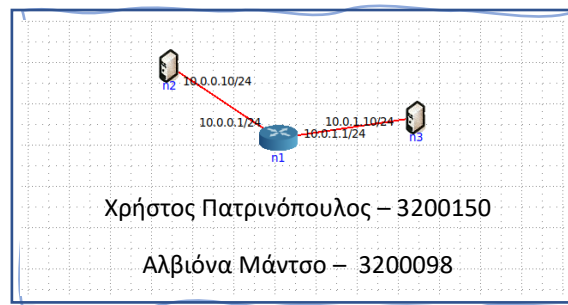


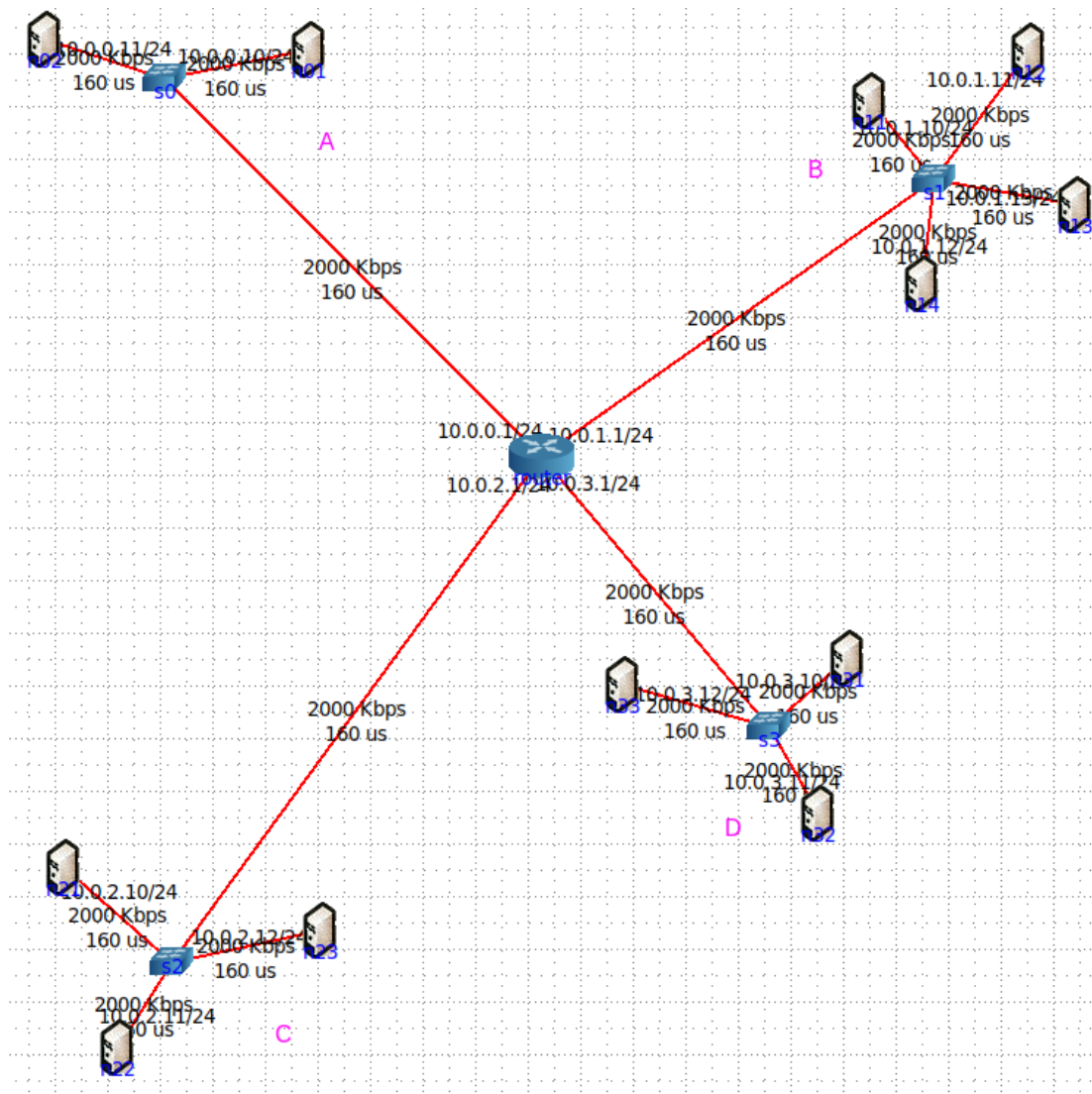
CORE



VM

A. Η διάρθρωση του δικτύου συνοψίζεται στον πίνακα και φαίνεται στην παρακάτω εικόνα:

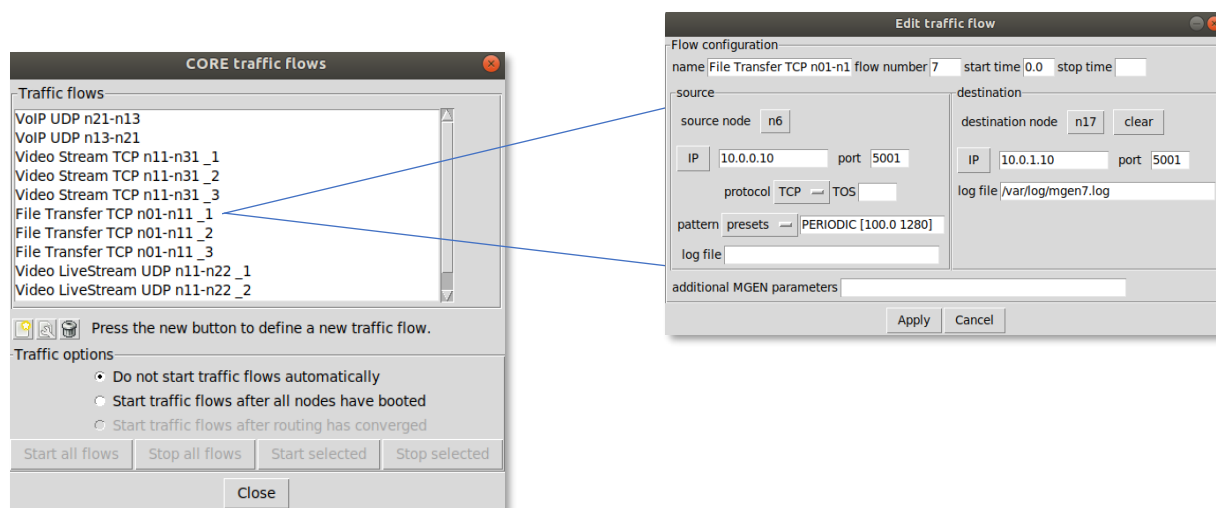
Subnet	Address	Number of hosts	Nodes
A	10.0.0.0/24	2	n01, n02
B	10.0.1.0/24	4	n11, n12, n13, n14
C	10.0.2.0/24	3	n21, n22, n23
D	10.0.3.0/24	3	n31, n32, n33



Σημειώνεται ότι οι γραμμές έχουν περιοριστεί σε bandwidth 2000Kbps ώστε να παρατηρηθούν ευκολότερα και πιο γρήγορα τα προβλήματα συμφορήσεων του δικτύου στα επόμενα ερωτήματα της εργασίας.

B. Τα flows που δημιουργούνται συνοψίζονται στον πίνακα και αναλύονται παρακάτω:

Name	Source Node	Destination Node	Source Port	Destination Port
File Transfer TCP n01-n11 _1	n01	n11	5001	5001
File Transfer TCP n01-n11 _2	n01	n11	5002	5002
File Transfer TCP n01-n11 _3	n01	n11	5003	5003
Video Stream TCP n11-n31 _1	n11	n31	5011	5011
Video Stream TCP n11-n31 _2	n11	n31	5012	5012
Video Stream TCP n11-n31 _3	n11	n31	5013	5013
Video LiveStream UDP n11-n22 _1	n11	n22	6001	6001
Video LiveStream UDP n11-n22 _2	n11	n22	6002	6002
Video LiveStream UDP n11-n22 _3	n11	n22	6003	6003
VoIP UDP n13-21	n13	n21	7001	7001
VoIP UDP n21-13	n21	n13	7002	7002



Σημειώνεται ότι τα Source Port και Destination Port έχουν ίδιο αριθμό σε κάθε flow για λόγους εύκολης διαχείρισης του εργαλείου iperf σε επόμενο ερώτημα, χωρίς δλδ αυτό να επιβάλλεται.

Video Streaming:

On demand Video Streaming

Video Stream TCP n11-n31 _1/_2:

Από n11 (B) προς n31(C)

Bitrate: Θεωρούμε ότι τα video είναι ανάλυσης 720p και 30fps οπότε το αντίστοιχο bitrate είναι 1.500 Kbps – 4.000 Kbps.

Transfer Protocol: Θα πρόκειται για on demand video streaming (π.χ. netflix), οπότε το καταλληλότερο πρωτόκολλο είναι το TCP (quality of service).

- **Flow Pattern & Protocol:** 2 flows «PERIODIC [100.0 1280]» (2 x 1024Kbps ώστε να προσομοιωθεί το παραπάνω bitrate) και TCP.



“

Currently MGEN supports four pattern types, "PERIODIC", "POISSON", "BURST", "JITTER", and "CLONE". Complex traffic patterns can be created by using a compound of multiple "flows" (with the same SRC/DST) with different pattern types and parameters.

”

Video Stream TCP n11-n31_3:

Από n11 (B) προς n31(C)

Χρησιμοποιείται ως διόρθωση του PERIODIC pattern για να προσομοιώσει τις τυχαίες χρονικά αλλαγές του bitrate, για παράδειγμα κατά την μετάβαση από παρόμοια διαδοχικά frames (στατικότητα εικόνας) σε γρήγορα εναλλασσόμενα διαφορετικά frames (π.χ. σε μια σκηνή δράσης).

Parameters: Θεωρούμε ότι οι ριπές συμβαίνουν κατά μέσω όρο κάθε 10.0 sec ακανόνιστα (συνεπώς RANDOM) και έχουν διάρκεια 3.0 sec FIXED, αυτά εφόσον πρόκειται για on demand video streaming (πιθανόν κάποια ταινία) και συνεπώς οι γρήγορα εναλλασσόμενες σκηνές δεν είναι πάρα πολύ συχνές (συγκριτικά με το live, βλ. παρακάτω), εξαρτάται φυσικά αυτό και από παράγοντες που έχουν να κάνουν με την ταινία. Τα packets στέλνονται με PERIODIC ροή με ένα bitrate των 256Kbps, γεγονός που αυξάνει κατά ένα λογικό ποσό το συνολικό bitrate.

Transfer Protocol: TCP, ίδιο με τη βασική (periodic) ροή (εφόσον είναι διόρθωση).

- **Flow Pattern & Protocol:** 1 flow « BURST [RANDOM 10.0 PERIODIC [25.0 1280] FIXED 3.0] » και TCP.

Live Video Streaming

Video Live Stream UDP n11-n22_1/_2:

Από n11 (B) προς n22(C)

Bitrate: Θεωρούμε ότι τα video είναι ανάλυσης 360p και 30fps οπότε το αντίστοιχο bitrate είναι 800 Kbps -1200 Kbps.

Transfer Protocol: Θα πρόκειται για live video streaming (π.χ. YouTube Livestream), οπότε το καταλληλότερο πρωτόκολλο είναι το UDP (faster).

- **Flow Pattern & Protocol:** 2 flows «PERIODIC [50.0 1280]» (2 x 512Kbps ώστε να προσομοιωθεί το παραπάνω bitrate) και UDP.

Video Live Stream UDP n11-n22_3:

Από n11 (B) προς n22(C)

Χρησιμοποιείται ως διόρθωση του PERIODIC pattern για να προσομοιώσει τις τυχαίες χρονικά αλλαγές του bitrate, για παράδειγμα κατά την μετάβαση από παρόμοια διαδοχικά frames (στατικότητα εικόνας) σε γρήγορα εναλλασσόμενα διαφορετικά frames (π.χ. σε μια σκηνή δράσης).

Transfer Protocol: UDP, ίδιο με τη βασική (periodic) ροή (εφόσον είναι διόρθωση).

Parameters: Θεωρούμε ότι οι ριπές συμβαίνουν κατά μέσω όρο κάθε 3.0 sec ακανόνιστα (συνεπώς RANDOM), έχουν διάρκεια κατά μέσο όρο 1.0 sec και ακολουθούν κάποια εκθετική

κατανομή (συνεπώς EXP), αυτά εφόσον πρόκειται για live video streaming και συνεπώς οι γρήγορα εναλλασσόμενες σκηνές παρουσιάζονται αρκετά συχνά με μια μικρή διάρκεια. Τα packets στέλνονται με PERIODIC ροή με ένα bitrate των 256Kbps, γεγονός που αυξάνει κατά ένα λογικό ποσό το συνολικό bitrate.

- **Flow Pattern & Protocol:** 1 flow « BURST [RANDOM 3.0 PERIODIC [25.0 1280] EXP 1.0] » και UDP.

File Transfer

File Transfer TCP n01-n11 _1/_2/_3 :

Από n01 (A) προς n11(B)

Γίνεται μεταφορά αρχείων από τον host n01 προς τον n11. Τα αρχεία αυτά αφορούν το βίντεο που γίνεται stream παραπάνω (το on demand video streaming). Οπότε θέλουμε έναν παρόμοιο ή μεγαλύτερο ρυθμό μετάδοσης, δηλ για να μεταδοθούν έγκαιρα από τον n11 στον n21 (streaming host προς client host) θα πρέπει να έχουν φτάσει έγκαιρα από τον n01 στον n11 (database host προς streaming host). Συνεπώς χρησιμοποιούνται 3 flows, ώστε επιτυγχάνεται ένα μεγαλύτερο bitrate σε σχέση με του streaming προς τον client host, με την ιδέα ότι γίνεται buffering στον streaming host.

