

Δίκτυα Επικοινωνιών

Εργαστηριακή Άσκηση 9

Νικόλαος Ζαρίφης
03112178

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.

Για το ονόματα των κόμβων προσθέτουμε το ακόλουθο:

```
for {set i 0} {$i < 5} {incr i} {  
  set n($i) [$ns node]  
  $n($i) label "Node $i"  
}
```

Ενώ για τις ταχύτητες - ουρά:

```
$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) $n(2) label "Queue 2,$i :10,$i->2:30,speed: 2Mb"
```

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

Προσθέτοντας τις ακόλουθες εντολές:

```
$ns duplex-link-op $n(2) $n(0) orient left-up  
$ns duplex-link-op $n(2) $n(3) orient right-up  
$ns duplex-link-op $n(2) $n(4) orient right-down  
$ns duplex-link-op $n(2) $n(1) orient left-down
```

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

Στον κόμβο 0 έχουμε CBR και FTP:

```
set cbr1 [new Application/Traffic/CBR]  
$cbr1 attach-agent $udp1  
$ns attach-agent $n(0) $udp1
```

```
$ns attach-agent $n(0) $tcp2  
set ftp2 [new Application/FTP]  
$ftp2 attach-agent $tcp2
```

Για το κόμβο 1 έχουμε EXP κι telnet:

```
$ns attach-agent $n(1) $udp3
set exp3 [new Application/Traffic/Exponential]
$exp3 attach-agent $udp3
$ns attach-agent $n(1) $tcp4
set telnet4 [new Application/Telnet]
$telnet4 attach-agent $tcp4
```

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

Για τα CBR και EXP έχουμε UDP ενώ για telnet και ftp έχουμε TCP.

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

Ο κόμβος 0 CBR στέλνει ως προς τον 3:

```
set null1 [new Agent/Null]
$ns attach-agent $n(3) $null1
$ns connect $udp1 $null1
```

Επίσης FTP ως προς τον 4:

```
set sink2 [new Agent/TCPSink]
$ns attach-agent $n(4) $sink2
$ns connect $tcp2 $sink2
```

Ο κόμβος 1 EXP στέλνει στον 3:

```
set null3 [new Agent/Null]
$ns attach-agent $n(3) $null3
$ns connect $udp3 $null3
```

Ενώ telnet στον 4:

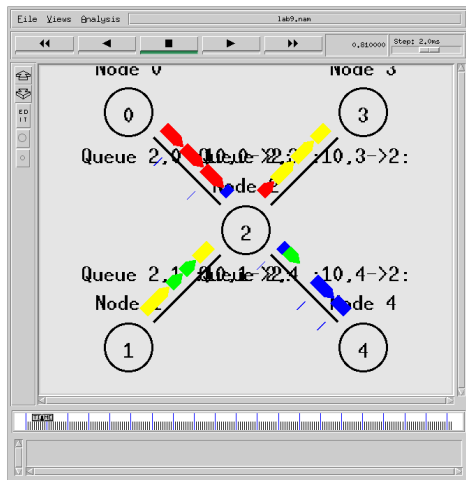
```
set sink4 [new Agent/TCPSink]
$ns attach-agent $n(4) $sink4
$ns connect $tcp4 $sink4
```

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

Εντολές:

```
$ns color 1 red
$ns color 2 blue
$ns color 3 yellow
```

\$ns color 4 green



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.

Το awk έβγαλε τα ακόλουθα αποτελέσματα:

flow 1

Total packets : 334
Total data : 501000
Total time : 4.144000
Total packets lost : 0
Total data lost : 0
Total packets sent from node0: 334
Total data sent from node0: 501000
Total packets received to node3: 334
Total data received from node3: 501000

flow 2

Total packets : 294
Total data : 441000
Total time : 4.492161
Total packets lost : 8
Total data lost : 12000
Total packets sent from node0: 294
Total data sent from node0: 441000
Total packets received to node4: 286
Total data received from node4: 429000

flow 3

Total packets : 104
Total data : 156000
Total time : 3.874000
Total packets lost : 0
Total data lost : 0

