

Assignment 2

Team 08

Περβολάρακης Ζαχαρίας AM4027

Σταματάκης Στυλιανός AM4041

Introduction

Στην πρώτη φάση του project που το υλοποιήσαμε χωρίς τα πολλαπλά streams ακολουθήσαμε την έξεις λογική. Ο τρόπος που υλοποιήσαμε το throttling είναι με την sleep της chrono. Αφού βρήκαμε τον αριθμό των πακέτων που πρέπει να σταλθούν ανά δευτερόλεπτο ώστε να πέτυχουμε το ζητούμενο bandwidth, παρατηρήσαμε ότι λογο του overhead της sleep δεν είναι βιώσιμο να κάνουμε sleep για κάθε πακέτο που στέλνουμε οπότε σκεφτήκαμε να φτιάξουμε ομάδες από πακέτα και αφού έχουμε στείλει εκείνη την ομάδα να κάνουμε μετα sleep. Με αυτό το τρόπο κάνουμε πολύ λιγότερες sleep αλλά και το bandwidth είναι αρκετά σταθερό και με λίγα spikes. Για να βρούμε πόση ώρα πρέπει να κοιμόμαστε ανά ομάδα πακέτων έπρεπε να λάβουμε υπόψη μας και ποσό χρόνο χρειάστηκε για να στείλει αυτά τα πακέτα. Ακόμα παρατηρήσαμε ότι παίζει μεγάλη σημασία και το μέγεθος του πακέτου. Πιο αναλυτικά άμα έχουμε ένα πολύ μεγάλο πακέτο σε εάν πολύ μικρό bandwidth είναι λογικό να χρειαστεί να στείλουμε λιγότερα πακέτα σε σχέση με εάν πακέτο μικρότερου μεγέθους. Αυτό επίσης επηρεάζει και στο packet loss έως ένα βαθμό. Η υλοποίηση που κάναμε είναι να υπολογίζεται κάπως δυναμικά το μέγεθος του πακέτου άμα δεν μας δώσει ο χρήστης κάποια τιμή. Συγκεκριμένα στα παραδείγματα μας αναφέρουμε ποσό είναι το μέγεθος του πακέτου.

Όλες οι μετρήσεις που κάναμε είτε στα μηχανήματα της σχολής είτε στην στο εργαστήριο ήταν αυτό ακριβώς που περιμέναμε. Το οποίο σημαίνει ότι είχαμε αρκετά σταθερό bandwidth και jitter, με το packet lost να είναι 0 – 2%, ανάλογα το bandwidth.

Στην δεύτερη φάση οπού αρχίσαμε να μεταφέρουμε το πρόγραμμα μας σε πολλαπλά streams αντιμετωπίσαμε κάποιες δυσκολίες όχι τόσο στην σύνδεση ή στην παραγωγή μηνυμάτων, ούτε στο να πέτυχουμε το ζητούμενο bandwidth. Εκεί που δυσκολευτήκαμε ήταν στο να πέτυχουμε το σωστό timing στο printing thread στον server. Πιο συγκεκριμένα το printing thread με το να τυπώνει σε άλλο χρόνο από ότι έρχονται τα πραγματικά δεδομένα οι μετρήσεις που δείχνει δεν είναι οι σωστές καθώς είτε ξανά μετράει πακέτα που έχει ήδη μετρήσει είτε τα δεδομένα που θεωρεί ότι έχει λάβει σε σχέση με τον χρόνο που έχει περάσει δεν είναι σωστά. Αυτός είναι και ο λόγος που στην απάντηση κάθε πειράματος έχουμε προσθέσει και την τελική αναφορά του server που είναι η σωστές μετρήσεις που έγιναν κατά την διάρκεια ολοι του πειράματος και όχι ανά iteration.

Τα πειράματα ακολουθούνε την έξεις μορφή: Καταρχάς όλα τα πειράματα έχουν τρέξει για 30 δευτερόλεπτα. Πάντα το πρώτο πείραμα είναι X Mbits/s για ένα stream και το δεύτερο είναι X/2 Mbits/s για δυο streams. Στην αρχή κάθε πειράματος αναγράφοντες σχετικές πληροφορίες.

