι σε αυτό το πείραμα είχαμε 1% αύξηση στα χίλια αιτήματα σε σχέση με το 3.2.1.

3.1.4 Ανάλυση Αποτελεσμάτων Σεναρίου Χρησιμοποιώντας Την Απλή Εντολή Του OpenRtsp Ζητώντας Διαφορετικά Αρχεία Από Τον Streaming Server Και Χρόνο Έκδοσης Των Αιτημάτων Μία Τυχαία Μεταβλητή

Πιο κάτω παρουσιάζουμε τα γραφικά αποτελέσματα του σεναρίου 3.3.2. Στην γραφική παράσταση αναγράφεται η μέση τιμή για την κάθε παραλλαγή του σεναρίου. Οι παραλλαγές του σεναρίου υπενθυμίζουμε είναι για πεντακόσια, επτακόσια, οκτακόσια και χίλια αιτήματα. Στο πιο κάτω διάγραμμα παρατηρούμε την αύξηση της χρήσης του CPU όσον αυξάνονται και τα αιτήματα.

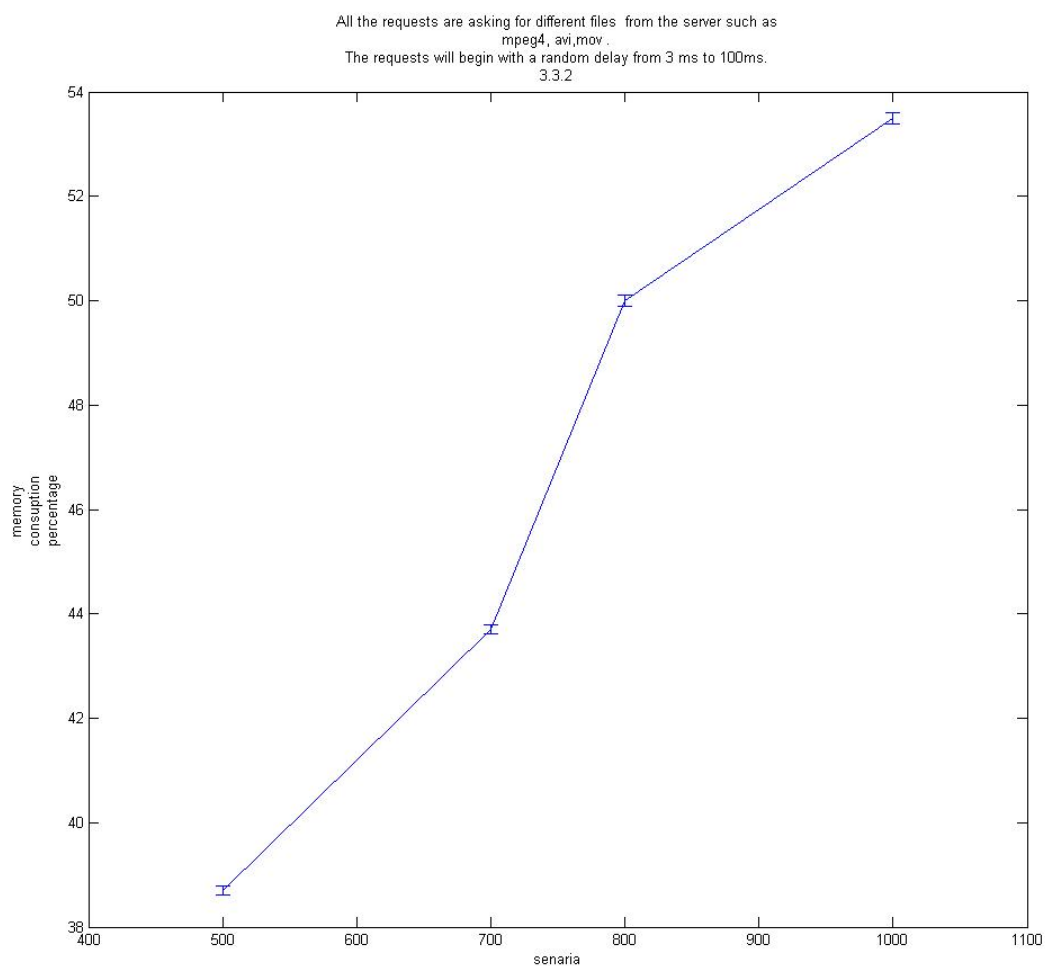


Γραφική 5.7: Μέση τιμή κατανάλωσης Cpu Χρησιμοποιώντας Την Απλή Εντολή Του OpenRtsp Ζητώντας Διαφορετικά Αρχεία Από Τον Streaming Server Και Χρόνο Έκδοσης Των Αιτημάτων Μία Τυχαία Μεταβλητή

Παρατηρούμε ότι για πεντακόσια αιτήματα η μέση τιμή κατανάλωσης CPU από την διεργασία του DSS είναι 31%. Για επτακόσια αιτήματα έχουμε μία αύξηση της τάξεως του 1% και η μέση τιμή ανεβαίνει στο 32%. Για οκτακόσια αιτήματα έχουμε μία αύξηση 7% και η μέση τιμή

ανεβαίνει στο 39%. Για χίλια αιτήματα η μέση τιμή εκτοξεύεται στο 59%. Παρατηρούμε μία λογική αύξηση της μέσης τιμής κατανάλωσης του CPU αφού ο διακομιστής μας αναγκάζεται να εξυπηρετήσει όλο και περισσότερα αιτήματα, και επιπλέον είναι και για διαφορετικά αρχεία.

Στην επόμενη γραφική παράσταση θα παρατηρήσουμε την συμπεριφορά της διεργασίας του DSS ως προς την κατανάλωση Memory. Αναγράφεται για κάθε παραλλαγή του σεναρίου η μέση τιμή κατανάλωσης μνήμης. Παρατηρούμε μία σημαντική αύξηση στο ποσοστό κατανάλωσης της μνήμης όσον αυξάνονται τα αιτήματα.



Γραφική 5.8: Μέση τιμή κατανάλωσης Memory Χρησιμοποιώντας Την Απλή Εντολή Του OpenRtsp Ζητώντας Διαφορετικά Αρχεία Από Τον Streaming Server Και Χρόνο Έκδοσης Των Αιτημάτων Μία Τυχαία Μεταβλητή

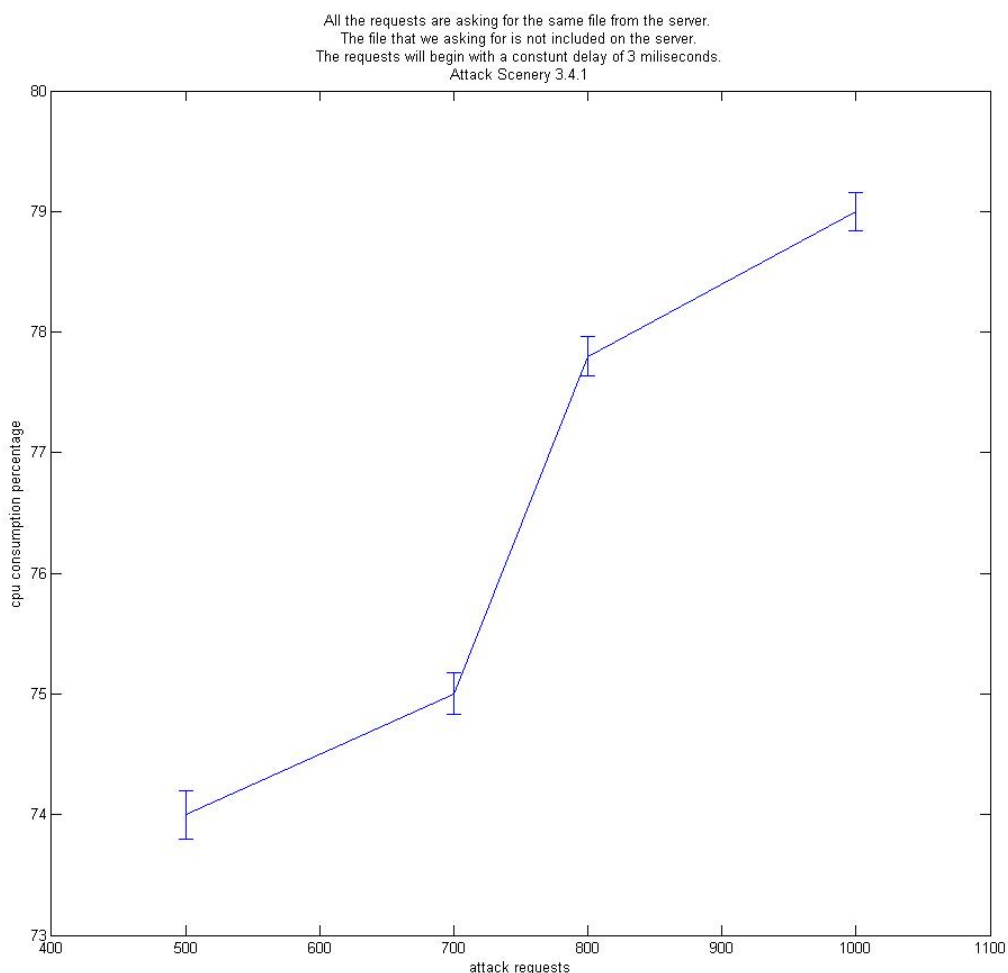
Παρατηρούμε ότι για πεντακόσια αιτήματα η μέση τιμή κατανάλωσης Memory από την διεργασία του DSS είναι 38,5%. Για επτακόσια αιτήματα έχουμε μία αύξηση της τάξεως του 5% και η μέση τιμή ανεβαίνει στο 43,5%. Για οκτακόσια αιτήματα έχουμε μία αύξηση 6,5% και η

μέση τιμή ανεβαίνει στο 50%. Για χίλια αιτήματα η μέση τιμή ανεβαίνει στο 53,5%. Παρατηρούμε μία λογική αύξηση της μέσης τιμής κατανάλωσης του CPU αφού ο διακομιστής μας αναγκάζεται να εξυπηρετήσει όλο και περισσότερα αιτήματα, και επιπλέον είναι και για διαφορετικά αρχεία.

Μπορούμε εδώ να αναφέρουμε ότι σημαντικό ρόλο στο στρεσάρισμα του DSS παίζει η χρονική άφιξης των αιτημάτων. Συγκρίνοντας το πείραμα 3.3.2 και το 3.3.1 που η μοναδική διαφορά στο σενάριο τους είναι η χρονική άφιξης των αιτημάτων, βλέπουμε ότι σε όλες τις παραλλαγές του πειράματος 3.3.1 όπου τα αιτήματα καταφθάνουν σχεδόν ταυτόχρονα οι μετρήσεις κατανάλωση των πόρων του DSS είναι πολύ ψιλότερες από το πείραμα 3.3.2, όπου τα αιτήματα καταφθάνουν με τυχαία χρονική επιλογή. Και αυτό συμβαίνει γιατί ο DSS έχει χρόνο για να αντιμετωπίσει αυτό το φαινόμενο και να εξυπηρετήσει όλα τα αιτήματα χωρίς να στρεσαριστεί.

5.1.5 Ανάλυση Αποτελεσμάτων Σεναρίου Χρησιμοποιώντας Την Απλή Εντολή Του OpenRtsp Ζητώντας Κάποιο Αρχείο Που Δεν Βρίσκεται Στον Streaming Server Και Χρόνο Έκδοσης Των Αιτημάτων Μία Σταθερή Μεταβλητή

Πιο κάτω παρουσιάζουμε τα γραφικά αποτελέσματα του σεναρίου 3.4.1. Στην γραφική παράσταση αναγράφεται η μέση τιμή για την κάθε παραλλαγή του σεναρίου. Οι παραλλαγές του σεναρίου υπενθυμίζουμε είναι για πεντακόσια, επτακόσια, οκτακόσια και χίλια αιτήματα. Στο πιο κάτω διάγραμμα παρατηρούμε την αύξηση της χρήσης του CPU όσον αυξάνονται και τα αιτήματα.



