****

Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Μάθημα: Δίκτυα Επικοινωνιών

9η Εργαστηριακή Άσκηση

Ονοματεπώνυμο : **Σταυρακάκης Δημήτριος**

ΑΜ : **03112017**

Εξάμηνο : 6ο

Ημερομηνία Παράδοσης: 3/6/2015

# communication_networks.jpg

**Ερωτήσεις**

1. *Να σχεδιάσετε την τοπολογία της προσομοίωσης, ώστε να φαίνονται τα ονόματα των κόμβων, οι ταχύτητες των μεταξύ τους ζεύξεων και το μέγεθος ουράς σε κάθε ζεύξη. Να αναφέρετε τις εντολές με τις οποίες ορίζονται τα παραπάνω.*

Προκειμένου να εμφανίζεται στην προσομοίωση στο NΑΜ τα ονόματα των κόμβων προσθέτω στο tcl script μου το ακόλουθο τμήμα του κώδικα:

for {set i 0} {$i < 5} {incr i} {

set n($i) [$ns node]

$n($i) label "Νode $i"

}

Με την εντολή $n($i) label "Νode $i" βάζουμε σαν label του κάθε κόμβου τον τίτλο Node ακολουθούμενο από τον αντίστοιχο αριθμό του.

Τώρα, Προκειμένου να εμφανίζεται στην προσομοίωση στο NΑΜ και η ταχύτητα των ζεύξεων μεταξύ των κόμβων καθώς και το μέγεθος της ουράς σε κάθε ζεύξη προσθέτω στο tcl script μου το ακόλουθο τμήμα του κώδικα:

for {set i 0} {$i < 5} {incr i} {

if {$i != 2} {

$ns duplex-link $n($i) $n(2) 2Mb 20ms DropTail

$ns queue-limit $n($i) $n(2) 30

$ns queue-limit $n(2) $n($i) 10

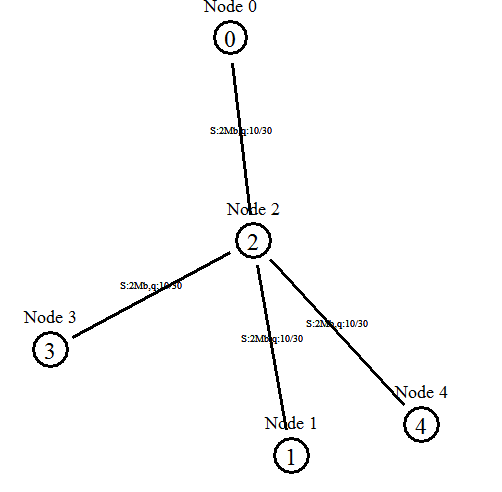
$ns duplex-link-op $n($i) $n(2) label "S:2Mb,q:10/30"

}

}

Με την εντολή $ns duplex-link-op $n($i) $n(2) label "S:2Mb,q:10/30" βάζουμε σαν label της κάθε ζεύξης, την ταχύτητα της σε Mb καθώς και το μέγεθος της ουράς της.

Μετά από την εκτέλεση αυτού του script λαμβάνουμε το ακόλουθο στιγμιότυπο:



1. *Τροποποιήστε τον κώδικα της προσομοίωσης, ώστε η τοπολογία να φαίνεται στο NAM όπως απεικονίζεται στο Σχήμα 1.*

Προσθέτουμε τις παρακάτω εντολές για να έχουμε την τοπολογία που ζητείται:

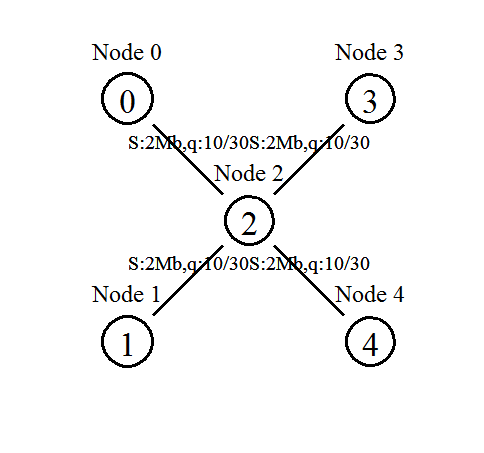
$ns duplex-link-op $n(2) $n(0) orient left-up

$ns duplex-link-op $n(2) $n(1) orient left-down

$ns duplex-link-op $n(2) $n(4) orient right-down

$ns duplex-link-op $n(2) $n(3) orient right-up

Το στιγμιότυπο από το NAM φαίνεται παρακάτω :



1. *Να αναφέρετε ποιες εφαρμογές τρέχουν σε κάθε κόμβο και δημιουργούν δεδομένα προς μετάδοση. Ποιες είναι οι σχετικές εντολές;*

Στον κόμβο 0, τρέχει εφαρμογή που αποστέλλει δεδομένα με σταθερό ρυθμό μετάδοσης CBR. Οι εντολές που μας το κάνουν αυτό φανερό είναι οι ακόλουθες:

set udp1 [new Agent/UDP]

$udp1 set fid\_ 1

$udp1 set packetSize\_ 1500

$ns attach-agent $n(0) $udp1

set null1 [new Agent/Null]

$ns attach-agent $n(3) $null1

$ns connect $udp1 $null1

**set cbr1 [new Application/Traffic/CBR]**

**$cbr1 attach-agent $udp1**

**$cbr1 set packetSize\_ 1500**

**$cbr1 set rate\_ 1.0mb**

**$cbr1 set random\_ off**

(udp1 προσαρτημένος δηλαδή στον κόμβο 0)

