

**ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ**  
**ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ**  
**ΥΠΟΛΟΓΙΣΤΩΝ**



**ΔΙΚΤΥΑ ΕΠΙΚΟΙΝΩΝΙΩΝ**

(2018-2019)

*5<sup>η</sup> ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΝΑΦΟΡΑ*

Ονοματεπώνυμο : Χρήστος Τσούφης

A.M. : 03117176

## 1. Εισαγωγή

Σε αυτή την άσκηση μελετάται η επίδοση του MAC πρωτοκόλλου IEEE 802.3. Σε γλώσσα tcl γράφεται ένα script για τον NS2, το οποίο θα προσομοιώνει ένα τοπικό δίκτυο IEEE 802.3 με ένα πλήθος σταθμών συνδεδεμένων σε αυτό. Οι σταθμοί θα παράγουν κίνηση, η οποία θα διέρχεται μέσα από το τοπικό δίκτυο. Από την προσομοίωση θα προκύψει αρχείο, από την ανάλυση του οποίου θα προκύψουν κάποια μέτρα επίδοσης του δικτύου. Συγκεκριμένα θα μετρηθεί το πλήθος των πακέτων και η ποσότητα των δεδομένων που μεταδόθηκαν επιτυχώς κατά τη διάρκεια της προσομοίωσης, καθώς και η μέση καθυστέρηση των πακέτων. Από αυτά τα δεδομένα θα γίνει υπολογισμός της χρησιμοποίησης του καναλιού και σύγκριση με τα θεωρητικά αναμενόμενα αποτελέσματα.

Στις ενότητες παρουσιάζονται οι λεπτομέρειες του σεναρίου προσομοίωσης, η διαδικασία ανάλυσης των αποτελεσμάτων της προσομοίωσης και η ανάλυση της διαδικασίας σύγκρισης θεωρητικών και πειραματικών αποτελεσμάτων.

## 2. Σενάριο προσομοίωσης

### 2.1 Παράμετροι Προσομοίωσης

Αρχικά δημιουργείται ένα τοπικό δίκτυο IEEE 802.3. Σημαντικότερες παράμετροι είναι: η διάρκεια της προσομοίωσης *opt(stopsim)*, το πλήθος των σταθμών στο LAN *opt(node)*, η καθυστέρηση διάδοσης *opt(delay)* και το μέγεθος των πακέτων *opt(packetsize)*. Αυτές οι παράμετροι πρέπει να αλλάζουν σε κάθε προσομοίωση.

*Πηγαίος κώδικας:*

```
#### Αρχείο προσομοίωσης για μελέτη επίδοσης πρωτοκόλλου IEEE 802.3 για τοπικά
#### δίκτυα. Το σενάριο αποτελείται από έναν αριθμό σταθμών συνδεδεμένων σε ένα
#### τοπικό δίκτυο (LAN) 802.3. Η γεννήτρια κίνησης είναι CBR (σταθερού ρυθμού) με
#### τέτοιο ρυθμό ώστε πάντα να υπάρχουν πακέτα για μετάδοση.
####
#### 0   1   2
#### |   |   |
#### =====
#### |   |   |
#### ... n-2 n-1
#### Τα αποτελέσματα καταγράφονται στα αρχεία lab5.nam (NAM) και lab5.tr (trace)
```

```
set opt(nam) "lab5.nam" ;# Αρχείο NAM
set opt(tr) "lab5.tr" ;# Αρχείο trace
```

```
set opt(seed) 0 ;# Παράμετρος για τη γεννήτρια τυχαίων αριθμών
set opt(starttraf) 0.25 ;# Έναρξη παραγωγής δεδομένων(sec)
set opt(stoptraf) 2.0 ;# Λήξη παραγωγής δεδομένων (sec)
set opt(stopsim) 3.0 ;# Διάρκεια προσομοίωσης (sec)
set opt(node) 18 ;# Αριθμός σταθμών στο LAN
set opt(qsize) 250 ;# Μέγεθος ουράς σε κάθε σταθμό
set opt(bw) 10000000 ;# Ρυθμός μετάδοσης (bps)
set opt(delay) 0.0000256 ;# Καθυστέρηση μετάδοσης (sec)
set opt(packetsize) 1024 ;# Μέγεθος πακέτου (byte)
```

```
set opt(rate) [expr 4*$opt(bw)/$opt(node)] ;# Ρυθμός παραγωγής δεδομένων κάθε σταθμού (bps)
```

```
### Παράμετροι του LAN  
set opt(ll) LL  
set opt(ifq) Queue/DropTail  
set opt(mac) Mac/802_3  
set opt(chan) Channel
```

## 2.2 Τοπολογία

Η τοπολογία της προσομοίωσης αποτελείται από έναν αριθμό κόμβων συνδεδεμένων σε ένα τοπικό δίκτυο. Με την ακόλουθη διαδικασία δημιουργείται αρχικά ένας αριθμός από κόμβους οι οποίοι τοποθετούνται σε λίστα. Στη συνέχεια, ενώνονται όλοι οι κόμβοι της λίστας που δημιουργήθηκε με ένα τοπικό δίκτυο (LAN). Οι παράμετροι του τοπικού δικτύου είναι στις μεταβλητές που ορίστηκαν αρχικά.

*Πηγαίος κώδικας:*

```
# Δημιουργία τοπολογίας δικτύου  
proc create-topology {} {  
    global ns opt  
    global lan node  
    set num $opt(node)
```

```
### Δημιουργία σταθμών του LAN  
for {set i 0} {$i < $num} {incr i} {  
    set node($i) [$ns node]  
    lappend nodelist $node($i)  
}
```

```
### Δημιουργία του LAN  
set lan [$ns newLan $nodelist $opt(bw) $opt(delay) \  
-llType $opt(ll) -ifqType $opt(ifq) \  
-macType $opt(mac) -chanType $opt(chan)]  
}
```

## 2.3 Γεννήτρια Κίνησης

Εδώ μελετάται η επίδοση του MAC πρωτοκόλλου IEEE 802.3 υπό συνθήκες κορεσμού, δηλαδή, όταν όλοι οι χρήστες έχουν πακέτα για μετάδοση. Για τον λόγο αυτόν, σε κάθε σταθμό του δικτύου προσαρμόζεται γεννήτρια κίνησης CBR (Constant Bit Rate) με ρυθμό `opt(rate)`, ο οποίος είναι τετραπλάσιος του ρυθμού μετάδοσης του καναλιού προς το συνολικό πλήθος των κόμβων. Για τη μεταφορά της κίνησης χρησιμοποιείται το πρωτόκολλο UDP. Η μετάδοση των πλαισίων ξεκινά τη χρονική στιγμή `opt(starttraf)` και ολοκληρώνεται τη χρονική στιγμή `opt(stoptraf)` για όλους του κόμβους.

*Πηγαίος κώδικας:*

```
### Δημιουργία κίνησης μεταξύ των σταθμών
```

```

proc create-connections { } {
    global ns opt
    global node udp sink cbr

    for {set i 1} {$i < $opt(node)} {incr i} {
        set udp($i) [new Agent/UDP]
        $udp($i) set packetSize_ $opt(packetSize)
        $ns attach-agent $node($i) $udp($i)
        set sink($i) [new Agent/Null]
        $ns attach-agent $node(0) $sink($i)
        $ns connect $udp($i) $sink($i)
        set cbr($i) [new Application/Traffic/CBR]
        $cbr($i) set rate_ $opt(rate)
        $cbr($i) set packetSize_ $opt(packetSize)
        $cbr($i) set random_ 1
        $cbr($i) attach-agent $udp($i)
        $ns at $opt(starttraf) "$cbr($i) start"
        $ns at $opt(stoptraf) "$cbr($i) stop"
    }
}

```

## 2.4 Δημιουργία και αποθήκευση αρχείου animation

Για να είναι δυνατή η αναπαράσταση του σεναρίου της προσομοίωσης με το πρόγραμμα Network Animator (NAM), πρέπει να ανοιχτεί το αρχείο όπου θα αποθηκεύονται οι πληροφορίες του animation, να δοθεί οδηγία στον προσομοιωτή να αποθηκεύσει την πληροφορία για το animation σε αυτό το αρχείο και, όταν ολοκληρωθεί η προσομοίωση, να κλείσει το αρχείο.

