



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Τομέας Επικοινωνιών, Ηλεκτρονικής και Συστημάτων
Πληροφορικής

Παροχή Ποιότητας Υπηρεσίας σε IP Δίκτυα με Χρήση
Βελτιστοποιημένων Συστημάτων Κινητών Πρακτόρων

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

του

Στυλιανού Δ. Καλογερόπουλου

Διπλωματούχου Ηλεκτρολόγου Μηχανικού
Πολυτεχνικής Σχολής Πανεπιστημίου Πατρών

Αθήνα, Μάρτιος 2003

ΠΡΟΛΟΓΟΣ

“Μή πάντα ἐρεῦνα, πολλά καί λαθεῖν καλόν”

-Σωκράτης

Η παρούσα διατριβή αποτελεί το αποτέλεσμα μιας ερευνητικής προσπάθειας, διάρκειας τεσσάρων ετών. Η εργασία αυτή δεν θα είχε πραγματοποιηθεί χωρίς την πολύτιμη βοήθεια και συμπαράσταση του επιβλέποντα καθηγητή κ. Βασίλη Λούμου, ο οποίος πέρα από τις επιστημονικές κατευθύνσεις που μου προσέφερε, αντιμετώπισε με το πηγαίο χιούμορ που διαθέτει, όλα τα εμπόδια που συνάντησα. Επίσης θέλω να ευχαριστήσω θερμά τον Καθηγητή ΕΜΠ κ. Ελευθέριο Καγιάφα για την βοήθεια και υποστήριξη που μου προσέφερε όλα αυτά τα χρόνια. Ευχαριστώ επίσης, τον Καθηγητή ΕΜΠ κ. Γ. Στασινόπουλο ο οποίος μου έδωσε την δυνατότητα να εργαστώ σε σημαντικά ερευνητικά προγράμματα. Χωρίς την δυνατότητα αυτή, η υλοποίηση της διατριβής θα ήταν αδύνατη.

Ένα μεγάλο ευχαριστώ σε όλα τα παιδιά του εργαστηρίου πολυμέσων του ΕΜΠ, με τα οποία είχα πάντα μια άψογη συνεργασία. Πιστεύω ότι στο εργαστήριο πολυμέσων πέρασα τα πιο εποικοδομητικά χρόνια της φοιτητικής μου ζωής και γνώρισα ανθρώπους οι οποίοι θα είναι για πάντα φίλοι μου.

Τέλος θέλω να εκφράσω τις ευχαριστίες και την ευγνωμοσύνη μου στην οικογένεια μου η οποία με στήριξε σε όλα τα βήματα της ζωής μου και μου παρείχε τα ηθικά και υλικά εφόδια για να ολοκληρώσω τις σπουδές μου.

Αθηνά Μάρτιος 2003

Καλογερόπουλος Δ. Στυλιανός

Στον Πατέρα μου

ΠΕΡΙΕΧΟΜΕΝΑ

ΚΕΦΑΛΑΙΟ 1

1. ΕΙΣΑΓΩΓΗ.....	1
1.1 Αντικείμενο της διατριβής	1
1.2 Δομή της εργασίας.....	4

ΚΕΦΑΛΑΙΟ 2

2. ΠΟΙΟΤΗΤΑ ΥΠΗΡΕΣΙΑΣ ΣΤΟ ΔΙΑΔΙΚΤΥΟ—ΟΡΙΣΜΟΙ ΚΑΙ ΒΑΣΙΚΕΣ ΕΝΝΟΙΕΣ	9
2.1 Η έννοια της ποιότητας υπηρεσίας στο διαδίκτυο	9
2.2 Τρόπος λειτουργία του διαδικτύου σήμερα και η ανάγκη για παροχή ΠοιΥπ.....	13
2.2.1 Συμβατική ροή της πληροφορίας στο διαδίκτυο	13
2.2.2 Λόγοι για τους οποίους τα δίκτυα με βάση το πρωτόκολλο του διαδικτύου δεν υποστήριζαν έως σήμερα ΠοιΥπ	13
2.2.3 Αναγκαιότητα παροχής ΠοιΥπ από το διαδίκτυο σήμερα	16
2.3 Το γενικό πλαίσιο για την παροχή ΠοιΥπ.....	20
2.3.1 Γενικές αρχές για την παροχή ΠοιΥπ.....	20
2.3.2 Προδιαγραφές για την παροχή ΠοιΥπ.....	20
2.4 Μηχανισμοί παροχής ΠοιΥπ	22
2.4.1 Μηχανισμοί προπαρασκευής ΠοιΥπ	22
2.4.2 Μηχανισμοί έλεγχου ΠοιΥπ	23
2.4.3 Μηχανισμός διαχείρισης ΠοιΥπ	24
2.5 Αρχιτεκτονικές παροχής ΠοιΥπ.....	26
2.5.1 Αρχιτεκτονική Ολοκληρωμένων Υπηρεσιών	27
2.5.1.1 Κλάσεις Υπηρεσιών	28
2.5.1.2 Το μοντέλο Ολοκληρωμένων Υπηρεσιών διαδικτύου.....	29
2.5.1.3 Το πρωτόκολλο RSVP	31
2.5.2 Διαφοροποιημένες Υπηρεσίες	33
2.5.2.1 Στοιχεία της αρχιτεκτονικής διαφοροποιημένων υπηρεσιών	34
2.5.2.2 Διάκριση ανάμεσα στη λειτουργία των δρομολογητών κορμού και άκρων	37
2.5.2.3 Πολλαπλά διαχειριστικά πεδία.....	38
2.5.2.4 Συμβόλαια Υπηρεσιών.....	40
2.5.2.5 Το πεδίο DS και συμβατότητα με το πεδίο TOS	41
2.5.2.6 Ανά κόμβο συμπεριφορές	42
2.5.2.7 Προτεινόμενες υπηρεσίες για προτυποποίηση.....	44
2.5.2.8 Χρονοπρογραμματιστές (schedulers) κατάλληλοι για την παροχή ΔΥ	46