Στον κόμβο 0, τρέχει εφαρμογή τύπου FTP. Οι εντολές που το δηλώνουν αυτό είναι οι παρακάτω:

set tcp2 [new Agent/TCP]

$tcp2 set packetSize\_ 1460

$tcp2 set fid\_ 2

$ns attach-agent $n(0) $tcp2

set sink2 [new Agent/TCPSink]

$ns attach-agent $n(4) $sink2

$ns connect $tcp2 $sink2

**set ftp2 [new Application/FTP]**

**$ftp2 attach-agent $tcp2**

(tcp2 προσαρτημένος δηλαδή στον κόμβο 0)

(sink-agent ο κόμβος 4)

Στον κόμβο 1, έχουμε εφαρμογή εκθετικής κίνησης. Οι εντολές είναι οι παρακάτω:

set udp3 [new Agent/UDP]

$udp3 set fid\_ 3

$udp3 set packetSize\_ 1500

$ns attach-agent $n(1) $udp3

**set exp3 [new Application/Traffic/Exponential]**

**$exp3 attach-agent $udp3**

**$exp3 set packetSize\_ 1500**

**$exp3 set rate\_ 1.2mb**

(udp3 προσαρτημένος στον κόμβο 1)

Στον κόμβο 1, έχουμε ακόμη εφαρμογή τύπου Telnet. Οι εντολές είναι οι παρακάτω:

set tcp4 [new Agent/TCP]

$tcp4 set packetSize\_ 960

$tcp4 set fid\_ 4

$ns attach-agent $n(1) $tcp4

set sink4 [new Agent/TCPSink]

$ns attach-agent $n(4) $sink4

$ns connect $tcp4 $sink4

**set telnet4 [new Application/Telnet]**

**$telnet4 attach-agent $tcp4**

**$telnet4 set interval\_ 0.001**

(tcp4 προσαρτημένος στον κόμβο 1)

(sink-agent ο κόμβος 4)

(Με τα bold γράμματα ορίζονται οι εφαρμογές κίνησης που έχουμε σε κάθε περίπτωση. Τα υπόλοιπα μέρη του κώδικα αφορούν ορισμούς των agents σε κάθε περίπτωση)

1. *Τι πρωτόκολλο στο επίπεδο μεταφοράς χρησιμοποιείται για την κίνηση κάθε εφαρμογής;*

Για τις εφαρμογές CBR και Exponential χρησιμοποιείται το πρωτόκολλο UDP, ενώ για τις εφαρμογές FTP και Telnet χρησιμοποιείται το πρωτόκολλο TCP, όπως φαίνεται ξεκάθαρα και στους κώδικες που παρατέθηκαν στο προηγούμενο ερώτημα.

1. *Πού κατευθύνεται η κίνηση κάθε εφαρμογής; Με ποιες εντολές του κώδικα ορίζεται;*

Από τον κόμβο 0 (εφαρμογή κίνησης τύπου CBR και πρωτόκολλο UDP) έχουμε αποστολή δεδομένων στον κόμβο 3 όπου έχουμε καθορίσει να υπάρχει ένας **null agent**. Οι εντολές είναι οι παρακάτω:

set udp1 [new Agent/UDP]

$udp1 set fid\_ 1

$udp1 set packetSize\_ 1500

$ns attach-agent $n(0) $udp1

set null1 [new Agent/Null]

$ns attach-agent $n(3) $null1

$ns connect $udp1 $null1

Από τον κόμβο 0 (εφαρμογή κίνησης τύπου FTP και πρωτόκολλο TCP) έχουμε αποστολή δεδομένων στον κόμβο 4 στον οποίο υπάρχει ένας TCPSink Agent. Οι εντολές είναι οι παρακάτω:

set tcp2 [new Agent/TCP]

$tcp2 set packetSize\_ 1460

$tcp2 set fid\_ 2

$ns attach-agent $n(0) $tcp2

set sink2 [new Agent/TCPSink]

$ns attach-agent $n(4) $sink2

$ns connect $tcp2 $sink2

Από τον κόμβο 1 (εφαρμογή κίνησης τύπου Exponential και πρωτόκολλο UDP) έχουμε αποστολή δεδομένων στον κόμβο 3 στον οποίο υπάρχει ένας null Agent. Οι εντολές είναι οι παρακάτω:

set udp3 [new Agent/UDP]

$udp3 set fid\_ 3

$udp3 set packetSize\_ 1500

$ns attach-agent $n(1) $udp3

set null3 [new Agent/Null]

$ns attach-agent $n(3) $null3

$ns connect $udp3 $null3

Από τον κόμβο 1 (εφαρμογή κίνησης τύπου Telnet και πρωτόκολλο TCP) έχουμε αποστολή δεδομένων στον κόμβο 4 στον οποίο υπάρχει ένας TCPSink Agent. Οι εντολές είναι οι παρακάτω:

set tcp4 [new Agent/TCP]

$tcp4 set packetSize\_ 960

$tcp4 set fid\_ 4

$ns attach-agent $n(1) $tcp4

set sink4 [new Agent/TCPSink]

$ns attach-agent $n(4) $sink4

$ns connect $tcp4 $sink4

1. *Χρωματίστε τις τέσσερις ροές δεδομένων, ώστε να διακρίνονται στο NAM. Παραθέστε ένα σχετικό screenshot.*

