

Δίκτυα Επικοινωνιών

2014-2015

7η εργαστηριακή άσκηση:

Δρομολόγηση Διανύσματος Αποστάσεων

Εξάμηνο : 6ο

Ονοματεπώνυμο : Δασούλας Γεώργιος

Α.Μ. :03112010

Επαλήθευση πρωτοκόλλου DV:

Στη συγκεκριμένη άσκηση μελετάται η επίδοση του πρωτοκόλλου DV. Στο πρώτο μέρος της άσκησης δημιουργούμε μια σύνθετη τοπολογία για την ανάδειξη της λειτουργίας του πρωτοκόλλου.

Για τη δημιουργία της ζητούμενης τοπολογίας , δημιουργούμε και χρησιμοποιούμε το παρακάτω tcl script:

```
set ns [new Simulator]
Agent/rtpProto/Direct set preference_ 200
$ns rtproto DV
set nf [open lab7a.nam w]
$ns namtrace-all $nf
set xf [open lab7a.tr w]
$ns trace-all $xf
for {set i 0} {$i < 8} {incr i} {
    set n($i) [$ns node]
}
for {set i 0} {$i<8} {incr i} {
    $ns duplex-link $n($i) $n([expr ($i+1)%8]) 1Mb 20ms DropTail
    $ns cost $n($i) $n([expr ($i+1)%8]) 2
    $ns cost $n([expr ($i+1)%8]) $n($i) 2
}
$ns duplex-link $n(7) $n(1) 1Mb 30ms DropTail
$ns cost $n(7) $n(1) 3
$ns cost $n(1) $n(7) 3
$ns duplex-link $n(7) $n(6) 1Mb 30ms DropTail
$ns cost $n(7) $n(6) 3
$ns cost $n(6) $n(7) 3
$ns duplex-link $n(5) $n(7) 1Mb 50ms DropTail
```

```
$ns cost $n(5) $n(7) 5
$ns cost $n(7) $n(5) 5
$ns duplex-link $n(6) $n(5) 1Mb 30ms DropTail
$ns cost $n(5) $n(6) 3
$ns cost $n(6) $n(5) 3
$ns duplex-link $n(3) $n(1) 1Mb 10ms DropTail
$ns cost $n(3) $n(1) 1
$ns cost $n(1) $n(3) 1
$ns duplex-link $n(5) $n(3) 1Mb 30ms DropTail
$ns cost $n(5) $n(3) 3
$ns cost $n(3) $n(5) 3
```

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

```
set tcp1 [new Agent/TCP]
$tcp1 set packetSize_ 500
$ns attach-agent $n(0) $tcp1
$tcp1 set fid_ 0
$ns color 0 blue
set sink1 [new Agent/TCPSink]
$ns attach-agent $n(5) $sink1
$ns connect $tcp1 $sink1
```

```
set tcp2 [new Agent/TCP]
$tcp2 set packetSize_ 500
$ns attach-agent $n(4) $tcp2
$tcp2 set fid_ 1
$ns color 3 red
set sink2 [new Agent/TCPSink]
$ns attach-agent $n(7) $sink2
$ns connect $tcp2 $sink2
```

```
set ftp1 [new Application/FTP]
$ftp1 attach-agent $tcp1
```

```

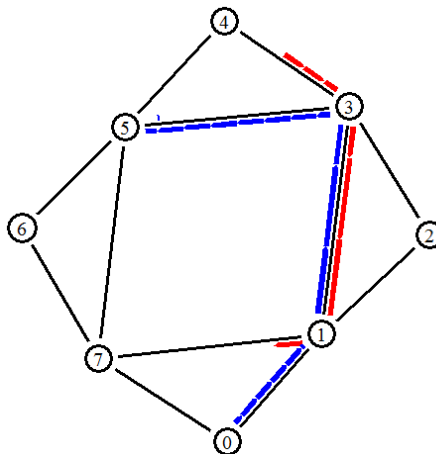
set ftp2 [new Application/FTP]
$ftp2 attach-agent $tcp2
$ns at 0.5 "$ftp1 produce 120"
$ns at 0.8 "$ftp2 produce 120"
$ns at 3 "finish"
$ns run

