ΔΙΚΤΥΑ ΕΠΙΚΟΙΝΩΝΙΩΝ

Εργαστηριακή Άσκηση 5

Επίδοση Τοπικών Δικτύων 802.3

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), § 4.3.

Αρχικά θα γραφεί σε γλώσσα tcl ένα script για τον NS 2, το οποίο θα προσομοιώνει ένα τοπικό δίκτυο 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 παράγραφο 1 παρουσιάζονται οι λεπτομέρειες του σεναρίου προσομοίωσης, στην παράγραφο 2 παρουσιάζεται η διαδικασία ανάλυσης των αποτελεσμάτων της προσομοίωσης και στην παράγραφο 3 αναλύεται η διαδικασία σύγκρισης θεωρητικών και πειραματικών αποτελεσμάτων.

1 Σενάριο προσομοίωσης.

1.1 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(stop)**, 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 όπως περιγράφεται στη συνέχεια.

```
set opt(tr)     "lantest.tr"    ;# Trace arxeio pou perilamvanei
                                ;#ta apotelesmata tis prosomoiosis

set opt(nam)     "lantest.nam"  ;# Arxeio katagrafis pliroforias nam

set opt(seed)   0               ;# Gia tin gennhtria tyxaion arithmon
set opt(stop)   1               ;# Diarkeia prosomoiosis se seconds
set opt(node)   6               ;# Arithmos stathmon sto LAN

set opt(qsize)  100             ;# Megethos buffer se kathe stathmo
set opt(bw)     10000000        ;# Rythmos metadoshs sto LAN
set opt(delay)  0.000000950     ;# Kathysterisi metadoshs sto LAN
set opt(packetsize) 1500        ;# Megethos paketou

set opt(rate)   [expr 2* $opt(bw)/ $opt(node)] ;# Rythmos paragogis
                                ;# dedomenonn kathe stathmou.

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

1.2 Τοπολογία

Η τοπολογία της προσομοίωσης αποτελείται από έναν αριθμό κόμβων συνδεδεμένων σε ένα τοπικό δίκτυο. Με το παρακάτω κομμάτι κώδικα δημιουργούμε έναν αριθμό από κόμβους, και τους τοποθετούμε σε λίστα.

```
        ### Dimiourgia Stathmon tou lan
        for {set i 0} {$i < $num} {incr i} {
                set node($i) [$ns node]
                lappend nodelist $node($i)
        }
```

Με αυτό το κομμάτι κώδικα ενώνουμε όλους τους κόμβους της λίστας που δημιουργήσαμε με ένα τοπικό δίκτυο (LAN). Οι παράμετροι του τοπικού δικτύου είναι στις μεταβλητές που ορίσαμε αρχικά.

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

1.3 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 διπλάσιο του ρυθμού μετάδοσης του καναλιού. Για τη μεταφορά της κίνησης χρησιμοποιούμε το πρωτόκολλο UDP. Όλες οι επικοινωνίες ξεκινούν τη χρονική στιγμή 0 και ολοκληρώνονται στο τέλος της προσομοίωσης.

```
### Dimiourgia Kinisis metaksy ton stathmon
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_ [expr $opt(packetsize) + 100]
                $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)

                $ns at 0.000$i "$cbr($i) start"
        }
}
```

1.4 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 τον ακόλουθο κώδικα.

```
###  Anoigma nam arxeiou gia tin animation ton symvanton tis prosomoiosis
proc create-nam-trace {} {
        global ns opt
        set namfd [open $opt(nam) w] ;# Anoigma tou arxeiou
```

```
        $ns namtrace-all $namfd ;# Entoli gia katagrafi gegonoton
        return $namfd
}
```

1.5 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.