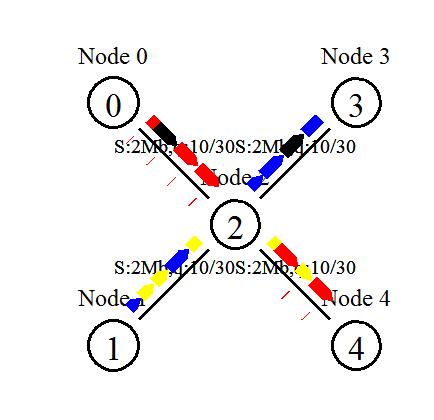
Χρωματίζουμε τις ροές: 0🡪3 μαύρη , 0🡪4 κόκκινη, 1🡪3 μπλε, 1🡪4 κίτρινη με τις παρακάτω εντολές :

$ns color 2 red

$ns color 3 blue

$ns color 4 yellow

Το στιγμιότυπο φαίνεται παρακάτω :

**

1. *Με βάση τα περιεχόμενα του αρχείου ίχνους (lab9.tr) και με τη βοήθεια κατάλληλου script σε γλώσσα awk, υπολογίστε τα ακόλουθα για κάθε μία από τις τέσσερις ροές δεδομένων:*
   1. *το χρόνο μετάδοσης πακέτων,*
   2. *το συνολικό πλήθος των πακέτων και τον όγκο δεδομένων (byte) που αποστέλλονται από*

*τους κόμβους πηγής n(0) και n(1) (εξαιρώντας τις αναμεταδόσεις και τις αποστολές*

*επιβεβαιώσεων),*

*3. το συνολικό πλήθος των πακέτων και τον όγκο δεδομένων που χάνονται,*

*4. το συνολικό πλήθος των πακέτων και τον όγκο δεδομένων που λαμβάνονται από τους*

*κόμβους προορισμού n(3) και n(4) (εξαιρώντας τις λήψεις επιβεβαιώσεων).*

*Σχολιάστε τα αποτελέσματά σας.*

Η εκτέλεση του αρχείου lab9.awk μέσω του cmd μας έδωσε τα εξής αποτελέσματα:

Results for flow 1

Total transfer time: 4,096881 sec

Total packets sent: 334 packets

Total data sent: 501000 bytes

Total packets lost: 0 packets

Total data lost: 0 bytes

Total packets received at node 3: 334 packets

Total data received at node 3: 501000 bytes

Results for flow 2

Total transfer time: 4,382640 sec

Total packets sent: 184 packets

Total data sent: 276000 bytes

Total packets lost: 6 packets

Total data lost: 9000 bytes

Total packets received at node 4: 178 packets

Total data received at node 4: 267000 bytes

Results for flow 3

Total transfer time: 3,850881 sec

Total packets sent: 104 packets

Total data sent: 156000 bytes

Total packets lost: 0 packets

Total data lost: 0 bytes

Total packets received at node 3: 104 packets

Total data received at node 3: 156000 bytes

Results for flow 4

Total transfer time: 21,198080 sec

Total packets sent: 3962 packets

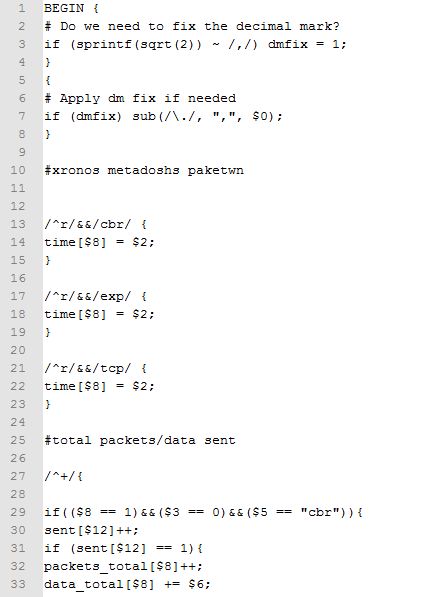
Total data sent: 3962000 bytes

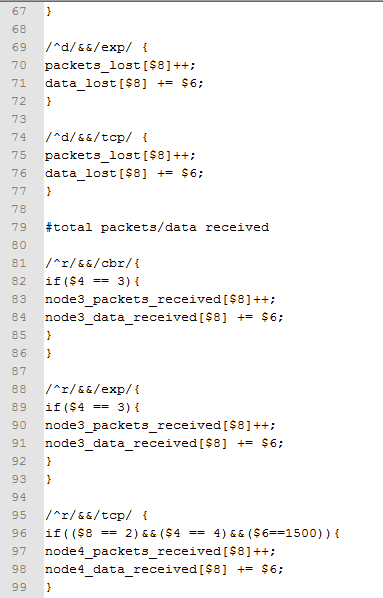
Total packets lost: 3 packets

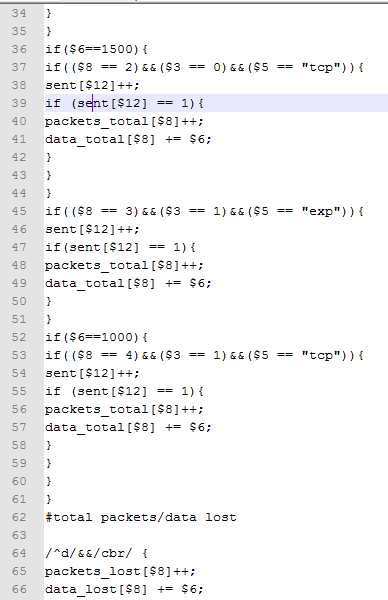
Total data lost: 3000 bytes

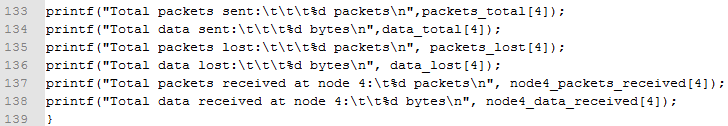
Total packets received at node 4: 3959 packets