Total packets sent from node0: 104
Total data sent from node1: 156000
Total packets received to node3: 104
Total data received from node3: 156000
flow 4
Total packets : 3963
Total data : 3963000
Total time : 19.529841
Total packets lost : 4
Total data lost : 4000
Total packets sent from node0: 3963
Total data sent from node0: 3963000
Total packets received to node4: 3959
Total data received from node4: 3959000

To script:

```
BEGIN {
    packets[1]=0;
    packets[2]=0;
    packets[3]=0;
    packets[4]=0;
    data[1]=0;
    data[2]=0;
    data[3]=0;
    data[4]=0;
    lpackets[1]=0;
    lpackets[2]=0;
    lpackets[3]=0;
    lpackets[4]=0;
    ldata[1]=0;
    ldata[2]=0;
    ldata[3]=0;
    ldata[4]=0;
    startpnode[1]=0;
    startpnode[2]=0;
    startpnode[3]=0;
    startpnode[4]=0;
    startdnode[1]=0;
    startdnode[2]=0;
    startdnode[3]=0;
    startdnode[4]=0;
}
/^r/&&(/cbr/||/exp/||/tcp/){
    ltime[$8]=2;
    if(($4 == 3) && ($6 > 40))
    {
        pnode3[$8]++;
        dnode3[$8]+=$6;
    }
    if(($4 == 4) && ($6 > 40))
    {
        pnode4[$8]++;
        dnode4[$8]+=$6;
    }
}
/^d/&&(/cbr/||/exp/||/tcp/){
    lpackets[$8]++;
    ldata[$8]+=$6;
```

```

    }

/^\/&&(cbr\/|exp\/|tcp\/){
    if($6 >40)
    {
        if($3 == 0 || $3 ==1){
            packets[$8]++;
            data[$8]+= $6;
            startpnode[$8]++;
            startdnode[$8]+=$6;}
    }
}
END{
    printf("flow 1\n");
    printf("Total packets : %d\n",packets[1]);
    printf("Total data : %d\n",data[1]);
    printf("Total time : %f\n",(ltime[1] -0.15));
    printf("Total packets lost : %d\n",lpackets[1]);
    printf("Total data lost : %d\n",ldata[1]);
    printf("Total packets sent from node0: %d\n" ,startpnode[1]);
    printf("Total data sent from node0: %d\n" ,startdnode[1]);
    printf("Total packets received to node3: %d\n" ,pnode3[1]);
    printf("Total data received from node3: %d\n" ,dnode3[1]);
    printf("flow 2\n");
    printf("Total packets : %d\n",packets[2]);
    printf("Total data : %d\n",data[2]);
    printf("Total time : %f\n",(ltime[2] -0.3));
    printf("Total packets lost : %d\n",lpackets[2]);
    printf("Total data lost : %d\n",ldata[2]);
    printf("Total packets sent from node0: %d\n" ,startpnode[2]);
    printf("Total data sent from node0: %d\n" ,startdnode[2]);
    printf("Total packets received to node4: %d\n" ,pnode4[2]);
    printf("Total data received from node4: %d\n" ,dnode4[2]);
    printf("flow 3\n");
    printf("Total packets : %d\n",packets[3]);
    printf("Total data : %d\n",data[3]);
    printf("Total time : %f\n",(ltime[3] -0.45));
    printf("Total packets lost : %d\n",lpackets[3]);
    printf("Total data lost : %d\n",ldata[3]);
    printf("Total packets sent from node0: %d\n" ,startpnode[3]);
    printf("Total data sent from node1: %d\n" ,startdnode[3]);
    printf("Total packets received to node3: %d\n" ,pnode3[3]);
    printf("Total data received from node3: %d\n" ,dnode3[3]);
    printf("flow 4\n");
    printf("Total packets : %d\n",packets[4]);
    printf("Total data : %d\n",data[4]);
    printf("Total time : %f\n",(ltime[4] -0.6));
    printf("Total packets lost : %d\n",lpackets[4]);
    printf("Total data lost : %d\n",ldata[4]);
    printf("Total packets sent from node0: %d\n" ,startpnode[4]);
    printf("Total data sent from node0: %d\n" ,startdnode[4]);
    printf("Total packets received to node4: %d\n" ,pnode4[4]);
    printf("Total data received from node4: %d\n" ,dnode4[4]);
}

```

Βλέπουμε πως απορρίπτονται πακέτα μόνο από τις ροές 2,4. Γιατί εκεί γεμίζει η ουρά κι χάνονται τα πακέτα που δεν χωράνε .