*Πηγαίος κώδικας:*

```

### Δημιουργία αρχείου NAM
proc create-nam-trace { } {
    global ns opt
    set namf [open $opt(nam) w]
    $ns namtrace-all $namf
    return $namf
}

```

## 2.5 Δημιουργία και αποθήκευση αρχείου ίχνους

Για να είναι δυνατή η παραπέρα επεξεργασία των αποτελεσμάτων της προσομοίωσης δημιουργείται ένα αρχείο ίχνους (trace file). Σε αυτό το αρχείο αποθηκεύονται όλα τα γεγονότα που συνέβησαν κατά τη διάρκεια της προσομοίωσης. Συγκεκριμένα για κάθε δημιουργία, αποστολή, προώθηση, απόρριψη, επιτυχημένη λήψη κάποιου πακέτου προστίθεται μια γραμμή στο αρχείο ίχνους, η οποία περιγράφει αναλυτικά το γεγονός που συνέβη.

Για να γίνει η καταγραφή των γεγονότων της προσομοίωσης πρέπει πρώτα να ανοιχθεί ένα αρχείο για εγγραφή, να δοθεί οδηγία στον προσομοιωτή να αποθηκεύσει την πληροφορία για

τα γεγονότα που θα συμβούν σε αυτό το αρχείο και, όταν ολοκληρωθεί η προσομοίωση, να κλείσει το αρχείο.

*Πηγαίος κώδικας:*

```
### Δημιουργία αρχείου trace
proc create-trace {} {
    global ns opt
    set trf [open $opt(tr) w]
    $ns trace-all $trf
    return $trf
}
```

```
### Διαδικασία τερματισμού
proc finish {} {
    global ns trf namf
    $ns flush-trace
    close $trf
    close $namf
    exit 0
}
```

## **2.6 Εκτέλεση του σεναρίου**

Τέλος ορίζονται τα γεγονότα της προσομοίωσης.

*Πηγαίος κώδικας:*

```
## MAIN ##
set ns [new Simulator]
set trf [create-trace]
set namf [create-nam-trace]
create-topology
create-connections
$ns at $opt(stopsim) "finish"
$ns run
```

## **3. Ανάλυση αρχείου ίχνους (trace file)**

Μετά τη δημιουργία του σεναρίου προσομοίωσης, της εκτέλεσής του και τη δημιουργία των αρχείων αποτελεσμάτων, τα αρχεία αυτά πρέπει να αναλυθούν ώστε να παρθούν οι κατάλληλες πληροφορίες.

### **3.1 Μορφή αρχείου ίχνους (trace file)**

Το αρχείο lab5.tr που δημιουργήθηκε στο προηγούμενως περιέχει πληροφορίες της μορφής:

```
+ 0.253998 13 18 cbr 1024 ----- 13 13.0 0.12 1 18
h 0.254408 9 18 cbr 1024 ----- 9 9.0 0.8 1 19
```

Κάθε γραμμή του αρχείου αυτού αντιστοιχεί σε ένα γεγονός που συνέβη κατά τη διάρκεια της προσομοίωσης.

### 3.2 Ανάλυση με τη γλώσσα awk

Η γλώσσα awk είναι σχεδιασμένη ώστε να επιτρέπει την εύκολη ανάλυση αρχείων με δεδομένα. Ένα πρόγραμμα awk αποτελείται από τρία τμήματα. Το πρώτο τμήμα του προγράμματος ορίζεται με την εντολή BEGIN { } και περιλαμβάνει όλες τις εντολές που θα γίνουν μία φορά, κατά την εκκίνηση της του προγράμματος. Εδώ μπορούν να αρχικοποιηθούν μεταβλητές, να ανοιχτούν αρχεία, κ.λπ.

Το δεύτερο τμήμα του προγράμματος αποτελείται από ένα σύνολο από κανόνες που θα εκτελεστούν για κάθε γραμμή του αρχείου εισόδου. Αυτοί οι κανόνες αποτελούνται από δύο κομμάτια. Το πρώτο κομμάτι ορίζει σε ποιες γραμμές του αρχείου εισόδου αναφέρεται ο κανόνας, και το δεύτερο ορίζει ποιες λειτουργίες θα πραγματοποιηθούν για αυτές τις γραμμές. Το τρίτο τμήμα ορίζεται από την εντολή END{ } και περιλαμβάνει τις λειτουργίες που θα πραγματοποιηθούν αφού διαβαστεί ολόκληρο το αρχείο εισόδου, στο τέλος της εκτέλεσης του προγράμματος.

### 3.3 Μέτρηση πλήθους πακέτων, μέσης καθυστέρησης και ποσότητας δεδομένων

Για τον υπολογισμό της χρησιμοποίησης (utilisation) του καναλιού, όταν χρησιμοποιείται το πρωτόκολλο IEEE 802.3, πρέπει να μετρηθεί η ποσότητα των δεδομένων που ελήφθησαν κατά τη διάρκεια της προσομοίωσης. Στη συνέχεια αυτός ο αριθμός θα διαιρεθεί με το συνολικό χρόνο μετάδοσης των εν λόγω δεδομένων, ώστε να υπολογιστεί ο ρυθμός διέλευσης δεδομένων πάνω από το κανάλι (throughput). Τέλος, η παραπάνω τιμή θα διαιρεθεί με τον ονομαστικό ρυθμό μετάδοσης του καναλιού ώστε να προκύψει η χρησιμοποίηση (utilisation). Πρέπει να εντοπισθούν οι γραμμές του αρχείου ίχνους που φανερώνουν ορθή λήψη πακέτου cbr. Αυτές οι γραμμές αρχίζουν με τον χαρακτήρα "r" και περιλαμβάνουν την λέξη cbr. Για κάθε τέτοια γραμμή που εντοπίζεται, η μεταβλητή total\_pkts\_recv (αριθμός πακέτων που ελήφθησαν) αυξάνεται κατά ένα.

Η καθυστέρηση κάθε πακέτου προκύπτει αν αφαιρεθεί ο χρόνος λήψης κάθε πακέτου από τον χρόνο αποστολής του. Ως χρόνος αποστολής ενός πακέτου θεωρείται η χρονική στιγμή κατά την οποία το πακέτο εξέρχεται από την ουρά αναμονής στον κόμβο αποστολής.

Ο χρόνος αποστολής αφαιρείται από τον χρόνο λήψης και προκύπτει η καθυστέρηση του κάθε πακέτου. Αθροίζοντας όλες τις καθυστερήσεις και διαιρώντας προς τον αριθμό των πακέτων προκύπτει η μέση καθυστέρηση.

*Πηγαίος κώδικας:*

```
BEGIN {
sum_delay = 0;
bufferspace = 100000;
total_pkts_sent = 0;
total_pkts_recv = 0;
total_pkts_dropped = 0;

# Do we need to fix the decimal mark?
if (sprintf(sqrt(2)) ~ /\./) dmfix = 1;
}

{
# Apply dm fix if needed
if (dmfix) sub(/\./, "", $0);
```

```

}

/^h/&&/cbr/ {
total_pkts_sent++;
}

/^d/&&/cbr/ {
total_pkts_dropped++;
}

/^-/&&/cbr/ {
sendtimes[$12%bufferspace] = $2;
}

/^r/&&/cbr/ {
total_pkts_rcv++;
sum_delay += $2 - sendtimes[$12%bufferspace];
}

END {
printf("Total Packets sent\t: %d\n", total_pkts_sent);
printf("Total Packets received\t: %d\n", total_pkts_rcv);
printf("Total Packets dropped\t: %d\n", total_pkts_dropped);
printf("Average Delay\t: %f sec\n", (1.0 * sum_delay)/ total_pkts_rcv);
}

```

### 3.4 Εκτέλεση του προγράμματος ανάλυσης

Για την εκτέλεση προγραμμάτων awk χρησιμοποιείται ο διερμηνέας (interpreter) awk με παραμέτρους το όνομα του αρχείου που περιγράφει τις διαδικασίες της ανάλυσης και το όνομα του αρχείου που περιλαμβάνει τα δεδομένα που θα αναλυθούν. Εάν ο παραπάνω κώδικας έχει αποθηκευτεί στο αρχείο lab5.awk και τα δεδομένα βρίσκονται στο αρχείο lab5.tr, τότε εκτελείται η εντολή: `awk95.exe -f lab5.awk < lab5.tr`

Αυτή η εντολή θα εκτυπώσει στην οθόνη τον αριθμό των πακέτων και το πλήθος των δεδομένων που ελήφθησαν.