Γραφική 5.9: Μέση τιμή κατανάλωσης Cpu Χρησιμοποιώντας Την Απλή Εντολή Του OpenRtsp Ζητώντας Κάποιο Αρχείο Που Δεν Βρίσκεται Στον Streaming Server Και Χρόνο Έκδοσης Των Αιτημάτων Μία Σταθερή Μεταβλητή

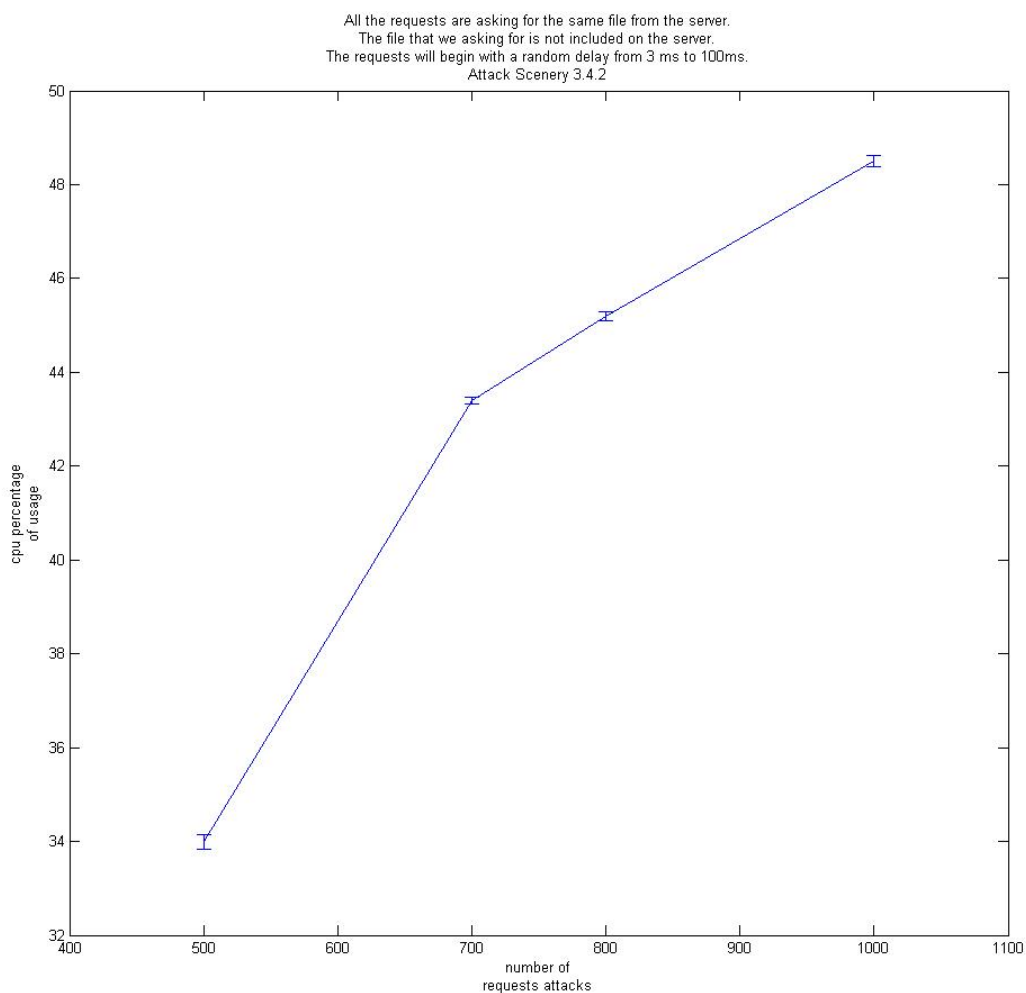
Παρατηρούμε ότι για πεντακόσια αιτήματα η μέση τιμή κατανάλωσης CPU από την διεργασία του DSS είναι 74%. Για επτακόσια αιτήματα έχουμε μία αύξηση της τάξεως του 1% και η μέση τιμή ανεβαίνει στο 75%. Για οκτακόσια αιτήματα έχουμε μία αύξηση 3% και η μέση τιμή

ανεβαίνει στο 78%. Για χίλια αιτήματα η μέση τιμή ανεβαίνει στο 79%. Παρατηρούμε μία αύξηση της μέσης τιμής κατανάλωσης του CPU όσον αυξάνονται τα αιτήματα. Ο DSS αναγκάζεται να σπαταλάει πόρους επεξεργαστική δύναμη αφού προσπαθεί να εξυπηρετήσει άσκοπα μηνύματα. Είναι άξιο αναφοράς ότι μέχρι τώρα έχουμε το πιο ψηλό ποσοστό χρήσης cpu και όλο αυτό για να εξυπηρετήσουμε ανώφελα αιτήματα.

Η μετρήσεις για κατανάλωση memory ήταν αμελητέες για αυτό και δεν παρουσιάζονται για σχολιασμό.

5.1.6 Ανάλυση Αποτελεσμάτων Σεναρίου Περιγραφή Πειράματος Χρησιμοποιώντας Την Απλή Εντολή Του OpenRtsp Ζητώντας Κάποιο Αρχείο Που Δεν Βρίσκεται Στον Streaming Server Και Χρόνο Εκδοσης Των Αιτημάτων Μία Τυχαία Μεταβλητή

Πιο κάτω παρουσιάζουμε τα γραφικά αποτελέσματα του σεναρίου 3.4.2. Στην γραφική παράσταση αναγράφεται η μέση τιμή για την κάθε παραλλαγή του σεναρίου. Οι παραλλαγές του σεναρίου υπενθυμίζουμε είναι για πεντακόσια, επτακόσια, οκτακόσια και χίλια αιτήματα. Στο πιο κάτω διάγραμμα παρατηρούμε την αύξηση της χρήσης του CPU όσον αυξάνονται και τα αιτήματα.



Γραφική 5.10: Μέση τιμή κατανάλωσης Cpu Χρησιμοποιώντας Την Απλή Εντολή Του OpenRtsp Ζητώντας Κάποιο Αρχείο Που Δεν Βρίσκεται Στον Streaming Server Και Χρόνο Εκδοσης Των Αιτημάτων Μία Τυχαία Μεταβλητή

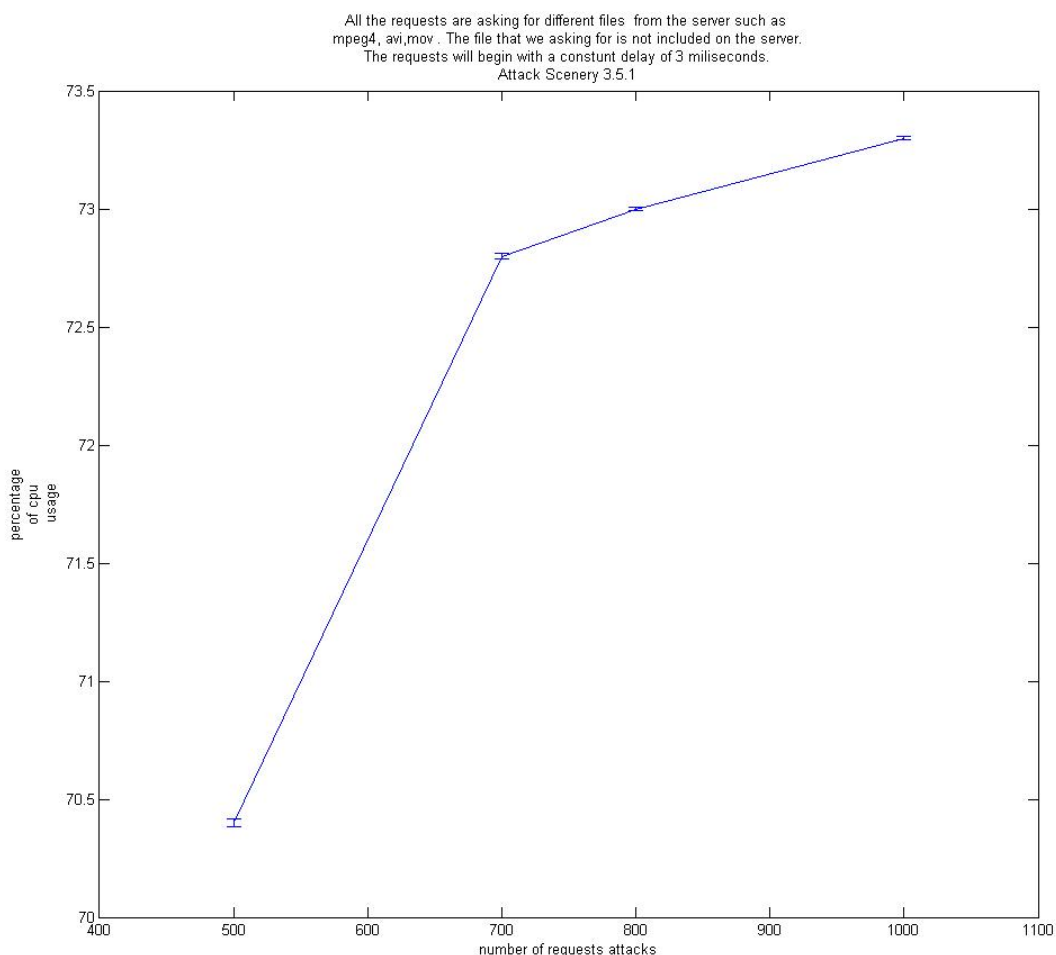
Παρατηρούμε ότι για πεντακόσια αιτήματα η μέση τιμή κατανάλωσης CPU από την διεργασία του DSS είναι 34%. Για επτακόσια αιτήματα έχουμε μία αύξηση της τάξεως του 9% και η μέση τιμή ανεβαίνει στο 43%. Για οκτακόσια αιτήματα έχουμε μία αύξηση 2% και η μέση τιμή ανεβαίνει στο 45%. Για χίλια αιτήματα η μέση τιμή ανεβαίνει στο 49%. Παρατηρούμε μία αύξηση της μέσης τιμής κατανάλωσης του CPU όσον αυξάνονται τα αιτήματα. Ο DSS αναγκάζεται να σπαταλάει πόρους επεξεργαστική δύναμη αφού προσπαθεί να εξυπηρετήσει άσκοπα μηνύματα.

Είναι άξιο αφοράς και πάλι ο χρόνος άφιξης των αιτημάτων αφού, εάν συγκρίνουμε το πείραμα 3.4.2 και 3.4.1 όπου η μόνη διαφορά τους είναι η χρονική αποστολή των μηνυμάτων θα παρατηρήσουμε τρομερές διαφορές στην αντίδραση του DSS αφού στο 3.4.1 η μετρήσεις εκτοξεύονται μέχρι και 79 % ενώ στο 3.4.2 φτάνουν μόνο στο 49 % . Αυτό το φαινόμενο συμβαίνει αφού ο DSS έχει χρόνο ώστε να εξυπηρετήσει όλες τις διεργασίες χωρίς να στρεσαριστεί, σε αντίθεση με το 3.4.1 όπου ο DSS κατακλύζεται από σχεδόν ταυτόχρονα αιτήματα εξυπηρέτησης.

Η μετρήσεις για κατανάλωση memory ήταν αμελητέες για αυτό και δεν παρουσιάζονται για σχολιασμό.

5.1.7 Ανάλυση Αποτελεσμάτων Σεναρίου Χρησιμοποιώντας Την Απλή Εντολή Του OpenRtsp Ζητώντας Αρχεία Που Δεν Βρίσκονται Στον Streaming Server Και Χρόνο Έκδοσης Των Αιτημάτων Μία Σταθερή Μεταβλητή

Πιο κάτω παρουσιάζουμε τα γραφικά αποτελέσματα του σεναρίου 3.5.1. Στην γραφική παράσταση αναγράφεται η μέση τιμή για την κάθε παραλλαγή του σεναρίου. Οι παραλλαγές του σεναρίου υπενθυμίζουμε είναι για πεντακόσια, επτακόσια, οκτακόσια και χίλια αιτήματα. Στο πιο κάτω διάγραμμα παρατηρούμε την αύξηση της χρήσης του CPU όσον αυξάνονται και τα αιτήματα.



Γραφική 5.11: Μέση τιμή κατανάλωσης Cpu Χρησιμοποιώντας Την Απλή Εντολή Του OpenRtsp Ζητώντας Αρχεία Που Δεν Βρίσκονται Στον Streaming Server Και Χρόνο Έκδοσης Των Αιτημάτων Μία Σταθερή Μεταβλητή

Παρατηρούμε ότι για πεντακόσια αιτήματα η μέση τιμή κατανάλωσης CPU από την διεργασία του DSS είναι 70,4%. Για επτακόσια αιτήματα έχουμε μία αύξηση της τάξεως του 2,3% και η

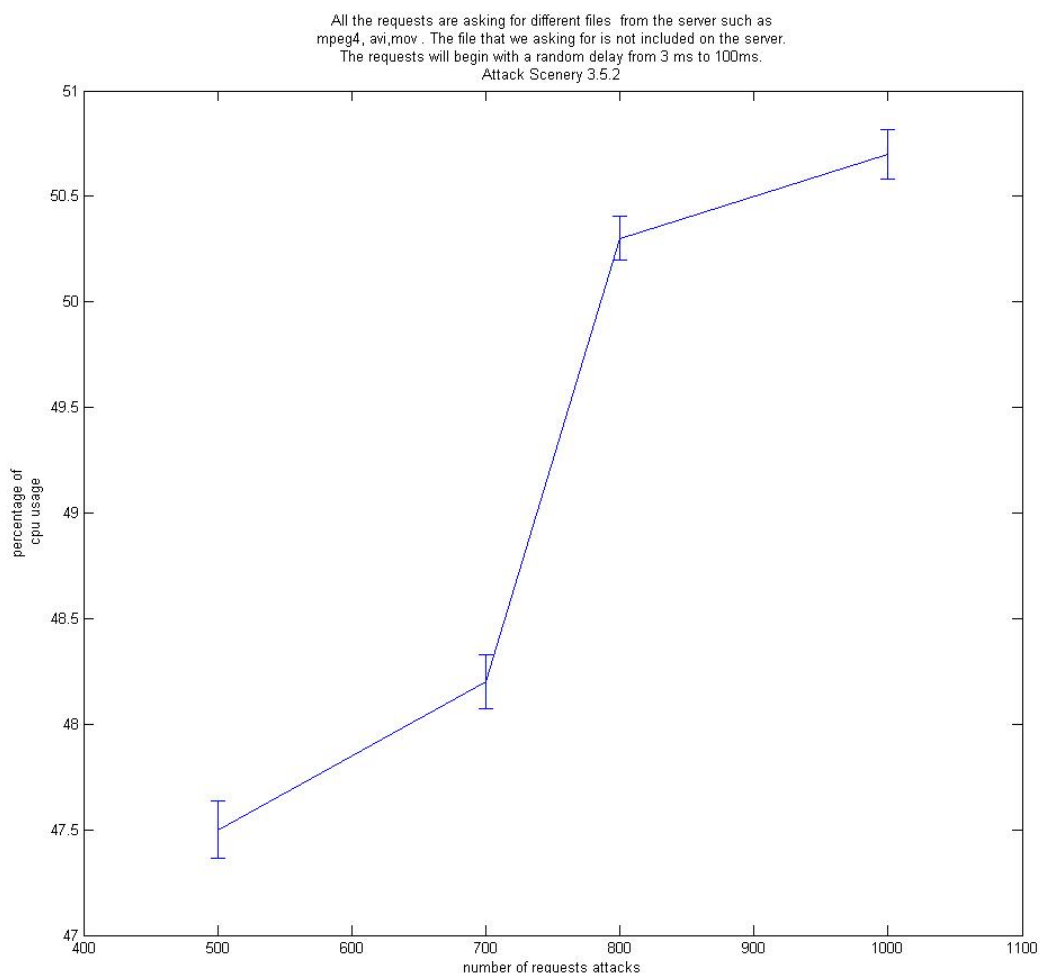
μέση τιμή ανεβαίνει στο 72,7%. Για οκτακόσια αιτήματα έχουμε μία αύξηση 0,3% και η μέση τιμή ανεβαίνει στο 73%. Για χίλια αιτήματα η μέση τιμή ανεβαίνει στο 73,4%. Παρατηρούμε μία αύξηση της μέσης τιμής κατανάλωσης του CPU όσον αυξάνονται τα αιτήματα. Ο DSS αναγκάζεται να σπαταλάει πόρους επεξεργαστική δύναμη αφού προσπαθεί να εξυπηρετήσει άσκοπα μηνύματα.

