

Εισαγωγικά:

Σκοπός αυτής της πέμπτης εργαστηριακής άσκησης είναι η μελέτη του πρωτοκόλλου go back N χωρίς όμως να υπάρχουν σφάλματα μετάδοσης. Η χρήση αυτού του πρωτοκόλλου θα μελετηθεί πάνω στην απλούστερη δυνατή τοπολογία δικτύου, δηλαδή αυτή που αποτελείται από δύο κόμβους. Σκοπός της χρήσης αυτού του πρωτοκόλλου είναι η εγγυημένη παράδοση δεδομένων πάνω από αναξιόπιστο δίκτυο. Οι λόγοι για τους οποίους συμβαίνει αυτό είναι κατά κύριο λόγο οι ακόλουθοι:

- Επιτρέπει αποδοτικότερη και πιο αποτελεσματική χρήση του πομπού καθώς δύναται να μεταφέρει περισσότερα πακέτα, ενώ περιμένει επιβεβαίωση (acknowledgment) για αυτά που ήδη έχουν σταλεί.
- Παράλληλα δε κάνει την όλη διαδικασία πιο σύνθετη, διότι όταν κάτι λείπει (π.χ ένα πακέτο πληροφορίας), τότε είναι πιο δύσκολο να βρεθεί τι είναι αυτό και συνακόλουθα να ξανασταλθεί.
- Επίσης πρέπει να έχουμε κατά νου ότι τόσο τα πακέτα πληροφορίας, όσο και τα μηνύματα επιβεβαίωσης έχουν και αυτά πιθανότητες να χαθούν εντός του διαύλου επικοινωνίας.

Η λειτουργία αυτού του πρωτοκόλλου βασίζεται κυρίως στην συνεχή μετάδοση πληροφοριών σύμφωνα με το ακόλουθο σχήμα:

- ✓ Ότι τα πακέτα συνεχίζουν να στέλνονται ορθά
- ✓ Η τήρηση ενός αντιγράφου για τα πακέτα είναι επιτακτική ανάγκη ειδικά για την περίπτωση αναμετάδοσης κάποιου πακέτου (ιδιότητα ανάκτησης της πληροφορίας)

- ✓ Περιορισμένο παράθυρο (δηλαδή συνολικό μέγεθος πακέτων τα οποία στέλνονται μαζί)
- ✓ Καθώς επιβεβαιώνεται η μετάδοση των πακέτων ,τότε έχουμε ολίσθηση του παραθύρου που πρόκειται να σταλεί.

Αξιίζει δε να σημειωθεί ,ότι στην συγκεκριμένη εργασία έγινε και χρήση της γλώσσας προγραμματισμού awk. Η awk είναι μια πολύ απλή γλώσσα script,η οποία προορίζεται κυρίως για την ανάλυση δεδομένων που συλλέγονται από την εκτέλεση μιας προσομοίωσης .Μέσω αυτής κατέστη εφικτή η ανάλυση και η επεξεργασία δεδομένων από τα αρχεία τα οποία ήταν σημαντικά για την μελέτη της προσομοίωσης και της εξαγωγής διαφόρων συμπερασμάτων ,πάνω στην απόδοση του δικτύου ενώ χρησιμοποιείται το εν λόγω πρωτόκολλο.