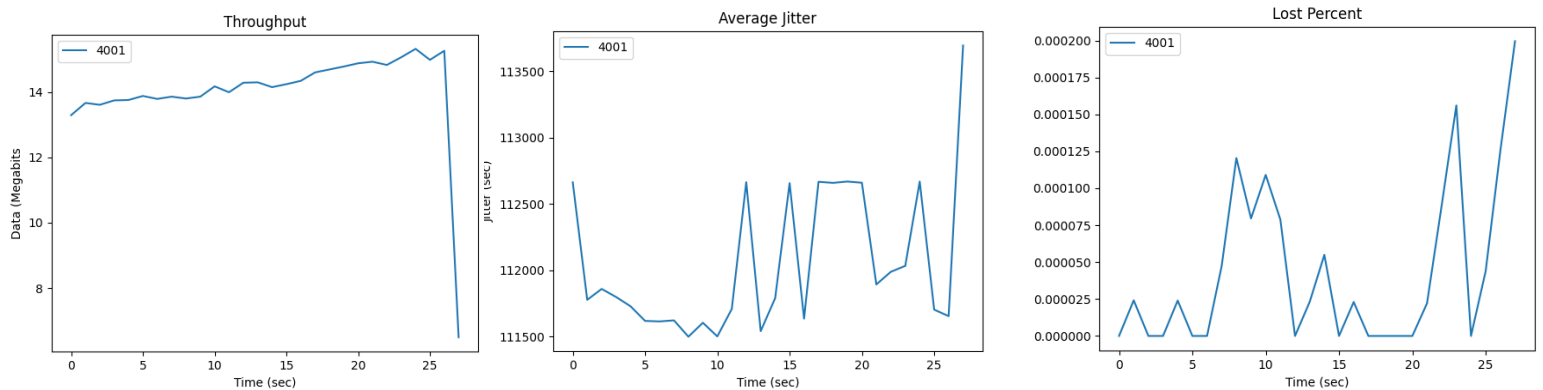
Experiment 1

Bandwidth: 100 Mbits/sec

Number of streams: 1

Packet Size: 1460 bytes

Run for 30 sec



Απάντηση:

Βλέπουμε ότι τα ζητούμενα 100Mbits/s δεν είναι πρόβλημα να τα πιάσει ο server, καθώς έχουμε και αρκετά μικρό και σταθερό jitter αλλά και πολύ μικρό packet loss. Παρατηρούμε ότι το throughput στο plot εμφανίζεται μεγαλύτερο από ότι μας δείχνουν παρακάτω οι τιμές. Αυτό συμβαίνει ακριβώς, καθώς όπως αναφέρθηκε και στην εισαγωγή η μέτρηση δεν είναι ακριβής ανά iteration (1s). Παρόλο που τα τελικά συμπεράσματα του server, τα οποία είναι και τα ακριβή, είναι αυτά που θα περιμέναμε να δούμε με εξαίρεση το ότι μας φαίνεται πιο λογικό να μην είχαμε χαμένα πακέτα, γνωρίζοντας την τοπολογία του συστήματος.

