

Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Μάθημα: Δίκτυα Επικοινωνιών

3η Εργαστηριακή Άσκηση

Ονοματεπώνυμο : Σταυρακάκης Δημήτριος

ΑΜ : 03112017

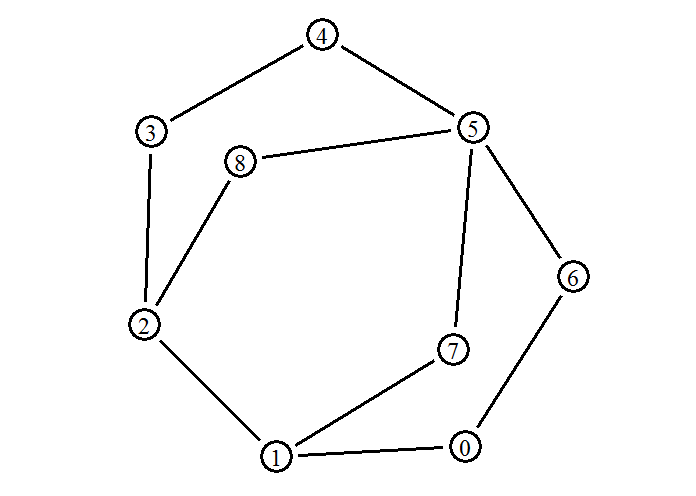
Εξάμηνο : 6ο

Ημερομηνία Παράδοσης: 1/4/2015

Μέρος 1: Μετάδοση δεδομένων σε δίκτυο με σύνθετη τοπολογία

**1.1: Τοπολογία**

Στο πρώτο αυτό ερώτημα της παρούσας άσκησης θα υλοποιήσουμε μια σύνθετη τοπολογία που περιλαμβάνει 9 κόμβους και θα παρατηρήσουμε τη μετάδοση δεδομένων σε αυτή όταν μεταβάλλουμε τον αριθμό των παρεμβαλλόμενων κόμβων και ζεύξεων και όταν μεταβάλλουμε το κόστος των ζεύξεων μεταξύ των κόμβων. Αρχικά ορίζονται οι εννέα κόμβοι στην τοπολογία και αμφίδρομες ζεύξεις μεταξύ των πρώτων επτά κόμβων με καθυστέρηση 40ms(οι υπόλοιπες διαφέρουν και ορίζονται στον κώδικα). Η τοπολογία που δίνει το animation όταν το τρέχουμε με το πρώτο αυτό τμήμα του κώδικα (κάνοντας το κατάλληλο re-layout για να έχουμε πιο ξεκάθαρο σχήμα ) είναι η ακόλουθη:

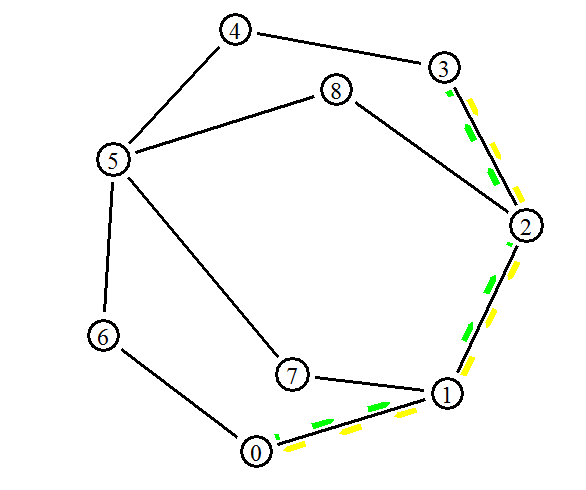


**1.2: Τα γεγονότα (events)**

Στο μέρος αυτός της άσκησης ήρθε η ώρα όπως και στις προηγούμενες ασκήσεις να δημιουργήσουμε τους agents μας. Όπως ζητείται δημιουργούμε 2 UDP agents που παράγουν κίνηση πακέτων CBR. Τους τοποθετούμε στους κόμβους n0 και n3. Στους ίδιους αυτούς κόμβους εν συνεχεία δημιουργούμε και 2 Sink Agents που λαμβάνουν τα δεδομένα. Ρυθμίζοντας τις χρονικές παραμέτρους του δικτύου μας, ορίζουμε τη χρονική διάρκεια κατά την οποία οι 2 UDP agents θα στέλνουν τα πακέτα τους ως εξής:

Ο πρώτος στέλνει από τη στιγμή 0.2 sec έως τη στιγμή 4.2 sec

Ενώ ο δεύτερος από τη στιγμή 0.7 έως τη στιγμή 3.7 sec

Σημειώνουμε ότι τα πακέτα που αποστέλλονται από τον κόμβο n0 στον κόμβο n3 έχουν πράσινο χρώμα ενώ αντίθετα, αυτά που ακολουθούν τη διαδρομή n3 -> n0 είναι κίτρινα. Αφού λοιπόν γράψουμε και τρέξουμε τον κώδικα που περιγράψαμε ο οποίος μάλιστα παρατίθεται στο τέλος της αναφοράς παίρνουμε το εξής στιγμιότυπο:

**1.3: Ερωτήσεις**

* **Ποια διαδρομή ακολουθούν τα πακέτα;**

Όπως παρουσιάζεται και παραπάνω στο στιγμιότυπο που παρατέθηκε τα πράσινα πακέτα ακλουθούν τη διαδρομή 0->1->2->3 ενώ τα κίτρινα πακέτα τη διαδρομή 3->2->1->0.