Για να αποθηκευτούν τα αποτελέσματα της ανάλυσης στο αρχείο results.txt εκτελείται η εντολή: `awk95.exe -f lab5.awk < lab5.tr > results.txt`

Σε περιβάλλον MS Windows ο διερμηνέας awk χρησιμοποιεί ως σύμβολο υποδιαστολής (decimal mark), το σύμβολο που είναι καθορισμένο από τα regional settings της εγκατάστασης. Σε πολλές περιπτώσεις αυτό το σύμβολο είναι το κόμμα (,) και όχι η τελεία (.), με αποτέλεσμα το awk να μην μπορεί να χειριστεί σωστά πραγματικούς αριθμούς από το αρχείο εισόδου. Για να αντιμετωπιστεί το πρόβλημα αυτό, πρέπει να προστεθεί ο ακόλουθος κανόνας ο οποίος αντικαθιστά όπου χρειάζεται τις τελείες με κόμματα.

*Πηγαίος κώδικας:*

```

{
# Apply dm fix if needed
if (dmfix) sub(/./, ",", $0); }

```

#### 4. Μελέτη απόδοσης τοπικού δικτύου IEEE 802.3

Με βάση την προσομοίωση που εκτελέσατε στην ενότητα 2 και την ανάλυση της ενότητας 3 να επαληθεύσετε κατά πόσον ισχύει ή όχι η εξίσωση:  $\eta = \frac{P}{P+2\tau e}$ .

Υπολογίζεται ότι:  $P = \frac{\text{packet\_size}}{\text{bitrate}} = \frac{1024 \text{ bytes}}{10 \text{ Mbps}} = \frac{8.192 \text{ bits}}{10.000.000 \frac{\text{bits}}{\text{sec}}} \Rightarrow P = 0,0008192 \text{ sec}$


Οπότε:  $\eta_{\text{θεωρ}} = \frac{0,0008192 \text{ sec}}{0,0008192 \text{ sec} + 2 * 0,0000256 \text{ sec} * 2,718281828} = \frac{0,0008192 \text{ sec}}{0,0,0009583760295936 \text{ sec}} = 0,854779 \Rightarrow \eta_{\text{θεωρ}} = 85,4779 \%$

Επίσης η απόδοση μπορεί να υπολογιστεί και πειραματικά με βάση τον τύπο:

$$\eta_{\text{πειρ}} = \frac{\text{bitrate\_actual}}{\text{bitrate}}$$

όπου  $\text{bitrate\_actual} = \frac{\text{total data received}}{\text{total transmission time}} = \frac{\text{total packets received} * \text{packet size}}{\text{total transmission time}}$

Η ανάλυση του awk, δίνει τα εξής αποτελέσματα:

 results - Σημειωματάριο

```
Αρχείο Επεξεργασία Μορφή Προβολή Βοήθεια
Total Packets sent      : 8058
Total Packets received  : 2754
Total Packets dropped    : 5304
Average Delay           : 0,002406 sec
Last packet received at : 2,711561 sec
Total Transmission time : 2,46156 sec
```

Οπότε,

$$\text{bitrate\_actual} = \frac{2.754 * 1.024 \text{ bytes}}{2,46156 \text{ sec}} = \frac{2.820.096 \text{ bytes}}{2,46156 \text{ sec}} = \frac{22.560.768 \text{ bits}}{2,46156 \text{ sec}} = 9.165.231,8042 \frac{\text{bits}}{\text{sec}}$$

Επομένως,

$$\eta_{\text{πειρ}} = \frac{\text{bitrate\_actual}}{10 \text{ Mbps}} = \frac{9.165.231,8042}{10.000.000} = 0,9165 \Rightarrow \eta_{\text{πειρ}} = 91,65\%$$

Παρατήρηση: ο πειραματικός ρυθμός μετάδοσης είναι σημαντικά μεγαλύτερος από τον θεωρητικό. Αυτό συμβαίνει διότι η κίνηση θεωρητικά ακολουθεί κατανομή Poisson, πράγμα που δεν συμβαίνει στην πράξη, και οι περισσότερες θεωρητικές αναλύσεις έχουν γίνει σε αφύσικα υψηλά φορτία.

##### 4.1 Επίδραση μεγέθους πλαισίων

Για την επαλήθευση της θεωρητικής εξίσωσης να πραγματοποιηθεί σειρά προσομοιώσεων με τις εξής παραμέτρους:

- 18 σταθμοί συνδεδεμένοι πάνω από ένα τοπικό δίκτυο IEEE 802.3.
- Ρυθμός μετάδοσης στο LAN 10 Mbps.
- Καθυστέρηση διάδοσης 25,6  $\mu\text{s}$ .
- Ρυθμός μετάδοσης δεδομένων κάθε σταθμού ίσος με το τετραπλάσιο του ονομαστικού ρυθμού μετάδοσης του καναλιού προς το πλήθος των σταθμών.

Η σειρά προσομοιώσεων πρέπει να πραγματοποιηθεί με μεταβαλλόμενο μέγεθος πακέτου από 64 Byte (ελάχιστο μέγεθος πλαισίου) μέχρι 1500 Byte (μέγιστο μέγεθος πλαισίου). Πρέπει να πραγματοποιηθούν τουλάχιστον δέκα προσομοιώσεις.

α) Για αυτές τις προσομοιώσεις να γίνει κοινή γραφική παράσταση της θεωρητικής και πειραματικής τιμής της χρησιμοποίησης του καναλιού (utilisation) σε σχέση με το μέγεθος του πλαισίου. Επίσης να γίνει γραφική παράσταση της μέσης καθυστέρησης που παρατηρείται



με χρήση του πρωτοκόλλου IEEE 802.3 συναρτήσει του μεγέθους των πλαισίων, καθώς και του χρόνου τελευταίας επιτυχημένης λήψης πλαισίου συναρτήσει του μεγέθους των πλαισίων. Για να προσδιορίσετε το τελευταίο, τροποποιείστε κατάλληλα τον κώδικα awk και παραθέστε τον στην αναφορά σας.

