Εργαστηριακή Άσκηση 4

Επίδοση πρωτοκόλλου Go Back N

Σε αυτή την άσκηση θα μελετηθεί η επίδοση του πρωτοκόλλου go back N απουσία σφαλμάτων μετάδοσης. Η λειτουργία του πρωτοκόλλου αυτού περιγράφεται αναλυτικά στο βιβλίο «Δίκτυα Υπολογιστών» (A. Tanenbaum). Το πρωτόκολλο αυτό μπορεί να χρησιμοποιηθεί, εκτός από το στρώμα ζεύξης δεδομένων, και στο στρώμα μεταφοράς, ώστε να παρέχεται υπηρεσία με εγγυημένη παράδοση δεδομένων πάνω από αναξιόπιστο δίκτυο.

Για να αρχίσετε, θα πρέπει να δημιουργήσετε ένα αρχείο, π.χ. "gobackn.tcl", με τον τρόπο που περιγράφεται σε προηγούμενες ασκήσεις. Όπως ειπώθηκε προηγουμένως, αυτός ο κώδικας θα είναι πάντοτε παρόμοιος. Θα πρέπει πάντοτε να δημιουργείτε ένα αντικείμενο προσομοίωσης, να αρχίζετε την προσομοίωση με την ίδια εντολή και, αν θέλετε να τρέχει το NAM και το Xgraph, θα πρέπει να ανοίγετε πάντα αρχεία trace ".tr" ή ".nam", να τα αρχικοποιείτε και να ορίζετε μια διαδικασία που να τα κλείνει.

Ένας εύκολος τρόπος για την ανάλυση των αποτελεσμάτων της προσομοίωσης, είναι η γλώσσα προγραμματισμού awk. Αυτή η γλώσσα είναι μια πολύ απλή γλώσσα script που επιτρέπει την ανάλυση και επεξεργασία δεδομένων από αρχεία με απλό και αποτελεσματικό τρόπο. Για το σκοπό αυτό θα δημιουργήσουμε ένα αρχείο με κατάληξη .awk που θα περιγράφει τις διαδικασίες που απαιτούνται για την ανάλυση των δεδομένων. Το αρχείο αυτό εκτελείται με τη βοήθεια του προγράμματος awk.exe.

Σημείωση: Είναι σκόπιμο να ανατρέξετε στο φυλλάδιο της Εργαστηριακής Άσκησης 1, για να θυμηθείτε τις διαδικασίες με τις οποίες σώζονται και τρέχουν τα αρχεία. Θυμίζουμε ότι πριν αρχίζετε την προσομοίωση με ένα αρχείο "*.tcl" πρέπει κάθε φορά να το σώζετε (File > Save). Για να εμφανιστεί ο δρομέας (cursor) στο "Command Prompt", όταν είναι ανοιχτά το NAM ή το Xgraph, πρέπει πρώτα να κλείσετε τα παράθυρά τους. Κλείστε τα μόνο πατώντας το "X" δεξιά επάνω στην μπάρα του παραθύρου , όχι από το μενού File. Στο τέλος του εργαστηρίου κάντε "log off" πριν φύγετε. Ο ολοκληρωμένος κώδικας στο τέλος της Ενότητας παρατίθεται για συμβουλευτικό σκοπό και μόνο, π.χ. σε περίπτωση λάθους κτλ. Στο επόμενο εργαστήριο θα πρέπει να παραδώσετε μια αναφορά με απαντήσεις και σχόλια σχετικά με τις ερωτήσεις αυτής της Εργαστηριακής Άσκησης. Είναι απαραίτητο επίσης να φέρνετε μαζί σας από μία δισκέτα.

1. Σενάριο προσομοίωσης.

1.1 Αρχικοποίηση Προσομοίωσης – Δημιουργία αρχείου ίχνους

Αρχικά πρέπει να δημιουργηθεί το αντικείμενο της προσομοίωσης.

```
#Create a simulator object
set ns [new Simulator]
```

Στις προηγούμενες ασκήσεις μάθαμε πώς να δημιουργούμε ένα animation της προσομοίωσης, ώστε να επαληθεύσουμε την ορθότητα του σεναρίου που δημιουργήσαμε, και να δημιουργούμε αρχεία για γραφική απεικόνιση με το Xgraph. Σε αυτή την άσκηση θα μάθουμε να χρησιμοποιούμε τα αρχεία ίχνους (trace file). Όπως θα δούμε και στη συνέχεια, στο αρχείο ίχνους αποθηκεύονται όλα τα γεγονότα που θα συμβούν κατά τη διάρκεια της προσομοίωσης.

```
#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 της προσομοίωσης είναι η απλούστερη δυνατή, με δύο κόμβους n1 και n2 να συνδέονται μεταξύ τους με μία πλήρως αμφίδρομη ζεύξη.