Transfer Protocol: Χρειαζόμαστε αξιοπιστία στη μεταφορά των αρχείων οπότε χρησιμοποιείται TCP.

- **Flow Pattern & Protocol:** 3 flows « PERIODIC [100.0 1280] » και TCP.

Push to Talk Voice Communication

VoIP UDP n13-n21:

Από n03 (B) προς n21(C)

VoIP UDP n21-n13:

Από n21 (C) προς n13(B)

Το Push to Talk VoIP στέλνει ήχο σε μικρές ακανόνιστες ριπές όταν ο χρήστης κρατάει πατημένο το κουμπί στη συσκευή του. Αυτό έχει ως αποτέλεσμα την κατά ριπές μεταφορά δεδομένων όπου περίοδοι χαμηλής ή μηδενικής μεταφοράς ακολουθούνται από αυτές τις ριπές πολλών δεδομένων όταν μιλάει ο χρήστης. Όμοια φαίνεται να προσεγγίζει το θέμα και το παρακάτω απόσπασμα από το pdf:

An example use of the BURST pattern would be to roughly emulate the "talk spurts" which might result from Voice Over IP (VOIP) applications. As a voice conversation commences, a user's burst of activity (talk spurts) might be RANDOM with some average interval and the duration talk spurts approximate EXPONENTIAL statistics. > When the talk spurt (burst) occurs, the voice compression codec might generate messages following something like a PERIODIC flow with packet rates and packet sizes dependent upon the voice codec in use.

Δημιουργούμε 2 αντίθετα κατευθυνόμενα flows ώστε στέλνοντας ο ένας κόμβος στον άλλον να επιτυγχάνεται η προσομοίωση της αμφίδρομης επικοινωνίας μεταξύ τους με VoIP.

Transfer Protocol: Το πρωτόκολλο που χρησιμοποιείται είναι το UDP (faster).

Parameters: Θεωρούμε ότι οι ριπές συμβαίνουν κατά μέσο όρο κάθε 8.0 sec ακανόνιστα (συνεπώς RANDOM), έχουν διάρκεια κατά μέσο όρο 4.0 sec και ακολουθούν κάποια εκθετική κατανομή (συνεπώς EXP) και τα packets στέλνονται στη συνέχεια με PERIODIC ροή με ένα bitrate των 100Kbps (τυπικές τιμές είναι 80Kbps-100Kbps).

■ **Flow Pattern & Protocol:** 2 flows «BURST [RANDOM 8.0 PERIODIC [10.0 1250] EXP 4.0]» και UDP.

Γ.

α) Διάγραμμα μετρήσεων καθυστέρησης (Average RTT)

Για την κατασκευή του διαγράμματος έγιναν τα εξής:

Αυξάνοντας διαδοχικά τα ενεργά flows (1. No flow – 2. VoIP – 3. Video Live Stream – 4. File Transfer – 5. Video Stream (Part 1) – 6. Video Stream (Part 2)) και συνεπώς την κίνηση στο δίκτυο:

1. Χρησιμοποιήθηκε το εργαλείο ping από τον κόμβο n23 (υποδίκτυο C) προς τον κόμβο n11 (υποδίκτυο B), με αποστολή 100 πακέτων κάθε φορά. Σώστε να έχουμε ένα στατιστικά ικανοποιητικό μέγεθος δείγματος.

```
root@n23:/tmp/pycore.39123/n23.conf# ping 10.0.1.10 -c 100
```

2. Από τα αποτελέσματα των 100 μηνυμάτων λήφθηκε το average RTT ως καθυστέρηση, τιμή η οποία χρησιμοποιείται στο διάγραμμα.
3. Οι τιμές του βήματος 2 (μια για κάθε ενεργοποίηση επιπλέον flow) συνιστούν το διάγραμμα που έγινε με τη χρήση Excel.

Παρακάτω φαίνονται screenshots από τις μετρήσεις όπως περιεγράφηκαν στα βήματα, και ακολουθεί το τελικό διάγραμμα.

Μέτρηση του RTT όταν...

```
root@n23:/tmp/pycore.38893/n23.conf
File Edit View Search Terminal Help
64 bytes from 10.0.1.10: icmp_seq=82 ttl=63 time=1.96 ms
64 bytes from 10.0.1.10: icmp_seq=83 ttl=63 time=1.16 ms
64 bytes from 10.0.1.10: icmp_seq=84 ttl=63 time=4.63 ms
64 bytes from 10.0.1.10: icmp_seq=85 ttl=63 time=1.22 ms
64 bytes from 10.0.1.10: icmp_seq=86 ttl=63 time=11.9 ms
64 bytes from 10.0.1.10: icmp_seq=87 ttl=63 time=6.15 ms
64 bytes from 10.0.1.10: icmp_seq=88 ttl=63 time=2.57 ms
64 bytes from 10.0.1.10: icmp_seq=89 ttl=63 time=3.11 ms
64 bytes from 10.0.1.10: icmp_seq=90 ttl=63 time=1.26 ms
64 bytes from 10.0.1.10: icmp_seq=91 ttl=63 time=2.62 ms
64 bytes from 10.0.1.10: icmp_seq=92 ttl=63 time=2.02 ms
64 bytes from 10.0.1.10: icmp_seq=93 ttl=63 time=5.81 ms
64 bytes from 10.0.1.10: icmp_seq=94 ttl=63 time=11.5 ms
64 bytes from 10.0.1.10: icmp_seq=95 ttl=63 time=14.0 ms
64 bytes from 10.0.1.10: icmp_seq=96 ttl=63 time=5.74 ms
64 bytes from 10.0.1.10: icmp_seq=97 ttl=63 time=9.89 ms
64 bytes from 10.0.1.10: icmp_seq=98 ttl=63 time=22.2 ms
64 bytes from 10.0.1.10: icmp_seq=99 ttl=63 time=11.0 ms
64 bytes from 10.0.1.10: icmp_seq=100 ttl=63 time=8.02 ms

--- 10.0.1.10 ping statistics ---
100 packets transmitted, 100 received, 0% packet loss, time 99277ms
rtt min/avg/max/mdev = 0.854/5.359/25.907/5.403 ms
root@n23:/tmp/pycore.38893/n23.conf#
```

...όταν κανένα flow δεν είναι ενεργό: "No flow"

```
root@n23:/tmp/pycore.38893/n23.conf
File Edit View Search Terminal Help
64 bytes from 10.0.1.10: icmp_seq=82 ttl=63 time=22.8 ms
64 bytes from 10.0.1.10: icmp_seq=83 ttl=63 time=1.73 ms
64 bytes from 10.0.1.10: icmp_seq=84 ttl=63 time=1.10 ms
64 bytes from 10.0.1.10: icmp_seq=85 ttl=63 time=6.39 ms
64 bytes from 10.0.1.10: icmp_seq=86 ttl=63 time=5.21 ms
64 bytes from 10.0.1.10: icmp_seq=87 ttl=63 time=9.91 ms
64 bytes from 10.0.1.10: icmp_seq=88 ttl=63 time=3.58 ms
64 bytes from 10.0.1.10: icmp_seq=89 ttl=63 time=1.90 ms
64 bytes from 10.0.1.10: icmp_seq=90 ttl=63 time=4.08 ms
64 bytes from 10.0.1.10: icmp_seq=91 ttl=63 time=1.47 ms
64 bytes from 10.0.1.10: icmp_seq=92 ttl=63 time=1.44 ms
64 bytes from 10.0.1.10: icmp_seq=93 ttl=63 time=1.54 ms
64 bytes from 10.0.1.10: icmp_seq=94 ttl=63 time=3.19 ms
64 bytes from 10.0.1.10: icmp_seq=95 ttl=63 time=6.71 ms
64 bytes from 10.0.1.10: icmp_seq=96 ttl=63 time=31.8 ms
64 bytes from 10.0.1.10: icmp_seq=97 ttl=63 time=13.9 ms
64 bytes from 10.0.1.10: icmp_seq=98 ttl=63 time=10.2 ms
64 bytes from 10.0.1.10: icmp_seq=99 ttl=63 time=18.0 ms
64 bytes from 10.0.1.10: icmp_seq=100 ttl=63 time=8.02 ms

--- 10.0.1.10 ping statistics ---
100 packets transmitted, 100 received, 0% packet loss, time 99296ms
rtt min/avg/max/mdev = 0.843/5.531/31.801/5.895 ms
root@n23:/tmp/pycore.38893/n23.conf#
```

...όταν ενεργοποιούνται τα VoIP flows: "VoIP"


```

root@n23: /tmp/pycore.38893/n23.conf
File Edit View Search Terminal Help
64 bytes from 10.0.1.10: icmp_seq=82 ttl=63 time=5.58 ms
64 bytes from 10.0.1.10: icmp_seq=83 ttl=63 time=2.71 ms
64 bytes from 10.0.1.10: icmp_seq=84 ttl=63 time=8.91 ms
64 bytes from 10.0.1.10: icmp_seq=85 ttl=63 time=2.13 ms
64 bytes from 10.0.1.10: icmp_seq=86 ttl=63 time=3.09 ms
64 bytes from 10.0.1.10: icmp_seq=87 ttl=63 time=4.54 ms
64 bytes from 10.0.1.10: icmp_seq=88 ttl=63 time=1.47 ms
64 bytes from 10.0.1.10: icmp_seq=89 ttl=63 time=16.1 ms
64 bytes from 10.0.1.10: icmp_seq=90 ttl=63 time=1.17 ms
64 bytes from 10.0.1.10: icmp_seq=91 ttl=63 time=5.46 ms
64 bytes from 10.0.1.10: icmp_seq=92 ttl=63 time=1.11 ms
64 bytes from 10.0.1.10: icmp_seq=93 ttl=63 time=10.3 ms
64 bytes from 10.0.1.10: icmp_seq=94 ttl=63 time=2.04 ms
64 bytes from 10.0.1.10: icmp_seq=95 ttl=63 time=9.83 ms
64 bytes from 10.0.1.10: icmp_seq=96 ttl=63 time=10.4 ms
64 bytes from 10.0.1.10: icmp_seq=97 ttl=63 time=7.38 ms
64 bytes from 10.0.1.10: icmp_seq=98 ttl=63 time=1.10 ms
64 bytes from 10.0.1.10: icmp_seq=99 ttl=63 time=2.93 ms
64 bytes from 10.0.1.10: icmp_seq=100 ttl=63 time=3.17 ms

--- 10.0.1.10 ping statistics ---
100 packets transmitted, 100 received, 0% packet loss, time 99364ms
rtt min/avg/max/mdev = 0.891/7.937/59.362/10.159 ms
root@n23: /tmp/pycore.38893/n23.conf#

```

...όταν ενεργοποιούνται επιπλέον τα Video Live Stream flows: "Video Live Stream"

```

root@n23: /tmp/pycore.38893/n23.conf
File Edit View Search Terminal Help
64 bytes from 10.0.1.10: icmp_seq=82 ttl=63 time=47.3 ms
64 bytes from 10.0.1.10: icmp_seq=83 ttl=63 time=55.8 ms
64 bytes from 10.0.1.10: icmp_seq=84 ttl=63 time=54.4 ms
64 bytes from 10.0.1.10: icmp_seq=85 ttl=63 time=54.9 ms
64 bytes from 10.0.1.10: icmp_seq=86 ttl=63 time=55.1 ms
64 bytes from 10.0.1.10: icmp_seq=87 ttl=63 time=68.2 ms
64 bytes from 10.0.1.10: icmp_seq=88 ttl=63 time=60.7 ms
64 bytes from 10.0.1.10: icmp_seq=89 ttl=63 time=61.7 ms
64 bytes from 10.0.1.10: icmp_seq=90 ttl=63 time=58.1 ms
64 bytes from 10.0.1.10: icmp_seq=91 ttl=63 time=66.9 ms
64 bytes from 10.0.1.10: icmp_seq=92 ttl=63 time=63.2 ms
64 bytes from 10.0.1.10: icmp_seq=93 ttl=63 time=96.9 ms
64 bytes from 10.0.1.10: icmp_seq=94 ttl=63 time=66.1 ms
64 bytes from 10.0.1.10: icmp_seq=95 ttl=63 time=59.8 ms
64 bytes from 10.0.1.10: icmp_seq=96 ttl=63 time=66.9 ms
64 bytes from 10.0.1.10: icmp_seq=97 ttl=63 time=62.4 ms
64 bytes from 10.0.1.10: icmp_seq=98 ttl=63 time=60.1 ms
64 bytes from 10.0.1.10: icmp_seq=99 ttl=63 time=66.4 ms
64 bytes from 10.0.1.10: icmp_seq=100 ttl=63 time=62.5 ms

--- 10.0.1.10 ping statistics ---
100 packets transmitted, 100 received, 0% packet loss, time 99209ms
rtt min/avg/max/mdev = 18.121/38.282/96.921/14.719 ms
root@n23: /tmp/pycore.38893/n23.conf#

