



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

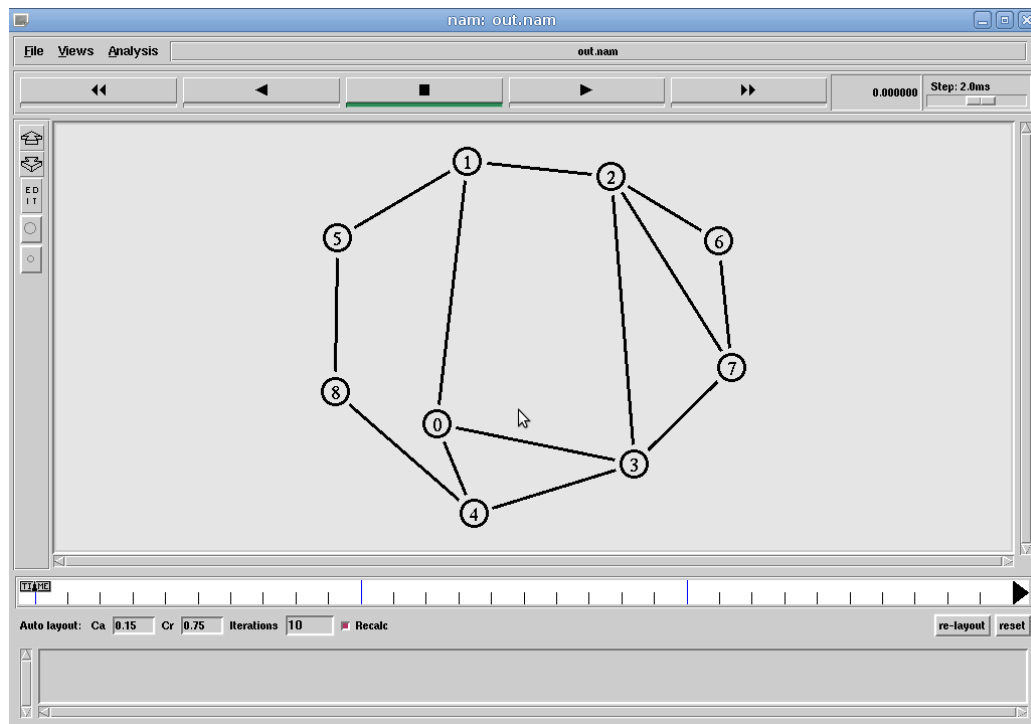
ΣΧΟΛΗ ΗΜ&ΜΥ
Δίκτυα Επικοινωνιών

3^η Άσκηση
Ακ. έτος 2011-2012

Γρηγόρης Λύρας Α.Μ.: 03109687

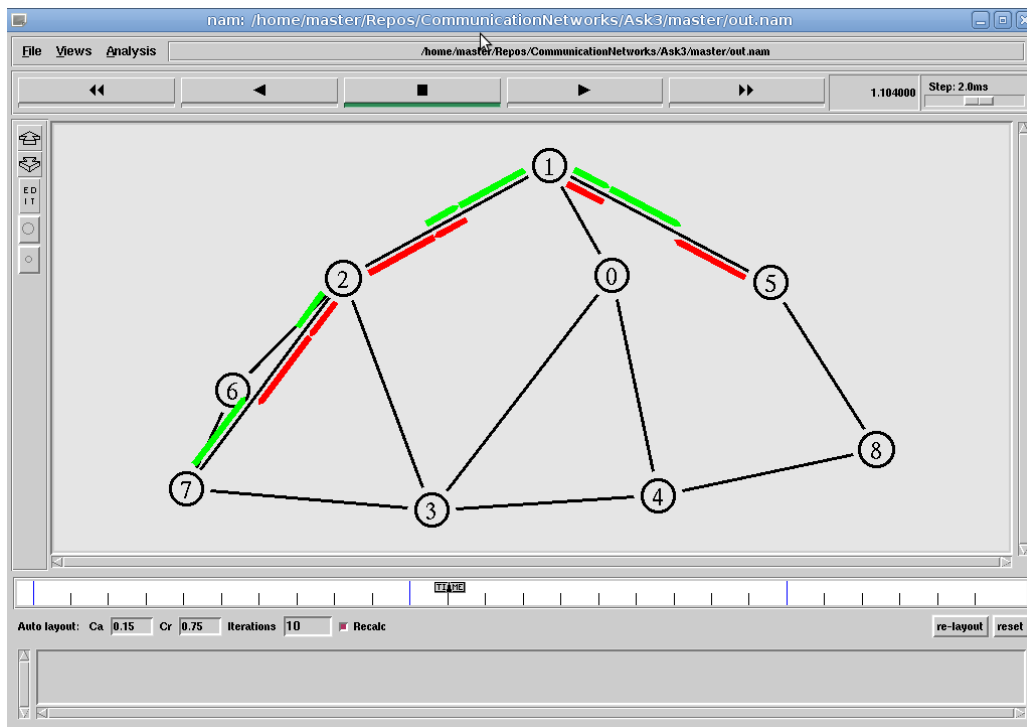
9 Μαΐου 2012

I.