2.5.2.8.1	Ο χρονοπρογραμματιστής CBQ	48
2.5.3	Πολυ-πρωτόκολλο Μεταγωγής Ετικέτας	52
2.6	Η Ποιότητα Υπηρεσίας στα πλαίσια της διατριβής	57
2.7	Σύνοψη κεφαλαίου	58
2.8	Βιβλιογραφικές αναφορές	59

ΚΕΦΑΛΑΙΟ 3

3.	ΤΕΧΝΟΛΟΓΙΑ ΚΙΝΗΤΩΝ ΠΡΑΚΤΩΡΩΝ	63
3.1	Εισαγωγή.....	63
3.2	Μια ιστορική ανάδρομη - Από την μέθοδο ανταλλαγής μηνυμάτων στην τεχνολογία κινητών πρακτόρων	66
3.3	Κινητοί πράκτορες.....	69
3.4	Συστήματα υποστήριξης κινητών πρακτόρων - πλατφόρμες κινητών πρακτόρων	71
3.4.1	Υπηρεσίες ΠΚΠ.....	72
3.4.1.1	Υπηρεσία επικοινωνίας.....	73
3.4.1.2	Υπηρεσία διαχείρισης	74
3.4.1.3	Υπηρεσία καταχώρησης.....	74
3.4.1.4	Υπηρεσία μεταφοράς	75
3.4.1.5	Υπηρεσία επιμονής	75
3.4.1.6	Υπηρεσία καταλόγου	75
3.4.1.7	Υπηρεσία ασφάλειας.....	76
3.4.2	Τεχνολογία κινητών πρακτόρων και η γλώσσα προγραμματισμού Java	78
3.4.2.1	Υποστήριξη της λειτουργίας κινητών πρακτόρων από μηχανισμούς της Java	79
3.5	Πλεονεκτήματα και εφαρμογές της τεχνολογίας κινητών πρακτόρων	80
3.6	Μειονεκτήματα της τεχνολογία κινητών πρακτόρων	83
3.7	Κινητοί πράκτορες στα πλαίσια της διατριβής.....	84
3.8	Σύνοψη κεφαλαίου	86
3.9	Βιβλιογραφικές αναφορές	88

ΚΕΦΑΛΑΙΟ 4

4.	ΕΦΑΡΜΟΓΗ ΤΗΣ ΤΕΧΝΟΛΟΓΙΑΣ ΚΙΝΗΤΩΝ ΠΡΑΚΤΩΡΩΝ ΣΕ ΣΥΣΤΗΜΑΤΑ ΠΑΡΟΧΗΣ ΠΟΙΟΤΗΤΑΣ ΥΠΗΡΕΣΙΑΣ ΣΤΟ ΔΙΑΔΙΚΤΥΟ	91
4.1	Εισαγωγή.....	91
4.2	Τεχνολογία κινητών πρακτόρων και συστήματα διαχείρισης δικτύου.....	93
4.3	Σύστημα διαχείρισης δικτύου διαφοροποιημένων υπηρεσιών	96

4.3.1	Αρχιτεκτονική του Συστήματος.....	96
4.3.2	Ανάλυση της λειτουργία του συστήματος.....	98
4.3.2.1	Υποσύστημα διάρθρωσης.....	99
4.3.2.1.1	Προπαρασκευή του συστήματος πριν την δημιουργία συνδέσεων.....	102
4.3.2.1.2	Δημιουργία σύνδεσης με χρήση του υποσυστήματος διάρθρωσης.....	103
4.3.2.2	Υποσύστημα Παρακολούθησης ΠοιΥπ.....	106
4.3.2.3	Υποσύστημα προσαρμογής.....	108
4.3.2.3.1	Λεπτομέρειες υλοποίησης του υποσυστήματος προσαρμογής.....	110
4.3.2.3.1.1	Διαχείριση δρομολογητών Cisco.....	111
4.3.2.3.1.1.1	Διάρθρωση.....	111
4.3.2.3.1.1.2	Παρακολούθηση.....	113
4.3.2.3.1.2	Διαχείριση δρομολογητών Linux για την παροχή διαφοροποιημένων υπηρεσιών.....	115
4.3.2.3.1.3	Το πρόγραμμα tc.....	118
4.3.3	Υποσύστημα εφαρμογής.....	121
4.3.4	Κοινή διεπαφή - ΠΚΠ (Common MA API).....	128
4.4	Πειραματική αξιολόγηση του συστήματος διαχείρισης.....	131
4.4.1	Κατανομή εύρους ζώνης.....	133
4.4.2	Καθυστερήση μετάδοσης.....	140
4.5	Σύνοψη κεφαλαίου - συμπεράσματα.....	143
4.6	Βιβλιογραφικές αναφορές.....	146

ΚΕΦΑΛΑΙΟ 5

5.	ΣΥΓΚΡΙΣΗ ΤΗΣ ΤΕΧΝΟΛΟΓΙΑ ΚΙΝΗΤΩΝ ΠΡΑΚΤΟΡΩΝ ΜΕ ΤΟ ΜΟΝΤΕΛΟ ΠΕΛΑΤΗ/ΕΞΥΠΗΡΕΤΗΤΗ.....	149
5.1	Εισαγωγή.....	149
5.2	Διάρθρωση δρομολογητών κορμού ενός δικτύου ΔΥ αποτελούμενο από δρομολογητές Linux.....	151
5.3	Τρόπος λειτουργίας των δυο υλοποιήσεων με διαφορετική πολυπλοκότητα του προς εκτέλεση έργου.....	155
5.3.1	Λειτουργία της εφαρμογής με βάση το μοντέλο πελάτη/εξυπηρετητή.....	156
5.3.2	Λειτουργία της εφαρμογής με χρήση της ΤΚΠ.....	161
5.4	Πειραματική σύγκριση.....	164
5.5	Σύνοψη κεφαλαίου – συμπεράσματα.....	169
5.6	Βιβλιογραφικές Αναφορές.....	171

