

*Διαμόρφωση Κυκλοφορίας Δεδομένων  
με tcng και tcsim*

***“A Hierarchical Token Bucket example”***

Ανέστης Κουτσούδης  
akoutsou@ipet.gr

## Περιεχόμενα

1.	Εισαγωγή.....	3
2.	Η δομή Hierarchical Token Bucket.....	4
3.	Διαμόρφωση εξερχόμενης κίνησης δεδομένων με χρήση tcng και HTB .....	4
3.1	Περιγραφή συμπεριφοράς <i>HTB</i> μέσα από γραφήματα.....	5
3.2	Ανάλυση της χρονικής στιγμής απώλειας του πρώτου πακέτου .....	6
3.3	Επιρροή των παραμέτρων του HTB στους χρόνους αποχώρησης πακέτων από την προσωρινή αποθήκη .....	9
3.4	Δανεισμός διαθέσιμου εύρους από άλλη κατηγορία .....	11
3.5	Ακρίβεια της διαμόρφωσης κυκλοφορίας στα επιθυμητά επίπεδα.....	14
4.	Αναφορές – Βιβλιογραφία.....	17

## Πίνακες

Πίνακας 3.1 - Χαρακτηριστικά κάδων rate και ceil της υποκατηγορίας ftp.....	7
Πίνακας 3.2 - Ανάλυση της κατάστασης της προσωρινής αποθήκης σε κάθε χρονική στιγμή .....	8
Πίνακας 3.3 - Ρυθμίσεις παραμέτρων των κάδων για κάθε σενάριο και στατιστικά κίνησης πακέτων .	9
Πίνακας 3.4 - Χρόνοι αναμονής του κάθε πακέτου στην προσωρινή αποθήκη δεδομένων .....	9
Πίνακας 3.5 - Τιμές παραμέτρων για κάθε σενάριο .....	12
Πίνακας 3.6 - Αποτελέσματα προσομοίωσης με rate/ceil στο 1Kbps.....	14
Πίνακας 3.7 - Αποτελέσματα προσομοίωσης με rate 1Kbps και ceil 5Kbps .....	15

## Γραφήματα

Γράφημα 3.1 - Συνολικός όγκος εισαγόμενων και εξαγόμενων δεδομένων HTB rate 1kbps, ceil 4kbps με sfq .....	5
Γράφημα 3.2 - Ρυθμός εισόδου και εξόδου δεδομένων HTB rate 1kbps, ceil 4kbps με sfq.....	5
Γράφημα 3.3 - Συνολικός όγκος εισαγόμενων και εξαγόμενων δεδομένων .....	6
Γράφημα 3.4 - Ρυθμός εισόδου και εξόδου δεδομένων HTB rate 1kbps, ceil 8kbps με sfq.....	6
Γράφημα 3.5 - Μεταβολή της χρονικής στιγμής αποστολής ενός πακέτου .....	10
Γράφημα 3.6 - Σενάριο 1 <sup>ο</sup> - HTB rate:256 Kbps, ceiling: 256 Kbps .....	12
Γράφημα 3.7 - Σενάριο 2 <sup>ο</sup> - HTB rate:256 Kbps, ceiling: 384 Kbps .....	12
Γράφημα 3.8 - Σενάριο 3 <sup>ο</sup> - HTB rate:256 Kbps, ceiling: 512 Kbps .....	13
Γράφημα 3.9 - Σενάριο 4 <sup>ο</sup> – HTB rate:256 Kbps, ceiling: 600 Kbps .....	13
Γράφημα 3.10 - Ταχύτητα ανταπόκρισης του κάδου rate μιας κατηγορίας.....	15
Γράφημα 3.11 - Ταχύτητα ανταπόκρισης των κάδων rate και ceiling μιας κατηγορίας.....	16

## Πηγαίος Κώδικας

Πηγαίος Κώδικας 3.1 - Ορισμός φίλτρων πακέτων και παραμέτρων της δομής htb .....	7
Πηγαίος Κώδικας 3.2 - Προσομοίωση κίνησης εισαγωγής πακέτων <i>ftp</i> στην κάρτα δικτύου <i>eth0</i> .....	7
Πηγαίος Κώδικας 3.3 - Φιλτράρισμα πακέτων και ορισμός εύρους για κάθε κατηγορία.....	11

## 1. Εισαγωγή

Ο όρος *ποιότητα παροχής υπηρεσιών* (Quality of Service) σε ένα δίκτυο υπολογιστών αναφέρεται συνήθως στην ικανότητα ενός δικτύου να αντεπεξέρχεται σε συγκεκριμένες απαιτήσεις έτσι ώστε να διατηρεί ή να διαμορφώνει την συνολική κίνηση δεδομένων σε κάποια επίπεδα τα οποία ορίζονται παραμετρικά. Έχει ως σκοπό επίσης να δημιουργεί, να συντηρεί και να διαχειρίζεται μηχανισμούς στον κορμό αλλά στην περιφέρεια ενός δικτύου για την εξασφάλιση των εκάστοτε εγγυήσεων απόδοσης του δικτύου. Ως εγγυήσεις λειτουργίας ενός δικτύου στην συγκεκριμένη περίπτωση ορίζονται η ταχύτητα μεταφοράς δεδομένων, ο αριθμός των πιθανών σφαλμάτων αλλά και πλήθος χαρακτηριστικών που δύναται να μετρηθούν και να βελτιωθούν με απώτερο σκοπό την απομάκρυνση της λειτουργίας του δικτύου από μεθοδολογίες – τεχνικές όπως αυτή της *καλύτερης προσπάθειας* (Best effort).

Ο *έλεγχος κυκλοφορίας* (Traffic control) είναι αναπόσπαστο κομμάτι της ποιότητας παροχής υπηρεσιών. Ο έλεγχος διενεργείται με την χρήση ενός συστήματος δρομολόγησης (packet queuing) όλων των εισερχόμενων πακέτων ενός δικτύου ή πιο απλά μίας κάρτας δικτύου. Ο έλεγχος κυκλοφορίας αποτελείται από ένα σύνολο υπολειτουργιών. Αυτές είναι οι εξής:

- i. *Ταξινόμηση* (Classifying) – Μηχανισμός αναγνώρισης προτεραιότητας των πακέτων και τοποθέτησης τους σε αντίστοιχες ροές (flows) ή κατηγορίες (class).
- ii. *Αστυνόμηση* (Policing) – Μηχανισμός περιορισμού του αριθμού των πακέτων ή των δεδομένων σε μία ροή με βάση τους εκάστοτε κανόνες ταξινόμησης.
- iii. *Προγραμματισμός* (Scheduling) – Διαδικασία λήψης αποφάσεων για την κατανομή και ανακατανομή της μετάδοσης των πακέτων.
- iv. *Διαμόρφωση* (Shaping) – Διαδικασία εισαγωγής καθυστέρησης στην μετάδοση των πακέτων για την ισορροπημένη και προβλέψιμη ροή δεδομένων.

Ο συνδυασμός των παραπάνω υπολειτουργιών επιτρέπει την δημιουργία πολύπλοκων μεθόδων διαφύλαξης τους εύρους για συγκεκριμένες εφαρμογές ή υπηρεσίες.

Ο έλεγχος κυκλοφορίας σε ελεύθερα λειτουργικά συστήματα που βασίζονται σε πυρήνα Linux, μπορεί να πραγματοποιηθεί ευκολότερα με την χρήση του εργαλείου *tcng* (Traffic Control Next Generation). Το *tcng* σχεδιάστηκε αρχικά για να παρέχει μια ευανάγνωστη δομή πυγαιού κώδικα (σε σύγκριση με την δομή κώδικα του εργαλείου *tc*) που θα περιγράφει τα διάφορα στοιχεία ελέγχου κυκλοφορίας του πυρήνα Linux. Θα μπορούσε κάποιος να το περιγράψει ως μια προσπάθεια ανασχεδιασμού της αρχιτεκτονικής ελέγχου κυκλοφορίας του *Linux*. Από την άλλη, το *tcng* συνοδεύεται και από ένα σύστημα προσομοίωσης της συμπεριφοράς του ελέγχου κυκλοφορίας στο δίκτυο γνωστό ως *tcsim*. Το *tcsim* προσομοιώνει με μεγάλη λεπτομέρεια τη συμπεριφορά του ελέγχου κυκλοφορίας με αποτέλεσμα να έχει εφαρμογές σε εργασίες δοκιμής, σε πειράματα και σε επικυρώσεις των παραμέτρων ρύθμισης της κυκλοφορίας.

Στα πλαίσια της παρούσας εργασίας πραγματοποιήθηκε η εγκατάσταση του συγκεκριμένου εργαλείου σε ένα υπολογιστικό σύστημα με μία διανομή *Ubuntu 5.10* με έκδοση πυρήνα *Linux 02.6.12-9*. Στα επόμενα κεφάλαια, παρουσιάζονται ο πηγαίος κώδικας που αναπτύχθηκε γύρω από την δομή του *Hierarchical Token Bucket*, τα σενάρια που εκτελέστηκαν στον προσομοιωτή *tcsim* καθώς και ο σχολιασμός των αποτελεσμάτων. Στα πειράματα που ακολουθούν παρουσιάζεται η ιδέα των διαφοροποιημένων υπηρεσιών με την χρήση φίλτρων για την ταξινόμηση των εισερχόμενων πακέτων *ftp* και *http* σε κατηγορίες όπου ελέγχονται από *κάδους κουπονιών* με διαφορετικούς ρυθμούς μετάδοσης.