Total data received at node 4: 3959000 bytes

Το αρχείο lab9.awk είναι το εξής:







**Σχόλιο** : Χάνονται πακέτα από την ροή 2 και τη ροή 4 με βάση τα αποτελέσματα που μας προκύπτουν. (FTP και TELNET)

1. *Πόσο είναι το ελάχιστο μέγεθος της ουράς αναμονής της ζεύξης μεταξύ των κόμβων n(2) και n(4), ώστε να μην παρατηρούνται απώλειες πακέτων; Περιγράψτε τη μεθοδολογία εντοπισμού του.*

Αν υποθέσουμε ότι το μέγεθος της ουράς αναμονής της ζεύξης μεταξύ των κόμβων n(2) και n(4) έχει την τιμή x. Η εντολή που το περιγράφει αυτό είναι η :

$ns queue-limit $n(2) $n(4) x

Για να βρεθεί το ελάχιστο μέγεθος της ουράς αναμονής αυτής της ζεύξης ώστε να μην έχουμε απώλειες πακέτων ξεκινάμε από το μέγεθος της ουράς που μας δίνεται το οποίο έχει την τιμή 10 και προσθέτουμε κάθε φορά τα πακέτα που χάνονταν, με βάση την ανάλυση του αρχείου ίχνους (trace file) που πραγματοποιήσαμε προηγουμένως. Έπειτα ξανατρέχουμε τον κώδικα μέχρι να είναι μηδενικές οι συνολικές απώλειες. Έτσι υπολογίζουμε το ελάχιστο μέγεθος της ουράς. Στην περίπτωσή μας αυτό είναι ίσο με **21**.

1. *Τροποποιήστε τον κώδικα προσομοίωσης του ερωτήματος (ζ), ώστε να δημιουργήσετε μια αμφίδρομη ζεύξη μεταξύ των κόμβων n(1) και n(4) με τα ακόλουθα χαρακτηριστικά: (i) εύρος ζώνης 1.2Mbps, (ii) ουρά τύπου DropTail, (iii) καθυστέρηση 50ms και (iv) μέγεθος ουράς αναμονής ίσο με 30. Τι παρατηρείτε ως προς τις διαδρομές που ακολουθούν τώρα οι ροές δεδομένων σε σχέση με τις διαδρομές που ακολουθούνται στην αρχική τοπολογία; Παραθέστε ένα σχετικό screenshot. Υπολογίστε το πλήθος των πακέτων που χάνονται σε κάθε κόμβο για κάθε μία από τις ροές δεδομένων.*

Στον ορισμό των ζεύξεων προσθέτουμε τις παρακάτω εντολές ώστε να οριστούν όσα αναφέρονται στην εκφώνηση στην προσομοίωσή μας.

$ns duplex-link $n(1) $n(4) 1.2Mb 50ms DropTail

$ns queue-limit $n(1) $n(4) 30

$ns queue-limit $n(4) $n(1) 30

Για να γίνει σωστός ο προσανατολισμός στο ΝΑΜ προσθέτουμε ακόμη την εντολή:

$ns duplex-link-op $n(1) $n(4) orient left-right

Οι διαδρομές που ακολουθούνται είναι οι παρακάτω:

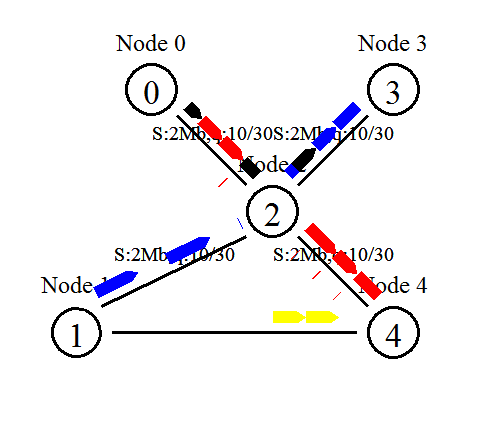
Ροή 1(μαύρη): 0🡪2🡪3

Ροή 2(κόκκινη): 0🡪2🡪4

Ροή 3(μπλε): 1🡪2🡪3

Ροή 4(κίτρινη): 1🡪4

Ένα ενδεικτικό στιγμιότυπο είναι το παρακάτω:



**Σχόλιο** : Για την τέταρτη ροή τα δεδομένα στέλνονται απευθείας από τον κόμβο 1 στον κόμβο 4 ενώ πριν ακολουθούσαν τη διαδρομή **1->2->4.**

Τρέχουμε ξανά το awk script και υπολογίζουμε τα πακέτα που χάνονται για κάθε ροή και πήραμε τα παρακάτω αποτελέσματα:

Results for flow 1

Total packets lost: 0 packets

Results for flow 2

Total packets lost: 0 packets

Results for flow 3

Total packets lost: 1 packets

Packet lost at node: 2

Results for flow 4

Total packets lost: 0 packets

**Σχόλιο:** Χάνεται όπως είναι φανερό ένα πακέτο της ροής 3 στον κόμβο 2. Ο εντοπισμός έγινε προσθέτοντας στον awk κώδικα ένα επιπλέον έλεγχο ο οποίος υλοποιείται με τις ακόλουθες εντολές:

/^d/&&/exp/ {

packets\_lost[$8]++;

data\_lost[$8] += $6;

node\_lost=$3;