* **Ελέγξτε αν η ροή των πακέτων και από τις δυο πλευρές ακολουθεί τη διαδρομή με τα λιγότερα βήματα.**

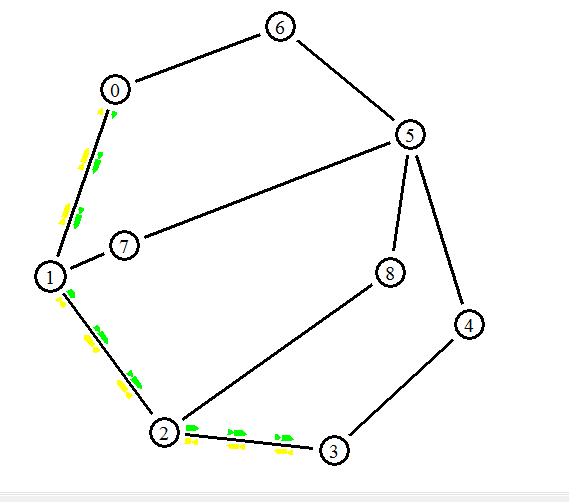
Τόσο τα πράσινα όσο και τα κίτρινα πακέτα ακολουθούν τη διαδρομή με τα λιγότερα βήματα. (3 βήματα) Κάθε άλλη διαφορετική από αυτή διαδρομή απαιτεί τουλάχιστον 4 βήματα προκειμένου να συνδεθούν οι κόμβοι n0 και n3. Συνεπώς, αυτή η ροή των πακέτων ακολουθεί την βέλτιστη διαδρομή.

* **Υπάρχει συντομότερη διαδρομή από αυτήν που ακολουθούν, όσον αφορά τη συνολική καθυστέρηση κάθε ροής;**

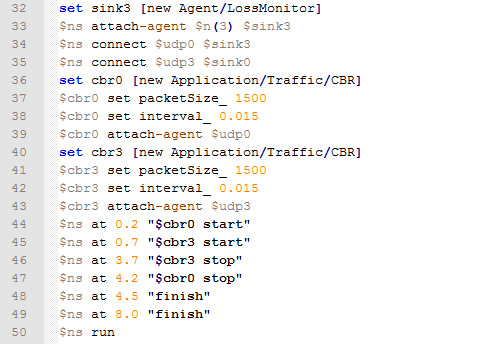
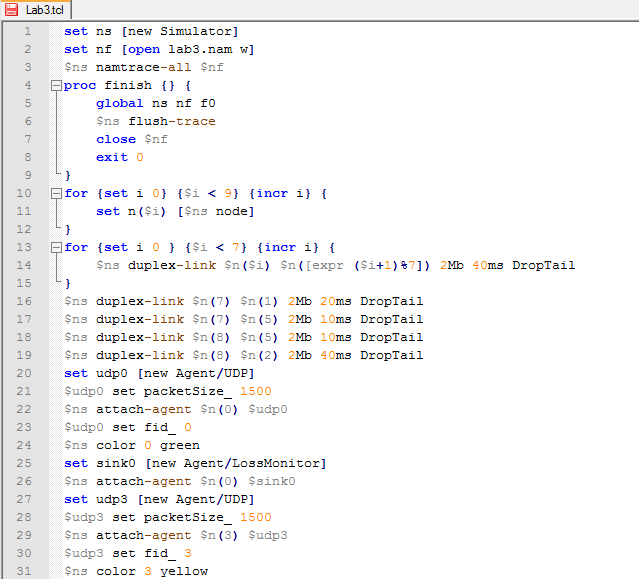
Παρατηρούμε πως η διαδρομή που ακολουθούν τα πακέτα έχει συνολική καθυστέρηση 3\*40ms=120ms (καθώς περνούν από 3 ζεύξεις με 40 ms καθυστέρηση έκαστη). Στη συνέχεια αν παρατηρήσουμε προσεκτικά τις άλλες πιθανές διαδρομές που μπορούν να ακολουθήσουν τα πακέτα μας ώστε να μεταβούν από τον κόμβο n0 στον n3 και αντίστροφα βλέπουμε πως δεν υπάρχει διαδρομή με μικρότερη συνολική καθυστέρηση. Αυτό γιατί οποιαδήποτε από τις πιθανές διαδρομές περνάει από τουλάχιστον 3 ζεύξεις με καθυστέρηση 40 ms( για τη διαδρομή 0-1-7-5-8-2-3 έχει και η 8-2 40 ms καθυστέρηση) αλλά και από άλλες επιπλέον ζεύξεις που καθιστούν την καθυστέρηση μεγαλύτερη από 120 ms. Η δεύτερη από άποψη καθυστέρησης διαδρομή όπως παρατηρούμε είναι η 0-1-7-5-4-3 που έχει 140 ms καθυστέρηση.

* **Ποιος είναι ο ρόλος των εντολών $udp0 set packetSize\_ 1500 και $udp3 set packetSize\_ 1500; Τι παρατηρείτε στις ροές των πακέτων αν αφαιρεθούν οι γραμμές αυτές από τον κώδικα της προσομοίωσης;**

Οι παραπάνω εντολές καθορίζουν το μέγεθος του κάθε πακέτου που αποστέλλουν οι UDP agents των κόμβων n0 και n3. Αν αφαιρεθούν από τον κώδικά μας θα προκύψει το εξής animation:

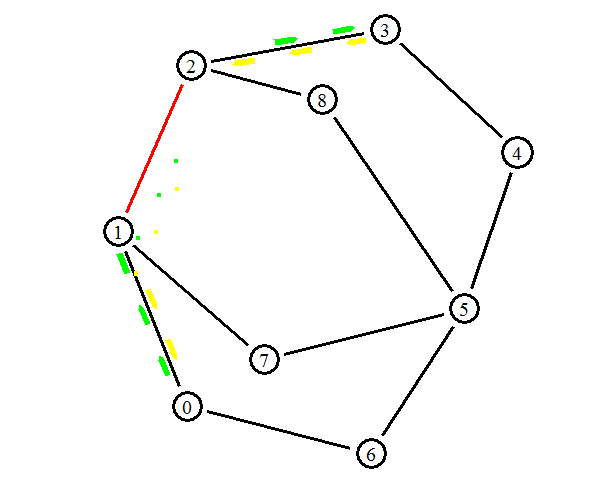
Οι εντολές αυτές ορίζουν ότι το μέγεθος των πακέτων που στέλνουν οι UDP agents στα 1500bytes. Παραλείποντας τις εντολές αυτές προκύπτει το στιγμιότυπο:

Βλέπουμε πως τώρα που αφαιρέσαμε τον ορισμό του μεγέθους του πακέτου , οι UDP agents στέλνουν 2 πακέτα τη φορά. 1 μεγάλο μεγέθους 1000bytes και ένα ακόμη μικρότερο μεγέθους 500bytes. Αυτό συμβαίνει για το λόγο ότι ορίσαμε ως χαρακτηριστικό της κίνησης CBR το μέγεθος του πακέτου να ανέρχεται στα 1500 bytes. Επομένως το δίκτυό μας εξακολουθεί να στέλνει 1500 bytes τη φορά αλλά εμπεριέχονται σε 2 πακέτα αντί για 1 που συνέβαινε μέχρι πριν την αφαίρεση των 2 γραμμών του κώδικα.

Ο συνολικός κώδικας που μας οδήγησε στα παραπάνω συμπεράσματα και στη δημιουργία των animation μαζί με τις 2 τελευταίες γραμμές που σβήσαμε είναι ο ακόλουθος:

# ΜΕΡΟΣ 2: Στατική και δυναμική δρομολόγηση

Στο μέρος αυτό της άσκησης θα εξετάσουμε πως συμπεριφέρεται το δίκτυο όταν χαρακτηρίζεται από δυναμική δρομολόγηση. Μέχρι τώρα όσα βλέπαμε αναφέρονταν σε στατική δρομολόγηση. Προκειμένου να εξετάσουμε τη νέα συμπεριφορά του δικτύου (δυναμική) διακόπτουμε τη ζεύξη που ενώνει τους κόμβους n1 και n2 στο χρονικό διάστημα 1.7-2.7 sec. Τρέχοντας λοιπόν τώρα το animation παρατηρούμε πως το διάστημα 1.7-2.7 sec η ζεύξη κόβεται και τα πακέτα που στέλνονται χάνονται. Παρόλα αυτά οι δύο κόμβοι εξακολουθούν να εκπέμπουν πακέτα. Το στιγμιότυπο που το δείχνει αυτό είναι (τη στιγμή 1.71 sec):

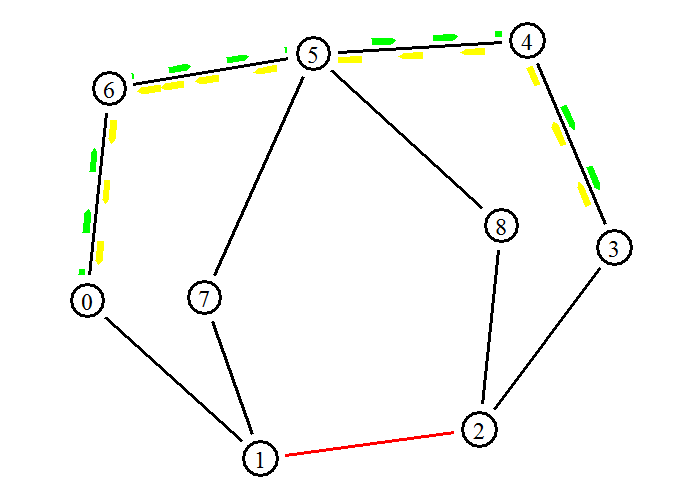


Μετά το πέρας των 2.7 sec αποκαθίσταται η ζεύξη και η ροή δεδομένων συνεχίστηκε κανονικά.

Τώρα θα κάνουμε τη δρομολόγηση δυναμική ώστε να δούμε τι αποτέλεσμα θα έχει στην αποστολή δεδομένων. Η δυναμική δρομολόγηση σημαίνει πως οι κόμβοι είναι σε θέση να καθορίζουν αυτόματα τα θέματα και την πορεία της δρομολόγησης ώστε αν αντιμετωπίζουν μόνοι τους προβλήματα όπως το “κόψιμο” μια ζεύξης. Για να το επιτύχουμε αυτό προσθέτουμε τις εξής γραμμές κώδικα :

Agent/rtProto/Direct set preference\_ 200

$ns rtproto DV

Με την προσθήκη του κώδικα αυτού το δίκτυο μας τις στιγμές που είναι κομμένη η ζεύξη αναδρομολογεί τα δεδομένα μέσω των κόμβων 0-6-5-4-3 ώστε να μην τα χάσει. Αυτό φαίνεται και στο παρακάτω στιγμιότυπο του NAM:

**2.1: Ερωτήσεις**

* **Εξηγήστε γιατί, με τη στατική δρομολόγηση, οι κόμβοι εξακολουθούν να στέλνουν πακέτα και μετά τη διακοπή της ζεύξης.**

Όταν έχουμε δίκτυο με στατική δρομολόγηση δεν κυκλοφορούν πακέτα μεταξύ των κόμβων που να περιέχουν την πληροφορία ότι μια ζεύξη διακόπηκε. Επομένως οι κόμβοι δεν ενημερώνονται για τη διακοπή αυτή (μεταξύ των κόμβων n1 και n2) και συνεχίζουν να στέλνουν πακέτα όπως έκανα και πριν , μόνο που κατά τη διάρκεια της διακοπής τα πακέτα αυτά χάνονται.