Πηγαίος κώδικας:

```
BEGIN {
sum_delay = 0;
bufferspace = 100000;
total_pkts_sent = 0;
total_pkts_recv = 0;
total_pkts_dropped = 0;

# Do we need to fix the decimal mark?
if (sprintf(sqrt(2)) ~ /\./) dmfix = 1;
}

{
# Apply dm fix if needed
if (dmfix) sub(/\./, "", $0);
}

/^h/&&/cbr/ {
total_pkts_sent++;
}

/^d/&&/cbr/ {
total_pkts_dropped++;
}

/^-/&&/cbr/ {
sendtimes[$12%bufferspace] = $2;
}

/^r/&&/cbr/ {
packet_size = $6;
last_ts = $2;
total_pkts_recv++;
sum_delay += $2 - sendtimes[$12%bufferspace];
}

{
tr_time = last_ts - 0.25;
th_ut = 100 * 8 * packet_size / (8 * packet_size + 1391.76);
exp_ut = 100 * (total_pkts_recv * packet_size * 8) / (tr_time * 10000000);
}
```

```

END {
printf("Packet size\t\t\t: %d\n", packet_size);
printf("Theoretical Utilization\t\t: %f\n", th_ut);
printf("Experimental Utilization\t: %f\n", exp_ut);
printf("Divergence\t\t\t: %f\n", (th_ut - exp_ut) * 100/ th_ut);
printf("Total Packets sent\t: %d\n", total_pkts_sent);
printf("Total Packets received\t: %d\n", total_pkts_recv);
printf("Total Packets dropped\t: %d\n", total_pkts_dropped);
printf("Average Delay\t\t: %f sec\n", (1.0 * sum_delay)/ total_pkts_recv);
printf("Last packet received at\t: %s sec\n", last_ts);
printf("Total Transmission time\t: %s sec\n", last_ts - 0.25);
}

```

#### results - Σημειωματάριο

Αρχείο	Επεξεργασία	Μορφή	Προβολή	Βοήθεια
Packet size				: 64
Theoretical Utilization				: 26,894146
Experimental Utilization				: 56,471383
Divergence				: -109,976483
Total Packets sent				: 128891
Total Packets received				: 20385
Total Packets dropped				: 108506
Average Delay				: 0,001074 sec
Last packet received at				: 2,098214 sec
Total Transmission time				: 1,84821 sec

#### results - Σημειωματάριο

Αρχείο	Επεξεργασία	Μορφή	Προβολή	Βοήθεια
Packet size				: 208
Theoretical Utilization				: 54,454538
Experimental Utilization				: 76,927439
Divergence				: -41,269104
Total Packets sent				: 39671
Total Packets received				: 9024
Total Packets dropped				: 30647
Average Delay				: 0,001618 sec
Last packet received at				: 2,201961 sec
Total Transmission time				: 1,95196 sec

#### results - Σημειωματάριο

Αρχείο	Επεξεργασία	Μορφή	Προβολή	Βοήθεια
Packet size				: 352
Theoretical Utilization				: 66,923969
Experimental Utilization				: 84,002925
Divergence				: -25,519939
Total Packets sent				: 23435
Total Packets received				: 6086
Total Packets dropped				: 17349
Average Delay				: 0,001622 sec
Last packet received at				: 2,290188 sec
Total Transmission time				: 2,04019 sec

#### results - Σημειωματάριο

Αρχείο	Επεξεργασία	Μορφή	Προβολή	Βοήθεια
Packet size				: 496
Theoretical Utilization				: 74,033166
Experimental Utilization				: 87,374296
Divergence				: -18,020478
Total Packets sent				: 16623
Total Packets received				: 4691
Total Packets dropped				: 11932
Average Delay				: 0,002235 sec
Last packet received at				: 2,380362 sec
Total Transmission time				: 2,13036 sec

#### results - Σημειωματάριο

Αρχείο	Επεξεργασία	Μορφή	Προβολή	Βοήθεια
Packet size				: 640
Theoretical Utilization				: 78,626976
Experimental Utilization				: 89,408428
Divergence				: -13,712153
Total Packets sent				: 12907
Total Packets received				: 3864
Total Packets dropped				: 9043
Average Delay				: 0,002056 sec
Last packet received at				: 2,462731 sec
Total Transmission time				: 2,21273 sec

#### results - Σημειωματάριο

Αρχείο	Επεξεργασία	Μορφή	Προβολή	Βοήθεια
Packet size				: 784
Theoretical Utilization				: 81,839724
Experimental Utilization				: 89,908915
Divergence				: -9,859749
Total Packets sent				: 10548
Total Packets received				: 3330
Total Packets dropped				: 7218
Average Delay				: 0,003064 sec
Last packet received at				: 2,572991 sec
Total Transmission time				: 2,32299 sec

#### results - Σημειωματάριο

Αρχείο Επεξεργασία Μορφή Προβολή Βοήθεια

Packet size : 928  
 Theoretical Utilization : 84,212819  
 Experimental Utilization : 92,037763  
 Divergence : -9,291868  
 Total Packets sent : 8889  
 Total Packets received : 2926  
 Total Packets dropped : 5963  
 Average Delay : 0,002338 sec  
 Last packet received at : 2,610186 sec  
 Total Transmission time : 2,36019 sec

#### results - Σημειωματάριο

Αρχείο Επεξεργασία Μορφή Προβολή Βοήθεια

Packet size : 1072  
 Theoretical Utilization : 86,037385  
 Experimental Utilization : 92,134676  
 Divergence : -7,086793  
 Total Packets sent : 7727  
 Total Packets received : 2666  
 Total Packets dropped : 5061  
 Average Delay : 0,003234 sec  
 Last packet received at : 2,731543 sec  
 Total Transmission time : 2,48154 sec

#### results - Σημειωματάριο

Αρχείο Επεξεργασία Μορφή Προβολή Βοήθεια

Packet size : 1216  
 Theoretical Utilization : 87,483903  
 Experimental Utilization : 92,685552  
 Divergence : -5,945836  
 Total Packets sent : 6798  
 Total Packets received : 2462  
 Total Packets dropped : 4336  
 Average Delay : 0,003560 sec  
 Last packet received at : 2,834042 sec  
 Total Transmission time : 2,58404 sec

#### results - Σημειωματάριο

Αρχείο Επεξεργασία Μορφή Προβολή Βοήθεια

Packet size : 1360  
 Theoretical Utilization : 88,658839  
 Experimental Utilization : 93,575359  
 Divergence : -5,545437  
 Total Packets sent : 6083  
 Total Packets received : 2261  
 Total Packets dropped : 3822  
 Average Delay : 0,003129 sec  
 Last packet received at : 2,878863 sec  
 Total Transmission time : 2,62886 sec

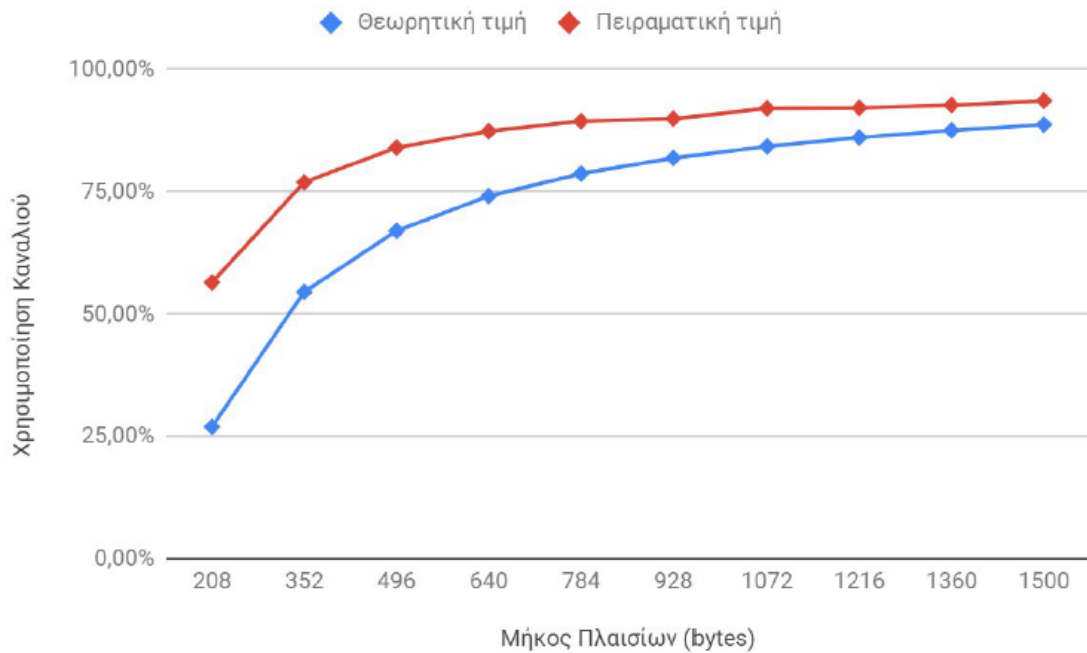
#### results - Σημειωματάριο

Αρχείο Επεξεργασία Μορφή Προβολή Βοήθεια

Packet size : 1500  
 Theoretical Utilization : 89,607341  
 Experimental Utilization : 94,249012  
 Divergence : -5,180012  
 Total Packets sent : 5503  
 Total Packets received : 2131  
 Total Packets dropped : 3372  
 Average Delay : 0,003932 sec  
 Last packet received at : 2,963238 sec  
 Total Transmission time : 2,71324 sec