## 2. Η δομή Hierarchical Token Bucket

Για την επίτευξη της ποιότητας παροχής υπηρεσιών σε ένα δίκτυο υπολογιστών βασικό ρόλο παίζει η διαμόρφωση του κυκλοφοριακού των δεδομένων. Πρόκειται για μια λειτουργία της εκάστοτε συσκευής επόπτευσης ενός δικτύου που στόχο έχει την ρύθμιση και τον συγχρονισμό του ρυθμού αποστολής δεδομένων. Κάτι τέτοιο επιτυγχάνεται αρχικά με την αλλοίωση του κυκλοφοριακού και στην συνέχεια με την σταδιακή διαμόρφωση του σε κάποια επίπεδα [1].

Υπάρχουν διάφορες προσεγγίσεις για την διαμόρφωση του κυκλοφοριακού, στην συγκεκριμένη εργασία γίνεται χρήση του *Ιεραρχικού Κάδου Κουπονιών* (Hierarchical Token bucket ή HTB) που βασίζεται στην τεχνολογία του *Κάδου Κουπονιών* (Token Bucket ή TB). Η διαφορά τους έγκειται στο γεγονός ότι ο μεν πρώτος μπορεί να αποτελείται από πολλαπλούς κάδους με διαφορετικά χαρακτηριστικά. Οι τιμές των παραμέτρων του κάθε κάδου κουπονιών, ορίζουν διαφορετικές κατηγορίες ροής δεδομένων μέσα σε ένα δίκτυο. Η τεχνολογία *TB* σε αντίθεση με αυτή του *Κάδου που Στάζει* (Leaky Bucket ή LB) επιτρέπει κάποια ξεσπάσματα (αύξηση) της ροής δεδομένων, επιβάλλοντας ένα μέγιστο όριο αύξησης της κίνησης. Αυτό επιτυγχάνεται με τον επαναλαμβανόμενο υπολογισμό του μέσου όρου μετάδοσης δεδομένων. Άρα σαν βασική διαφορά ανάμεσα στην τεχνολογία *TB* και *LB* μπορούμε να αναγνωρίσουμε το γεγονός ότι η πρώτη επιτρέπει την κυκλοφορία δεδομένων που εμφανίζουν ξαφνικές αυξήσεις στους ρυθμούς μετάδοσης τους. Ο μέγιστος ρυθμός μετάδοσης σε καταστάσεις ξαφνικής αύξησης ορίζεται από ένα κατώφλι.

Η δομή *Hierarchical Token Bucket* (HTB) αποτελεί μια εξέλιξη της ιδέας του *qdisc*<sup>1</sup> και αποτελεί πλέον αναπόσπαστο τμήμα του μηχανισμού διαμόρφωσης κίνησης δεδομένων του πυρήνα Linux. Αποσκοπεί στη διαμόρφωση της εξερχόμενης κίνησης και επιτρέπει την αποδοτικότερη χρήση ενός δικτύου με την εφαρμογή ενός πολύπλοκου μοντέλου *δανεισμού* (*borrowing*) του διαθέσιμου εύρους. Μια δομή *HTB* αποτελείται από ένα πλήθος κάδων οργανωμένους σε μία δενδροειδής ιεραρχία. Για κάθε μία κλάση (κατηγορία) μέσα στην ιεραρχία υπάρχουν δύο κάδοι κουπονιών. Ο κάθε κάδος αντιπροσωπεύει τις δύο βασικές παραμέτρους που ορίζουν τον ρυθμό μεταφοράς δεδομένων σε μια κατηγορία. Η πρώτη παράμετρος *rate* ορίζει το εγγυημένο εύρος που θα έχει η κατηγορία ενώ η παράμετρος *ceiling* καθορίζει το μέγιστο εύρος που μπορεί να χρησιμοποιήσει μια κατηγορία με την μέθοδο του δανεισμού κάτω από συνθήκες ξαφνικής αύξησης της ροής των εισερχόμενων δεδομένων. Για κάθε κάδο υπάρχουν και άλλες δύο βασικές παράμετροι, η *burst* και *cburst* αντίστοιχα. Οι δύο αυτοί παράμετροι καθορίζουν την χωρητικότητα κουπονιών του εκάστοτε κάδου, ενώ ο ανεφοδιασμός κουπονιών γίνεται με ρυθμούς που ορίζει η παράμετρος *rate*. Όταν δεν υπάρχουν διαθέσιμα κουπόνια μέσα στους κάδους, τότε τα πακέτα τοποθετούνται στις προσωρινές αποθήκες δεδομένων και παραμένουν εκεί μέχρι να εμφανιστούν και πάλι διαθέσιμα κουπόνια. Με τον τρόπο αυτό εμφανίζεται μια αύξηση του χρόνου αναμονής των εισερχόμενων πακέτων στις προσωρινές αποθήκες με αποτέλεσμα να περιορίζεται ο ρυθμός των εξερχόμενων πακέτων στους ρυθμούς που ορίζουν οι παράμετροι *rate* και *ceiling*. Τα κουπόνια στον κάδο *rate* ονομάζονται *tokens* ενώ αυτά του κάδου *ceiling*, *ctokens*. Να σημειωθεί επίσης ότι το κάθε κουπόνι μεταφέρει *1 byte* πληροφορίας αλλά για *100 bytes* δεδομένων αναγκαιούν και 100 κουπόνια. Η επιλογή της κατηγορίας που ανήκει το κάθε εισερχόμενο πακέτο γίνεται με την χρήση φίλτρων. Η προσωρινή αποθήκη δεδομένων της κάθε κατηγορίας μπορεί να ακολουθεί διαφορετικές στρατηγικές δρομολόγησης των πακέτων. Επιπροσθέτως, η παράμετρος *quantum* σε μία *HTB* δομή, περιγράφει τον τρόπο διαχωρισμού του διαθέσιμου εύρους κάτω από συνθήκες δανεισμού. Ενώ τέλος, η παράμετρος *prio* επιτρέπει τον ορισμό προτεραιότητας εξυπηρέτησης για κάθε μία κατηγορία. Η χρήση της προτεραιότητας έχει εφαρμογή σε περιπτώσεις όπου δύο οι περισσότερες κατηγορίες με κοινές παραμέτρους έχουν ανάγκη χρήσης εύρους μεγαλύτερου από το εγγυημένο τους.

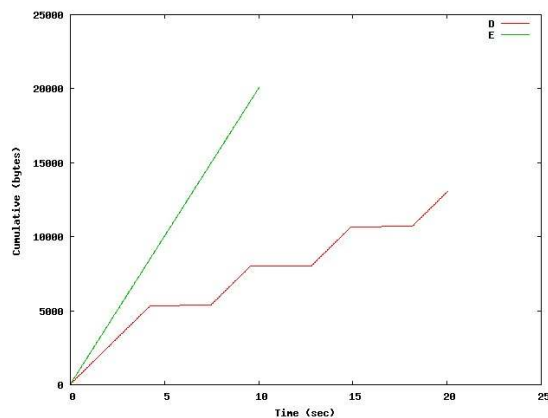
---

<sup>1</sup> *qdisc* είναι η συντομογραφία του Queue Discipline, μιας συγκεκριμένης στρατηγικής με εφαρμογές στον έλεγχο προσωρινών αποθηκών δεδομένων (queues).

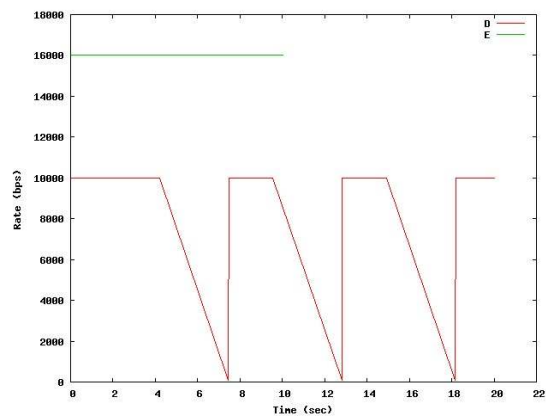
### 3. Διαμόρφωση εξερχόμενης κίνησης δεδομένων με χρήση tcng και HTB