```

...όταν ενεργοποιούνται επιπλέον τα File Stream flows: "File Transfer"

```

root@n23: /tmp/pycore.38893/n23.conf
File Edit View Search Terminal Help
64 bytes from 10.0.1.10: icmp_seq=78 ttl=63 time=5639 ms
64 bytes from 10.0.1.10: icmp_seq=82 ttl=63 time=5587 ms
64 bytes from 10.0.1.10: icmp_seq=83 ttl=63 time=5483 ms
64 bytes from 10.0.1.10: icmp_seq=84 ttl=63 time=5435 ms
64 bytes from 10.0.1.10: icmp_seq=85 ttl=63 time=5397 ms
64 bytes from 10.0.1.10: icmp_seq=86 ttl=63 time=5572 ms
64 bytes from 10.0.1.10: icmp_seq=87 ttl=63 time=5743 ms
64 bytes from 10.0.1.10: icmp_seq=88 ttl=63 time=5668 ms
64 bytes from 10.0.1.10: icmp_seq=89 ttl=63 time=5638 ms
64 bytes from 10.0.1.10: icmp_seq=91 ttl=63 time=5633 ms
64 bytes from 10.0.1.10: icmp_seq=92 ttl=63 time=5637 ms
64 bytes from 10.0.1.10: icmp_seq=93 ttl=63 time=5501 ms
64 bytes from 10.0.1.10: icmp_seq=94 ttl=63 time=5272 ms
64 bytes from 10.0.1.10: icmp_seq=95 ttl=63 time=5203 ms
64 bytes from 10.0.1.10: icmp_seq=96 ttl=63 time=5145 ms
64 bytes from 10.0.1.10: icmp_seq=97 ttl=63 time=5102 ms
64 bytes from 10.0.1.10: icmp_seq=98 ttl=63 time=5160 ms
64 bytes from 10.0.1.10: icmp_seq=99 ttl=63 time=5354 ms
64 bytes from 10.0.1.10: icmp_seq=100 ttl=63 time=5541 ms

--- 10.0.1.10 ping statistics ---
100 packets transmitted, 91 received, 9% packet loss, time 99729ms
rtt min/avg/max/mdev = 2990.936/4444.578/5743.015/778.812 ms, pipe 6
root@n23: /tmp/pycore.38893/n23.conf#

```

...όταν ενεργοποιείται επιπλέον το πρώτο από τα Video Stream flows: "Video Stream (Part 1)"

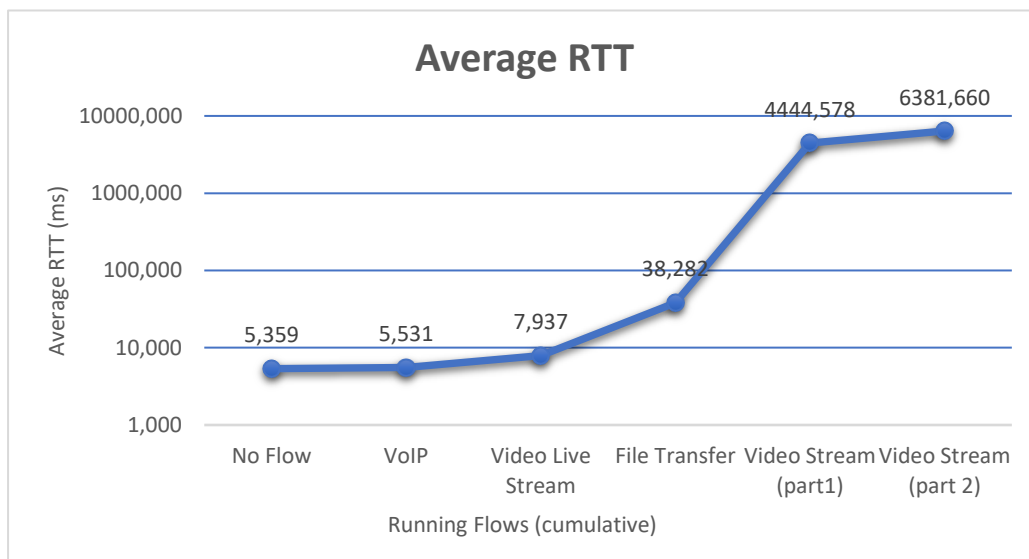
```

root@n23: /tmp/pycore.34839/n23.conf
File Edit View Search Terminal Help
64 bytes from 10.0.1.10: icmp_seq=95 ttl=63 time=6528 ms
64 bytes from 10.0.1.10: icmp_seq=96 ttl=63 time=6755 ms
64 bytes from 10.0.1.10: icmp_seq=97 ttl=63 time=6949 ms
64 bytes from 10.0.1.10: icmp_seq=98 ttl=63 time=7254 ms
64 bytes from 10.0.1.10: icmp_seq=99 ttl=63 time=7569 ms
64 bytes from 10.0.1.10: icmp_seq=100 ttl=63 time=7600 ms

--- 10.0.1.10 ping statistics ---
100 packets transmitted, 90 received, 10% packet loss, time 99389ms
rtt min/avg/max/mdev = 5433.465/6381.660/7600.613/452.314 ms, pipe 7
root@n23: /tmp/pycore.34839/n23.conf#

```

...όταν ενεργοποιούνται επιπλέον τα τελευταία δύο από τα Video Stream flows: "Video Stream (Part 2)"



β) Διαγράμματα μετρήσεων throughput για κάθε source-destination ζευγάρι κόμβων

Κατασκευάστηκε ένα διάγραμμα ξεχωριστά για κάθε ένα από τα 4 σενάρια επικοινωνίας (VoIP, Video Live Stream, File Transfer, Video Stream). Για την κατασκευή του **καθενός** από τα 4 αυτά διαγράμματα έγιναν τα εξής:

Step by step

Για τις τρεις περιπτώσεις No flow (κανένα flow ενεργοποιημένο), Current Flow Running (το εξεταζόμενο σε αυτό το διάγραμμα σενάριο επικοινωνίας - σενάριο από flows - είναι ενεργοποιημένο), All Flows Running (όλα τα flows του δικτύου είναι ενεργοποιημένα):

1. Χρησιμοποιήθηκε το εργαλείο iperf3 με server τον source κόμβο των flows και client τον destination κόμβο των flows.
2. Από τα αποτελέσματα λήφθηκαν από το terminal του client οι τιμές του bandwidth (throughput) του sender και receiver και οι τιμές αυτές χρησιμοποιούνται στο αντίστοιχο κάθε φορά διάγραμμα.
3. Οι τιμές του βήματος 2 συνιστούν το εκάστοτε διάγραμμα (4 συνολικά) που έγινε με τη χρήση Excel.

VoIP: No Flow, Current Flow Running, All Flows Running (εδώ ως source και destination θεωρήθηκαν αυτοί του ενός εκ των δύο flows)

```
root@n21:/tmp/pycore.34839/n21.conf# iperf3 -c 10.0.1.13 -p 3000
Connecting to host 10.0.1.13, port 3000
[ 4] local 10.0.2.10 port 38156 connected to 10.0.1.13 port 3000
[ ID] Interval      Transfer    Bandwidth  Retr  Cwnd
[ 4] 0.00-1.00 sec  526 KBytes  4.31 Mbits/sec  0    43.8 KBytes
[ 4] 1.00-2.00 sec  255 KBytes  2.09 Mbits/sec  0    55.1 KBytes
[ 4] 2.00-3.00 sec  318 KBytes  2.61 Mbits/sec  0    67.9 KBytes
[ 4] 3.00-4.00 sec  318 KBytes  2.61 Mbits/sec  0    79.2 KBytes
[ 4] 4.00-5.00 sec  255 KBytes  2.09 Mbits/sec  0    91.9 KBytes
[ 4] 5.00-6.00 sec  445 KBytes  3.65 Mbits/sec  0    129 KBytes
[ 4] 6.00-7.00 sec  509 KBytes  4.17 Mbits/sec  0    181 KBytes
[ 4] 7.00-8.00 sec  573 KBytes  4.69 Mbits/sec  0    249 KBytes
[ 4] 8.00-9.00 sec  509 KBytes  4.17 Mbits/sec  0    327 KBytes
[ 4] 9.00-10.00 sec 954 KBytes  7.80 Mbits/sec  0    438 KBytes

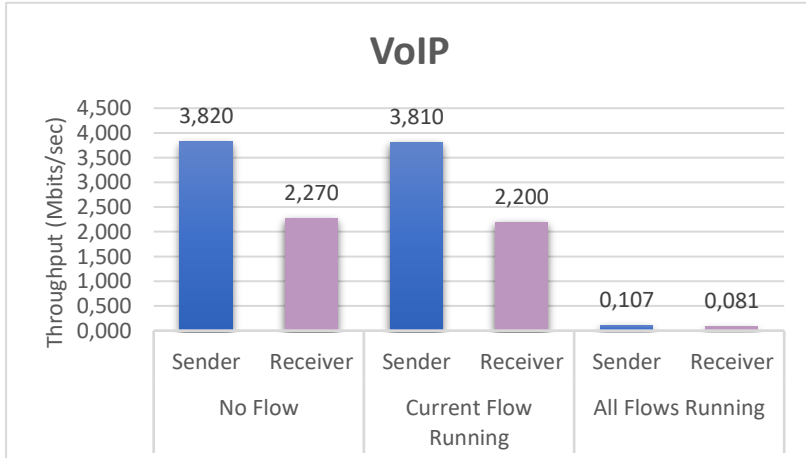
[ ID] Interval      Transfer    Bandwidth  Retr
[ 4] 0.00-10.00 sec 4.55 MBytes 3.82 Mbits/sec  0
[ 4] 0.00-10.00 sec 2.71 MBytes 2.27 Mbits/sec  0
sender
receiver
```

```
root@n21:/tmp/pycore.34839/n21.conf# iperf3 -c 10.0.1.13 -p 3000
Connecting to host 10.0.1.13, port 3000
[ 4] local 10.0.2.10 port 38168 connected to 10.0.1.13 port 3000
[ ID] Interval      Transfer    Bandwidth  Retr  Cwnd
[ 4] 0.00-1.00 sec  519 KBytes  4.25 Mbits/sec  0    43.8 KBytes
[ 4] 1.00-2.00 sec  255 KBytes  2.08 Mbits/sec  0    55.1 KBytes
[ 4] 2.00-3.00 sec  255 KBytes  2.08 Mbits/sec  0    65.0 KBytes
[ 4] 3.00-4.00 sec  318 KBytes  2.61 Mbits/sec  0    76.4 KBytes
[ 4] 4.00-5.00 sec  318 KBytes  2.60 Mbits/sec  0    90.5 KBytes
[ 4] 5.00-6.00 sec  382 KBytes  3.14 Mbits/sec  0    127 KBytes
[ 4] 6.00-7.00 sec  509 KBytes  4.16 Mbits/sec  0    180 KBytes
[ 4] 7.00-8.00 sec  636 KBytes  5.23 Mbits/sec  0    249 KBytes
[ 4] 8.00-9.00 sec  636 KBytes  5.21 Mbits/sec  0    328 KBytes
[ 4] 9.00-10.00 sec 827 KBytes  6.78 Mbits/sec  0    436 KBytes

[ ID] Interval      Transfer    Bandwidth  Retr
[ 4] 0.00-10.00 sec 4.55 MBytes 3.81 Mbits/sec  0
[ 4] 0.00-10.00 sec 2.63 MBytes 2.20 Mbits/sec  0
sender
receiver
```

```
root@n21:/tmp/pycore.34839/n21.conf# iperf3 -c 10.0.1.13 -p 3000
Connecting to host 10.0.1.13, port 3000
[ 4] local 10.0.2.10 port 38174 connected to 10.0.1.13 port 3000
[ ID] Interval      Transfer    Bandwidth  Retr  Cwnd
[ 4] 0.00-3.36 sec  76.4 KBytes  186 Kbits/sec  2    14.1 KBytes
[ 4] 3.36-3.36 sec  0.00 Bytes  0.00 bits/sec  0    14.1 KBytes
[ 4] 3.36-4.00 sec  0.00 Bytes  0.00 bits/sec  0    14.1 KBytes
[ 4] 4.00-5.00 sec  0.00 Bytes  0.00 bits/sec  0    14.1 KBytes
[ 4] 5.00-6.00 sec  0.00 Bytes  0.00 bits/sec  0    14.1 KBytes
[ 4] 6.00-7.00 sec  31.1 KBytes  255 Kbits/sec  0    28.3 KBytes
[ 4] 7.00-8.00 sec  0.00 Bytes  0.00 bits/sec  0    28.3 KBytes
[ 4] 8.00-9.00 sec  0.00 Bytes  0.00 bits/sec  0    28.3 KBytes
[ 4] 9.00-10.00 sec 22.6 KBytes  185 Kbits/sec  0    56.6 KBytes

[ ID] Interval      Transfer    Bandwidth  Retr
[ 4] 0.00-10.00 sec  130 KBytes  107 Kbits/sec  2
[ 4] 0.00-10.00 sec 99.0 KBytes  81.1 Kbits/sec  0
sender
receiver
```