}

1. *Τροποποιήστε τον κώδικα του προηγούμενου ερωτήματος ώστε οι διαδρομές που ακολουθούν οι τέσσερις ροές δεδομένων στη νέα τοπολογία να είναι ίδιες με τις αρχικές. Χρησιμοποιήστε την έννοια του κόστους ζεύξης για να το επιτύχετε αυτό, θέτοντας το ανάλογο της καθυστέρησης της εκάστοτε ζεύξης (π.χ. 10 msec -> 1 μονάδα κόστους). Προσοχή: Σε περίπτωση ροής TCP, θα πρέπει τόσο τα πακέτα δεδομένων όσο και οι επιβεβαιώσεις τους να κινούνται στην ίδια διαδρομή (προφανώς με αντίθετες φορές). Παραθέστε ένα σχετικό screenshot.*

Ορίζουμε το κόστος στο βρόχο δημιουργίας τον ζεύξεων και στο τμήμα του κώδικα που ορίζεται η δημιουργία της ζεύξης 1🡪4 με τις εντολές:

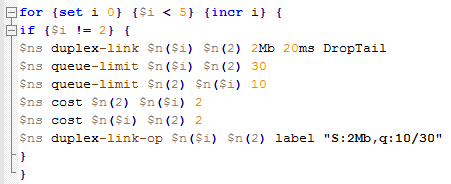
* $ns cost $n(2) $n($i) 2

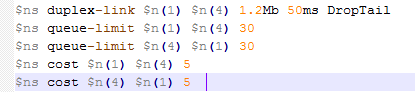
$ns cost $n($i) $n(2) 2 (ανάλογο της καθυστέρησης που είναι 20 ms)

* $ns cost $n(1) $n(4) 5

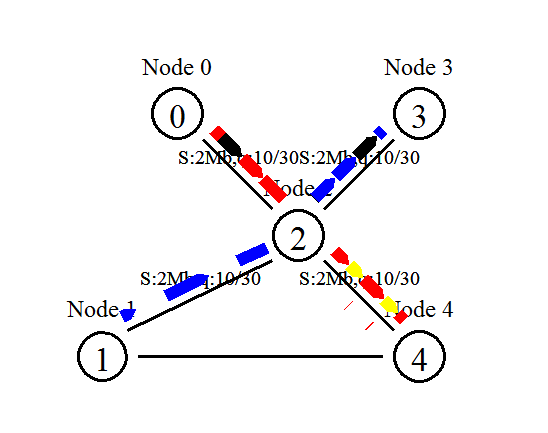
$ns cost $n(4) $n(1) 5 (ανάλογο της καθυστέρησης που είναι 50 ms)

Δηλαδή τμηματικά το tcl script τροποποιείται όπως φαίνεται παρακάτω:





Ακολουθούνται οι αρχικές διαδρομές καθώς η διαδρομή 1🡪4 απέκτησε μεγαλύτερο κόστος από την αρχική διαδρομή που είναι η 1🡪2🡪4 (5 έναντι 4 μονάδων κόστους) ενώ οι άλλες διαδρομές παρέμειναν αμετάβλητες. Το αντίστοιχο screenshot φαίνεται στη συνέχεια :



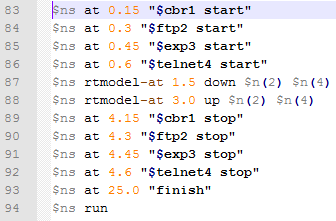
1. *Τροποποιήστε τον κώδικα του προηγούμενου ερωτήματος ώστε να συμβαίνει διακοπή της ζεύξης μεταξύ των κόμβων n(2) και n(4) μεταξύ των χρονικών στιγμών 1.5 και 3.0 sec. Πόσα πακέτα χάνονται στην περίπτωση αυτή για κάθε μία από τις ροές δεδομένων; Σε ποια χρονική στιγμή μετά την επαναφορά της ζεύξης συνεχίζουν οι ροές δεδομένων από τους κόμβους n(0) και n(1) στον n(4);*

Στον κώδικά μας προσθέτουμε τις ακόλουθες εντολές που ορίζουν το συγκεκριμένο γεγονός:

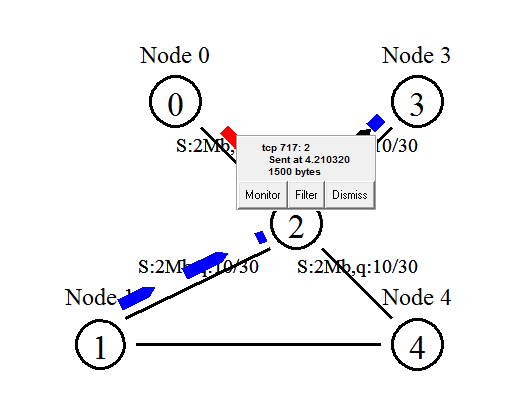
$ns rtmodel-at 1.5 down $n(2) $n(4)

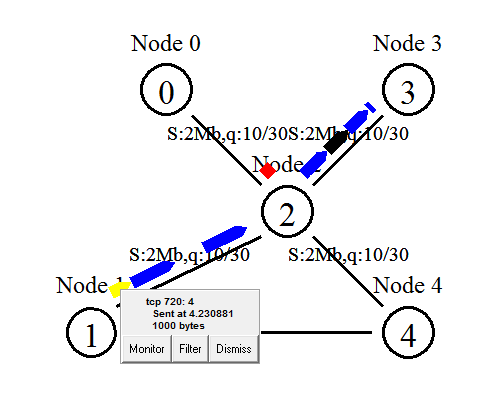
$ns rtmodel-at 3.0 up $n(2) $n(4)