Τρέχουμε το animation και παρατηρούμε την εξέλιξη της πορείας μετάδοσης των πακέτων:

- Αρχικά, για $t < 0.5\text{sec}$, δεν έχουμε ροή πακέτων.
- Για $0.5\text{sec} < t < 1\text{sec}$, έχουμε ροή πακέτων μόνο από τον 5 στον 7, μέσω της μικρότερης διαδρομής ($5 \rightarrow 1 \rightarrow 2 \rightarrow 7$) με κόκκινα πακέτα.
- Για $1\text{sec} < t < 2\text{sec}$ έχουμε και ροή πράσινων πακέτων από τον κόμβο 7 προς τον 5, ακολουθώντας την αντίστροφη διαδρομή από προηγουμένως.
- Η μετάδοση των κόκκινων πακέτων διακόπτεται σε χρόνο 2sec, ενώ η μετάδοση των πράσινων πακέτων σε 2.5sec. Τέλος, για $t = 3\text{sec}$, ολοκληρώνεται η όλη εφαρμογή, μέσα από τη διαδικασία finish.



Τα πακέτα ακολουθούν τη διαδρομή (5 -> 1 -> 2 -> 7) για να φτάσουν στον προορισμό τους. Η διαδρομή αυτή είναι η συντομότερη δυνατή. Ακολουθείται και κατά τις δύο κατευθύνσεις και δεν υπάρχει συντομότερη διαδρομή με την παρούσα συνδεσμολογία μας και το κόστος στις ζεύξεις είναι ίσο και σταθερό.

```
set ns [new Simulator]
set nf [open out.nam w]
$ns namtrace-all $nf

for {set i 0} {$i < 9} {incr i} {
    set n($i) [$ns node]
}

for {set i 0} {$i < 9} {incr i} {
    $ns duplex-link $n($i) $n([expr ($i+1)%5]) 2Mb 10ms DropTail
}

$ns duplex-link $n(3) $n(0) 2Mb 10ms DropTail
$ns duplex-link $n(5) $n(8) 2Mb 10ms DropTail
$ns duplex-link $n(6) $n(7) 2Mb 10ms DropTail
$ns duplex-link $n(2) $n(7) 2Mb 10ms DropTail

set udp5 [new Agent/UDP]
$ns attach-agent $n(5) $udp5
$udp5 set fid_ 5
$ns color 5 red
set sink5 [new Agent/LossMonitor]
$ns attach-agent $n(5) $sink5

set udp7 [new Agent/UDP]
$ns attach-agent $n(7) $udp7
$udp7 set fid_ 7
$ns color 7 green
set sink7 [new Agent/LossMonitor]
$ns attach-agent $n(7) $sink7

# Σ
$ns connect $udp5 $sink7
$ns connect $udp7 $sink5
# Σ

set cbr5 [new Application/Traffic/CBR]
$cbr5 set packetSize_ 1500
$cbr5 set interval_ 0.01
$cbr5 attach-agent $udp5
set cbr7 [new Application/Traffic/CBR]
$cbr7 set packetSize_ 1500
```

```
$cbr7 set interval_ 0.01
```

```
$cbr7 attach-agent $udp7
```

```
proc finish {} {  
    global ns nf  
    $ns flush-trace  
    close $nf  
    exit 0  
}
```

```
$ns at 0.5 "$cbr5 start"  
$ns at 1 "$cbr7 start"  
$ns at 2 "$cbr5 stop"  
$ns at 2.5 "$cbr7 stop"  
$ns at 3 "finish"
```

```
$ns run
```

II.

Σε αυτήν ενότητα θα δούμε τη διαφορά μεταξύ στατικής και δυναμικής δρομολόγησης και την εκάστοτε αντιμετώπιση από το δίκτυο. Μεταβάλλουμε λίγο τον κώδικα, διακόπτοντας την σύνδεση ανάμεσα σε 1 και 2 για 0.4sec.