Video Live Stream: No Flow, Current Flow Running, All Flows Running

```
root@n22:/tmp/pycore.34839/n22.conf# iperf3 -c 10.0.1.10 -p 3000
Connecting to host 10.0.1.10, port 3000
[ 4] local 10.0.2.11 port 43534 connected to 10.0.1.10 port 3000
[ ID] Interval      Transfer    Bandwidth  Retr  Cwnd
[ 4] 0.00-1.00    sec 519 KBytes 4.25 Mbits/sec  0  43.8 KBytes
[ 4] 1.00-2.00    sec 255 KBytes 2.09 Mbits/sec  0  55.1 KBytes
[ 4] 2.00-3.00    sec 318 KBytes 2.60 Mbits/sec  0  67.9 KBytes
[ 4] 3.00-4.00    sec 318 KBytes 2.61 Mbits/sec  0  79.2 KBytes
[ 4] 4.00-5.00    sec 318 KBytes 2.61 Mbits/sec  0  91.9 KBytes
[ 4] 5.00-6.00    sec 382 KBytes 3.13 Mbits/sec  0  127 KBytes
[ 4] 6.00-7.00    sec 509 KBytes 4.17 Mbits/sec  0  181 KBytes
[ 4] 7.00-8.00    sec 636 KBytes 5.21 Mbits/sec  0  247 KBytes
[ 4] 8.00-9.00    sec 636 KBytes 5.21 Mbits/sec  0  325 KBytes
[ 4] 9.00-10.00   sec 764 KBytes 6.25 Mbits/sec  0  436 KBytes

[ ID] Interval      Transfer    Bandwidth  Retr
[ 4] 0.00-10.00   sec 4.55 MBytes 3.81 Mbits/sec  0      sender
[ 4] 0.00-10.00   sec 2.70 MBytes 2.27 Mbits/sec              receiver
```

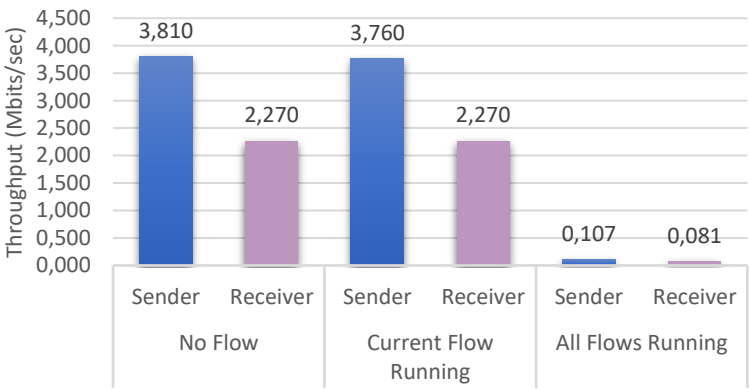
```
root@n22:/tmp/pycore.34839/n22.conf# iperf3 -c 10.0.1.10 -p 3000
Connecting to host 10.0.1.10, port 3000
[ 4] local 10.0.2.11 port 43546 connected to 10.0.1.10 port 3000
[ ID] Interval      Transfer    Bandwidth  Retr  Cwnd
[ 4] 0.00-1.00    sec 76.4 KBytes 623 Kbits/sec  3  14.1 KBytes
[ 4] 1.00-2.00    sec 0.00 Bytes 0.00 bits/sec  0  14.1 KBytes
[ 4] 2.00-3.00    sec 0.00 Bytes 0.00 bits/sec  0  14.1 KBytes
[ 4] 3.00-4.00    sec 31.1 KBytes 254 Kbits/sec  0  28.3 KBytes
[ 4] 4.00-5.00    sec 0.00 Bytes 0.00 bits/sec  0  28.3 KBytes
[ 4] 5.00-6.00    sec 0.00 Bytes 0.00 bits/sec  0  28.3 KBytes
[ 4] 6.00-7.00    sec 0.00 Bytes 0.00 bits/sec  0  28.3 KBytes
[ 4] 7.00-8.00    sec 22.6 KBytes 186 Kbits/sec  0  56.6 KBytes
[ 4] 8.00-9.00    sec 0.00 Bytes 0.00 bits/sec  0  56.6 KBytes
[ 4] 9.00-10.00   sec 0.00 Bytes 0.00 bits/sec  0  56.6 KBytes

[ ID] Interval      Transfer    Bandwidth  Retr
[ 4] 0.00-10.00   sec 130 KBytes 107 Kbits/sec  3      sender
[ 4] 0.00-10.00   sec 99.0 KBytes 81.1 Kbits/sec              receiver
```

```
root@n22:/tmp/pycore.34839/n22.conf# iperf3 -c 10.0.1.10 -p 3000
Connecting to host 10.0.1.10, port 3000
[ 4] local 10.0.2.11 port 43538 connected to 10.0.1.10 port 3000
[ ID] Interval      Transfer    Bandwidth  Retr  Cwnd
[ 4] 0.00-1.00    sec 519 KBytes 4.25 Mbits/sec  0  43.8 KBytes
[ 4] 1.00-2.00    sec 255 KBytes 2.08 Mbits/sec  0  55.1 KBytes
[ 4] 2.00-3.00    sec 318 KBytes 2.61 Mbits/sec  0  67.9 KBytes
[ 4] 3.00-4.00    sec 318 KBytes 2.61 Mbits/sec  0  79.2 KBytes
[ 4] 4.00-5.00    sec 255 KBytes 2.09 Mbits/sec  0  91.9 KBytes
[ 4] 5.00-6.00    sec 445 KBytes 3.65 Mbits/sec  0  129 KBytes
[ 4] 6.00-7.00    sec 509 KBytes 4.17 Mbits/sec  0  181 KBytes
[ 4] 7.00-8.00    sec 636 KBytes 5.20 Mbits/sec  0  249 KBytes
[ 4] 8.00-9.00    sec 636 KBytes 5.22 Mbits/sec  0  327 KBytes
[ 4] 9.00-10.00   sec 700 KBytes 5.72 Mbits/sec  0  437 KBytes

[ ID] Interval      Transfer    Bandwidth  Retr
[ 4] 0.00-10.00   sec 4.48 MBytes 3.76 Mbits/sec  0      sender
[ 4] 0.00-10.00   sec 2.71 MBytes 2.27 Mbits/sec              receiver
```

Video Live Stream



File Transfer: No Flow, Current Flow Running, All Flows Running

```
root@n11:/tmp/pycore.34839/n11.conf# iperf3 -c 10.0.0.10 -p 3000
Connecting to host 10.0.0.10, port 3000
[ 4] local 10.0.1.10 port 48892 connected to 10.0.0.10 port 3000
[ ID] Interval      Transfer    Bandwidth  Retr  Cwnd
[ 4] 0.00-1.00    sec 519 KBytes 4.25 Mbits/sec  0  43.8 KBytes
[ 4] 1.00-2.00    sec 255 KBytes 2.09 Mbits/sec  0  55.1 KBytes
[ 4] 2.00-3.00    sec 318 KBytes 2.60 Mbits/sec  0  67.9 KBytes
[ 4] 3.00-4.00    sec 318 KBytes 2.61 Mbits/sec  0  79.2 KBytes
[ 4] 4.00-5.00    sec 255 KBytes 2.09 Mbits/sec  0  91.9 KBytes
[ 4] 5.00-6.00    sec 445 KBytes 3.65 Mbits/sec  0  129 KBytes
[ 4] 6.00-7.00    sec 509 KBytes 4.17 Mbits/sec  0  181 KBytes
[ 4] 7.00-8.00    sec 636 KBytes 5.21 Mbits/sec  0  250 KBytes
[ 4] 8.00-9.00    sec 636 KBytes 5.21 Mbits/sec  0  328 KBytes
[ 4] 9.00-10.00   sec 827 KBytes 6.77 Mbits/sec  0  436 KBytes

[ ID] Interval      Transfer    Bandwidth  Retr
[ 4] 0.00-10.00   sec 4.61 MBytes 3.86 Mbits/sec  0      sender
[ 4] 0.00-10.00   sec 2.70 MBytes 2.27 Mbits/sec              receiver
```

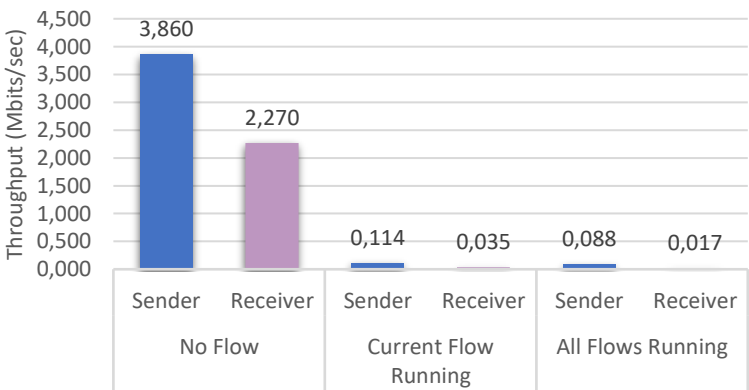
```
root@n11:/tmp/pycore.34839/n11.conf# iperf3 -c 10.0.0.10 -p 3000
Connecting to host 10.0.0.10, port 3000
[ 4] local 10.0.1.10 port 48904 connected to 10.0.0.10 port 3000
[ ID] Interval      Transfer    Bandwidth  Retr  Cwnd
[ 4] 0.00-1.00    sec 76.4 KBytes 624 Kbits/sec  1  14.1 KBytes
[ 4] 1.00-2.00    sec 0.00 Bytes 0.00 bits/sec  0  14.1 KBytes
[ 4] 2.00-3.00    sec 0.00 Bytes 0.00 bits/sec  0  14.1 KBytes
[ 4] 3.00-4.00    sec 0.00 Bytes 0.00 bits/sec  0  14.1 KBytes
[ 4] 4.00-5.00    sec 0.00 Bytes 0.00 bits/sec  0  14.1 KBytes
[ 4] 5.00-6.00    sec 0.00 Bytes 0.00 bits/sec  0  14.1 KBytes
[ 4] 6.00-7.00    sec 0.00 Bytes 0.00 bits/sec  0  14.1 KBytes
[ 4] 7.00-8.00    sec 0.00 Bytes 0.00 bits/sec  0  14.1 KBytes
[ 4] 8.00-9.00    sec 0.00 Bytes 0.00 bits/sec  0  14.1 KBytes
[ 4] 9.00-10.00   sec 31.1 KBytes 255 Kbits/sec  0  17.0 KBytes

[ ID] Interval      Transfer    Bandwidth  Retr
[ 4] 0.00-10.00   sec 107 KBytes 88.0 Kbits/sec  1      sender
[ 4] 0.00-10.00   sec 21.2 KBytes 17.4 Kbits/sec              receiver
```

```
root@n11:/tmp/pycore.34839/n11.conf# iperf3 -c 10.0.0.10 -p 3000
Connecting to host 10.0.0.10, port 3000
[ 4] local 10.0.1.10 port 48964 connected to 10.0.0.10 port 3000
[ ID] Interval      Transfer    Bandwidth  Retr  Cwnd
[ 4] 0.00-4.41    sec 76.4 KBytes 142 Kbits/sec  2  14.1 KBytes
[ 4] 4.41-4.41    sec 0.00 Bytes 0.00 bits/sec  0  14.1 KBytes
[ 4] 4.41-4.41    sec 0.00 Bytes 0.00 bits/sec  0  14.1 KBytes
[ 4] 4.41-4.41    sec 0.00 Bytes 0.00 bits/sec  0  14.1 KBytes
[ 4] 4.41-5.00    sec 0.00 Bytes 0.00 bits/sec  0  14.1 KBytes
[ 4] 5.00-6.00    sec 0.00 Bytes 0.00 bits/sec  0  14.1 KBytes
[ 4] 6.00-7.00    sec 0.00 Bytes 0.00 bits/sec  0  14.1 KBytes
[ 4] 7.00-8.00    sec 0.00 Bytes 0.00 bits/sec  0  14.1 KBytes
[ 4] 8.00-9.00    sec 62.2 KBytes 510 Kbits/sec  0  28.3 KBytes
[ 4] 9.00-10.00   sec 0.00 Bytes 0.00 bits/sec  0  28.3 KBytes

[ ID] Interval      Transfer    Bandwidth  Retr
[ 4] 0.00-10.00   sec 139 KBytes 114 Kbits/sec  2      sender
[ 4] 0.00-10.00   sec 42.4 KBytes 34.8 Kbits/sec              receiver
```