Total run time: 30.001454463 seconds

Throughput: 12.625395774841309 MegaBytes/sec

Bandwidth: 101.0016390764159 Megabits/sec

Average Jitter: 113444 nanoseconds

Lost/Total: 1100 / 265562

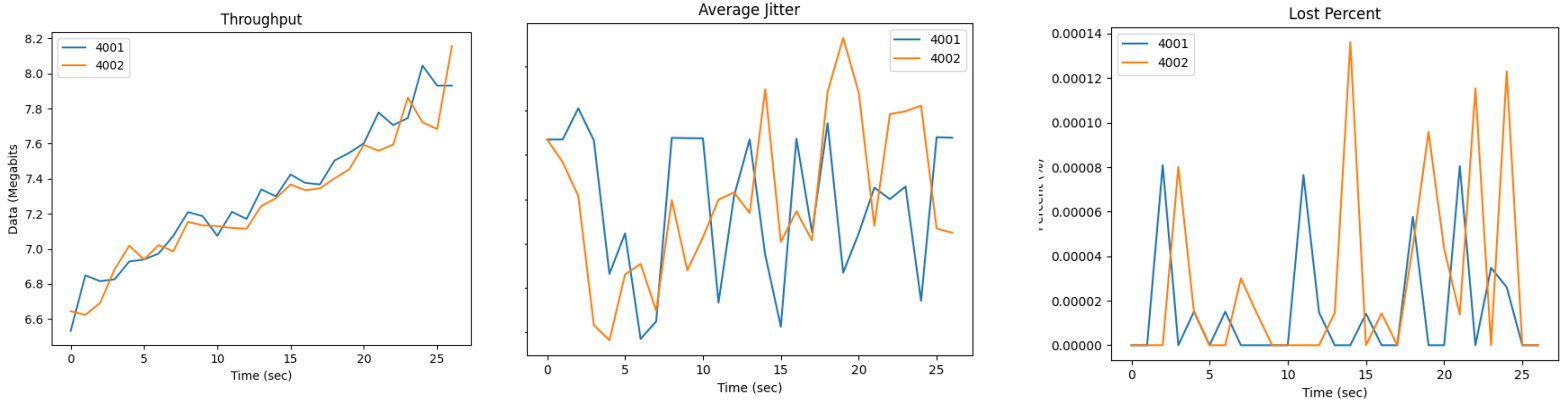
Lost percent: 0.41421588932151443 %

Bandwidth: 50 Mbits/sec

Number of streams: 2

Packet Size: 1460 bytes

Run for 30 sec



Απάντηση:

Έχοντας δημιουργήσει δυο streams με bandwidth 50Mbits/s το καθένα. Πάλι έχουμε αρκετά μικρό και σταθερό jitter αλλά και πολύ μικρό packet loss. Παρατηρούμε και πάλι το ίδιο πρόβλημα στο plot του throughput. Κοιτώντας όμως τις τελικές τιμές βλέπουμε ότι και τα δυο streams δεν είχαν κανένα πρόβλημα και είχαμε και μια αύξηση στο συνολικό packet loss. Επίσης... κάτι που μας φαίνεται ενδιαφέρον στο plot του average jitter, είναι το γεγονός ότι όταν το ένα stream έχει peak, το άλλο πέφτει.

Stream 0:

Total run time: 30.012517547 seconds
Throughput: 6.325227737426758 MegaBytes/sec
Bandwidth: 50.600949866125305 Megabits/sec
Average Jitter: 226302 nanoseconds
Lost/Total: 209 / 132782
Lost percent: 0.15740085252519165 %

Stream 1:

Total run time: 30.012517547 seconds
Throughput: 6.314632415771484 MegaBytes/sec
Bandwidth: 50.51618603962877 Megabits/sec
Average Jitter: 226732 nanoseconds
Lost/Total: 461 / 132782
Lost percent: 0.3471856125077194 %