Άρα το τμήμα ορισμού των γεγονότων γίνεται το ακόλουθο:



Από τον κόμβο n(0) στον κόμβο n(4) το πρώτο πακέτο μετά την επαναφορά της ζεύξης των κόμβων n(2) και n(4) στέλνεται την χρονική στιγμή t=4,210320 sec ενώ από τον κόμβο n(1) στον κόμβο n(4) την χρονική στιγμή t=4,230881 sec. Τα αντίστοιχα στιγμιότυπα προς απόδειξη των παραπάνω είναι τα ακόλουθα:





Τα πακέτα που χάνονται σε αυτήν την περίπτωση υπολογίζονται μέσω του awk script από την εκτέλεση του οποίου προκύπτει:

Results for flow 1

Total packets lost: 0 packets

Results for flow 2

Total packets lost: 4 packets

Results for flow 3

Total packets lost: 0 packets

Results for flow 4

Total packets lost: 2 packets

Άρα έχουμε απώλειες 4 πακέτων στη ροή 2 και 2 πακέτων στη ροή 4. Οι ροές 1 και 3 δεν έχουν καμία απώλεια πακέτων.

1. *Τροποποιήστε τον κώδικα της προσομοίωσης του ερωτήματος ΧΙ, ώστε οι ροές από τους κόμβους n(0) και n(1) προς τον n(4) να συνεχίζουν να δρομολογούνται και μετά τη διακοπή της ζεύξης n(2)-n(4) μέσω εναλλακτικής διαδρομής. Πόσα πακέτα χάνονται στην περίπτωση αυτή για κάθε μία από τις ροές δεδομένων; Σε ποια χρονική στιγμή μετά τη διακοπή της ζεύξης συνεχίζουν οι ροές δεδομένων από τους κόμβους n(0) και n(1) στον n(4) και μέσω ποιας διαδρομής; Σε ποια χρονική στιγμή μετά την επαναφορά της ζεύξης συνεχίζουν οι ροές δεδομένων από τους κόμβους n(0) και n(1) στον n(4) μέσω της n(2)-n(4);*

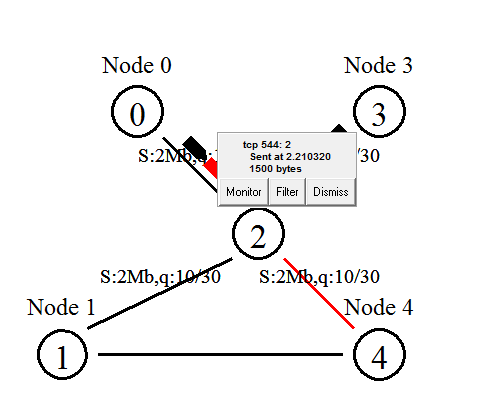
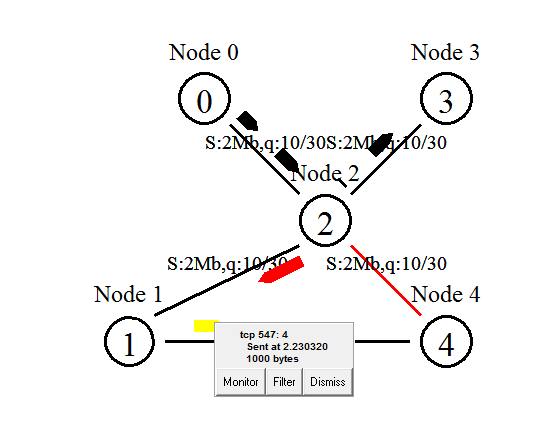
Ορίζουμε στην τοπολογία τη δυνατότητα δυναμικής δρομολόγησης. Προκειμένου να το πετύχουμε αυτό προσθέτουμε στον κώδικά μας τις εντολές :

Agent/rtProto/Direct set preference\_ 200

$ns rtproto DV

Μετά τη διακοπή της ζεύξης 2🡪4 και μέχρι να επανέλθει σε λειτουργία ακολουθείται η διαδρομή 0🡪2🡪1🡪4 για την ροή 2 και το πρώτο πακέτο στέλνεται τη χρονική στιγμή t= 2,21032 sec ενώ για την ροή 4 η διαδρομή που ακολουθείται είναι η 1🡪4 και το πρώτο πακέτο στέλνεται τη χρονική στιγμή t= 2,23032 sec. Οι υπόλοιπες ροές παραμένουν αμετάβλητες.

Για του λόγου το αληθές:



Τρέχουμε τον κώδικα σε γλώσσα awk για να δούμε και πάλι τα πακέτα που χάνονται σε αυτήν την περίπτωση και παίρνουμε τα ακόλουθα αποτελέσματα:

Results for flow 1

Total packets lost: 0 packets

Results for flow 2

Total packets lost: 5 packets

Results for flow 3

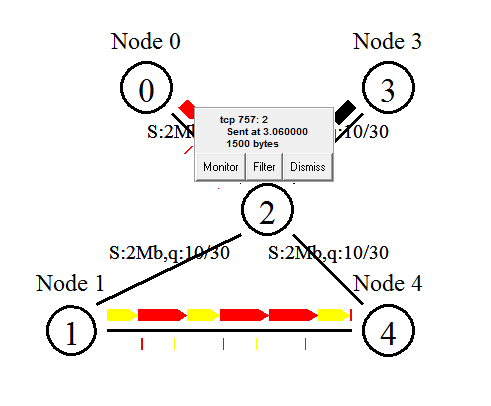
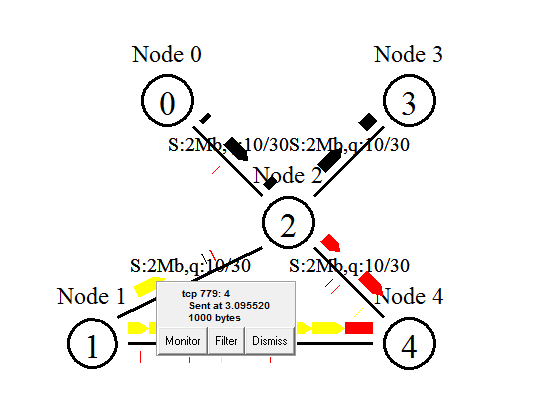
Total packets lost: 0 packets

