

ΜΥΥ703: Δίκτυα Υπολογιστών Ι

Εργαστήρια

Άσκηση 1^η
Εισαγωγή στον Network Simulator (Ns-2)

Πανεπιστήμιο Ιωαννίνων
Τμήμα Μηχανικών Η/Υ και Πληροφορικής
Διδάσκων: Παπαπέτρου Ευάγγελος

1 Εισαγωγή

Η άσκηση έχει στόχο την εκμάθηση των πρώτων βασικών εντολών για τη δημιουργία μιας προσομοίωσης με τον προσομοιωτή Network Simulator - ns. Ο ns¹ είναι ένας γενικής χρήσης προσομοιωτής δικτύων που αρχικά αναπτύχθηκε από το πανεπιστήμιο UC at Berkeley. Η λογική του είναι να παρέχει στους χρήστες κλάσεις που υλοποιούν τα βασικά συστατικά στοιχεία ενός δικτύου όπως κόμβοι, ζεύξεις, πηγές πληροφορίας, πρωτόκολλα, κ.α., ώστε αυτοί να μπορούν εύκολα να δημιουργήσουν το επιθυμητό δίκτυο και να προσομοιώσουν τη λειτουργία του.

Η δημιουργία μια προσομοίωσης στον ns γίνεται με απλές εντολές της γλώσσας Tcl, που είναι μια γλώσσα φιλική προς τον χρήστη. Οι εντολές αυτές αποθηκεύονται σε ένα απλό αρχείο κειμένου με κατάληξη tcl (π.χ. example.tcl), το οποίο και καλείται αρχείο προσομοίωσης (simulation script). Στην ενότητα 2 παρουσιάζονται οι βασικές εντολές για τη δημιουργία μιας προσομοίωσης.² Μια προσομοίωση εκτελείται όταν δοθεί η εντολή:

```
user@host:...$ ns example.tcl
```

όπου ns είναι το εκτελέσιμο αρχείο του ns και example.tcl το αρχείο προσομοίωσης. Στα εργαστήρια του Τμήματος Μηχανικών Η/Υ και Πληροφορικής το εκτελέσιμο του ns βρίσκεται στο path /usr/bin/. Για να εκτελέσετε μια προσομοίωση, πρέπει να εκτελέσετε τον ns από το directory στο οποίο βρίσκεται και το αρχείο example.tcl. Με άλλα λόγια, αν το αρχείο της προσομοίωσής σας βρίσκεται στο directory lab1, πρέπει να εκτελέσετε τις εντολές:

```
user@host:~$ cd lab1
```

```
user@host:~/lab1$ /usr/bin/ns example.tcl
```

2 Προγραμματισμός στον Ns-2

Στο σημείο αυτό θα δούμε ένα απλό παράδειγμα για το πως μπορεί ένας χρήστης να δημιουργήσει ένα απλό αρχείο προσομοίωσης (simulation script) που θα χρησιμοποιηθεί με τον ns. Το παράδειγμα που θα εξετάσουμε αναφέρεται στις εντολές που υπάρχουν στο αρχείο lab1_tutorial.tcl (διατίθεται μαζί με την παρούσα εκφώνηση). Όταν κατασκευάζουμε ένα αρχείο προσομοίωσης στόχος είναι να συνδυαστούν οι κλάσεις που είναι διαθέσιμες από τον ns για να δημιουργηθεί ένα δίκτυο, το οποίο και θα προσομοιωθεί. Μερικές από τις κλάσεις που είναι διαθέσιμες στον ns είναι: κόμβος (node), αμφίδρομη ζεύξη (duplex-link), πηγή πληροφορίας (Application), κ.λ.π.