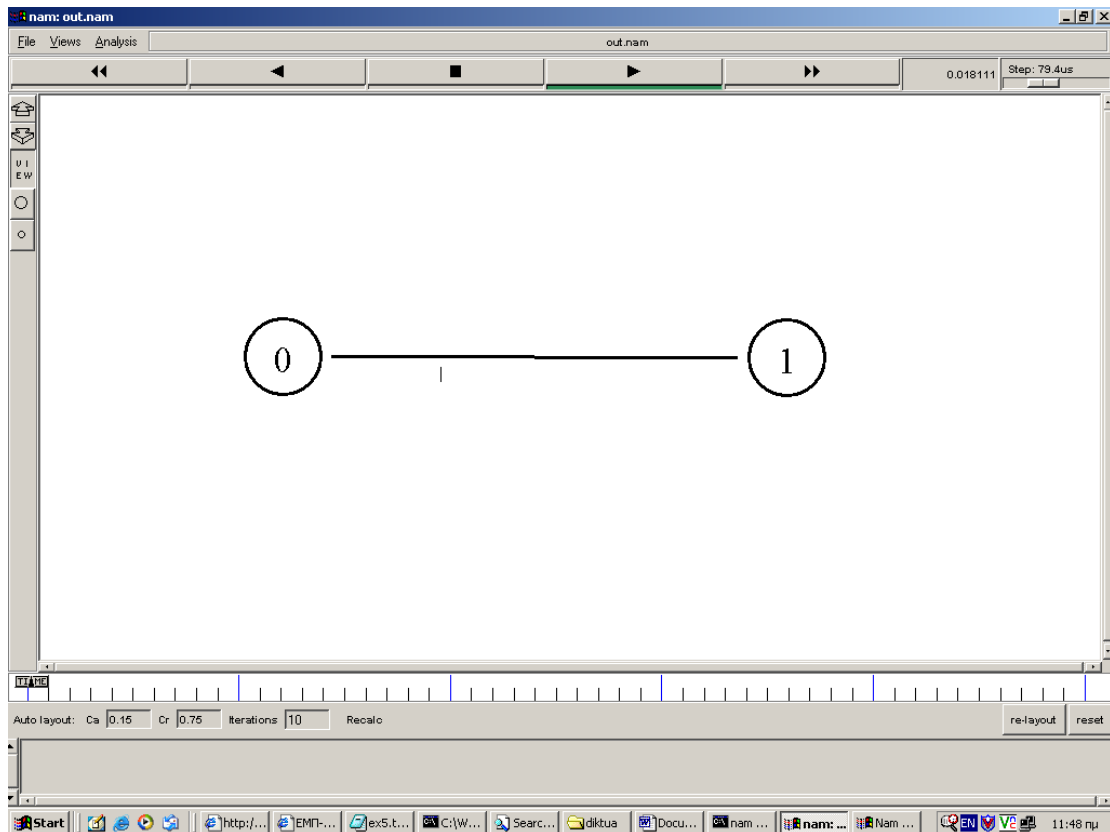
1.1Αρχικοποίηση Προσομοίωσης-Δημιουργία αρχείου ίχνους:

Σε αυτό το σημείο της εργαστηριακής άσκησης που είναι το εισαγωγικό ,κύριο μας μέλημα είναι να δημιουργήσουμε ένα αρχείο ίχνους(trace file).Σε αυτό το αρχείο αποθηκεύονται όλα τα δεδομένα της προσομοίωσης του δικτύου.Όπως είπαμε και πρωτύτερα αυτό το τμήμα θα μας χρειαστεί αργότερα στην πορεία της άσκησης.. Παρακάτω υπάρχει ο κώδικας που χρησιμοποιήθηκε για λόγους πληρότητας:

```
#Open the nam trace file
set nf [open out.nam w]
$ns namtrace-all $nf
set trf [open out.tr w]
$ns trace-all $trf
#Define a 'finish' procedure proc finish {} {
global ns nf trf
$ns flush-trace
#Close the trace file close
$nf close
$trf exit 0
}
```

1.2 Τοπολογία:

Η τοπολογία που χρησιμοποιήθηκε σε αυτήν την εργασία ήταν η απλούστερη δυνατή ,και η οποία πιο συγκεκριμένα ήταν αμφίδρομης ζεύξης και με εύρος ζώνης 10 Mbps.Οι κόμβοι που συνδέονται είναι οι n1 και n2.



Σχήμα1:Παραπάνω εικονίζεται η χρησιμοποιούμενη τοπολογία δικτύου για την μελέτη του πρωτοκόλλου go back N.

1.3 Το στρώμα μεταφοράς και το στρώμα εφαρμογής:

Το πρωτόκολλο μεταφοράς που χρησιμοποιείται εδώ είναι το TCP.Βέβαιο το πρωτόκολλο go back N περιέχεται στο προαναφερθέν πρωτόκολλο. Στο πρόγραμμα κάναμε τις ακόλουθες ρυθμίσεις.. Αρχικά ορίσαμε το μέγεθος του παραθύρου σε 5 πακέτα, και με βάση αυτά αρχικά δουλέψαμε. Επίσης το μέγεθος του πακέτου σε 1000 bits. Ιδού και ο κώδικας που χρησιμοποιήθηκε:

- ❑ set tcp0 [\$ns create-connection TCP/Reno \$n0 TCPSink \$n1 0]
 \$tcp0 set window_ 5
 \$tcp0 set packetSize_ 1000

Επίσης για να δημιουργήσουμε την κίνηση εντός του διαύλου θα θεωρήσουμε την μεταφορά ενός αρχείου μέσω του FTP το οποίο να έχει άπειρο μέγεθος για να έχουμε συνεχώς πληροφορία προς μετάδοση.Προγραμματιστικά αυτό έγινε με τις ακόλουθες εντολές::