ΚΕΦΑΛΑΙΟ 6

6. ΤΕΧΝΙΚΕΣ ΒΕΛΤΙΩΣΗΣ ΤΗΣ ΑΠΟΔΟΣΗΣ ΚΑΤΑΝΕΜΗΜΕΝΩΝ ΣΥΣΤΗΜΑΤΩΝ ΚΙΝΗΤΩΝ ΠΡΑΚΤΟΡΩΝ	173
6.1 Εισαγωγή.....	173
6.2 Καθορισμός του προβλήματος.....	176
6.3 Βελτίωση της απόδοσης με αναζήτηση και ανάθεση της βέλτιστης διαδρομής... ..	178
6.3.1 Βελτίωση της απόδοσης βάσει του φόρτου εργασίας των μηχανημάτων-στόχων.....	179
6.3.1.1 Πειραματικό σύστημα διαχείρισης δικτύου.....	181
6.3.1.2 Πειραματικά δεδομένα.....	184
6.3.2 Βελτίωση της απόδοσης βάσει της καθυστέρησης μετάδοσης του δικτύου.....	188
6.3.2.1 Βασικά της θεωρίας γραφών με βάρη.....	189
6.3.2.1.1 Ελάχιστο εκτεινόμενο δέντρο.....	190
6.3.2.1.2 Μικρότερο μονοπάτι	191
6.3.2.1.3 Το πρόβλημα περιοδεύοντος πωλητή (Traveling Salesman Problem-TSP)	193
6.3.2.2 Αλγόριθμος εύρεσης της βέλτιστης διαδρομής με βάση την καθυστέρηση μετάδοσης του δικτύου	193
6.3.2.3 Πειραματική εφαρμογή.....	195
6.3.2.4 Πειραματικά Δεδομένα	196
6.4 Βελτίωση της απόδοσης με κατάτμηση μιας εργασίας σε πολλούς πράκτορες. ...	200
6.4.1 Μεθοδολογία Βελτιστοποίησης του αριθμού ΚΠ ενός ΣΠΠ	201
6.4.2 Σχεδίαση και υλοποίηση του πειραματικού ΣΠΠ	203
6.4.3 Πειραματικά αποτελέσματα	206
6.5 Σύνοψη κεφαλαίου – συμπεράσματα	212
6.6 Βιβλιογραφικές Αναφορές	214

ΚΕΦΑΛΑΙΟ 7

7. ΣΥΜΠΕΡΑΣΜΑΤΑ.....	217
7.1 Καινοτομικά στοιχεία της διατριβής.....	217
7.2 Θέματα προς μελλοντική διερεύνηση.....	224

ΚΕΦΑΛΑΙΟ 1

ΕΙΣΑΓΩΓΗ

1.1 Αντικείμενο της διατριβής

Η παρούσα διατριβή πραγματεύεται ερευνητικά θέματα τα οποία έχουν προκύψει από την ραγδαία ανάπτυξη των δικτύων επικοινωνιών και κυρίως από την μεγάλη εξάπλωση του διαδικτύου. Συγκεκριμένα, οι ερευνητικές περιοχές στις οποίες εστιάζει η διατριβή είναι οι παρακάτω:

1. Η ανάπτυξη, υλοποίηση και αξιολόγηση συστημάτων διαχείρισης IP δικτύων, με δυνατότητα παροχής και παρακολούθησης ποιότητας υπηρεσίας (ΠοιΥπ). Τα συστήματα αυτά στηρίζονται σε μηχανισμούς που παρέχουν την δυνατότητα αξιοποίησης των δικτυακών πόρων, ώστε να παρέχεται ΠοιΥπ στους χρηστές, καλύτερη από αυτή της “καλύτερης προσπάθειας” που υποστηρίζεται σήμερα από το διαδίκτυο.
2. Την χρήση και αξιολόγηση της Τεχνολογία Κινητών Πρακτόρων (ΤΚΠ) σε συστήματα παροχής και παρακολούθησης ΠοιΥπ. Η σχετικά νέα αυτή τεχνολογία δημιουργίας κατανεμημένων εφαρμογών χρησιμοποιείται, στα πλαίσια της διατριβής, για την υλοποίηση μιας σειράς συστημάτων παροχής ΠοιΥπ. Οι κινητοί πράκτορες δεν αντιμετωπίζονται μόνο ως εργαλείο υλοποίησης των εφαρμογών αλλά και ως αντικείμενο μελέτης. Η απόδοση τους συγκρίνεται με κλασικό μοντέλο κεντροποιημένης διαχείρισης και προτείνονται τεχνικές (μηχανισμοί) βελτίωσης της.

Με δεδομένη την ευρεία διάδοση του Internet Protocol (IP) και την μεγάλη εγκατεστημένη βάση των IP δικτύων, η ανάγκη για να αποκτήσουν αυτά τα δίκτυα δυνατότητες παροχής ΠοιΥπ είναι επιτακτική. Στην περίπτωση του διαδικτύου, που η ευρεία χρήση του το έχει καθιερώσει σαν ένα μέσο μετάδοσης διαφόρων ειδών δεδομένων, υπάρχει μεγάλη ποικιλία εφαρμογών τα πακέτα των οποίων χρειάζεται να

εξυπηρετηθούν με ειδικό τρόπο. Τέτοιες εφαρμογές είναι η μετάδοση φωνής (Voice over IP-VoIP), κινούμενης εικόνας και εφαρμογές πολυμέσων. Στις περιπτώσεις αυτών των εφαρμογών, τα δεδομένα πρέπει να λαμβάνουν διαφορετική μεταχείριση από το δίκτυο, από την υπόλοιπη κίνηση. Δυστυχώς σήμερα, το διαδίκτυο δεν προσφέρει τη δυνατότητα για διάκριση μεταξύ των πακέτων δεδομένων και συνεπώς τη διαφορετική μεταχείρισή τους.