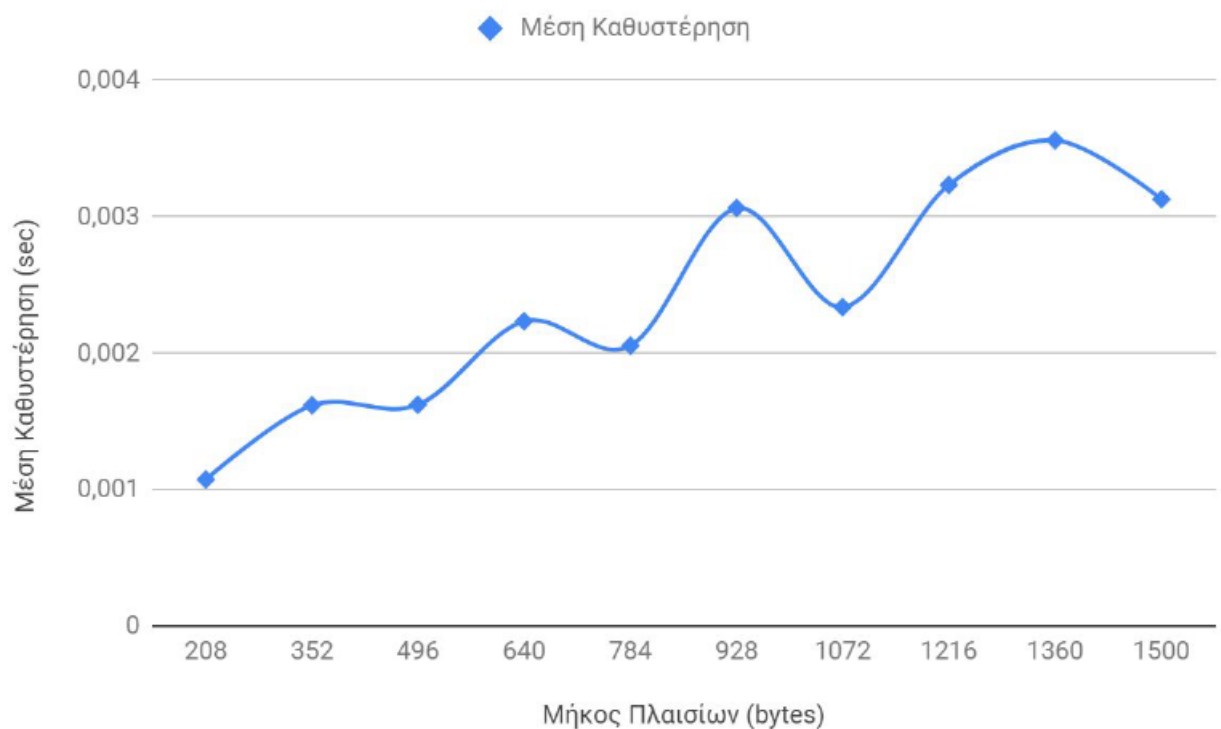
### Πίνακας

Packet size (bytes)	Theoretical Utilization	Experimental Utilization	Total Packets Received	Total Transmission Time
64	26,89%	56,47%	20385	1,848214
208	54,45%	76,93%	9024	1,951961
352	66,93%	84,00%	6086	2,040188
496	74,04%	87,37%	4691	2,130362
640	78,63%	89,41%	3864	2,212731
784	81,84%	89,91%	3330	2,322991
928	84,22%	92,04%	2926	2,360186
1072	86,04%	92,13%	2666	2,481543
1216	87,49%	92,69%	2462	2,584042
1360	88,66%	93,58%	2261	2,628863
1500	89,61%	94,25%	2131	2,713238

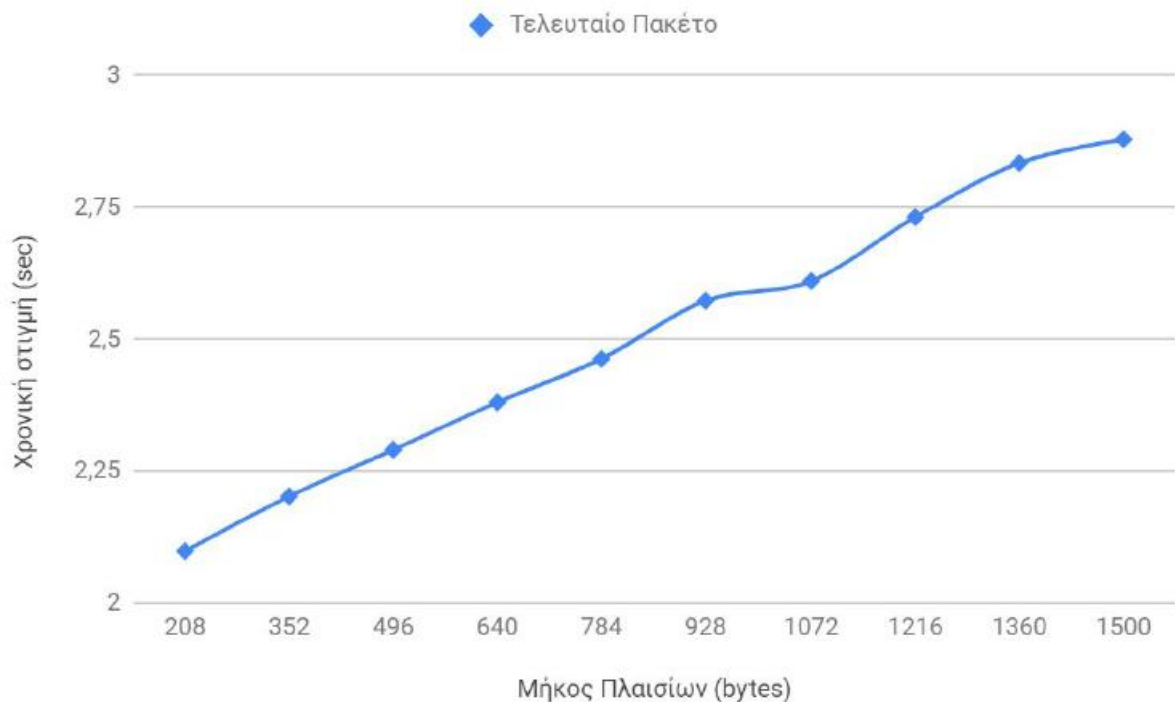


*Η γραφική παράσταση της θεωρητικής και πειραματικής τιμής της χρησιμοποίησης του καναλιού (utilisation) σε σχέση με το μέγεθος του πλαισίου*

Μετά από τις κατάλληλες τροποποιήσεις στον κώδικα και την καταγραφή των δεδομένων που προέκυψαν από το αρχείο awk προκύπτουν τα εξής:



*Γραφική παράσταση της μέσης καθυστέρησης που παρατηρείται με χρήση του πρωτοκόλλου IEEE 802.3 συναρτήσει του μεγέθους των πλαισίων*



*Γραφική παράσταση χρόνου τελευταίας επιτυχημένης λήψης πλαισίου συναρτήσει του μεγέθους των πλαισίων*

b) Βάσει των γραφικών παραστάσεων που σχεδιάσατε, απαντήσατε αν είναι προτιμότερο να χρησιμοποιούνται μικρά ή μεγάλα πλαίσια. Αιτιολογήστε και θεωρητικά την απάντησή σας. Μετά από παρατήρηση της γραφικής παράστασης χρησιμοποίησης καναλιού προκύπτει ότι όσο μεγαλύτερα είναι τα πακέτα τόσο μεγαλύτερη είναι και η απόδοση. Αυτό επιβεβαιώνεται τόσο από τη θεωρητική όσο και από την πειραματική μελέτη. Παρ' όλα αυτά, μετά την επεξεργασία των διαγραμμάτων μέσης καθυστέρησης της ζεύξης και χρονικής λήψης του τελευταίου πακέτου διαπιστώνεται, ότι όσο αυξάνεται το μέγεθος των πακέτων τόσο πιο αργή είναι η ζεύξη. Άρα, εφόσον σκοπός είναι μια γρήγορη ζεύξη θα πρέπει το μέγεθος των πακέτων που χρησιμοποιείται να είναι μικρό. Οι παραπάνω ισχυρισμοί επιβεβαιώνονται και από τον Πίνακα.

c) Κάντε τις απαραίτητες τροποποιήσεις ώστε να εντοπίζετε τη ροή με το μέγιστο ρυθμό μετάδοσης για μέγεθος πλαισίου 64 Byte, 512 Byte, 1024 Byte και 1500 Byte.