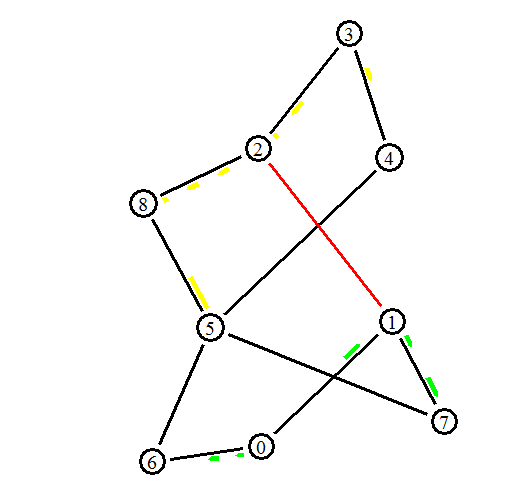
* **Τα πακέτα που χάθηκαν, θα ξαναμεταδοθούν από τους αντίστοιχους κόμβους, όταν επανέλθει η σύνδεση;**

Όχι τα πακέτα που χάνονται δε θα επαναμεταδοθούν από τους αντίστοιχους κόμβους, καθώς το δίκτυο με στατική δρομολόγηση δεν ενημερώνει τους κόμβους για τη διακοπή της ζεύξης ούτε και για την απώλεια των πακέτων δεδομένων.

* **Τι παρατηρείτε όταν γίνεται διακοπή ζεύξης και έχουμε δυναμική δρομολόγηση; Περιγράψτε με απλά λόγια τη διαδικασία που λαμβάνει χώρα στο animation. Συμπίπτει η αρχική με τη μόνιμη διαδρομή δρομολόγησης για τις δύο ροές κατά τη διάρκεια της διακοπής;**

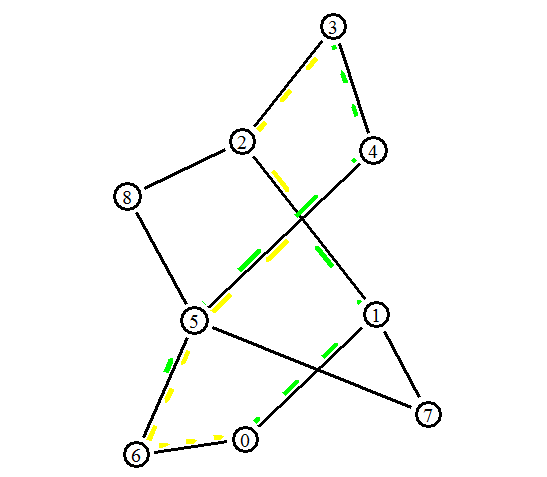
Όταν έχουμε δυναμική δρομολόγηση, με τη διακοπή της ζεύξης αλλάζει η πορεία ροής των δεδομένων. Συγκεκριμένα, τα πακέτα που έχουν την πληροφορία για την ζεύξη, ενημερώνουν τους κόμβους αποστολής n0 και n3 ότι πραγματοποιείται διακοπή της ζεύξης μεταξύ των κόμβων 1-2 και έτσι αναζητείται μια νέα συντομότερη διαδρομή καθώς οι κόμβοι γνωρίζουν ότι αυτή που ακολουθούσαν ως τώρα διεκόπη. Όσα πράσινα πακέτα τη στιγμή της διακοπής έχουν φτάσει από τον κόμβο n0 στον n1 ακλουθούν τη διαδρομή 0–1–7–5–4–3 και όσα κίτρινα έχουν πάει από τον κόμβο n3 στον n2 ακλουθούν την εναλλακτική διαδρομή 3-2-8-5-6-0 για να φτάσουν στον προορισμό τους, η οποία είναι η πιο σύντομη από τον κόμβο που βρίσκεται το πακέτο κατά τη διακοπή. Ο κόμβος n2, ο οποίος είναι ο κοντινότερος στο σημείο που διεκόπη η ζεύξη ενημερώνει και τους υπόλοιπους για αναζήτηση σύντομης νέας διαδρομής. Έτσι, μετά από λίγο τα πακέτα με την πληροφορία της δρομολόγησης ενημερώνουν τους κόμβους αποστολής ότι η συντομότερη διαδρομή είναι μέσω των κόμβων 0–6–5–4–3 και έτσι και οι δύο ροές δεδομένων ακολουθούν αυτήν την πορεία. Όταν αποκατασταθεί η ζεύξη 1 – 2, τα πακέτα ενημερώνουν και πάλι τους κόμβους και έτσι η κατάσταση του δικτύου επανέρχεται στην αρχική, όπως είχε και πριν τη διακοπή ζεύξης 1-2.

* **Με βάση το animation, προσδιορίστε για κάθε ροή τη χρονική στιγμή όπου παρατηρείται η μόνιμη διαδρομή δρομολόγησης κατά τη διάρκεια διακοπής.**
* Από 0 έως 0.2sec το δίκτυο, όπως το έχουμε ορίσει, δεν μεταφέρει κανένα δεδομένο.
* Από 0.2 έως 0.7sec αποστέλλονται τα κίτρινα πακέτα απ’ τον 0 στον 3 μέσω της διαδρομής 0-1-2-3 που είναι η συντομότερη.
* Από 0.7 έως 1.7sec ξεκινάνε να στέλνονται και τα πράσινα πακέτα απ’ τον 3 στον 0 μέσω της διαδρομής 3-2-1-0 που είναι επίσης η συντομότερη δυνατή.
* Από 1.7 έως 2.7sec διακόπτεται η ζεύξη 1-2, όπως ορίζουμε στον κώδικα, και η ανταλλαγή δεδομένων γίνεται μέσω της διαδρομής 0-6-5-4-3. (πλην των πρώτων στιγμών που τα κίτρινα πακέτα που βρίσκονται στον κόμβο 2 μεταφέρονται μέσω της διαδρομής 3-2-8-5-6-0 ενώ τα πράσινα που βρίσκονται ήδη στον κόμβο 1 ακολουθούν την πιο σύντομη για αυτά εκείνη τη στιγμή διαδρομή, δηλαδή τη 0–1–7–5–4–3 ) Ύστερα από μια μεταβατική κατάσταση, την t=1.847368sec μονιμοποιείται η παραπάνω διαδρομή. Παρακάτω φαίνεται η στιγμή διακοπής της ζεύξης που όπως ανέφερα τα πράσινα πακέτα του κόμβου 1 ακολουθούν τη διαδρομή 0–1–7–5–4–3 ενώ τα κίτρινα που έχουν ήδη φτάσει στον κόμβο 2 τη διαδρομή 3-2-8-5-6-0.