Για να καλυφθούν οι απαιτήσεις των νέων εφαρμογών, το διαδίκτυο πρέπει να μεταβεί από ένα μοντέλο υπηρεσίας “καλύτερης προσπάθειας”, όπου όλες οι μεταδόσεις θεωρούνται ισάξιες και δεν υπάρχουν εγγυήσεις μεταφοράς, σε ένα μοντέλο ικανό να παρέχει προβλεπόμενα και εγγυημένα επίπεδα υπηρεσίας. Η μετάβαση αυτή οδηγείται τόσο από τις ποιοτικά διαφορετικές απαιτήσεις των νέων εφαρμογών, όσο και από την ώθηση της αγοράς για διαφοροποίηση των υπηρεσιών. Ποικίλοι μηχανισμοί και πρωτόκολλα που έχουν προταθεί για ολοκληρωμένες (Integrated) και διαφοροποιημένες (Differentiated) υπηρεσίες προσπαθούν να παράσχουν αλληλεπιδραστικές και προσαρμόσιμες λύσεις, εφαρμόσιμες σε ένα δίκτυο.

Υπάρχουν πολλοί τρόποι με τους οποίους κάποιος μπορεί να αντιληφθεί την έννοια της ΠοιΥπ στα δίκτυα επικοινωνιών. Στα πλαίσια της διατριβής η ΠοιΥπ αντιμετωπίζεται περισσότερο ως η ικανότητα ενός δικτυακού στοιχείο να εξασφαλίσει ένα συγκεκριμένο επίπεδο εξυπηρέτησης. Η διατριβή εστιάζει στην ΠοιΥπ σε επίπεδο δικτύου. Έτσι λοιπόν, οι παράμετροι ΠοιΥπ είναι η καθυστέρηση μετάδοσης, η απόκλιση καθυστέρησης, η απώλεια πακέτων και το εύρος ζώνης. Σε πρακτικό επίπεδο η παροχή ποιότητας υπηρεσίας επιτυγχάνεται με την κατάλληλη διάρθρωση των δρομολογητών του δικτύου, ώστε να παρέχουν εγγυήσεις για τις τιμές των παραπάνω παραμέτρων.

Οι τρεις σημαντικότερες προτεινόμενες λύσεις στο πρόβλημα παροχής υπηρεσιών, σε IP δίκτυα, είναι οι εξής: η Αρχιτεκτονική Ολοκληρωμένων Υπηρεσιών, η Αρχιτεκτονική Διαφοροποιημένων Υπηρεσιών (Δ.Υ) και το Πολυ-πρωτόκολλο Μεταγωγής Ετικέτας. Η Αρχιτεκτονική ολοκληρωμένων υπηρεσιών εμφάνισε αρκετά μειονεκτήματα με κυριότερο αυτό της μη επεκτασιμότητας σε μεγάλα δίκτυα. Έτσι το μοντέλο αυτό δεν κατέστη κατάλληλο για εφαρμογή στο διαδίκτυο. Η αρχιτεκτονική ΔΥ παρέχει ένα μεγαλύτερο βαθμό ευελιξίας στη διαχείριση ενός δικτύου, με εγγυήσεις στην ΠοιΥπ, σε

σχέση με τις άλλες προτεινόμενες αρχιτεκτονικές. Για τον λόγο αυτό έχει κερδίσει την γενική αποδοχή της κοινότητας του διαδικτύου. Το Πολυ-πρωτόκολλο Μεταγωγής Ετικέτας είναι ένα πρωτόκολλο δρομολόγησης το οποίο μπορεί να χρησιμοποιεί για την παροχή ΠοιΥπ, σε συνδυασμό με τις δυο παραπάνω αρχιτεκτονικές.

Η συνεισφορά της διατριβής στην ανάπτυξη συστημάτων διαχείρισης δικτύου, επικεντρώνεται στην σχεδίαση, υλοποίηση και αξιολόγηση ενός ολοκληρωμένου συστήματος παροχής και παρακολούθησης ΠοιΥπ IP δικτύων. Το σύστημα είναι σχεδιασμένο με βάση την αρχιτεκτονική ΔΥ και υλοποιημένο με χρήση της ΤΚΠ. Βασική επιδίωξη της εργασίας είναι η εφαρμογή των μηχανισμών της αρχιτεκτονικής των ΔΥ και η αξιολόγηση της ΠοιΥπ που παρέχεται από αυτούς. Παρουσιάζονται οι βασικές σχεδιάσθηκες αρχές συστημάτων διαχείρισης με χρήση της ΤΚΠ και προτείνονται λύσεις σε διάφορα προβλήματα υλοποίησής τους.

Τα αποτελέσματα της έρευνας στην περιοχή της ΤΚΠ, έχουν αναδείξει τη χρησιμότητα αυτού του παραδείγματος στην κατασκευή μιας πλειάδας κατανεμημένων εφαρμογών και συστημάτων. Τα τελευταία χρόνια, έχουμε παραστεί μάρτυρες ενός ενθουσιώδους ενδιαφέροντος για τη ΤΚΠ και μια πληθώρα ερευνητικών δραστηριοτήτων που αφορούν αυτή την τεχνολογία έχουν πραγματοποιηθεί. Η ΤΚΠ παρουσιάζει μια σειρά πλεονεκτημάτων έναντι του μοντέλου πελάτη/εξυπηρετητή που χρησιμοποιείται συνήθως στην υλοποίηση κατανεμημένων συστημάτων. Μερικά από αυτά είναι η μείωση της κίνησης στο δίκτυο, η ευκολία επέκτασης των λειτουργιών και η ανεξαρτησία από την διαθεσιμότητα του δικτύου. Παρόλη την δεδομένη ύπαρξη αυτών των πλεονεκτημάτων, ένας αριθμός ζητημάτων που αφορούν κυρίως την απόδοση εφαρμογών βασισμένων στην ΤΚΠ παραμένουν ανοιχτά. Η απόδοση ενός συστήματος διαχείρισης δικτύου, υλοποιημένο με χρήση της ΤΚΠ, συγκρίνεται με αυτή ενός παρόμοιου συστήματος υλοποιημένο με βάση το μοντέλο πελάτη/εξυπηρετητή. Η σύγκριση των δυο προσεγγίσεων βασίζεται στην πολυπλοκότητα του προς εκτέλεση έργου. Τα πειραματικά αποτελέσματα αποδεικνύουν ότι η απόδοση ενός συστήματος υλοποιημένο με χρήση της ΤΚΠ είναι υποδεέστερη από αυτή ενός συστήματος υλοποιημένο με βάση το μοντέλο πελάτη/εξυπηρετητή, για την εκτέλεση απλών διεργασιών

