

Δίκτυα Υπολογιστών

Εργαστήρια

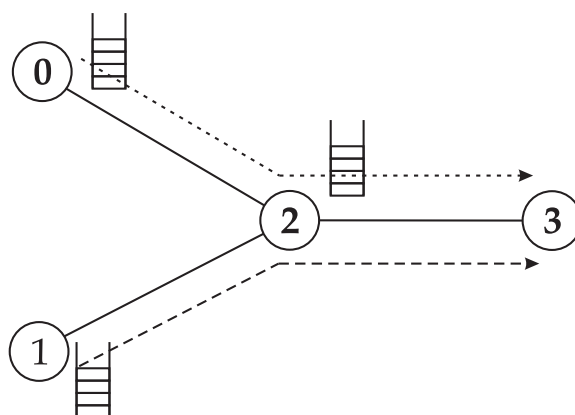
Άσκηση 2^η

Βασικές έννοιες δικτύων, ανάλυση
και επεξεργασία αποτελεσμάτων στον ns-2

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

1 Εισαγωγή - Περιγραφή άσκησης

Η άσκηση αυτή έχει στόχο την εκμάθηση βασικών εννοιών στην περιοχή των δικτύων υπολογιστών. Για το σκοπό αυτό θα χρησιμοποιηθεί ο προσομοιωτής ns και συγκεκριμένα η τοπολογία του Σχ.1. Στο δίκτυο του σχήματος οι κόμβοι 0 και 1 αποστέλλουν δεδομένα στον κόμβο 3. Τα δεδομένα αυτά διέρχονται από τον κόμβο 2, ο οποίος λειτουργεί ως δρομολογητής και προωθεί τα πακέτα στον κόμβο 3. Βασική προϋπόθεση για την ορθή λειτουργία ενός δικτύου είναι ο ρυθμός νέων πακέτων που εισέρχονται ή δημιουργούνται σε έναν κόμβο να είναι μικρότερος ή ίσος με τον ρυθμό με τον οποίο εξυπηρετούνται από αυτόν. Για τους κόμβους 0 και 1 που δημιουργούν νέα πακέτα, η παραπάνω αρχή σημαίνει ότι οι εφαρμογές που βρίσκονται στους κόμβους αυτούς πρέπει να δημιουργούν πακέτα με μικρότερο ρυθμό από αυτόν με τον οποίο το δίκτυο μπορεί να τα μεταδώσει. Αν και είναι απλό να τηρηθεί η παραπάνω προϋπόθεση σε έναν κόμβο που δημιουργεί τα πακέτα, κάποια πακέτα είναι δυνατόν να πρέπει να αποθηκευτούν προσωρινά λόγω στοχαστικών μεταβολών στους χρόνους δημιουργίας νέων πακέτων. Το τελευταίο αυτό χαρακτηριστικό είναι εγγενές στη συμπεριφορά των εφαρμογών¹, οι οποίες και δημιουργούν τα πακέτα. Το χαρακτηριστικό αυτό αποτελεί το λόγο που επιβάλλει τη χρήση ουρών προσωρινής αποθήκευσης (buffers) για τα εξερχόμενα πακέτα στους κόμβους 0 και 1. Ο κόμ-



Σχήμα 1: Τοπολογία αναφοράς

βος 2 στην ορολογία των δικτύων καλείται bottleneck, γιατί στο σημείο αυτό η χωρητικότητα της εξερχόμενης σύνδεσης είναι μικρότερη αυτής των εισερχόμενων συνδέσεων. Και στον κόμβο αυτό, για τη σωστή λειτουργία του δικτύου, πρέπει ο ρυθμός των εισερχόμενων πακέτων να είναι μικρότερος ή ίσος με το

¹Π.χ. ο τρόπος με τον οποίο μια εφαρμογή τηλεφωνίας δημιουργεί πακέτα είναι στοχαστικός και εξαρτάται από το χρήστη της εφαρμογής.

ρυθμό εξυπηρέτησής τους. Ωστόσο και πάλι πακέτα μπορεί να αποθηκευτούν στην ουρά εξόδου οπότε και καθυστερούν (queueing delay). Η αναγκαιότητα μιας ουράς σε κάποιον ενδιάμεσο κόμβο του δικτύου οφείλεται στο γεγονός ότι το ίδιο το δίκτυο μπορεί να παράγει τυχαιότητα στους χρόνους που χρειάζεται για να μεταδώσει τα δεδομένα μέχρι κάποιον κόμβο. Η τυχαιότητα αυτή μπορεί να οφείλεται είτε στη λειτουργία του δικτύου (π.χ. χρήση μηχανισμών ανταγωνισμού) ή στους τυχαίους χρόνους καθυστέρησης (queueing delays) στους ενδιάμεσους κόμβους.