- ❑ set ftp0 [\$tcp0 attach-app FTP]

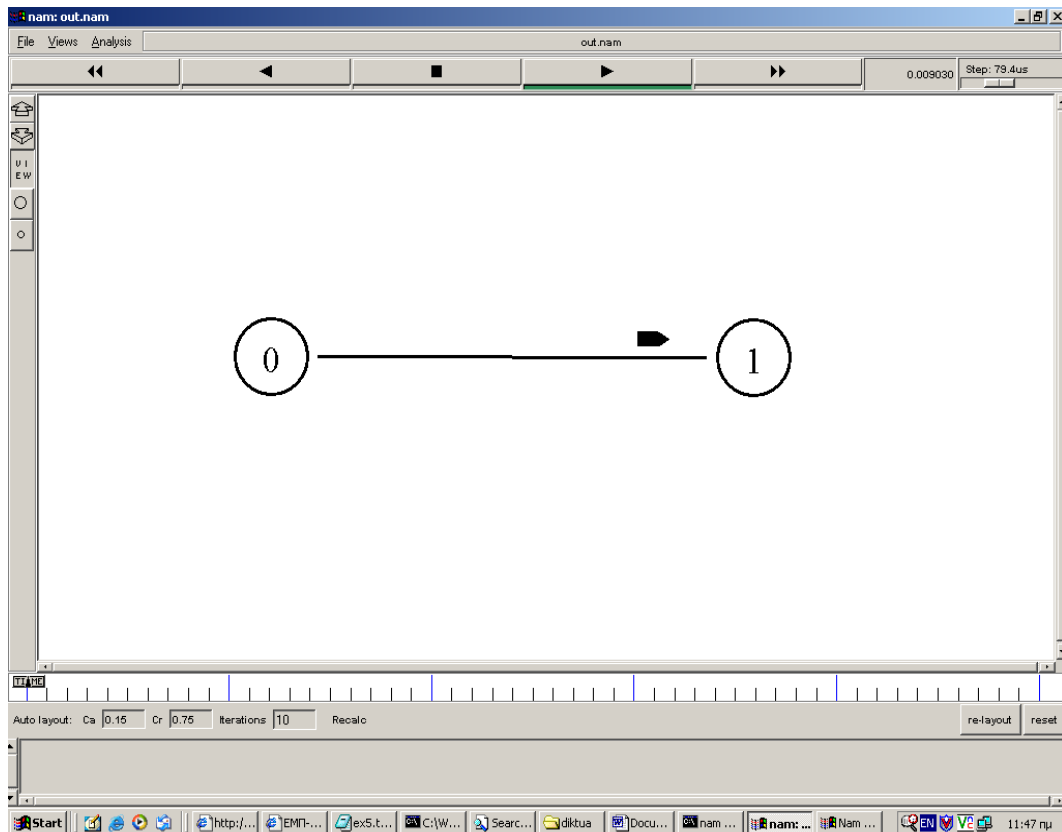
```
$ns at 0.0 "$ftp0 start"
```

1.4 Εκτέλεση του σεναρίου:

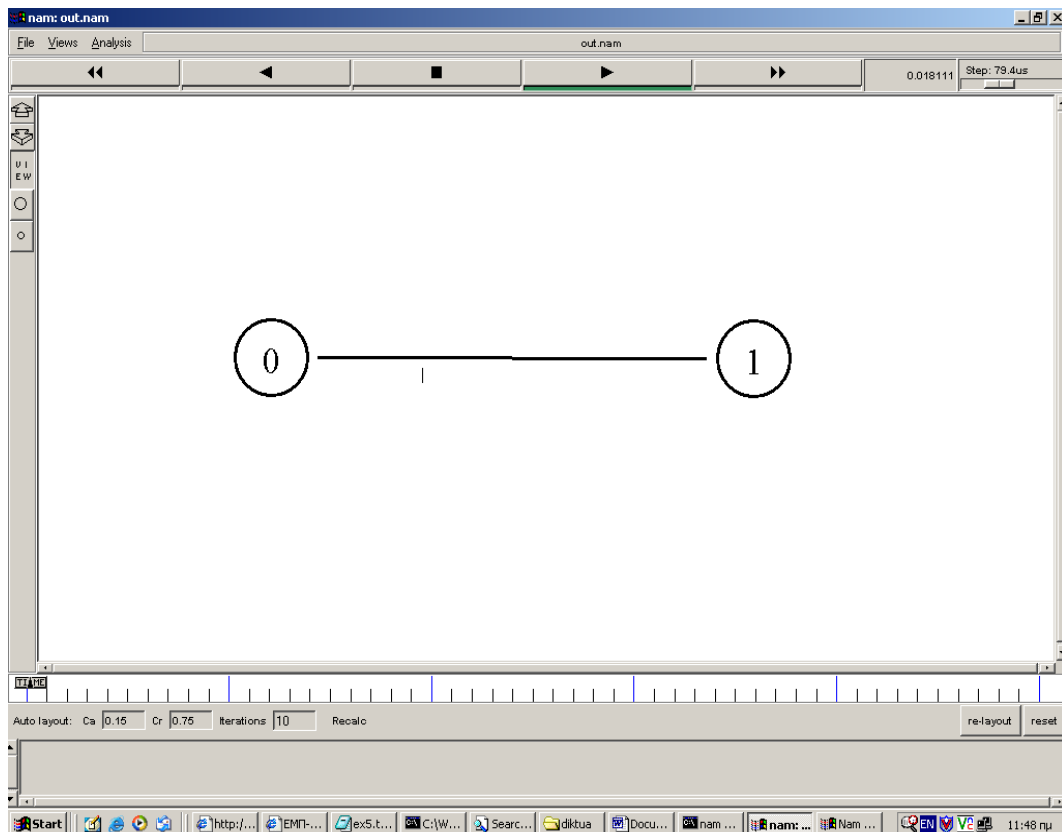
Ο κώδικας που τελικά χρησιμοποιήθηκε είναι ο ακόλουθος ,και παρακάτω δείχνουμε το εξαγόμενο του κώδικα μέσω του animation στο Nam:

```
#Create a simulator object
set ns [new Simulator]
#Open the nam trace file
set nf [open out.nam w]
$ns namtrace-all $nf
set trf [open out.tr w]
$ns trace-all $trf
#Define a 'finish' procedure
proc finish {} { global ns nf trf
$ns flush-trace
#Close the trace file
close $nf
close $trf
exit 0
}
#Create two nodes
set n0 [$ns node]
set n1 [$ns node]
#Create a duplex link between the nodes
$ns duplex-link $n0 $n1 10Mb 10ms DropTail
set tcp0 [$ns create-connection TCP/Reno $n0 TCPSink $n1 0]
$tcp0 set window_ 5
$tcp0 set packetSize_ 1000
set ftp0 [$tcp0 attach-app FTP]
$ns at 0.0 "$ftp0 start"
$ns at 5.0 "finish"
#Run the simulation
$ns run
```

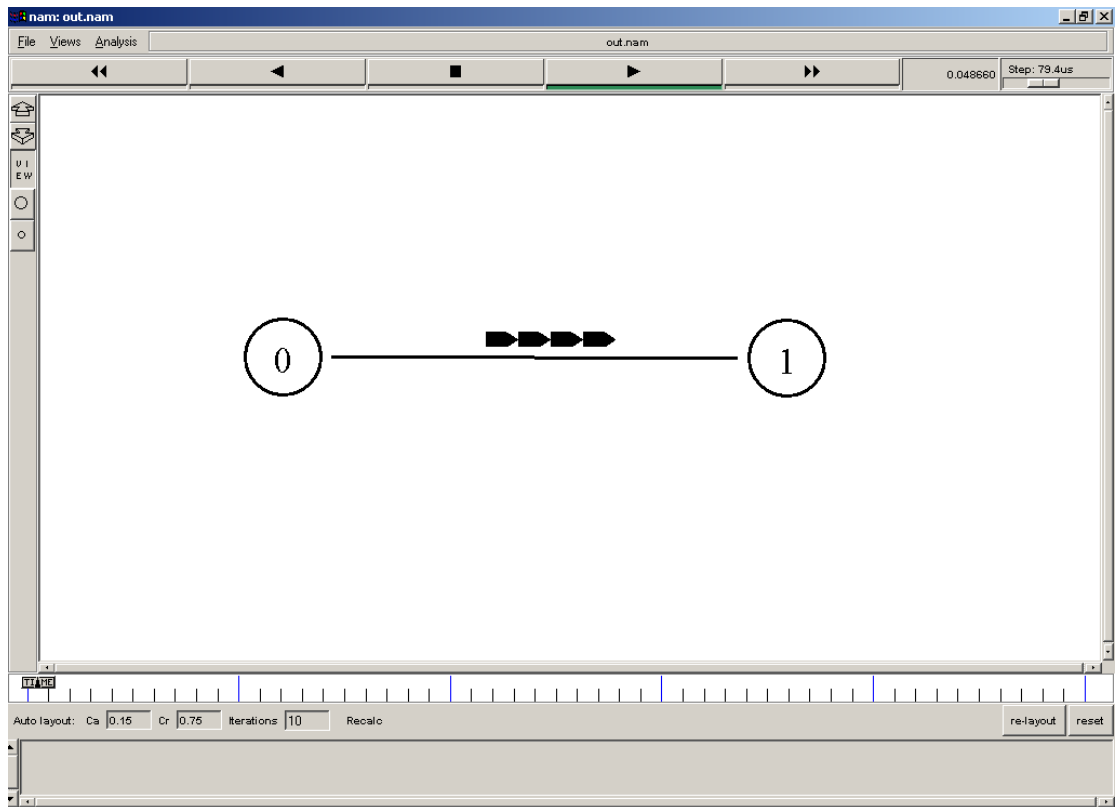
Παρακάτω εικονίζονται μερικά στιγμιότυπα από το animation:



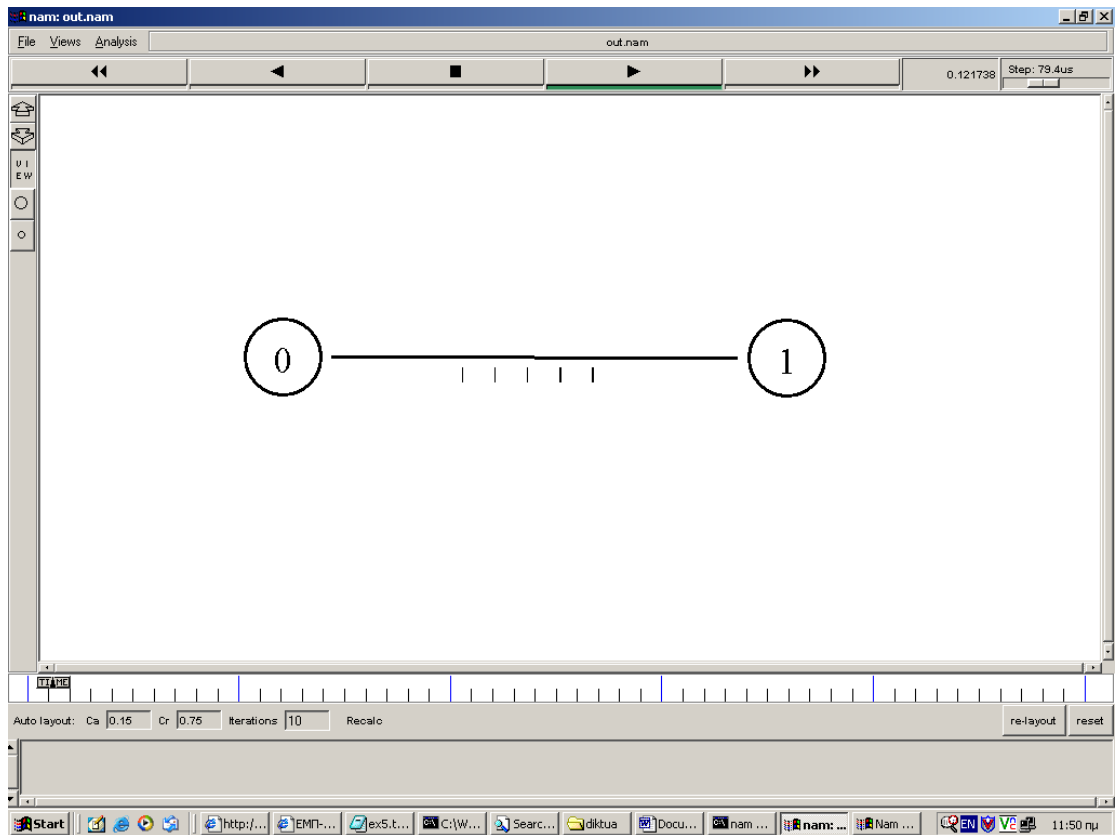
Σχήμα2:Στιγμιότυπο από την αρχή του animation,όπου μεταδίδεται πρώτα ένα πακέτο πληροφορίας.