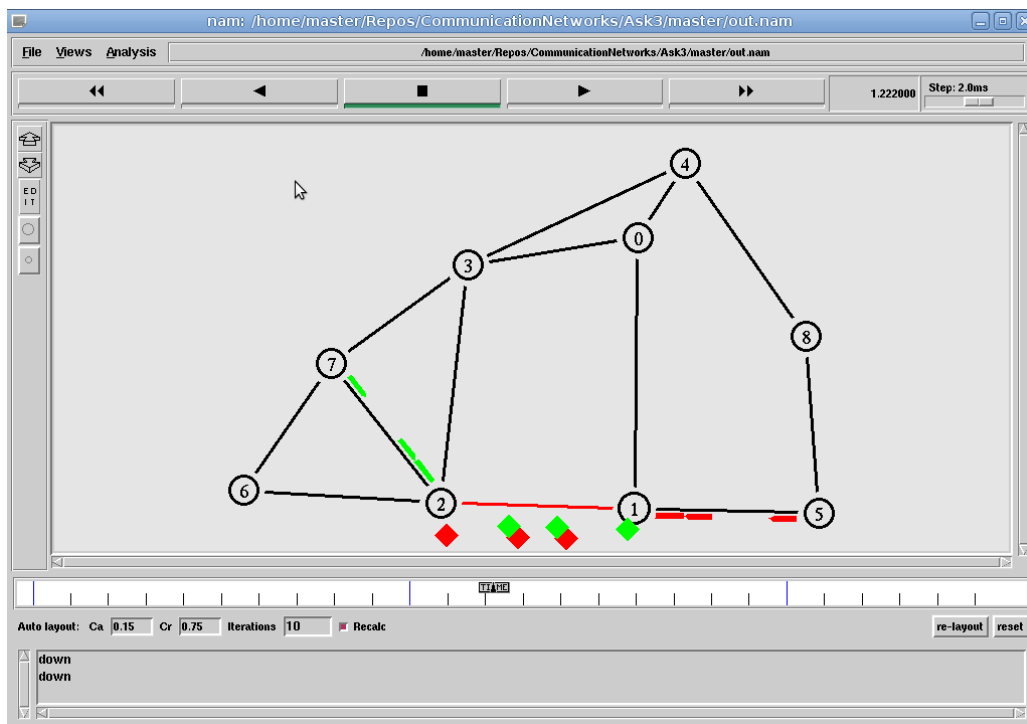
```
set ns [new Simulator]  
set nf [open out.nam w]  
$ns namtrace-all $nf  
  
for {set i 0} {$i < 9} {incr i} {  
    set n($i) [$ns node]  
}  
  
for {set i 0} {$i < 9} {incr i} {  
    $ns duplex-link $n($i) $n([expr ($i+1)%5]) 2Mb 10ms DropTail  
}  
$ns duplex-link $n(3) $n(0) 2Mb 10ms DropTail  
$ns duplex-link $n(5) $n(8) 2Mb 10ms DropTail  
$ns duplex-link $n(6) $n(7) 2Mb 10ms DropTail  
$ns duplex-link $n(2) $n(7) 2Mb 10ms DropTail  
  
set udp5 [new Agent/UDP]  
$ns attach-agent $n(5) $udp5  
$udp5 set fid_ 5  
$ns color 5 red  
set sink5 [new Agent/LossMonitor]  
$ns attach-agent $n(5) $sink5  
  
set udp7 [new Agent/UDP]  
$ns attach-agent $n(7) $udp7  
$udp7 set fid_ 7  
$ns color 7 green  
set sink7 [new Agent/LossMonitor]  
$ns attach-agent $n(7) $sink7  
  
# Σ  
$ns connect $udp5 $sink7  
$ns connect $udp7 $sink5  
# Σ  
set cbr5 [new Application/Traffic/CBR]  
$cbr5 set packetSize_ 1500  
$cbr5 set interval_ 0.01  
$cbr5 attach-agent $udp5  
set cbr7 [new Application/Traffic/CBR]  
$cbr7 set packetSize_ 1500  
$cbr7 set interval_ 0.01
```

```
$cbr7 attach-agent $udp7
```

```
proc finish {} {  
    global ns nf  
    $ns flush-trace  
    close $nf  
    exit 0  
}
```

```
$ns at 0.5 "$cbr5 start"  
$ns at 1 "$cbr7 start"  
$ns rtmodel-at 1.2 down $n(1) $n(2)  
$ns rtmodel-at 1.6 up $n(1) $n(2)  
$ns at 2 "$cbr5 stop"  
$ns at 2.5 "$cbr7 stop"  
$ns at 3 "finish"
```

```
$ns run
```