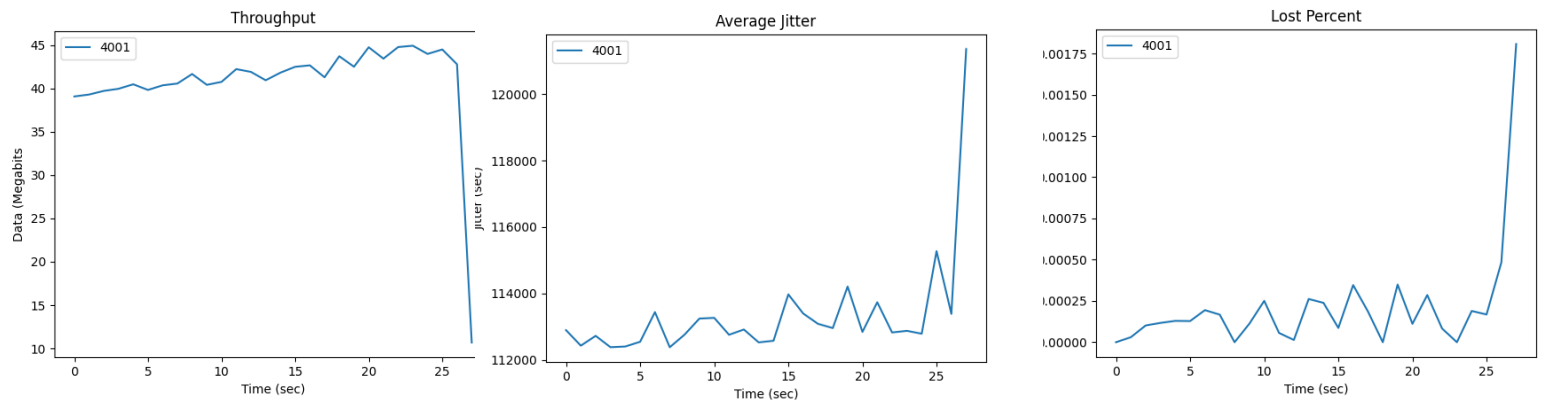
Experiment 2

Bandwidth: 300 Mbits/sec

Number of streams: 1

Packet Size: 1460 * 3 bytes

Run for 30 sec



Απάντηση:

Δεν περιμένουμε να δούμε καμία μεγάλη διαφορά στην συμπεριφορά συγκριτικά με 100Mbits/s καθώς ακόμα δεν έχουμε φτάσει στα άκρα του συστήματος. Και πάλι τα plots του jitter και Lost packet είναι αρκετά χαμηλά και σταθερά, ενώ στο throughput παρατηρούμε την ίδια λάθος συμπεριφορά.

Total run time: 30.001035177 seconds

Throughput: 36.62287902832031 MegaBytes/sec

Bandwidth: 292.98270195265735 Megabits/sec

Average Jitter: 115137 nanoseconds

Lost/Total: 4963 / 265532

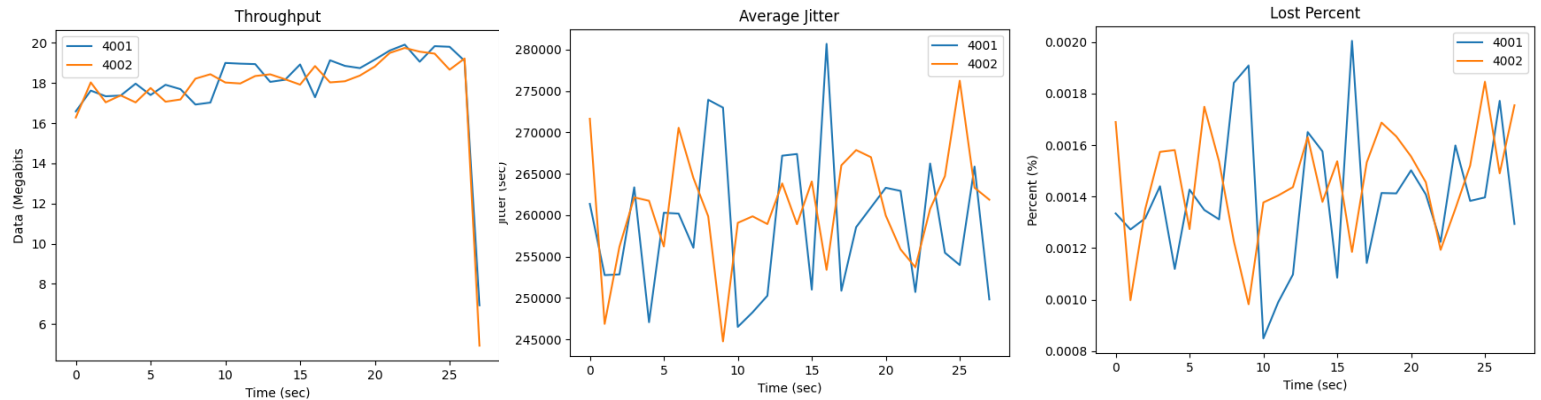
Lost percent: 1.8690779265775874 %

Bandwidth: 150 Mbits/sec

Number of streams: 2

Packet Size: 1460 * 3 bytes

Run for 30 sec



Απάντηση:

Αρχίζοντας από το plot του Jitter παρατηρούμε ότι αυξήθηκε συγκριτικά με το ένα stream όμως και πάλι διατηρείτε κάπως σταθερό. Παρατηρούμε πολύ μεγαλύτερο ποσοστό συνολικών χαμένων πακέτων καθώς και μικρότερο συνολικό throughput. Η οποία είναι μια περίεργη συμπεριφορά καθώς η μέτρηση είναι μέσα στην δυνατότητα της τοπολογία του δικτύου μας. Υποθέτουμε ότι αυτό συμβαίνει λόγω κάποιων πιθανών collisions μεταξύ των πακέτων των δυο stream.