* Από 2.7 – 2.83sec έχει ξανασυνδεθεί ο 1 με τον 2 αλλά τα δεδομένα συνεχίζουν στην ίδια διαδρομή 0-6-5-4-3.
* Από 2.83 και μετά, τα δεδομένα επαναπροσδιορίζουν την διαδρομή και ακολουθούν ξανά την 0-1-2-3, που είναι εκ νέου η συντομότερη δυνατή αφού η ζεύξη 1-2 αποκαταστάθηκε. Η μεταβατική κατάσταση τελειώνει την t=2.996702sec, δηλ. ύστερα από αυτή τη στιγμή έχουμε την μόνιμη μεταφορά απ’ την διαδρομή 0-1-2-3.

Η μεταβατική κατάσταση που αναφέρθηκε παρουσιάζεται παρακάτω:



* Η προσομοίωση σταματά την t=3.7sec για τα κίτρινα πακέτα και την t=4.2sec για τα πράσινα.
* **Για ποιο λόγο τα πακέτα ακολουθούν τις συγκεκριμένες διαδρομές αφότου πέσει η σύνδεση;**

Τα πακέτα σε αυτό το δυναμικό δίκτυο ακολουθούν τις συγκεκριμένες διαδρομές , καθώς ενημερώνονται οι κόμβοι για τη διακοπή της ζεύξης 1-2 και με δεδομένες τις υπόλοιπες που παραμένουν ενεργές η διαδρομή αυτή είναι η συντομότερη, όπως πληροφορούνται οι κόμβοι με βάση τα πακέτα που περιέχουν την πληροφορία της δρομολόγησης.

* **Θα μπορούσαν να δρομολογηθούν από άλλους κόμβους;**

Τα πακέτα που αποστέλλονται προφανώς και θα μπορούσαν να ακολουθήσουν οποιαδήποτε άλλη διαδρομή που συνδέει τον κόμβο-αποστολέα και τον κόμβο-λήπτη. Παρόλα αυτά έχουμε εμείς ορίσει το δίκτυο μας να επιλέγει τη συντομότερη δυνατή διαδρομή.

* **Ποιος από όλους τους κόμβους καθορίζει από ποια διαδρομή θα προωθηθούν κάθε φορά τα πακέτα;**

Τα πακέτα που περιέχουν τις πληροφορίες δρομολόγησης σύμφωνα με τον κώδικα ακολουθούν το πρωτόκολλο Direct. Αυτό θα πει πως ο κόμβος όπου βρίσκεται κάθε φορά το κάθε πακέτο καλείται και αποφασίζει προς τα πού θα προωθηθεί το πακέτο, πάντα ανάλογα με την κατάσταση των γειτονικών κόμβων που γνωρίζει.

# 

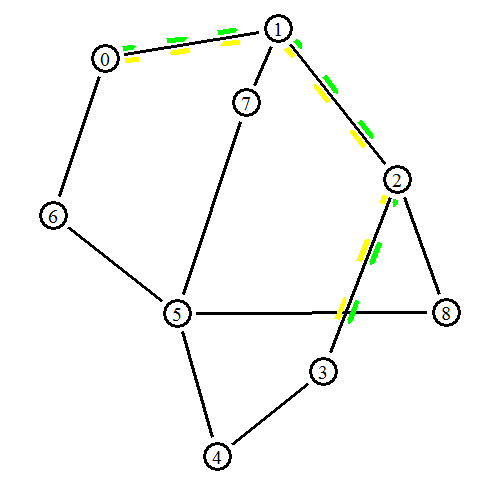
Μέρος 3: Καθορισμός κόστους ζεύξης

Μέχρι στιγμής , το δίκτυο θεωρεί μια διαδρομή ως συντομότερη δυνατή όταν αυτή περιέχει το μικρότερο αριθμό ζεύξεων. Αυτό ισχύει γιατί το κόστος κάθε ζεύξης ήταν ίσο με τη μονάδα. Όμως στην πραγματικότητα η διαδρομή επιλέγεται με βάση το κόστος των ζεύξεων. Θα θεωρήσουμε εδώ όπως αναφέρεται στην εκφώνηση πως το κόστος μια ζεύξης είναι ανάλογο της καθυστέρησης και ορίζεται από τον τύπο d/10 , όπου d είναι η καθυστέρηση της ζεύξης σε ms. Για να το κάνουμε αυτό συμπληρώνουμε στον κώδικά μας εντολές και το μορφοποιούμε κατάλληλα όπως εξηγείται στην εκφώνηση. (ο κώδικας θα παρατεθεί στο τέλος της άσκησης).

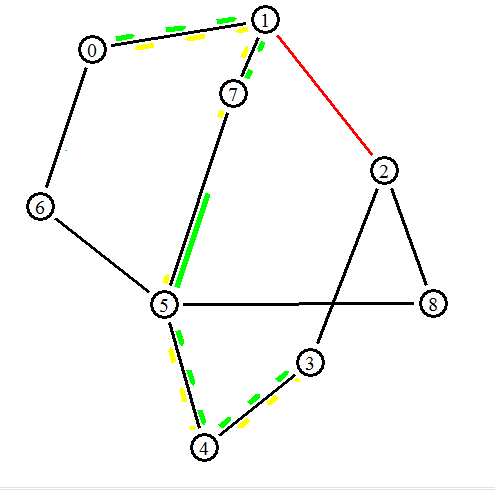
**3.1: Ερωτήσεις**

* **Ποιες διαδρομές ακολουθούν τα πακέτα πριν, μετά και κατά τη διάρκεια της πτώσης της σύνδεσης;**