#### 3.1 Περιγραφή συμπεριφοράς HTB μέσα από γραφήματα

Στην παράγραφο αυτή, σχολιάζονται κάποια βασικά χαρακτηριστικά της συμπεριφορά μιας δομής HTB μέσα από γραφήματα προσομοίωσης που παρουσιάζουν τον ρυθμό εισόδου και εξόδου των δεδομένων από μία κάρτα δικτύου. Στο συγκεκριμένο πείραμα, εκτελούνται δύο σενάρια προσομοίωσης κατά τα οποία αυξάνεται η τιμή της παραμέτρου *ceiling* της κατηγορίας πακέτων *ftp* με αποτέλεσμα να αυξάνεται και ο ρυθμός εξόδου των πακέτων από την κάρτα δικτύου. Με τον πηγαίο κώδικα του συγκεκριμένου πειράματος ορίστηκε μια κατηγορία για τα πακέτα *ftp* όπου ο μέγιστος ρυθμός εξόδου (*ceiling*) από τον κάδο την πρώτη φορά αγγίζει τα 4kbps (4000 bits per second στην περίπτωση του *tcsm*) και την δεύτερη τα 8kbps. Κατά την προσομοίωση δημιουργείται για 10 δευτερόλεπτα είσοδος πακέτων *ftp* μεγέθους 40 bytes με ρυθμό 0.02 sec (2000bytes/sec). Επειδή δεν εμφανίζεται κάποια άλλη κίνηση, ο κάδος επιτρέπει τον ρυθμό εξόδου των πακέτων να αγγίζει τα όρια της παραμέτρου *ceiling* (κατάσταση *borrowing*). Σημαντικό είναι να παρατηρήσει κανείς τις παύσεις που εμφανίζονται στα παρακάτω γραφήματα κατά την διαδικασία εξόδου των πακέτων. Με την καθυστέρηση αυτή επιτυγχάνεται ο περιορισμός του μέσου όρου της εξερχόμενης κίνησης στα επίπεδα που έχουν οριστεί για την κατηγορία πακέτων *ftp* (4kbps). Οι παύσεις εμφανίζονται εξαιτίας της έλλειψης διαθέσιμων κουπονιών με αποτέλεσμα τα πακέτα να παραμένουν στους προσωρινούς αποθηκευτικούς χώρους<sup>2</sup> έως ότου δημιουργηθούν νέα κουπόνια. Με αυτό το τρόπο επιτυγχάνεται και η διαμόρφωση της κίνησης που εξάγεται (outbound traffic) από την κάρτα δικτύου. Η μείωση των παύσεων στο δεύτερο σενάριο οφείλεται στο γεγονός της αύξησης των διαθέσιμων κουπονιών άρα και του διαθέσιμου εύρους της κατηγορίας με αποτέλεσμα στην ίδια χρονική περίοδο να εμφανίζεται μια μόνο παύση στην έξοδο των πακέτων (Γράφημα 3.4).



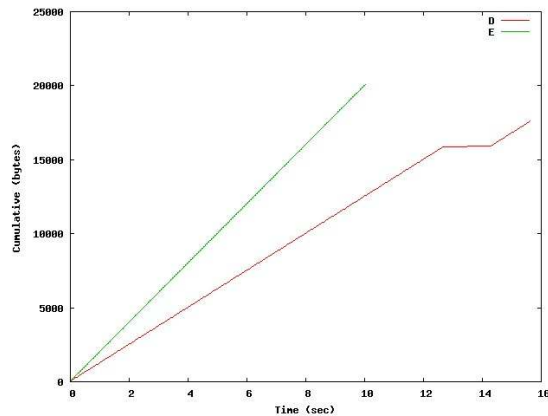
Γράφημα 3.1 - Συνολικός όγκος εισαγόμενων και εξαγόμενων δεδομένων HTB rate 1kbps, ceil 4kbps με sfq



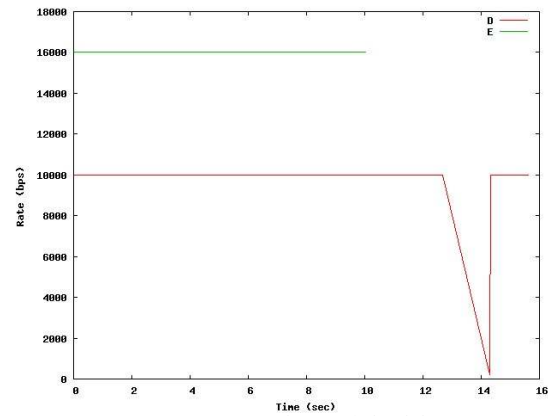
Γράφημα 3.2 - Ρυθμός εισόδου και εξόδου δεδομένων HTB rate 1kbps, ceil 4kbps με sfq

Σενάριο 1ο - Κατηγορία πακέτων *ftp* με *ceiling* στα 4kbps

<sup>2</sup> Ο τρόπος λειτουργίας των προσωρινών αποθηκευτικών χώρων αναλύεται στο επόμενο πείραμα



Γράφημα 3.3 - Συνολικός όγκος εισαγόμενων και εξαγόμενων δεδομένων  
HTB rate 1kbps, ceil 8kbps με sfq



Γράφημα 3.4 - Ρυθμός εισόδου και εξόδου δεδομένων  
HTB rate 1kbps, ceil 8kbps με sfq

### Σενάριο 2ο - Κατηγορία πακέτων *ftp* με *ceiling* στα 8kbps

Στα γραφήματα του συνολικού όγκου εισαγόμενων και εξαγόμενων δεδομένων (cumulative amount of data) παρατηρείται η ομαλή και συνεχής είσοδος των δεδομένων για 10 δευτερόλεπτα ενώ στους ρυθμούς των εξερχόμενων δεδομένων υπάρχουν χρονικές περιόδους (έλλειψη κουπονιών) κατά τις οποίες δεν υπάρχει αύξηση του συνολικού όγκου των εξερχόμενων δεδομένων (φαινόμενο που οφείλεται στο *outbound traffic shaping*). Το ίδιο γίνεται αντιληπτό και στα γραφήματα ρυθμού δεδομένων.

## 3.2 Ανάλυση της χρονικής στιγμής απώλειας του πρώτου πακέτου

Το πείραμα αυτό, αποσκοπεί στην περαιτέρω ανάλυση του τρόπου λειτουργίας του *HTB* και της χρονικής στιγμής κατά την οποία εμφανίζεται η πρώτη απώλεια πακέτου εξ αιτίας της έλλειψης διαθέσιμου προσωρινού χώρου αποθήκευσης των εισερχόμενων πακέτων.

Για την απλότητα του πειράματος ορίστηκε στο κάδο της κατηγορίας πακέτων *ftp*, μία προσωρινή αποθήκη που λειτουργεί με τρόπο *First In First Out* (FIFO). Η δομή *HTB* ορίζει πως σε περιπτώσεις όπου δεν υπάρχουν διαθέσιμα κουπόνια (rate και ceiling), τότε τα εισερχόμενα πακέτα αναμένουν στην εκάστοτε προσωρινή αποθήκη (στην συγκεκριμένη περίπτωση μία *FIFO*). Όταν στην συνέχεια υπάρχουν και πάλι διαθέσιμα κουπόνια και ο προσωρινός χώρος αποθήκευσης δεν έχει γεμίσει, τότε δεν παρατηρείται καμία απώλεια πακέτων ενώ συνεχίζεται και η αποστολή τους ακολουθώντας της πολιτική που ορίζουν οι παράμετροι του *HTB*. Όπως φαίνεται και στον παρακάτω κώδικα (Πηγαίος Κώδικας 2.1), μία συσκευή με ονομασία *eth0* αντιπροσωπεύει την κάρτα δικτύου σε ένα υπολογιστικό σύστημα. Ορίζεται μια αρχική κατηγορία που περιορίζει τον μέγιστο όγκο εξερχόμενων δεδομένων από τον κάδο στο *1Kbps*. Σημειώνεται όπως και στο προηγούμενο πείραμα πως στην περίπτωση του *tcng/tcsim* το *1Kbps* μεταφράζεται σε *1000 bits/sec* και όχι σε *1024 bits/sec* άρα το συνολικό εύρος αντιστοιχεί σε *125 bytes/sec*. Στην αρχική κατηγορία (*root*) ανήκει μια υποκατηγορία που εξυπηρετεί αποκλειστικά πακέτα της υπηρεσίας *ftp*. Η υποκατηγορία αυτή μπορεί να εκμεταλλεύεται συνέχεια το μέγιστο εύρος που φέρει η αρχική κατηγορία εξ αιτίας της κοινής τιμής που φέρουν στην παράμετρο *rate*. Μια τέτοια ρύθμιση είναι σαφές ότι δεν έχει νόημα κάτω από πραγματικές συνθήκες διαμόρφωσης της κίνησης ενός δικτύου. Παρόλα αυτά, έχει εφαρμογή για να προσδιοριστεί εύκολα η χρονική στιγμή κατά την οποία εμφανίζεται η απώλεια (*drop*) του πρώτου πακέτου.

```
#include "fields.tc"
#include "ports.tc"
#define INTERFACE eth0
dev INTERFACE {
    egress {
        class ( <$ftp> ) if tcp_sport == PORT_FTP;
        drop if tcp_sport != PORT_FTP;
        htb () {
            class (rate 1kpbs, ceil 1kpbs) { // Root class
                $ftp = class (rate 1kpbs, ceil 1kpbs, burst 125B, cburst 125B) {fifo limit 125B;};
            }
        }
    }
}
```

### Πηγαίος Κώδικας 3.1 - Ορισμός φίλτρων πακέτων και παραμέτρων της δομής htb

Οι παράμετροι των δύο κάδων της υποκατηγορίας *ftp* είναι :

rate	1kpbs = 125 bytes/sec άρα και 125 tokens/sec
ceiling	1kpbs = 125 bytes/sec άρα και 125 ctokens/sec
burst	125 bytes
cburst	125 bytes