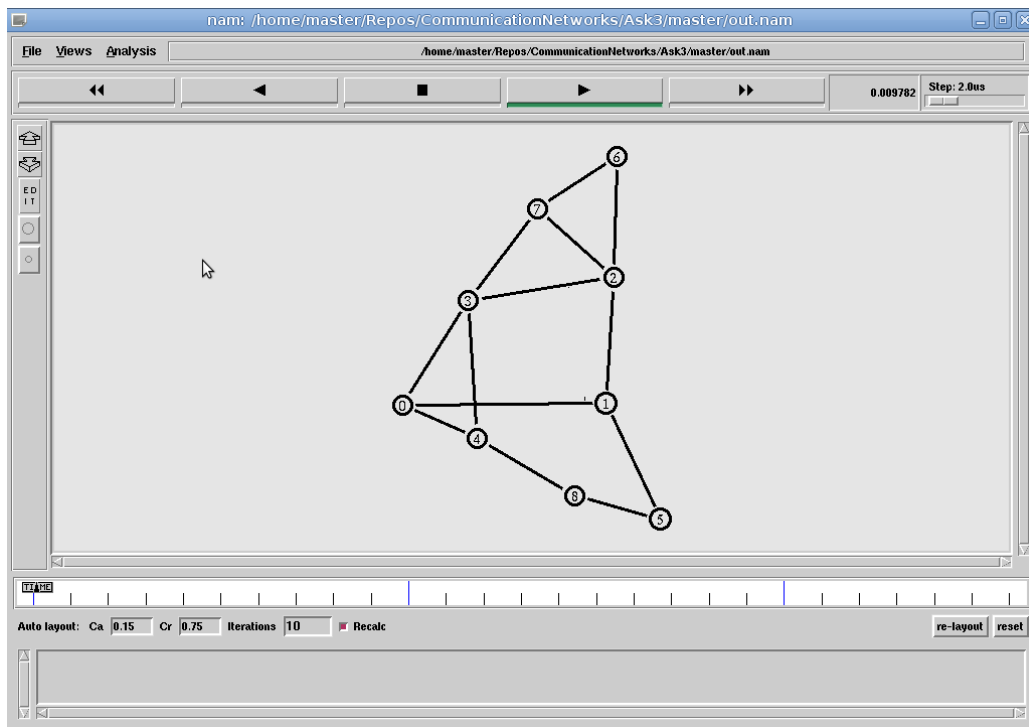


Σχήμα 1: Για χρόνο 1.2sec η σύνδεση διακόπτεται στη ζεύξη 1-2.

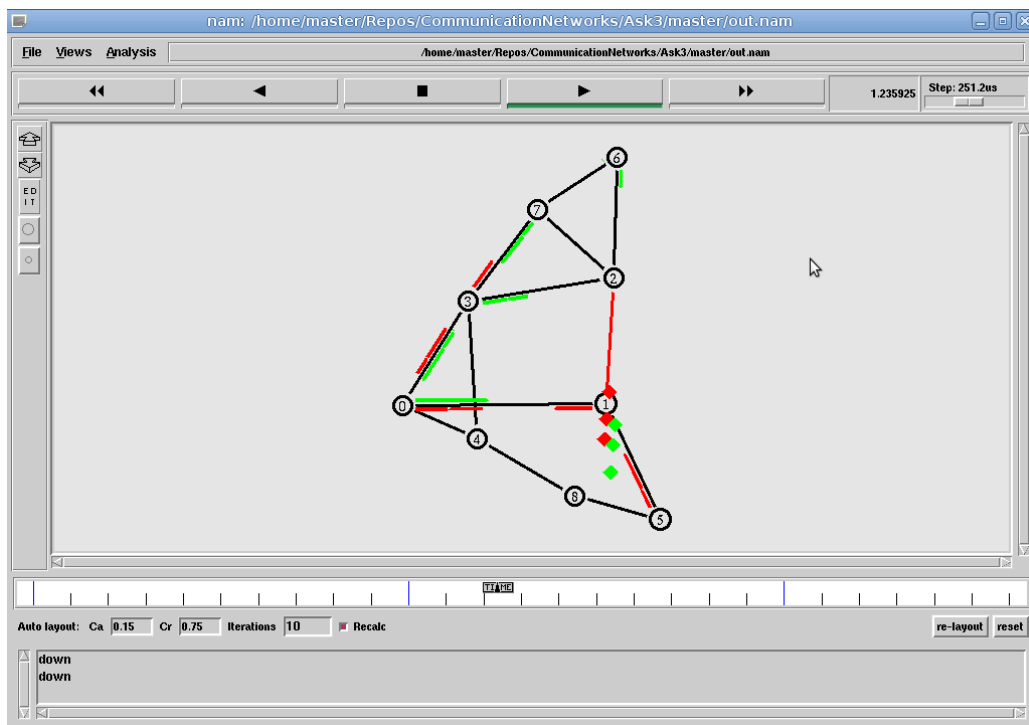
Προκειμένου να διορθώσουμε το παραπάνω πρόβλημα, αλλάζουμε τη σύνδεση σε δυναμική με την ακόλουθη εντολή:

```
$ns rtproto DV
```