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.

Θα χρησιμοποιήσω δυαδική αναζήτηση κι πειραμάτικα θα βρω το ελάχιστο μέγεθος. Και πράγματι βλέπουμε ότι για το `$ns queue-limit $n(2) $n(4) 21` δεν έχουμε καμία απώλεια. Ενώ για 20 έχουμε ένα χαμένο πακέτο στην ροή 4.

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.

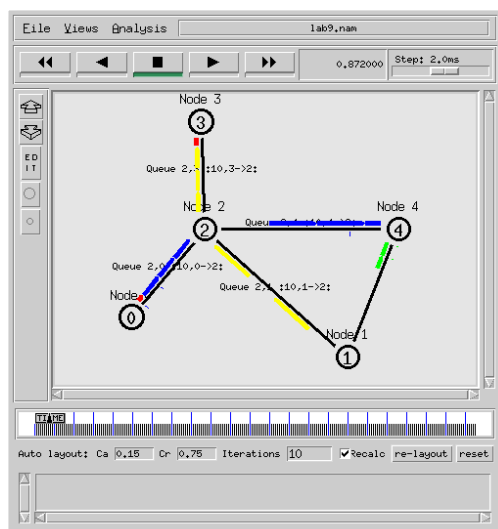
Στην αρχική τοπολογία η Ροή 1: 0->2->3, Ροή 2:0->2->4, Ροή 3: 1->2->3 και Ροή 4: 1->2->4. Βάζοντας τις γραμμές στον κώδικα

```
$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
```

Οι ροές 1,2,3 δεν αλλάζουν αλλά η ροή 4 ακολουθεί την απευθεία διαδρομή.



Awk:

flow 1

Total packets : 334

Total data : 501000

Total time : 4.216881

Total packets lost : 0

Total data lost : 0

Total packets sent from node0: 334

Total data sent from node0: 501000

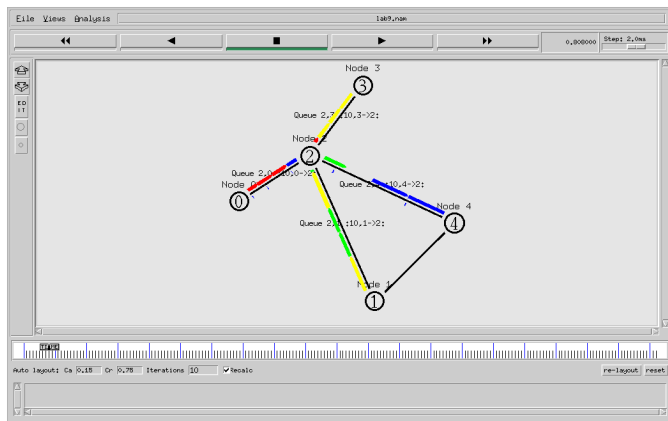
Total packets received to node3: 334

Total data received from node3: 501000

flow 2

Total packets : 341
Total data : 511500
Total time : 4.120000
Total packets lost : 0
Total data lost : 0
Total packets sent from node0: 341
Total data sent from node0: 511500
Total packets received to node4: 341
Total data received from node4: 511500
flow 3
Total packets : 104
Total data : 156000
Total time : 3.886881
Total packets lost : 1
Total data lost : 1500
Total packets sent from node0: 104
Total data sent from node1: 156000
Total packets received to node3: 103
Total data received from node3: 154500
flow 4
Total packets : 3615
Total data : 3615000
Total time : 24.399428
Total packets lost : 0
Total data lost : 0
Total packets sent from node0: 3615
Total data sent from node0: 3615000
Total packets received to node4: 3603
Total data received from node4: 3603000

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



Χρησιμοποιώντας αναλογία 10:1 πετύχαμε το αποτέλεσμα μας. Γιατί $2+2 < 5$.