Στα πλαίσια της διατριβής προτείνονται και αξιολογούνται μια σειρά από τεχνικές βελτίωσης της απόδοσης συστημάτων βασισμένων στη ΤΚΠ. Ένας κινητός πράκτορας (ΚΠ) είναι ένα πρόγραμμα το οποίο δεν είναι δεσμευμένο στο σύστημα το οποίο εκτελείται, αλλά έχει την ικανότητα να μετακινεί τον εαυτό του από ένα σύστημα σε ένα άλλο σε ένα δίκτυο υπολογιστών. Κατά την διάρκεια της λειτουργίας του συνήθως επισκέπτεται μια σειρά από μηχανήματα-στόχους, για την εκτέλεση ενός συγκεκριμένου έργου. Η βελτίωση της απόδοσης ενός συστήματος ΚΠ ανάγεται σε ένα πρόβλημα ανεύρεσης της βέλτιστης διαδρομής, που πρέπει να ακολουθήσει ο πράκτορας. Η παρούσα εργασία παρουσιάζει και αξιολογεί δυο διαφορετικούς αλγορίθμους επίλυσης του προβλήματος ανεύρεσης βέλτιστης διαδρομής. Οι αλγόριθμοι υπολογίζουν την βέλτιστη διαδρομή με βάση μια παράμετρο που επηρεάζει την απόδοση ενός ΚΠ (φόρτος εργασίας την κεντρικής μονάδας επεξεργασίας, καθυστέρηση μετάδοσης).

Παρουσιάζεται επίσης, μια μέθοδος βελτίωσης της απόδοσης μιας εφαρμογής ΚΠ με κατάτμηση της εργασίας ενός πράκτορα σε πολλούς, που συνεργάζονται για να φέρουν σε πέρας την ίδια εργασία. Με βάση την θεώρηση ότι η κατάτμηση εργασίας σε περισσότερους του ενός πράκτορες οδηγεί σε βελτίωση της απόδοσης, προκύπτει το εξής ερώτημα: “ποιος είναι ο αριθμός πρακτόρων που βελτιστοποιεί την απόδοση του συστήματος πολλαπλών πρακτόρων;”. Αποτέλεσμα της ερευνάς σε αυτή την κατεύθυνση, είναι μια μεθοδολογία ανεύρεσης του αριθμού ΚΠ που βελτιστοποιεί την απόδοση ενός συστήματος πολλαπλών πρακτόρων. Η συνεισφορά της διατριβής στην περιοχή της ΤΚΠ είναι πολλαπλή.

- Παρουσιάζονται οι βασικές σχεδιάσθηκες αρχές υλοποίησης συστημάτων κατανεμημένων εφαρμογών με χρήση της ΤΚΠ.
- Συγκρίνεται η ΤΚΠ με το μοντέλο πελάτη/εξυπηρετητή και υποδεικνύεται υπό προϋποθέσεις είναι καταλληλότερη η χρήση της.
- Προτείνονται και αξιολογούνται μια σειρά από τεχνικές βελτίωσης της απόδοσης συστημάτων κινητών πρακτόρων.

1.2 Δομή της εργασίας

Η παρούσα εργασία χωρίζεται σε επτά βασικές ενότητες-κεφάλαια. Στις πρώτες δυο ενότητες που ακολουθούν την εισαγωγή, παρουσιάζεται το θεωρητικό υπόβαθρο που

καλύπτει θέματα που σχετίζονται με την ΠοιΥπ στο διαδίκτυο και την ΤΚΠ. Οι επόμενες ενότητες αποτελούν το κυρίως σώμα της εργασίας, όπου παρουσιάζονται και αξιολογούνται το σύστημα παροχής ΠοιΥπ και οι μηχανισμοί βελτίωσης την απόδοσης εφαρμογών ΚΠ. Συγκεκριμένα η δομή της εργασίας είναι η ακόλουθη.

Στο **δεύτερο κεφάλαιο** προσδιορίζεται η έννοια της ΠοιΥπ στο χώρο των δικτύων επικοινωνιών. Παρουσιάζεται η ροή της πληροφορίας στο διαδίκτυο και οι λόγοι για τους οποίους η ΠοιΥπ που παρέχεται είναι χαμηλή. Η ανάγκη για την παροχή ΠοιΥπ από το διαδίκτυο, δημιουργήθηκε από τις ανάγκες των νέων εφαρμογών (VoIP, κινούμενης εικόνας, κ.α). Παρουσιάζοντας τις νέες δικτυακές εφαρμογές αποδεικνύεται ότι το πρόβλημα ΠοιΥπ δεν σχετίζεται μόνο με το εύρος ζώνης, αλλά και τις αυστηρές απαιτήσεις που έχουν οι εφαρμογές αυτές όσον αφορά την καθυστέρηση μετάδοσης. Οι νέες εφαρμογές δεν αύξησαν μόνο την κίνηση δεδομένων στο διαδίκτυο, αλλά άλλαξαν και την φύση της κυκλοφορίας αυτής. Κατόπιν, παρουσιάζονται οι γενικές αρχές, οι προδιαγραφές και οι μηχανισμοί που πρέπει να υποστηρίζονται από ένα δίκτυο επικοινωνιών, για την παροχή ΠοιΥπ. Η παράθεση αυτού του θεωρητικού υπόβαθρου, κρίθηκε απαραίτητη για την ευκολότερη κατανόηση των αρχιτεκτονικών παροχής ΠοιΥπ, των οποίων η παρουσίαση κλείνει αυτήν την ενότητα.