Results for flow 4

Total packets lost: 2 packets

Άρα έχουμε απώλειες 5 πακέτων στη ροή 2 και 2 πακέτων στη ροή 4. Οι ροές 1 και 3 δεν έχουν καμία απώλεια πακέτων.

Μετά την επαναφορά της ζεύξης n(2)🡪n(4) το πρώτο πακέτο της ροής 2 (κόκκινη) , στέλνεται από τον κόμβο n(0) μέσω της ζεύξης n(2)🡪n(4) ακολουθώντας τη διαδρομή 0🡪2🡪4 τη χρονική στιγμή: t= 3,06 sec και το πρώτο πακέτο της ροής 4 (κίτρινη) ,αποστέλλεται από τον κόμβο n(1) μέσω της ζεύξης n(2)🡪n(4) μέσω της διαδρομής 1🡪2🡪4 τη χρονική στιγμή: t= 3,09552 sec. Αυτά φαίνονται στα ακόλουθα στιγμιότυπα:



1. *Τροποποιήστε τον αρχικό κώδικα, ώστε να μετατρέψετε την ενσύρματη τοπολογία του δικτύου σε ασύρματη, χρησιμοποιώντας το πρότυπο IEEE 802.11. Επίσης, αντικαταστήστε την εκθετική κίνηση (Exponential) με κίνηση CBR πακέτων μεγέθους 1500 byte και ρυθμού 1.2 Mbps. Χρησιμοποιήστε επίπεδο πλέγμα μήκους 600 m και πλάτους 400 m και τοποθετήστε τον κόμβο n(0) στο κέντρο του, ενώ τους υπόλοιπους κόμβους σε κορυφές τετραγώνου, οι οποίες να απέχουν από τον n(0) απόσταση ίση με 200 m. Χρησιμοποιήστε τις τιμές παραμέτρων που ορίζονται ακολούθως:*

*set opt(chan) Channel/WirelessChannel*

*set opt(prop) Propagation/TwoRayGround*

*set opt(ant) Antenna/OmniAntenna*

*set opt(ll) LL*

*set opt(ifq) Queue/DropTail/PriQueue*

*set opt(ifqlen) 30*

*set opt(netif) Phy/WirelessPhy*

*set opt(mac) Mac/802\_11*

*set opt(rp) AODV*

*set opt(nn) 5*

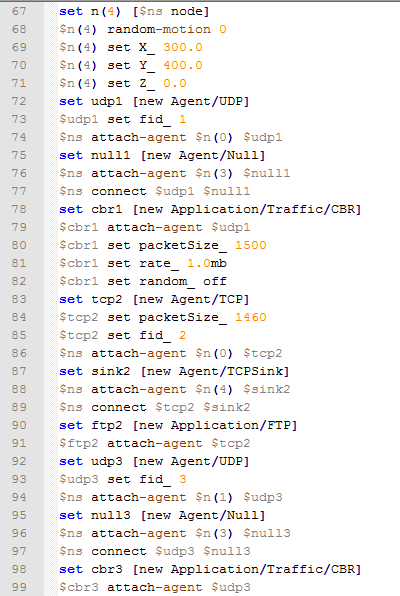
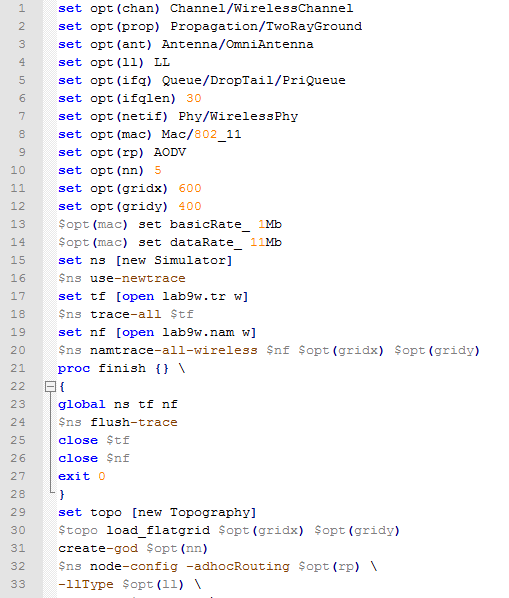
*set opt(gridx) 600*