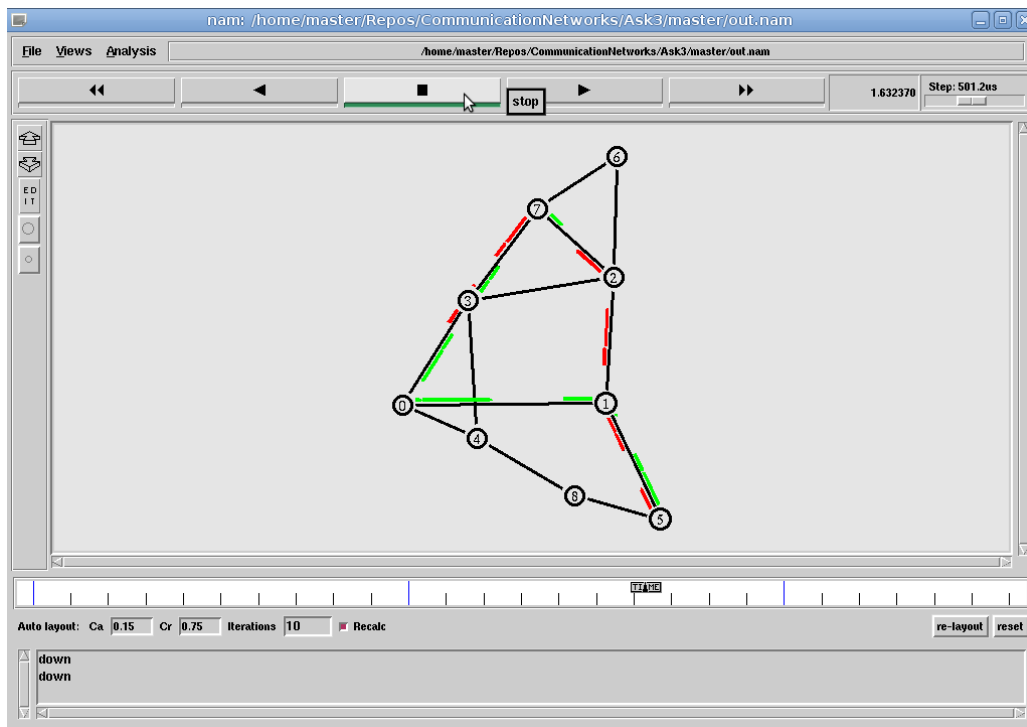
Παρατηρούμε ορισμένες αλλαγές σε σχέση με πριν. Αρχικά, εξετάζεται με δοκιμαστικά πακέτα η συνδεσιμότητα των γραμμών:



Μετά τη διακοπή της σύνδεσης 1-2 επιλέγεται εναλλακτική διαδρομή, όταν παρατηρείται η πρώτη απώλεια πακέτων:



Τέλος, μόλις η σύνδεση ενεργοποιηθεί ξανά, αρχίζει και πάλι η μετάδοση πακέτων από αυτή. Ασφαλώς υπάρχει και ένας χρόνος που μεσολαβεί κατά τον οποίο και οι δύο διαδρομές έχουν πακέτα.