Στο **τρίτο κεφάλαιο** παρουσιάζονται τα βασικά στοιχεία της ΤΚΠ. Αρχικά δίνεται ένας ορισμός της έννοια των λογισμικών πρακτόρων και μια κατηγοριοποίηση τους σε έναν τρισδιάστατο χώρο, στον οποίο η κάθε διάσταση είναι μια από τις ικανότητες τους (νοημοσύνη, τρόπος δράσης, κινητικότητα). Στην συνέχεια παρουσιάζεται η εξέλιξη του τρόπου λειτουργίας των κατανεμημένων συστημάτων και πως η τεχνολογία οδηγήθηκε από την μέθοδο της απλής ανταλλαγής μηνυμάτων στην ΤΚΠ. Διακρίνονται δυο τρόποι με τους οποίους ένας πράκτορας μπορεί να μετακινηθεί σε ένα δίκτυο υπολογιστών, η ισχυρή μετανάστευση και η ασθενής μετανάστευση. Ακολούθως, παρουσιάζεται η αρχιτεκτονική και οι βασικές υπηρεσίες ενός συστήματος υποστήριξης ΚΠ (τα συστήματα αυτά ονομάζονται και Πλατφόρμες Κινητών Πρακτόρων-ΠΚΠ). Οι περισσότερες ΠΚΠ είναι υλοποιημένες στην γλώσσα προγραμματισμού Java. Έτσι, στην συνέχεια παρουσιάζονται τα χαρακτηριστικά και οι μηχανισμοί που παρέχονται από την Java, που την καθιστούν την πλέον κατάλληλη για την ανάπτυξη εφαρμογών ΚΠ. Οι ΚΠ

χρησιμοποιούνται για την δημιουργία κατανεμημένων εφαρμογών, διότι παρουσιάζουν μια σειρά από πλεονεκτήματα σε σχέση με το μοντέλο πελάτη/εξυπηρετητή. Μαζί με τα πλεονεκτήματα παρουσιάζεται και ο τρόπος με τον οποίο επωφελούνται οι κατανεμημένες εφαρμογές από την χρήση των ΚΠ. Υπάρχουν φυσικά και κάποια μειονεκτήματα τα οποία παρουσιάζονται επίσης. Τέλος, παρουσιάζεται ο τρόπος με τον οποίο αντιμετωπίζονται οι ΚΠ στα πλαίσια της διατριβής και γίνεται μια αναφορά στις πλατφόρμες οι οποίες θα χρησιμοποιηθούν για την υλοποίησή τους.

Στο **τέταρτο κεφάλαιο** παρουσιάζεται ένα σύστημα παροχής ΠοιΥπ σε IP δίκτυα. Το σύστημα είναι ικανό να προσφέρει υπηρεσίες διάρθρωσης και παρακολούθησης ενός IP δικτύου, με βάση με την αρχιτεκτονική των ΔΥ. Το σύστημα είναι υλοποιημένο με χρήση της ΤΚΠ. Όλες οι λειτουργίες διάρθρωσης και παρακολούθησης πραγματοποιούνται με χρήση κινητών και στατικών πρακτόρων (οι στατικοί πράκτορες δεν διαθέτουν την ικανότητα μετανάστευσης). Οι ΚΠ μεταφέρουν τις εντολές διάρθρωσης και παρακολούθησης στους κόμβους του δικτύου. Επίσης, επιστρέφουν τα αποτελέσματα της εκτέλεσης των εντολών στην κεντρική μονάδα του συστήματος, μέσω της οποίας παρουσιάζονται στον χρήστη. Οι στατικοί πράκτορες που βρίσκονται στους κόμβους του δικτύου (δρομολογητές) δρουν ως διεπαφές μεταξύ των ΚΠ και του λογισμικού του κόμβου. Δηλαδή όλες οι αλληλεπιδράσεις μεταξύ των ΚΠ και των κόμβων γίνονται διάμεσου των στατικών πρακτόρων. Το σύστημα είναι σε θέση να διαχειριστεί δρομολογητές τύπου Linux και τύπου Cisco. Δυστυχώς, δεν υπάρχει μια κοινή βάση διαχείρισης της πληροφορίας (Management Information Base-MIB) ΔΥ που να υποστηρίζεται και από του δυο τύπους. Η ύπαρξη μια τέτοιας MIB θα επέτρεπε την διαχείριση των δυο τύπων δρομολογητών, με ένα ενιαίο τρόπο. Το πρόβλημα αυτό αντιμετωπίστηκε με την δημιουργία ενός υποσυστήματος προσαρμογής το οποίο αποκρύπτει τις λεπτομέρειες διαχείρισης των δρομολογητών από τους ΚΠ. Η λειτουργία του υποσυστήματος αυτού στηρίζεται στην ύπαρξη μιας σειράς στατικών πρακτόρων που διαθέτουν όλη την απαραίτητη πληροφορία για την παρακολούθηση και διάρθρωση των δρομολογητών. Ένα πρόβλημα που αντιμετωπίζουν όλες οι εφαρμογές ΚΠ είναι η εξάρτηση της λειτουργίας τους από μια και μόνο ΠΚΠ. Το πρόβλημα αυτό ξεπεράστηκε με την δημιουργία μια κοινής Διεπαφής Προγραμματισμού Εφαρμογής (ΔΠΕ) για τις δυο ΠΚΠ που συνήθως χρησιμοποιούνται (Grasshopper και Voyager). Η χρήση της κοινής

ΔΠΕ επιτρέπει την εκτέλεση των πρακτόρων του συστήματος και στις δυο πλατφόρμες με ελάχιστη μεταβολή στον κώδικα τους. Για την διάθρωση των δρομολογητών υλοποιήθηκε η ΔΠΕ Jain/Parlay. Σύμφωνα με την λογική της ΔΠΕ Jain/Parlay ένα χρήστης είναι σε θέση να δημιουργήσει εικονικούς ιδιωτικούς δίαυλους (Virtual Private Pipes, VPrP) με άλλους χρηστές του συστήματος. Οι δίαυλοι αυτοί είναι συνδέσεις που παρέχουν μια συγκεκριμένη ΠοιΥπ στην επικοινωνία των χρηστών. Επίσης ο χρήστης είναι σε θέση να παρακολουθήσει την ΠοιΥπ που του προσφέρει το σύστημα, ώστε να μπορεί να την συγκρίνει με αυτήν που απαιτήσε. Η παρακολούθηση της παρεχόμενης ΠοιΥπ έχει ιδιαίτερη σημασία στα συστήματα παροχής ΠοιΥπ, διότι ο χρήστης πρέπει να είναι σε θέση να γνωρίζει αν η ΠοιΥπ που του παρέχεται είναι σύμφωνη με το συμβόλαιο υπηρεσιών (SLA) για το οποίο έχει πληρώσει. Τέλος Η εφαρμογή αξιολογείται βάσει πειραματικών μετρήσεων που διεξάγονται σε ένα IP δίκτυο ΔΥ, εγκατεστημένο ακριβώς για αυτό τον σκοπό.

