

Εθνικό Μετσόβιο Πολυτεχνείο

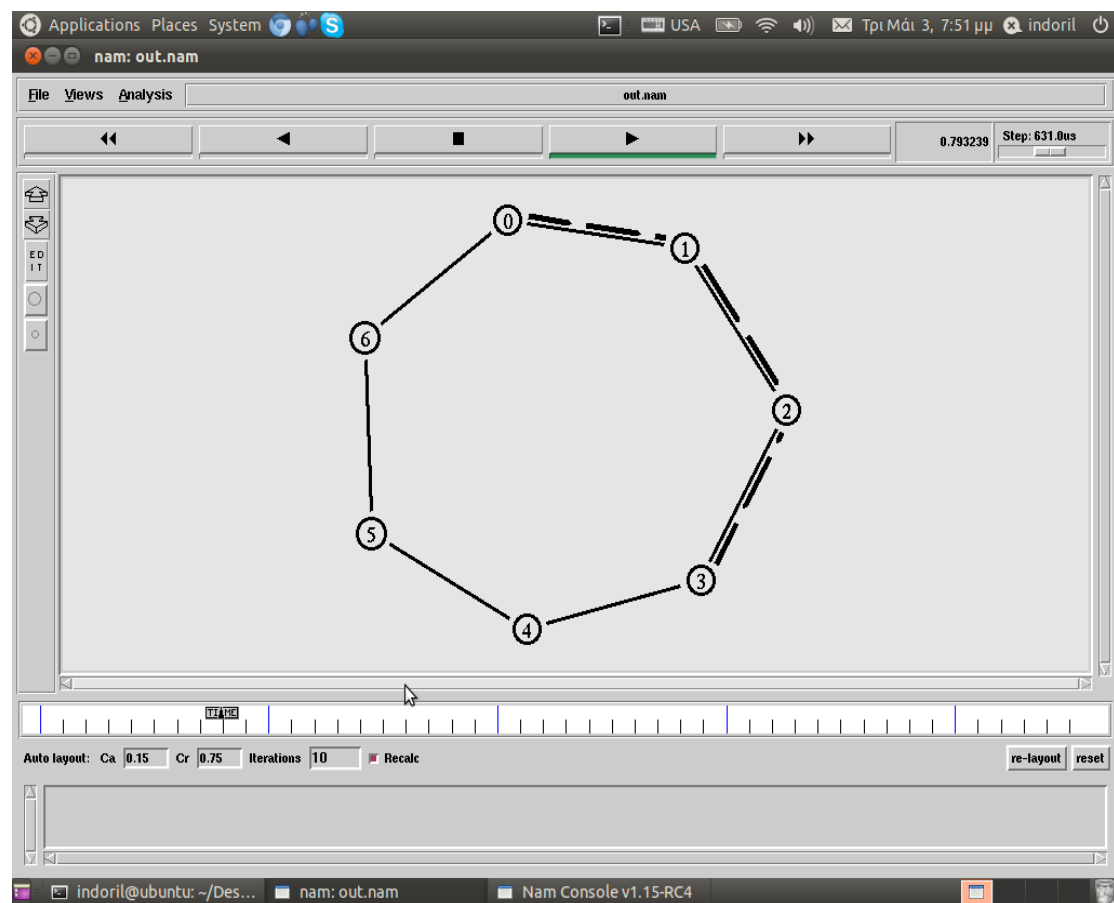
Σχολή Ηλεκτρολόγων Μηχ. Και Μηχ. Υπολογιστών

Δίκτυα Επικοινωνιών, 6<sup>ο</sup> εξάμηνο

Ακαδημαϊκή Περίοδος : 2010-2011



## 2<sup>η</sup> Εργαστηριακή Άσκηση



Γερακάρης Βασίλης  
Α.Μ. :03108092

- Σκοπός:

Σκοπός της άσκησης αυτής είναι η προσομοίωση πιο σύνθετων τοπολογιών δικτύων μέσω του λογισμικού Network Simulator 2 και η επίλυση σύνθετων προβλημάτων πάνω σε αυτά.

Συγκεκριμένα, ακολουθώντας τις υποδείξεις της εκφώνησης, στο πρώτο σκέλος της άσκησης μελετήσαμε την περίπτωση δικτύου επικοινωνιών που προκύπτει μεταξύ τεσσάρων κόμβων (2 UDP agents, 1 δρομολογητής και 1 sink agent) και τη συμπεριφορά του για διαφορετικές υλοποιήσεις της σειράς αναμονής.

Στο δεύτερο σκέλος της άσκησης μελετάμε ένα δίκτυο 7 κόμβων σε συνδεσμολογία δακτυλίου και τη δυναμική του συμπεριφορά και αναπροσαρμογή στην περίπτωση που κόπει κάποια ζεύξη.

Όπως και στην πρώτη εργαστηριακή άσκηση, έγινε και εδώ χρήση των εργαλείων NAM και XGRAPH του NS2 για οπτικοποίηση των αποτελεσμάτων και καλύτερη κατανόηση του κυκλώματος.