File Transfer

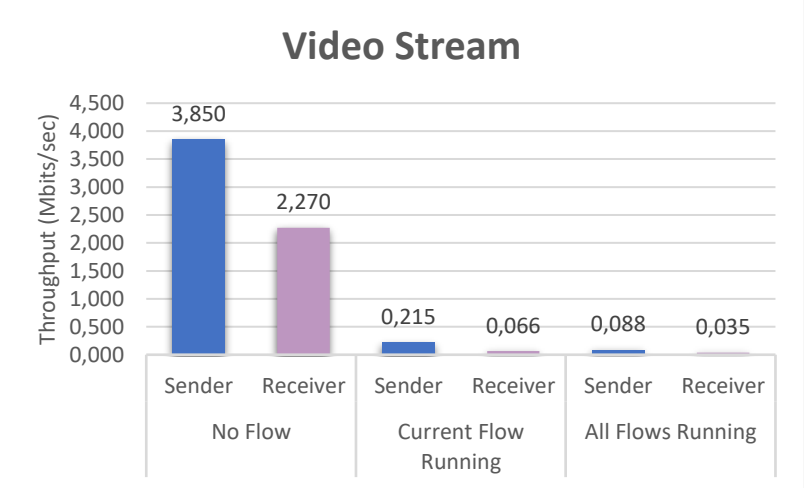


Video Stream (on demand): No Flow, Current Flow Running, All Flows Running

```
root@n31:/tmp/pycore.34839/n31.conf# iperf3 -c 10.0.1.10 -p 3000
Connecting to host 10.0.1.10, port 3000
[ 4] local 10.0.3.10 port 35042 connected to 10.0.1.10 port 3000
[ ID] Interval      Transfer    Bandwidth  Retr  Cwnd
[ 4] 0.00-1.00    sec    498 KBytes  4.08 Mbits/sec    0   43.8 KBytes
[ 4] 1.00-2.00    sec    318 KBytes  2.61 Mbits/sec    0   55.1 KBytes
[ 4] 2.00-3.00    sec    255 KBytes  2.08 Mbits/sec    0   67.9 KBytes
[ 4] 3.00-4.00    sec    318 KBytes  2.61 Mbits/sec    0   79.2 KBytes
[ 4] 4.00-5.00    sec    318 KBytes  2.61 Mbits/sec    0   91.9 KBytes
[ 4] 5.00-6.00    sec    445 KBytes  3.65 Mbits/sec    0   129 KBytes
[ 4] 6.00-7.00    sec    509 KBytes  4.17 Mbits/sec    0   181 KBytes
[ 4] 7.00-8.00    sec    573 KBytes  4.69 Mbits/sec    0   249 KBytes
[ 4] 8.00-9.00    sec    636 KBytes  5.21 Mbits/sec    0   327 KBytes
[ 4] 9.00-10.00   sec    827 KBytes  6.78 Mbits/sec    0   436 KBytes
--
[ ID] Interval      Transfer    Bandwidth  Retr
[ 4] 0.00-10.00   sec    4.59 MBytes  3.85 Mbits/sec    0      sender
[ 4] 0.00-10.00   sec    2.70 MBytes  2.27 Mbits/sec              receiver
```

```
root@n31:/tmp/pycore.34839/n31.conf# iperf3 -c 10.0.1.10 -p 3000
Connecting to host 10.0.1.10, port 3000
[ 4] local 10.0.3.10 port 35052 connected to 10.0.1.10 port 3000
[ ID] Interval      Transfer    Bandwidth  Retr  Cwnd
[ 4] 0.00-4.15    sec    76.4 KBytes  151 Kbits/sec    2   14.1 KBytes
[ 4] 4.15-4.15    sec     0.00 Bytes  0.00 bits/sec    0   14.1 KBytes
[ 4] 4.15-4.15    sec     0.00 Bytes  0.00 bits/sec    0   14.1 KBytes
[ 4] 4.15-4.15    sec     0.00 Bytes  0.00 bits/sec    0   14.1 KBytes
[ 4] 4.15-5.00    sec     0.00 Bytes  0.00 bits/sec    0   14.1 KBytes
[ 4] 5.00-6.00    sec     0.00 Bytes  0.00 bits/sec    0   14.1 KBytes
[ 4] 6.00-7.00    sec     0.00 Bytes  0.00 bits/sec    0   14.1 KBytes
[ 4] 7.00-8.00    sec    31.1 KBytes  256 Kbits/sec    0   28.3 KBytes
[ 4] 8.00-9.00    sec     0.00 Bytes  0.00 bits/sec    0   28.3 KBytes
[ 4] 9.00-10.00   sec     0.00 Bytes  0.00 bits/sec    0   28.3 KBytes
--
[ ID] Interval      Transfer    Bandwidth  Retr
[ 4] 0.00-10.00   sec    107 KBytes  88.0 Kbits/sec    2      sender
[ 4] 0.00-10.00   sec    42.4 KBytes  34.7 Kbits/sec              receiver
```

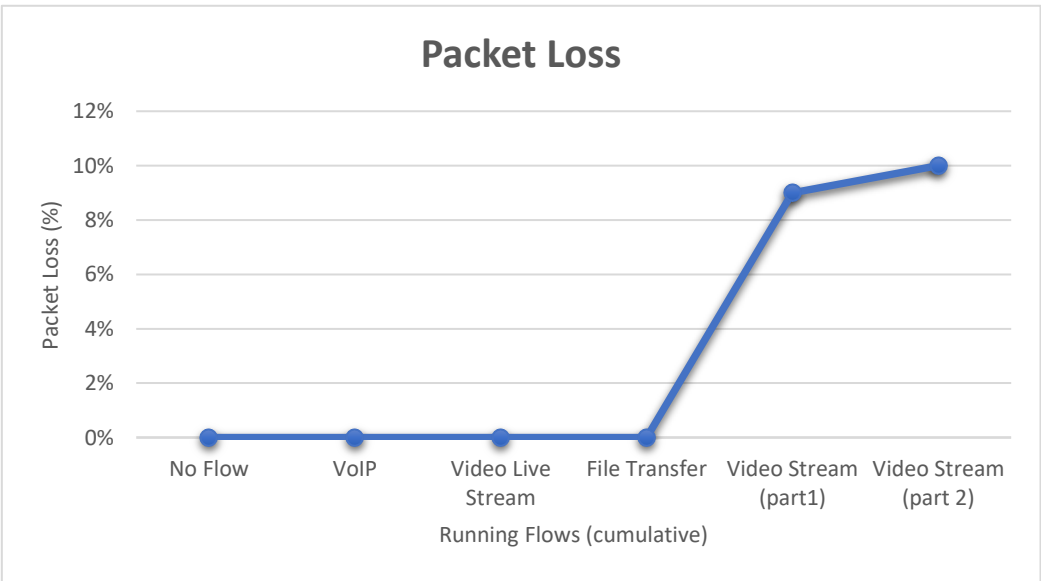
```
root@n31:/tmp/pycore.34839/n31.conf# iperf3 -c 10.0.1.10 -p 3000
Connecting to host 10.0.1.10, port 3000
[ 4] local 10.0.3.10 port 35046 connected to 10.0.1.10 port 3000
[ ID] Interval      Transfer    Bandwidth  Retr  Cwnd
[ 4] 0.00-2.72    sec    76.4 KBytes  230 Kbits/sec    1   14.1 KBytes
[ 4] 2.72-2.72    sec     0.00 Bytes  0.00 bits/sec    0   14.1 KBytes
[ 4] 2.72-3.00    sec     0.00 Bytes  0.00 bits/sec    0   14.1 KBytes
[ 4] 3.00-4.00    sec     0.00 Bytes  0.00 bits/sec    0   14.1 KBytes
[ 4] 4.00-5.01    sec     0.00 Bytes  0.00 bits/sec    0   14.1 KBytes
[ 4] 5.01-6.00    sec    31.1 KBytes  256 Kbits/sec    0   28.3 KBytes
[ 4] 6.00-7.00    sec     0.00 Bytes  0.00 bits/sec    0   28.3 KBytes
[ 4] 7.00-8.00    sec     0.00 Bytes  0.00 bits/sec    0   28.3 KBytes
[ 4] 8.00-9.00    sec     0.00 Bytes  0.00 bits/sec    0   33.9 KBytes
[ 4] 9.00-10.00   sec    156 KBytes  1.28 Mbits/sec    0   38.2 KBytes
--
[ ID] Interval      Transfer    Bandwidth  Retr
[ 4] 0.00-10.00   sec    263 KBytes  215 Kbits/sec    1      sender
[ 4] 0.00-10.00   sec    80.6 KBytes  66.0 Kbits/sec              receiver
```



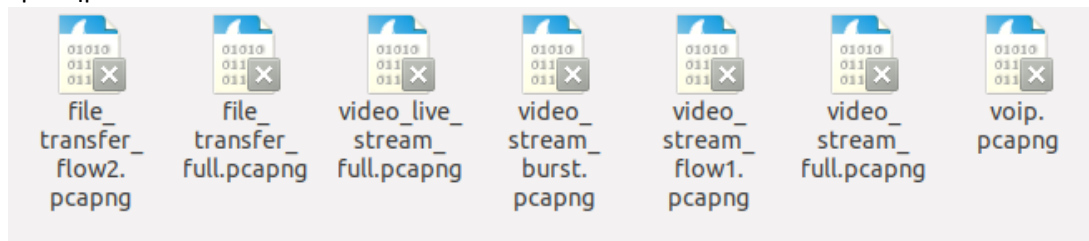
γ) Διάγραμμα μέτρησης packet loss

Αξιοποιούμε τις μετρήσεις του ερωτήματος (α) αφού το εργαλείο ring μας πληροφορεί και για το ποσοστό των πακέτων που χάθηκαν.

Αυξάνοντας λοιπόν διαδοχικά τα ενεργά flows (1. No flow – 2. VoIP – 3. Video Live Stream – 4. File Transfer – 5. Video Stream (Part 1) – 6. Video Stream (Part 2)) και συνεπώς την κίνηση στο δίκτυο, οι μετρήσεις packet loss (%) που λάβαμε ήταν όπως φαίνονται στο διάγραμμα:



Δ. Ένα screenshot από τα εξαχθέντα .pcap files που χρησιμοποιήθηκαν για το επόμενο ερώτημα:

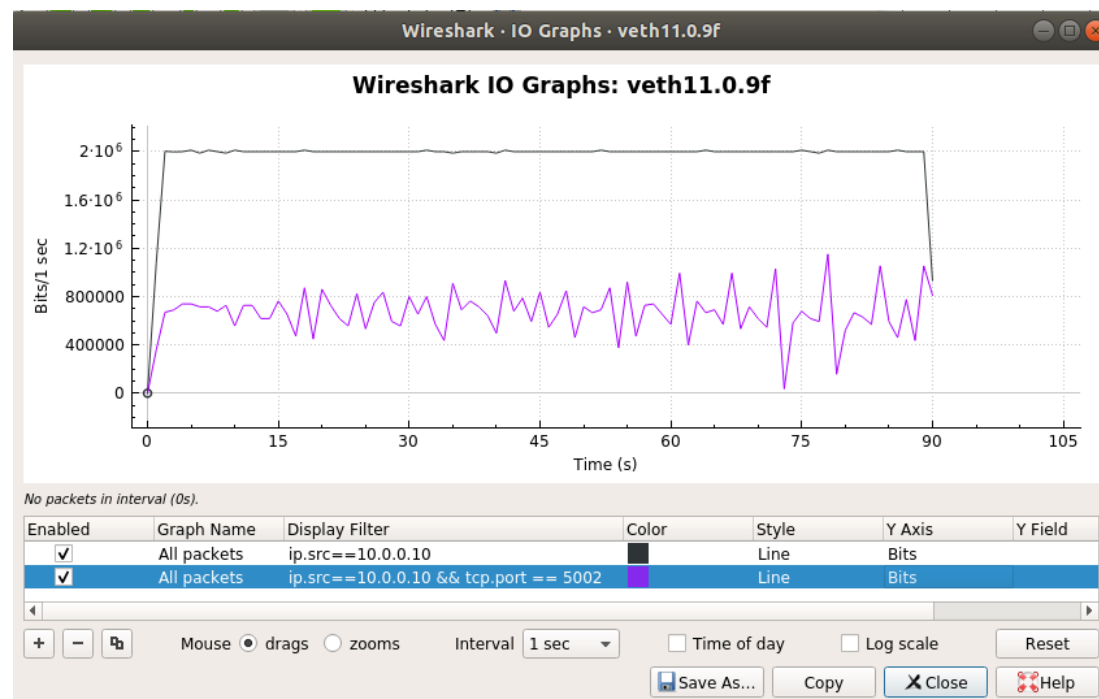


Ε.

File Transfer

Καταγραφή με τα 3 flows ενεργά
(File Transfer TCP n01-n11_1/_2/_3)

Ενεργοποιώντας και τα τρία flows που χρησιμοποιούνται για την μεταφορά αρχείων (κόμβος n01 του υποδικτύου Α προς κόμβο n11 του υποδικτύου Β), και με χρήση του εργαλείου Statistics/IO Graphs παρατηρούμε την εξής κίνηση:

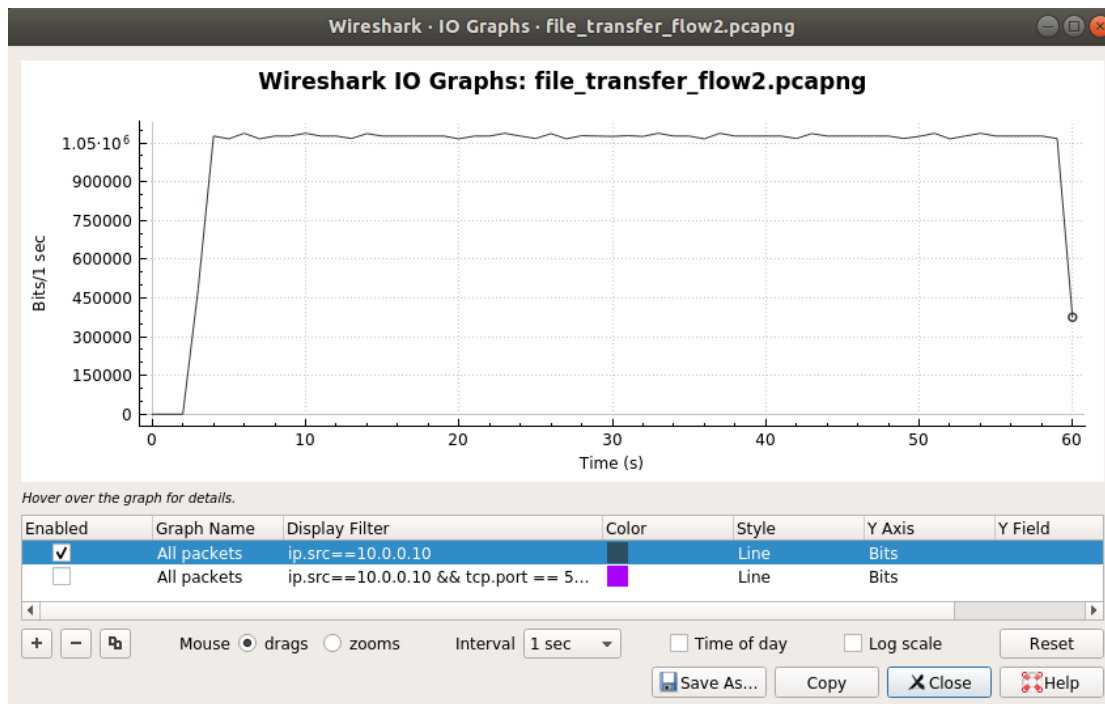


Από τη μαύρη γραμμή του διαγράμματος της οποίας η τιμή είναι στα 2Mbps επιβεβαιώνεται ότι λόγω του περιορισμού του bandwidth των γραμμών (2000Kbps=2Mbps) δεν χωράει η εισερχόμενη κίνηση των 3x1024Kbps (τα 3 file transfer flows) και συνεπώς το όριο αυτό δεν ξεπερνάται. (Χρησιμοποιείται filter *ip.src == 10.0.0.10*, δηλ η IP του n01, για να απομονωθεί και εξεταστεί η εισερχόμενη προς το n11 κίνηση).