Πίνακας 3.1 - Χαρακτηριστικά κάδων *rate* και *ceil* της υποκατηγορίας *ftp*

Από τις παραπάνω ρυθμίσεις προκύπτει πως σε ένα δευτερόλεπτο δημιουργούνται 125 κουπόνια και περιορίζουμε το συνολικό όγκο αποστολής δεδομένων από την *eth0* σε 125 bytes/sec. Σε συνδυασμό με τις υπόλοιπες παραμέτρους των δύο κάδων (*rate* και *ceil*) της υποκατηγορίας *ftp* και για να μην εμφανιστεί το φαινόμενο της απόρριψης εισερχόμενων πακέτων, θα πρέπει ο ρυθμός των πακέτων που εισάγουμε στην *eth0* να μην ξεπερνά τα 125 bytes/sec. Σε περίπτωση αύξησης του ρυθμού, παρατηρείται η εμφάνιση της πρώτης απώλειας πακέτου. Όταν γεμίσει ο αποθηκευτικός χώρος της αποθήκης *FIFO* και ο ρυθμός που εξέρχονται τα πακέτα (εξ αιτίας της πολιτικής που ορίζει η δομή *HTB*) είναι μικρότερος από τον ρυθμό με τον οποίο καταφτάνουν τότε είναι φυσικό να υπάρχουν απώλειες λόγω του μη-διαθέσιμου προσωρινού χώρου αποθήκευσης. Η παρατήρηση επαληθεύεται με την προσομοίωση της κίνησης δεδομένων σε μία κάρτα δικτύου *eth0* εύρους μόλις 1kpbs και ακολουθώντας δύο σενάρια βασισμένα πάνω στον παρακάτω κώδικα (Πηγαίος Κώδικας 2.3). Η μοναδική παραλλαγή ανάμεσα στα δύο σενάρια εντοπίζεται στον ρυθμό δημιουργίας πακέτων μεγέθους 40 bytes. Στο πρώτο σενάριο ο ρυθμός αυτός είναι 0.32 sec άρα και ο ρυθμός εισόδου δεδομένων είναι 125 bytes/sec ενώ στο δεύτερο 0.2 sec άρα 200 bytes/sec.

```
#include "packet.def"
#include "ports.tc"
#include "meters.tc"
#define INTERFACE eth0
dev eth0 1kpbs {
    #include "htb.tc"
}
    every 0.32s send TCP_PCK($tcp_sport = PORT_FTP); // Traffic equal to 125 bytes/sec
    time 2s
end
```

### Πηγαίος Κώδικας 3.2 - Προσομοίωση κίνησης εισαγωγής πακέτων *ftp* στην κάρτα δικτύου *eth0*

Εκτελώντας τον παραπάνω κώδικα προσομοίωσης για 2 δευτερόλεπτα, τα αποτελέσματα στα οποία καταλήγει η προσομοίωση του *tcsim* είναι αναμενόμενα, εφόσον παρατηρήσει κάποιος τις μεταβολές που δέχεται ο διαθέσιμος αποθηκευτικός χώρος του κάδου σε κάθε χρονική στιγμή. Στον ακόλουθο πίνακα 3.2 εμφανίζεται ο όγκος των δεδομένων που φιλοξενούνται στην αποθήκη σε κάθε χρονική στιγμή της προσομοίωσης. Η έλλειψη του απαραίτητου διαθέσιμου χώρου στην αποθήκη όταν εκτελείται το δεύτερο σενάριο έχει ως αποτέλεσμα την πρώτη απώλεια πακέτου. Κάτι τέτοιο είναι φυσικά αναμενόμενο αφού όπως προαναφέρθηκε ο ρυθμός εξαγωγής των πακέτων που ορίζει ο *HTB* είναι μικρότερος από τον ρυθμό που εισέρχονται. Έτσι, στο χρονικό σημείο της προσομοίωσης των 1.8 secs, το πακέτο απορρίπτεται αφού το μέγεθος του ξεπερνά κατά 35 bytes τον διαθέσιμο χώρο στην αποθήκη *FIFO*.

<i>Αποτελέσματα προσομοίωσης του 1<sup>ου</sup> σεναρίου</i> 0.32 sec για κάθε αποστολή πακέτου των 40 bytes <i>Όγκος δεδομένων που αποστέλλονται: 125 bytes/sec</i>	
Έξοδος προσομοίωσης tcsim	Όγκος δεδομένων στην αποθήκη σε κάθε δεδομένη χρονική στιγμή (bytes)
0.000000 E : 0x80b75a0 40 : eth0: 45000028 00000000 4006 ...	40
0.000000 D : 0x80b75a0 40 : eth0: 45000028 00000000 4006 ...	0
0.320000 E : 0x80b75a0 40 : eth0: 45000028 00000000 4006 ...	40
0.320000 D : 0x80b75a0 40 : eth0: 45000028 00000000 4006 ...	0
0.640000 E : 0x80b75a0 40 : eth0: 45000028 00000000 4006 ...	40
0.640000 D : 0x80b75a0 40 : eth0: 45000028 00000000 4006 ...	0
0.960000 E : 0x80b75a0 40 : eth0: 45000028 00000000 4006 ...	40
0.960000 D : 0x80b75a0 40 : eth0: 45000028 00000000 4006 ...	0
1.280000 E : 0x80b75a0 40 : eth0: 45000028 00000000 4006 ...	40
1.280000 D : 0x80b75a0 40 : eth0: 45000028 00000000 4006 ...	0
1.600000 E : 0x80b75a0 40 : eth0: 45000028 00000000 4006 ...	40
1.600000 D : 0x80b75a0 40 : eth0: 45000028 00000000 4006 ...	0
1.920000 E : 0x80b75a0 40 : eth0: 45000028 00000000 4006 ...	40
1.920000 D : 0x80b75a0 40 : eth0: 45000028 00000000 4006 ...	0
2.240000 E : 0x80b75a0 40 : eth0: 45000028 00000000 4006 ...	40
2.240000 D : 0x80b75a0 40 : eth0: 45000028 00000000 4006 ...	0

<i>Αποτελέσματα προσομοίωσης του 2<sup>ου</sup> σεναρίου</i> 0.2 sec για κάθε αποστολή πακέτου των 40 bytes <i>Όγκος δεδομένων που αποστέλλονται : 200 bytes/sec</i>	
Έξοδος προσομοίωσης tcsim	Όγκος δεδομένων στην αποθήκη σε κάθε δεδομένη χρονική στιγμή (bytes)
0.000000 E : 0x80b75a0 40 : eth0: 45000028 00000000 4006 ...	40
0.000000 D : 0x80b75a0 40 : eth0: 45000028 00000000 4006 ...	0
0.200000 E : 0x80b76a8 40 : eth0: 45000028 00000000 4006 ...	40
0.320000 D : 0x80b76a8 40 : eth0: 45000028 00000000 4006 ...	0
0.400000 E : 0x80b75a0 40 : eth0: 45000028 00000000 4006 ...	40
0.600000 E : 0x80b7750 40 : eth0: 45000028 00000000 4006 ...	80
0.640000 D : 0x80b75a0 40 : eth0: 45000028 00000000 4006 ...	40
0.800000 E : 0x80b76a8 40 : eth0: 45000028 00000000 4006 ...	80
0.960000 D : 0x80b7750 40 : eth0: 45000028 00000000 4006 ...	40
1.000000 E : 0x80b75a0 40 : eth0: 45000028 00000000 4006 ...	80
1.200000 E : 0x80b7828 40 : eth0: 45000028 00000000 4006 ...	120
1.280000 D : 0x80b76a8 40 : eth0: 45000028 00000000 4006 ...	80
1.400000 E : 0x80b7750 40 : eth0: 45000028 00000000 4006 ...	120
1.600000 D : 0x80b75a0 40 : eth0: 45000028 00000000 4006 ...	80
1.600000 E : 0x80b76a8 40 : eth0: 45000028 00000000 4006 ...	120
<b>1.800000 E : 0x80c2258 40 : eth0: 45000028 00000000 4006 ...</b>	<b>160 &gt; 125</b>
<b>1.800000 * : 0x80c2258 40 : eth0: enqueue returns DROP (1)</b>	<b>package loss</b>
1.920000 D : 0x80b7828 40 : eth0: 45000028 00000000 4006 ...	80
2.000000 E : 0x80c2258 40 : eth0: 45000028 00000000 4006 ...	120
<b>2.200000 E : 0x80b75a0 40 : eth0: 45000028 00000000 4006 ...</b>	<b>160 &gt; 125</b>
<b>2.200000 * : 0x80b75a0 40 : eth0: enqueue returns DROP (1)</b>	<b>package loss</b>
2.240000 D : 0x80b7750 40 : eth0: 45000028 00000000 4006 ...	80
2.560000 D : 0x80b76a8 40 : eth0: 45000028 00000000 4006 ...	40
2.880000 D : 0x80c2258 40 : eth0: 45000028 00000000 4006 ...	0

Πίνακας 3.2 - Ανάλυση της κατάστασης της προσωρινής αποθήκης δεδομένων σε κάθε χρονική στιγμή