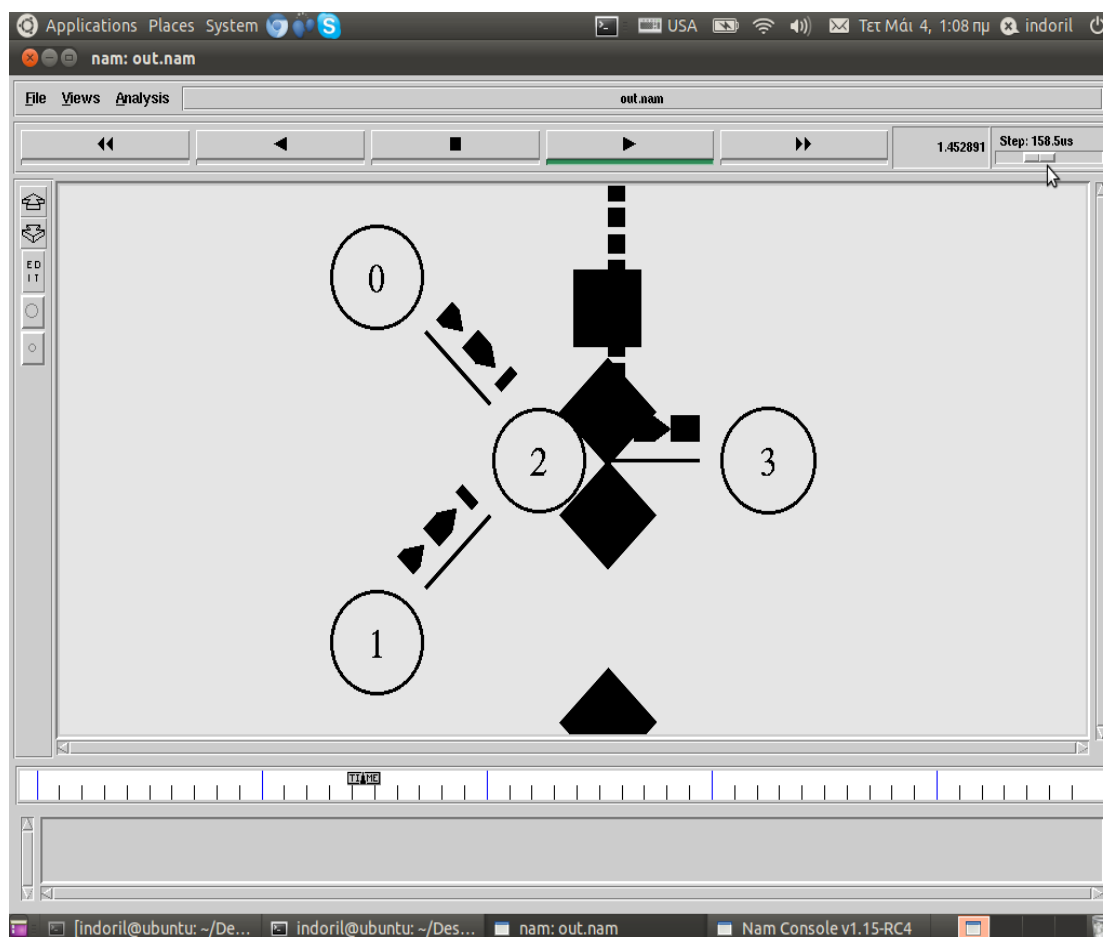
## Ενότητα 1

### • 1.1 - Δημιουργία Τοπολογίας

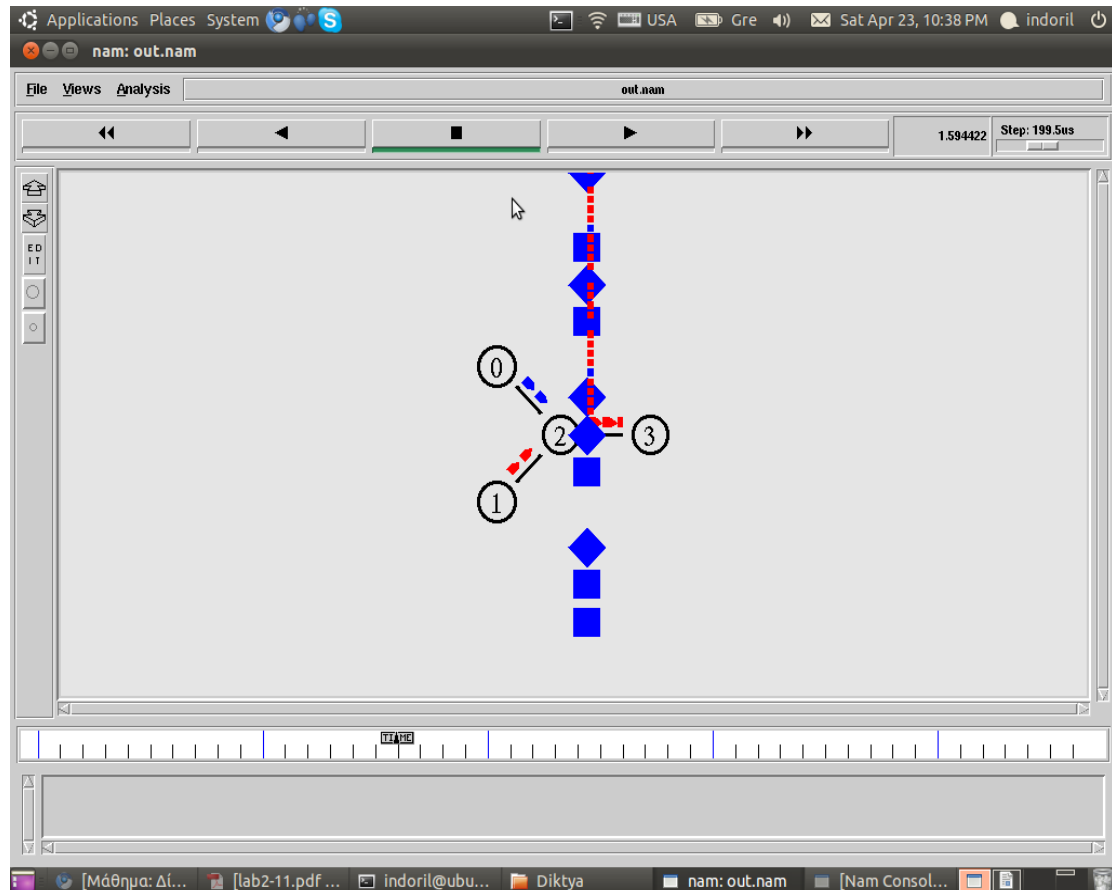
Δημιουργούμε την τοπόλογία όπως περιγράφεται από την εκφώνηση. Αρχικά δημιουργούμε τους 2 UDP agents στους κόμβους n0 και n1, και τοποθετούμε πηγές που παράγουν δεδομένα CBR (Constant Bit Rate). Στον κόμβο n3 τοποθετούμε τον sink agent, ενώ ο n2 θα λειτουργήσει ως δρομολογητής στο δίκτυο που σχεδιάσαμε.

Μεταξύ των κόμβων χρησιμοποιήσαμε ζεύξεις με εύρος ζώνης 1 Mbps και καθυστέρηση 10 msec.