Η μωβ γραμμή αναδεικνύει το παραπάνω γεγονός αφού έχοντας απομονώσει το ένα εκ των τριών flows (τίθεται επιπλέον filter *tcp.port == 5002*) παρατηρούμε ότι έχει διακυμάνσεις (αντίθετα από τη φύση του ως PERIODIC) γιατί τα 3 flows ανταγωνίζονται συνεχώς για το περιορισμένο bandwidth.

Καταγραφή με το 1 εκ των τριών flows ενεργό

Απομονώνουμε το ένα εκ των τριών (που θα χωράει τώρα στη γραμμή), ώστε να μπορέσουμε να διακρίνουμε τα patterns που το ταυτοποιούν. Παίρνουμε π.χ. παρακάτω το flow με source port 5002 (τα άλλα 2 flows θα είχαν όμοια συμπεριφορά).



(Χρησιμοποιείται filter `ip.src == 10.0.0.10`, δηλ η IP του n01, για να απομονωθεί και εξεταστεί η εισερχόμενη προς το n11 κίνηση).

! Υπενθύμιση: «PERIODIC [100.0 1280]» και TCP.

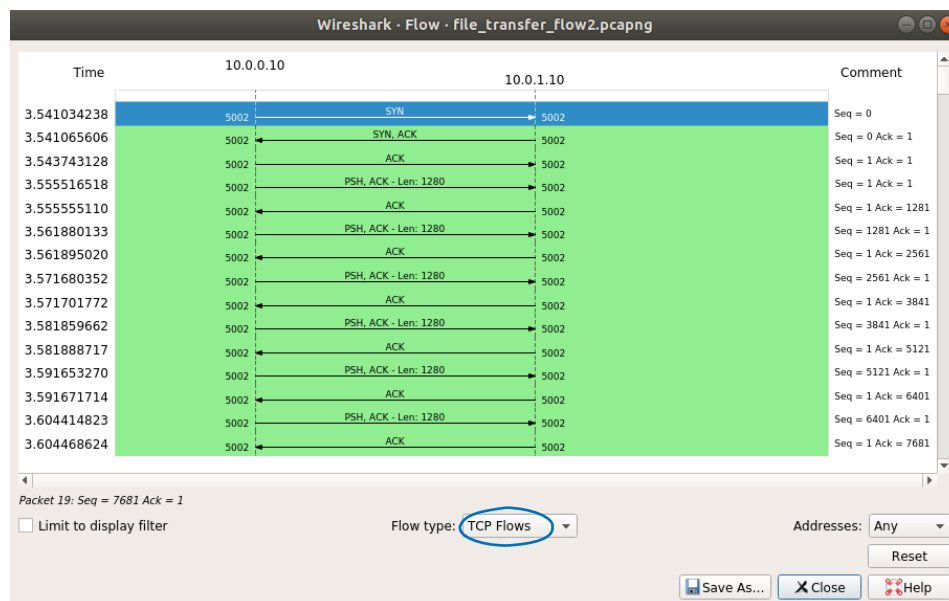
- ✓ Στο διάγραμμα φαίνεται ένα αρκετά σταθερό bitrate στα 1Mbps που συμφωνεί πλήρως με το γεγονός ότι το pattern είναι PERIODIC, και το υπολογιζόμενο bitrate του είναι:

$$100 \text{ messages/sec} \times 1280 \text{ Bytes/message} \times 8 = 1\text{Mbps}$$

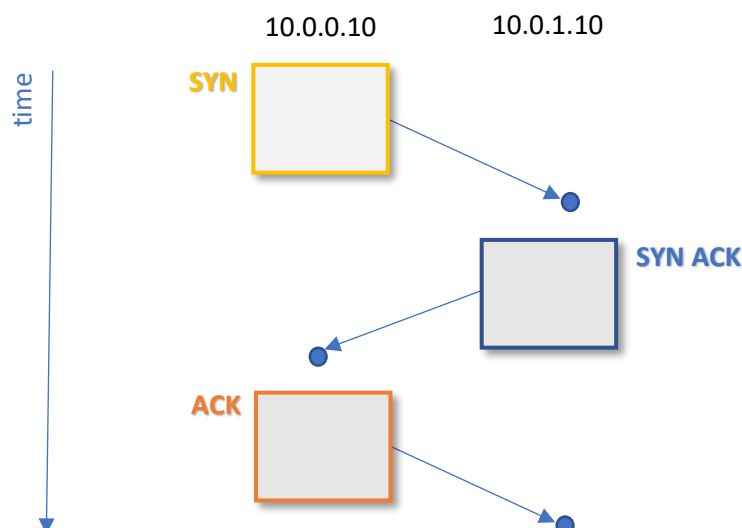
- ✓ Επίσης φαίνεται το μέγεθος των μηνυμάτων ότι είναι όσο είχε καθοριστεί στο pattern, δηλ 1280 Bytes:

```
▼ Transmission Control Protocol, Src Port: 5002, Dst Port: 5002, Seq: 8961, Ack: 1, Len: 1280
  Source Port: 5002
  Destination Port: 5002
  [Stream index: 0]
  [TCP Segment Len: 1280]
  Sequence number: 8961 (relative sequence number)
  [Next sequence number: 10241 (relative sequence number)]
  Acknowledgment number: 1 (relative ack number)
  1000 .... = Header Length: 32 bytes (8)
```

Ως προς το πρωτόκολλο μεταφοράς, επιβεβαιώνονται όσα θα αναμέναμε για το TCP, αφού με τη χρήση του εργαλείου Statistics/Flow Graph παρατηρούμε τα εξής:



- ✓ Τριπλή χειραψία για την εγκαθίδρυση σύνδεσης:
 - Ο client (n01) στέλνει πακέτο στον server (n11) με sequence num = 0 και **SYN** bit=1.
 - Ο server απαντάει στον client με ACK πακέτο με sequence num=0, **SYN** bit=1 και **ACK** bit=1, ACK num = 1. Εγκαθιδρύεται η σύνδεση στον client μόλις αυτό ληφθεί.
 - Ο client επιβεβαιώνει με ACK πακέτο με sequence num=1 και **ACK** bit=1, ACK num= 1. Εγκαθιδρύεται η σύνδεση στον server μόλις αυτό ληφθεί.



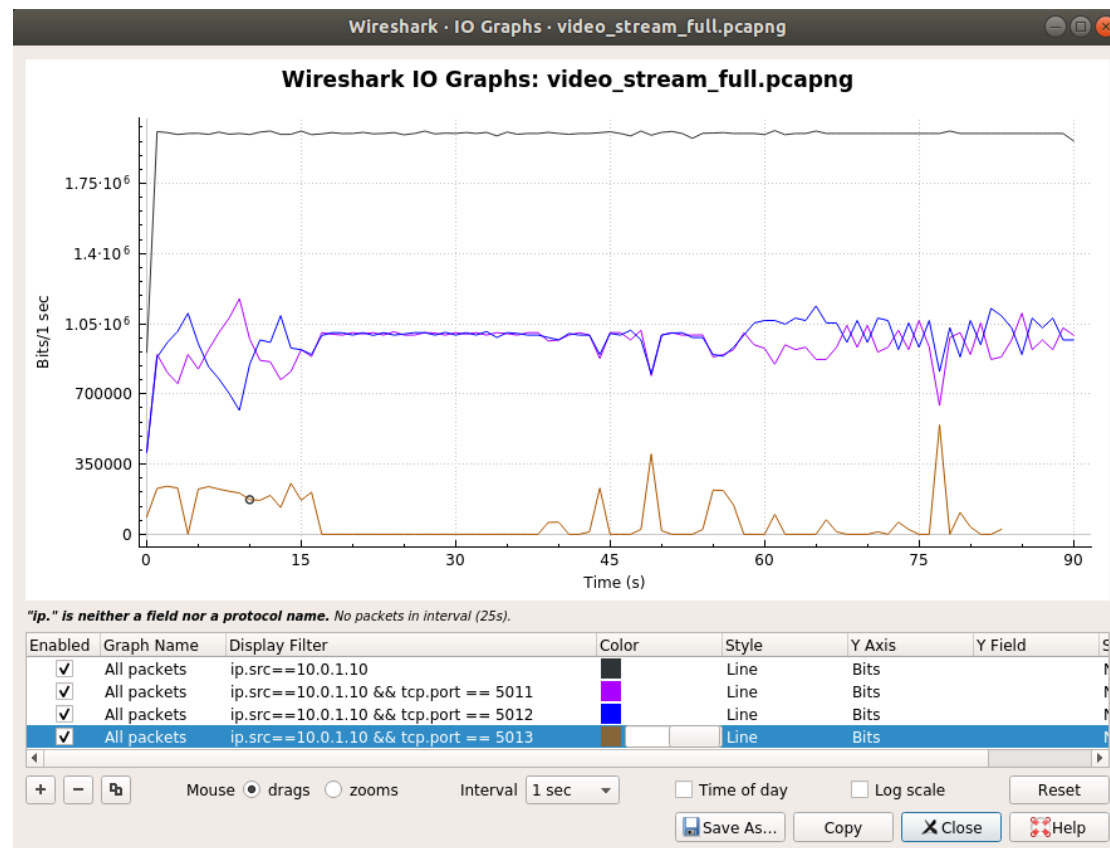
- ✓ Υπάρχει επιβεβαίωση των πακέτων με ACK. Στο Comment κομμάτι του screenshot φαίνεται η ακολουθία των sequence numbers και ACK numbers, όπου εφόσον κάθε πακέτο έχει μέγεθος 1280 Bytes, τα ACK κάθε φορά ζητούν το επόμενο αναμενόμενο Byte (TCP) που προκύπτει να είναι +1280 από το ληφθέν.

On demand Video Streaming

Καταγραφή με τα 3 flows ενεργά

(Video Stream TCP n11-n31_1/_2/_3)

Ενεργοποιώντας και τα τρία flows που χρησιμοποιούνται για την μεταφορά αρχείων (κόμβος n11 του υποδικτύου B προς κόμβο n31 του υποδικτύου D), και με χρήση του εργαλείου Statistics/IO Graphs παρατηρούμε την εξής κίνηση:

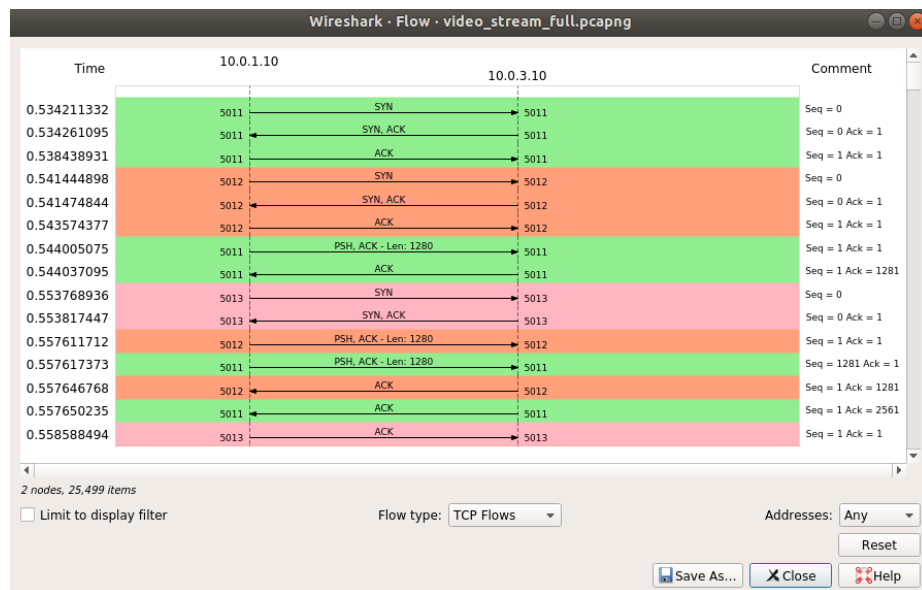


Από τη μαύρη γραμμή του διαγράμματος της οποίας η τιμή είναι στα 2Mbps επιβεβαιώνεται ότι λόγω του περιορισμού του bandwidth των γραμμών (2000Kbps=2Mbps) πιθανόν να μην χωράει η εισερχόμενη κίνηση των 2x1024 + burst Kbps (τα 3 on demand video streaming flows) και συνεπώς το όριο αυτό δεν ξεπερνάται. (Χρησιμοποιείται filter *ip.src == 10.0.1.10*, δλδ η IP του n11, για να απομονωθεί και εξεταστεί η εισερχόμενη προς το n31 κίνηση).