```

Ερωτήσεις -Απαντήσεις:

- Ποιες διαδρομές ακολουθούν τα πακέτα δεδομένων για τις δύο μεταδόσεις; Ποιες διαδρομές ακολουθούν τα πακέτα επιβεβαιώσεων για τις δύο μεταδόσεις; Παραθέστε τον ολοκληρωμένο κώδικα προσομοίωσης.

Απάντηση : Για την πρώτη ροή (μπλε) τα πακέτα δεδομένων ακολουθούν την εξής διαδρομή: **0 - 1 - 3 - 5** , η οποία διαδρομή έχει συνολική καθυστέρηση : $20+10+30=60$ ms . Τα πακέτα επιβεβαιώσεων ακολουθούν την εξής διαδρομή : **5 - 3 - 1 - 0**.
Για τη δεύτερη ροή(κόκκινη) τα πακέτα δεδομένων ακολουθούν την εξής διαδρομή : **4 - 3 - 1 - 7**, η οποία έχει συνολική καθυστέρηση : $20+10+30=60$ ms . Τα πακέτα επιβεβαιώσεων ακολουθούν την ανάποδη διαδρομή , δηλαδή την : **7 - 1 - 3 - 4** .
Ο κώδικας έχει παρατεθεί πιο πάνω. Ακολουθεί μια εικόνα από την προσομοίωση στο nam.



- Να εκτελέσετε με το χέρι τον αλγόριθμο διανύσματος αποστάσεων και να επαληθεύσετε την ορθή λειτουργία του πρωτοκόλλου που οδηγεί στην δρομολόγηση μέσω των διαδρομών ελαχίστου κόστους.

Απάντηση : Ο αλγόριθμος διανύσματος αποστάσεων ορίζει το παρακάτω :
Αν ο οδηγός-κόμβος Α έχει ενημερωθεί ότι το κόστος προς τον κόμβο Β είναι $f(A,B)$ μονάδες κόστους , τότε ενημερώνεται άμεσα ότι το κόστος προς τον οδηγό-κόμβο Γ είναι $f(A,B) + f(B,\Gamma)$ μονάδες κόστους. Θα δοκιμάσουμε τη διαδρομή **0 - 1 - 3 - 5**. Άρα, παίρνουμε ως αρχικό κόμβο τον 0 ξεκινώντας , λοιπόν, τη διαδικασία βήμα βήμα δημιουργούμε τα παρακάτω στοιχεία :
Για το πρώτο βήμα της δρομολόγησης :

κόμβοι-προορισμοί	κόστος από τον αρχικό	γείτονας του αρχικού 0
1	2	1
2	4	1
3	3	1
4	5	1
5	6	1
6	5	7
7	2	7

Συνεπώς, βλέπουμε ότι για το πρώτο βήμα της δρομολόγησης , προτιμάμε τον κόμβο 1. Συνεχίζοντας τη διαδικασία του αλγορίθμου βλέπουμε ότι προκύπτει η διαδρομή 0 - 1 - 3 - 5 , όπως προέκυψε στο NAM.

- Χρησιμοποιώντας κατάλληλο script σε γλώσσα awk, βάσει του αρχείου ίχνους της προσομοίωσης, να προσδιοριστεί το πλήθος των πακέτων TCP που λαμβάνονται από τους κόμβους προορισμού για κάθε μία από τις δύο ροές. Ολοκληρώνεται η μετάδοση κάθε ροής κατά τη διάρκεια της προσομοίωσης; Εάν ναι, καταγράψτε τις χρονικές στιγμές ολοκλήρωσης των δύο μεταδόσεων. Παραθέστε τον κώδικα awk που χρησιμοποιήσατε.

