

ΔΙΚΤΥΑ ΕΠΙΚΟΙΝΩΝΙΩΝ

Εργαστηριακή Άσκηση 6

Επίδοση Τοπικών Δικτύων IEEE 802.3

1. Εισαγωγή

Σε αυτή την άσκηση θα μελετηθεί η επίδοση του MAC πρωτοκόλλου IEEE 802.3. Η λειτουργία του πρωτοκόλλου αυτού περιγράφεται αναλυτικά στο βιβλίο «Δίκτυα Υπολογιστών» (A. Tanenbaum), ενότητα 4.3.

Αρχικά θα γραφεί σε γλώσσα tcl 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 ίχνους (trace file), 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 (Εξίσωση 4-6 του βιβλίου του μαθήματος).

Στην ενότητα 2 παρουσιάζονται οι λεπτομέρειες του σεναρίου προσομοίωσης, στην ενότητα 3 παρουσιάζεται η διαδικασία ανάλυσης των αποτελεσμάτων της προσομοίωσης και στην ενότητα 4 αναλύεται η διαδικασία σύγκρισης θεωρητικών και πειραματικών αποτελεσμάτων.

2. Σενάριο προσομοίωσης

2.1 Παράμετροι Προσομοίωσης

Αρχικά δημιουργούμε ένα τοπικό δίκτυο IEEE 802.3. Στο τμήμα του κώδικα που ακολουθεί καθορίζονται οι παράμετροι της προσομοίωσης. Σημαντικότερες από αυτές τις παραμέτρους 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(packetize)*. Αυτές οι παράμετροι πρέπει να αλλάζουν σε κάθε προσομοίωση, όπως περιγράφεται στη συνέχεια.

```
### 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 lab6.nam (NAM) και lab6.tr (trace)
set opt (nam)           "lab6.nam"           ;# Αρχείο NAM
```

```

set opt(tr)          "lab6.tr"                ;# Αρχείο trace

set opt(seed)        0                       ;# Παράμετρος για τη γεννήτρια
;# τυχαίων αριθμών

set opt(starttraf)   0.5                     ;# Έναρξη παραγωγής δεδομένων (sec)
set opt(stoptraf)    2.5                     ;# Λήξη παραγωγής δεδομένων (sec)
set opt(stopsim)     3.0                     ;# Διάρκεια προσομοίωσης (sec)
set opt(node)        8                       ;# Αριθμός σταθμών στο LAN
set opt(qsize)       200                     ;# Μέγεθος ουράς σε κάθε σταθμό
set opt(bw)          10000000                ;# Ρυθμός μετάδοσης (bps)
set opt(delay)       0.0000128              ;# Καθυστέρηση μετάδοσης (sec)
set opt(packetsize) 1500                     ;# Μέγεθος πακέτου (byte)
set opt(rate)        [expr 2*$opt(bw)/$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 δημιουργούμε αρχικά έναν αριθμό από κόμβους και τους τοποθετούμε σε λίστα. Στη συνέχεια, ενώνουμε όλους τους κόμβους της λίστας που δημιουργήσαμε με ένα τοπικό δίκτυο (LAN). Οι παράμετροι του τοπικού δικτύου είναι στις μεταβλητές που ορίσαμε αρχικά.

```

# 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

Μας ενδιαφέρει να μελετήσουμε την 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 προσαρμόζουμε γεννήτρια κίνησης CBR (Constant Bit Rate) με ρυθμό `opt(rate)`, ο οποίος είναι διπλάσιος του ρυθμού μετάδοσης του καναλιού προς το συνολικό πλήθος των κόμβων. Για τη μεταφορά της κίνησης χρησιμοποιούμε το πρωτόκολλο UDP. Η μετάδοση των πλαισίων ξεκινά τη χρονική στιγμή `opt(starttraf)` και ολοκληρώνεται τη χρονική στιγμή `opt(stoptraf)` για όλους του κόμβους.

```
### 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 ανοίξουμε το αρχείο όπου θα αποθηκεύονται οι πληροφορίες του *animation*, να δώσουμε οδηγία στον προσομοιωτή να αποθηκεύσει την πληροφορία για το *animation* σε αυτό το αρχείο και, όταν ολοκληρωθεί η προσομοίωση, να κλείσουμε το αρχείο. Αυτά γίνονται με την ακόλουθη διαδικασία:

```
### 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 δημιουργούμε 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, όπως φαίνεται και στον παρακάτω κώδικα, πρέπει πρώτα να ανοίξουμε ένα αρχείο για εγγραφή, να δώσουμε οδηγία στον προσομοιωτή να αποθηκεύσει την πληροφορία για τα γεγονότα που θα συμβούν σε αυτό το αρχείο και, όταν ολοκληρωθεί η προσομοίωση, να κλείσουμε το αρχείο.

```

### 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
}

```

Η μορφή των καταχωρήσεων στο αρχείο ίχνους περιγράφεται στην επόμενη υποενότητα.

2.6 Εκτέλεση του σεναρίου

Τέλος, ορίζουμε τα γεγονότα της προσομοίωσης:

```

## 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

```

Με βάση τα παραπάνω δημιουργούμε το αρχείο **lab6.tcl**, ενώ για την εκτέλεση της προσομοίωσης χρησιμοποιούμε την εντολή **ns lab6.tcl**. Μετά την εκτέλεση αυτής της εντολής θα πρέπει να έχουν δημιουργηθεί τα αρχεία **lab6.tr** και **lab6.nam**. Με την εντολή **nam lab6.nam** μπορούμε να δούμε την τοπολογία του δικτύου καθώς και την κίνηση που έχει δημιουργηθεί*.

Το αρχείο **lab6.tr** περιέχει πληροφορίες για όλα τα γεγονότα που συνέβησαν κατά την προσομοίωση. Στην επόμενη ενότητα θα εξηγηθεί πως αναλύονται αυτά τα δεδομένα.

3. Ανάλυση αρχείου ίχνους (trace file)

Αφού έχουμε δημιουργήσει το σενάριο προσομοίωσης, το έχουμε εκτελέσει και έχουμε δημιουργήσει τα αρχεία αποτελεσμάτων, τα αρχεία αυτά πρέπει να αναλυθούν ώστε να πάρουμε τις πληροφορίες που θέλουμε.

3.1 Μορφή αρχείου ίχνους (trace file)

Το αρχείο **lab6.tr** που δημιουργήθηκε στο προηγούμενο περιέχει πληροφορίες της μορφής:

```

+ 0.251188 7 8 cbr 1500 ----- 0 7.0 0.6 51 370
- 0.251188 7 8 cbr 1500 ----- 0 7.0 0.6 51 370
h 0.252103 3 8 cbr 1500 ----- 0 3.0 0.2 50 371
+ 0.252129 3 8 cbr 1500 ----- 0 3.0 0.2 50 371
- 0.252129 3 8 cbr 1500 ----- 0 3.0 0.2 50 371
d 0.252417 5 8 cbr 1500 ----- 0 5.0 0.4 29 203
r 0.252429 8 0 cbr 1500 ----- 0 7.0 0.6 51 370
- 0.25243 5 8 cbr 1500 ----- 0 5.0 0.4 30 210
- 0.253651 5 8 cbr 1500 ----- 0 5.0 0.4 31 216
r 0.253671 8 0 cbr 1500 ----- 0 5.0 0.4 30 210

```

Κάθε γραμμή του αρχείου αυτού αντιστοιχεί σε ένα γεγονός που συνέβη κατά τη διάρκεια της προσομοίωσης. Ο πρώτος χαρακτήρας κάθε γραμμής υποδηλώνει το είδος του γεγονότος που συνέβη. Ο χαρακτήρας “+” σημαίνει είσοδο του πακέτου σε ουρά αναμονής, ο χαρακτήρας “-” σημαίνει αποχώρηση από ουρά αναμονής, ο χαρακτήρας “h” σημαίνει προώθηση πακέτου, ο χαρακτήρας “r” σημαίνει επιτυχημένη λήψη πακέτου, και ο χαρακτήρας “d” σημαίνει απόρριψη πακέτου.

Η δεύτερη λέξη της κάθε γραμμής είναι η χρονική στιγμή κατά την οποία συνέβη το γεγονός που καταγράφεται. Οι επόμενες δύο λέξεις περιγράφουν μεταξύ ποιων κόμβων βρίσκεται το πακέτο, η

* Το τοπικό δίκτυο ίσως και να μην εμφανίζεται πολύ καλά στο nam.

