

Εθνικό Μετσόβιο Πολυτεχνείο

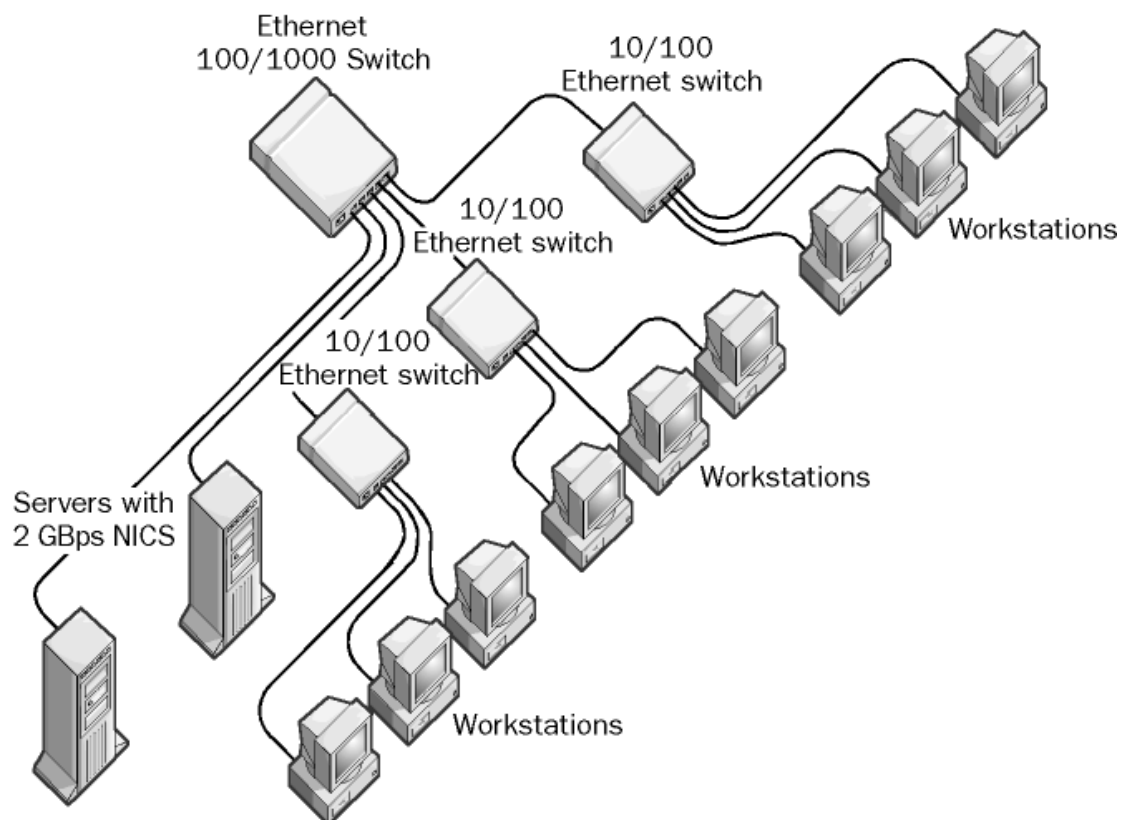
Σχολή Ηλεκτρολόγων Μηχ. Και Μηχ. Υπολογιστών