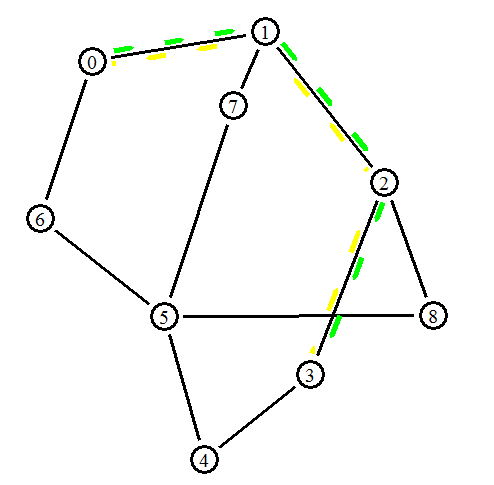
Πριν τη διακοπή της ζεύξης τα πακέτα ακολουθούν τη διαδρομή: 0-1-2-3, όπως φαίνεται και παρακάτω:



Κατά τη διακοπή της ζεύξης τα πακέτα ακολουθούν τη διαδρομή :0-1-7-5-4-3. Σε animation:



Μετά που επανέρχεται η ζεύξη τα δεδομένα ακολουθούν και πάλι τη διαδρομή 0-1-2-3. Δηλαδή το δίκτυο δείχνει ως εξής:



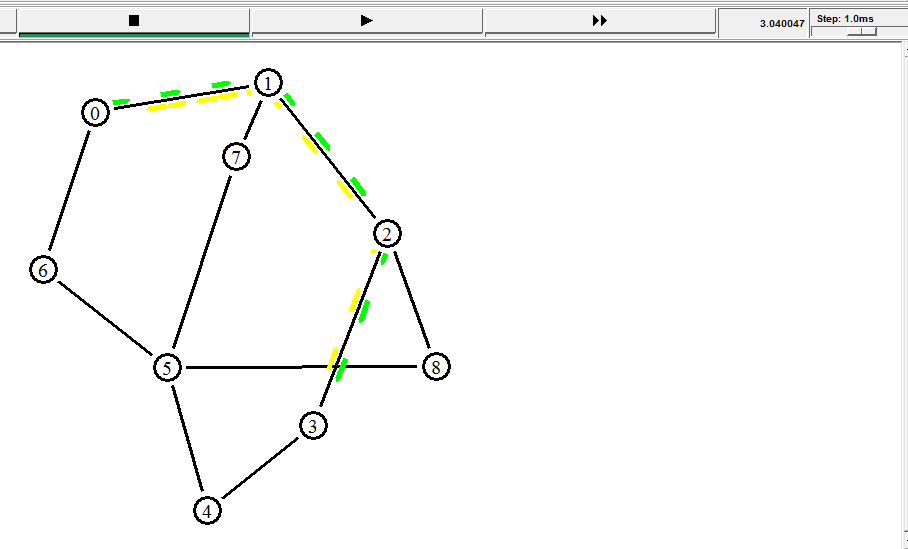
* **Για ποιον λόγο τα πακέτα ακολουθούν τις συγκεκριμένες διαδρομές;**

Τα πακέτα θα ακολουθούν τώρα τις διαδρομές με το ελάχιστο δυνατό κόστος κάθε δεδομένη στιγμή. Επομένως πριν τη διακοπή της ζεύξης η διαδρομή που ακολουθεί έχει κόστος : 40/10+40/10+40/10 =12 που είναι το μικρότερο όλων των δυνατών διαδρομών. Επίσης, κατά τη διακοπή της ζεύξης πάλι ακολουθείται η διαδρομή με το ελάχιστο κόστος, καθώς η διαδρομή 0-1-7-5-4-3 έχει κόστος 4+2+1+4+4=15, που για τη δεδομένη περίοδο έχει όντως το μικρότερο κόστος. Όταν επανέρχεται η σύνδεση, τα πακέτα συνεχίζουν να ακλουθούν για μικρό διάστημα τη διαδρομή 0-1-7-5-4-3 που δεν έχει το ελάχιστο κόστος, καθώς μεσολαβεί κάποιο χρονικό διάστημα μέχρι να ενημερωθούν οι δρομολογητές ότι ξανά είναι διαθέσιμη η οικονομικότερη διαδρομή από πλευρά κόστους.

* **Θα μπορούσαν να δρομολογηθούν από άλλους κόμβους;**

Φυσικά και θα μπορούσαν. Όμως με βάση τις εντολές που έχουν δοθεί στο πρόγραμμα το δίκτυο αναζητεί τη συντομότερη διαδρομή από άποψη κόστους. Επομένως, θα μπορούσαμε να αλλάξουμε τη δρομολόγηση των πακέτων αν μεταβάλλαμε τα κόστη των ζεύξεων καθώς τότε το πρόγραμμά μας θα αναζητούσε εκ νέου τη βέλτιστη διαδρομή η οποία θα μπορούσε να διαφέρει από αυτή που έχουμε τώρα.

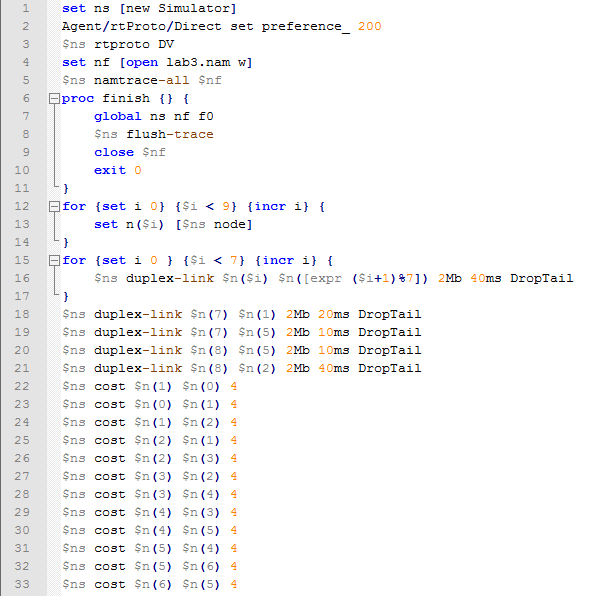
* **Μετά την αποκατάσταση της ζεύξης μεταξύ των κόμβων “1” και “2”, προσδιορίστε με βάση το animation τη χρονική στιγμή όπου παρατηρείται η μόνιμη διαδρομή δρομολόγησης για κάθε ροή.**