Εκτελώντας την προσομοίωση, το αποτέλεσμα του NAM ήταν το παρακάτω:



Παρατηρούμε ότι υπάρχει απώλεια πακέτων από τον κόμβο 2, αλλά στην παρούσα μορφή είναι αδύνατο να προσδιορίσουμε την προέλευσή τους, γι αυτό κάνουμε μια μετατροπή στο αρχικό script, ώστε να χρωματίζει τα πακέτα ανάλογα με τον UDP agent από τον οποίο προέρχονται. Εκτελώντας το script, έχουμε ως έξοδο του NAM:



## • 1.2 - Ανάλυση Δικτύου μέσω NAM

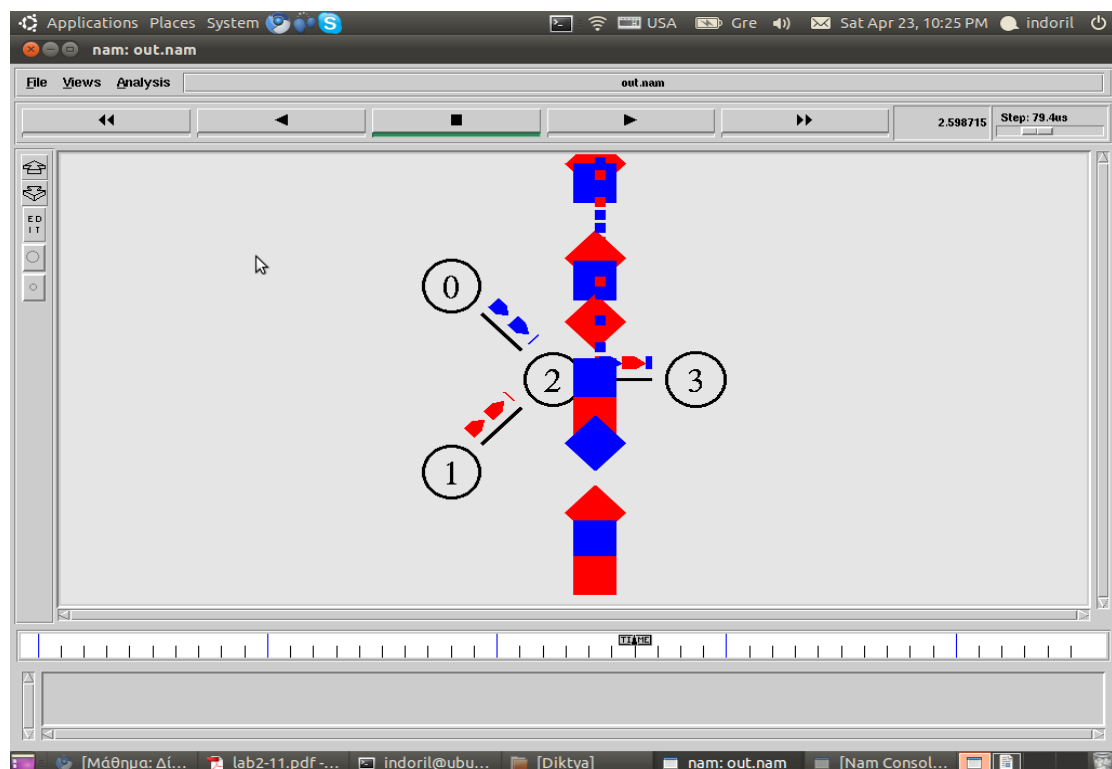
Παρατηρούμε ότι στον κόμβο 2 υπάρχει η ουρά αναμονής όπου συσσωρεύονται τα πακέτα και στη συνέχεια δρομολογούνται στον κόμβο 3 με απώλειες λόγω της χωρητικότητας της ζεύξης. Πιο συγκεκριμένα, ο n0 CBR agent αρχίζει την εκπομπή πακέτων όταν  $t=0.5\text{sec}$ , ενώ ο n1 για  $t=1\text{sec}$ . Όταν  $t=1.35\text{sec}$  αρχίζουν να παρατηρούνται οι πρώτες απώλειες πακέτων.

## DropTail

Στην αρχική μας υλοποίηση χρησιμοποιούμε ουρά δεδομένων (buffer) **DropTail**, ή οποία έχει δομή FIFO (First In – First Out). Όπως είναι φανερό και από την έξοδο του NAM, στην υλοποίηση αυτή οι απώλειες προκύπτουν εξ'ολοκλήρου από τον κόμβο n0 (μπλέ πακέτα), ενώ δεν υπάρχουν απώλειες από τον κόμβο n1 (κόκκινα πακέτα). Αυτό προφανώς δεν είναι επιθυμητό σε πραγματικές εφαρμογές, αφού αποτελεί μια άδικη ουρά αναμονής.

## SFQ

Στη συνέχεια θα χρησιμοποιήσουμε ένα διαφορετικό μοντέλο ουράς αναμονής, προκειμένου να καταλήξουμε σε μια πιο “δίκαιη” διανομή των δεδομένων μέσα στο δίκτυο. Το μοντέλο αυτό ονομάζεται **SFQ** (Stochastic Fair Queuing) και υλοποιεί μια πιο δίκαιη ουρά αναμονής μέσω της χρήσης στοχαστικής μεταβλητής για την κατανομή των απωλειών. Αφού μεταβάλλουμε και εκτελέσουμε το script, έχουμε την ακόλουθη έξοδο στο NAM:



- **1.3 - Απαντήσεις ερωτήσεων**

- *Ποιο είναι το ποσοστό των πακέτων που χάνονται από την μπλε και κόκκινη ροή, όταν χρησιμοποιείται η ουρά DropTail;*

Παρατηρούμε πως για κάθε 4 πακέτα του κόμβου n1 που διέρχονται από τον δρομολογητή χωρίς απώλειες, αντιστοιχεί 1 από τον κόμβο n0, ενώ τα άλλα 3 (μπλέ) πακέτα χάνονται. Επομένως, το ποσοστό των απωλειών για τα μπλέ πακέτα είναι 75%, ενώ για τα κόκκινα 0%.