Εντολές:

```
$ns cost $n($i) $n(2) 2
$ns cost $n(2) $n($i) 2
$ns cost $n(1) $n(4) 5
$ns cost $n(4) $n(1) 5
```

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 $n(4)$;

Εντολές:

```
$ns rtmodel-at 1.5 down $n(2) $n(4)
$ns rtmodel-at 3.0 up $n(2) $n(4)
```

awk:

```
flow 1
Total packets : 334
Total data : 501000
Total time : 4.096881
```



```

Total packets lost : 0
Total data lost : 0
Total packets sent from node0: 334
Total data sent from node0: 501000
Total packets received to node3: 334
Total data received from node3: 501000
flow 2
Total packets : 97
Total data : 145500
Total time : 3.990320
Total packets lost : 5
Total data lost : 7500
Total packets sent from node0: 97
Total data sent from node0: 145500
Total packets received to node4: 90
Total data received from node4: 135000
flow 3
Total packets : 104
Total data : 156000
Total time : 3.850881
Total packets lost : 0
Total data lost : 0
Total packets sent from node0: 104
Total data sent from node1: 156000
Total packets received to node3: 104
Total data received from node3: 156000
flow 4
Total packets : 3968
Total data : 3968000
Total time : 22.068240
Total packets lost : 6
Total data lost : 6000
Total packets sent from node0: 3968
Total data sent from node0: 3968000
Total packets received to node4: 3959
Total data received from node4: 3959000

```

Η 2 ροή ξεκίνα στο 3,8 , η 4 στο 4,6 κι η 3 στο 2,5 .

Τροποποιήστε τον κώδικα της προσομοίωσης του ερωτήματος (ια), 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 $n(2)$ - $n(4)$;

Βάζοντας δυναμική δρομολόγηση μπορούμε να το επιτύχουμε:

```
Agent/rtProto/Direct set preference_200
```

```
$ns rtproto DV
```

```
awk:
```

```
flow 1
```

Total packets : 334
Total data : 501000
Total time : 4.074881
Total packets lost : 0
Total data lost : 0
Total packets sent from node0: 334
Total data sent from node0: 501000
Total packets received to node3: 334
Total data received from node3: 501000
flow 2
Total packets : 208
Total data : 312000
Total time : 4.066980
Total packets lost : 5
Total data lost : 7500
Total packets sent from node0: 208
Total data sent from node0: 312000
Total packets received to node4: 183
Total data received from node4: 274500
flow 3
Total packets : 104
Total data : 156000
Total time : 3.924881
Total packets lost : 0
Total data lost : 0
Total packets sent from node0: 104
Total data sent from node1: 156000
Total packets received to node3: 104
Total data received from node3: 156000
flow 4
Total packets : 3965
Total data : 3965000
Total time : 19.873320
Total packets lost : 6
Total data lost : 6000
Total packets sent from node0: 3965
Total data sent from node0: 3965000
Total packets received to node4: 3959
Total data received from node4: 3959000

Η πράσινη ροή ακολουθεί την 1->4 κι ξεκινά στο 1,65, η μπλε 1,8 και ακολουθεί 0->2->1->4, η μπλε επαναφέρεται στο 3,1 όπως κι η πράσινη , αυτά τα παρατήρησα απο το nam.

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 ns [new Simulator]
set tf [open lab9.tr w]
set nf [open lab9.nam w]
```