```
###  Anoigma trace arxeiou gia tin katagrafi ton symvanton tis prosomoiosis
proc create-trace {} {
        global ns opt
        set trfd [open $opt(tr) w] ;# Anoigma tou arxeiou
        $ns trace-all $trfd ;# Entoli gia katagrafi gegonoton
        return $trfd
}

### Diadikasia termatismou - Klesimo anoixton arxeion
proc finish {} {
        global ns trfd namfd

        $ns flush-trace
        close $trfd
        close $namfd
        exit 0
}
```

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

1.6 Εκτέλεση του σεναρίου

Με βάση τα παραπάνω δημιουργούμε το αρχείο **lantest.tcl**, το οποίο υπάρχει ολοκληρωμένο στο παράρτημα.

Αφού δημιουργήσουμε και αποθηκεύσουμε το αρχείο, το εκτελούμε με την εντολή **ns lantest.tcl.** Με την εκτέλεση αυτής της εντολής θα πρέπει να έχουν δημιουργηθεί τα αρχεία **lantest.tr** και **lantest.nam.** Με την εντολή **nam lantest.nam** μπορούμε να δούμε την τοπολογία του δικτύου καθώς και την κίνηση που έχει δημιουργηθεί[*].

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

Προσοχή: Ο χρόνος εκτέλεσης της προσομοίωσης είναι ανάλογος του αριθμού των πακέτων που αποστέλλονται. Για τον λόγο αυτό, η διάρκεια της προσομοίωσης (**opt(stop)**) πρέπει να ελαττώνεται, όταν το μέγεθος των πακέτων είναι μικρό. Αντίθετα, όταν το μέγεθος των πακέτων είναι μεγάλο, θα πρέπει να αυξάνεται η διάρκεια της προσομοίωσης, ώστε τα αποτελέσματα να έχουν στατιστική εγκυρότητα.

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

2 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.

2.1 Μορφή αρχείου ίχνους (trace file)

Το αρχείο **out.tr** που δημιουργήθηκε στο προηγουμένως περιέχει πληροφορίες της μορφής:

```
r 0.242572 6 0 cbr 1500 ------- 0 4.0 0.3 63 318
- 0.242576 4 6 cbr 1500 ------- 0 4.0 0.3 64 323
r 0.243792 6 0 cbr 1500 ------- 0 4.0 0.3 64 323
- 0.243797 4 6 cbr 1500 ------- 0 4.0 0.3 65 328
h 0.2448 1 6 cbr 1500 ------- 0 1.0 0.0 68 340
h 0.2448 2 6 cbr 1500 ------- 0 2.0 0.1 68 341
h 0.2448 3 6 cbr 1500 ------- 0 3.0 0.2 68 342
h 0.2448 4 6 cbr 1500 ------- 0 4.0 0.3 68 343
h 0.2448 5 6 cbr 1500 ------- 0 5.0 0.4 68 344
+ 0.244801 1 6 cbr 1500 ------- 0 1.0 0.0 68 340
+ 0.244801 2 6 cbr 1500 ------- 0 2.0 0.1 68 341
d 0.244801 2 6 cbr 1500 ------- 0 2.0 0.1 68 341
+ 0.244801 3 6 cbr 1500 ------- 0 3.0 0.2 68 342
```

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 είσοδος 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 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.

2.2 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 κτλ.

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.

2.3 Μέτρηση πλήθους πακέτων, μέσης καθυστέρησης και ποσότητας δεδομένων

Για να μπορούμε να υπολογίσουμε την χρησιμοποίηση (utilization) 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 αριθμός θα διαιρεθεί με τη διάρκεια της προσομοίωσης, ώστε να υπολογιστεί ο ρυθμός διέλευσης δεδομένων πάνω από το κανάλι (throughput), και αυτή η τιμή θα διαιρεθεί με τον ονομαστικό ρυθμό μετάδοσης του καναλιού ώστε να προκύψει η χρησιμοποίηση (utilization).

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 την οδηγία