Γίνονται οι κατάλληλες αλλαγές – προσθήκες στον κώδικα του προγράμματος awk έτσι ώστε αυτό να αποθηκεύει τα συνολικά δεδομένα, τον συνολικό χρόνο μετάδοσης αυτών καθώς και να υπολογίζει τον ρυθμό μετάδοσης για κάθε μία από τις 18 ροές του δικτύου. Μεταβάλλοντας το μέγεθος πλαισίου σε 500, 1000 και 1500 bytes εκτελούνται 3 νέες προσομοιώσεις. Τα αποτελέσματα αυτών φαίνονται παρακάτω:

*Πηγαίος κώδικας:*

```
BEGIN {
sum_delay = 0;
bufferspace = 100000;
total_pkts_sent = 0;
total_pkts_recv = 0;
total_pkts_dropped = 0;
flow_1 = 0;
flow_2 = 0;
flow_3 = 0;
flow_4 = 0;
flow_5 = 0;
flow_6 = 0;
flow_7 = 0;
flow_8 = 0;
flow_9 = 0;
flow_10 = 0;
flow_11 = 0;
flow_12 = 0;
flow_13 = 0;
flow_14 = 0;
flow_15 = 0;
flow_16 = 0;
flow_17 = 0;
if (sprintf(sqrt(2)) ~ /./) dmfix = 1;
}

{
if (dmfix) sub(/\./, "", $0);
}
/^h/&&/cbr/ {
total_pkts_sent++;
}
/^d/&&/cbr/ {
total_pkts_dropped++;
}
/^-/&&/cbr/ {
sendtimes[$12%bufferspace] = $2;
}
/^r/&&/cbr/ {
total_pkts_recv++;
sum_delay += $2 -
sendtimes[$12%bufferspace];
last = $2;
}
```

```
/^r/&&/1.0/ {
flow_1+=$6
time_1 = $2 - 0.25;
}
/^r/&&/2.0/ {
flow_2+=$6
time_2 = $2 - 0.25;
}
/^r/&&/3.0/ {
flow_3+=$6
time_3 = $2 - 0.25;
}
/^r/&&/4.0/ {
flow_4+=$6
time_4 = $2 - 0.25;
}
/^r/&&/5.0/ {
flow_5+=$6
time_5 = $2 - 0.25;
}
/^r/&&/6.0/ {
flow_6+=$6
time_6 = $2 - 0.25;
}
/^r/&&/7.0/ {
flow_7+=$6
time_7 = $2 - 0.25;
}
/^r/&&/8.0/ {
flow_8+=$6
time_8 = $2 - 0.25;
}
/^r/&&/9.0/ {
flow_9+=$6
time_9 = $2 - 0.25;
}
/^r/&&/10.0/ {
flow_10+=$6
time_10 = $2 - 0.25;
}
/^r/&&/11.0/ {
flow_11+=$6
time_11 = $2 - 0.25;
}
```

```

/^r/ && /12.0/ {
    flow_12 += $6
    time_12 = $2 - 0.25;
}
/^r/ && /13.0/ {
    flow_13 += $6
    time_13 = $2 - 0.25;
}
/^r/ && /14.0/ {
    flow_14 += $6
    time_14 = $2 - 0.25;
}

/^r/ && /15.0/ {
    flow_15 += $6
    time_15 = $2 - 0.25;
}
/^r/ && /16.0/ {
    flow_16 += $6
    time_16 = $2 - 0.25;
}
/^r/ && /17.0/ {
    flow_17 += $6
    time_17 = $2 - 0.25;
}

END {
    printf("Total Packets sent\t: %d\n", total_pkts_sent);
    printf("Total Packets received\t: %d\n", total_pkts_rcv);
    printf("Total Packets dropped\t: %d\n", total_pkts_dropped);
    printf("Average Delay\t: %f sec\n", (1.0 * sum_delay)/ total_pkts_rcv);
    printf("Last package recieved\t: %f\n", last);
    printf("\n");
    printf("Rate 1\t: %d bps\n", 8*flow_1/time_1);
    printf("Rate 2\t: %d bps\n", 8*flow_2/time_2);
    printf("Rate 3\t: %d bps\n", 8*flow_3/time_3);
    printf("Rate 4\t: %d bps\n", 8*flow_4/time_4);
    printf("Rate 5\t: %d bps\n", 8*flow_5/time_5);
    printf("Rate 6\t: %d bps\n", 8*flow_6/time_6);
    printf("Rate 7\t: %d bps\n", 8*flow_7/time_7);
    printf("Rate 8\t: %d bps\n", 8*flow_8/time_8);
    printf("Rate 9\t: %d bps\n", 8*flow_9/time_9);
    printf("Rate 10\t: %d bps\n", 8*flow_10/time_10);
    printf("Rate 11\t: %d bps\n", 8*flow_11/time_11);
    printf("Rate 12\t: %d bps\n", 8*flow_12/time_12);
    printf("Rate 13\t: %d bps\n", 8*flow_13/time_13);
    printf("Rate 14\t: %d bps\n", 8*flow_14/time_14);
    printf("Rate 15\t: %d bps\n", 8*flow_15/time_15);
    printf("Rate 16\t: %d bps\n", 8*flow_16/time_16);
    printf("Rate 17\t: %d bps\n", 8*flow_17/time_17);
}

```

Για 64 bytes:

results - Σημειωματάριο				
Αρχείο	Επεξεργασία	Μορφή	Προβολή	Βοήθεια
<hr/>				
Total Packets sent	:	128891		
Total Packets received	:	20385		
Total Packets dropped	:	108506		
Average Delay	:	0,001074 sec		
Last package recieved	:	2,098214		
<hr/>				
Rate 1	:	1549673 bps		
Rate 2	:	1056847 bps		
Rate 3	:	1041888 bps		
Rate 4	:	1029604 bps		
Rate 5	:	1352825 bps		
Rate 6	:	848388 bps		
Rate 7	:	875596 bps		
Rate 8	:	5647138 bps		
Rate 9	:	759105 bps		
Rate 10	:	301438 bps		
Rate 11	:	543689 bps		
Rate 12	:	279240 bps		
Rate 13	:	481194 bps		
Rate 14	:	417283 bps		
Rate 15	:	498742 bps		
Rate 16	:	272325 bps		
Rate 17	:	289648 bps		

Η ροή 8 έχει μέγιστο ρυθμό μετάδοσης.

Για 1024 bytes:

results - Σημειωματάριο				
Αρχείο	Επεξεργασία	Μορφή	Προβολή	Βοήθεια
<hr/>				
Total Packets sent	:	8058		
Total Packets received	:	2754		
Total Packets dropped	:	5304		
Average Delay	:	0,002406 sec		
Last package recieved	:	2,711561		
<hr/>				
Rate 1	:	1799674 bps		
Rate 2	:	2359714 bps		
Rate 3	:	1246787 bps		
Rate 4	:	1723888 bps		
Rate 5	:	1187593 bps		
Rate 6	:	1371539 bps		
Rate 7	:	1674540 bps		
Rate 8	:	9165228 bps		
Rate 9	:	856645 bps		
Rate 10	:	554672 bps		
Rate 11	:	455325 bps		
Rate 12	:	868704 bps		
Rate 13	:	384793 bps		
Rate 14	:	556203 bps		
Rate 15	:	607275 bps		
Rate 16	:	680843 bps		
Rate 17	:	802931 bps		

Η ροή 8 έχει μέγιστο ρυθμό μετάδοσης.