```

$ns use-newtrace
$ns trace-all $tf
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
$ns namtrace-all-wireless $nf $opt(gridx) $opt(gridy)
proc finish {} {
    global ns tf nf
    $ns flush-trace
    close $tf
    close $nf
    exit 0
}
set topo [new Topography]
$topo load_flatgrid $opt(gridx) $opt(gridy)
create-god $opt(nn)
$ns node-config -adhocRouting $opt(rp) \
    -llType $opt(ll) \
    -macType $opt(mac) \
    -ifqType $opt(ifq) \
    -ifqLen $opt(ifqlen) \
    -antType $opt(ant) \
    -propType $opt(prop) \
    -phyType $opt(netif) \
    -channel [new $opt(chan)] \
    -topoInstance $topo \
    -agentTrace ON \
    -routerTrace ON \
    -macTrace OFF \
    -movementTrace OFF

set n(0) [$ns node]
$n(0) random-motion 0
$n(0) set X_ 300.0
$n(0) set Y_ 200.0
$n(0) set Z_ 0.0
set n(1) [$ns node]
$n(1) random-motion 0
$n(1) set X_ 100.0
$n(1) set Y_ 200.0
$n(1) set Z_ 0.0
set n(2) [$ns node]
$n(2) random-motion 0
$n(2) set X_ 300.0
$n(2) set Y_ 0.0
$n(2) set Z_ 0.0
set n(3) [$ns node]
$n(3) random-motion 0
$n(3) set X_ 500.0
$n(3) set Y_ 200.0
$n(3) set Z_ 0.0

```

```

set n(4) [$ns node]
$n(4) random-motion 0
$n(4) set X_ 300.0
$n(4) set Y_ 400.0
$n(4) set Z_ 0.0

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
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
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
set cbr3 [new Application/Traffic/CBR]
$cbr3 attach-agent $udp3
$cbr3 set packetSize_ 1500
$cbr3 set rate_ 1.2mb
$cbr3 set random_ off
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
# Events
$ns at 0.15 "$cbr1 start"
$ns at 0.3 "$ftp2 start"
$ns at 0.45 "$cbr3 start"
$ns at 0.6 "$telnet4 start"
$ns at 4.15 "$cbr1 stop"
$ns at 4.3 "$ftp2 stop"
$ns at 4.45 "$cbr3 stop"
$ns at 4.6 "$telnet4 stop"
$ns at 25.0 "finish"
$ns run

```

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.

Το awk μας έβγαλε:

flow 1

Total packets : 334

Total data : 501000

Total time : 4.063527

Total packets lost : 6

Total data lost : 501000

Total packets sent from node0: 334

Total data sent from node0: 501000

Total packets received to node3: 328

Total data received from node3: 498560

flow 2

Total packets : 498

Total data : 747000

Total time : 4.076536

Total packets lost : 9

Total data lost : 13500

Total packets sent from node0: 498

Total data sent from node0: 747000

Total packets received to node4: 489

Total data received from node4: 733500

flow 3

Total packets : 401

Total data : 601500

Total time : 4.126072

Total packets lost : 135

Total data lost : 601500

Total packets sent from node0: 401

Total data sent from node1: 601500

Total packets received to node3: 266

Total data received from node3: 404320

flow 4

Total packets : 2569

Total data : 2569000

Total time : 24.399519

Total packets lost : 16

Total data lost : 16000

Total packets sent from node0: 2569

Total data sent from node0: 2569000

Total packets received to node4: 2553

Total data received from node4: 2553000

script:

```
BEGIN {
    packets[1]=0;
    packets[2]=0;
    packets[3]=0;
    packets[4]=0;
    data[1]=0;
    data[2]=0;
    data[3]=0;
    data[4]=0;
    lpackets[1]=0;
    lpackets[2]=0;
    lpackets[3]=0;
    lpackets[4]=0;
    ldata[1]=0;
    ldata[2]=0;
    ldata[3]=0;
    ldata[4]=0;
    startpnode[1]=0;
    startpnode[2]=0;
    startpnode[3]=0;
    startpnode[4]=0;
    startdnode[1]=0;
    startdnode[2]=0;
    startdnode[3]=0;
    startdnode[4]=0;
}
/^r/ && (/cbr/ || /exp/ || /tcp/) && /AGT/ {
    ltime[$39]=$3;
    if(($7 == 3) && ($37 > 40))
    {
        pnode3[$39]++;
        dnode3[$39]+=$37;
    }
    if(($7 == 4) && ($37 > 40))
    {
        pnode4[$39]++;
        dnode4[$39]+=$37;
    }
}