Δίκτυα Επικοινωνιών , 6^ο εξάμηνο

Ακαδημαϊκή Περίοδος : 2010-2011



7^η Εργαστηριακή Άσκηση



Γερακάρης Βασίλης
Α.Μ. :03108092

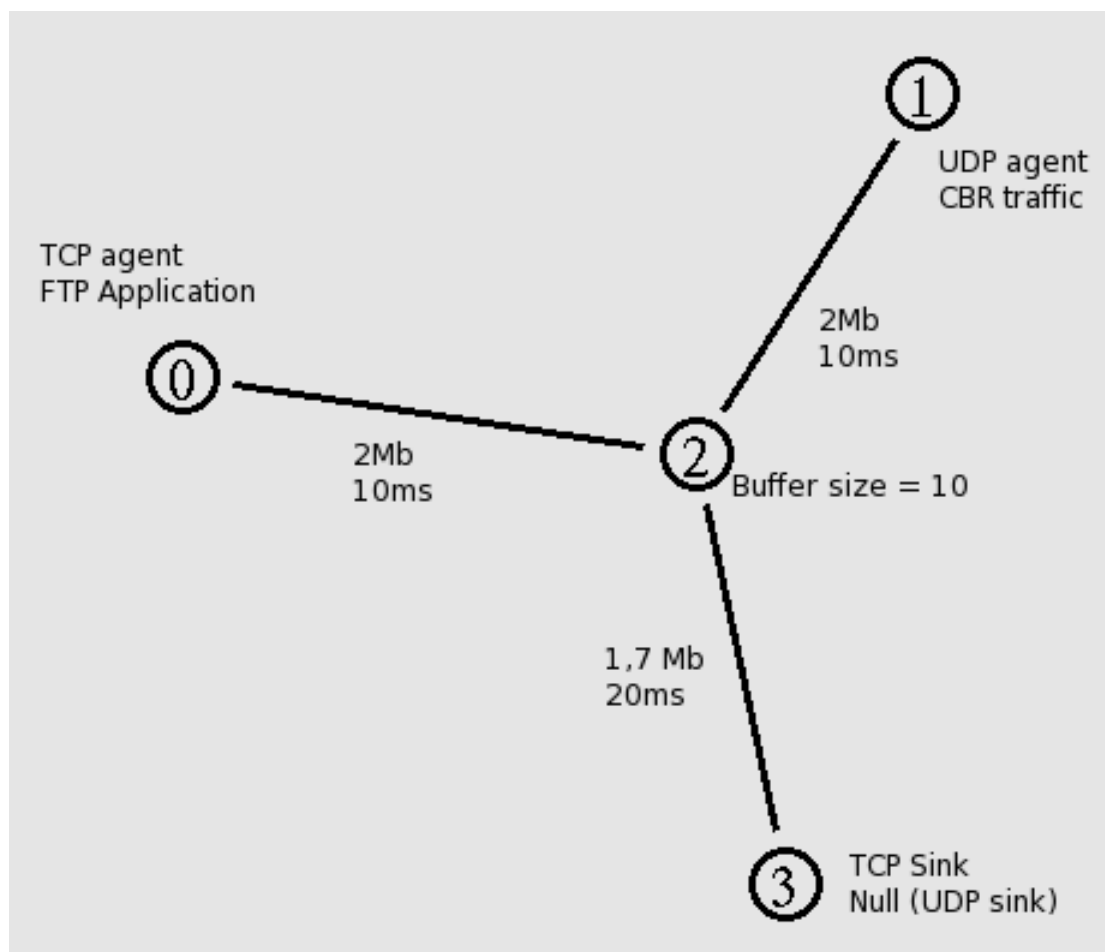
- Σκοπός:

Σε αυτή την εργαστηριακή άσκηση, κληθήκαμε να χρησιμοποιήσουμε τις προαποκτιθείσες γνώσεις μας πάνω στη χρήση του λογισμικού NetworkSimulator2 για να δημιουργήσουμε μόνοι μας την τοπολογία που περιγράφει η εκφώνηση και τα γεγονότα πάνω σε αυτή.

(α) – Τοπολογία

Η τοπολογία που εξετάζουμε είναι σχετικά απλή, αποτελείται από 4 κόμβους και αμφίδρομες ζεύξεις. Οι κόμβοι n0 και n1 παράγουν δεδομένα προς αποστολή, ο n2 λειτουργεί ως δρομολογητής (router), και ο n3 ως καταβόθρα (sink) δεδομένων.

Η σχηματική αναπαράσταση φαίνεται στο παρακάτω σχήμα:



Οι εντολές με τις οποίες ορίζονται οι κόμβοι και η σύνδεση μεταξύ τους είναι οι εξής:

```
22 #####
23 ### TOPOLOGIA ###
24 #####
25
26 ##Dimiourgia 4 komvwn##
27 set n0 [$ns node]
28 set n1 [$ns node]
29 set n2 [$ns node]
30 set n3 [$ns node]
31
32 ##Dimiourgia Zeyksewn##
33 $ns duplex-link $n0 $n2 2Mb 10ms DropTail
34 $ns duplex-link $n1 $n2 2Mb 10ms DropTail
35 $ns duplex-link $n2 $n3 1.7Mb 20ms DropTail
36 #Kathorizoume to megethos tis ouras (buffer)
37 $ns queue-limit $n2 $n3 10 ;
```