Επίσης εδώ μπορούμε να αναφέρουμε ότι σε σύγκριση με το πείραμα 3.4.1 όπου η μόνη διαφορά ότι σε αυτό το πείραμα 3.5.1 είναι ότι ζητούμε αρκετά αρχεία δεν βλέπουμε και τρομερή διαφορά. Η διαφορά στις δύο μετρήσεις εάν πάρουμε την παραλλαγή του πειράματος για χίλια αιτήματα είναι τις τάξεως του 6% αφού για το 3.4.1 ανέρχεται σε 79% και στο 3.5.1 στο 73%.

Η μετρήσεις για κατανάλωση memory ήταν αμελητέες για αυτό και δεν παρουσιάζονται για σχολιασμό.

5.1.8 Ανάλυση Αποτελεσμάτων Σεναρίου Περιγραφή Πειράματος Χρησιμοποιώντας Την Απλή Εντολή Του OpenRtsp Ζητώντας Αρχεία Που Δεν Βρίσκονται Στον Streaming Server Και Χρόνο Εκδοσης Των Αιτημάτων Μία Τυχαία Μεταβλητή

Πιο κάτω παρουσιάζουμε τα γραφικά αποτελέσματα του σεναρίου 3.5.2. Στην γραφική παράσταση αναγράφεται η μέση τιμή για την κάθε παραλλαγή του σεναρίου. Οι παραλλαγές του σεναρίου υπενθυμίζουμε είναι για πεντακόσια, επτακόσια, οκτακόσια και χίλια αιτήματα. Στο πιο κάτω διάγραμμα παρατηρούμε την αύξηση της χρήσης του CPU όσον αυξάνονται και τα αιτήματα.



Γραφική 5.12: Μέση τιμή κατανάλωσης Cpu Χρησιμοποιώντας Την Απλή Εντολή Του OpenRtsp Ζητώντας Αρχεία Που Δεν Βρίσκονται Στον Streaming Server Και Χρόνο Εκδοσης Των Αιτημάτων Μία Τυχαία Μεταβλητή

Παρατηρούμε ότι για πεντακόσια αιτήματα η μέση τιμή κατανάλωσης CPU από την διεργασία του DSS είναι 47,5%. Για επτακόσια αιτήματα έχουμε μία αύξηση της τάξεως του 2,3% και η

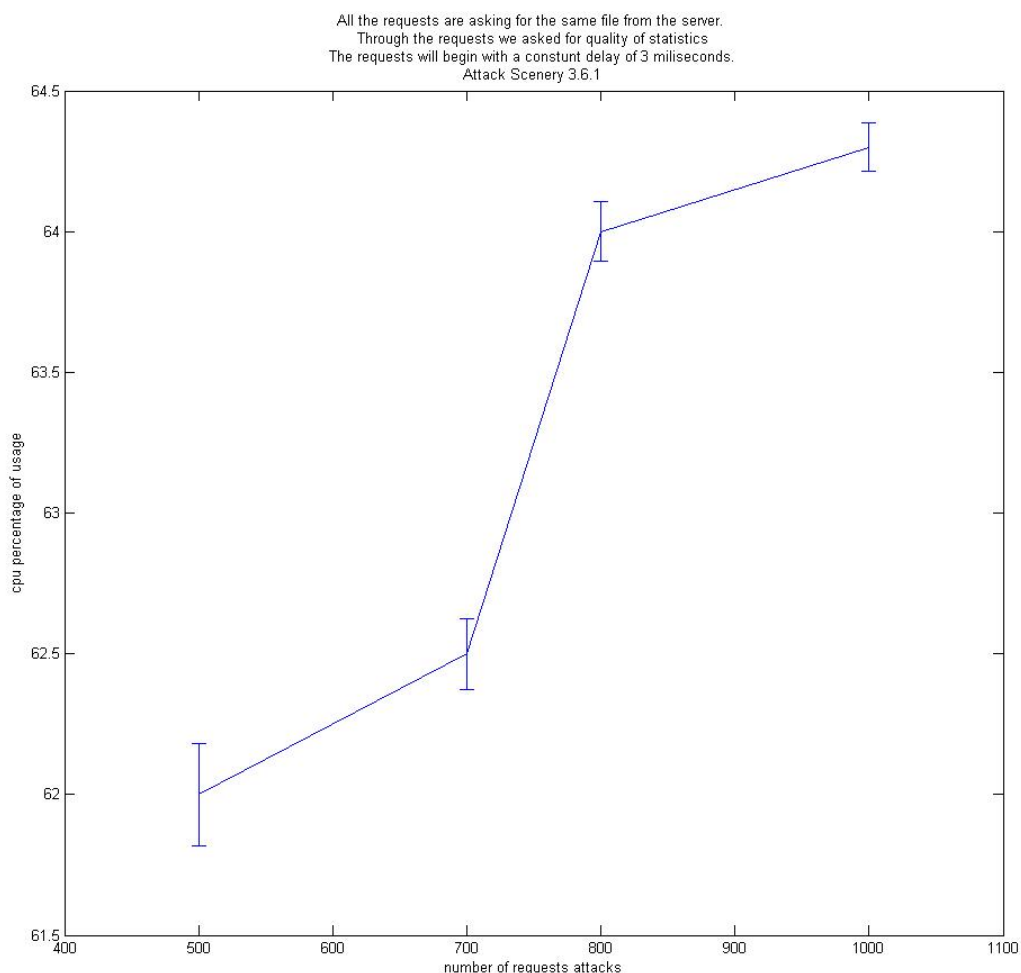
μέση τιμή ανεβαίνει στο 48,8%. Για οκτακόσια αιτήματα έχουμε μία αύξηση 1,5 % και η μέση τιμή ανεβαίνει στο 50,3 %. Για χίλια αιτήματα η μέση τιμή ανεβαίνει στο 50,8%. Παρατηρούμε μία αύξηση της μέσης τιμής κατανάλωσης του CPU όσον αυξάνονται τα αιτήματα. Ο DSS αναγκάζεται να σπαταλάει πόρους επεξεργαστική δύναμη αφού προσπαθεί να εξυπηρετήσει άσκοπα μηνύματα.

Ακόμη σε ένα άλλο παράδειγμα φαίνεται πόσο σημαντικό ρόλο παίζει η χρονική άφιξης των αιτημάτων στον διακομιστή. Εάν αναλογιστούμε το προηγούμενο παράδειγμα 3.5.1 και αυτό 3.5.2 όπου η μόνη διαφορά είναι η χρονική άφιξης των μηνυμάτων αφού στο ένα πείραμα (3.5.2) γίνεται με τυχαία επιλογή ενώ στο προηγούμενο (3.5.1) με σταθερή μεταβλητή, θα παρατηρήσουμε ότι σε όλες τις παραλλαγές των πειραμάτων η μέση τιμή του 3.5.1 είναι κατά πολύ ψιλότερη από του 3.5.2.

Η μετρήσεις για κατανάλωση memory ήταν αμελητέες για αυτό και δεν παρουσιάζονται για σχολιασμό.

5.1.9 Ανάλυση Αποτελεσμάτων Σεναρίου Χρησιμοποιώντας Την Εντολή Του OpenRtsp Για Quality Of Service Και Χρόνο Έκδοσης Των Αιτημάτων Μία Σταθερή Μεταβλητή

Πιο κάτω παρουσιάζουμε τα γραφικά αποτελέσματα του σεναρίου 3.6.1. Στην γραφική παράσταση αναγράφεται η μέση τιμή για την κάθε παραλλαγή του σεναρίου. Οι παραλλαγές του σεναρίου υπενθυμίζουμε είναι για πεντακόσια, επτακόσια, οκτακόσια και χίλια αιτήματα. Στο πιο κάτω διάγραμμα παρατηρούμε την αύξηση της χρήσης του CPU όσον αυξάνονται και τα αιτήματα.

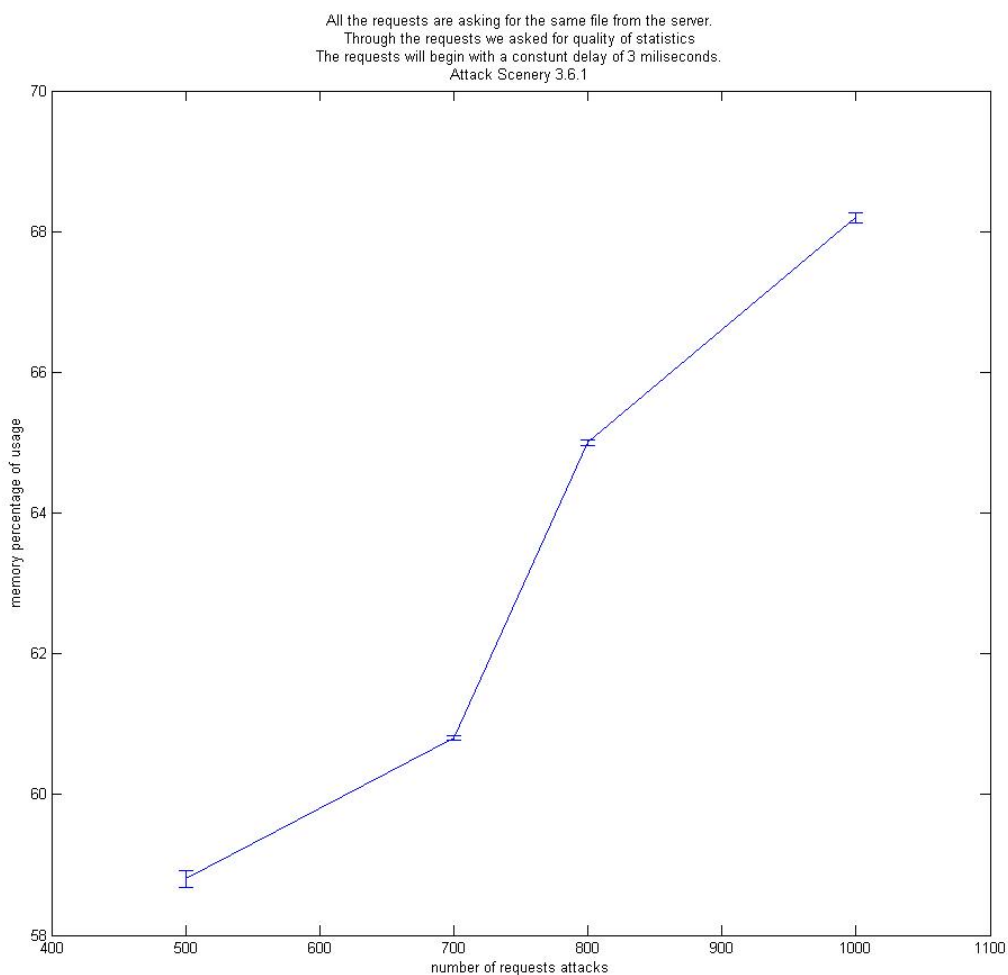


Γραφική 5.12: Μέση τιμή κατανάλωσης Cpu Χρησιμοποιώντας Την Εντολή Του OpenRtsp Για Quality Of Service Και Χρόνο Έκδοσης Των Αιτημάτων Μία Σταθερή Μεταβλητή

Παρατηρούμε ότι για πεντακόσια αιτήματα η μέση τιμή κατανάλωσης CPU από την διεργασία του DSS ανέρχεται στο 62%. Για επτακόσια αιτήματα έχουμε μία αύξηση της τάξεως του 0,5% και η μέση τιμή ανεβαίνει στο 62%. Για οκτακόσια αιτήματα έχουμε μία αύξηση 1,5% και η μέση τιμή ανεβαίνει στο 64%. Για χίλια αιτήματα η μέση τιμή ανεβαίνει στο 64,4%. Παρατηρούμε μία

λογική αύξηση της μέσης τιμής κατανάλωσης του CPU αφού ο διακομιστής μας αναγκάζεται να εξυπηρετήσει όλο και περισσότερα αιτήματα σχεδόν ταυτόχρονα. Επίσης αποστέλλει και στατιστικά ποιότητας παράδοσης των δεδομένων, όπου κάνει τον φόρτο εργασίας του διακομιστή πιο δύσκολο.