```
#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
```

1.3 Το στρώμα μεταφοράς

Το πρωτόκολλο go back N εμπεριέχεται στο πρωτόκολλο μεταφοράς TCP. Για το λόγο αυτό θα δημιουργήσουμε μια TCP σύνδεση από τον κόμβο n1 στον κόμβο n2. Αυτό γίνεται με την εντολή

```
set tcp0 [$ns create-connection TCP/Reno $n0 TCPSink $n1 0]
```

Το μέγεθος του παραθύρου του go back N (βλέπε βιβλίο) καθορίζεται με την εντολή

```
$tcp0 set window_ 5
```

Το μέγεθος του tcp πακέτου καθορίζεται με την εντολή

```
$tcp0 set packetSize_ 1000
```

1.3 Το στρώμα εφαρμογής.

Ως εφαρμογή που θα δημιουργήσει την κίνηση θεωρούμε τη μεταφορά ενός αρχείου απείρου μεγέθους με FTP, έτσι ώστε να υπάρχει πάντα πληροφορία για μετάδοση. Η δημιουργία της γεννήτριας κίνησης FTP, η σύνδεσή της με το στρώμα μεταφοράς και η εκκίνησή της γίνονται με τις παρακάτω εντολές:

```
set ftp0 [$tcp0 attach-app FTP]
$ns at 0.0 "$ftp0 start"
```

1.4 Εκτέλεση του σεναρίου

Με βάση τα παραπάνω δημιουργούμε το αρχείο **gobackn.tcl**, με το εξής περιεχόμενο:

```
#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
```

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

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

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 προηγουμένως περιέχει πληροφορίες της μορφής:

```
+ 0.12688 0 1 tcp 1040 ------- 0 0.0 1.0 25 47
- 0.12688 0 1 tcp 1040 ------- 0 0.0 1.0 25 47
r 0.127712 1 0 ack 40 ------- 0 1.0 0.0 21 43
+ 0.127712 0 1 tcp 1040 ------- 0 0.0 1.0 26 48
- 0.127712 0 1 tcp 1040 ------- 0 0.0 1.0 26 48
r 0.135216 0 1 tcp 1040 ------- 0 0.0 1.0 22 44
+ 0.135216 1 0 ack 40 ------- 0 1.0 0.0 22 49
- 0.135216 1 0 ack 40 ------- 0 1.0 0.0 22 49
r 0.136048 0 1 tcp 1040 ------- 0 0.0 1.0 23 45
```

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 "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 την διεύθυνση αποστολέα και προορισμού (IP.port),

από τον αύξοντα αριθμό (sequence number) του πακέτου και τέλος από έναν μοναδικό αριθμό (unique number) του πακέτου.

2.2 Ανάλυση με την 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 ανοιχθούν 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/&&/tcp/ {
     packets++;
     data+=$6;
}
```

ορίζει ότι, όταν συναντήσουμε μια γραμμή του αρχείου εισόδου που να ξεκινάει με τον χαρακτήρα “r” (/^r/) και (&&) που περιλαμβάνει τη λέξη tcp (/tcp/), τότε η τιμή της μεταβλητής packets αυξάνεται κατά 1, και η τιμή της μεταβλητής data αυξάνεται κατά την τιμή που βρίσκεται στην έκτη λέξη ($6) της γραμμής την οποία εξετάζουμε.

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 Υπολογισμός του πλήθους των πακέτων και της ποσότητας των δεδομένων που ελήφθησαν

Για να μπορούμε να υπολογίσουμε την επίδοση του πρωτοκόλλου go back N πρέπει να μετρήσουμε την ποσότητα των δεδομένων που ελήφθησαν κατά τη διάρκεια της προσομοίωσης. Για κάθε πακέτο που έλαβε 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 TCP. Αυτές οι γραμμές αρχίζουν με τον χαρακτήρα “r” και περιλαμβάνουν την λέξη tcp. 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. Έτσι δημιουργείται το παρακάτω πρόγραμμα awk:

```
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);
}
```

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

```
awk.exe -f trace.awk < out.tr
```

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

3. Μελέτη απόδοσης go back N

Με βάση την προσομοίωση που εκτελέσατε στο τμήμα 1 και την ανάλυση του τμήματος 2 να επαληθεύσετε κατά πόσον ισχύει ή όχι η εξίσωση

$$\eta = \min\left\{\frac{W \times TRANSP}{S}, 1\right\}$$

Όπου:

- S = TRANSP + 2xPROP + TRANSA
- TRANSP είναι ο χρόνος μετάδοσης του πακέτου δεδομένων, (μήκος πακέτου) /(ρυθμός μετάδοσης)
- PROP είναι η καθυστέρηση διάδοσης του πακέτου
- TRANSA είναι ο χρόνος μετάδοσης της επαλήθευσης.
- W είναι το μέγεθος του παραθύρου
- η είναι η απόδοση του πρωτοκόλλου (πραγματικός ρυθμός μετάδοσης δεδομένων)/(ρυθμός μετάδοσης ζεύξης)

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

(β) Με βάση την παραπάνω εξίσωση, υπολογίστε τη 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;

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

(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.

(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;