Stream 0:

Total run time: 30.002336422 seconds
Throughput: 16.043209075927734 MegaBytes/sec
Bandwidth: 128.34424029629037 Megabits/sec
Average Jitter: 262767 nanoseconds
Lost/Total: 18593 / 132782
Lost percent: 14.00265096172674 %

Stream 1:

Total run time: 30.002336422 seconds
Throughput: 15.933711051940918 MegaBytes/sec
Bandwidth: 127.4682663424913 Megabits/sec
Average Jitter: 264661 nanoseconds
Lost/Total: 19423 / 132782
Lost percent: 14.627735687066018 %

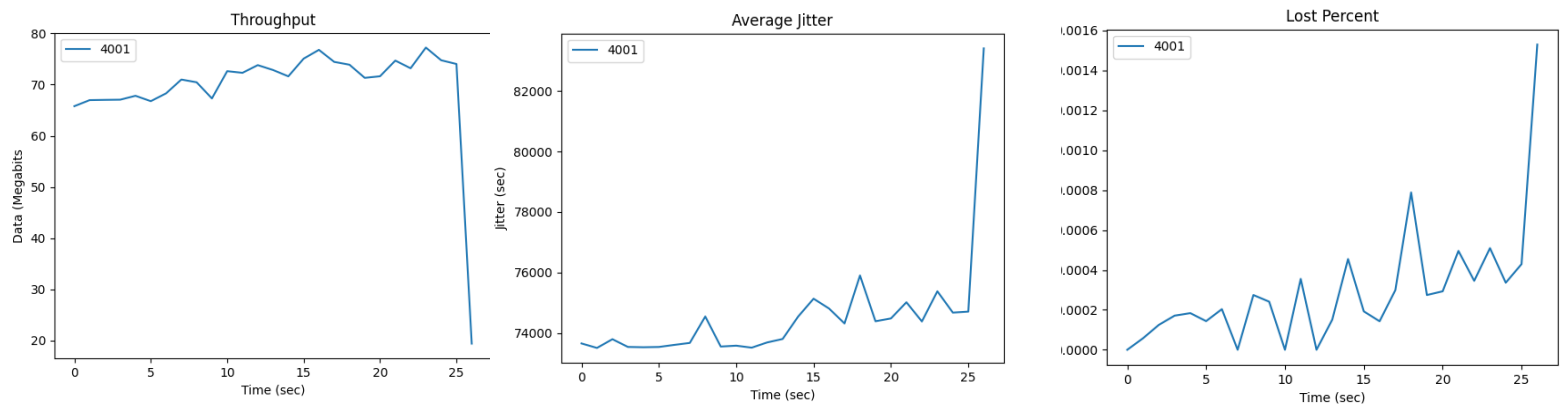
Experiment 3

Bandwidth: 500 Mbits/sec

Number of streams: 1

Packet Size: 1460 * 3.3 bytes

Run for 30 sec



Απάντηση:

Τρεχοντας το πειραμα και για τα 500Mbits/s, βλεπουμε ότι τα plots του jitter και Lost packet είναι αρκετά χαμηλά και σταθερά, ενώ στο throughput παρατηρούμε την ίδια λάθος συμπεριφορά την οποία και περιμεναμε. Κοιτοντας και τα παρακατω αποτελεσματα διαπιστώνουμε ότι καταφεραμε να ειμαστε σταθερα κοντα στο ζητουμενο bandwidth.