- Εξηγήστε γιατί με τη στατική δρομολόγηση, οι κόμβοι εξακολουθούν να στέλνουν πακέτα μετά τη διακοπή της ζεύξης.
Καθώς, η δρομολόγηση είναι στατική, δεν ενημερώνονται οι κόμβοι με κανένα τρόπο για τη διακοπή της σύνδεσης. Επομένως δεν υπάρχει κάποια ανατροφοδότηση που θα τους οδηγούσε σε σχεδιασμό νέας διαδρομής.
- Τα πακέτα που χάνονται θα ξαναμεταδοθούν από τους αντίστοιχους κόμβους, όταν επανέλθει η σύνδεση;
Αφού δεν υπάρχει ενημέρωση σχετικά με την απώλεια τους, ασφαλώς και οι κόμβοι δεν έχουν καμία πληροφορία για το αν χάθηκαν πακέτα και ποια είναι αυτά. Οπότε τα συγκεκριμένα πακέτα δε θα μεταδοθούν ξανά όταν επανέλθει η σύνδεση.
- Τι παρατηρείτε όταν γίνεται διακοπή ζεύξης και έχουμε δυναμική δρομολόγηση;
Περιγράψτε με απλά λόγια τη διαδικασία που λαμβάνει χώρα στο animation. Στη δυναμική δρομολόγηση, υπάρχει διαρκής ενημέρωση σχετικά με τη συνδεσιμότητα της κάθε γραμμής με τη βοήθεια βοηθητικών πακέτων. Αρχικά επιλέγεται η διαδρομή 5-1-2-7 που είναι και η συντομότερη. Μόλις κοπεί η σύνδεση 1-2, αμέσως βρίσκεται η συντομότερη διαδρομή που δε χρησιμοποιεί την αποσυνδεδεμένη ζεύξη. Αυτή είναι η 5-1-0-3-7. Προφανώς η διαδρομή αυτή είναι πιο αργή από την πρώτη. Για αυτό το λόγο, μόλις ενεργοποιηθεί ξανά η γραμμή 1-2, αυτόματα η μεταφορά πακέτων αρχίζει να ξαναγίνεται από εκεί.
- Ποιος από όλους τους κόμβους καθορίζει από ποια διαδρομή θα προωθηθούν κάθε φορά τα πακέτα;
Αρχικά η πληροφορία για τη συνδεσιμότητα φθάνει στους κόμβους που έχουν επαφή με την αποσυνδεδεμένη ζεύξη και εκεί επανακαθορίζεται η διαδρομή που θα ακολουθηθεί για τον τελικό προορισμό. Σταδιακά η πληροφορία αυτή μεταβιβάζεται και στους γειτονικούς κόμβους και μόλις φτάσει στον αρχικό κόμβο, αυτός υπολογίζει ξανά τη διαδρομή λαμβάνοντας υπ' όψιν τις αποσυνδεδεμένες ζεύξεις.
- Για ποιο λόγο τα πακέτα ακολουθούν τις συγκεκριμένες διαδρομές αφότου πέσει η σύνδεση;
Ο λόγος που γίνεται αυτό είναι επειδή οι συγκεκριμένες διαδρομές είναι οι συντομότερες για τον προορισμό που δεν χρησιμοποιούν την κομμένη ζεύξη. Επί της ουσίας, επανασχεδιάζεται η διαδρομή στον κόμβο που εντοπίζεται η διακοπή ζεύξης και, όταν η πληροφορία φτάσει στον αρχικό κόμβο σε αυτόν, για να βρεθεί ο συντομότερος τρόπος ώστε να προσεγγιστεί ο προορισμός χωρίς αυτή τη σύνδεση.
- Θα μπορούσαν να δρομολογηθούν από άλλους κόμβους;
Ασφαλώς η δρομολόγηση θα μπορούσε να γίνει και από άλλους κόμβους, με μεγαλύτερο πλήθος κόμβων να διαρρέεται αλλά κάτι τέτοιο είναι χρονοβόρο χωρίς να προσφέρει επί της ουσίας κάτι.

III.

Σε αυτό το κομμάτι της άσκησης αλλάζουμε το κόστος των διαδρομών όπως φαίνεται στον κώδικα παρακάτω.

```
set ns [new Simulator]
$ns rtproto DV
set nf [open out.nam w]
$ns namtrace-all $nf

for {set i 0} {$i < 9} {incr i} {
    set n($i) [$ns node]
}

for {set i 0} {$i < 9} {incr i} {
    $ns duplex-link $n($i) $n([expr ($i+1)%5]) 2Mb 10ms DropTail
}

$ns duplex-link $n(3) $n(0) 2Mb 40ms DropTail
$ns duplex-link $n(5) $n(8) 2Mb 40ms DropTail
$ns duplex-link $n(6) $n(7) 2Mb 30ms DropTail
$ns duplex-link $n(2) $n(7) 2Mb 20ms DropTail

$ns cost $n(3) $n(0) 4
$ns cost $n(5) $n(8) 4
$ns cost $n(6) $n(7) 3
$ns cost $n(2) $n(7) 2

$ns cost $n(0) $n(3) 4
$ns cost $n(8) $n(5) 4
$ns cost $n(7) $n(6) 3
$ns cost $n(7) $n(2) 2

set udp5 [new Agent/UDP]
$ns attach-agent $n(5) $udp5
$udp5 set fid_ 5
$ns color 5 red
set sink5 [new Agent/LossMonitor]
$ns attach-agent $n(5) $sink5

set udp7 [new Agent/UDP]
$ns attach-agent $n(7) $udp7
$udp7 set fid_ 7
$ns color 7 green
set sink7 [new Agent/LossMonitor]
$ns attach-agent $n(7) $sink7

# Σ
$ns connect $udp5 $sink7
$ns connect $udp7 $sink5
# Σ
set cbr5 [new Application/Traffic/CBR]
$cbr5 set packetSize_ 1500
$cbr5 set interval_ 0.01
$cbr5 attach-agent $udp5
set cbr7 [new Application/Traffic/CBR]
$cbr7 set packetSize_ 1500
$cbr7 set interval_ 0.01

$cbr7 attach-agent $udp7

proc finish {} {
    global ns nf
    $ns flush-trace
    close $nf
    exit 0
}

$ns at 0.5 "$cbr5 start"
$ns at 1 "$cbr7 start"
```