Σχήμα3:Εδώ εικονίζεται το πακέτο επιβεβαίωσης (acknowledgement),το οποίο βασίζεται στο πρωτόκολλο του FTP και κατ'επέκταση του go back



Σχήμα4:Στιγμιότυπο όπου εικονίζονται τέσσερα πακέτα πληροφορίας ,δηλαδή για μέγεθος παραθύρου 4



Σχήμα5:Εδώ εικονίζεται το στιγμιότυπο που στέλνονται πέντε πακέτα επιβεβαίωσης για μέγεθος παραθύρου 5

2 Ανάλυση αρχείου ίχνους(trace file):

Καθώς δημιουργήσαμε το αρχείο ίχνους ,το οποίο βασίστηκε στα δεδομένα της προσομοίωσης ,οπού χρησιμοποιήθηκε ο κώδικας που παρουσιάστηκε προωύτερα.. Αυτά τα αρχεία έχουν μία συγκεκριμένη μορφή η οποία παρουσιάζει όλα τα απαραίτητα στοιχεία για περαιτέρω ανάλυση .Σε αυτά τα αρχεία περιέχονται πληροφορίες και για τα acknowledgement files καθώς και τις διευθύνσεις αποστολέα αλλά και προορισμού (IP.port).Αυτό έγινε με την βοήθεια της script γλώσσας awk.Με την βοήθεια αυτής μπορέσαμε να υπολογίσουμε το πλήθος των πακέτων και της ποσότητας των δεδομένων που ελήφθησαν. Ο κώδικας που χρησιμοποιήθηκε ήταν ο ακόλουθος::

```
BEGIN {  
  data=0;  
  packets=0;  
}  
/^r/&&/tcp/  
{ data+=$6;  
  packets++;  
}  
END {  
  printf("Total Data received\t: %d Bytes\n", data);  
  printf("Total Packets received\t: %d\n", packets);  
}
```

Συνακόλουθα εκτελούμε το πρόγραμμα όπως προστάζει το φυλλάδιο της εργαστηριακής άσκησης. Εκτελούμε το πρόγραμμα ανάλυσης ,όπου το εφοδιάζουμε με τα απαραίτητα στοιχεία που απαιτούνται ,δηλαδή το αρχείο ίχνους, μετέπειτα εκτυπώνεται ο αριθμός των πακέτων και το πλήθος των δεδομένων. Συγκεκριμένα τυπώθηκαν τα εξής

```
Total Data received   : 1192000 Bytes  
Total Packets received : 1192
```

Αυτά τα στοιχεία είναι πάρα πολύ σημαντικά καθώς πάνω σε αυτά θα βασισθεί η μελέτη της απόδοσης του go back N,και η οποία ακολουθεί.

3 Μελέτη απόδοσης go back N:

Σε αυτό το τμήμα της μελέτης θα ευρεθεί η απόδοση του υπό μελέτη πρωτοκόλλου.Συνακόλουθα θα γίνει προσπάθεια αιτιολόγησης της αποδόσεως του πρωτοκόλλου.

α) Ποιος είναι ο αριθμός των πακέτων που παραλήφθηκαν , πόσα δεδομένα παραλήφθηκαν από τον παραλήπτη κατά τη διάρκεια της προσομοίωσης ; Σε πόσο χρόνο απεστάλησαν αυτά τα δεδομένα , ποιος ο ρυθμός μετάδοσης δεδομένων , και ποια η χρησιμοποίηση του καναλιού ;

Με βάση το πρόγραμμα που χρησιμοποιήθηκε προωύτερα .

- ❖ Βρέθηκε ότι παραλήφθηκαν 1192 πακέτα
- ❖ Συνολικά 1192000 Bytes=1192000*8 bits=9536000 bits.
- ❖ Γνωρίζουμε ότι είχαμε ζεύξη για 5 δευτερόλεπτα ,οπότε για να βρούμε τον καθαρό ρυθμό μετάδοσης, αρκεί να διαιρέσουμε την συνολική ποσότητα των bits δια τον χρόνο που απαιτήθηκε για να σταλεί αυτό το μέγεθος .Ο λόγος για τον οποίο ορίζουμε τον χρόνο ίσο με 5 δευτερόλεπτα είναι στο ότι τόσο ορίσαμε τον χρόνο προσομοίωσης με την βοήθεια της script γλώσσας. Συνεπώς έχουμε:

$$\text{Καθαρος_ρυθμος_μεταδ.} = \frac{1192000 * 8}{5} \cdot \frac{\text{bits}}{\text{sec}} = 1.907.200 \frac{\text{bits}}{\text{sec}} \cong 1.9 \text{Mb} / \text{sec}$$