Total run time: 30.001109158 seconds

Throughput: 60.956382751464844 MegaBytes/sec

Bandwidth: 487.64928328527867 Megabits/sec

Average Jitter: 76027 nanoseconds

Lost/Total: 12606 / 407219

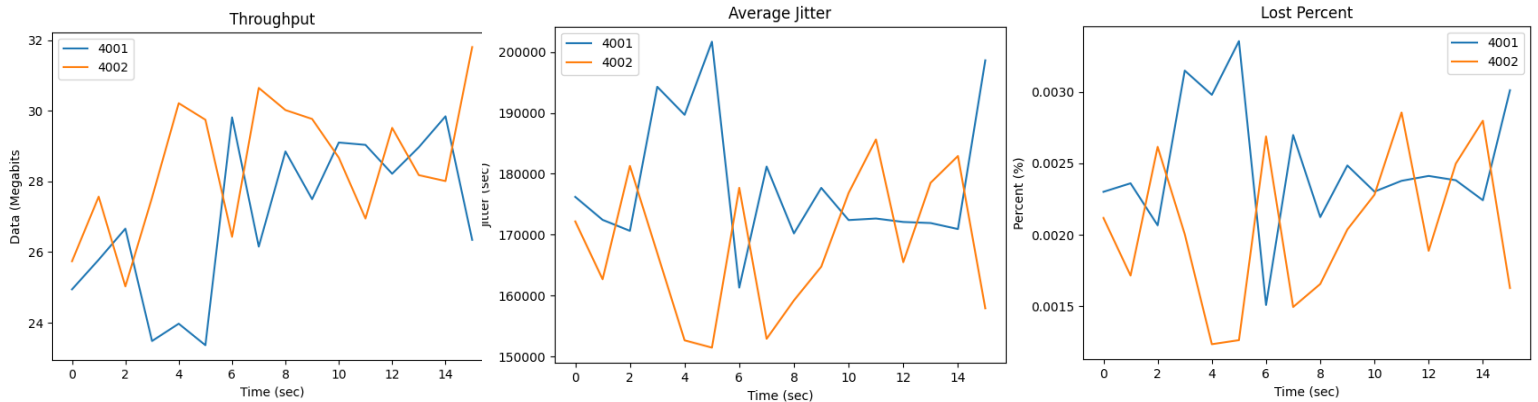
Lost percent: 3.095631588899339 %

Bandwidth: 250 Mbits/sec

Number of streams: 2

Packet Size: 1460 * 3 bytes

Run for 30 sec



Απάντηση:

Ξανά... μένουμε στα όρια της γραμμής, οπότε δεν παρατηρούμε κάτι πολύ διαφορετικό από τις προηγούμενες εκτελέσεις του πειράματος. Παρόλα αυτά... σε αυτή την διεξαγωγή του πειράματος που υπάρχει πιο έντονη μεταφορά δεδομένων έχουμε μεγαλύτερο ποσοστό χαμένων πακέτων. Αυτό μας δίνει συνολικά ένα χαμηλότερο ποσοστό bandwidth σε σχέση με το target bandwidth.

Stream 0:

Total run time: 17.851451478 seconds
Throughput: 23.33381462097168 MegaBytes/sec
Bandwidth: 186.66578840319363 Megabits/sec
Average Jitter: 180510 nanoseconds
Lost/Total: 32753 / 131576
Lost percent: 24.89283759956223 %

Stream 1:

Total run time: 17.851451478 seconds
Throughput: 24.65960121154785 MegaBytes/sec
Bandwidth: 197.27182058388152 Megabits/sec
Average Jitter: 170942 nanoseconds
Lost/Total: 27242 / 131677
Lost percent: 20.688502927618337 %