Στην επόμενη γραφική παράσταση θα παρατηρήσουμε την συμπεριφορά της διεργασίας του DSS ως προς την κατανάλωση Memory. Αναγράφεται για κάθε παραλλαγή του σεναρίου η μέση τιμή κατανάλωσης μνήμης. Παρατηρούμε μία σημαντική αύξηση στο ποσοστό κατανάλωσης της μνήμης όσον αυξάνονται τα αιτήματα.



Γραφική 5.13: Μέση τιμή κατανάλωσης Memory Χρησιμοποιώντας Την Εντολή Του OpenRtsp Για Quality Of Service Και Χρόνο Έκδοσης Των Αιτημάτων Μία Σταθερή Μεταβλητή

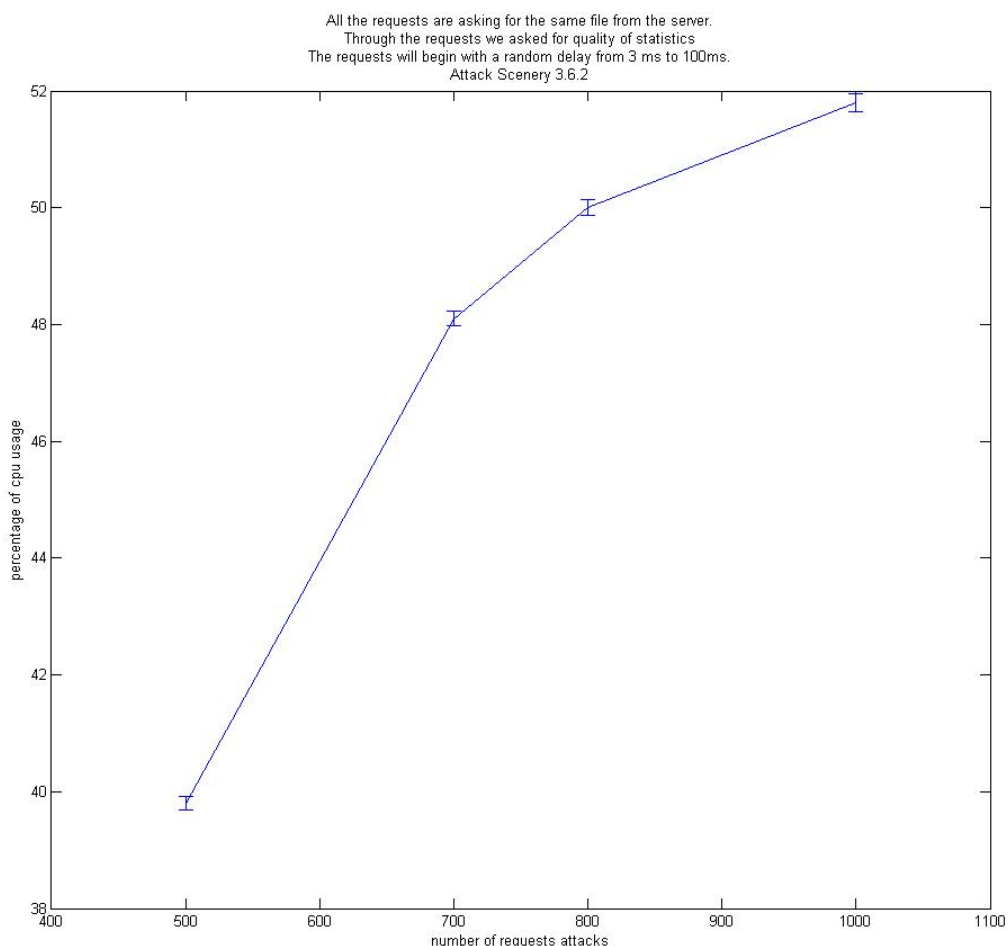
Παρατηρούμε ότι για πεντακόσια αιτήματα η μέση τιμή κατανάλωσης Memory από την διεργασία του DSS ανέρχεται στο 59%. Για επτακόσια αιτήματα έχουμε μία αύξηση της τάξεως του 2% και η μέση τιμή ανεβαίνει στο 61%. Για οκτακόσια αιτήματα έχουμε μία αύξηση 5% και η μέση τιμή ανεβαίνει στο 65%. Για χίλια αιτήματα η μέση τιμή ανεβαίνει στο 68%. Παρατηρούμε

μία λογική αύξηση της μέσης τιμής κατανάλωσης του CPU αφού ο διακομιστής μας αναγκάζεται να εξυπηρετήσει όλο και περισσότερα αιτήματα, και επιπλέον είναι και για διαφορετικά αρχεία.

Το γενικό συμπέρασμα που αποκομίζουμε από τις πιο πάνω γραφικές παραστάσεις είναι ότι η κατανάλωση cpu και memory από την διεργασία του DSS ανεβαίνει όσο ανεβαίνουν και τα ταυτόχρονα αιτήματα. Ο διακομιστής θέλοντας να εξυπηρετήσει όλα τα αιτήματα αφού αυτός είναι και ο σκοπός του, καταναλώνει όλο και περισσότερους πόρους όσο τα αιτήματα στο socket του συσσωρεύονται.

5.1.10 Ανάλυση Αποτελεσμάτων Σεναρίου Χρησιμοποιώντας Την Εντολή Του OpenRtsp Για Quality Of Service Και Χρόνο Έκδοσης Των Αιτημάτων Μία Τυχαία Μεταβλητή

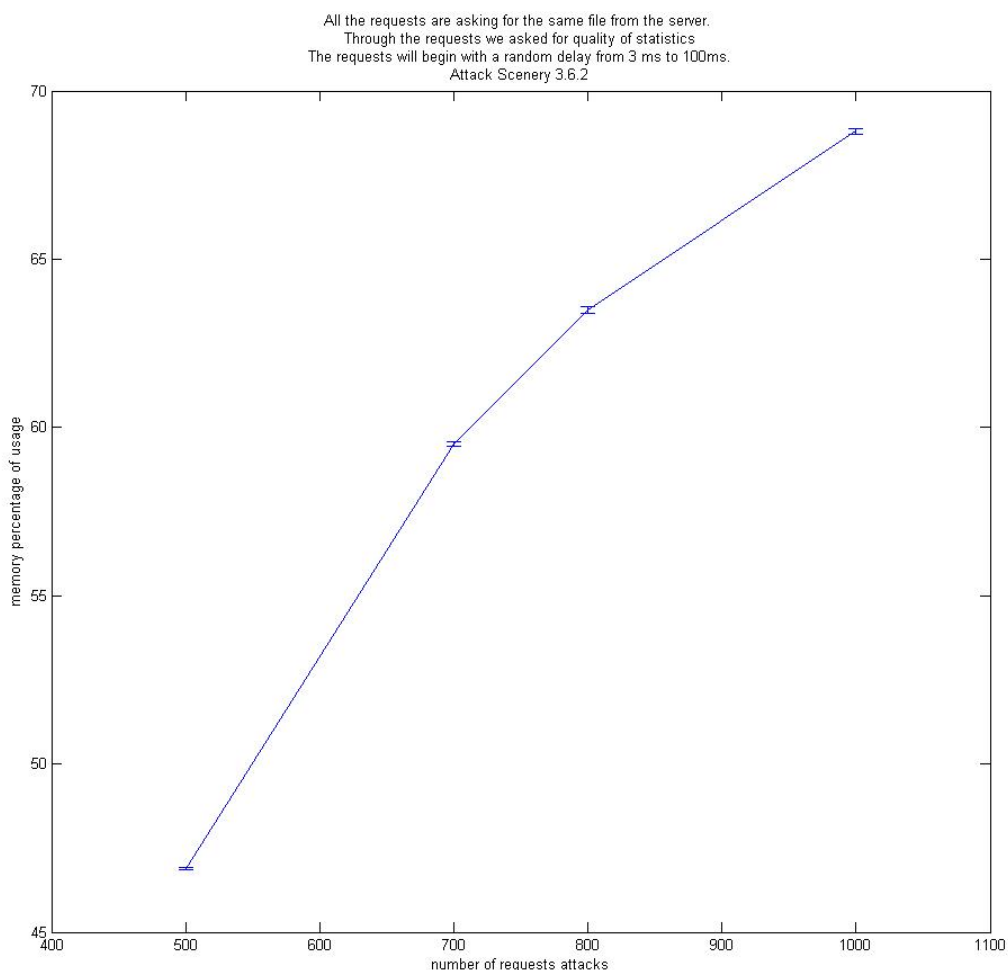
Πιο κάτω παρουσιάζουμε τα γραφικά αποτελέσματα του σεναρίου 3.6.2. Στην γραφική παράσταση αναγράφεται η μέση τιμή για την κάθε παραλλαγή του σεναρίου. Οι παραλλαγές του σεναρίου υπενθυμίζουμε είναι για πεντακόσια, επτακόσια, οκτακόσια και χίλια αιτήματα. Στο πιο κάτω διάγραμμα παρατηρούμε την αύξηση της χρήσης του CPU όσον αυξάνονται και τα αιτήματα.



Γραφική 5.14: Μέση τιμή κατανάλωσης CPU Χρησιμοποιώντας Την Εντολή Του OpenRtsp Για Quality Of Service Και Χρόνο Έκδοσης Των Αιτημάτων Μία Σταθερή Μεταβλητή

Παρατηρούμε ότι για πεντακόσια αιτήματα η μέση τιμή κατανάλωσης CPU από την διεργασία του DSS είναι 40 %. Για επτακόσια αιτήματα έχουμε μία αύξηση της τάξεως του 8 % και η μέση τιμή ανεβαίνει στο 48%. Για οκτακόσια αιτήματα έχουμε μία αύξηση 2 % και η μέση τιμή ανεβαίνει στο 50 %. Για χίλια αιτήματα η μέση τιμή ανεβαίνει στο 52%. Παρατηρούμε μία αύξηση της μέσης τιμής κατανάλωσης του CPU όσον αυξάνονται τα αιτήματα.

Στην επόμενη γραφική παράσταση θα παρατηρήσουμε την συμπεριφορά της διεργασίας του DSS ως προς την κατανάλωση Memory. Αναγράφεται για κάθε παραλλαγή του σεναρίου η μέση τιμή κατανάλωσης μνήμης. Παρατηρούμε μία σημαντική αύξηση στο ποσοστό κατανάλωσης της μνήμης όσον αυξάνονται τα αιτήματα.



Γραφική 5.15: Μέση τιμή κατανάλωσης Memory Χρησιμοποιώντας Την Εντολή Του OpenRtsp Για Quality Of Service Και Χρόνο Έκδοσης Των Αιτημάτων Μία Σταθερή Μεταβλητή

Παρατηρούμε ότι για πεντακόσια αιτήματα η μέση τιμή κατανάλωσης Memory από την διεργασία του DSS ανέρχεται στο 46%. Για επτακόσια αιτήματα έχουμε μία αύξηση της τάξεως του 13,8% και η μέση τιμή ανεβαίνει στο 59,4%. Για οκτακόσια αιτήματα έχουμε μία αύξηση 4,6% και η μέση τιμή ανεβαίνει στο 64%. Για χίλια αιτήματα η μέση τιμή ανεβαίνει στο 69%.

Παρατηρούμε μία λογική αύξηση της μέσης τιμής κατανάλωσης του CPU αφού ο διακομιστής μας αναγκάζεται να εξυπηρετήσει όλο και περισσότερα αιτήματα, και επιπλέον είναι και για διαφορετικά αρχεία.

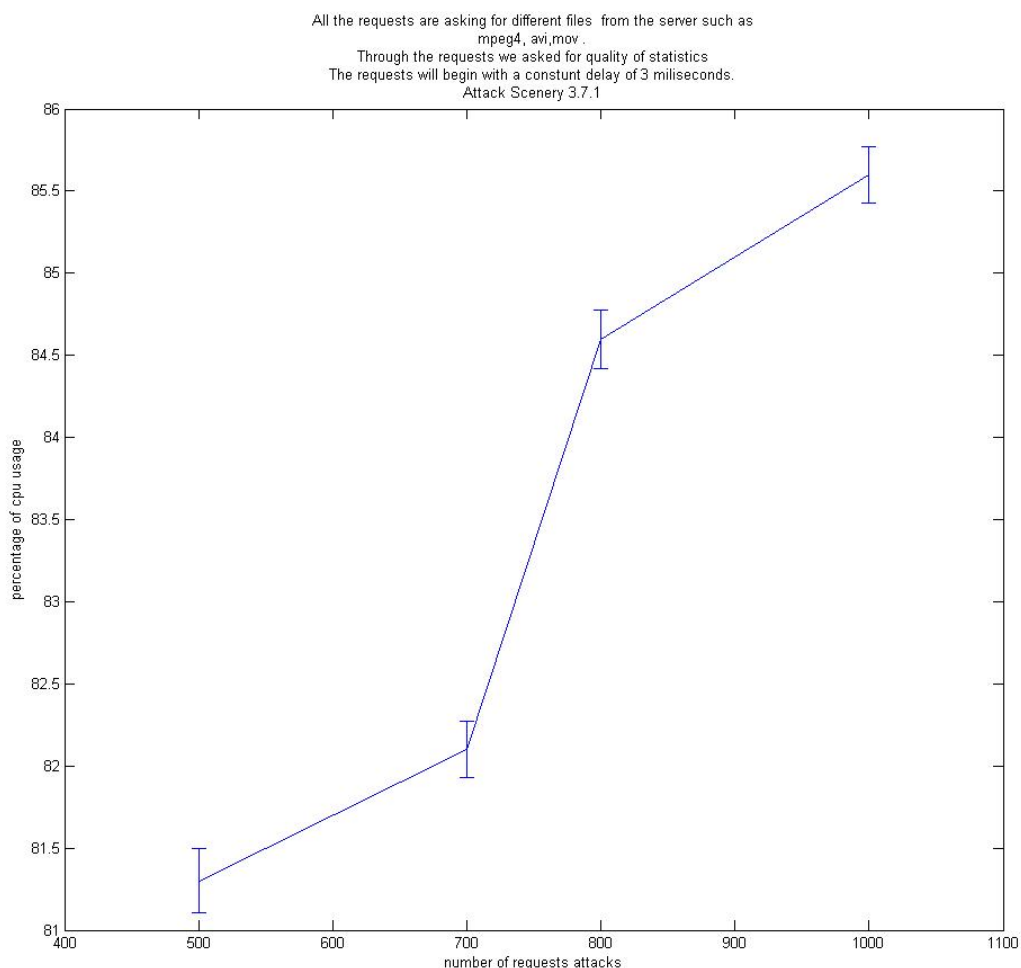
Το γενικό συμπέρασμα που αποκομίζουμε από τις πιο πάνω γραφικές παραστάσεις είναι ότι η κατανάλωση cpu και memory από την διεργασία του DSS ανεβαίνει όσο ανεβαίνουν και τα ταυτόχρονα αιτήματα. Ο διακομιστής θέλοντας να εξυπηρετήσει όλα τα αιτήματα αφού αυτός είναι και ο σκοπός του, καταναλώνει όλο και περισσότερους πόρους όσο τα αιτήματα στο socket του συσσωρεύονται.