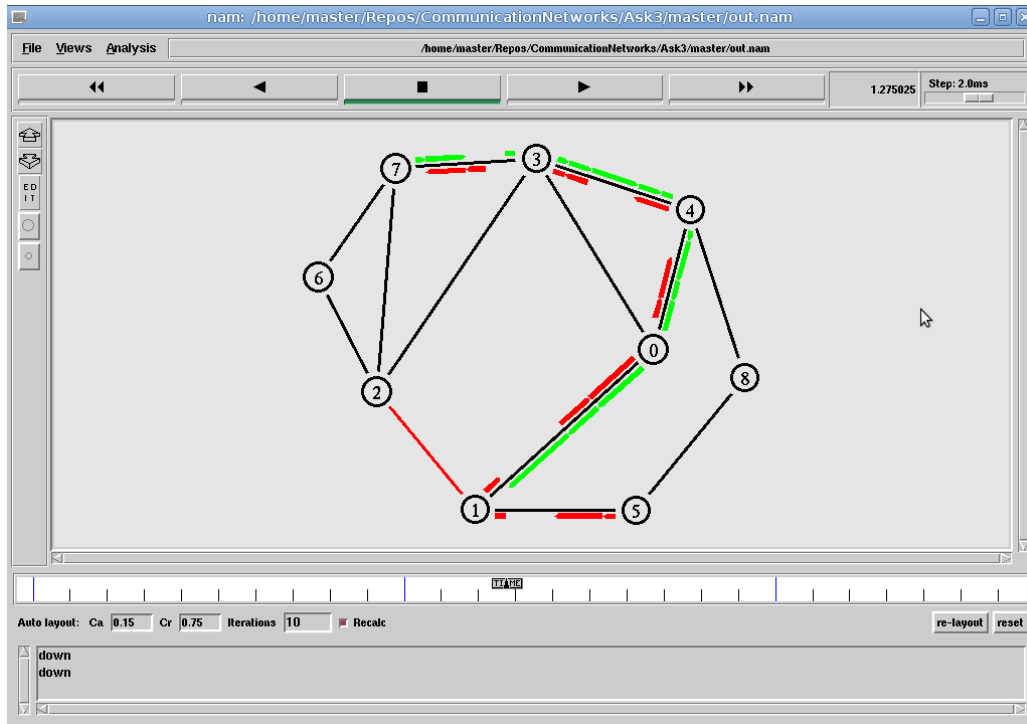
```

$ns rtmodel-at 1.2 down $n(1) $n(2)
$ns rtmodel-at 1.6 up $n(1) $n(2)
$ns at 2 "$cbr5 stop"
$ns at 2.5 "$cbr7 stop"
$ns at 3 "finish"

$ns run

```

Εκτελώντας την προσομοίωση και πάλι παρατηρούμε ότι πλέον όταν κόβεται η ζεύξη 1-2 επιλέγεται διαφορετική πορεία από πριν. Καθώς αυτή τη φορά συνοπολογίζουμε και το διαφορετικό κόστος κάθε ζεύξης 5-0-3-7.