(β,γ) – Εφαρμογές στους κόμβους

Στον κόμβο n0 προσαρτήθηκε μια πηγή FTP η οποία θα αποστέλλει τα δεδομένα της μέσω του πρωτοκόλλου TCP.

```
45 #Dimiourgia TCP agent ston komvo n0
46 set tcp [new Agent/TCP]
47 $tcp set class_ 2
48 $ns attach-agent $n0 $tcp
49
50 #Prosartisi pigis FTP ston komvo n0
51 set ftp [new Application/FTP]
52 $ftp attach-agent $tcp
53 $ftp set type_ FTP
```

Στον κόμβο n1 αντίστοιχα, έχουμε πηγή δεδομένων CBR προσαρτημένη σε ένα UDP agent (που καθορίζει το πρωτόκολλο αποστολής δεδομένων)

```
61 #Dimiourgia UDP agent ston komvo n1
62 set udp [new Agent/UDP]
63 $ns attach-agent $n1 $udp
64 $udp set class_ 1
65
66 #Prosartisi pigis CBR ston komvo n1
67 set cbr [new Application/Traffic/CBR]
68 $cbr attach-agent $udp
69 $cbr set type_ CBR
70 $cbr set packet_size_ 1000 ; #Megethos paketwn
71 $cbr set rate_ 1mb ; #Rythmos metadosis dedomenwn
72 $cbr set random_ false ; #Den yparxei random thoryvos sti metadosi
```

(δ) – Κατεύθυνση δεδομένων

Τα δεδομένα που παράγονται στον κόμβο n0 στέλνονται προς ένα TCP sink agent που προσαρτήσαμε στον κόμβο n3:

```
55 #Dimiourgia TCP sink agent ston komvo n3
56 set sink [new Agent/TCPSink]
57 $ns attach-agent $n3 $sink
58 $ns connect $tcp $sink ;           #Syndesi TCP me to sink
```

Αντίστοιχα, το CBR traffic που παράγει ο UDP agent στον κόμβο n1 συνδέεται με το null sink που επίσης προσαρτούμε στον κόμβο n3:

```
74 #Dimiourgia null agent ston komvo n3
75 set null [new Agent/Null]
76 $ns attach-agent $n3 $null
77 $ns connect $udp $null ;           #Syndesi UDP me to sink
```

(ε) – Προσθήκη TCP agent με άπειρη ποσότητα δεδομένων

Με τις παρακάτω γνωστές εντολές, ορίζουμε ένα TCP agent στον κόμβο n1 και σε αυτόν προσαρτούμε μια εφαρμογή FTP, ώστε να έχουμε άπειρη ποσότητα δεδομένων προς αποστολή. Τέλος, δημιουργούμε και θέτουμε ως προορισμό το TCP sink στον κόμβο n3, όπως παραπάνω:

```
81 ### Erwtima (e) ###
82
83 #Dimiourgia TCP agent ston komvo n1
84 set tcp2 [new Agent/TCP]
85 $tcp2 set class_ 3
86 $ns attach-agent $n1 $tcp2
87
88 #Prosartisi pigis FTP ston komvo n1
89 set ftp2 [new Application/FTP]
90 $ftp2 attach-agent $tcp2
91 $ftp2 set type_ FTP
```

(στ) – Χρωματισμός Ροών

Χρησιμοποιώντας τις εντολές:

```
$udp set class_ 1  
$tcp set class_ 2  
$tcp2 set class_ 3
```

αντιστοιχίζουμε σε κάθε ροή ένα αριθμό που θα μας χρειαστεί, ώστε με τις παρακάτω εντολές να χρωματίσουμε τα δεδομένα που μεταφέρονται:

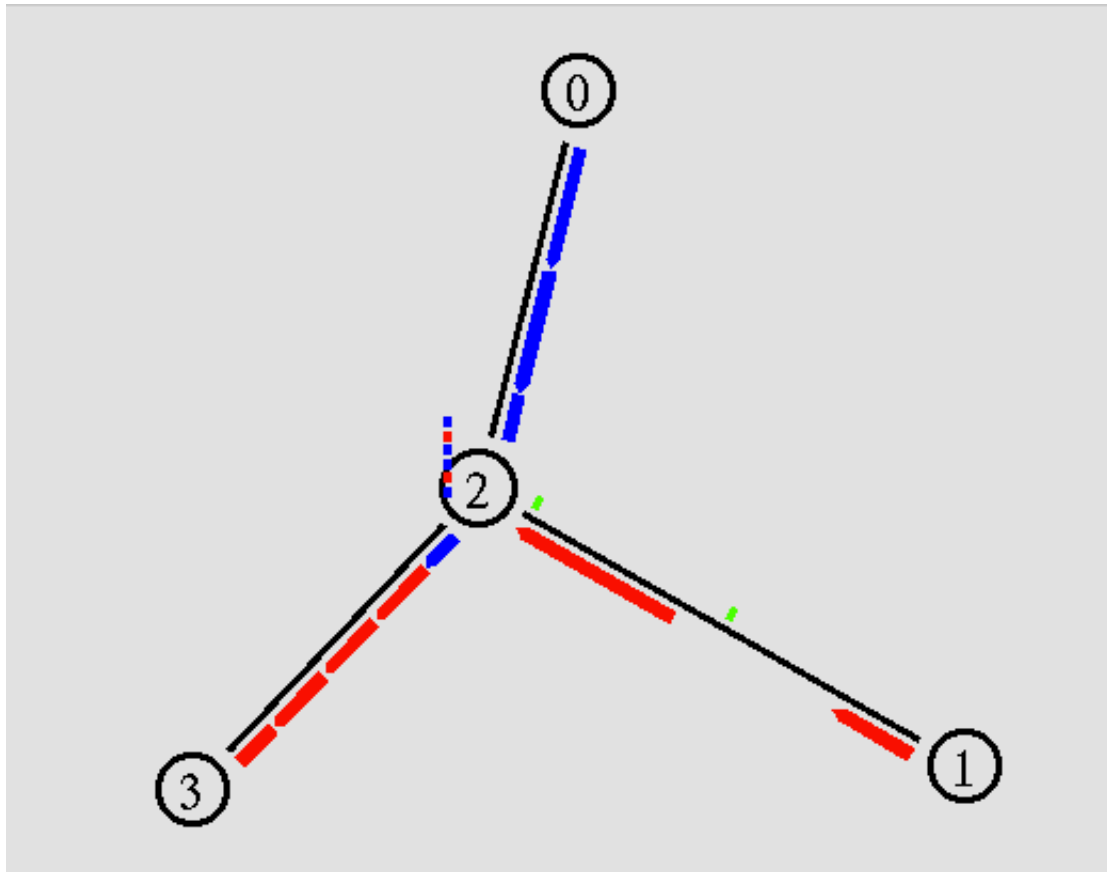
```
102 ### Xrwmatismos rown – Erwtima (st) ###  
103 $ns color 1 Red  
104 $ns color 2 Blue  
105 $ns color 3 Green
```

(ζ) – Ουρά αναμονής

Για να κάνουμε ορατή την ουρά αναμονής, αρκεί να προσθέσουμε την παρακάτω εντολή, μετά τον ορισμό των ζεύξεων:

```
39 ### Queue-monitoring gia ti zeyksi tw n2,n3 – Erwtima (z) ###  
40 $ns duplex-link-op $n2 $n3 queuePos 0.5
```

Από την παρακάτω εικόνα, παρατηρούμε πως όντως υλοποιήθηκαν σωστά και τα χρώματα και η ουρά αναμονής.



(η) – Χρήση awk για ανάλυση αποτελεσμάτων

Στο τελευταίο αυτό κομμάτι της άσκησης έπρεπε να δημιουργήσουμε ένα awk script, ώστε να αναλύσουμε τη ροή του TCP agent που δημιουργήσαμε στο ερώτημα (ε). Δημιουργώντας λοιπόν το παρακάτω script και τρέχοντας το σε περιβάλλον bash terminal με την εντολή

awk -f awkscript.awk < out.tr προέκυψαν τα παρακάτω αποτελέσματα.

```
>>= awk -f awkscript.awk < out.tr
Total Data received      : 656400 Bytes
Total Data lost : 21840 Bytes
Percentage      : 96.779901
```

Το script που χρησιμοποιήθηκε ήταν το εξής:

```
1 BEGIN {
2     data=0;
3     datalost=0;
4 }
5
6
7 /^r/ && /tcp/ && /1\.\./ {
8     data+=$6;
9 }
10
11 /^d/ && /tcp/ && /1\.\./ {
12     datalost+=$6;
13 }
14
15 END{
16     printf("Total Data received\t: %d Bytes\n", data);
17     printf("Total Data lost\t: %d Bytes\n", datalost);
18     printf("Percentage \t: %f \n", 100*data/(data+datalost));
19 }
```