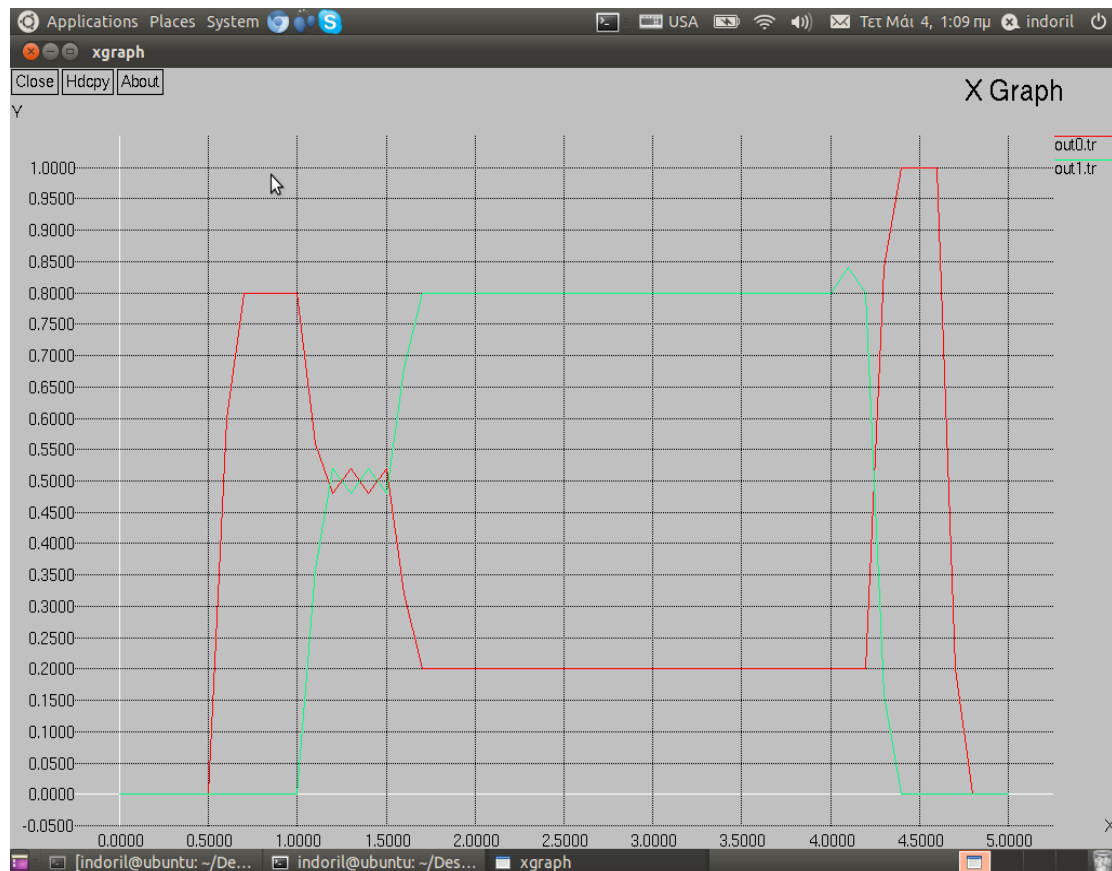
- *Παρατηρώντας το animation, εκτιμήστε το ποσοστό των πακέτων που χάνονται από την μπλε και από την κόκκινη ροή, όταν χρησιμοποιείται η ουρά SFQ;*

Παρατηρούμε πως παρουσιάζεται ίδιος αριθμός μπλε και κόκκινων χαμένων πακέτων, σε περίοδο 2 “κύκλων”. Δηλαδή εναλλάσσονται τα σύνολα {2 χαμένα μπλε + 1 κόκκινο} και {2 χαμένα κόκκινα + 1 μπλε}. Έτσι έχουμε συνολικές απώλειες για κάθε UDP agent:  $3/8 = 37,5\%$

- **1.4 - Μελέτη τοπολογίας με Xgraph**

Σε αυτό το κομμάτι της άσκησης θα χρησιμοποιήσουμε το πρόγραμμα Xgraph για να σχεδιάσουμε τις γραφικές παραστάσεις της κίνησης που χαρακτηρίζει το δίκτυο που δημιουργήσαμε.

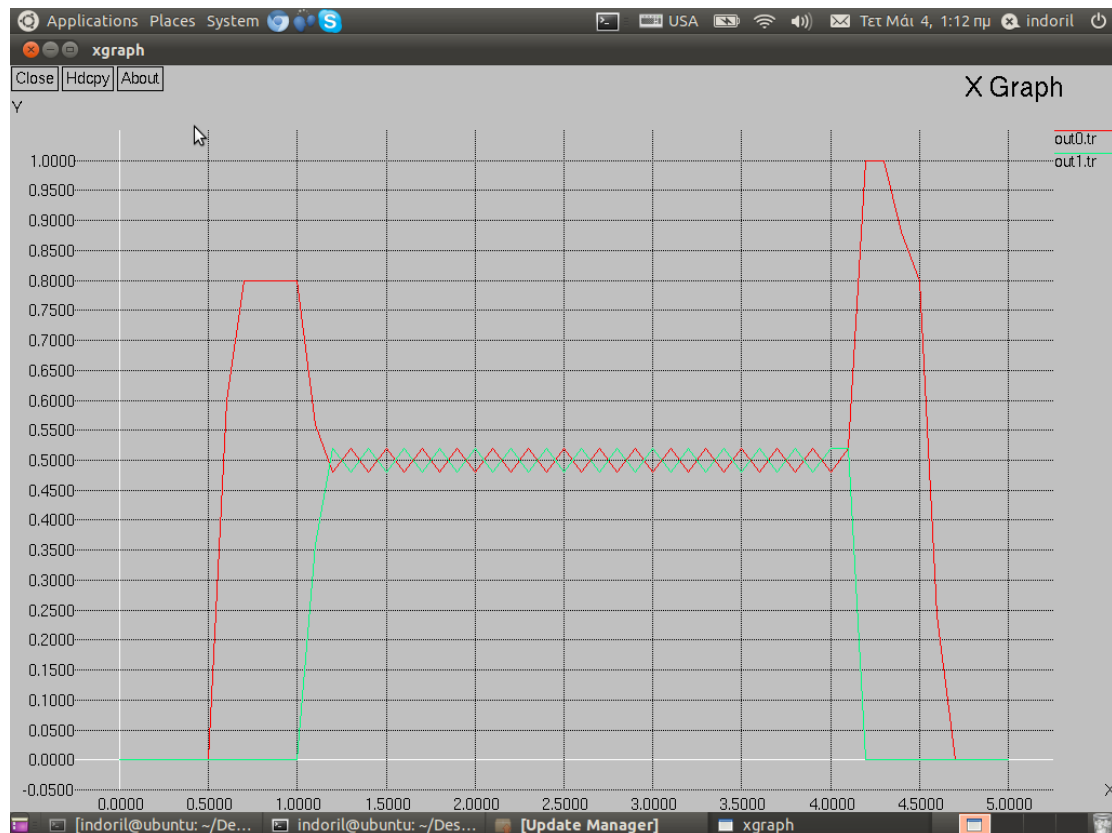
Για την ουρά τύπου DropTail είχαμε την ακόλουθη έξοδο του xgraph:



Παρατηρούμε πώς τα πακέτα που προέρχονται από τον κόμβο n0 αρχίζουν να παρουσιάζουν απώλειες, όταν  $t = 1\text{sec}$ , με ποσοστό 75%, ενώ στα πακέτα του κόμβου n1 δεν παρατηρούνται απώλειες. Το αποτέλεσμα αυτό επαληθεύει τη θεωρία και τον χαρακτηρισμό της DropTail ως “άδικης” στον τρόπο διανομής των πακέτων.

Η διαφορά αυτή γίνεται καταφανής εάν κάποιος παρατηρήσει στη γραφική παράσταση τη διαφορά του εύρους ζώνης που χρησιμοποιείται για τις δύο γραμμές.

Για την ουρά τύπου SFQ είχαμε την ακόλουθη έξοδο του xgraph:



Η γραφική αυτή παράσταση επιβεβαιώνει τον ισχυρισμό μας ότι τα ποσοστά απωλειών είναι ίσα και για τις 2 γραμμές. Το μέσο εύρος ζώνης που χρησιμοποιεί η κάθε γραμμή κατά τη χρονικό διάστημα που αποστέλλουν δεδομένα και οι 2 UDP agents είναι τα 0,5Mb. Επομένως οι ζητούμενες απώλειες θα είναι:  $(0,8 - 0,5) / 0,5 = 0,375$  ή 37,5%.

Τέλος, παρατηρούμε πως γίνεται ιση κατανομή των πόρων, αφού οι 2 γραμμές χρησιμοποιούν το ίδιο εύρος ζώνης.



- Ποια είναι η μέγιστη και ποια η ελάχιστη τιμή του ρυθμού μεταφοράς των δεδομένων για τις δυο ροές;

Τη χρονική περίοδο που εκπέμπουν και οι 2 agents έχουμε:

#### DropTail

n0:	Max = 0.8Mbps	Min = 0.2Mbps
n1:	Max = 0.8Mbps	Min = 0.8Mbps

#### SFQ

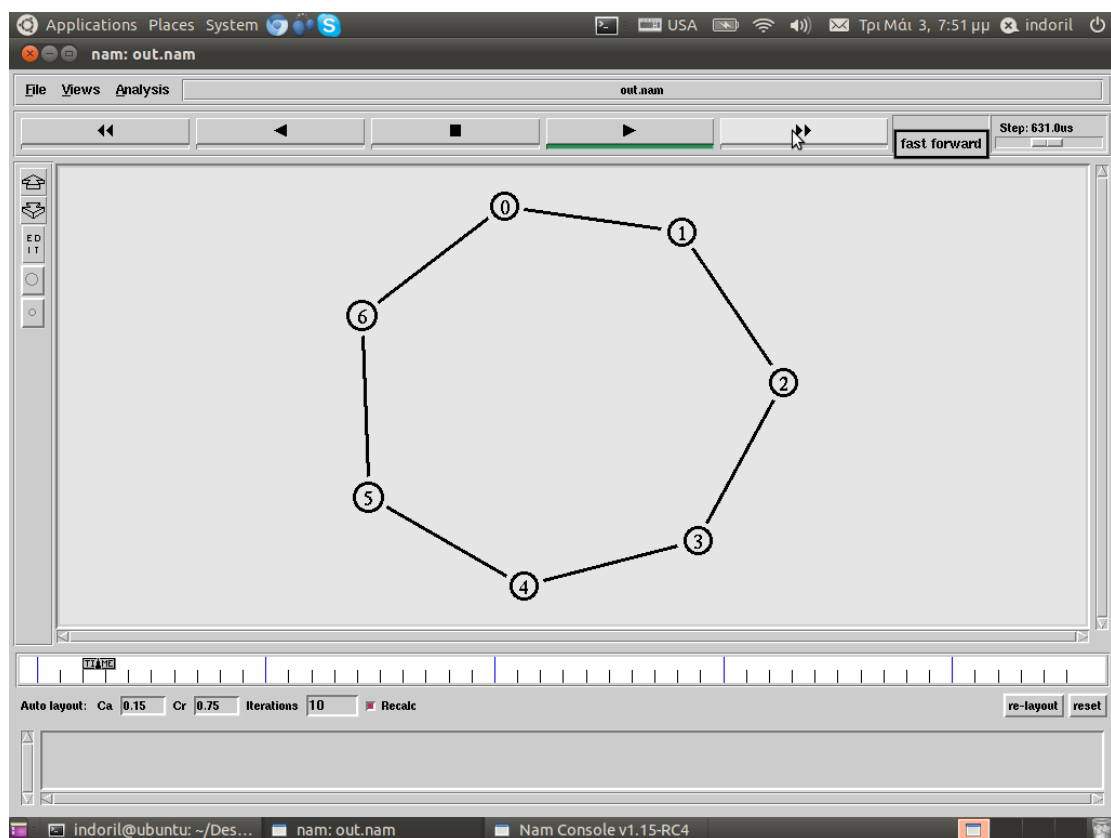
n0:	Max = 1.0Mbps	Min = 0.5Mbps
n1:	Max = 0.8Mbps	Min = 0.5Mbps