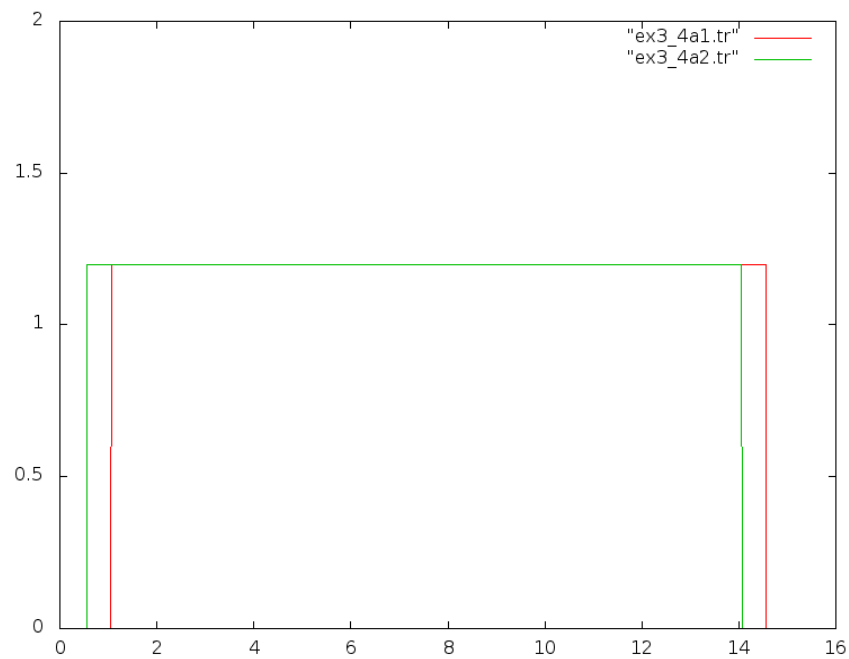


- Ποιές διαδρομές ακολουθούν τα πακέτα πριν , μετά και κατά τη διάρκεια της πτώσης της σύνδεσης;
Τα πακέτα όπως ήταν αναμενόμενο ακολουθούν πριν και μετά την πτώση της σύνδεσης 1-2 την ίδια διαδρομή με τα προηγούμενα ερωτήματα. Κατά τη διάρκεια της πτώσης επιλέγουν η διαδρομή 5-1-0-4-3-7 .
- Για ποιόν λόγο τα πακέτα ακολουθούν τις συγκεκριμένες διαδρομές;
Ο λόγος για τον οποίο τα πακέτα ακολουθούν τις παραπάνω διαδρομές είναι ότι με βάση τα βάρη που έχουν οι ζεύξεις, αυτές είναι οι διαδρομές που κοστίζουν λιγότερο.
- Θα μπορούσαν να δρομολογηθούν από άλλους κόμβους;
Προφανώς θα μπορούσαν να δρομολογηθούν από άλλους κόμβους τα πακέτα (βλ. Μέσω του κόμβου 8) αλλά αυτό θα ήταν ακριβότερο γι'αυτό επιλέγεται η παραπάνω διαδρομή.

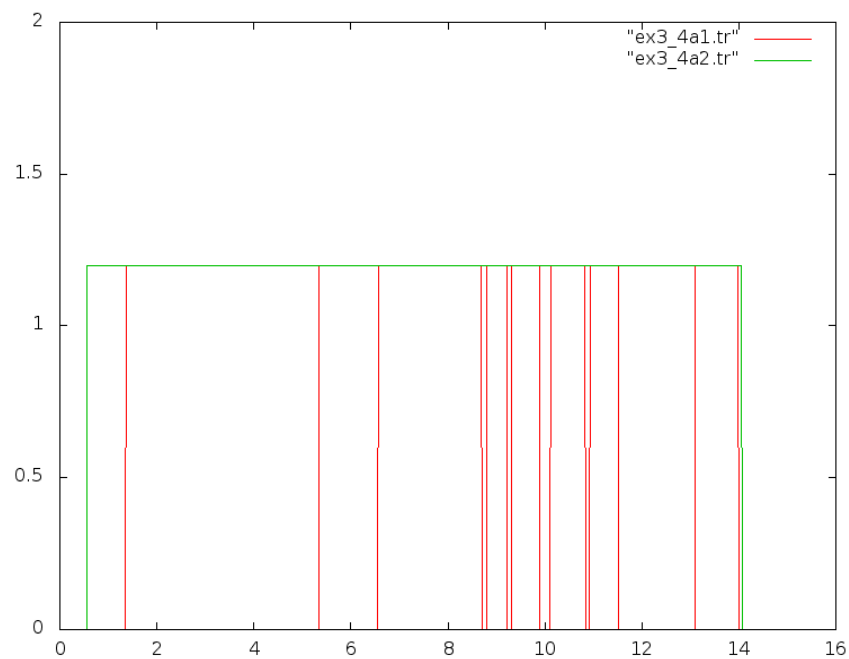
IV.

Προσθέτουμε τις κατάλληλες εντολές και ο τελικός κώδικας διαμορφώνεται όπως φαίνεται παρακάτω. Σβήνουμε τις εντολές που προκαλούν τη διακοπή της σύνδεσης όταν $t=1.2\text{sec}$ και την αποκατάστασή της όταν $t=1.6\text{sec}$ ώστε να μην υπάρχουν προβλήματα στη ροή δεδομένων. Ακόμη ρυθμίζουμε το χρόνο προσομοίωσης στα 15sec ώστε να έχουμε αντιπροσωπευτικά αποτελέσματα.

IV.1 CBR



IV.2 Exponential



- Ποιός είναι ο μέγιστος αριθμός μετάδοσης που επιτυγχάνεται για τις δύο περιπτώσεις κίνησης, με βάση τις γραφικές παραστάσεις που σχεδιάσατε;
Ο μέγιστος ρυθμός μετάδοσης είναι και για τις δύο περιπτώσεις κίνησης 0.18.
- Αιτιολογήστε τις τιμές που θέσατε παραπάνω για τις δύο τιμές κίνησης (cbr και exp);
Η παραπάνω τιμή είναι λογική αφού είναι ελαφρώς μικρότερη από το εύρος ζώνης της σύνδεσης (δηλ 2Mb).