Η λειτουργία μιας ουράς σε έναν ενδιάμεσο κόμβο είναι πολύ σημαντική καθώς μπορεί να επιβάλλει σημαντικές καθυστερήσεις (queueing delay) ή ακόμα και την απόρριψη πακέτων όταν αυτή γεμίσει. Αποτέλεσμα των φαινομένων αυτών είναι η μείωση της ποιότητας επικοινωνίας που το δίκτυο προσφέρει. Η λειτουργία της ουράς εξαρτάται από το μέγεθός της, το ρυθμό εισερχομένων πακέτων, το ρυθμό εξυπηρέτησης της εξερχόμενης γραμμής, το μέγεθος των πακέτων και την αρχή εξυπηρέτησης. Στη συγκεκριμένη άσκηση θα μελετηθεί η συμπεριφορά του δικτύου μεταβάλλοντας κάποιες από τις παραπάνω παραμέτρους. Πιο συγκεκριμένα για να αξιολογηθεί η συμπεριφορά του δικτύου θα μελετηθούν τα παρακάτω μεγέθη:

- Ποσοστό επιτυχούς παράδοσης πακέτων (Delivery Ratio)
- Μέση καθυστέρηση στην ουρά αναμονής (Mean Queueing Delay)
- Μέση καθυστέρηση πακέτου (Mean Packet Delay)
- Ποσοστό αξιοποίησης ζεύξης (Link Utilization)

2 Οδηγίες για την άσκηση

Μαζί με την παρούσα εκφώνηση διανέμεται και το αρχείο `lab2.tcl`. Το αρχείο αυτό προσομοιώνει την τοπολογία του Σχ.1. Οι κόμβοι 0 και 1 είναι αποστολείς πακέτων με σταθερό ρυθμό ενώ ο κόμβος 2 είναι ενδιάμεσος κόμβος που προωθεί τα πακέτα προς τον κόμβο 3, ο οποίος είναι και ο τελικός προορισμός.² Το αρχείο `lab2.tcl` έχει κατασκευαστεί με τις βασικές εντολές που περιγράφηκαν στην προηγούμενη άσκηση. Ωστόσο έχουν προστεθεί κάποιες νέες εντολές που αξίζει να σχολιαστούν.

Στον ns μία ουρά αναμονής κατασκευάζεται αυτόματα σε κάθε ζεύξη που συνδέει δύο κόμβους και με προκαθορισμένο μέγεθος ίσο με 50 πακέτα. Μετά τον ορισμό των ζεύξεων μεταξύ των κόμβων δηλώνουμε στον ns ότι επιθυμούμε να παρατηρήσουμε (στον nam) τις ουρές που βρίσκονται μεταξύ των κόμβων {0,2},{1,2} και {2,3}. Αυτό γίνεται με τις εντολές:

```
$ns duplex-link-op $n2 $n3 queuePos 0.5
$ns duplex-link-op $n0 $n2 queuePos 0.8
```

²Για την ευκολότερη παρατήρηση της οπτικής προσομοίωσης, τα πακέτα του κόμβου 0 έχουν μπλε χρώμα, ενώ τα πακέτα του κόμβου 1 έχουν κόκκινο χρώμα.

```
$ns duplex-link-op $n1 $n3 queuePos 0.45
```

Πρέπει να σημειωθεί ότι ιδιαίτερο ενδιαφέρον παρουσιάζει η ουρά μεταξύ των κόμβων {2,3} και στη συνέχεια θα επικεντρώσουμε την προσοχή μας σε αυτήν. Κατόπιν, θέτουμε το μέγεθος της ουράς αυτής ίσο με 10. Αυτό γίνεται με την εντολή:

```
$ns queue-limit $n2 $n3 10
```