Για 512 bytes:

results - Σημειωματάριο				
Αρχείο	Επεξεργασία	Μορφή	Προβολή	Βοήθεια
<hr/>				
Total Packets sent	:	16071		
Total Packets received	:	4563		
Total Packets dropped	:	11508		
Average Delay	:	0,001893 sec		
Last package recieved	:	2,406193		
<hr/>				
Rate 1	:	1842655 bps		
Rate 2	:	1698073 bps		
Rate 3	:	1645447 bps		
Rate 4	:	1476023 bps		
Rate 5	:	1374286 bps		
Rate 6	:	1423970 bps		
Rate 7	:	1550630 bps		
Rate 8	:	8668077 bps		
Rate 9	:	1164209 bps		
Rate 10	:	548299 bps		
Rate 11	:	532575 bps		
Rate 12	:	440963 bps		
Rate 13	:	616435 bps		
Rate 14	:	500283 bps		
Rate 15	:	523447 bps		
Rate 16	:	513313 bps		
Rate 17	:	661760 bps		

Η ροή 8 έχει μέγιστο ρυθμό μετάδοσης.

Για 1500 bytes:

results - Σημειωματάριο				
Αρχείο	Επεξεργασία	Μορφή	Προβολή	Βοήθεια
<hr/>				
Total Packets sent	:	5503		
Total Packets received	:	2131		
Total Packets dropped	:	3372		
Average Delay	:	0,003932 sec		
Last package recieved	:	2,963238		
<hr/>				
Rate 1	:	9424901 bps		
Rate 2	:	2173018 bps		
Rate 3	:	1415919 bps		
Rate 4	:	1431859 bps		
Rate 5	:	9424901 bps		
Rate 6	:	1512583 bps		
Rate 7	:	1268975 bps		
Rate 8	:	9424901 bps		
Rate 9	:	818947 bps		
Rate 10	:	694565 bps		
Rate 11	:	521246 bps		
Rate 12	:	916251 bps		
Rate 13	:	661344 bps		
Rate 14	:	623719 bps		
Rate 15	:	9424901 bps		
Rate 16	:	576826 bps		
Rate 17	:	469600 bps		

Οι ροές 1, 5, 8 έχουν μέγιστο ρυθμό μετάδοσης.



#### 4.2 Επίδραση αριθμού σταθμών

Στη συνέχεια θα γίνει σειρά προσομοιώσεων με μεταβλητό πλήθος σταθμών, και θα μελετηθεί η επίδραση του πλήθους των σταθμών στην χρησιμοποίηση του καναλιού και στην μέση καθυστέρηση. Για το σκοπό θα γίνει σειρά προσομοιώσεων με τις εξής παραμέτρους:

- Ρυθμός μετάδοσης στο LAN 10 Mbps
- Καθυστέρηση διάδοσης 25,6  $\mu$ s.
- Ρυθμός μετάδοσης δεδομένων κάθε σταθμού ίσος με το τετραπλάσιο του ονομαστικού ρυθμού μετάδοσης του καναλιού προς το πλήθος των σταθμών.
- Μέγεθος πλαισίου 1024 bytes.
- Διάρκεια προσομοίωσης 5 sec.
- Θα γίνουν τέσσερις προσομοιώσεις. Στην πρώτη θα συμμετέχουν 6 σταθμοί, στην δεύτερη 12, στην τρίτη 18 και στην τέταρτη 24

α) Για τις παραπάνω προσομοιώσεις να γίνει γραφική παράσταση της πειραματικής τιμής της χρησιμοποίησης του καναλιού (*utilisation*), της μέσης καθυστέρησης, καθώς και του ποσοστού του απορριφθέντων πλαισίων (επί του συνολικού αριθμού πλαισίων που αποστέλλονται) συναρτήσει του πλήθους των σταθμών στο δίκτυο.

Για την μελέτη της επίδραση του αριθμού των σταθμών στην απόδοση εκτελούνται 4 προσομοιώσεις μεταβάλλοντας στο tel script των αριθμό των σταθμών σε 6, 12, 18 και 24 κρατώντας σταθερές τις υπόλοιπες παραμέτρους. Τροποποιείται ξανά το πρόγραμμα awk που χρησιμοποιήθηκε και πριν ώστε να υπολογίζει και να εμφανίζει (εκτός των άλλων) το ποσοστό των απορριφθέντων πλαισίων επί του συνολικού αριθμού πλαισίων που αποστέλλονται.

Πηγαίος κώδικας:

```
BEGIN {
sum_delay = 0;
bufferspace = 100000;
total_pkts_sent = 0;
total_pkts_recv = 0;
total_pkts_dropped = 0;
if (sprintf(sqrt(2)) ~ /\./) dmfix = 1;
}

{
if (dmfix) sub(/\./, "", $0);
}

/^h/&&/cbr/ {
stations = $4;
total_pkts_sent++;
}

/^d/&&/cbr/ {
total_pkts_dropped++;
}
```

```


/^-/&&/cbr/ {
sendtimes[$12%bufferspace] = $2;
}

/^r/&&/cbr/ {
packet_size = $6;
total_pkts_rcv++;
sum_delay += $2 - sendtimes[$12%bufferspace];
last_ts = $2;
}

{
tr_time = last_ts - 0.25;
exp_ut = 100 * (total_pkts_rcv * packet_size * 8) / (tr_time * 10000000);
lost_packets_percentage = total_pkts_dropped * 100 / total_pkts_sent;
}
END {
printf("Percentage of lost packets\t: %f\n", lost_packets_percentage);
printf("Number of stations\t: %d\n", stations);
printf("Experimental Utilization\t: %f\n", exp_ut);
printf("Total Packets sent\t: %d\n", total_pkts_sent);
printf("Total Packets received\t: %d\n", total_pkts_rcv);
printf("Total Packets dropped\t: %d\n", total_pkts_dropped);
printf("Average Delay\t: %f sec\n", (1.0 * sum_delay) / total_pkts_rcv);
printf("Last packet received at\t: %s sec\n", last_ts);
printf("Total Transmission time\t: %s sec\n", last_ts - 0.25);
}


```

Για 6 σταθμούς προκύπτει:

 results - Σημειωματάριο

Αρχείο	Επεξεργασία	Μορφή	Προβολή	Βοήθεια
Percentage of lost packets	:	68,012379		
Number of stations	:	6		
Experimental Utilization	:	94,544089		
Total Packets sent	:	7109		
Total Packets received	:	2274		
Total Packets dropped	:	4835		
Average Delay	:	0,001627 sec		
Last packet received at	:	2,220362 sec		
Total Transmission time	:	1,97036 sec		

Για 12 σταθμούς προκύπτει:

 results - Σημειωματάριο

Αρχείο	Επεξεργασία	Μορφή	Προβολή	Βοήθεια
Percentage of lost packets	:	67,813739		
Number of stations	:	12		
Experimental Utilization	:	93,646806		
Total Packets sent	:	7817		
Total Packets received	:	2516		
Total Packets dropped	:	5301		
Average Delay	:	0,001864 sec		
Last packet received at	:	2,450937 sec		
Total Transmission time	:	2,20094 sec		