Απάντηση : Αφού δημιουργήσαμε το κατάλληλο script το οποίο παραθέτουμε παρακάτω , προέκυψαν τα παρακάτω αποτελέσματα :

```
received packets from flow 1: 120
received packets from flow 2: 120
finishing time for packet transfer from flow 1: 1.816240 sec
finishing time for packet transfer from flow 2: 2.110880 sec
```

Από τα δεδομένα φαίνεται πως η μετάδοση της κάθε ροής ολοκληρώθηκε επιτυχώς μέσα στο χρόνο της προσομοίωσης. Το awk αρχείο που δημιουργήσαμε είναι το παρακάτω :

```
BEGIN {
    revpacks1=0;
    revpacks2=0;
    time1=0.0;
    time2=0.0;
    if (sprintf(sqrt(2)) ~ /,/ ) dmfix = 1;
}
{
    if (dmfix) sub(/\. /, ", ", $0);
}
/^r/&&/tcp/ {
    if ($6 == 540){
        if ($4 == 5 && $8 == 0){
            revpacks1++;
        }
    }
}
```

```

        if (revpacks1 == 120) time1=$2;
    }
    if ($4 == 7 && $8 == 1){
        revpacks2++;
        if (revpacks2 == 120) time2=$2;
    }
}
}
END{
    printf("received packets from flow 1: %d\n",revpacks1);
    printf("received packets from flow 2: %d\n",revpacks2);
    printf("finishing time for packet transfer from flow 1: %f sec\n",time1);
    printf("finishing time for packet transfer from flow 2: %f sec\n",time2);
}

```

- Ποιο είναι το μέγιστο κόστος μετάδοσης της ζεύξης 1-3, ώστε να διατηρούνται βέλτιστες οι διαδρομές που προσδιορίσατε στο προηγούμενο ερώτημα για τις δύο μεταδόσεις; Είναι οι μόνες συντομότερες διαθέσιμες διαδρομές σε αυτήν την περίπτωση; Επαληθεύστε την απάντησή σας τόσο θεωρητικά (εκτέλεση αλγορίθμου DV με το χέρι), όσο και πειραματικά με τη βοήθεια του NS2, υποδεικνύοντας όλες τις διαδρομές ελάχιστου κόστους για κάθε μετάδοση.

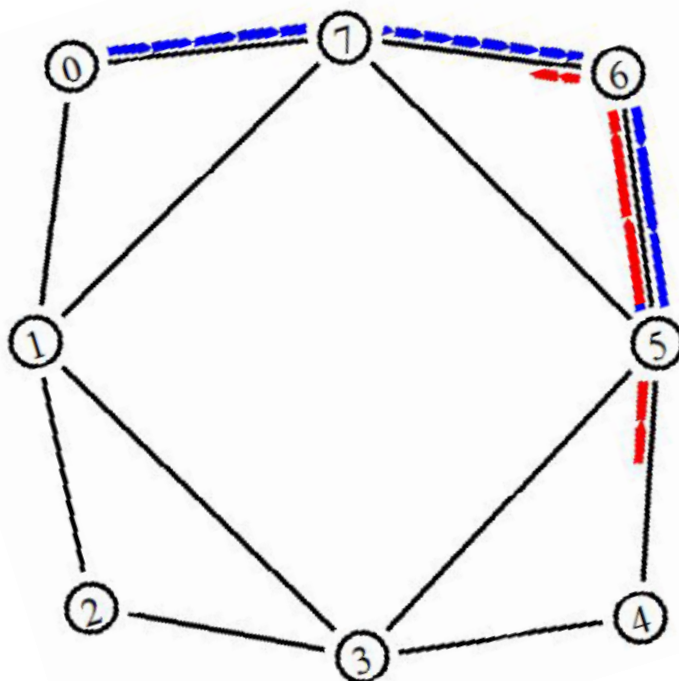
Απάντηση: Από την τοπολογία βλέπουμε πως μετά τη διαδρομή **0 - 1 - 3 - 5** , η αμέσως επόμενη λιγότερο κοστοβόρα έχει καθυστέρηση 70 ms και είναι η **0 - 7 - 5** , η οποία δεν περιέχει τη ζεύξη 1 - 3 . Συνεπώς , θα πρέπει να είναι τέτοιο το μέγιστο κόστος της 1-3 , ώστε να μην ξεπερνάει η συνολική καθυστέρηση $(20+k+30)$ τα 70 ms. Άρα, το μέγιστο κόστος της 1 -3 είναι 2 (δηλαδή 20 ms καθυστέρηση), αν θέλουμε να διατηρείται η διαδρομή **0 - 1 - 3 - 5** ως βέλτιστη . Ομοίως, για τη διαδρομή **4 - 3 - 1 - 7** , η αμέσως επόμενη λιγότερο κοστοβόρα έχει καθυστέρηση 70 ms και είναι η **4 - 5 - 7** , που κι αυτή δεν περιέχει τη ζεύξη 1 - 3 . Άρα, τελικά το μέγιστος κόστος μετάδοσης της ζεύξης είναι 2. Θεωρητικά , εκτελώντας με το χέρι τον αλγόριθμο και χρησιμοποιώντας τον ορισμό φτιάχνουμε τον παρακάτω πίνακα: (θα εκτελέσουμε μόνο για τη διαδρομή με αρχικό τον 0)