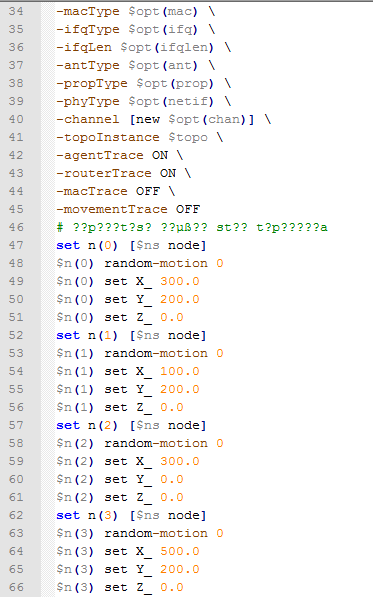
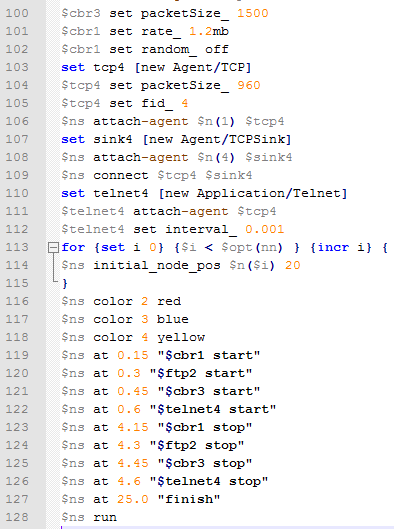
*set opt(gridy) 400*

*$opt(mac) set basicRate\_ 1Mb*

*$opt(mac) set dataRate\_ 11Mb*

Για να μετατρέψουμε την ενσύρματη τοπολογία του δικτύου σε ασύρματη, χρησιμοποιώντας το πρότυπο IEEE 802.11 τροποποιήσαμε τον κώδικα όπως φαίνεται παρακάτω:





1. *Με τη βοήθεια κατάλληλου script σε γλώσσα awk, υπολογίστε τα ακόλουθα για κάθε μία από τις τέσσερις ροές δεδομένων: 1. το χρόνο μετάδοσης πακέτων, 2. το συνολικό πλήθος των πακέτων και τον όγκο δεδομένων (byte) που αποστέλλονται από τους κόμβους πηγής n(0) και n(1) (εξαιρώντας τις αναμεταδόσεις και τις αποστολές επιβεβαιώσεων), 3. το συνολικό πλήθος των πακέτων και τον όγκο δεδομένων που χάνονται, 4. το συνολικό πλήθος των πακέτων και τον όγκο δεδομένων που λαμβάνονται από τους κόμβους προορισμού n(3) και n(4) (εξαιρώντας τις λήψεις επιβεβαιώσεων). Σχολιάστε τα αποτελέσματά σας.*

Δημιουργώντας παρόμοιο script με εκείνο που χρησιμοποιήσαμε στα προηγούμενα ερωτήματα λαμβάνουμε τα ακόλουθα αποτελέσματα για την προσομοίωσή μας:

Results for flow 1  
Total transfer time: 4,005710 sec  
Total packets sent: 334 packets  
Total data sent: 501000 bytes  
Total packets lost: 4 packets  
Total data lost: 6080 bytes  
Total packets received at node 3: 330 packets  
Total data received at node 3: 494920 bytes

Results for flow 2  
Total transfer time: 4,390680 sec  
Total packets sent: 207 packets  
Total data sent: 310500 bytes  
Total packets lost: 7 packets   
Total data lost: 10640 bytes  
Total packets received at node 3: 200 packets  
Total data received at node 3: 299860 bytes

Results for flow 3  
Total transfer time: 4,087510 sec  
Total packets sent: 401 packets  
Total data sent: 601500 bytes  
Total packets lost: 51 packets   
Total data lost: 77520 bytes  
Total packets received at node 4: 350 packets  
Total data received at node 4: 523980 bytes

Results for flow 4  
Total transfer time: 24,398600 sec  
Total packets sent: 2907 packets  
Total data sent: 2907000 bytes  
Total packets lost: 14 packets  
Total data lost: 14280 bytes  
Total packets received at node 4: 2893 packets  
Total data received at node 4: 2892720 bytes

1. *Με βάση την απάντησή σας στο προηγούμενο ερώτημα, έχουν ολοκληρωθεί όλες οι ροές*

*δεδομένων; Αν όχι, πόσο πρέπει να αυξηθεί ο χρόνος προσομοίωσης ώστε να ολοκληρωθούν και οι τέσσερις ροές;*

Για να είμαστε σίγουροι για τα αποτελέσματα μας αυξάνουμε τη διάρκεια παραγωγής δεδομένων κάθε ροής στα 20sec και τη διάρκεια της προσομοίωσης στα 40 sec. Με αυτές τις τροποποιήσεις λαμβάνουμε από το awk script τα ακόλουθα αποτελέσματα για την προσομοίωσή μας:   
  
Results for flow 1  
Total transfer time: 19,996200 sec  
Total packets sent: 1667 packets  
Total data sent: 2500500 bytes  
Total packets lost: 26 packets   
Total data lost: 39520 bytes   
Total packets received at node 3: 1641 packets  
Total data received at node 3: 2460980 bytes

Results for flow 2  
Total transfer time: 20,507700 sec  
Total packets sent: 1087 packets  
Total data sent: 1630500 bytes  
Total packets lost: 24 packets  
Total data lost: 36480 bytes  
Total packets received at node 3: 1063 packets  
Total data received at node 3: 1594020 bytes

Results for flow 3  
Total transfer time: 20,053700 sec  
Total packets sent: 2000 packets  
Total data sent: 3000000 bytes  
Total packets lost: 238 packets  
Total data lost: 361760 bytes  
Total packets received at node 4: 1762 packets  
Total data received at node 4: 2638240 bytes

Results for flow 4  
Total transfer time: 44,398500 sec  
Total packets sent: 3679 packets  
Total data sent: 3679000 bytes  
Total packets lost: 41 packets  
Total data lost: 41820 bytes  
Total packets received at node 4: 3638 packets  
Total data received at node 4: 3637180 bytes