Για 18 σταθμούς προκύπτει:

results - Σημειωματάριο

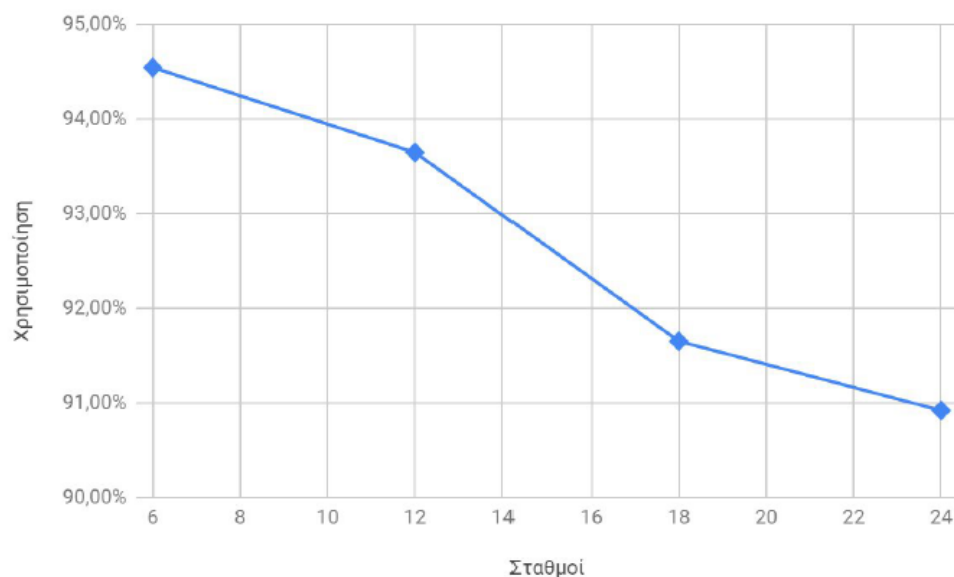
Αρχείο	Επεξεργασία	Μορφή	Προβολή	Βοήθεια
Percentage of lost packets	:	65,822785		
Number of stations	:	18		
Experimental Utilization	:	91,652281		
Total Packets sent	:	8058		
Total Packets received	:	2754		
Total Packets dropped	:	5304		
Average Delay	:	0,002406 sec		
Last packet received at	:	2,711561 sec		
Total Transmission time	:	2,46156 sec		

Για 24 σταθμούς προκύπτει:

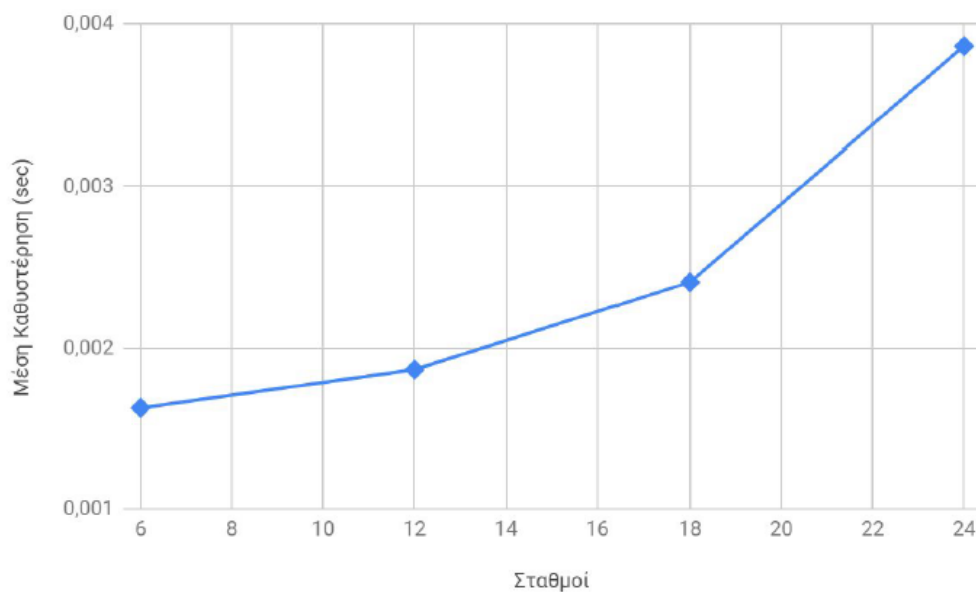
results - Σημειωματάριο

Αρχείο	Επεξεργασία	Μορφή	Προβολή	Βοήθεια
Percentage of lost packets	:	63,267049		
Number of stations	:	24		
Experimental Utilization	:	90,920327		
Total Packets sent	:	8197		
Total Packets received	:	3011		
Total Packets dropped	:	5186		
Average Delay	:	0,003862 sec		
Last packet received at	:	2,962937 sec		
Total Transmission time	:	2,71294 sec		

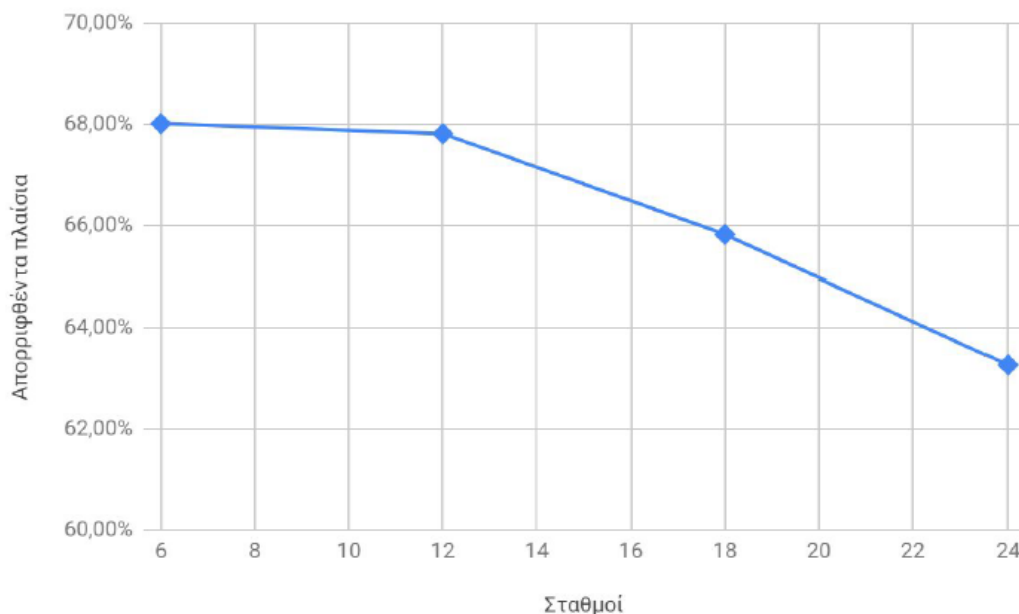
Και προκύπτουν οι παρακάτω γραφικές παραστάσεις:



*Πειραματική τιμή της χρησιμοποίησης του καναλιού συναρτήσει του πλήθους των σταθμών στο δίκτυο*



*Μέση καθυστέρηση συναρτήσει του πλήθους των σταθμών στο δίκτυο*



*Ποσοστό των απορριφθέντων πλαισίων συναρτήσει του πλήθους των σταθμών στο δίκτυο*

*b) Τι παρατηρείτε; Πώς δικαιολογείται η συμπεριφορά του δικτύου;*

Παρατηρείται από το γράφημα της χρησιμοποίησης του καναλιού ότι όσο λιγότεροι οι σταθμοί του δικτύου τόσο μεγαλύτερη και η χρησιμοποίηση. Από το δύο τελευταία γραφήματα προκύπτει ότι για λιγότερους σταθμούς η μέση καθυστέρηση είναι μεγαλύτερη, ενώ τα απορριφθέντα πλαίσια φαίνονται να είναι περισσότερα. Τα συμπεράσματα που εξάγονται είναι λογικά, γιατί η ύπαρξη περισσότερων σταθμών προκαλεί αύξηση της μέσης καθυστέρησης αλλά και αποδοτικότερη εξυπηρέτηση, με αποτέλεσμα να μην χάνονται πακέτα. Παράλληλα, όπως φαίνεται πέφτει και η απόδοση-χρησιμοποίηση αφού το δίκτυο επιβαρύνεται με παραπάνω σταθμούς.