κόμβοι - προορισμοί	κόστος από τον αρχικό	γείτονας του αρχικού 0
1	2	1
2	4	1
3	4	1
4	6	1
5	7	επιλογή ανάμεσα σε 1 και 7
6	5	7
7	2	7

Βλέπουμε , συνεπώς , ότι για το πρώτο βήμα της δρομολόγησης προέκυψε πως μετά το 0 μπορούμε να πάμε στο 1 ή στο 7 (για κόστος 2 στην ζεύξη 1 -3) . Αν συνεχίσουμε τη διαδικασία για όλα τα βήματα παρατηρούμε πως επιβεβαιώνεται η επιλογή του μέγιστου κόστους. Ομοίως, για την περίπτωση της διαδρομής με αρχικό κόμβο τον 4.

- Θεωρώντας το κόστος της ζεύξης 1-3 ίσο με την τιμή που προσδιορίστηκε στο τρίτο ερώτημα, να υπολογιστεί η ελάχιστη μεταβολή τους κόστους μετάδοσης της ζεύξης 5-6, η οποία θα οδηγήσει στη συμμετοχή της ζεύξης αυτής στη μία εκ των παραπάνω δύο βέλτιστων διαδρομών για τις δύο μεταδόσεις και άρα στην τροποποίησή της; Επαληθεύστε την απάντησή σας πειραματικά με τη βοήθεια του NS2 και παραθέστε τα σχετικά στιγμιότυπα από το NAM.

Απάντηση : Θεωρούμε το κόστος της ζεύξης ίσο με 2. Άρα, έχουμε ελάχιστο κόστος διαδρομής ίσο με 70 ms. Για να έχουμε τη ζεύξη 5-6 στη βέλτιστη διαδρομή, θα πρέπει το κόστος της 0 - 7 - 6 - 5 να μην ξεπερνάει τα 70ms, κάτι που δείχνει ότι αρκεί η 5 - 6 να έχει κόστος 2. Ομοίως, για τη διαδρομή 4 - 5 - 6 - 7 έχουμε καθυστέρηση για κόστος της 5-6 2. Συνεπώς , η ελάχιστη μεταβολή του κόστους της 5 - 6 είναι να μειωθεί από 3 σε 2. Αυτό μας προέκυψε και πειραματικά , μέσα από την προσομοίωση στο NAM.