¹<http://www.isi.edu/nsnam/ns/>

²Επίσης μπορείτε να βρείτε παραδείγματα στην ιστοσελίδα του ns, <http://www.isi.edu/nsnam/ns/tutorial/>

Στο παράδειγμα που θα εξετάσουμε θα δημιουργήσουμε δύο κόμβους, συνδεδεμένους με μία αμφίδρομη ζεύξη, την οποία και θα χρησιμοποιήσουμε για να στείλουμε πληροφορία από τον ένα στον άλλο κόμβο. Το πρώτο βήμα είναι να δημιουργήσουμε το βασικό αντικείμενο του προσομοιωτή. *Η δήλωση αυτή χρησιμοποιείται σε κάθε προσομοίωση.*

```
set ns [new Simulator]
```

Το επόμενο βήμα είναι να δημιουργήσουμε δύο κόμβους

```
set n0 [$ns node]
set n1 [$ns node]
```

Με τις παραπάνω δηλώσεις δημιουργούνται δύο μεταβλητές (n0 και n1), στις οποίες αποθηκεύονται τα αντικείμενα κόμβοι που δημιουργήσαμε. Κατόπιν συνδέουμε τους δύο κόμβους με μία αμφίδρομη ζεύξη

```
$ns duplex-link $n0 $n1 1.5Mb 10ms DropTail
```

Με την παραπάνω εντολή δηλώνουμε ότι η ζεύξη υλοποιείται μεταξύ των κόμβων n0 και n1, ότι η ταχύτητα μετάδοσης πληροφορίας στη ζεύξη είναι $C = 1,5Mbps$ και ότι η καθυστέρηση διάδοσης πάνω σε αυτήν είναι $10ms$. Τέλος, δηλώνουμε ότι κάθε κόμβος που συνδέεται στη ζεύξη (στην περίπτωση μας οι κόμβοι n0 και n1) διατηρεί μια ουρά αποθήκευσης πακέτων (buffer), στην οποία η αρχή εξυπηρέτησης είναι FIFO (First In First Out) και όταν αυτή γεμίσει, απορρίπτει το τελευταίο πακέτο (DropTail).³ Στο σημείο αυτό έχουμε ολοκληρώσει τη δημιουργία της τοπολογίας του δικτύου. Περισσότερο πολύπλοκες τοπολογίες φυσικά είναι εφικτές.

Στο επόμενο στάδιο πρέπει να εισάγουμε στο δίκτυο τηλεπικοινωνιακή κίνηση. Αρχικά δημιουργούμε μια πηγή πληροφορίας (δηλαδή μια εφαρμογή) που θα αποστέλλει δεδομένα από τον ένα κόμβο στον άλλο. Στην περίπτωση μας χρησιμοποιούμε μία εφαρμογή που στέλνει δεδομένα με σταθερό ρυθμό (Constant BitRate, CBR), δηλαδή σε ένα καθορισμένο χρονικό διάστημα στέλνει πάντα τον ίδιο αριθμό πακέτων. Στην εφαρμογή αυτή δίνουμε το όνομα cbr0

```
set cbr0 [new Application/Traffic/CBR]
```

³Πρέπει να σημειωθεί ότι όπως φαίνεται και στο παράδειγμα της κλάσης duplex-link, υπάρχει η δυνατότητα να καθοριστούν μια σειρά παραμέτρων των κλάσεων, δίνοντας έτσι τη δυνατότητα στο χρήστη να μελετήσει την επίδρασή τους στην απόδοση του συστήματος.

Υπάρχει η δυνατότητα να καθορίσουμε κάποια χαρακτηριστικά για την εφαρμογή `cbr0` όπως το μέγεθος του πακέτου (`packetSize_`) και το χρονικό διάστημα ανάμεσα στις αποστολές των πακέτων (`interval_`). Στο παράδειγμά μας, το μέγεθος του πακέτου τίθεται ίσο με 48 bytes και το χρονικό διάστημα ίσο με 0.01 seconds

```
set ps 48
set int 0.01
$cbr0 set packetSize_ $ps
$cbr0 set interval_ $int
```