Παράλληλα εάν κοιτάξουμε το πείραμα 3.6.1 η μόνη διαφορά είναι η χρονική άφιξης των αιτημάτων. Θα συμπεράνουμε ότι ο κανόνας που δημιουργήθηκε ήδη και από την ανάλυση των προηγούμενων πειραμάτων ισχύει και σε αυτά τα δύο, δηλαδή το πείραμα 3.6.1 έχει πολύ ψιλότερες μετρήσεις όσον αφορά την κατανάλωση cpu από την διεργασία.

Τώρα όσον αφορά την κατανάλωση μνήμης αυτό που παρατηρούμε για πρώτη φορά είναι ότι οι τιμές στις μετρήσεις των δύο πειραμάτων είναι πολύ κοντά. Να υπενθυμίσουμε ότι για χίλια αιτήματα το 3.6.1 ανέρχεται στο 68,3% και το 3.6.2 στο 69%.

5.1.11 Ανάλυση Αποτελεσμάτων Σεναρίου Χρησιμοποιώντας Την Εντολή Του OpenRtsp Για Quality Of Service Ζητώντας Διαφορετικά Αρχεία Και Χρόνο Έκδοσης Των Αιτημάτων Μία Σταθερή Μεταβλητή

Πιο κάτω παρουσιάζουμε τα γραφικά αποτελέσματα του σεναρίου 3.7.1. Στην γραφική παράσταση αναγράφεται η μέση τιμή για την κάθε παραλλαγή του σεναρίου. Οι παραλλαγές του σεναρίου υπενθυμίζουμε είναι για πεντακόσια, επτακόσια, οκτακόσια και χίλια αιτήματα. Στο πιο κάτω διάγραμμα παρατηρούμε την αύξηση της χρήσης του CPU όσον αυξάνονται και τα αιτήματα.



Γραφική 5.16: Μέση τιμή κατανάλωσης CPU Χρησιμοποιώντας Την Εντολή Του OpenRtsp Για Quality Of Service Ζητώντας Διαφορετικά Αρχεία Και Χρόνο Έκδοσης Των Αιτημάτων Μία Σταθερή Μεταβλητή

Παρατηρούμε ότι για πεντακόσια αιτήματα η μέση τιμή κατανάλωσης CPU από την διεργασία του DSS είναι 81,3 %. Για επτακόσια αιτήματα έχουμε μία αύξηση της τάξεως του 0,7 % και η μέση τιμή ανεβαίνει στο 82%. Για οκτακόσια αιτήματα έχουμε μία αύξηση 2,5 % και η μέση τιμή

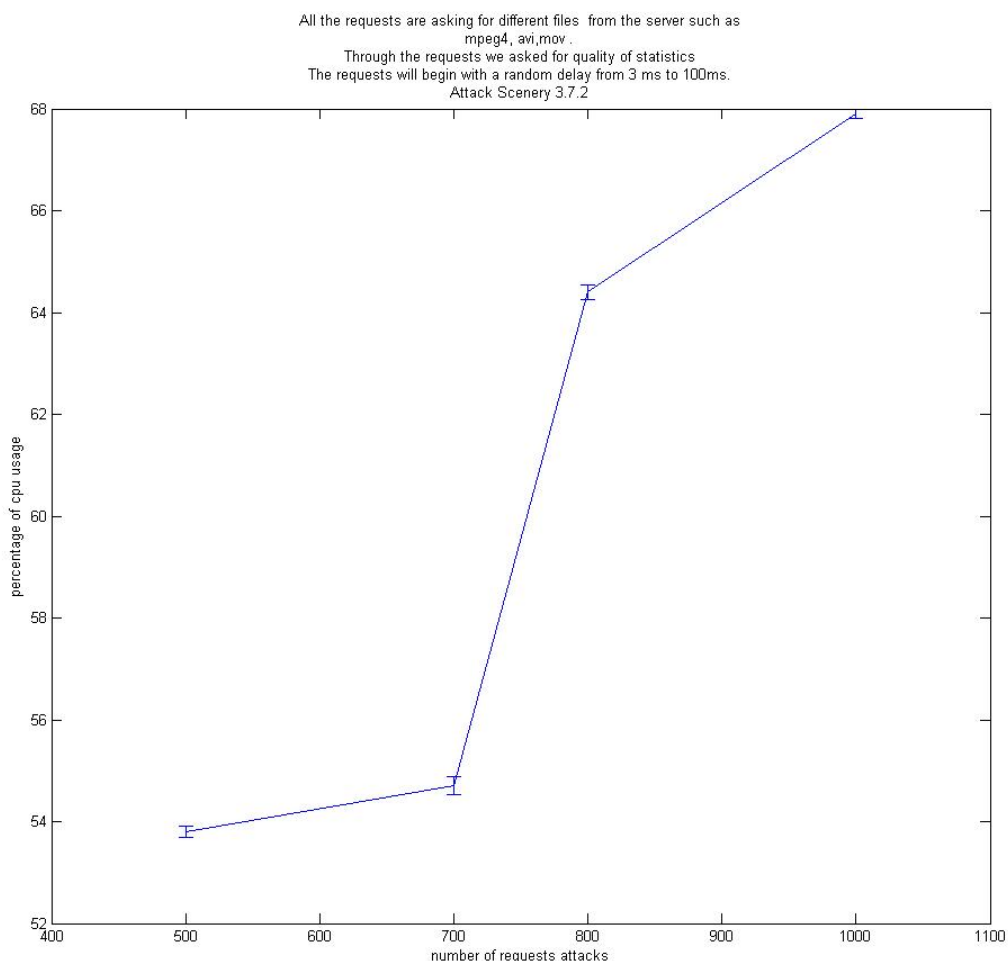
ανεβαίνει στο 84,5 %. Για χίλια αιτήματα η μέση τιμή ανεβαίνει στο 85,5%. Παρατηρούμε μία αύξηση της μέσης τιμής κατανάλωσης του CPU όσον αυξάνονται τα αιτήματα. Είναι η πρώτη φορά που συναντούμε τέτοια χρησιμοποίηση του Cpu από την διεργασία του DSS κορυφαία απόδοση να ανέρχεται στο 85,5% όταν τα αιτήματα είναι χίλια.

Επιπλέον και εδώ παρατηρούμε ότι ένας κανόνας που δημιουργήθηκε και προηγούμενος εφαρμόζεται και πάλι αφού εάν συγκρίνουμε το πείραμα 3.6.1 και 3.7.1 η μόνη διαφορά τους είναι η διαφορετικότητα των αρχείων που ζητούνται, όταν ζητούμε διαφορετικά αρχεία τότε η κατανάλωση πόρων cpu αυξάνετε.

Η μετρήσεις για κατανάλωση memory ήταν αμελητέες για αυτό και δεν παρουσιάζονται για σχολιασμό.

5.1.12 Ανάλυση Αποτελεσμάτων Σεναρίου Την Εντολή Του OpenRtsp Για Quality Of Service Ζητώντας Διαφορετικά Αρχεία Και Χρόνο Έκδοσης Των Αιτημάτων Μία Τυχαία Μεταβλητή

Πιο κάτω παρουσιάζουμε τα γραφικά αποτελέσματα του σεναρίου 3.7.2. Στην γραφική παράσταση αναγράφεται η μέση τιμή για την κάθε παραλλαγή του σεναρίου. Οι παραλλαγές του σεναρίου υπενθυμίζουμε είναι για πεντακόσια, επτακόσια, οκτακόσια και χίλια αιτήματα. Στο πιο κάτω διάγραμμα παρατηρούμε την αύξηση της χρήσης του CPU όσον αυξάνονται και τα αιτήματα.



Γραφική 5.17: Μέση τιμή κατανάλωσης CPU Χρησιμοποιώντας Την Εντολή Του OpenRtsp Για Quality Of Service Ζητώντας Διαφορετικά Αρχεία Και Χρόνο Έκδοσης Των Αιτημάτων Μία Τυχαία Μεταβλητή

Παρατηρούμε ότι για πεντακόσια αιτήματα η μέση τιμή κατανάλωσης CPU από την διεργασία του DSS είναι 54 %. Για επτακόσια αιτήματα έχουμε μία αύξηση της τάξεως του 1 % και η μέση τιμή ανεβαίνει στο 51%. Για οκτακόσια αιτήματα έχουμε μία αύξηση 13% και η μέση τιμή

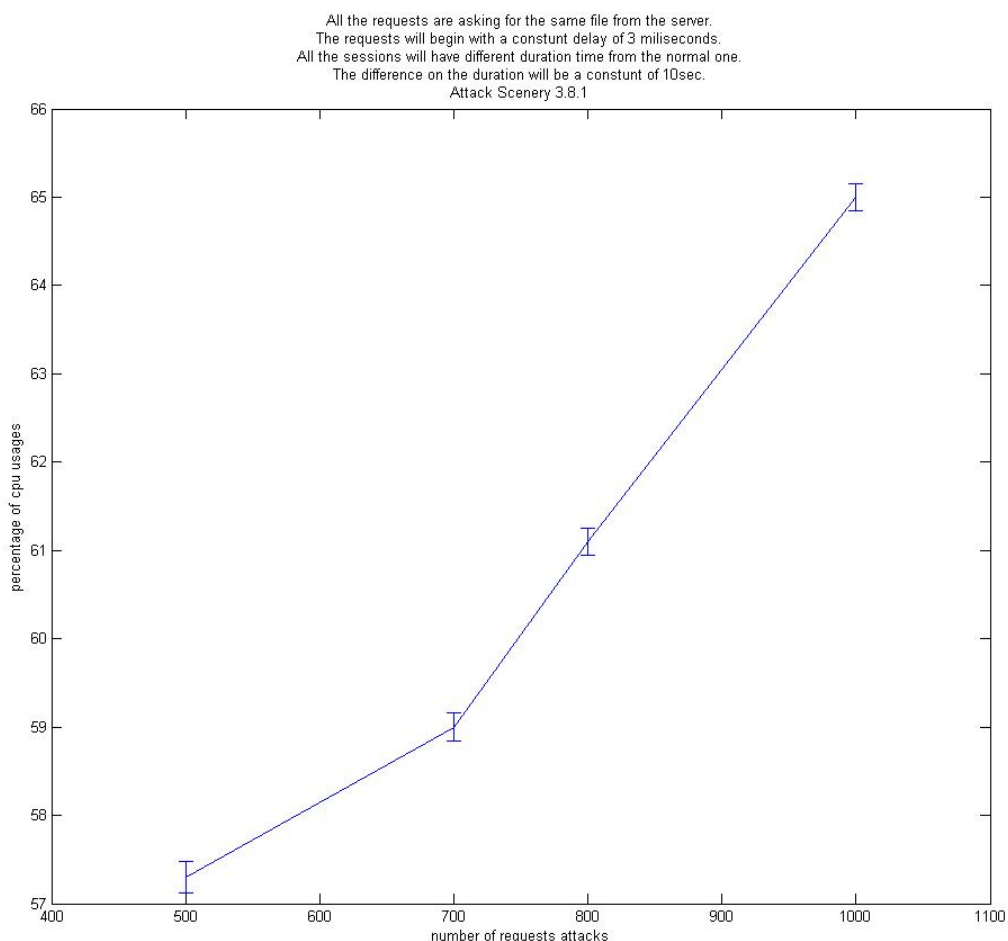
ανεβαίνει στο 64%. Για χίλια αιτήματα η μέση τιμή ανεβαίνει στο 68%. Παρατηρούμε μία αύξηση της μέσης τιμής κατανάλωσης του CPU όσον αυξάνονται τα αιτήματα.

Ακολούθως διαπιστώνουμε ξανά ότι ο κανόνας με την χρονική άφιξη των αιτημάτων ισχύει και εδώ αφού εάν συγκρίνουμε το πείραμα 3.7.1 και 3.7.2 όπου η μόνη διαφορά που έχουν είναι η χρονική άφιξης των αιτημάτων θα διαπιστώσουμε ότι στο 3.7.1 όπου τα αιτήματα καταφθάνουν στον διακομιστή σχεδόν ταυτόχρονα έχουμε περισσότερη κατανάλωση CPU

Η μετρήσεις για κατανάλωση memory ήταν αμελητέες για αυτό και δεν παρουσιάζονται για σχολιασμό.