## Ενότητα 2

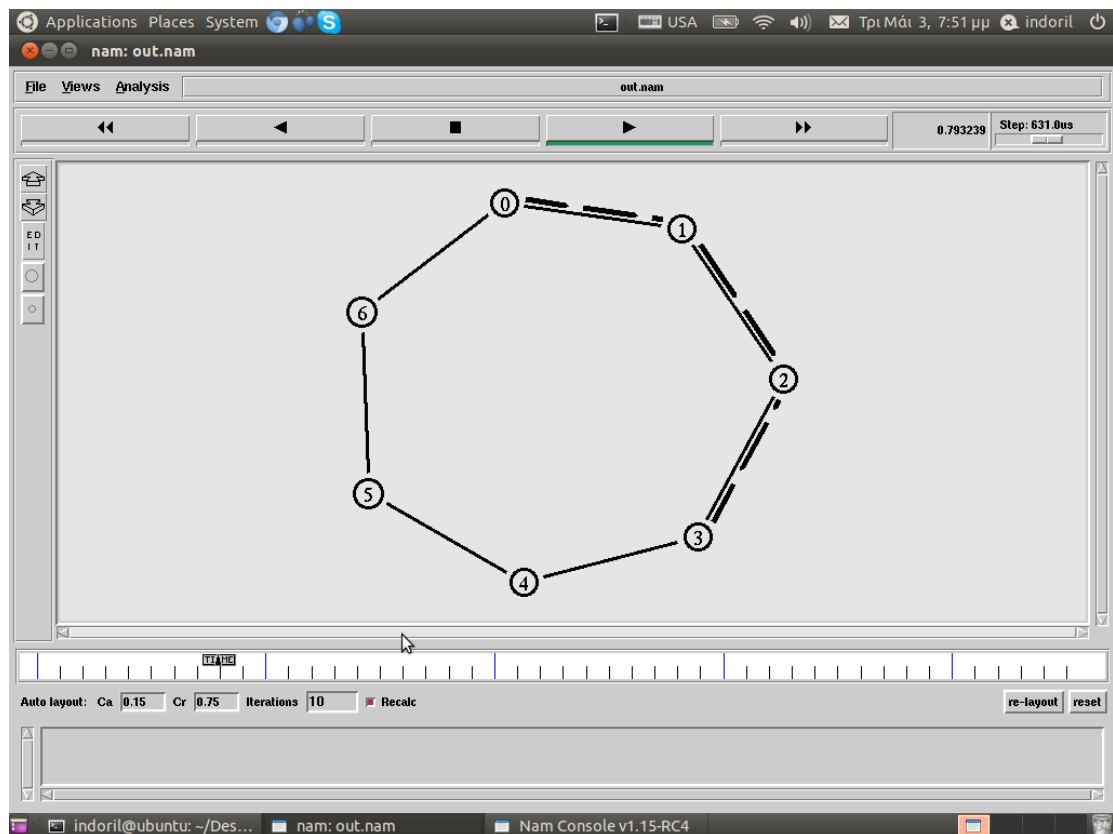
### • 2.1 – Δημιουργία Μεγαλύτερης Τοπολογίας

Στο κομμάτι αυτό της άσκησης, καλούμαστε να μελετήσουμε και να αναλύσουμε τη δυναμική συμπεριφορά ενός δικτύου. Ακολουθώντας τις οδηγίες της εκφώνησης δημιουργούμε τον παρακάτω δακτύλιο 7 κόμβων, όπου αποστέλονται δεδομένα από τον κόμβο n0 (UDP agent – πηγή CBR δεδομένων) προς τον κόμβο n3 (sink agent).

Η τοπολογία που προκύπτει φαίνεται στο παρακάτω σχήμα:

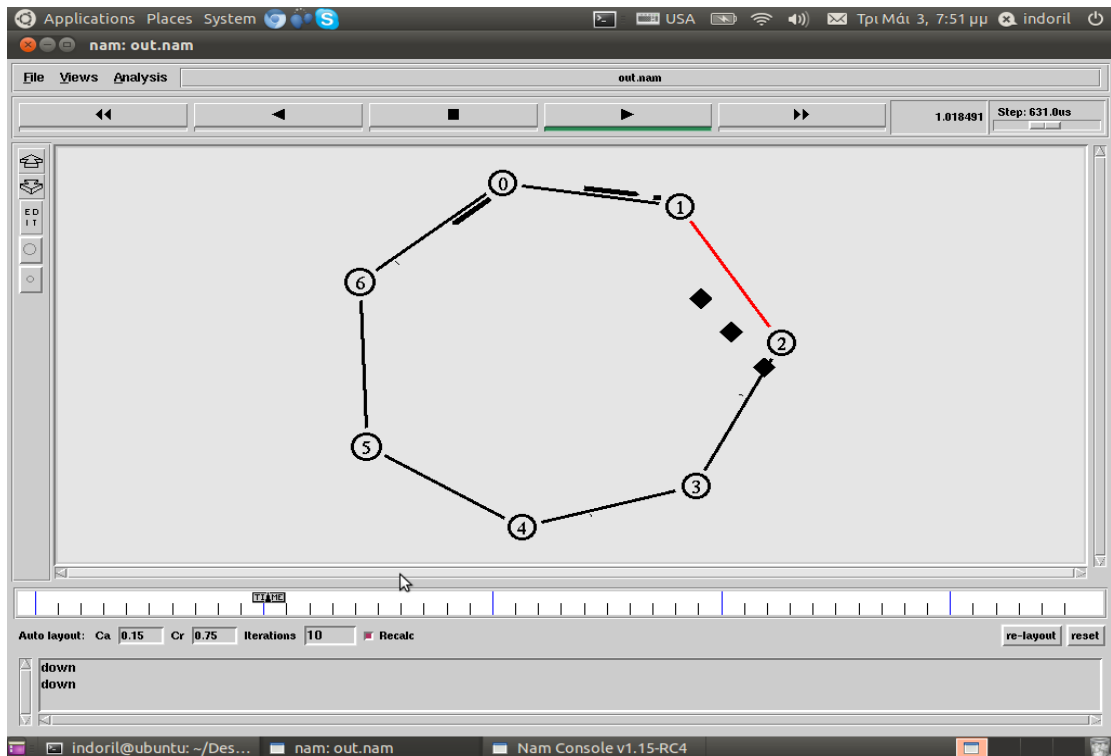


Εκτελώντας την προσομοίωση του script, βλέπουμε να πραγματοποιείται η ροή των δεδομένων.