Οι μωβ και μπλε γραμμές αναδεικνύουν το παραπάνω γεγονός αφού έχοντας απομονώσει το ένα εκ των PERIODIC flows κάθε φορά (τίθεται επιπλέον filter *tcp.port == [...] κάθε φορά*) παρατηρούμε ότι έχει διακυμάνσεις (αντίθετα από τη φύση του ως PERIODIC) γιατί τα 3 flows ανταγωνίζονται συνεχώς για το περιορισμένο bandwidth.

Η καφέ γραμμή αντιστοιχεί στο BURST flow το οποίο κινείται ακανόνιστα χαμηλότερα από τα άλλα δύο.

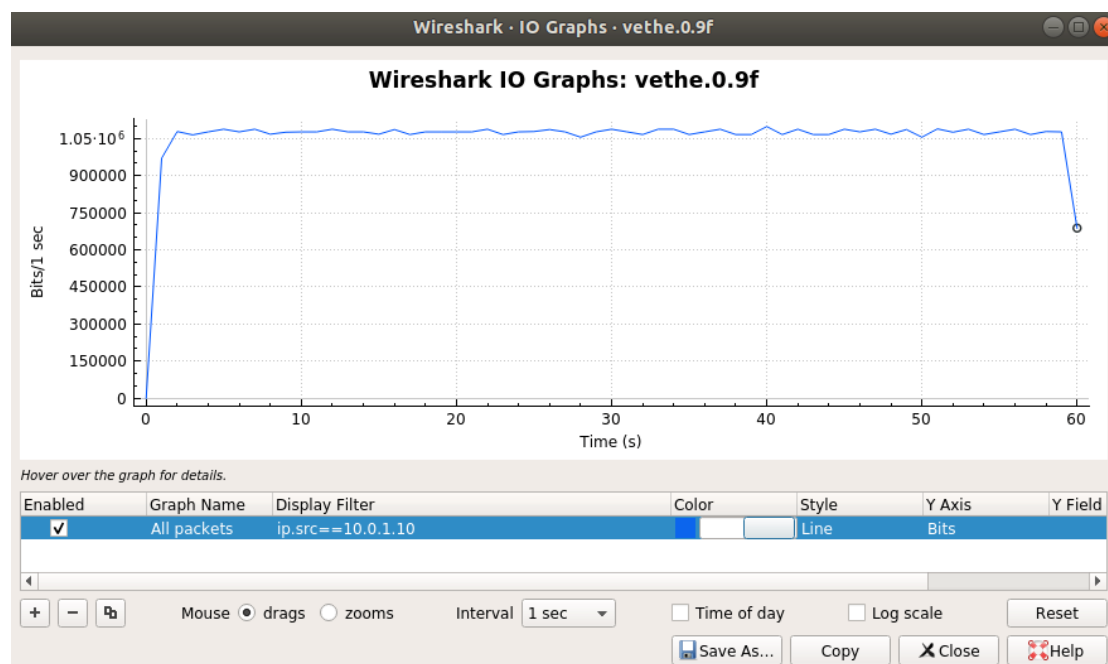
Με χρήση του εργαλείου Statistics/Flow Grpahs:



Η ανάλυση είναι η ίδια που έγινε προηγουμένως για την ταυτοποίηση του TCP πρωτοκόλλου, παρατηρούμε όμως επιπλέον ότι εφόσον τρέχουν και τα 3 flows στην καταγραφή και είναι και τα τρία TCP ανοίγουν **3 συνδέσεις** (ports 5011, 5012, 5013, **τριπλή χειραψία 3 φορές**) και η ακολουθία πακέτων στη συνέχεια φαίνεται με διαφορετικό χρώμα για κάθε σύνδεση/flow.

Καταγραφή με ένα εκ των δύο PERIODIC flows ενεργό

Απομονώνουμε το ένα εκ των δύο PERIODIC flows (που θα χωράει τώρα στη γραμμή), ώστε να μπορέσουμε να διακρίνουμε τα patterns που το ταυτοποιούν (το άλλο flow θα είχε όμοια συμπεριφορά).



! Υπενθύμιση: «PERIODIC [100.0 1280]»

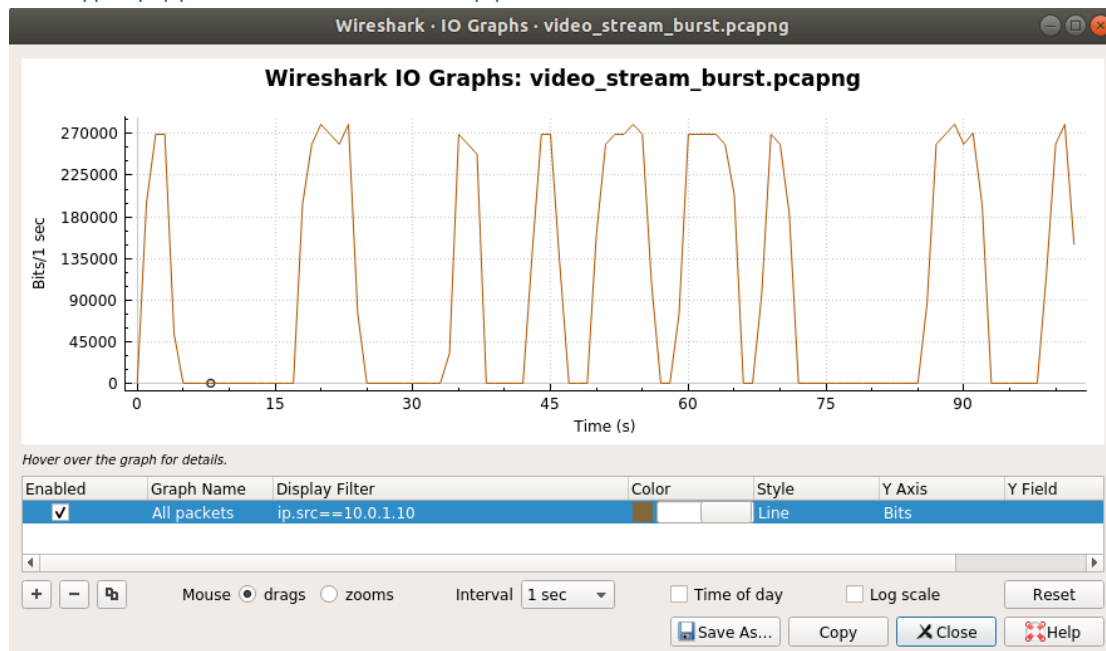
- ✓ Στο διάγραμμα φαίνεται ένα αρκετά σταθερό bitrate στα 1Mbps που συμφωνεί πλήρως με το γεγονός ότι το pattern είναι PERIODIC, και το υπολογιζόμενο bitrate του είναι:

$$100 \text{ messages/sec} \times 1280 \text{ Bytes/message} \times 8 = 1\text{Mbps}$$

- ✓ Επίσης φαίνεται το μέγεθος των μηνυμάτων ότι είναι όσο είχε καθοριστεί στο pattern, δηλ 1280 Bytes:

```
▼ Transmission Control Protocol, Src Port: 5011, Dst Port: 5011, Seq: 2671361, Ack: 1, Len: 1280
  Source Port: 5011
  Destination Port: 5011
  [Stream index: 0]
  [TCP Segment Len: 1280]
```

Καταγραφή με το BURST flow ενεργό



! Υπενθύμιση: «BURST [RANDOM 10.0 PERIODIC [25.0 1280] FIXED 3.0]» και TCP

- ✓ Από τη μορφή του διαγράμματος επιβεβαιώνεται ότι πρόκειται για BURST pattern, εφόσον έχουμε ριπές αποστολής μηνυμάτων που ακολουθούνται από χρονικά διαστήματα απουσίας δραστηριότητας. Παρατηρούμε ότι η διάρκεια των ριπών όπως φαίνεται από το πλάτος που έχουν τα «βουνά» που σχηματίζονται είναι κοντά στις ρυθμίσεις (~3 sec), όπως επίσης και η συχνότητα δημιουργίας ριπής (κατά μέσο όρο 10 sec) που γίνεται αντιληπτή στον οριζόντιο άξονα βλέποντας τα χρονικά διαστήματα με bitrate 0. Επιπλέον κάθε ριπή μεμονωμένα έχει PERIODIC pattern αφού είναι σχετικά ομαλές οι κορυφές των «βουνών» (πρέπει να ληφθεί υπόψιν και το εύρος των τιμών στον άξονα γ). Το ύψος των ριπών βρίσκεται κοντά στη ρύθμιση των:

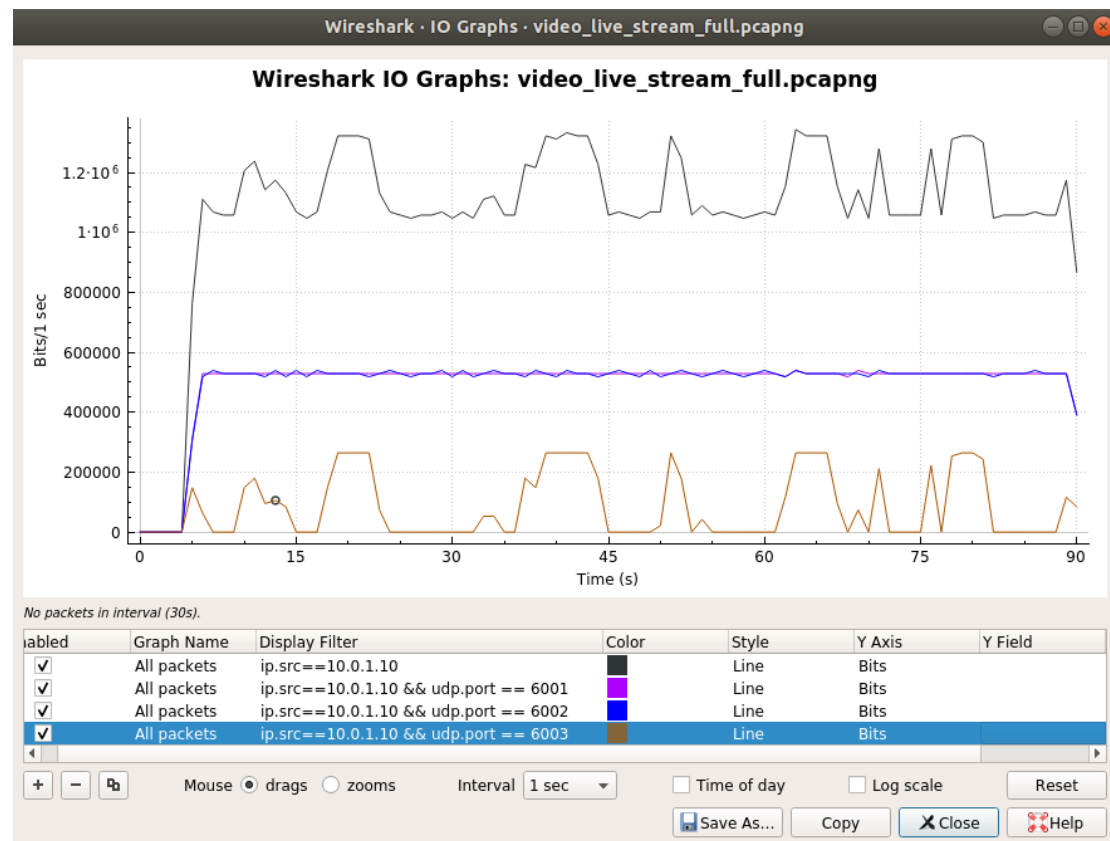
$$25 \text{ messages/sec} \times 1280 \text{ Bytes/message} \times 8 = 256\text{Kbps}$$

Live Video Streaming

Καταγραφή με τα 3 flows ενεργά

(Video Stream TCP n11-n22_1/_2/_3)

Ανοίγοντας και τα τρία flows που χρησιμοποιούνται για την μεταφορά αρχείων (κόμβος n11 του υποδικτύου B προς κόμβο n22 του υποδικτύου C), και με χρήση του εργαλείου Statistics/IO Graphs παρατηρούμε την εξής κίνηση:



! Υπενθυμίσεις: «PERIODIC [50.0 1280]» και UDP

«BURST [RANDOM 3.0 PERIODIC [25.0 1280] EXP 1.0]» και UDP

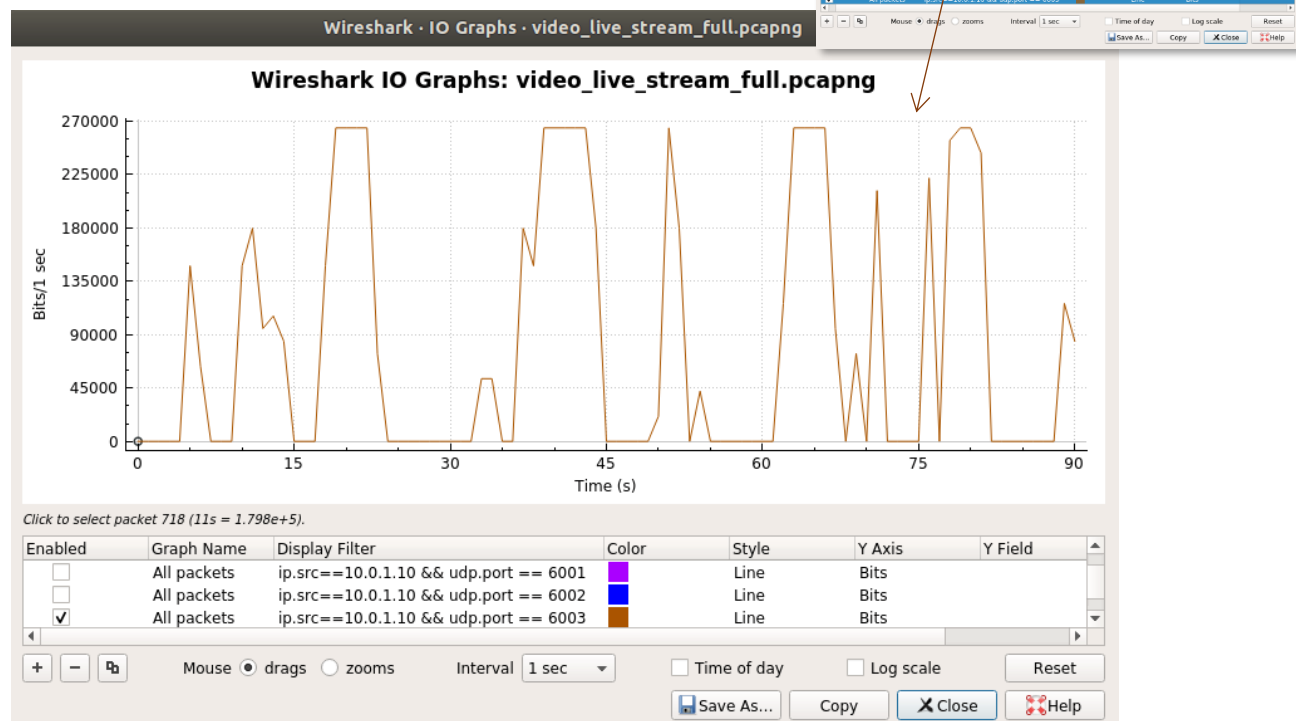
- ✓ Η μαύρη γραμμή απεικονίζει τη ροή και των τριών flow αθροιστικά ως προς το bitrate, οπότε το συνολικό bitrate κυμαίνεται από 1Mbps έως 1.25Mbps.
- ✓ Οι μωβ και μπλε γραμμές αντιστοιχούν στα δύο PERIODIC flows με bitrate που κυμαίνεται στα 512Kbps για το καθένα, γεγονός που συμφωνεί με τις ρυθμίσεις του pattern:

$$50 \text{ messages/sec} \times 1280 \text{ Bytes/message} \times 8 = 512\text{Kbps.}$$

- ✓ Επίσης φαίνεται το μέγεθος των μηνυμάτων ότι είναι όσο είχε καθοριστεί στο pattern, δηλ 1280 Bytes:

```
▼ User Datagram Protocol, Src Port: 6001, Dst Port: 6001
  Source Port: 6001
  Destination Port: 6001
  Length: 1288
  Checksum: 0x4134 [unverified]
  [Checksum Status: Unverified]
  [Stream index: 0]
```


Η καφέ γραμμή αντιστοιχεί στο BURST flow (την εμφανίζουμε παρακάτω απομονωμένη για διευκόλυνση).



! Υπενθύμιση (ξανά): «BURST [RANDOM 3.0 PERIODIC [25.0 1280] EXP 1.0]» και UDP

- ✓ Από τη μορφή του διαγράμματος επιβεβαιώνεται ότι πρόκειται για BURST pattern, εφόσον έχουμε ριπές αποστολής μηνυμάτων που ακολουθούνται από χρονικά διαστήματα απουσίας δραστηριότητας. Η συχνότητα δημιουργίας ριπής είναι κοντά στις ρυθμίσεις (κατά μέσο όρο 3 sec) και γίνεται αντιληπτή στον οριζόντιο άξονα βλέποντας τα χρονικά διαστήματα με bitrate 0. Επιπλέον κάθε ριπή μεμονωμένα έχει PERIODIC pattern αφού είναι σχετικά ομαλές οι κορυφές των «βουνών». Το ύψος των ριπών βρίσκεται κοντά στη ρύθμιση των:

$$25 \text{ messages/sec} \times 1280 \text{ Bytes/message} \times 8 = 256 \text{ Kbps}$$

Ως προς το πρωτόκολλο μεταφοράς του συνολικού σεναρίου (δλδ και των τριών flows συνολικά) επιβεβαιώνονται όσα θα αναμέναμε για το UDP, αφού παρατηρούμε τα εξής:

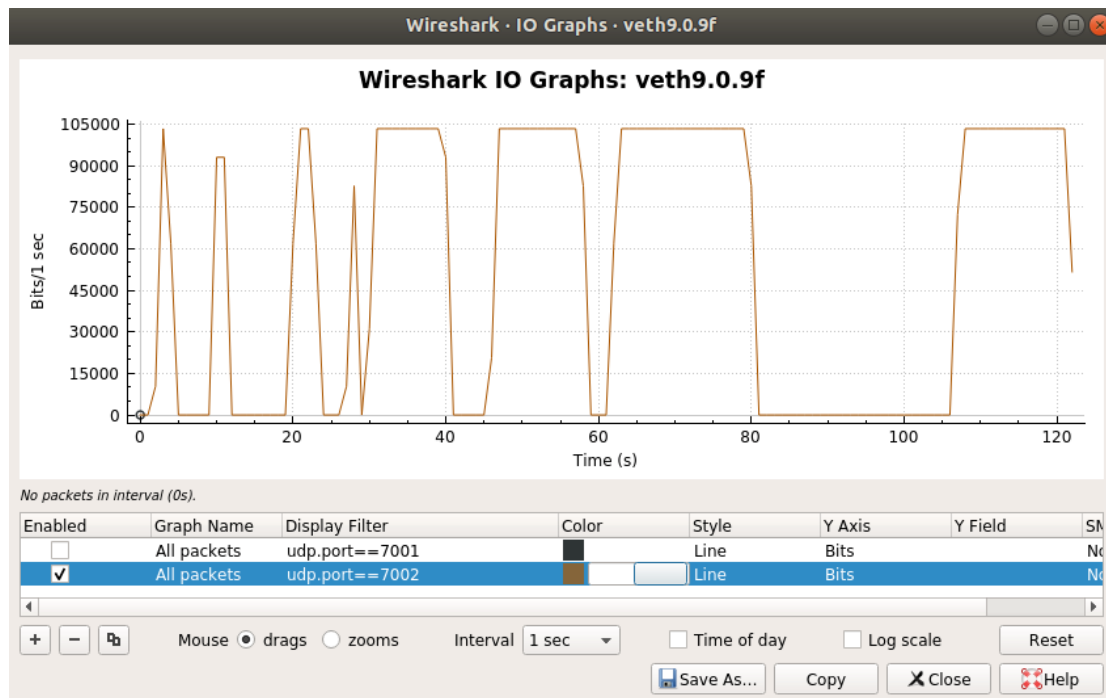
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.0.2.1	224.0.0.5	OSPF	78	Hello Packet
2	2.000072678	10.0.2.1	224.0.0.5	OSPF	78	Hello Packet
3	4.001847697	10.0.2.1	224.0.0.5	OSPF	78	Hello Packet
4	4.837392342	fe80::200:ff:feaa:1	ff02::5	OSPF	90	Hello Packet
5	5.429143571	10.0.1.10	10.0.2.11	UDP	1322	6001 → 6001 Len=1280
6	5.436633989	10.0.1.10	10.0.2.11	UDP	1322	6002 → 6002 Len=1280
7	5.442462866	10.0.1.10	10.0.2.11	UDP	1322	6003 → 6003 Len=1280
8	5.450313834	10.0.1.10	10.0.2.11	UDP	1322	6001 → 6001 Len=1280
9	5.461135680	10.0.1.10	10.0.2.11	UDP	1322	6002 → 6002 Len=1280
10	5.469406531	10.0.1.10	10.0.2.11	UDP	1322	6001 → 6001 Len=1280
11	5.483613966	10.0.1.10	10.0.2.11	UDP	1322	6002 → 6002 Len=1280
12	5.488572170	10.0.1.10	10.0.2.11	UDP	1322	6003 → 6003 Len=1280
13	5.489501538	10.0.1.10	10.0.2.11	UDP	1322	6001 → 6001 Len=1280
14	5.498544147	10.0.1.10	10.0.2.11	UDP	1322	6002 → 6002 Len=1280
15	5.509264101	10.0.1.10	10.0.2.11	UDP	1322	6001 → 6001 Len=1280
16	5.525439366	10.0.1.10	10.0.2.11	UDP	1322	6002 → 6002 Len=1280
17	5.529326837	10.0.1.10	10.0.2.11	UDP	1322	6003 → 6003 Len=1280
18	5.529666865	10.0.1.10	10.0.2.11	UDP	1322	6001 → 6001 Len=1280

- ✓ Δεν υπάρχει εγκαθίδρυση σύνδεσης (όπως υπήρχε στο TCP). Φαίνεται στο screenshot ότι η αποστολή μηνυμάτων ξεκινάει κατευθείαν.
- ✓ Δεν υπάρχουν επιβεβαιώσεις (ACKs) και συνεπώς πακέτα στέλνονται μόνο από τον κόμβο n11 (10.0.1.10) προς τον n22 (10.0.2.11).

Push to Talk Voice Communication

Καταγραφή με τα 2 συμμετρικά flows ενεργά

(VoIP UDP n13-n21, VoIP UDP n21-n13)



! Υπενθύμιση: «BURST [RANDOM 8.0 PERIODIC [10.0 1250] EXP 4.0]» και UDP.

- ✓ Από τη μορφή του διαγράμματος επιβεβαιώνεται ότι πρόκειται για BURST pattern, εφόσον έχουμε ριπές αποστολής μηνυμάτων που ακολουθούνται από χρονικά διαστήματα απουσίας δραστηριότητας. Παρατηρούμε ότι η διάρκεια των ριπών όπως φαίνεται από το πλάτος που έχουν τα «βουνά» που σχηματίζονται συμφωνεί με τις ρυθμίσεις (διάρκεια κατά μέσο όρο 4 sec), όπως επίσης και η συχνότητα δημιουργίας ριπής (κατά μέσο όρο 8 sec) που γίνεται αντιληπτή στον οριζόντιο άξονα βλέποντας τα χρονικά διαστήματα με bitrate 0. Επιπλέον κάθε ριπή μεμονωμένα έχει PERIODIC pattern αφού είναι σχετικά ομαλές οι κορυφές των «βουνών». Το ύψος των ριπών βρίσκεται κοντά στη ρύθμιση των:

$$10 \text{ messages/sec} \times 1250 \text{ Bytes/message} \times 8 = 100 \text{ Kbps}$$

- ✓ Επίσης φαίνεται το μέγεθος των μηνυμάτων ότι είναι όσο είχε καθοριστεί στο pattern, δηλ 1280 Bytes:

```

▼ User Datagram Protocol, Src Port: 7001, Dst Port: 7001
  Source Port: 7001
  Destination Port: 7001
  Length: 1258
  Checksum: 0x64d1 [unverified]
  [Checksum Status: Unverified]
  [Stream index: 1]
▶ Data (1250 bytes)

```

Ως προς το πρωτόκολλο μεταφοράς, επιβεβαιώνονται όσα θα αναμέναμε για το UDP, αφού παρατηρούμε τα εξής:

Apply a display filter ... <Ctrl-/> Expression..						
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.0.2.1	224.0.0.5	OSPF	78	Hello Packet
2	2.001828490	10.0.2.1	224.0.0.5	OSPF	78	Hello Packet
3	2.984899124	10.0.2.10	10.0.1.13	UDP	1292	7002 → 7002 Len=1250
4	2.998387465	10.0.1.13	10.0.2.10	UDP	1292	7001 → 7001 Len=1250
5	3.085703820	10.0.2.10	10.0.1.13	UDP	1292	7002 → 7002 Len=1250
6	3.097317400	10.0.1.13	10.0.2.10	UDP	1292	7001 → 7001 Len=1250
7	3.185478655	10.0.2.10	10.0.1.13	UDP	1292	7002 → 7002 Len=1250
8	3.200679724	10.0.1.13	10.0.2.10	UDP	1292	7001 → 7001 Len=1250
9	3.287806103	10.0.2.10	10.0.1.13	UDP	1292	7002 → 7002 Len=1250
10	3.298391870	10.0.1.13	10.0.2.10	UDP	1292	7001 → 7001 Len=1250
11	3.385978392	10.0.2.10	10.0.1.13	UDP	1292	7002 → 7002 Len=1250
12	3.397179929	10.0.1.13	10.0.2.10	UDP	1292	7001 → 7001 Len=1250
13	3.485509205	10.0.2.10	10.0.1.13	UDP	1292	7002 → 7002 Len=1250

- ✓ Δεν υπάρχει εγκαθίδρυση σύνδεσης (όπως υπήρχε στο TCP). Φαίνεται στο screenshot ότι η αποστολή μηνυμάτων ξεκινάει κατευθείαν.
- ✓ Δεν υπάρχουν επιβεβαιώσεις (ACKs) και συνεπώς πακέτα στέλνονται από τον κόμβο n21 (10.0.2.10) προς τον n13 (10.0.1.13) στα ports 7002 προς 7002 και από τον n13 (10.0.1.13) προς τον n21 (10.0.2.10) στα ports 7001 προς 7001 αλλά κανένα από αυτά δεν είναι ACK. Στέλνοντας ο ένας κόμβος στον άλλον επιτυγχάνεται η προσομοίωση της αμφίδρομης επικοινωνίας μεταξύ τους με VoIP.