### 3.3 Επιρροή των παραμέτρων του HTB στους χρόνους αποχώρησης πακέτων από την προσωρινή αποθήκη

Από τον τρόπο λειτουργίας της δομής *HTB*, είναι γνωστό πως ο ρυθμός εισαγωγής των πακέτων δεν επηρεάζεται από την εκάστοτε πολιτική του *HTB*. Από την άλλη πλευρά, όπως φαίνεται και στο παρακάτω πείραμα, επαληθεύεται η επιρροή της πολιτικής του *HTB* κατά την διάρκεια της αποχώρησης των πακέτων από την εκάστοτε προσωρινή αποθήκη. Με την προσομοίωση δύο σεναρίων παρατηρούμε την διαφοροποίηση των χρόνων αποχώρησης των πακέτων (dequeing) καθώς αυξάνουμε το ρυθμό δημιουργίας κουπονιών αλλά και τη χωρητικότητα κουπονιών του κάδου της κάθε υποκατηγορίας. Παρατηρείται η κύρια εφαρμογή του *HTB*, που δεν είναι άλλη από την διαμόρφωση της ροής δεδομένων στην έξοδο της κάρτας δικτύου. Στην συγκεκριμένη περίπτωση γίνεται για άλλη μια φορά χρήση της αποθήκης *FIFO* με μέγεθος μόλις *80 bytes* ικανά για την ταυτόχρονη αποθήκευση 2 εισερχόμενων πακέτων των *40 bytes*. Τα σενάρια που προσομοιώθηκαν είναι τα εξής δύο (Πίνακας 3.3):

	<i>Σενάριο 1ο</i>	<i>Σενάριο 2ο</i>
Ταχύτητα eth0 (Kbps)	1	1
HTB \$ftp class Rate (Bytes per second)	40	80
HTB \$ftp class Ceiling (Bytes per second)	40	80
Πλήθος Εισερχόμενων Πακέτων	7	7
Πλήθος Εξερχόμενων Πακέτων	5	7
Πλήθος Χαμένων Πακέτων	2	0

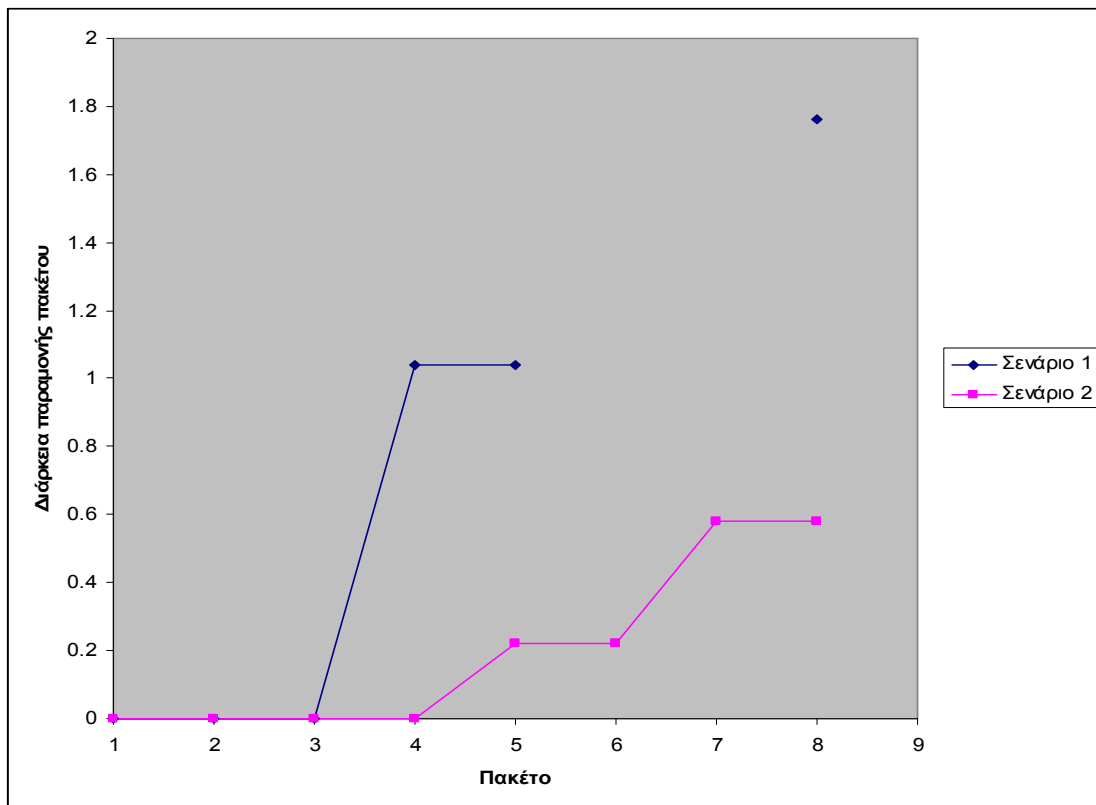
Πίνακας 3.3 - Ρυθμίσεις παραμέτρων των κάδων για κάθε σενάριο και στατιστικά κίνησης πακέτων

Ο ρυθμός που εμφανίζονται τα πακέτα των *40 bytes* και στα δύο σενάρια είναι *0.32 sec*. Ο ρυθμός αυτός δημιουργίας πακέτων, μεταφράζεται σε μία ροή της τάξης των *125 bytes/sec*. Στο πρώτο σενάριο ο κάδος μπορεί να αποστείλει ένα πακέτο ανά δευτερόλεπτο ενώ μπορεί να φιλοξενήσει μόλις 2 στον προσωρινό αποθηκευτικό χώρο. Για τον λόγο αυτό εμφανίζεται και η απόρριψη των δύο από τα οκτώ συνολικά πακέτων που εισάχθηκαν στα 2 δευτερόλεπτα λειτουργίας της προσομοίωσης. Παρόλα αυτά, ενδιαφέρον παρουσιάζει η χρονική διαφορά εξαγωγής των πακέτων από την αποθήκη καθώς αλλάζει η πολιτική του *HTB*. Με τον διπλασιασμό των κουπονιών που παράγονται ανά δευτερόλεπτο (παράμετρος *rate*) βλέπουμε την μείωση αυτής της χρονικής διαφοράς αφού η ύπαρξη κουπονιών στον κάδο επιτρέπει πλέον τον διπλάσιο όγκο εξαγωγής πακέτων σε ένα δευτερόλεπτο. Στον επόμενο πίνακα (Πίνακας 3.4) εμφανίζονται οι χρόνοι κατά τους οποίους εισάχθηκαν και εξάχθηκαν τα πακέτα.

A/A	<i>Σενάριο 1ο</i>			<i>Σενάριο 2ο</i>		
	Χρονική Στιγμή Εισόδου (sec)	Χρονική Στιγμή Εξόδου (sec)	Χρόνος Αναμονής (sec)	Χρονική Στιγμή Εισόδου (sec)	Χρονική Στιγμή Εξόδου (sec)	Χρόνος Αναμονής (sec)
1	0.00	0.00	0.00	0.00	0.00	0.00
2	0.32	0.32	0.00	0.32	0.32	0.00
3	0.64	0.64	0.00	0.64	0.64	0.00
4	0.96	2.00	1.04	0.96	0.96	0.00
5	1.28	2.32	1.04	1.28	1.50	0.22
6	1.60	απόρριψη	-	1.60	1.82	0.22
7	1.92	απόρριψη	-	1.92	2.50	0.58
8	2.24	4.00	1.76	2.24	2.82	0.58

Πίνακας 3.4 - Χρόνοι αναμονής του κάθε πακέτου στην προσωρινή αποθήκη δεδομένων

Στο γράφημα 3.6 γίνεται πιο εμφανής η χρονική διαφορά εξόδου του κάθε πακέτου. Για το 6<sup>ο</sup> και 7<sup>ο</sup> κατά σειρά εισόδου πακέτο δεν υπάρχουν τιμές για το σενάριο ένα, αφού απωλέστηκαν λόγω της έλλειψης προσωρινού χώρου αποθήκευσης κατά την χρονική στιγμή άφιξης τους. Για τα τρία πρώτα πακέτα και στα δύο σενάρια ο χρόνος εξόδου των πακέτων είναι κοινός για το πρώτο δευτερόλεπτο. Καθώς όμως η μορφοποίηση της κίνησης στο πρώτο σενάριο πρέπει να περιοριστεί σε πολύ χαμηλά επίπεδα (40 bytes/sec) παρατηρούμε μια δραματική, για την συγκεκριμένη χρονική κλίμακα, αύξηση του χρόνου αναμονής του τέταρτου πακέτου στην προσωρινή αποθήκη. Η χρονική καθυστέρηση που δημιουργήσε η δομή *HTB*, για το σύνολο των οκτώ πακέτων του δεύτερου σεναρίου, ακολουθώντας της πολιτική που ορίστηκε από τις παραμέτρους *rate* και *ceil* είναι 0.56 sec. Η είσοδος των πακέτων στον κάδο ολοκληρώθηκε μέσα σε 2 δευτερόλεπτα (όπως ορίστηκε στον πηγαίο κώδικα της προσομοίωσης), η έξοδος όμως τους καθυστέρησε για 0.56 δευτερόλεπτα παραπάνω με απώτερο σκοπό την μορφοποίηση της κίνησης των εξερχόμενων δεδομένων σε συγκεκριμένα επίπεδα.