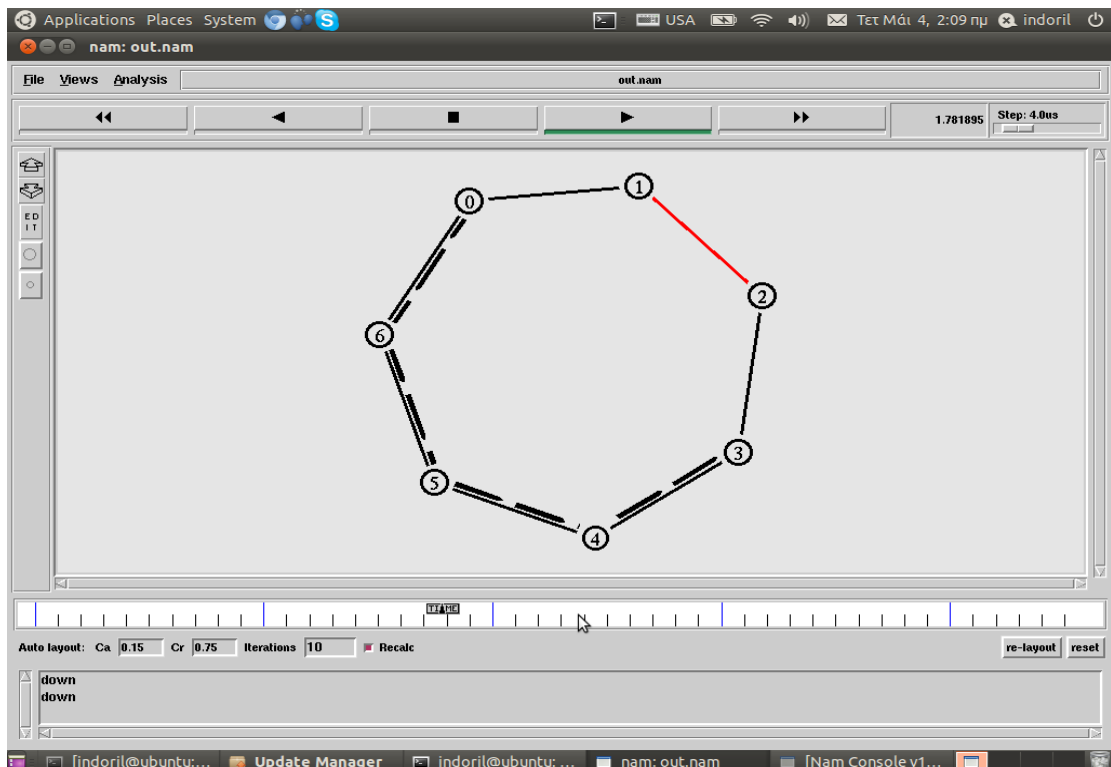


## - 2.2 – Διακοπή ζεύξης

Εδώ, μελετάμε τη δυναμική συμπεριφορά και δρομολόγηση του δικτύου στην περίπτωση που παρουσιαστεί κάποιο πρόβλημα, όπως για παράδειγμα η διακοπή μιας ζεύξης. Στην περίπτωση αυτή, το δίκτυο ενημερώνεται για τη “βλάβη” μέσω της χρήσης μικρότερων πακέτων που ονομάζονται rtpeto DV και αναδρομολογεί την κίνηση μέσω άλλων, διαθέσιμων κόμβων. Με τον τρόπο αυτό, αποτρέπεται η εκτενής απώλεια πακέτων. Στο παράδειγμά μας, η κίνηση αναδρομολογείται από τους κόμβους 0 -> 1 -> 2 -> 3 στους 0 -> 6 -> 5 -> 4 -> 3.