επόμενη λέξη είναι το είδος του πακέτου και η έκτη λέξη περιγράφει το μέγεθος του πακέτου (συμπεριλαμβάνονται οι επικεφαλίδες TCP και IP). Οι παύλες αντιστοιχούν σε πεδία που δεν χρησιμοποιούνται στο παράδειγμα.

Ο πρώτος αριθμός μετά τις παύλες είναι το `flow_id` της ροής στην οποία ανήκει το πακέτο, που ακολουθείται από τη διεύθυνση αποστολέα και προορισμού (`IP.port`), από τον αύξοντα αριθμό (sequence number) του πακέτου και τέλος από ένα `unique number` του πακέτου.

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. Για παράδειγμα ο κανόνας:

```
 /^r/&&/cbr/ {  
    data+=$6;  
    packets++;  
    sumDelay += $2 - sendtimes[$12%bufferspace];  
}
```

ορίζει ότι όταν συναντήσουμε μια γραμμή του αρχείου εισόδου που να ξεκινάει με τον χαρακτήρα “r” (`/^r/`) και (`&&`) που περιλαμβάνει τη λέξη `cbr` (`/cbr/`), τότε η τιμή της μεταβλητής *packets* αυξάνεται κατά 1, η τιμή της μεταβλητής *data* αυξάνεται κατά την τιμή που βρίσκεται στην έκτη λέξη (`$6`) της γραμμής την οποία εξετάζουμε, και η αύξηση της τιμής της μεταβλητής *sumDelay* ισούται με τη διαφορά της τιμής που βρίσκεται στη δεύτερη θέση της εξεταζόμενης γραμμής, μείον την τιμή του πίνακα *sendtimes* στη θέση `$12%bufferspace`.

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 να μπορούμε να υπολογίσουμε την χρησιμοποίηση (utilisation) του καναλιού, όταν χρησιμοποιείται το πρωτόκολλο IEEE 802.3, πρέπει να μετρήσουμε την 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).

Για κάθε πακέτο που λαμβάνει ο αποδέκτης, η ποσότητα των δεδομένων αυξάνεται κατά το μέγεθος του πακέτου. Συνεπώς πρέπει να εντοπίσουμε τις 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 *packets* (αριθμός πακέτων που ελήφθησαν) αυξάνεται κατά ένα, ενώ η μεταβλητή *data* (πλήθος δεδομένων που ελήφθησαν) αυξάνεται κατά το μέγεθος του πακέτου (το οποίο βρίσκεται στην έκτη λέξη κάθε γραμμής του αρχείου ίχνους). Έτσι

καταγράφεται η ποσότητα των δεδομένων που ελήφθησαν.

Η καθυστέρηση κάθε πακέτου προκύπτει αν αφαιρεθεί ο χρόνος λήψης κάθε πακέτου από τον χρόνο αποστολής του. Ως χρόνο αποστολής ενός πακέτου θεωρούμε τη χρονική στιγμή κατά την οποία το πακέτο εξέρχεται από την ουρά αναμονής στον κόμβο αποστολής. Με το ακόλουθο τμήμα κώδικα:

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

εντοπίζονται οι γραμμές που αναφέρονται σε αποστολή πακέτου. Η χρονική στιγμή αποστολής (\$2) αποθηκεύεται στον πίνακα *sendtimes*, σε θέση που προκύπτει από το υπόλοιπο της διαίρεσης του id του πακέτου που εστάλη δια του χώρου αποθήκευσης στον πίνακα (*bufferspace*). Η μεταβλητή *bufferspace* χρησιμοποιείται, ώστε να περιοριστεί το μέγεθος του πίνακα αποθήκευσης.

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.