Υπενθυμίζεται ότι χρησιμοποιήθηκε η ουρά τύπου DropTail. Αυτό σημαίνει ότι αν η ουρά χρειαστεί να αποθηκεύσει παραπάνω από 10 πακέτα τότε θα απορρίψει το επόμενο πακέτο που θα λάβει. Επίσης η αρχή εξυπηρέτησης της ουράς είναι FIFO (First In First Out), δηλαδή τα πακέτα εξυπηρετούνται με τη σειρά με την οποία εισέρχονται σε αυτήν. Η πολιτική εξυπηρέτησης της ουράς DropTail μπορεί να δημιουργήσει προβλήματα άνισης συμπεριφοράς σε πακέτα που έρχονται από διαφορετικούς κόμβους (π.χ. στο παράδειγμα μπορεί να απορρίπτονται μόνο πακέτα από τον κόμβο 0 ή 1). Ένας τρόπος να εισαχθεί περισσότερη δικαιοσύνη στην πολιτική της ουράς αναμονής είναι να αλλάξει το είδος της. Προαιρετικά, μπορείτε να χρησιμοποιήσετε μια ουρά που ονομάζεται Fair Queue. Αυτό μπορεί να γίνει αν αντικαταστήσετε τη λέξη DropTail με την λέξη FQ στο αρχείο lab2.tcl.

2.1 Trace

Ένα καινούργιο στοιχείο που χρησιμοποιείται στο αρχείο lab2.tcl είναι η δυνατότητα του ns να καταγράφει, σε αρχεία κειμένου, δεδομένα σχετικά με τη λειτουργία του δικτύου. Για την ακρίβεια ο ns καταγράφει στοιχεία σχετικά με τα πακέτα που μεταδίδονται μέσα στο δίκτυο. Η λειτουργία αυτή καλείται *trace* και για να την ενεργοποιήσουμε ακολουθούμε τα παρακάτω βήματα. Πρώτα χρησιμοποιούμε την εντολή:

```
set f [open lab2.tr w]
```

για να δηλώσουμε ότι το αρχείο καταγραφής θα έχει όνομα lab2.tr. Κατόπιν δηλώνουμε στον ns ότι επιθυμούμε να ενεργοποιήσει τη διαδικασία trace με την εντολή:

```
$ns trace-all $f
```

Τέλος, κατά τον τερματισμό της προσομοίωσης κλείνουμε το αρχείο καταγραφής με την εντολή:

```
close $f
```

Η λειτουργία trace είναι πολύ σημαντική γιατί δίνει τη δυνατότητα της καταγραφής της πορείας κάθε πακέτου μέσα στο δίκτυο. Τα δεδομένα αυτά μπορεί κανείς να τα *επεξεργαστεί* για να βγάλει σημαντικά συμπεράσματα για τη συμπεριφορά του δικτύου και την απόδοσή του. Στη συνέχεια της άσκησης θα επεξεργαστούμε τα δεδομένα κάθε προσομοίωσης με τη βοήθεια του script lab.sh που διανέμεται με την εκφώνηση της άσκησης αυτής. Περισσότερες πληροφορίες σχετικά

με τη δομή των αρχείων καταγραφής μπορείτε να βρείτε στο Παράρτημα Α' της παρούσας εκφώνησης.

3 Μεθοδολογία

Όπως έχει αναφερθεί για να εκτελέσετε μια προσομοίωση θα πρέπει να αντιγράψετε το αρχείο `lab2.tcl` στο λογαριασμό σας και να πληκτρολογήσετε την εντολή:

```
user@host:...$ /usr/bin/ns lab2.tcl
```

Μετά το τέλος της προσομοίωσης τα αποτελέσματα αποθηκεύονται σε ένα αρχείο κειμένου με το όνομα `lab2.tr`³. Επίσης η προσομοίωση δημιουργεί και ένα αρχείο `lab.nam`, το οποίο θα χρησιμοποιηθεί από το γραφικό περιβάλλον `network animator (nam)`. Το γραφικό περιβάλλον εκκινεί αυτόματα μετά το τέλος της προσομοίωσης.

Προσοχή: Πρέπει να σημειωθεί ότι το μέγεθος του αρχείου `lab2.tr` (και γενικά των αρχείων `trace`) μπορεί να είναι μεγάλο. Επομένως θα πρέπει να βεβαιωθείτε ότι υπάρχει διαθέσιμος χώρος στο λογαριασμό σας.