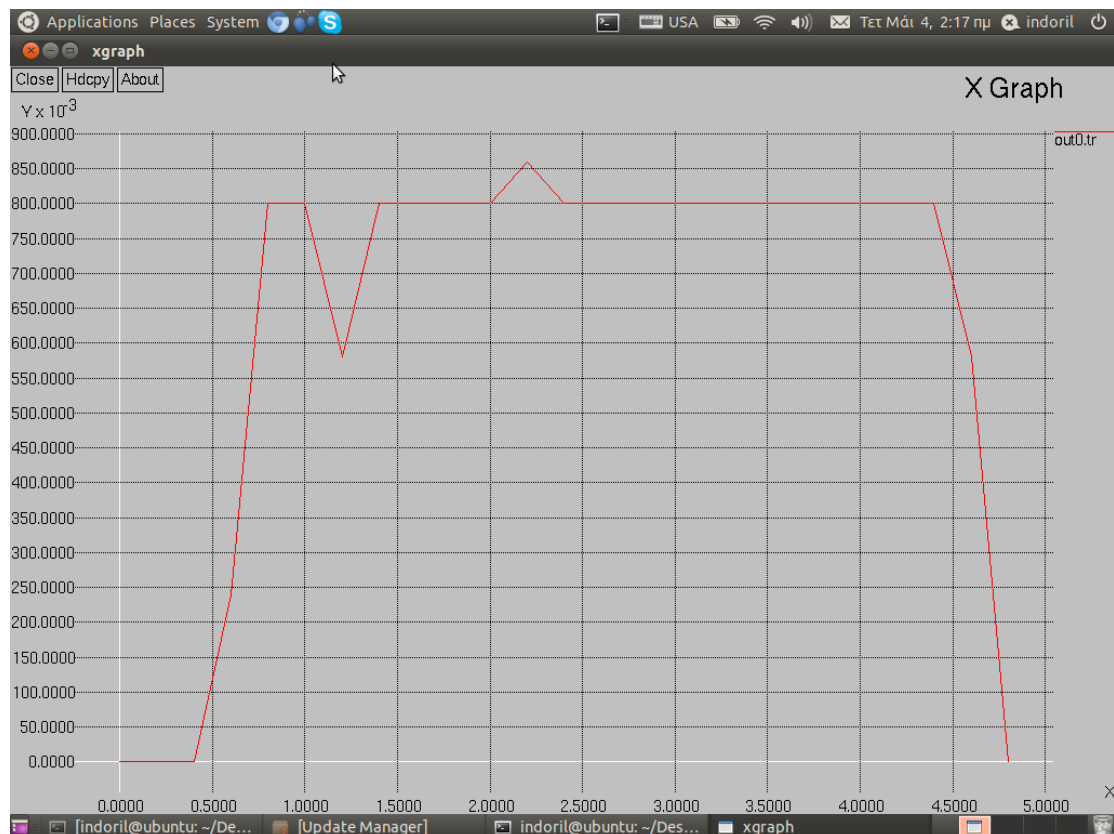


Στην εικόνα αυτή φαίνεται η χρονική στιγμή που διακόπτεται η ζεύξη και χάνονται τα πακέτα που βρίσκονται πάνω σε αυτή. Στη συνέχεια εκπέμπονται τα πακέτα rtproto DV, τα οποία όταν φτάσουν στον κόμβο n0, θα γίνει η αναδρομολόγηση των πακέτων, όπως φαίνεται στην παρακάτω εικόνα:

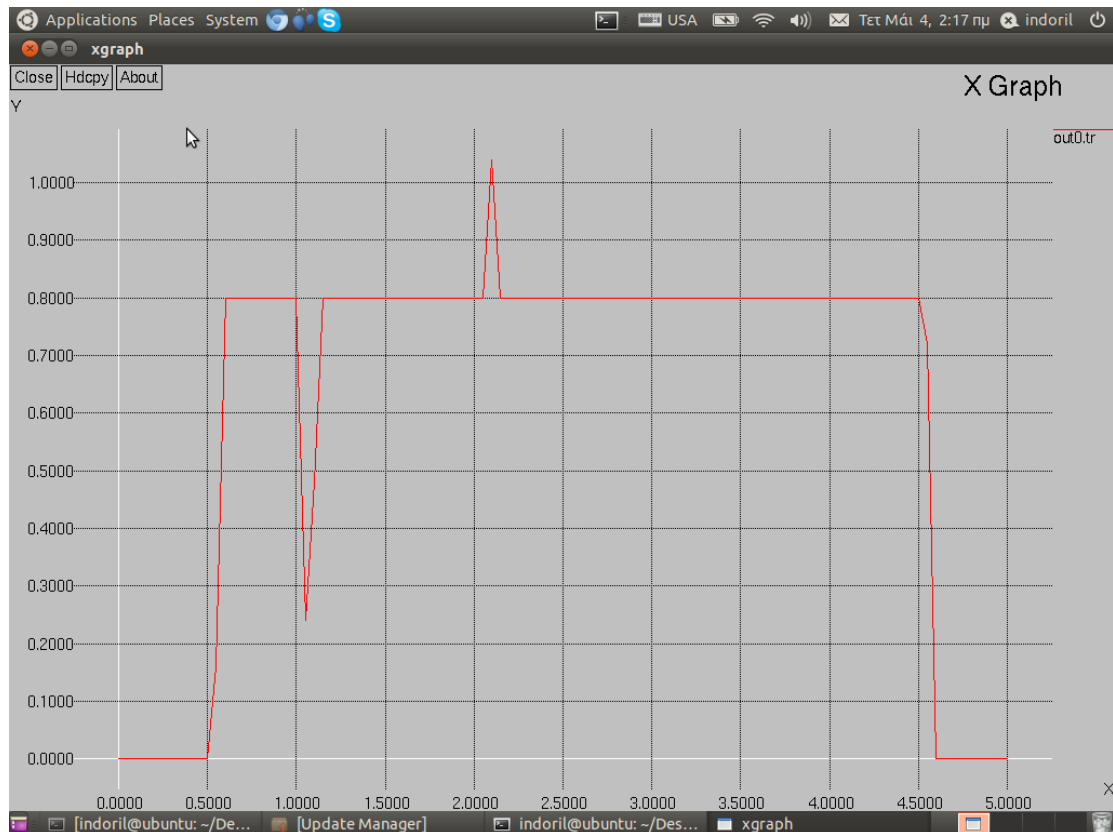


### • 2.3 - Προσομοίωση στο X-graph

Εκτελώντας το xgraph για τα δεδομένα της εκφώνησης προκύπτει η παρακάτω γραφική:



Παρατηρούμε πως ο χρόνος καταγραφής της συνάρτησης record είναι πολύ μεγάλος για να καταγράψει τη μεταβολή που πραγματοποιείται στο δίκτυο. Για να έχουμε μια πιο έγκυρη εικόνα του τι συμβαίνει θα πρέπει να μειώσουμε τον χρόνο καταγραφής από 0,2sec σε 0,05sec. Τότε, προκύπτει το παρακάτω σχήμα:



Εδώ παρατηρείται μια απότομη μείωση των πακέτων όταν  $t = 1\text{sec}$  που προκύπτει από τη διακοπή της ζεύξης. Σύντομα όμως επανέρχεται, αφού πραγματοποιείται η αναδρομολόγηση μέσω των κόμβων  $6 \rightarrow 5 \rightarrow 4$ . Όταν  $t = 2\text{ sec}$ , η ζεύξη επανέρχεται και η διανομή των πακέτων αρχίζει και πάλι να πραγματοποιείται μέσω της συντομότερης διαδρομής, όπως και στην αρχή. Για ένα ελάχιστο χρονικό διάστημα, στον κόμβο  $n3$  θα καταφτάνουν πακέτα μέσω και των 2 διαδρομών, γεγονός που δικαιολογεί την αιχμή στην κίνηση που φαίνεται στο παραπάνω γράφημα.