Στο **πέμπτο κεφάλαιο** παρουσιάζεται μια σύγκριση της ΤΚΠ με το μοντέλο πελάτη/εξυπηρετητή. Συγκρίνεται η κεντροποιημένη διαχείριση που εκφράζεται από το μοντέλο πελάτη/εξυπηρετητή με την αποκεντροποιημένη διαχείριση που εκφράζεται από τους ΚΠ. Για τον λόγο αυτό δημιουργήθηκαν δυο εφαρμογές, μια για κάθε προσέγγιση, που εκτελούν την αρχική διάρθρωση ενός IP δικτύου ΔΥ, αποτελούμενο από δρομολογητές Linux. Η σύγκριση είναι βασισμένη στο χρόνο εκτέλεσης που απαιτεί κάθε μια από τις δυο προσεγγίσεις, για την εκτέλεση μιας σειράς διεργασιών διαφορετικής πολυπλοκότητας. Η διαφορετική πολυπλοκότητα του προς εκτέλεση έργου, επέτρεψε την διερεύνηση της επίδρασης που έχει στην απόδοση των εφαρμογών η λογική που εμπεριέχεται στο έργο που καλούνται να εκτελέσουν. Ένας άλλος παράγοντας ο οποίος διερευνήθηκε, ήταν ο τρόπος με το οποίο χρησιμοποιούν το εύρος ζώνης του δικτύου οι δυο εφαρμογές. Ένα πειραματικό IP δίκτυο αποτελούμενο από δρομολογητές Linux, χρησιμοποιήθηκε για την διεξαγωγή των πειραμάτων.

Στο **έκτο κεφάλαιο** παρουσιάζονται τρεις διαφορετικοί μέθοδοι (μηχανισμοί) βελτίωσης της απόδοσης εφαρμογών διαχείρισης δικτύου, υλοποιημένων με χρήση της ΤΚΠ. Ακολουθήθηκαν δυο διαφορετικές προσεγγίσεις για την επίλυση του προβλήματος. Στη πρώτη προσέγγιση η βελτίωση επιτυγχάνεται με ανεύρεση και ανάθεση στον πράκτορα

της βέλτιστης διαδρομής, με βάση μια παράμετρο που επηρεάζει την απόδοση του. Στην δεύτερη προσέγγιση η βελτίωση είναι αποτέλεσμα της κατάτμησης μιας εργασίας σε πολλούς πράκτορες, οι οποίοι δρουν ταυτόχρονα για την επίτευξη της. Οι παράμετροι οι οποίοι επηρεάζουν την απόδοση ενός ΚΠ, είναι: η κατάσταση του υποκείμενου δικτύου και η κατάσταση των μηχανημάτων-στόχων. Υλοποιήθηκαν δυο αλγόριθμοι ανεύρεσης βέλτιστης διαδρομής. Κάθε αλγόριθμος υπολογίζει την βέλτιστη διαδρομή με βάση μια από τις δυο παραπάνω παραμέτρους. Με αυτόν τον τρόπο, είναι δυνατή η διερεύνηση της επίδρασης στην απόδοση κάθε παραμέτρου ξεχωριστά. Στην δεύτερη προσέγγιση του προβλήματος, η βελτίωση επιτυγχάνεται με κατάτμηση μιας εργασίας σε πολλούς πράκτορες. Προτείνεται λοιπόν μια μεθοδολογία ανεύρεσης του αριθμού πρακτόρων, που βελτιστοποιούν την απόδοση ενός συστήματος πολλαπλών πρακτόρων. Όλοι οι μηχανισμοί βελτίωσης αξιολογούνται και συγκρίνονται βάσει πειραματικών μετρήσεων, που λαμβάνονται με την χρήση πρωτότυπων υλοποιήσεων συστημάτων διαχείρισης δικτύου.

Το **έβδομο** και τελευταίο **κεφάλαιο** της διατριβής συνοψίζει τα σημαντικότερα σημεία συνεισφοράς της εργασίας και περιγράφει τα κυριότερα θέματα για μελλοντική διερεύνηση.

ΚΕΦΑΛΑΙΟ 2

ΠΟΙΟΤΗΤΑ ΥΠΗΡΕΣΙΑΣ ΣΤΟ ΔΙΑΔΙΚΤΥΟ – ΟΡΙΣΜΟΙ ΚΑΙ ΒΑΣΙΚΕΣ ΕΝΝΟΙΕΣ

2.1 Η έννοια της ποιότητας υπηρεσίας στο διαδίκτυο

Η ποιότητα υπηρεσίας είναι ένα από τα πιο λανθάνοντα, συγκεχυμένα και απροσδιόριστα θέματα στον κόσμο των δικτύων σήμερα. Γιατί όμως μια τέτοια εκ πρώτης όψεως απλή έννοια έχει φθάσει σε τόσο υψηλά επίπεδα σύγχυσης; Ολόκληρη η βιομηχανία των τηλεπικοινωνιών φαίνεται να χρησιμοποιεί τον όρο με προφανή ευκολία και η χρήση του όρου είναι πολύ κοινή. Επομένως, θα ήταν λογικό να περιμένουμε να υπάρχει ένα κοινό επίπεδο κατανόησης του όρου. Για να φθάσουμε σε ένα αποδεκτό ορισμό της ποιότητας υπηρεσίας, θα ξεκινήσουμε από τις έννοιες από τις οποίες συνίσταται: την **ποιότητα** και τις **υπηρεσίες**.