5.1.13 Ανάλυση Αποτελεσμάτων Σεναρίου Χρησιμοποιώντας Την Εντολή Του OpenRtsp Για Αλλαγή στην Διάρκεια Αναπαραγωγής Του Αρχείου Και Χρόνο Έκδοσης Των Αιτημάτων Μία Σταθερή Μεταβλητή

Πιο κάτω παρουσιάζουμε τα γραφικά αποτελέσματα του σεναρίου 3.8.1. Στην γραφική παράσταση αναγράφεται η μέση τιμή για την κάθε παραλλαγή του σεναρίου. Οι παραλλαγές του σεναρίου υπενθυμίζουμε είναι για πεντακόσια, επτακόσια, οκτακόσια και χίλια αιτήματα. Στο πιο κάτω διάγραμμα παρατηρούμε την αύξηση της χρήσης του CPU όσον αυξάνονται και τα αιτήματα.

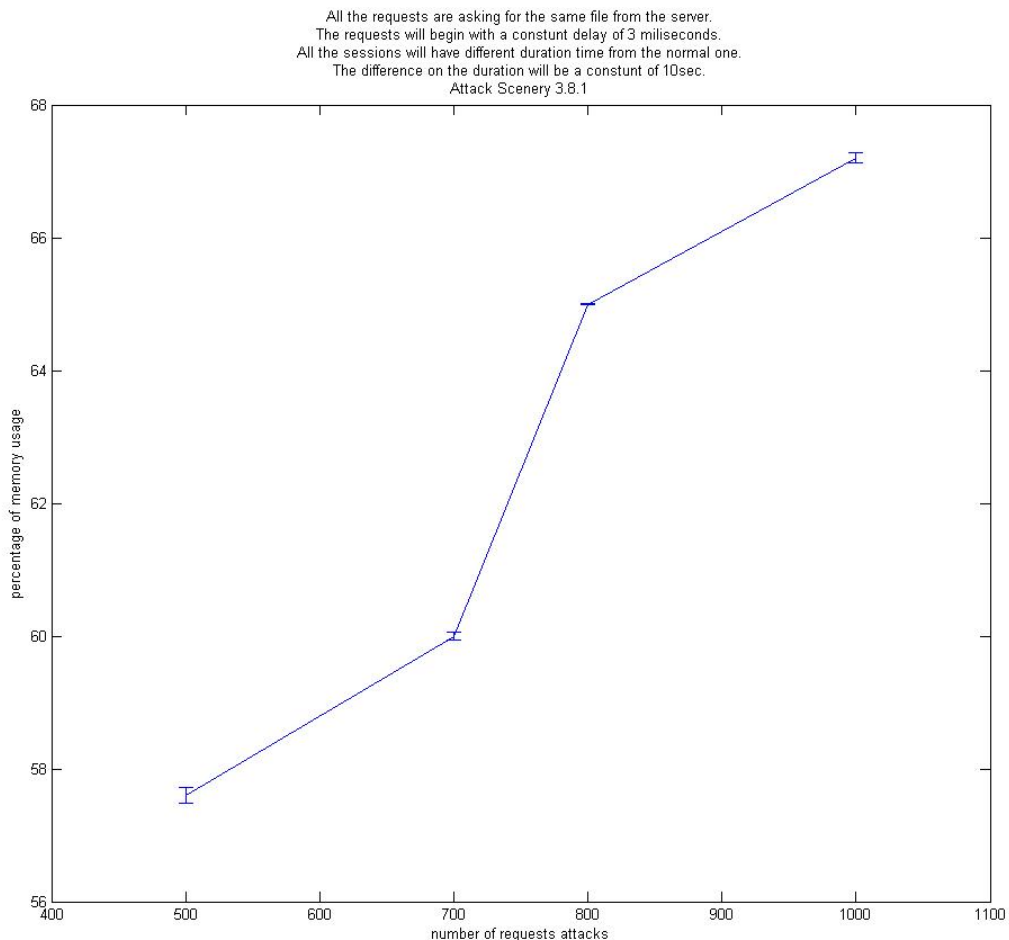


Γραφική 5.18: Μέση τιμή κατανάλωσης CPU Χρησιμοποιώντας Την Εντολή Του OpenRtsp Για Quality Of Service Ζητώντας Διαφορετικά Αρχεία Και Χρόνο Έκδοσης Των Αιτημάτων Μία Τυχαία Μεταβλητή

Παρατηρούμε ότι για πεντακόσια αιτήματα η μέση τιμή κατανάλωσης CPU από την διεργασία του DSS είναι 57,3 %. Για επτακόσια αιτήματα έχουμε μία αύξηση της τάξεως του 1,7 % και η μέση τιμή ανεβαίνει στο 59%. Για οκτακόσια αιτήματα έχουμε μία αύξηση 2 % και η μέση τιμή

ανεβαίνει στο 61 %. Για χίλια αιτήματα η μέση τιμή ανεβαίνει στο 65%. Παρατηρούμε μία αύξηση της μέσης τιμής κατανάλωσης του CPU όσον αυξάνονται τα αιτήματα.

Στην επόμενη γραφική παράσταση θα παρατηρήσουμε την συμπεριφορά της διεργασίας του DSS ως προς την κατανάλωση Memory. Αναγράφεται για κάθε παραλλαγή του σεναρίου η μέση τιμή κατανάλωσης μνήμης. Παρατηρούμε μία σημαντική αύξηση στο ποσοστό κατανάλωσης της μνήμης όσον αυξάνονται τα αιτήματα.



Γραφική 5.19: Μέση τιμή κατανάλωσης Memory Χρησιμοποιώντας Την Εντολή Του OpenRtsp Για Quality Of Service Ζητώντας Διαφορετικά Αρχεία Και Χρόνο Έκδοσης Των Αιτημάτων Μία Τυχαία Μεταβλητή

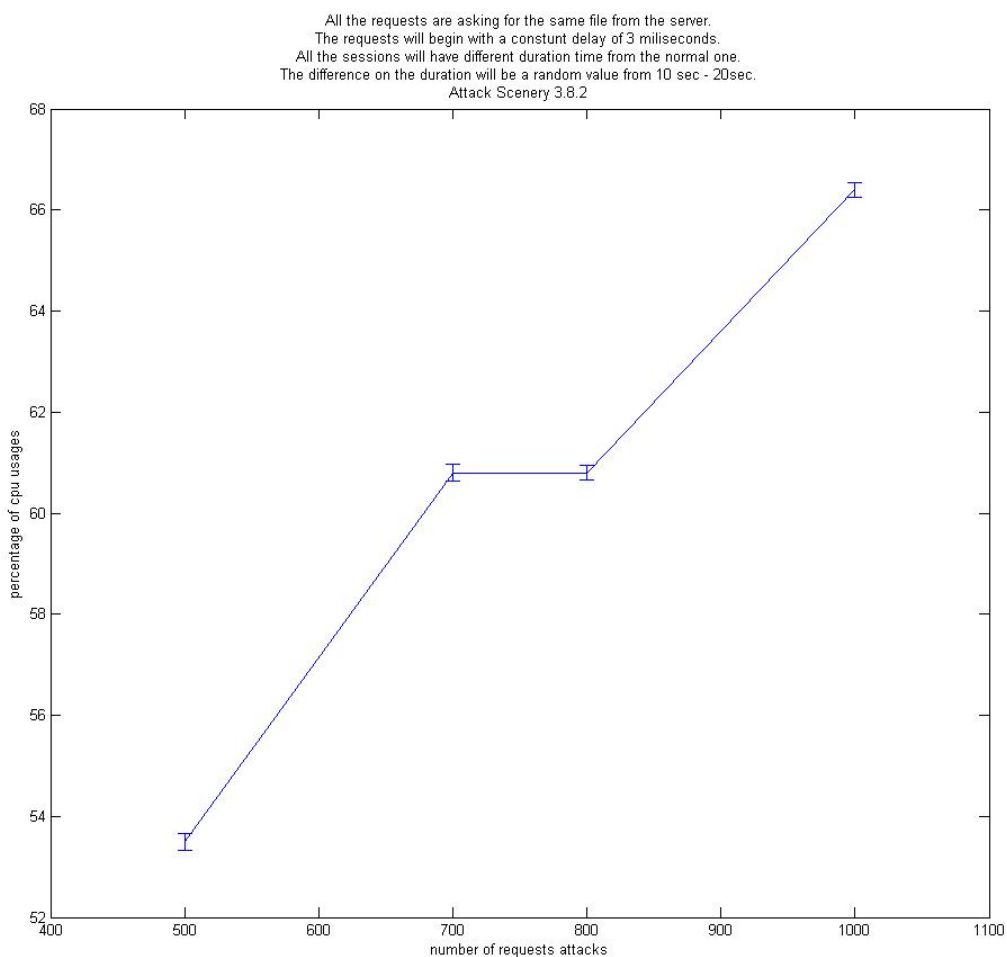
Παρατηρούμε ότι για πεντακόσια αιτήματα η μέση τιμή κατανάλωσης Memory από την διεργασία του DSS ανέρχεται στο 57,5%. Για επτακόσια αιτήματα έχουμε μία αύξηση της τάξεως του 2,5% και η μέση τιμή ανεβαίνει στο 60%. Για οκτακόσια αιτήματα έχουμε μία αύξηση 5% και η μέση τιμή ανεβαίνει στο 65%. Για χίλια αιτήματα η μέση τιμή ανεβαίνει στο 67%.

Παρατηρούμε μία λογική αύξηση της μέσης τιμής κατανάλωσης του CPU αφού ο διακομιστής μας αναγκάζεται να εξυπηρετήσει όλο και περισσότερα αιτήματα, και επιπλέον οι σύνοδοι καθυστερούν να κλείσουν κατά 10 δεύτερα.

Το γενικό συμπέρασμα που αποκομίζουμε από τις πιο πάνω γραφικές παραστάσεις είναι ότι η κατανάλωση cpu και memory από την διεργασία του DSS ανεβαίνει όσο ανεβαίνουν και τα ταυτόχρονα αιτήματα. Ο διακομιστής θέλοντας να εξυπηρετήσει όλα τα αιτήματα αφού αυτός είναι και ο σκοπός του, καταναλώνει όλο και περισσότερους πόρους όσο τα αιτήματα στο socket του συσσωρεύονται.

5.1.14 Ανάλυση Αποτελεσμάτων Σεναρίου Χρησιμοποιώντας Την Εντολή Του OpenRtsp Για Αλλαγή στην Διάρκεια Αναπαραγωγής Του Αρχείου Και Χρόνο Έκδοσης Των Αιτημάτων Μία Τυχαία Μεταβλητή

Πιο κάτω παρουσιάζουμε τα γραφικά αποτελέσματα του σεναρίου 3.8.2. Στην γραφική παράσταση αναγράφεται η μέση τιμή για την κάθε παραλλαγή του σεναρίου. Οι παραλλαγές του σεναρίου υπενθυμίζουμε είναι για πεντακόσια, επτακόσια, οκτακόσια και χίλια αιτήματα. Στο πιο κάτω διάγραμμα παρατηρούμε την αύξηση της χρήσης του CPU όσον αυξάνονται και τα αιτήματα.

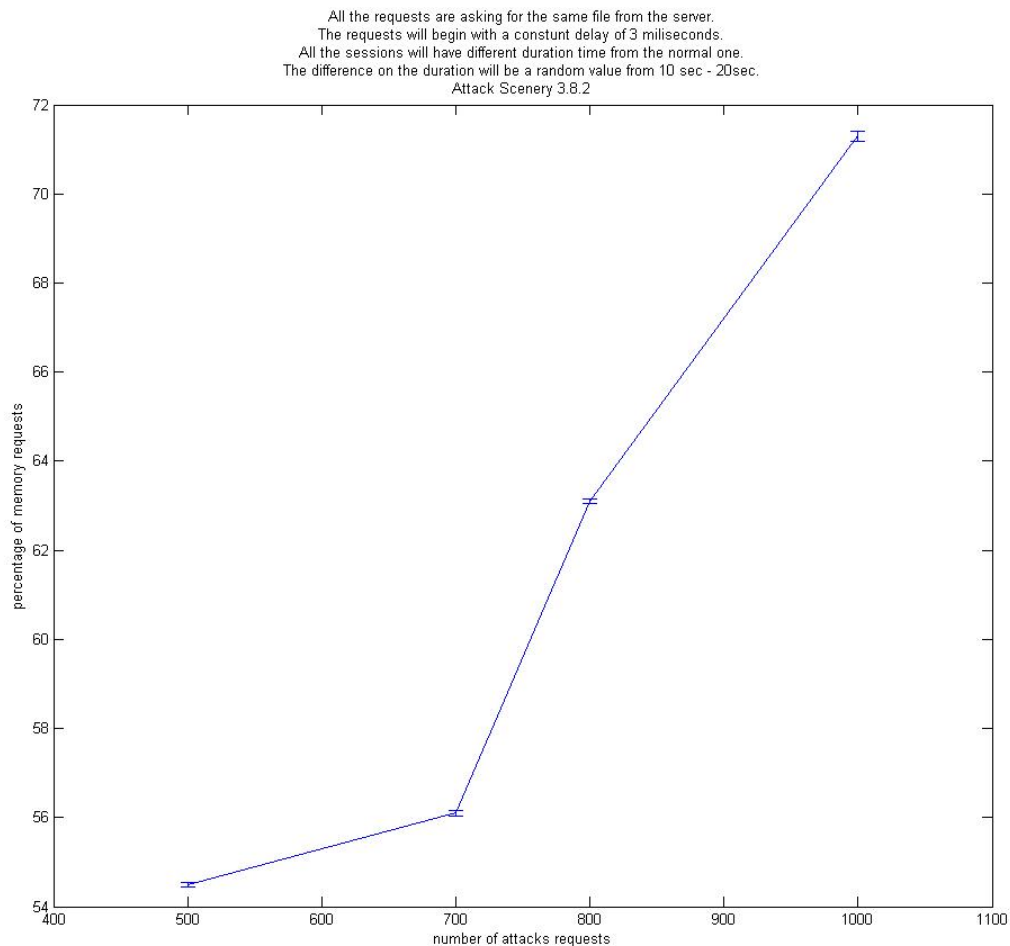


Γραφική 5.20: Μέση τιμή κατανάλωσης CPU Χρησιμοποιώντας Την Εντολή Του OpenRtsp Για Quality Of Service Ζητώντας Διαφορετικά Αρχεία Και Χρόνο Έκδοσης Των Αιτημάτων Μία Τυχαία Μεταβλητή

Παρατηρούμε ότι για πεντακόσια αιτήματα η μέση τιμή κατανάλωσης CPU από την διεργασία του DSS είναι 53,8 %. Για επτακόσια αιτήματα έχουμε μία αύξηση της τάξεως του 6,2 % και η μέση τιμή ανεβαίνει στο 60%. Για οκτακόσια αιτήματα δεν έχουμε καμία αύξηση και η μέση τιμή

διατηρείτε στο 60 %. Για χίλια αιτήματα η μέση τιμή ανεβαίνει στο 66%. Παρατηρούμε μία αύξηση της μέσης τιμής κατανάλωσης του CPU όσον αυξάνονται τα αιτήματα.