Το επόμενο βήμα είναι να επεξεργαστούμε τα δεδομένα που βρίσκονται στο αρχείο αποτελεσμάτων `lab2.tr` με σκοπό να εξάγουμε συμπεράσματα για τη λειτουργία του δικτύου. Αυτό γίνεται με τη βοήθεια του αρχείου `lab2.sh`. Το αρχείο αυτό διαβάζει τα δεδομένα στο αρχείο `lab2.tr` και υπολογίζει: α) το ποσοστό επιτυχούς παράδοσης πακέτων (*delivery ratio*), β) τη μέση καθυστέρηση στην ουρά αναμονής (*mean queueing delay*), γ) τη μέση καθυστέρηση ενός πακέτου (*mean packet delay*), και δ) το ποσοστό αξιοποίησης (*link utilization*) της ζεύξης μεταξύ των κόμβων {2,3}. Για να χρησιμοποιήσετε το αρχείο `lab2.sh` θα πρέπει πρώτα να το αντιγράψετε στον ίδιο κατάλογο που αποθηκεύσατε και το `lab2.tr` και να εκτελέσετε την εντολή:

```
user@host:...$ sh lab2.sh
```

Με την εκτέλεση της εντολής αυτής τα επιθυμητά αποτελέσματα θα τυπωθούν στην οθόνη.

Σκοπός της άσκησης είναι να μελετηθεί η συμπεριφορά του δικτύου μεταβάλλοντας την τιμή διαφόρων παραμέτρων λειτουργίας όπως π.χ. το μέγεθος της ουράς, ο ρυθμός αποστολής πακέτων, η ταχύτητα μετάδοσης των συνδέσεων, κα. Για να γίνει κατανοητό το πως κάθε μια παράμετρος ξεχωριστά επηρεάζει τη συμπεριφορά του δικτύου η σωστή μεθοδολογία είναι κάθε φορά να μεταβάλλουμε μια μόνο παράμετρο λειτουργίας και να εκτελούμε προσομοιώσεις για διάφορες τιμές της παραμέτρου αυτής. Συνήθως ονομάζουμε τη διαδικασία αυτή *Πείραμα*. Για παράδειγμα, ένα πείραμα στην άσκηση αυτή θα ήταν η εκτέλεση

³Τα αρχεία αποθηκεύονται στο φάκελο μέσα από τον οποίο εκτελέσατε την προσομοίωση, ανεξάρτητα από το αν το εκτελέσιμο του ns βρίσκεται σε άλλο μονοπάτι.

προσομοιώσεων μεταβάλλοντας το μέγεθος της ουράς. Σκοπός ενός πειράματος είναι η καταγραφή διαφόρων μεγεθών της απόδοσης του δικτύου (όπως π.χ. τα μεγέθη που εξάγει το αρχείο `lab2.sh`) για διαφορετικές τιμές μιας παραμέτρου λειτουργίας. Η καταγραφή αυτή γίνεται σε γραφικές παραστάσεις (ή σε πίνακες) και υπάρχει μια γραφική παράσταση (ή πίνακας) για κάθε μέγεθος που καταγράφεται. Στα πλαίσια της άσκησης αυτής θα πραγματοποιήσετε πολλαπλά πειράματα και σε κάθε πείραμα θα καταγράψετε τα μεγέθη που υπολογίζει το αρχείο `lab2.sh`.

4 Παραδείγματα ζητούμενων

Αφού μελετήσετε το αρχείο `lab2.tcl` και εξοικειωθείτε με τη μορφή των δεδομένων του αρχείου `lab2.tr` θα πρέπει να εκτελέσετε μία σειρά από πειράματα και να απαντήσετε σε ορισμένες ερωτήσεις:

- 1ο πείραμα. Μεταβάλετε το μέγεθος της ουράς της ζεύξης {2,3} στις ακόλουθες τιμές: 3, 5, 8, 12 και 15 πακέτα. Καταγράψτε σε έναν πίνακα τις τιμές των μεγεθών που υπολογίζει το αρχείο `lab2.sh`. Σχολιάστε και αιτιολογήστε τη μεταβολή των μεγεθών που καταγράψατε.
- 2ο πείραμα. Διατηρώντας το μέγεθος της ουράς ίσο με 10 πακέτα, μεταβάλετε την παράμετρο `interval` της κίνησης `cbr0` και της κίνησης `cbr1` στις ακόλουθες τιμές: 0.0008, 0.0009, 0.002, 0.005 και 0.01 secs. Καταγράψτε σε έναν πίνακα τις τιμές των μεγεθών που υπολογίζει το αρχείο `lab2.sh`. Σχολιάστε και αιτιολογήστε τη μεταβολή των μεγεθών που καταγράψατε.
- 3ο πείραμα. Διατηρώντας το μέγεθος της ουράς ίσο με 10 πακέτα και την παράμετρο `interval` της κίνησης `cbr0` και της κίνησης `cbr1` ίση με 0.001, μεταβάλετε την παράμετρο `bandwidth` της ζεύξης {2,3} στις ακόλουθες τιμές: 0.5, 0.8, 1, 2 και 2.5 Mbps. Καταγράψτε σε έναν πίνακα τις τιμές των μεγεθών που υπολογίζει το αρχείο `lab2.sh`. Σχολιάστε και αιτιολογήστε τη μεταβολή των μεγεθών που καταγράψατε.
- 1η ερώτηση. Διατηρώντας το μέγεθος της ουράς ίσο με 16 πακέτα, την παράμετρο `interval` της κίνησης `cbr0` και της κίνησης `cbr1` ίση με 0.001 και το `bandwidth` της ζεύξης {2,3} στην τιμή 2.1 Mbps εκτελέστε μία φορά την προσομοίωση. Στη συνέχεια ανοίξτε το αρχείο `lab2.tr` με έναν επεξεργαστή κειμένου και εντοπίστε το πακέτο με `uid` (unique id) = 3900 + ομάδα_εργασίας. Περιγράψτε με σαφήνεια τον κύκλο ζωής του πακέτου, δηλαδή τα χρονικά στάδια της πορείας του (δημιουργία, πηγή, προορισμός, εισαγωγή σε ουρά κλπ).

- 2η ερώτηση. Διατηρώντας το μέγεθος της ουράς ίσο με 21 πακέτα και το bandwidth της ζεύξης {2,3} στην τιμή 1.1 Mbps υπολογίστε την ελάχιστη (κοινή) τιμή της παραμέτρου interval της κίνησης cbr0 και cbr1 ώστε το δίκτυο να λειτουργεί ομαλά στον κόμβο 2.

Α' Παράρτημα: Δομή αρχείου trace

Το αρχείο καταγραφής lab2.tr είναι ένα ascii αρχείο στο οποίο αποθηκεύονται δεδομένα σχετικά με τα πακέτα που διακινούνται στο δίκτυο. Τα δεδομένα αποθηκεύονται σε 12 πεδία, τα οποία χωρίζονται μεταξύ τους με κενό. Τα πιο σημαντικά πεδία που θα μας απασχολήσουν στην παρούσα εργασία είναι:

- Πεδίο1: ο τύπος του γεγονότος. Μπορεί να περιέχει ένα από τα σύμβολα r, +, - και d. Το r σημαίνει ότι το πακέτο παραλήφθηκε στο τέλος μίας ζεύξης, το + σημαίνει ότι το πακέτο εισήλθε σε μία ουρά, το - σημαίνει ότι το πακέτο εξήλθε από μία ουρά και το d σημαίνει ότι το πακέτο απορρίφθηκε.
- Πεδίο2: η ακριβής χρονική στιγμή του παραπάνω γεγονότος.
- Πεδίο3: ο κόμβος εισόδου της ζεύξης (input node) όπου έλαβε χώρα το παραπάνω γεγονός.
- Πεδίο4: ο κόμβος εξόδου της ζεύξης (output node) όπου έλαβε χώρα το παραπάνω γεγονός.
- Πεδίο5: ο τύπος του πακέτου (είναι αντίστοιχο με το όνομα που δηλώθηκε στο αντικείμενο εφαρμογής, π.χ. cbr).
- Πεδίο6: το μέγεθος του πακέτου.
- Πεδίο7: τιμές σημαίας (δεν θα αναλυθούν στην παρούσα άσκηση).
- Πεδίο8: το id ροής (flow id, fid) που αντιστοιχεί στα διαφορετικά χρώματα της προσομοίωσης.
- Πεδίο9: η διεύθυνση του κόμβου αποστολέα του πακέτου (source node) στη μορφή διεύθυνση.θύρα (address.port).
- Πεδίο10: η διεύθυνση του κόμβου προορισμού στη μορφή διεύθυνση.θύρα (address.port).
- Πεδίο11: σειριακός αριθμός πακέτου ανά πρωτόκολλο. Αν και τα πακέτα UDP δεν χρησιμοποιούν τέτοιο σειριακό αριθμό, ο ns το κάνει για λόγους ανάλυσης.
- Πεδίο12: (ένα από τα σημαντικότερα πεδία) ένας μοναδικός σειριακός αριθμός ταυτοποίησης (unique id) που προσδιορίζει ένα πακέτο μέσα στον ns.