/^s/ && (/cbr/ || /exp/ || /tcp/) && /AGT/ {

    if($37 > 40)
    {
        if($5 == 0 || $5 == 1){
            packets[$39]++;
            data[$39]+=$37;
            startpnode[$39]++;
            startdnode[$39]+=$37;}
    }
}
END{
    printf("flow 1\n");
    printf("Total packets : %d\n",packets[1]);
    printf("Total data : %d\n",data[1]);
    printf("Total time : %f\n",(ltime[1] -0.15));
    printf("Total packets lost : %d\n",startpnode[1]-pnode3[1]);
```

```

printf("Total data lost : %d\n",startdnode[1]-dnode[3]);
printf("Total packets sent from node0: %d\n",startpnode[1]);
printf("Total data sent from node0: %d\n",startdnode[1]);
printf("Total packets received to node3: %d\n",pnode3[1]);
printf("Total data received from node3: %d\n",dnode3[1]);
printf("flow 2\n");
printf("Total packets : %d\n",packets[2]);
printf("Total data : %d\n",data[2]);
printf("Total time : %f\n",(ltime[2]-0.3));
printf("Total packets lost : %d\n",startpnode[2]-pnode4[2]);
printf("Total data lost : %d\n",startdnode[2]-dnode4[2]);
printf("Total packets sent from node0: %d\n",startpnode[2]);
printf("Total data sent from node0: %d\n",startdnode[2]);
printf("Total packets received to node4: %d\n",pnode4[2]);
printf("Total data received from node4: %d\n",dnode4[2]);
printf("flow 3\n");
printf("Total packets : %d\n",packets[3]);
printf("Total data : %d\n",data[3]);
printf("Total time : %f\n",(ltime[3]-0.45));
printf("Total packets lost : %d\n",startpnode[3]-pnode3[3]);
printf("Total data lost : %d\n",startdnode[3]-dnode[3]);
printf("Total packets sent from node0: %d\n",startpnode[3]);
printf("Total data sent from node1: %d\n",startdnode[3]);
printf("Total packets received to node3: %d\n",pnode3[3]);
printf("Total data received from node3: %d\n",dnode3[3]);
printf("flow 4\n");
printf("Total packets : %d\n",packets[4]);
printf("Total data : %d\n",data[4]);
printf("Total time : %f\n",(ltime[4]-0.6));
printf("Total packets lost : %d\n",startpnode[4]-pnode4[4]);
printf("Total data lost : %d\n",startdnode[4]-dnode4[4]);
printf("Total packets sent from node0: %d\n",startpnode[4]);
printf("Total data sent from node0: %d\n",startdnode[4]);
printf("Total packets received to node4: %d\n",pnode4[4]);
printf("Total data received from node4: %d\n",dnode4[4]);
}

```

Υποψιαζόμαστε ότι από την ώρα που ήρθε το τελευταίο πακέτο στην ροή 4 ότι δεν στέλνονται όλα τα πακέτα.

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;

Τρέχοντας την προσομοίωση για 50 δεύτερα βλέπουμε:

```

flow 1
Total packets : 334
Total data : 501000
Total time : 4.063527
Total packets lost : 6
Total data lost : 501000
Total packets sent from node0: 334
Total data sent from node0: 501000
Total packets received to node3: 328
Total data received from node3: 498560
flow 2
Total packets : 498
Total data : 747000

```

Total time : 4.076536
Total packets lost : 9
Total data lost : 13500
Total packets sent from node0: 498
Total data sent from node0: 747000
Total packets received to node4: 489
Total data received from node4: 733500

flow 3

Total packets : 401
Total data : 601500
Total time : 4.126072
Total packets lost : 135
Total data lost : 601500
Total packets sent from node0: 401
Total data sent from node1: 601500
Total packets received to node3: 266
Total data received from node3: 404320

flow 4

Total packets : 4012
Total data : 4012000
Total time : 35.004620
Total packets lost : 7
Total data lost : 7000
Total packets sent from node0: 4012
Total data sent from node0: 4012000
Total packets received to node4: 4005
Total data received from node4: 4005000

Άρα για 35,1 second θα πρέπει να τρέξουμε την προσομοίωση .