Στην επόμενη γραφική παράσταση θα παρατηρήσουμε την συμπεριφορά της διεργασίας του DSS ως προς την κατανάλωση Memory. Αναγράφεται για κάθε παραλλαγή του σεναρίου η μέση τιμή κατανάλωσης μνήμης. Παρατηρούμε μία σημαντική αύξηση στο ποσοστό κατανάλωσης της μνήμης όσον αυξάνονται τα αιτήματα.



Γραφική 5.21: Μέση τιμή κατανάλωσης memory Χρησιμοποιώντας Την Εντολή Του OpenRtsp Για Quality Of Service Ζητώντας Διαφορετικά Αρχεία Και Χρόνο Έκδοσης Των Αιτημάτων Μία Τυχαία Μεταβλητή

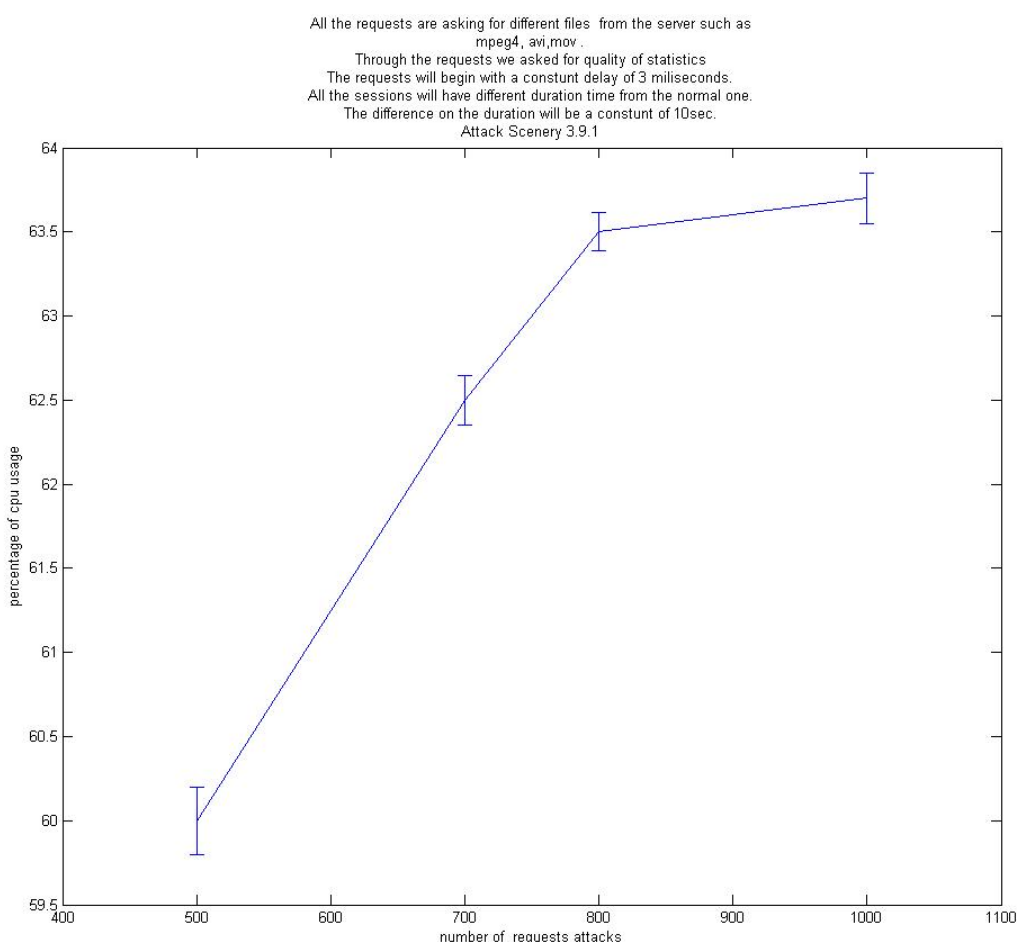
Παρατηρούμε ότι για πεντακόσια αιτήματα η μέση τιμή κατανάλωσης Memory από την διεργασία του DSS ανέρχεται στο 54,3%. Για επτακόσια αιτήματα έχουμε μία αύξηση της τάξεως του 1,7% και η μέση τιμή ανεβαίνει στο 56%. Για οκτακόσια αιτήματα έχουμε μία αύξηση 7% και η μέση τιμή ανεβαίνει στο 63%. Για χίλια αιτήματα η μέση τιμή ανεβαίνει στο 71%.

Παρατηρούμε μία λογική αύξηση της μέσης τιμής κατανάλωσης του CPU και Memory αφού ο διακομιστής μας αναγκάζεται να εξυπηρετήσει όλο και περισσότερα αιτήματα, και επιπλέον οι σύνοδοι καθυστερούν να κλείσουν κατά 10 δεύτερα.

Το γενικό συμπέρασμα που αποκομίζουμε από τις πιο πάνω γραφικές παραστάσεις είναι ότι η κατανάλωση cpu και memory από την διεργασία του DSS ανεβαίνει όσο ανεβαίνουν και ο αριθμός των αιτημάτων. Ο διακομιστής θέλοντας να εξυπηρετήσει όλα τα αιτήματα αφού αυτός είναι και ο σκοπός του, καταναλώνει όλο και περισσότερους πόρους όσο τα αιτήματα στο socket του συσσωρεύονται. Παρόλα αυτά εάν συγκρίνουμε το πείραμα 3.8.1 και 3.8.2 τα οποία διαφέρουν μόνο στην χρονική άφιξη των μηνυμάτων θα διαπιστώσουμε ότι τα επίπεδα αύξησης του CPU φτάνει στα ίδια επίπεδα αφού είναι στο 65% για το 3.8.1 και 66% για το 3.8.2. Επιπλέον και στην κατανάλωση μνήμη τα επίπεδα αύξησης δεν διαφέρουν κατά πολύ αφού στο 3.8.1 ανέρχεται στο 67% και για το 3.8.2 ανέρχεται στο 71%.

5.1.15 Ανάλυση Αποτελεσμάτων Σεναρίου Χρησιμοποιώντας Την Εντολή Του OpenRtsp Για Αλλαγή στην Διάρκεια Αναπαραγωγής Του Αρχείου Και Quality Of Service Και Χρόνο Έκδοσης Των Αιτημάτων Μία Σταθερή Μεταβλητή

Πιο κάτω παρουσιάζουμε τα γραφικά αποτελέσματα του σεναρίου 3.9.1. Στην γραφική παράσταση αναγράφεται η μέση τιμή για την κάθε παραλλαγή του σεναρίου. Οι παραλλαγές του σεναρίου υπενθυμίζουμε είναι για πεντακόσια, επτακόσια, οκτακόσια και χίλια αιτήματα. Στο πιο κάτω διάγραμμα παρατηρούμε την αύξηση της χρήσης του CPU όσον αυξάνονται και τα αιτήματα.

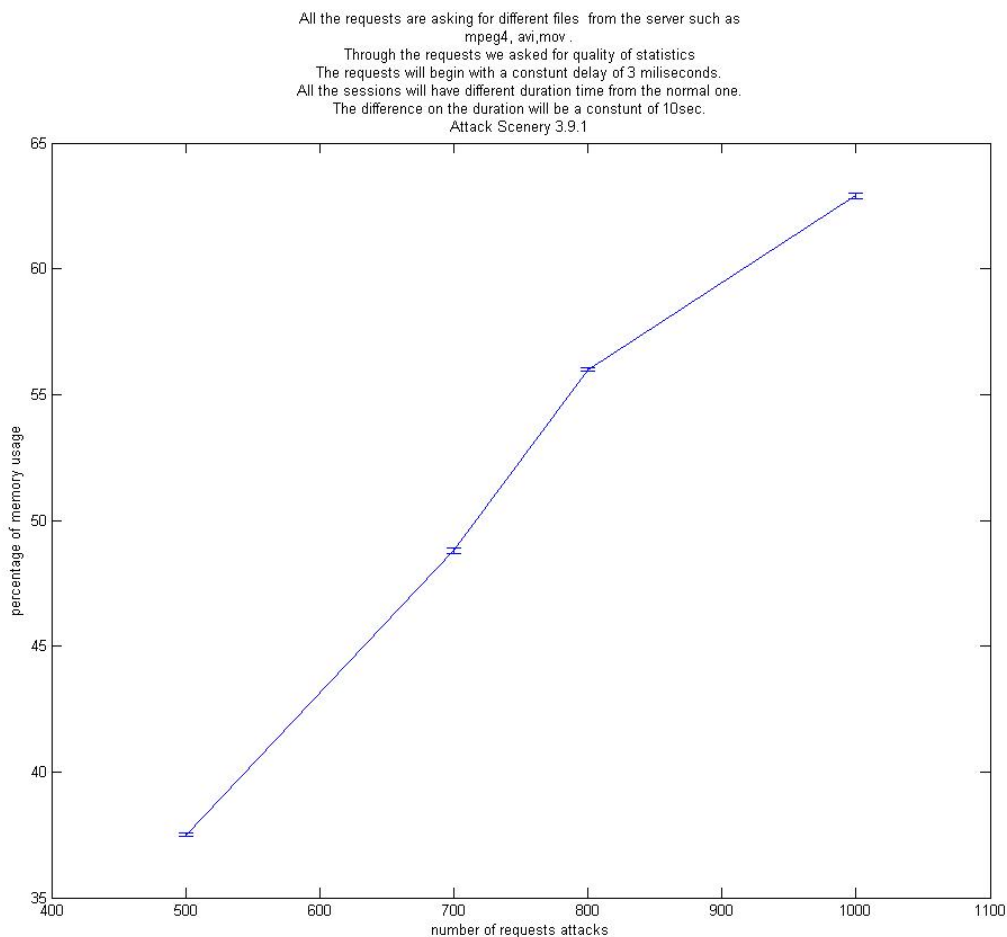


Γραφική 5.22: Μέση τιμή κατανάλωσης CPU Χρησιμοποιώντας Την Εντολή Του OpenRtsp Για Quality Of Service Ζητώντας Διαφορετικά Αρχεία Και Χρόνο Έκδοσης Των Αιτημάτων Μία Τυχαία Μεταβλητή

Παρατηρούμε ότι για πεντακόσια αιτήματα η μέση τιμή κατανάλωσης CPU από την διεργασία του DSS είναι 60%. Για επτακόσια αιτήματα έχουμε μία αύξηση της τάξεως του 2,5 % και η μέση τιμή ανεβαίνει στο 62,5%. Για οκτακόσια αιτήματα η αύξηση ανέρχεται 1% στο και η μέση τιμή

ανεβαίνει στο 63%. Για χίλια αιτήματα η μέση τιμή ανεβαίνει στο 63,8%. Παρατηρούμε μία αύξηση της μέσης τιμής κατανάλωσης του CPU όσον αυξάνονται τα αιτήματα.

Στην επόμενη γραφική παράσταση θα παρατηρήσουμε την συμπεριφορά της διεργασίας του DSS ως προς την κατανάλωση Memory. Αναγράφεται για κάθε παραλλαγή του σεναρίου η μέση τιμή κατανάλωσης μνήμης. Παρατηρούμε μία σημαντική αύξηση στο ποσοστό κατανάλωσης της μνήμης όσον αυξάνονται τα αιτήματα.