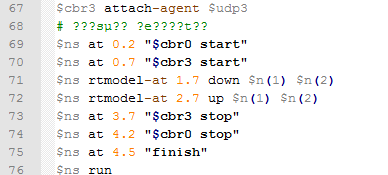
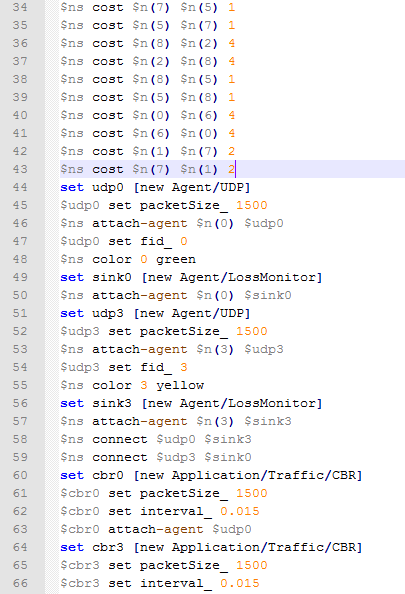


Όπως φαίνεται από το παραπάνω στιγμιότυπο του animation, τη στιγμή 3,040047sec μονιμοποιείται εκ νέου η διαδρομή 0-1-2-3. Δηλαδή παρατηρούμε πως παρά το γεγονός ότι η ζεύξη 1-2 αποκαταστάθηκε ήδη από τα 2.7 sec υπήρξε μια μεταβατική κατάσταση που έλαβε τέλος την παραπάνω χρονική στιγμή που παρουσιάζω.

* **Ποιος είναι ο ρόλος της εντολής Agent/rtProto/Direct set preference\_ 200; Τι παρατηρείτε στη δρομολόγηση των πακέτων αν αφαιρεθεί η εντολή αυτή από τον κώδικα της προσομοίωσης; Αιτιολογείστε γιατί συμβαίνει αυτό.**

Η εντολή αυτή στην ουσία ορίζει το πρωτόκολλο με το οποίο γίνεται η δυναμική δρομολόγηση του δικτύου. Η ρύθμιση στο Direct στην ουσία δίνει τη δυνατότητα στους κόμβους να συλλέγουν στοιχεία για τους γειτονικούς τους κόμβους οπότε να ξέρουν πώς να ανταποκριθούν αν μια ζεύξη διακοπεί. Αν από το δίκτυό μας αφαιρεθεί αυτή η γραμμή κώδικα αυτό που μπορούμε να παρατηρήσουμε μέσω του animation είναι ότι με την εντολή αυτή η αποστολή πακέτων ξεκινά τη χρονική στιγμή 2.834 sec ενώ χωρίς τη χρονική στιγμή 2.858sec.

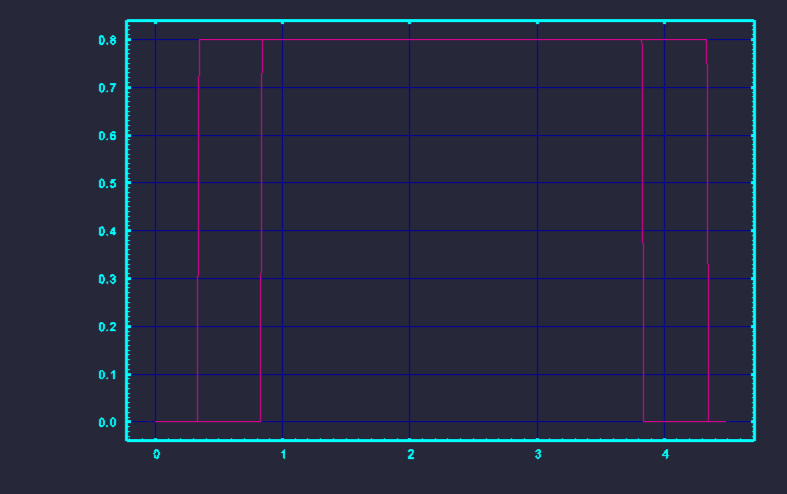
Ο κώδικας τη άσκησης που προηγήθηκε είναι:

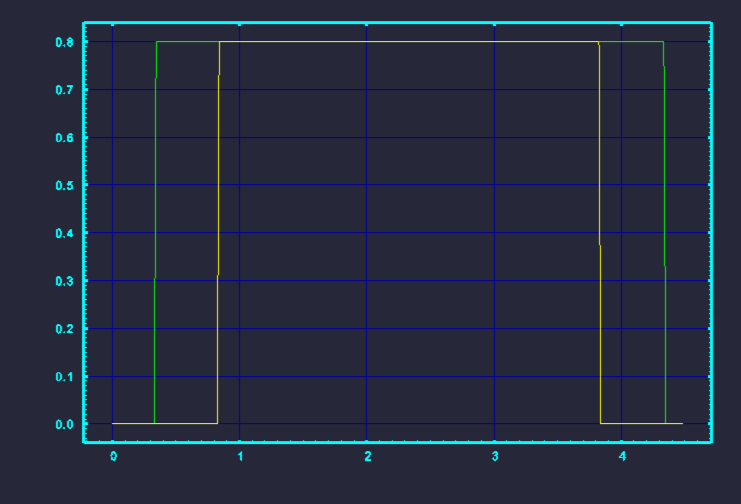


Μέρος 4: Παρακολούθηση εκθετικής κίνησης με το Xgraph

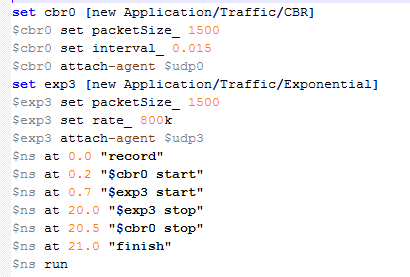
Στο 4ο και τελευταίο αυτό κομμάτι της άσκησης θα παρουσιάσουμε με γραφικές παραστάσεις την κίνηση των δεδομένων του δικτύου μας. Αυτό θα γίνει όπως και στις 2 προηγούμενες ασκήσεις με το Xgraph. Ακολουθώντας τις οδηγίες της εκφώνησης διαμορφώνουμε τον κώδικα κατάλληλα ώστε τρέχοντας των να μας δίνει τις γραφικές που ζητάμε. Παραλείπουμε επίσης τις εντολές που ορίζουν τη διακοπή και την αποκατάσταση της ζεύξης 1-2 προκειμένου να μην υπάρχουν προβλήματα με τη ροή δεδομένων.