Γράφημα 3.5 - Μεταβολή της χρονικής στιγμής αποστολής ενός πακέτου

### 3.4 Δανεισμός διαθέσιμου εύρους από άλλη κατηγορία

Με το συγκεκριμένο πείραμα παρατηρείται η δυνατότητα μίας κατηγορίας να δανείζεται (borrowing) από το διαθέσιμο εύρος μίας άλλης. Ο δανεισμός δύναται μόνο σε περιπτώσεις όπου η τιμή της παραμέτρου *ceiling* είναι μεγαλύτερη από αυτή της *rate*. Στόχος του δανεισμού είναι η ικανότητα εξυπηρέτησης ροών δεδομένων με ξαφνικές αυξήσεις στον όγκο των δεδομένων τους. Τέτοιου είδους ξαφνικές αυξήσεις του μεταδιδόμενου όγκου εμφανίζονται στην υπηρεσία *http*, καθώς με την αίτηση του χρήστη να δει μια ιστοσελίδα, ξεκινά αμέσως μια ξαφνική ροή δεδομένων (burst), η οποία μετά μειώνεται σημαντικά και τέλος σταματά για όση ώρα ο χρήστης διαβάζει την ιστοσελίδα που κατέβασε. Ο όρος *δανεισμός* όπως εμφανίζεται στην βιβλιογραφία δεν είναι δόκιμος, αφού στην πραγματικότητα το εύρος που χρησιμοποιήθηκε *δεν επιστρέφεται* ποτέ στην υποκατηγορία που ανήκε.

Στο πείραμα ορίζεται το εύρος μίας κάρτας δικτύου *eth0* στα *600 kbps*. Γίνεται ένας ίσος διαχωρισμός του εύρους σε δύο υποκατηγορίες (*http* και *ftp* πακέτων). Κάτι τέτοιο δεν είναι πρακτικό σε πραγματικές καταστάσεις αφού συνήθως δίνεται μεγαλύτερο τμήμα του εύρους στην υποκατηγορία των *http* πακέτων. Κατά την διάρκεια της προσομοίωσης παράγονται μόνο *ftp* πακέτα. Η παράμετρος *ceiling* ορίζεται διαφορετικά σε κάθε σενάριο, έτσι ώστε να επιτρέπει τον σταδιακό δανεισμό εύρους από την υποκατηγορία *http*. Στον κώδικα (Πηγαίος Κώδικας 3.3) που ακολουθεί γίνεται ένα αρχικό φιλτράρισμα των εισερχόμενων πακέτων για την δρομολόγηση τους στον κατάλληλο κάδο.

```
#include "fields.tc"
#include "ports.tc"
#define INTERFACE eth0

dev INTERFACE {
  egress {
    class ( <$ftp> ) if tcp sport == PORT FTP;
    class ( <$http> ) if tcp sport == PORT HTTP;
    drop if tcp sport != PORT FTP || tcp sport != PORT HTTP;
    htb () {
      class (rate 600kbps, ceil 600kbps) {
        $ftp = class (rate 256kbps, ceil 256kbps) {sfq;};
        $http = class (rate 256kbps, ceil 256kbps) {sfq;};
      }
    }
  }
}
```

**Πηγαίος Κώδικας 3.3 - Φιλτράρισμα πακέτων και ορισμός εύρους για κάθε κατηγορία**

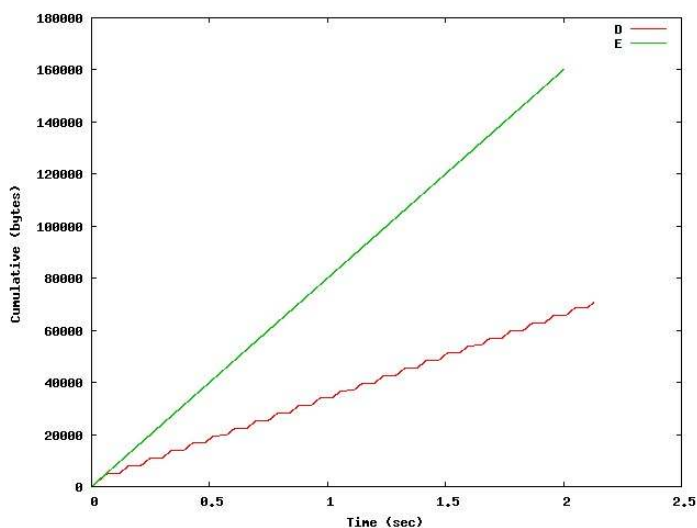
Με το συγκεκριμένο φιλτράρισμα γίνονται αποδεκτά μόνο πακέτα που αφορούν τις δύο υπηρεσίες, ενώ κάποιο πακέτο για οποιαδήποτε άλλη υπηρεσία απορρίπτεται αμέσως. Η χρήση της προσωρινής αποθήκης *Stochastic Fair Queuing (sfq)* αντί μίας απλής *FIFO* βελτιώνει την απόδοση του δικτύου σε πραγματικές συνθήκες αφού ο τρόπος λειτουργίας της, απαγορεύει την δέσμευση όλου του εύρους μίας υποκατηγορίας από έναν και μόνο χρήστη (σύνδεση). Η συγκεκριμένη στρατηγική προτείνεται από πολλούς ως ένας γρήγορος και δίκαιος τρόπος κατανομής του εύρους σε πολλούς χρήστες. Στο συγκεκριμένο πείραμα κάτι τέτοιο δεν είναι εμφανές αφού ο παραπάνω κώδικας προσομοιώνει την δημιουργία πακέτων από μια συγκεκριμένη πηγή.

Οι παραλλαγές στις τιμές των παραμέτρων για κάθε ένα σενάριο είναι οι παρακάτω:

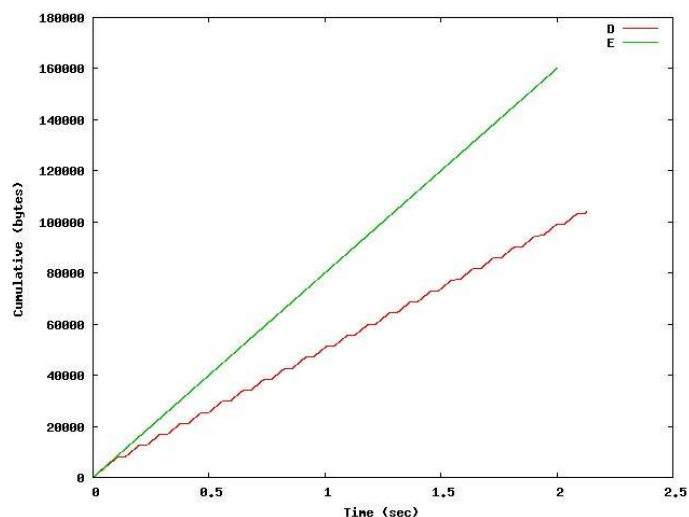
	Σενάριο 1ο	Σενάριο 2ο	Σενάριο 3ο	Σενάριο 4ο
<b>Ταχύτητα eth0 (Kbps)</b>	600	600	600	600
<b>HTB \$ftp class Rate (Kbps)</b>	256	256	256	256
<b>HTB \$ftp class Ceiling (Kbps)</b>	256	384	512	600
<b>Χρόνος απόρριψης του 1ου πακέτου (sec)</b>	0.164000	0.218500	0.409000	1.008500
<b>Πλήθος Εισαγόμενων Πακέτων</b>	4002	4002	4002	4002
<b>Πλήθος Εξαγόμενων Πακέτων</b>	1768	2599	3377	3877
<b>Πλήθος Χαμένων Πακέτων</b>	2234	1403	625	125

**Πίνακας 3.5 - Τιμές παραμέτρων για κάθε σενάριο**

Ο ρυθμός δημιουργίας των πακέτων *ftp* (μεγέθους *40 bytes*) είναι *0.0005 sec* με αποτέλεσμα να παράγεται ένα συνολικό φορτίο της τάξης των *625Kbps*. Κατά την διάρκεια της προσομοίωσης εισάγονται μόνο πακέτα *ftp* με αποτέλεσμα να γίνεται δανεισμός και χρήση του εύρους που ήταν αρχικά καταναμημένο αποκλειστικά για πακέτα υπηρεσίας *http*. Παρόλα αυτά, η έλλειψη ύπαρξης πακέτων *http*, επιτρέπει στον μηχανισμό λειτουργίας του *HTB* να εκμεταλλεύεται καλύτερα το συνολικό διαθέσιμο εύρος του δικτύου με το να επιτρέπει τη χρήση του από άλλες κατηγορίες. Το μέγεθος του εύρους που θα καλύψει ο δανεισμός ορίζεται από την τιμή του ανώτατου ορίου (παράμετρος *ceiling*). Όπως φαίνεται και από τα γραφήματα συνολικού όγκου εισαγόμενων και εξαγόμενων δεδομένων στην επόμενη σελίδα, η σταδιακή αύξηση του ανώτατου ορίου έχει ως αποτέλεσμα την αύξηση του αριθμού των εξερχόμενων πακέτων. Η αύξηση μεγέθους του *ceiling* μειώνει όπως είναι φυσικό και την χρονική περίοδο παραμονής του κάθε πακέτου στην προσωρινή αποθήκη *sfq* της δομής *HTB*. Με την αύξηση του διαθέσιμου εύρους στην υποκατηγορία *ftp* μειώνεται το πλήθος των χαμένων πακέτων αφού αυξάνονται και τα διαθέσιμα κουπόνια σε κάθε δευτερόλεπτο.