Οπότε παρατηρούμε ότι ο ρυθμός μεταφοράς δεδομένων είναι ιδιαίτερα χαμηλός και αυτό άλλωστε το επιβεβαιώσαμε και με το animation,μέσω του ότι καθώς στέλνονταν τα πακέτα παρατηρούσαμε αρκετά διαστήματα όπου ο διαυλος μετάδοσης παρέμενε σχετικά ανεκμετάλλευτος .Για να βρούμε την χρησιμοποίηση του καναλιού θα βρούμε τι ποσοστό αποτελεί ο καθαρός ρυθμός ως προς το μέγιστο εύρος μετάδοσης του καναλιού .Άρα έχουμε:

$$\text{❖ } n\%(\text{χρησιμοποίηση_διαυλου}) = \frac{1.9 \text{Mb} / \text{sec}}{10 \text{Mb} / \text{sec}} = 19\% . \text{Παρατηρούμε ότι όντως}$$

πρόκειται για μία σχετικά φτωχή εκμετάλλευση του καναλιού και αυτό οφείλεται στο ότι το χρησιμοποιούμενο μέγεθος παραθύρου είναι το βέλτιστο δυνατό, διότι τα πακέτα επιβεβαίωσης καθυστερούν πολύτιμο χρόνο από την γραμμή μας.

(β) Με βάση την παραπάνω εξίσωση , υπολογίστε την θεωρητική τιμή της χρησιμοποίησης του καναλιού , θεωρώντας ότι το μέγεθος των πακέτων αυξάνεται κατά 40 byte λόγω επικεφαλίδων TCP και IP, και ότι οι επαληθεύσεις (ACK) έχουν μέγεθος 40 byte. Ισχύει η εξίσωση ; Αν όχι , πού οφείλεται η απόκλιση ;

Μας δίδεται στο φυλλάδιο της εργαστηριακής άσκησης ο ακόλουθος τύπος ο οποίος μας δείχνει την θεωρητική τιμή χρησιμοποίησης του καναλιού:

$$\text{❖ } n = \min \left\{ \frac{W * \text{TRANSP}}{S}, 1 \right\} , \text{μέσω αυτού θα προσπαθήσουμε να βρούμε την}$$

απόδοση. Έτσι θα κάνουμε τις ακόλουθες αντικαταστάσεις καθώς με βάση τα δεδομένα της άσκησης ισχύει ότι :

$$\text{TRANSP} = \frac{\text{μεγεθος_πακετου}}{\text{ρυθμο_μεταδοσης}} = \frac{1000 \text{bytes}}{10 \text{Mbits} / \text{sec}} \Rightarrow$$

$$\text{TRANSP} = \frac{1000 * 8 \text{bits}}{10 * 10^6 \text{bits} / \text{sec}} = 0,0008 \text{sec}$$

Όμοια το PROP είναι γνωστό από την αρχή καθώς το ορίσαμε ίσο με 10ms=0.01sec.Όμοια για τον χρόνο επαλήθευσης

$$\text{TRANSA} = \frac{\text{μεγεθος_πακετου_επαληθευσης}}{\text{ρυθμο_μεταδοσης}} = \frac{40 \text{bytes}}{10 \text{Mbits} / \text{sec}} \Rightarrow$$

$$\text{TRANSA} = \frac{40 * 8 \text{bits}}{10 * 10^6 \text{bits} / \text{sec}} = 0,000032 \text{sec}$$

Συνεπώς έχοντας όλα αυτά γνωστά τα αντικαθιστούμε στο S και κατ' επένταση στον δοθέντα τύπο έχουμε ότι: $n=0,1920$. Παρατηρούμε ότι αυτές οι τιμές είναι σχεδόν ίσες καθώς λογικό είναι να έχουμε μια πολύ μικρή διαφορά, η οποία όμως δεν μας ανησυχεί ιδιαίτερα καθώς οφείλεται σε στοιχεία που έχουν να κάνουν με την προσομοίωση που έγινε. Αλλά και ίσως στο γεγονός ότι κατά την αποστολή του πακέτου δεν συνυπολογίστηκε η επικεφαλίδα του πακέτου (δηλαδή υπολογίσαμε για 1000bytes αντί για 1040bytes). Πάντως δεν επηρέασε τα αποτελέσματα, καθώς οι επικεφαλίδες ήταν αρκετά πιο μικρές από το πακέτο που εστάλη κάθε φορά.