Γραφική 5.23: Μέση τιμή κατανάλωσης memory Χρησιμοποιώντας Την Εντολή Του OpenRtsp Για Quality Of Service Ζητώντας Διαφορετικά Αρχεία Και Χρόνο Έκδοσης Των Αιτημάτων Μία Τυχαία Μεταβλητή

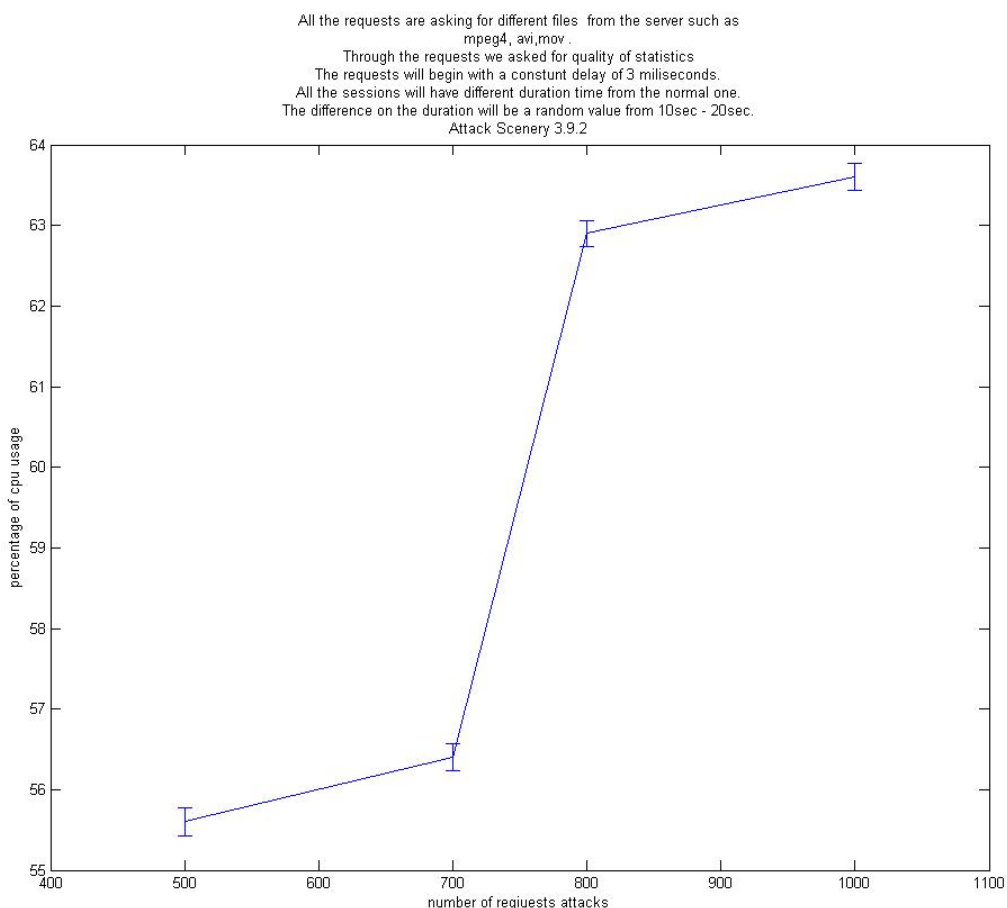
Παρατηρούμε ότι για πεντακόσια αιτήματα η μέση τιμή κατανάλωσης Memory από την διεργασία του DSS ανέρχεται στο 37%. Για επτακόσια αιτήματα έχουμε μία αύξηση της τάξεως του 10% και η μέση τιμή ανεβαίνει στο 47%. Για οκτακόσια αιτήματα έχουμε μία αύξηση 9% και η μέση τιμή ανεβαίνει στο 56%. Για χίλια αιτήματα η μέση τιμή ανεβαίνει στο 65%.

Παρατηρούμε μία λογική αύξηση της μέσης τιμής κατανάλωσης του CPU και Memory αφού ο διακομιστής μας αναγκάζεται να εξυπηρετήσει όλο και περισσότερα αιτήματα, και επιπλέον οι σύνοδοι καθυστερούν να κλείσουν κατά 10 δεύτερα και ταυτόχρονα πρέπει να αποστέλλει και στατιστικά ποιότητας παράδοσης των δεδομένων.

Το γενικό συμπέρασμα που αποκομίζουμε από τις πιο πάνω γραφικές παραστάσεις είναι ότι η κατανάλωση cpu και memory από την διεργασία του DSS ανεβαίνει όσο ανεβαίνουν και ο αριθμός των αιτημάτων. Ο διακομιστής θέλοντας να εξυπηρετήσει όλα τα αιτήματα αφού αυτός είναι και ο σκοπός του, καταναλώνει όλο και περισσότερους πόρους όσο τα αιτήματα στο socket του συσσωρεύονται.

5.1.16 Ανάλυση Αποτελεσμάτων Σεναρίου Την Εντολή Του OpenRtsp Για Αλλαγή στην Διάρκεια Αναπαραγωγής Του Αρχείου Και Quality Of Service Και Χρόνο Έκδοσης Των Αιτημάτων Μία Τυχαία Μεταβλητή

Πιο κάτω παρουσιάζουμε τα γραφικά αποτελέσματα του σεναρίου 3.9.2. Στην γραφική παράσταση αναγράφεται η μέση τιμή για την κάθε παραλλαγή του σεναρίου. Οι παραλλαγές του σεναρίου υπενθυμίζουμε είναι για πεντακόσια, επτακόσια, οκτακόσια και χίλια αιτήματα. Στο πιο κάτω διάγραμμα παρατηρούμε την αύξηση της χρήσης του CPU όσον αυξάνονται και τα αιτήματα.

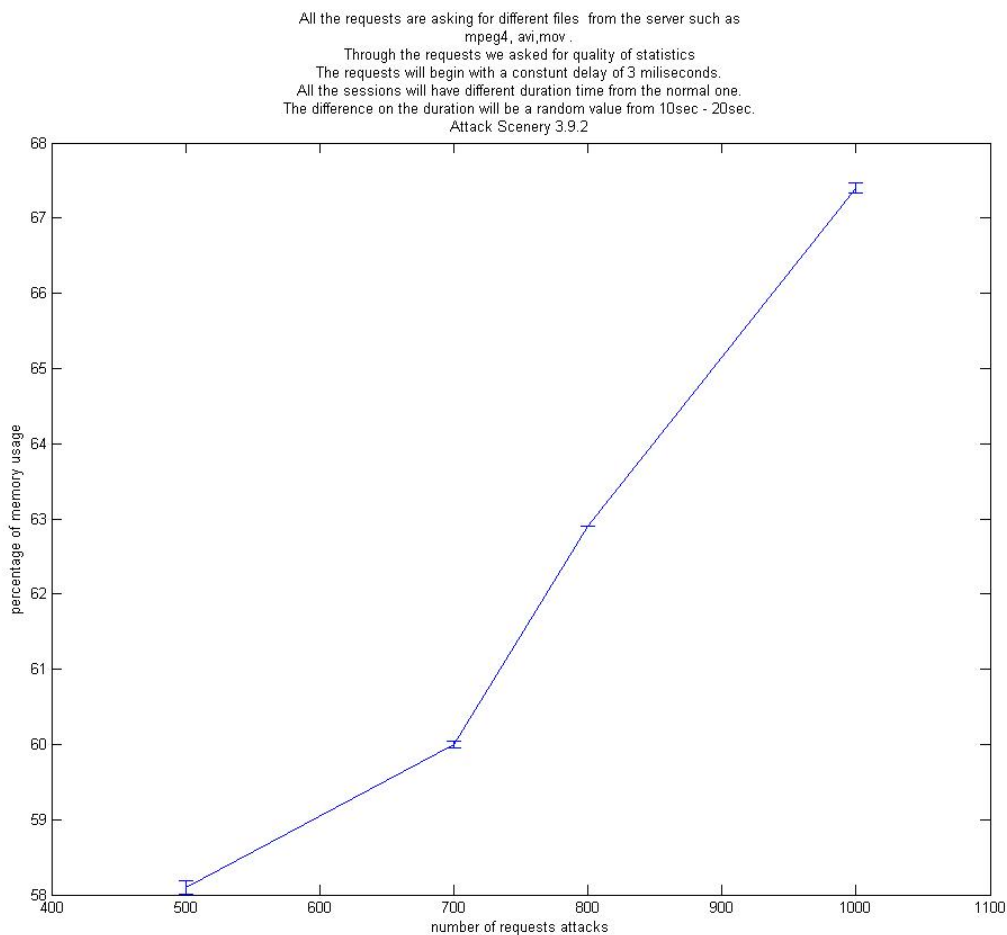


Γραφική 5.24: Μέση τιμή κατανάλωσης CPU Χρησιμοποιώντας Την Εντολή Του OpenRtsp Για Quality Of Service Ζητώντας Διαφορετικά Αρχεία Και Χρόνο Έκδοσης Των Αιτημάτων Μία Τυχαία Μεταβλητή

Παρατηρούμε ότι για πεντακόσια αιτήματα η μέση τιμή κατανάλωσης CPU από την διεργασία του DSS είναι 55,8%. Για επτακόσια αιτήματα έχουμε μία αύξηση της τάξεως του 0,7 % και η μέση τιμή ανεβαίνει στο 56,5%. Για οκτακόσια αιτήματα η αύξηση ανέρχεται στο 6,4% και η

μέση τιμή ανεβαίνει στο 63%. Για χίλια αιτήματα η μέση τιμή ανεβαίνει στο 63,8%. Παρατηρούμε μία αύξηση της μέσης τιμής κατανάλωσης του CPU όσον αυξάνονται τα αιτήματα.

Στην επόμενη γραφική παράσταση θα παρατηρήσουμε την συμπεριφορά της διεργασίας του DSS ως προς την κατανάλωση Memory. Αναγράφεται για κάθε παραλλαγή του σεναρίου η μέση τιμή κατανάλωσης μνήμης. Παρατηρούμε μία σημαντική αύξηση στο ποσοστό κατανάλωσης της μνήμης όσον αυξάνονται τα αιτήματα.



Γραφική 5.25: Μέση τιμή κατανάλωσης memory Χρησιμοποιώντας Την Εντολή Του OpenRtsp Για Quality Of Service Ζητώντας Διαφορετικά Αρχεία Και Χρόνο Έκδοσης Των Αιτημάτων Μία Τυχαία Μεταβλητή

Παρατηρούμε ότι για πεντακόσια αιτήματα η μέση τιμή κατανάλωσης Memory από την διεργασία του DSS ανέρχεται στο 58,2%. Για επτακόσια αιτήματα έχουμε μία αύξηση της τάξεως του 2,2% και η μέση τιμή ανεβαίνει στο 60%. Για οκτακόσια αιτήματα έχουμε μία αύξηση 3% και η μέση τιμή ανεβαίνει στο 63%. Για χίλια αιτήματα η μέση τιμή ανεβαίνει στο 67,5%.

Παρατηρούμε μία λογική αύξηση της μέσης τιμής κατανάλωσης του CPU και Memory αφού ο διακομιστής μας αναγκάζεται να εξυπηρετήσει όλο και περισσότερα αιτήματα, και επιπλέον οι σύνοδοι καθυστερούν να κλείσουν κατά 10 δεύτερα και ταυτόχρονα πρέπει να αποστέλλει και στατιστικά ποιότητας παράδοσης των δεδομένων.

Επιπλέον, εδώ μπορούμε να επισημάνουμε ότι ο κανόνας που αφορά την χρονική άφιξη των αιτημάτων δεν επαναλαμβάνεται εδώ αφού εάν συγκρίνουμε το πείραμα 3.9.1 και 3.9.2 όπου η μεταξύ τους διαφορά είναι στην χρονική άφιξη των αιτημάτων, οι μετρήσεις έχουν πολύ μικρή διαφορά με αποτέλεσμα στα χίλια αιτήματα όσον αφορά το cpu να ανέρχεται στο 63% για το 3.9.1 και στο 63,8% για το 3.9.2. Στην κατανάλωση μνήμης έχουμε διαφορά αλλά δεν εμπίπτει στον κανόνα που βλέπαμε μέχρι τώρα, δηλαδή όταν καταφθάνουν σχεδόν ταυτόχρονα τα αιτήματα να είναι και πιο αυξημένες οι μετρήσεις.

Βιβλιογραφία

- [01] H. Schulzrinne, A. Rao, and R. Lanphier. Real Time Streaming Protocol (RTSP). RFC 2326.
- [02] Tom Anderson and Timothy Roscoe, «Preventing internet denial-of-service with capabilities», In ACM SIGCOMM Computer Communication Review, volume 34, 2004.
- [03] B. Bloom, “Space/time trade-offs in hash coding with allowable errors”, Communications of the ACM, 1970
- [04] Stephen Northcut, Lenny Zeltser, Scott Winters, Karen Kent, Ronald W. Ritchey “Inside Network Perimeter Security”, Second Edition, 2005
- [05] James F. Kurose, Keith W. Ross “Computer Networking A Top –Down Approach”, Fifth Edition, 2008
- [06] Open Rtp Tool, <http://www.live555.com/openRTSP>
- [07] IETF, Real Time Streaming Protocol, <http://www.ietf.org/rfc/rfc2326.txt>
- [08] Real Time Streaming Protocol, <http://www.cs.columbia.edu/~hgs/rtsp/>
- [09] Real Time Streaming Protocol, <http://www.cse.iitd.ernet.in/~pkalra/siv864/streaming.pdf>
- [10] Nihat Altıparmak, Ali Tekeoğlu, Ali Saman Tosun, “DoS Resilience Of Real Time Streaming Protocol”, 2011 International Conference on Innovations in Information Technology, University Of Texas At San Antonio
- [11] CHEN Lei, YE Dejian, “DoS and DDoS Attack’s Possibility Verification on Streaming Media Application”, 2008 International Symposium on Information Science and Engineering, Shanghai China