Με τις παραπάνω εντολές, πρώτα δημιουργήσαμε δύο μεταβλητές (`ps` και `int`) και τους δώσαμε τις τιμές που μας ενδιαφέρουν και μετά θέσαμε στις παραμέτρους που μας ενδιαφέρουν τις τιμές που περιείχαν οι μεταβλητές. Όπως είναι γνωστό, κάθε εφαρμογή για να μπορέσει να αποστείλει δεδομένα πρέπει να χρησιμοποιήσει το επίπεδο μεταφοράς (μοντέλο OSI). Επομένως, πρέπει να δημιουργήσουμε έναν πράκτορα (`agent`) που θα παίζει το ρόλο του επιπέδου μεταφοράς

```
set udp0 [new Agent/UDP]
```

και να συνδέσουμε την εφαρμογή με τον πράκτορα αυτό.

```
$cbr0 attach-agent $udp0
```

Το επόμενο βήμα είναι να δημιουργούμε έναν επιπλέον πράκτορα που θα λαμβάνει τα δεδομένα στον δεύτερο κόμβο.

```
set null0 [new Agent/Null]
```

Ο πράκτορας έχει το όνομα `null0` και παίζει το ρόλο της εφαρμογής που λαμβάνει και να καταναλώνει τα πακέτα⁴. Τέλος, θα πρέπει να δηλώσουμε ποιός πράκτορας θα βρίσκεται σε ποιόν κόμβο. Με άλλα λόγια πρέπει να δηλώσουμε ποια εφαρμογή θα τρέχει σε ποιόν κόμβο. Αυτό γίνεται με τις εντολές

```
$ns attach-agent $n0 $udp0
$ns attach-agent $n1 $null0
```

όπου δηλώνεται ότι η εφαρμογή `cbr0`, μέσω του πράκτορα `udp0` θα βρίσκεται

⁴Για συνδέσεις TCP ο πράκτορας στον κόμβο-αποστολέα ονομάζεται `Agent/TCP` και ο πράκτορας στον κόμβο-παραλήπτη `Agent/TCPSink`.

στον κόμβο `n0`, ενώ η εφαρμογή που θα λαμβάνει τα πακέτα θα βρίσκεται στον κόμβο `n1`. Στο σημείο αυτό έχουμε ολοκληρώσει τη δημιουργία των εφαρμογών που θα δημιουργούν και θα καταναλώνουν την τηλεπικοινωνιακή κίνηση.

Το επόμενο βήμα είναι να δημιουργήσουμε τη σύνδεση μεταξύ των εφαρμογών, δηλαδή να δηλώσουμε ότι η εφαρμογή `cbr0`, μέσω του πράκτορα `udp0` πρέπει να στέλνει δεδομένα στον πράκτορα `null0`

```
$ns connect $udp0 $null0
```

Στη συνέχεια πρέπει να δηλώσουμε πότε θα ξεκινήσει η ροή της πληροφορίας από την πηγή ως εξής

```
$ns at 1.0 "$cbr0 start"
```

όπου δηλώνουμε ότι η ροή πληροφορίας θα ξεκινήσει *1sec* μετά την έναρξη της προσομοίωσης⁵. Μπορούμε αντίστοιχα να ορίσουμε και το χρόνο τερματισμού της ροής πληροφορίας ως εξής

```
$ns at 4.0 "$cbr0 stop"
```

Στο σημείο αυτό μπορούμε να κάνουμε δύο απλούς υπολογισμούς σχετικά με την ροή πληροφορίας `cbr0`. Πρώτον, υπολογίζουμε και τυπώνουμε στην οθόνη τον ρυθμό αποστολής πακέτων ανά `second`

```
puts "cbr0 produces [expr (1/$int)] packets per second"
```

Δεύτερον υπολογίζουμε και τυπώνουμε στην οθόνη το ρυθμό αποστολής `bytes` ανά `second`

```
puts "cbr0 produces [expr (1/$int)*$ps] bytes per second"
```