Ο κώδικας awk για την εκτέλεση αυτής της ανάλυσης είναι ο εξής:

```
BEGIN {  
    data = 0;  
    packets = 0;  
    buffersize = 10;  
    sumDelay = 0;  
    bufferspace = 10000;  
  
    # Do we need to fix the decimal mark?  
    if (sprintf(sqrt(2)) ~ /,/) dmfix = 1;  
}  
  
{  
    # Apply dm fix if needed  
    if (dmfix) sub(/\. /, ", ", $0);  
}  
  
/^-/&&/cbr/ {  
    sendtimes[$12%bufferspace] = $2;  
}  
  
/^r/&&/cbr/ {  
    data += $6;  
    packets++;  
    sumDelay += $2 - sendtimes[$12%bufferspace];  
}
```

```

END{
    printf("Total Data received\t: %d Bytes\n", data);
    printf("Total Packets received\t: %d\n", packets);
    printf("Average Delay\t\t: %f sec\n", (1.0 * sumDelay)/ packets);
}

```

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 **lab6.awk** και τα δεδομένα βρίσκονται στο αρχείο **lab6.tr**, τότε εκτελούμε την εντολή:

```
awk.exe -f lab6.awk < lab6.tr
```

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

Για να αποθηκεύσουμε τα αποτελέσματα της ανάλυσης στο αρχείο **results.txt** εκτελούμε την εντολή

```
awk.exe -f lab6.awk < lab6.tr > results.txt
```

Προσοχή: 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, έχουμε προσθέσει τον ακόλουθο κανόνα ο οποίος αντικαθιστά όπου χρειάζεται τις τελείες με κόμματα.

```

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

```

Εναλλακτικά, θα μπορούσαμε να ορίσουμε την τελεία ως το σύμβολο για την υποδιαστολή αλλάζοντας τις ρυθμίσεις των Windows (Control Panel -> Regional and Language Options).

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:

$$\eta = \frac{P}{P + 2\tau e}$$

Όπου:

- η είναι η απόδοση του διαύλου (πραγματικός ρυθμός μεταφοράς δεδομένων / ονομαστικό ρυθμό μετάδοσης ζεύξης)
- P είναι ο χρόνος μετάδοσης κάθε πλαισίου (Μήκος πλαισίου σε bit / ονομαστικό ρυθμό μετάδοσης ζεύξης)
- τ είναι η μέγιστη επιτρεπόμενη καθυστέρηση διάδοσης πάνω στο καλώδιο, την οποία μπορείτε να θεωρήσετε ίση με 12,8 μ s.
- $e = 2,718281828$

4.1 Επίδραση μεγέθους πλαισίων

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

- 8 σταθμοί συνδεδεμένοι πάνω από ένα τοπικό δίκτυο IEEE 802.3.
- Ρυθμός μετάδοσης στο LAN 10 Mbps.
- Καθυστέρηση διάδοσης 12,8 μ s.
- Ρυθμός μετάδοσης δεδομένων κάθε σταθμού ίσος με το διπλάσιο 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 2000 Byte (μέγιστο μέγεθος πλαισίου). Πρέπει να πραγματοποιηθούν τουλάχιστον δέκα προσομοιώσεις.

- 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.
- Βάσει των γραφικών παραστάσεων που σχεδιάσατε, απαντήσατε αν είναι προτιμότερο να χρησιμοποιούνται μικρά ή μεγάλα πλαίσια. Αιτιολογήστε και θεωρητικά την απάντησή σας.
- Θα εμπιστευόσασταν περισσότερο τους θεωρητικούς υπολογισμούς ή την προσομοίωση ως προς την εγκυρότητα των αποτελεσμάτων σας;

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 12,8 μ s.
- Ρυθμός μετάδοσης δεδομένων κάθε σταθμού ίσος με το διπλάσιο του ονομαστικού ρυθμού μετάδοσης του καναλιού προς το πλήθος των σταθμών.
- Μέγεθος πλαισίου 1500 bytes.
- Διάρκεια προσομοίωσης 5 sec.
- Θα γίνουν τρεις προσομοιώσεις. Στην πρώτη θα συμμετέχουν 16 σταθμοί, στην δεύτερη 32, και

στην τρίτη 48.

- (α) Για τις παραπάνω προσομοιώσεις να γίνει γραφική παράσταση της πειραματικής τιμής της χρησιμοποίησης του καναλιού (utilisation), καθώς και της μέσης καθυστέρησης συναρτήσει του πλήθους των σταθμών στο δίκτυο.
- (β) Τι παρατηρείτε; Πώς δικαιολογείται η συμπεριφορά του δικτύου;