Αποτυχία πρωτοκόλλου Distance Vector:

Για το δεύτερο κομμάτι δημιουργούμε το παρακάτω αρχείο tcl για την κατάλληλη τοπολογία :

```
set ns [new Simulator]
Agent/rtpProto/Direct set preference_ 200
$ns rtpProto DV
set nf [open lab7b.nam w]
$ns namtrace-all $nf
set xf1 [open lab7b1.tr w]
set xf2 [open lab7b2.tr w]
$ns trace-all $xf1
$ns trace-all $xf2
$ns color 0 blue
$ns color 1 red

for {set i 0} {$i < 4} {incr i} {
    set n($i) [$ns node]
}

for {set i 1} {$i<3} {incr i} {
    $ns duplex-link $n($i) $n([expr ($i+1)]) 1.2Mb 10ms DropTail
}
$ns duplex-link $n(3) $n(1) 1.2Mb 10ms DropTail
$ns duplex-link $n(0) $n(1) 1.2Mb 10ms DropTail

proc record {} {
    global sink0 sink1 xf1 xf2
    set ns [Simulator instance]
    set time 0.2
    set bw0 [$sink0 set bytes_]
    set bw1 [$sink1 set bytes_]
    set now [$ns now]
    puts $xf1 "$now [expr $bw0/$time*8/1000000]"
    puts $xf2 "$now [expr $bw1/$time*8/1000000]"
    $sink0 set bytes_ 0
    $sink1 set bytes_ 0
    $ns at [expr $now+$time] "record"
}

proc finish {} {
    global ns nf xf1 xf2
    $ns flush-trace
    close $nf
    close $xf1
    close $xf2
}
```

```

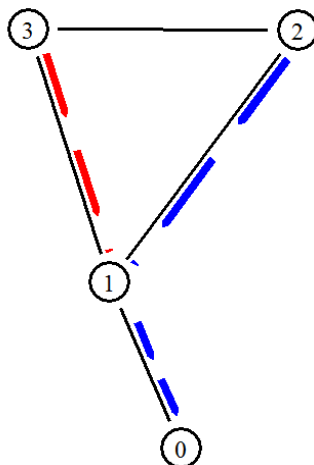
    exit 0
}

set udp2 [new Agent/UDP]
$udp2 set packetSize_ 500
$ns attach-agent $n(2) $udp2
$udp2 set fid_ 0
set sink0 [new Agent/LossMonitor]
$ns attach-agent $n(0) $sink0
$ns connect $udp2 $sink0
set udp3 [new Agent/UDP]
$udp3 set packetSize_ 500
$ns attach-agent $n(3) $udp3
$udp3 set fid_ 1
set sink1 [new Agent/LossMonitor]
$ns attach-agent $n(1) $sink1
$ns connect $udp3 $sink1
set cbr2 [new Application/Traffic/CBR]
$cbr2 set packetSize_ 500
$cbr2 set interval_ 0.005
$cbr2 attach-agent $udp2
set cbr3 [new Application/Traffic/CBR]
$cbr3 set packetSize_ 500
$cbr3 set interval_ 0.005
$cbr3 attach-agent $udp3

$ns at 0.0 "record"
$ns at 0.25 "$cbr2 start"
$ns at 0.25 "$cbr3 start"
$ns rtmodel-at 1 down $n(0) $n(1)
$ns rtmodel-at 1.5 up $n(0) $n(1)
$ns at 3 "finish"
$ns at 3.5 "finish"
$ns run

```

Προέκυψε η παρακάτω τοπολογία :



Ερωτήσεις -Απαντήσεις:

- Ποια διαδρομή ακολουθούν τα πακέτα που στέλνει ο κόμβος 2 πριν και μετά τη διακοπή της ζεύξης 0-1; Εξηγήστε αυτή τη συμπεριφορά.