Για τον τερματισμό της προσομοίωσης καλούμε μία διαδικασία με όνομα `finish` που θα ορίσουμε αργότερα. Η κλήση της γίνεται στα `5 seconds` που δηλώνει το χρόνο της προσομοίωσης

```
$ns at 5.0 "finish"
```

Η διαδικασία `finish` έχει τη δομή

⁵Προσοχή: Οι χρόνοι που αναφέρονται αφορούν το χρόνο της προσομοίωσης και όχι τον πραγματικό χρόνο καθώς ο `ns` είναι ένας `event-driven` προσομοιωτής

```
proc finish {
    exit 0
}
```

Τελευταίο βήμα είναι η εκκίνηση της προσομοίωσης

```
$ns run
```

2.1 Χρήση Οπτικής Προσομοίωσης

Ο ns δίνει τη δυνατότητα οπτικοποίησης της προσομοίωσης που παράγει μέσω του εργαλείου nam (Network Animator). Στη συνέχεια θα περιγράψουμε πως μπορεί να γίνει αυτό, προσθέτοντας ορισμένες γραμμές κώδικα στο προηγούμενο παράδειγμα. Πρώτα πρέπει να δημιουργηθεί ένα αρχείο στο οποίο ο ns αποθηκεύει δεδομένα σχετικά με την προσομοίωση, τα οποία κατόπιν διαβάζει το εργαλείο nam. Η δημιουργία του αρχείου (`lab1_tutorial.nam`) γίνεται με την εντολή

```
set nf [open lab1_tutorial.nam w]
```

Κατόπιν πρέπει να δηλώσουμε στον ns ότι θέλουμε να δημιουργήσει τα δεδομένα που θα χρησιμοποιήσει ο nam. Αυτό γίνεται με την εντολή

```
$ns namtrace-all $nf
```

Κατά τη διάρκεια της προσομοίωσης, ο ns θα γράφει στο αρχείο που δηλώσαμε όλα τα απαραίτητα δεδομένα. Με το τέλος της προσομοίωσης, δηλαδή στην ρουτίνα `finish` θα πρέπει να πούμε στον ns να κλείσει το αρχείο και να καλέσει το εργαλείο nam που θα οπτικοποιήσει την προσομοίωση. Αυτό γίνεται προσθέτοντας τις παρακάτω εντολές στη ρουτίνα `finish`

```
global ns nf
$ns flush-trace
close $nf
exec /usr/bin/nam lab1_tutorial.nam &
```

Η πρώτη εντολή δηλώνει τις μεταβλητές του ns του αρχείου ώστε να είναι προσβάσιμες και μέσα από τη ρουτίνα `finish`. Με τις δύο επόμενες εντολές γράφουμε όλα τα δεδομένα στο αρχείο και το κλείνουμε και τέλος καλούμε τον nam για να ξεκινήσει η οπτικοποίηση της προσομοίωσης.

Τέλος, για την βελτίωση της οπτικής αναπαράστασης μπορούμε να ορίσουμε και χρώματα για κάθε επικοινωνία, δηλαδή για κάθε εφαρμογή που αποστέλλει δεδομένα. Αυτό γίνεται προσθέτοντας στην προσομοίωσή μας τις εντολές

```
$ns color 1 Red
$udp0 set class_ 1
```

Με τις εντολές αυτές, τα πακέτα που στέλνει η εφαρμογή που είναι συνδεδεμένη στον πράκτορα `udp0` θα έχουν κόκκινο χρώμα.