```
/^-/&&/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;
}

/^-/&&/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);
}
```

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

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

Προσοχή: Σε περιβάλλον windows ο διερμηνέας awk χρησιμοποιεί ως σύμβολο υποδιαστολής, το σύμβολο που είναι καθορισμένο από τα *regional setting* 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 πρέπει είτε να καθοριστεί το σωστό σύμβολο για την υποδιαστολή (Control Panel -> Regional and Language Options), είτε να αντικατασταθούν όλες οι υποδιαστολές (.) με κόμματα (,) στο αρχείο ίχνους. Η αντικατάσταση αυτή μπορεί να γίνει εύκολα με οποιοδήποτε text editor, αλλά καλύτερα να αποφεύγονται τα wordpad και notepad γιατί καθυστερούν στην εκτέλεση. Προτείνεται η χρήση του word, στο οποίο η αντικατάσταση γίνεται ως εξής: 1. Ανοίγουμε το αρχείο lantest.tr με το word. 2) Πατάμε Ctrl+H 3) Βάζουμε στο πάνω text box την τελεία και στο κάτω το κόμμα. 4) Πατάμε “Replace All” 5) Αποθηκεύουμε το αρχείο.

3 Μελέτη απόδοσης τοπικού δικτύου IEEE 802.3

Με βάση την προσομοίωση που εκτελέσατε στο τμήμα 1 και την ανάλυση του τμήματος 2 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 (Για το 802.3 στα 10Mbps η τιμή αυτή έχει καθοριστεί στα 950ns)
- $e = 2{,}718$

3.1 Επίδραση μεγέθους πλαισίων

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

- 6 σταθμοί συνδεδεμένοι πάνω από ένα τοπικό δίκτυο 802.3
- Ρυθμός μετάδοσης στο LAN 10Mbps

- Καθυστέρηση διάδοσης 950ns
- 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 64Byte (ελάχιστο μέγεθος πλαισίου) μέχρι 1500 byte (μέγιστο μέγεθος πλαισίου). Πρέπει να πραγματοποιηθούν τουλάχιστον δέκα προσομοιώσεις.

Για αυτές τις προσομοιώσεις να γίνει γραφική παράσταση της θεωρητικής και πειραματικής τιμής της χρησιμοποίησης του καναλιού (utilization) σε σχέση με το μέγεθος του πλαισίου. Επίσης να γίνει γραφική παράσταση της μέσης καθυστέρησης που προκαλείται από το πρωτόκολλο 802.3.

Τελικά είναι προτιμότερο να χρησιμοποιούνται μικρά ή μεγάλα πλαίσια;

Θα εμπιστευόσαστε περισσότερο τους θεωρητικούς υπολογισμούς ή την προσομοίωση ως προς την εγκυρότητα των αποτελεσμάτων;

3.2 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 γένει σειρά προσομοιώσεων με τις εξής παραμέτρους:

- Ρυθμός μετάδοσης στο LAN 10Mbps
- Καθυστέρηση διάδοσης 950ns
- Ρυθμός μετάδοσης δεδομένων κάθε σταθμού ίσος με το διπλάσιο του ονομαστικού ρυθμού μετάδοσης του καναλιού προς το πλήθος των σταθμών.
- Μέγεθος πλαισίου 1024 bytes.
- Θα γίνουν τρεις προσομοιώσεις. Στην πρώτη θα συμμετέχουν 6 σταθμοί, στην δεύτερη 20, και στην τρίτη 40.
- Διάρκεια προσομοίωσης: Μεγαλύτερη από 1 δευτερόλεπτο.

Για αυτές τις προσομοιώσεις να γίνει γραφική παράσταση της πειραματικής τιμής της χρησιμοποίησης του καναλιού (utilization), καθώς και της μέσης καθυστέρησης σε σχέση με το πλήθος των σταθμών στο δίκτυο.

Τι παρατηρείτε;

4 Παράρτημα

4.1 Κώδικας αρχείου lantest.tcl για την εκτέλεση της προσομοίωσης

```
###  Arxeio prosomopoiosis gia meleth epidosis prostokollou 802.3
###  gia topika diktya. To senario apoteleitai apo enan arithmo
###  stathmon syndedemonon pano se ena topiko diktyo LAN 802.3.
###  H gennitria kinisis einai CBR (statherou rythmou metadosis me
###  tetoio ryrhmo oste panta na yparxoun paketa gia metadosi.