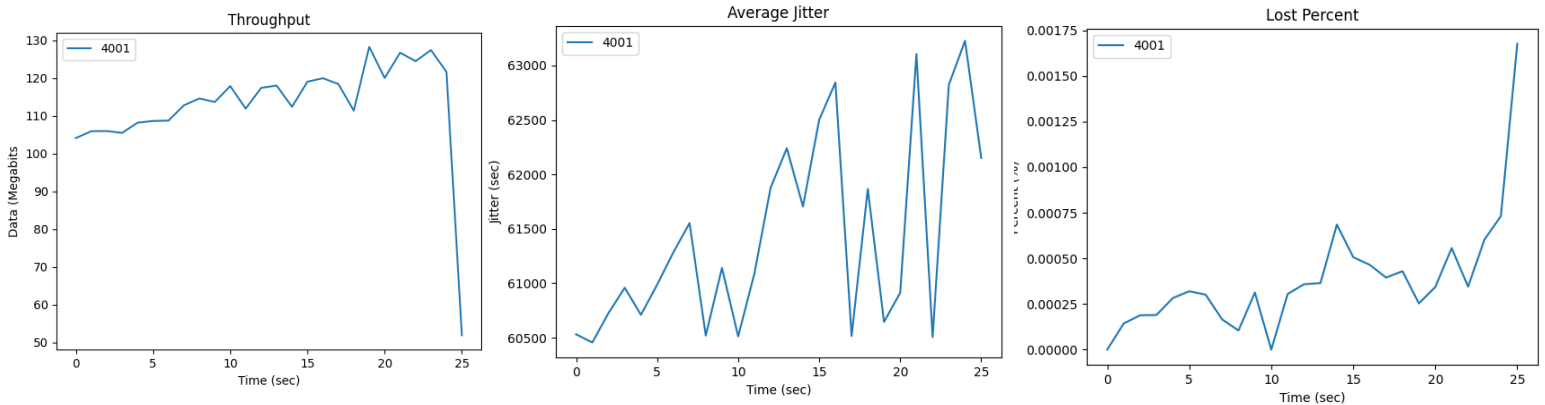
Experiment 4

Bandwidth: 800 Mbits/sec

Number of streams: 1

Packet Size: 1460 * 4 bytes

Run for 30 sec



Απάντηση:

Παρόμοια με τα υπόλοιπα πειράματα.

Total run time: 30.000987744 seconds

Throughput: 95.30323028564453 MegaBytes/sec

Bandwidth: 762.4007607376412 Megabits/sec

Average Jitter: 63237 nanoseconds

Lost/Total: 21292 / 495714

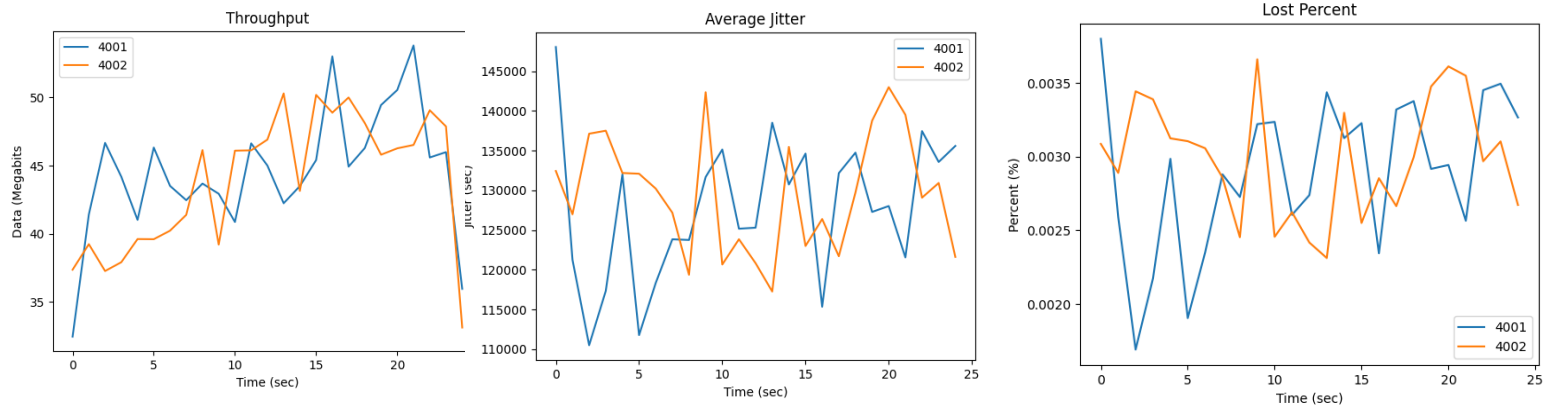
Lost percent: 4.2952186139588555 %

Bandwidth: 400 Mbits/sec

Number of streams: 2

Packet Size: 1460 * 3.3 bytes

Run for 30 sec



Απάντηση:

Παρόμοια με τα προηγούμενα πειράματα των 2 streams.

Stream 0:

Total run time: 30.009026564 seconds

Throughput: 35.863346099853516 MegaBytes/sec

Bandwidth: 286.90650237228465 Megabits/sec

Average Jitter: 129186 nanoseconds

Lost/Total: 95282 / 327526

Lost percent: 29.091430909301856 %

Stream 1:

Total run time: 30.009026564 seconds

Throughput: 35.312889099121094 MegaBytes/sec

Bandwidth: 282.50285910282605 Megabits/sec

Average Jitter: 131212 nanoseconds

Lost/Total: 98878 / 327526

Lost percent: 30.189359012719603 %