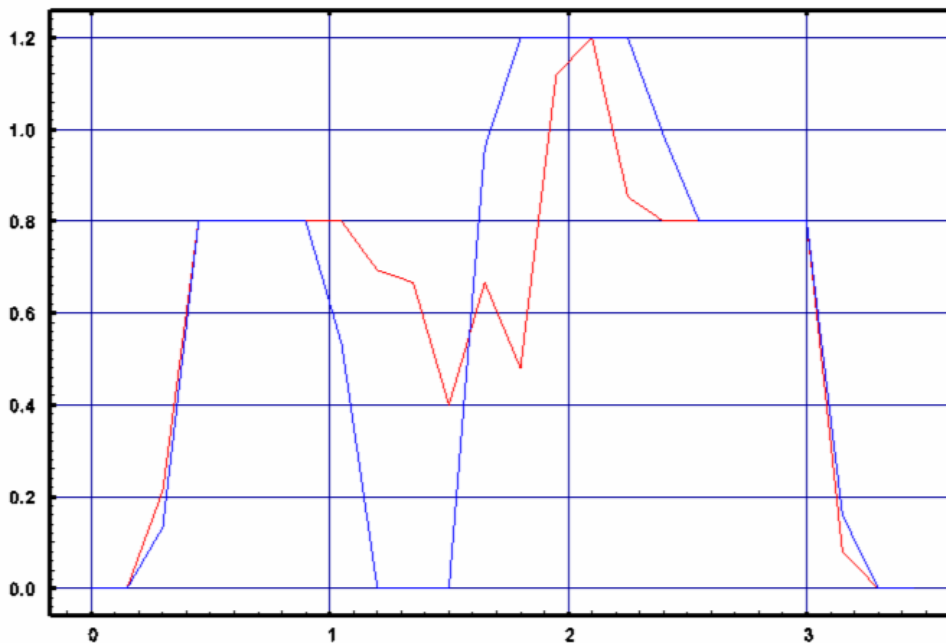
Απάντηση : Για τη μπλε ροή έχουμε πριν τη διακοπή ζεύξης τη διαδρομή : 2 - 1 - 0.

Για την κόκκινη ροή πριν την διακοπή ζεύξης έχουμε τη διαδρομή : 3 - 1. Μετά τη διακοπή της ζεύξης , για την κόκκινη ροή δεν αλλάζει τίποτα, παραμένει ίδια : 3 - 1. Για τη μπλε ροή προκύπτει το πρόβλημα της μέτρησης ως το άπειρο μέσω του πρωτοκόλλου DV. Αρχικά , μέχρι να ενημερωθεί ο 2 ότι η ζεύξη έχει διακοπεί συνεχίζει να στέλνει πακέτα στον 1, τα οποία χάνονται. Στη συνέχεια , μόλις φτάσουν τα πρωτόκολλα protoDV και ενημερωθεί ο 2 , θα στείλει πακέτα προς τον 3. Η διαδρομή , όμως που ακολουθούν τα πακέτα δεν είναι γνωστή ακόμα από τον 2. Μόλις φτάσουν πακέτα στον 3 , μερικά θα γυρίσουν πίσω στον 2 και άλλα θα κατευθυνθούν προς τον 1, τα οποία από τον 1 θα κατευθυνθούν προς τον κόμβο 2 . Τελικά , τα πακέτα δημιουργούν κύκλους και τελικά πολλά από αυτά στοιβάζονται στις DropTail ουρές αναμονής . Μετά την επαναφορά της ζεύξης , τη στιγμή που ενημερωθούν οι κόμβοι για τη λειτουργικότητα της ζεύξης 0 - 1, θα αρχίσουν να στέλνουν πακέτα από την αρχική διαδρομή . Αυτά τα πακέτα θα είναι τόσο από τις ουρές αναμονής στους κόμβους 2,3 (μέχρι να αδειάσουν) όσο και νέα πακέτα .

- Να παραθέσετε τα διαγράμματα του xgraph που δείχνουν τον ρυθμό άφιξης δεδομένων στους προορισμούς τους. Σχολιάστε τους παρατηρούμενους ρυθμούς άφιξης δεδομένων για κάθε ροή.