###
###             0       1       2
###             |       |       |
###          ======================
###             |       |       |
###             3       4       5

### Gia tin ektelesi tou arxeiou pliktrologoume
###    ns lantest.tcl

### Ta apotelesmata vriskontai sta arxeia lantest.nam (Animation)
### kai lantest.tr (Trace File)

### FileName: lantest.tcl
### Author:   D. J. Vergados

set opt(tr)     "lantest.tr"    ;# Trace arxeiou pou perilamvanei
                                ;#ta apotelesmata tis prosomoiosis

set opt(nam)     "lantest.nam"  ;# Arxeio katagrafis pliroforias nam

set opt(seed)   5               ;# Gia ti gennhtria tyxaion arithmon
set opt(stop)   1               ;# Diarkeia prosomoiosis se deyterolepta
set opt(node)   6               ;# Arithmos xriston sto LAN

set opt(qsize)  100             ;# Megethos buffer se kathe stathmo
set opt(bw)     10000000        ;# Rythmos metadoshs sto LAN
set opt(delay)  0.000000950     ;# Kathysterisi metadoshs sto LAN
set opt(packetsize) 1024        ;# Megethos paketou

set opt(rate)   [expr 2* $opt(bw)/ $opt(node)] ;# Rythmos paragogis
                                ;# dedomenonn kathe stathmou.

### Parametroi tou LAN
set opt(ll)     LL
set opt(ifq)    Queue/DropTail
set opt(mac)    Mac/802_3
set opt(chan)   Channel

# Dimiourgia topologias diktyou
proc create-topology {} {
        global ns opt
        global lan node

        set num $opt(node)

        ### Dimiourgia Stathmon tou lan
        for {set i 0} {$i < $num} {incr i} {
                set node($i) [$ns node]
                lappend nodelist $node($i)
        }

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

### Dimiourgia Kinisis metaksy ton stathmon
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_ [expr $opt(packetsize) + 100]
                $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)

                $ns at 0.000$i "$cbr($i) start"
        }
}

###  Anoigma trace arxeiou gia tin katagrafi ton symbanton tis
###  prosomoiosis
proc create-trace {} {
        global ns opt
        set trfd [open $opt(tr) w] ;# Anoigma tou arxeiou
        $ns trace-all $trfd ;# Entoli gia katagrafi gegonoton
        return $trfd
}

###  Anoigma nam arxeiou gia tin animation ton symbanton tis
###  prosomoiosis
proc create-nam-trace {} {
        global ns opt
        set namfd [open $opt(nam) w] ;# Anoigma tou arxeiou
        $ns namtrace-all $namfd ;# Entoli gia katagrafi gegonoton
        return $namfd
}

### Diadikasia termatismou - Klesimo anoixton arxeion
proc finish {} {
        global ns trfd namfd

        $ns flush-trace
        close $trfd
        close $namfd
        exit 0
}

## MAIN ##
set ns [new Simulator]
set trfd [create-trace]
set namfd [create-nam-trace]
create-topology
create-connections
$ns at $opt(stop) "finish"
$ns run
```

4.2 Κώδικας αρχείου lantest.awk για την επεξεργασία των αποτελεσμάτων της προσομοίωσης

```
# Arxeio gia tin analysi dedomenon prosomoiosis pano apo
# topiko diktyo 802.3, kai gia ton ypologismo metron epidosis
#
# Ypologizetai to plithos ton paketon poy elifthisan, to plhthos
# to dedomenon pou elifthisan, kai mesi kathysterisi
#
# Gia na ektelestei to arxeio pliktrologoume
#    awk -f lantest.awk < lantest.tr
#
# FileName: lantest.awk
# Author:   D. J. Vergados

BEGIN {
       data=0;
       packets=0;
       buffersize=10;
       sumDelay=0;
       bufferspace=10000;
}

/^-/&&/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);
}
```