3 Οδηγίες για την άσκηση

Όπως αναφέρθηκε η εκτέλεση μιας προσομοίωσης με τον `ns` προϋποθέτει την ύπαρξη κάποιου αρχείου προσομοίωσης στη γλώσσα `OTcl`. Αρχικά, για τις ανάγκες της άσκησης διατίθεται ένα απλό δείγμα προσομοίωσης, το `lab1_tutorial.tcl` για την μελέτη των πρώτων βασικών εντολών του προσομοιωτή. Για να εκτελέσετε το αρχείο θα πρέπει να το αντιγράψετε στο λογαριασμό σας και να πληκτρολογήσετε την εντολή:

```
user@host:...$ /usr/bin/ns lab1_tutorial.tcl
```

Μετά το τέλος της προσομοίωσης τα αποτελέσματα αποθηκεύονται με την προέκταση `nam`⁶ αναφορικά με το γραφικό περιβάλλον `Network Animator (nam)`. Πιο συγκεκριμένα η προσομοίωση με το αρχείο `lab1_tutorial.tcl` παράγει το αρχείο `lab1_tutorial.nam`. Μετά την εκτέλεση της παραπάνω εντολής θα παρατηρήσετε την εκκίνηση του προγράμματος `nam` που εμφανίζει το αποτέλεσμα της προσομοίωσης σε γραφικό περιβάλλον. Για το συγκεκριμένο παράδειγμα θα παρατηρήσετε δύο κύκλους (κόμβους), αριθμημένους με 0 και 1, που συνδέονται με μία γραμμή. Εάν επιλέξετε την εκκίνηση του γραφικού περιβάλλοντος θα παρατηρήσετε μία σταθερή ροή (κόκκινων) πακέτων από τον κόμβο 0 στον κόμβο 1 με σταθερό ρυθμό από το 1^ο έως και το 4^ο δευτερόλεπτο.

4 Παραδείγματα ζητούμενων

Μαζί με την παρούσα εκφώνηση σας έχουν δοθεί και τρία αρχεία προσομοίωσης με ονόματα `lab1_1.tcl`, `lab1_2.tcl` και `lab1_3.tcl`. Αφού εκτελέσετε και μελετήσετε το παράδειγμα που σας έχει ήδη δοθεί θα πρέπει να συμπληρώσετε τις απαραίτητες γραμμές κώδικα σε ένα από τα παραπάνω αρχεία `tcl`.

⁶Τα αρχεία αποθηκεύονται στο φάκελο μέσα από τον οποίο εκτελέσατε την προσομοίωση, ανεξάρτητα από το αν το εκτελέσιμο του `ns` βρίσκεται σε άλλο μονοπάτι.

Ενδεικτικές ερωτήσεις:

- 1η ερώτηση. Αυξήστε την παράμετρο `packetSize` σε 300 και μετά σε 600. Τι παρατηρείτε; Υπολογίστε το ρυθμό αποστολής πακέτων ανά δευτερόλεπτο και το ρυθμό αποστολής bytes ανά δευτερόλεπτο για κάθε περίπτωση.
- 2η ερώτηση. Διατηρώντας την παράμετρο `packetSize` ίση με 300 αυξήστε την παράμετρο `interval`. Συγκεκριμένα θα πρέπει να λαμβάνει τις τιμές 0.05 και 0.1. Τι παρατηρείτε; Υπολογίστε και πάλι το ρυθμό αποστολής πακέτων ανά δευτερόλεπτο και το ρυθμό αποστολής bytes ανά δευτερόλεπτο για κάθε περίπτωση.
- 3η ερώτηση. Διατηρώντας την παράμετρο `packetSize` ίση με 300 και την παράμετρο `interval` ίση με 0.05 αυξήστε το `bandwidth` των ζεύξεων από τις οποίες διέρχεται μια ροή `cbr` σε 3Mbps. Τι παρατηρείτε;
- 4η ερώτηση. Διατηρώντας την παράμετρο `packetSize` ίση με 300, την παράμετρο `interval` ίση με 0.05 και το `bandwidth` των ζεύξεων από τις οποίες διέρχεται μια ροή `cbr` ίσο με 3Mbps μειώστε το `delay` στις ζεύξεις αυτές σε 5 ms. Τι παρατηρείτε;