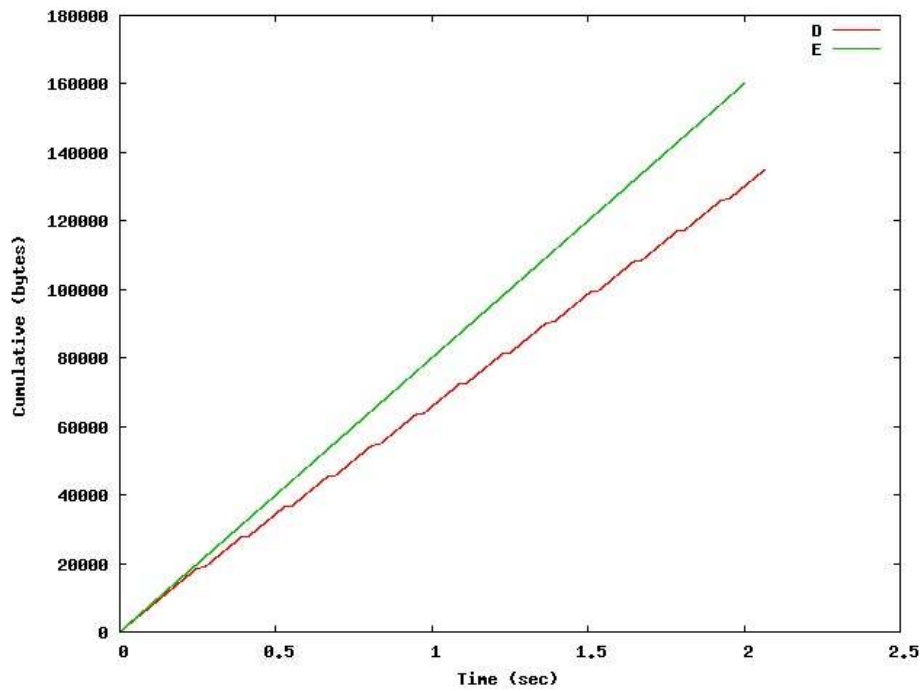


Γράφημα 3.6- Σενάριο 1° - HTB rate:256 Kbps, ceiling: 256 Kbps

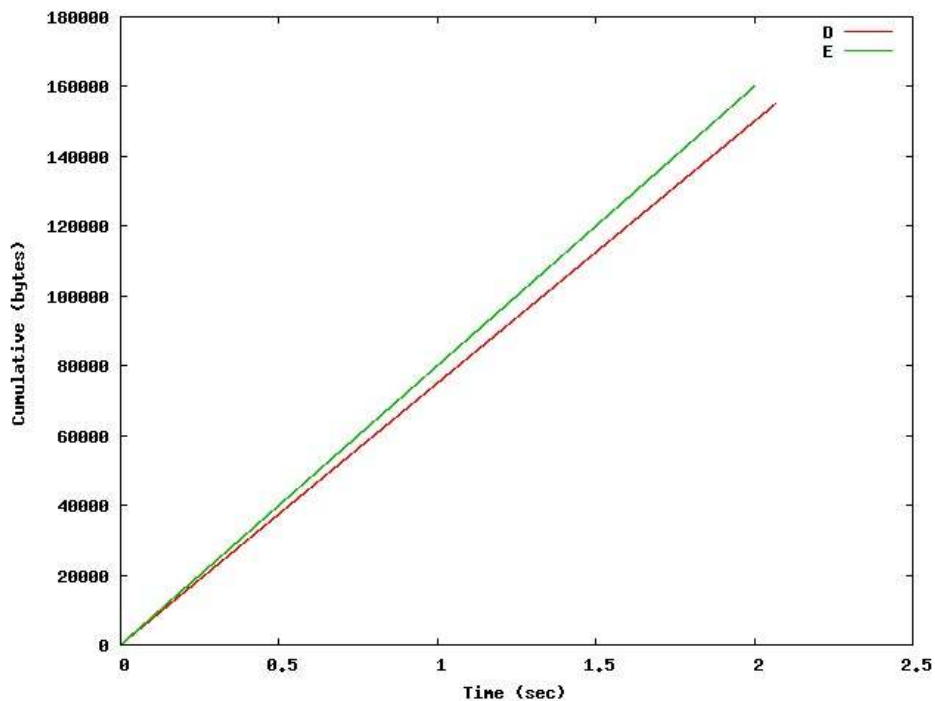


Γράφημα 3.7 – Σενάριο 2° - HTB rate:256 Kbps, ceiling: 384 Kbps

Συγκρίνοντας τα τέσσερα γραφήματα παρατηρείται μια σταδιακή μείωση της χρονικής περιόδου κατά την οποία τα πακέτα αναμένουν στην προσωρινή αποθήκη. Καθώς αυξάνεται το εύρος που χρησιμοποιεί η κατηγορία *ftp*, μειώνεται η χρονική διάρκεια των παύσεων που δημιουργούνται στην έξοδο των πακέτων σε σημείο που δεν είναι εμφανής (π.χ. Γράφημα 3.8 και 3.9).



Γράφημα 3.8 – Σενάριο 3<sup>ο</sup> - HTB rate:256 Kbps, ceiling: 512 Kbps



Γράφημα 3.9 – Σενάριο 4<sup>ο</sup> – HTB rate:256 Kbps, ceiling: 600 Kbps

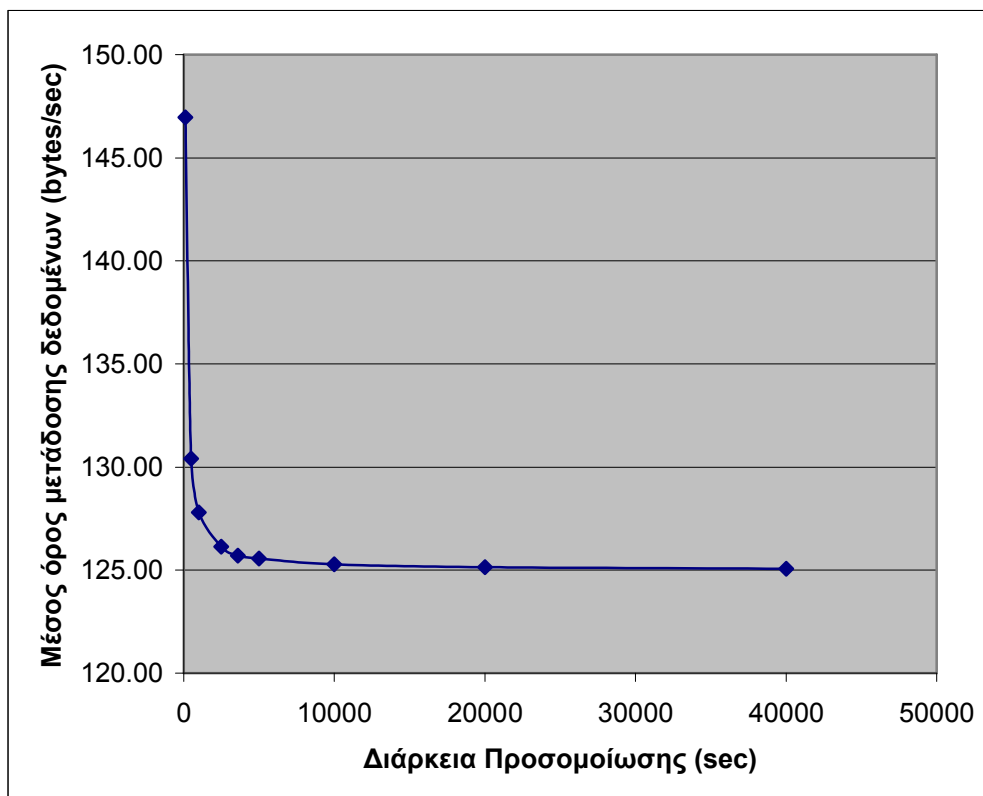
### 3.5 Ακρίβεια της διαμόρφωσης κυκλοφορίας στα επιθυμητά επίπεδα

Με τα δύο σενάρια προσομοίωσης που περιγράφονται στην παρούσα παράγραφο, γίνεται μια προσπάθεια προσδιορισμού της άμεσης διαμόρφωσης της κυκλοφορίας σε συγκεκριμένα επίπεδα από μια δομή *HTB*. Για το συγκεκριμένο πείραμα εκτελέστηκε πολλές φορές ο ίδιος κώδικας προσομοίωσης για διαφορετικής διάρκειας χρονικές περιόδους. Σε κάθε προσομοίωση, μια εικονική πηγή εισήγαγε στην κάρτα δικτύου ένα φορτίο δεδομένων της τάξης των 4.000 bytes/sec. Το φορτίο αυτό αποτελείται και πάλι από πακέτα υπηρεσίας *ftp*. Ο εγγυημένος ρυθμός εξόδου των δεδομένων περιορίστηκε στα 125 bytes/sec χωρίς κανένα δικαίωμα περαιτέρω δανεισμού από άλλη κατηγορία. Αυτό επιτυγχάνεται με τον ορισμό της όμοιας τιμής στους κάδους *rate* και *ceil*.