(γ) Διατηρώντας σταθερό το μέγεθος του παραθύρου, αλλάζετε το μήκος των πακέτων ώστε η απόδοση του πρωτοκόλλου να λάβει τη μέγιστη τιμή της.

Παρατηρήσαμε ότι η απόδοση χρήσης του καναλιού είναι αρκετά μικρή για αυτό θα το επιχειρήσουμε να το λύσουμε αυτό σε αυτή την φάση με την χρήση του δοθέντος θεωρητικού τύπου. Δηλαδή:

$$n = \min \left\{ \frac{W * TRANSP}{S}, 1 \right\} = 1.$$

Βέβαια καθώς αλλάζει μέγεθος το πακέτο, τότε επηρεάζεται άμεσα ο χρόνος αποστολής, άρα θα μεταβληθούν λίγο τα πράγματα, οπότε ισοδύναμα θα έχουμε:

$$(packet_size) * (w - 1) = 2 * PROP * B + (ACK_size) * 8$$

Από την παραπάνω σχέση προκύπτει ότι: **packet size = 6260 Byte**. Για αυτό το λόγο κάνουμε τις απαραίτητες αλλαγές στο script παραπάνω και καθώς το τρέχουμε, συνακόλουθα χρησιμοποιούμε την γλώσσα awk για να υπολογίσουμε τον αριθμό των ληφθέντων bytes και πακέτων:

Total Data received : 6184880 Bytes

Total Packets received: 988. Συνεπώς εάν αντικαταστήσουμε στον δοθέντα τύπο τα δεδομένα που τώρα συλλέξαμε θα έχουμε: (Πάλι ο χρόνος αποστολής ήταν 5 δευτερόλεπτα)

$$καθαρος_ρυθμος_μεταδοσης = \frac{6184880 * 8}{5} \cdot \frac{bits}{sec} = 9.895.808 \frac{bits}{sec} \cong 9.9 Mb/sec$$

Οπότε κατά αντιστοιχία με ότι κάναμε πρωύτερα τώρα θα έχουμε σχετικά με την απόδοση:

$$n \% \cong \frac{9.9 Mb/sec}{10 Mb/sec} \cong 99 \% \Rightarrow 100 \%$$

(δ) Αυξήστε στη συνέχεια στο δεκαπλάσιο την ταχύτητα μετάδοσης και ρυθμίστε το μέγεθος του παραθύρου, ώστε και πάλι η απόδοση να λάβει τη μέγιστη τιμή της.

Σε αυτό το ερώτημα γνωρίζουμε τι η ταχύτητα μετάδοσης δεκαπλασιάζεται οπότε τώρα θα έχουμε 100Mbps/sec, αυτό επηρεάζει άμεσα τους χρόνους μετάδοσης της επαλήθευσης, καθώς και το ίδιο το πακέτο που αποστέλλεται κάθε φορά. Συνεπώς απαιτούμε

$$πάλι \quad n = \min \left\{ \frac{W * TRANSP}{S}, 1 \right\} = 1 \text{ και ισοδύναμα}$$

$(packet_size) * (w - 1) = 2 * PROP * B + (ACK_size) * 8$, άρα από αυτήν την σχέση έπεται ότι $w = 38.9 \rightarrow w = 39$. Άρα το βέλτιστο μέγεθος παραθύρου είναι 39 πακέτα.

(ε) Εφαρμόστε τώρα το πρωτόκολλο με το παράθυρο που βρήκατε στο ερώτημα (γ) σε ζεύξη με δεκαπλάσια καθυστέρηση διάδοσης ; Ποια είναι η απόδοση του πρωτοκόλλου στη νέα αυτή ζεύξη ;

Σε αυτό το τελευταίο κομμάτι της εργασίας θα έχουμε ότι θα μεταβάλλουμε την καθυστέρηση διάδοσης κατά δέκα φορές .άρα θα επηρεαστεί άμεσα ο χρόνος PROP κατά κύριο λόγο,οπότε πιστεύουμε πως θα επηρεαστεί και η απόδοση του καναλιού .Συνεπώς έχοντας αυτό κατά νου και εφαρμόζοντας το πρωτόκολλο, θα έχουμε μέγεθος παραθύρου $w=39$ packets,με εύρος ζώνης 100 Mbs/sec.Για το PROP ισχύει ότι είναι ίσο με 100 msec .Άρα εφαρμόζοντας την δοθείσα από το φυλλάδιο σχέση για την απόδοση και κάνοντας αντικατάσταση έχουμε : $n=0,097 = 9,7\%$