Τρέχοντας στο xgraph το παραπάνω παίρνω:

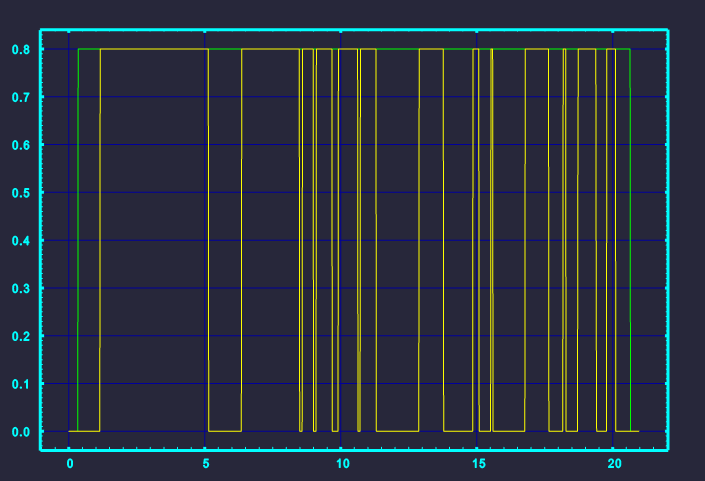


Προσθέτοντας στην αρχή των αρχείων out0.tr και out3.tr τις εντολές color = green & color = yellow αντίστοιχα ώστε να ξεχωρίζω για ποια ροή δεδομένων είναι ποια γραφική παίρνω:

Μετά από αυτή την αλλαγή χρώματος, αλλάζω την κίνηση του ενός αποστολέα σταθερής ροής(CBR) και την κάνω εκθετική βάζοντας Exponential όπου υπάρχει CBR. Σύμφωνα με την εκφώνηση για να λάβουμε σωστά αποτελέσματα και να εξάγουμε συμπεράσματα πρέπει να αυξήσουμε το χρόνο αποστολής της κίνησης και ανάλογα της προσομοίωσης τουλάχιστον σε 20 δευτερόλεπτα. Αυτό επιτυγχάνεται τροποποιώντας ένα κομμάτι του κώδικα και μεταβάλλοντας τις παραμέτρους του όπως δείχνω παρακάτω:



Ξανατρέχοντας το script λαμβάνω την εξής γραφική:



**4.1: Ερωτήσεις**

* **Ποιος είναι ο μέγιστος ρυθμός μετάδοσης που επιτυγχάνεται για τις δύο περιπτώσεις κίνησης, βάσει των γραφικών παραστάσεων που σχεδιάσατε;**

Παρατηρώντας τις παραπάνω γραφικές παραστάσεις παρατηρούμε ότι τόσο στην περίπτωση της CBR κίνησης δεδομένων όσο και της εκθετικής έχουμε μέγιστο ρυθμό μετάδοσης 0.8 Mbps.

* **Αιτιολογείστε τις μέγιστες τιμές που προσδιορίσατε παραπάνω, χρησιμοποιώντας τις παραμέτρους που θέσατε για τη διαμόρφωση των δύο πηγών κίνησης (CBR και Exponential).**

Στην περίπτωση της CBR κίνησης δεδομένων έχουμε ορίσει να στέλνονται πακέτα των 1500bytes ανά 0.015 sec. Επομένων για αυτή την περίπτωση ο μέγιστος ρυθμός μετάδοσης δεδομένων τόσο για τη ροη 0->3 όσο και για τη 3->0 θα είναι 1500/0.015\*8=0.8Mbps.

Στην περίπτωση της εκθετικής κίνησης δεδομένων ήδη από τον κώδικα έχουμε ορίσει το μέγιστο ρυθμό μετάδοσης δεδομένων από την εντολή

$exp3 set rate\_ 800k. (800Kbps=0.8Mbps)

* **Υπολογίστε το πλήθος των bytes που λαμβάνονται επιτυχώς στον προορισμό για κάθε ροή, θεωρώντας ότι και οι δύο ροές ολοκληρώνονται σε χρόνο t=20+(a/10)sec, όπου a τα δύο τελευταία ψηφία του αριθμού μητρώου σας.**

Και οι δύο ροές ολοκληρώνονται σε χρόνο t= 20+17/10=21.7 sec.Εφόσον κάθε 0,015sec στέλνονται 1500 bytes, τότε στα 21.7sec στέλνονται

* Για τη ροή CBR στους κόμβους 0->3:

(1500\*(21.7-0.2))/0,015 = 2150000bytes

(αφού t=0.2 sec ο χρόνος εκκίνησης του CBR agent του κόμβου 0.)

* Για τη ροή CBR στους κόμβους 3->0:

(1500\*(21.7-0.7))/0,015 = 2100000bytes

(αφού t=0.7 sec ο χρόνος εκκίνησης του CBR agent του κόμβου 3.)

* Για τη ροή Exponential στους κόμβους 3->0:

(21.7-0.7)\*800=16800 Kbytes

(αφού t=0.7 sec ο χρόνος εκκίνησης του Exponential agent του κόμβου 3.)