Στην συνέχεια για το σύνολο των πακέτων που κατάφεραν να εξαχθούν από τη δομή *HTB* υπολογίστηκε ο μέσος όρος ρυθμού αποστολής δεδομένων για διαφορετικές χρονικές στιγμές της προσομοίωσης. Ο υπολογισμός αυτός βασίστηκε στο συνολικό όγκο δεδομένων προς τον χρόνο εξόδου του τελευταίου πακέτου από την κάρτα δικτύου. Η έξοδος (*trace*) του προσομοιωτή *tcsim* παρέιχε όλες τις απαραίτητες πληροφορίες για τους παραπάνω υπολογισμούς. Οι χρόνοι προσομοίωσης ξεκινούν από μερικά δευτερόλεπτα και φτάνουν έως και αρκετές ώρες. Στον πίνακα 3.6 παρουσιάζονται οι μέσοι όροι ρυθμού μετάδοσης δεδομένων μίας κατηγορίας για διαφορετικές διάρκειες προσομοίωσης. Ενώ στο γράφημα 3.10 παρατηρείται ο περιορισμός των εξερχόμενων πακέτων στο επίπεδο των 125 bytes/sec για μια μεγάλη χρονική περίοδο.

Διάρκεια Προσομοίωσης (sec)	Χρονική Στιγμή Εξαγωγής του τελευταίου πακέτου (sec)	Φορτίο Δεδομένων (bytes/sec)	Σύνολο Απεσταλμένων Πακέτων	Σύνολο Απεσταλμένων Δεδομένων (bytes)	Μέσος Όρος ρυθμού μετάδοσης (bytes/sec)
100	132	4.000	485	19.400	146.96
500	535,2	-/-	1.745	69.800	130.41
1.000	1039,2	-/-	3.320	132.800	127.79
2.500	2536,8	-/-	8.000	320.000	126.14
3.600	3631,2	-/-	11.420	456.800	<b>125.70</b>
5.000	5042,4	-/-	15.830	633.200	<b>125.57</b>
10.000	10.039,2	-/-	31.445	1.257.800	<b>125.28</b>
20.000	20.032,8	-/-	62.675	2.507.000	<b>125.14</b>
40.000	40.934,4	-/-	125.180	5.007.200	<b>125.07</b>

Πίνακας 3.6 - Αποτελέσματα προσομοίωσης με rate/ceil στο 1Kbps

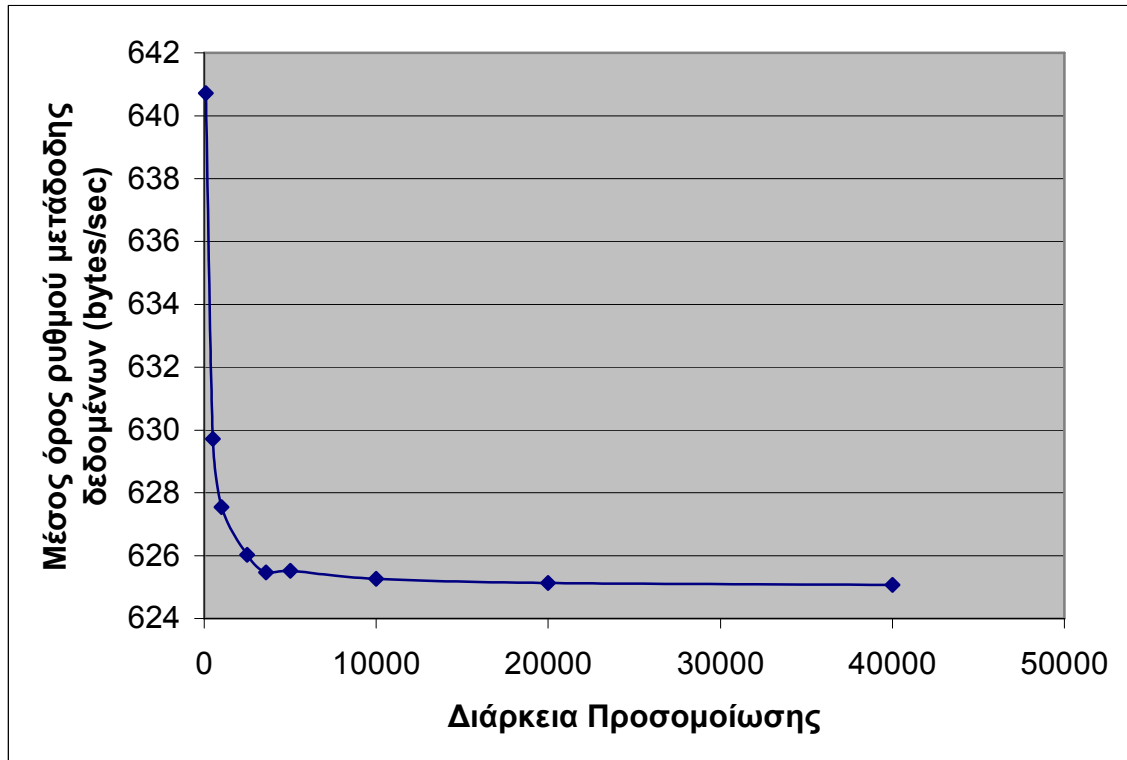


Γράφημα 3.10 – Ταχύτητα ανταπόκρισης του κώδου rate μιας κατηγορίας

Στο δεύτερο σενάριο εκτελέστηκε η ίδια προσομοίωση αλλά αυτή τη φορά επιτρεπόταν και ο δανεισμός εύρους από άλλες κατηγορίες έως και το εύρος των 625bytes/sec.

Διάρκεια Προσομοίωσης (sec)	Χρονική Στιγμή Εξαγωγής του τελευταίου πακέτου (sec)	Φορτίο Δεδομένων (bytes/sec)	Σύνολο Απεσταλμένων Πακέτων	Σύνολο Απεσταλμένων Δεδομένων (bytes)	Μέσος Όρος ρυθμού μετάδοσης (bytes/sec)
100	109,18	4.000	1.749	69.960	640,72
500	509,17	-//-	8.016	320.640	629,72
1.000	1007,33	-//-	15.804	632.160	627,55
2.500	2507,21	-//-	39.240	1.569.600	626,03
3.600	3609,42	-//-	56.440	2.257.600	<b>625,47</b>
5.000	5007,42	-//-	78.307	3.132.280	<b>625,52</b>
10.000	10.006,83	-//-	156.426	6.257.040	<b>625,27</b>
20.000	20.007,29	-//-	312.688	12.507.520	<b>625,14</b>
40.000	40.008,20	-//-	625.209	25.008.360	<b>625,08</b>

Πίνακας 3.7 - Αποτελέσματα προσομοίωσης με rate 1Kbps και ceil 5Kbps



Γράφημα 3.11 -Ταχύτητα ανταπόκρισης των κώδων rate και ceiling μιας κατηγορίας

Παρατηρώντας και τα δύο σενάρια, διακρίνεται η άμεση ανταπόκριση της δομής *HTB* στην διαμόρφωση της ροής των δεδομένων στα επιθυμητά επίπεδα ενώ ταυτόχρονα παρατηρούνται και κάποιες ελάχιστες διορθωτικές μεταβολές στην σταθερότητα του μετά από μία αρκετά μεγάλη χρονική περίοδο προσομοίωσης. Κάτι τέτοιο όμως δεν συμβαίνει σχεδόν ποτέ αφού σε πραγματικές συνθήκες λειτουργίας ενός δικτύου η διάρκεια της σταθερής ροής εισερχόμενων δεδομένων είναι απίθανο να αγγίξει τόσο μεγάλες τιμές (π.χ. 40.000 secs).

#### 4. Συμπεράσματα

Η δομή *HTB* αποτελεί μία εύχρηστη και χαμηλού κόστους μεθοδολογία για την αντιμετώπιση ροών δεδομένων με ξεσπάσματα. Η δημιουργία μιας πολύπλοκης ιεραρχίας σε συνεργασία με ένα σύστημα φίλτρων για την ταξινόμηση των εισερχόμενων πακέτων επιτρέπει τη διαμόρφωση του κυκλοφοριακού σε ένα δίκτυο. Με τα προηγούμενα πειράματα έγινε μια πρώτη προσέγγιση στην ανάλυση λειτουργίας μιας δομής *HTB* αγγίζοντας θέματα όπως η ακρίβεια της δομής και η επεκτασιμότητα της.



## 5. Αναφορές – Βιβλιογραφία

1. Ferguson P., Huston G., *Quality of Service: Delivering QoS on the Internet and in Corporate Networks*, John Wiley & Sons, Inc., 1998. ISBN 0-471-24358-2.
2. *Packet Shaping* [http://gentoo-wiki.com/HOWTO\\_Packet\\_Shaping](http://gentoo-wiki.com/HOWTO_Packet_Shaping)
3. **HTB Linux queuing discipline manual** <http://luxik.cdi.cz/~devik/qos/htb/manual/userg.htm>
4. **Traffic Control Next Generation Reference Manual**, Werner Almesberger, September 28, 2004
5. **HTB performance**,  
[http://home.comcast.net/~msethuraman/htbtest/#1\\_5\\_Mbits\\_sec\\_link\\_bandwidth](http://home.comcast.net/~msethuraman/htbtest/#1_5_Mbits_sec_link_bandwidth)
6. **Classful Queuing Disciplines (Qdiscs)**  
<http://www.tldp.net/HOWTO/Traffic-Control-HOWTO/classful-qdiscs.html#qc-htb>
7. **Information on HTB** <http://www.docum.org/docum.org/docs/htb/>
8. **Linux Advanced Routing & Traffic Control** <http://www.lartc.org/>
9. **Network QoS with Linux** <http://wiki.litux.org/NetworkQos>