Απάντηση

:



Παρατηρούμε ότι τη στιγμή που διακόπτεται η ζεύξη, διακόπτεται και η μπλε ροή. Επίσης , για τον αρχικό ρυθμό μετάδοσης βλέπουμε ότι επιβεβαιώνεται , καθώς είναι : $\frac{500 \times 8}{5 \times 10^{-3}} = 0.8 \text{ Mbits/sec}$. Τέλος , βλέπουμε ότι μετά την επαναφορά της ζεύξης αυξάνεται ο ρυθμός μετάδοσης και φτάνει στα 1.2 Mbits/sec , μεγαλύτερο από τον αρχικό ρυθμό μετάδοσης, κάτι που οφείλεται στα πακέτα που έρχονται από τις ουρές αναμονής.

- Επηρεάζεται η ροή πακέτων από τον κόμβο 3 προς τον 1 λόγω της αστοχίας της ζεύξης 0-1;

Απάντηση: Η ροή των πακέτων θα επηρεαστεί κι αυτό οφείλεται στο ότι, όταν ο 2 αρχίζει να στέλνει πακέτα στον 3, τότε από τον 3 μπλε πακέτα θα πάνε πίσω στον κόμβο 2 και άλλα μπλε θα πάνε προς τον κόμβο 1, καταλαμβάνοντας έτσι χωρητικότητα ζεύξης που χρησιμοποιεί ταυτόχρονα η κόκκινη ροή, στοιβάζοντας στην ουρά αναμονής τόσο μπλε όσο και κόκκινα πακέτα.

- Υπολογίστε το πλήθος των πακέτων που χάνεται για κάθε ροή, με τη βοήθεια του κατάλληλου script σε γλώσσα awk.

Απάντηση: Το πρόγραμμα awk που χρησιμοποιήθηκε είναι το παρακάτω :

```
BEGIN {
    packet1=0
    packet2=0}
/^d/&&/cbr/ {
    if($8 == 0)
        packet1++;
    if($8 ==1)
        packet2++;}
END{
    printf("Packets lost flow1 %d\n",packet1);
    printf("Packets lost flow2 %d\n",packet2);}
```

Πήραμε τα παρακάτω αποτελέσματα :

```
Packets lost flow1 40
Packets lost flow2 13
```

- Προτείνετε κάποια λύση για την αντιμετώπιση αυτής της συμπεριφοράς.

Απάντηση: Θα μπορούσαμε να κατασκευάσουμε έτσι το δίκτυό μας, ώστε να γίνεται συνεχής ενημέρωση όλων των κόμβων σε όλους τους κόμβους, έτσι ώστε όταν ένας κόμβος (στην περίπτωση μας ο 2) στέλνει πακέτα σε έναν άλλον (στην περίπτωσή μας στον 3) να ενημερώνεται για το αν η διαδρομή των πακέτων περιέχει τον ίδιο κόμβο, έτσι ώστε να μη δημιουργείται κύκλος που θα φορτώνει τις χωρητικότητες ζεύξεων.

- Όσο η ζεύξη 0-1 είναι εκτός λειτουργίας, διαφοροποιείται καθόλου η κατάσταση εάν μεταβληθεί ο μέγιστος ρυθμός μετάδοσης που μπορεί να υποστηριχθεί από τις υπόλοιπες ζεύξεις; Αιτιολογείστε την απάντησή σας τόσο θεωρητικά, όσο και πειραματικά με τη βοήθεια του NS2.

Απάντηση: Αυτό που μπορούμε να πετύχουμε με μεγαλύτερο ρυθμό μετάδοσης είναι να πετύχουμε μια αποσυμφόρηση, όσον αφορά τα πακέτα που μπαίνουν στις ουρές αναμονής, λόγω της αδυναμίας της ζεύξης να “χωρέσει” όλα τα πακέτα που στέλνονται.

Παραδείγματος χάριν , για ρυθμό μετάδοσης 15 Mbits/sec , προέκυψε το παρακάτω :