Ο όρος **ποιότητα** μπορεί να συμπεριλάβει πολλές ιδιότητες των δικτύων, αλλά χρησιμοποιείται συνήθως για να περιγραφεί η διαδικασία της μετάδοσης των δεδομένων με ένα αξιόπιστο τρόπο ή έστω με ένα τρόπο καλύτερο από τον κανονικό. Αυτή η μέθοδος περιλαμβάνει τον παράγοντα της απώλειας δεδομένων, την ελάχιστη (ή όχι) προκαλούμενη καθυστέρηση και τη δυνατότητα να καθορίζουμε την αποδοτικότερη χρήση των πόρων του δικτύου (όπως την ελάχιστη απόσταση ανάμεσα σε δύο άκρα ή τη μέγιστη αποδοτικότητα της χωρητικότητας του κυκλώματος). Ποιότητα μπορεί επίσης να σημαίνει ένα χαρακτηριστικό γνώρισμα. Έτσι, ο κόσμος χρησιμοποιεί τον όρο ποιότητα για να καθορίσει συγκεκριμένα χαρακτηριστικά εφαρμογών δικτύων ή πρωτοκόλλων.

Ο όρος **υπηρεσίες** επίσης μπορεί να έχει πολλά νοήματα και εξαρτάται από το πώς ένας οργανισμός ή μια επιχείρηση είναι δομημένοι. Συνήθως χρησιμοποιείται για να περιγράψει κάτι που προσφέρεται στους τελικούς χρήστες ενός δικτύου, όπως επικοινωνία από άκρο-σε-άκρο (end-to-end) ή εφαρμογές Πελάτη/Εξυπηρετητή (Client/Server). Οι υπηρεσίες μπορεί να καλύπτουν ένα ευρύ φάσμα παροχών, από

ηλεκτρονικό ταχυδρομείο μέχρι video και από πλοήγηση στο WWW μέχρι χώρους για συνομιλίες.

Παραδοσιακά οι παροχείς υπηρεσιών δικτύου χρησιμοποιούν μια ποικιλία από μεθόδους για να παρέχουν εγγυήσεις υπηρεσιών στους συνδρομητές τους, οι περισσότερες από τις οποίες είναι συμβατικές. Η διαθεσιμότητα του δικτύου παραδείγματος χάριν, είναι μια από τις πιο διαδεδομένες μετρήσεις σε μια συμφωνία επιπέδου υπηρεσιών ανάμεσα στον παροχέα και στο συνδρομητή. Εδώ η πρόσβαση στο δίκτυο είναι η βασική υπηρεσία και η αδυναμία να παρέχεις αυτή την υπηρεσία είναι αδυναμία να ανταποκριθείς στις συμβατικές σου υποχρεώσεις. Αν το δίκτυο δεν είναι προσπελάσιμο, η ποιότητα της υπηρεσίας τίθεται καθαρά υπό αμφιβολία. Περιστασιακά οι παροχείς υπηρεσιών έχουν εισάγει επιπρόσθετα κριτήρια για εγγυημένη μετάδοση, όπως η ποσότητα της κυκλοφορίας που μεταδόθηκε. Αν ένας παροχέας μεταδίδει στον τελικό προορισμό μόνο το 98% της κυκλοφορίας, τότε μπορεί να θεωρηθεί ότι η υπηρεσία υστερεί σε ποιότητα. Όπως γίνεται αντιληπτό, ένα κομμάτι της κοινότητας των δικτύων αποστρέφεται τη χρήση του όρου *εγγύηση* επειδή μπορεί να είναι μια διφορούμενη και μη ευδιάκριτη έννοια και να οδηγήσει σε αντιφάσεις. Το να παρέχεις μια εγγυημένη υπηρεσία κάποιου είδους υπονοεί ότι όχι μόνο δε θα υπάρχουν απώλειες πακέτων, αλλά και ότι η απόδοση του δικτύου θα είναι σταθερή και προβλέψιμη. Στον κόσμο των δικτύων που βασίζονται σε πακέτα, αυτή είναι μια πολύ μεγάλη πρόκληση.

Από την ανάλυση των όρων ποιότητα και υπηρεσίες παίρνουμε έναν γενικό ορισμό της Ποιότητας Υπηρεσίας: *Η ποιότητα υπηρεσίας ένα μέτρο του πόσο καλά συμπεριφέρεται το δίκτυο και μια προσπάθεια να καθοριστούν τα χαρακτηριστικά και οι ιδιότητες συγκεκριμένων υπηρεσιών* [Ferguson98].

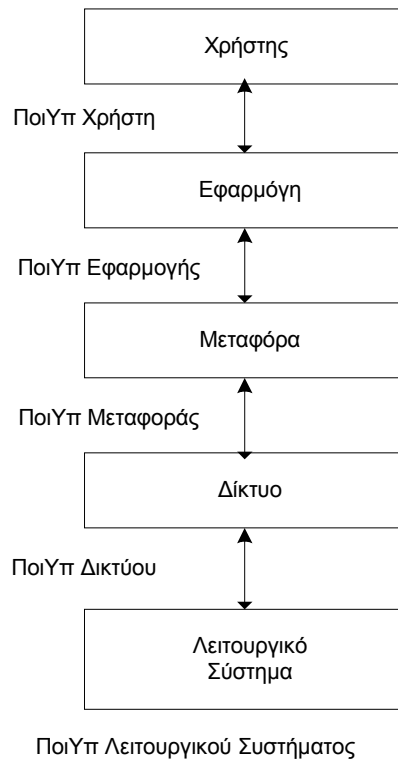
Το γεγονός ότι δεν υπάρχει ένας καθολικά αποδεκτός ορισμός για την ποιότητα υπηρεσίας στον χώρο των δικτύων, οδήγησε διάφορους οργανισμούς να δώσουν τον δικό τους ορισμό. Ενδεικτικά αναφέρονται οι παρακάτω:

- Ο διεθνής οργανισμός πρότυπων ISO (International standard Organization) ορίζει την ποιότητα υπηρεσίας σε σχέση με το πόσο καλή είναι μια δικτυακή υπηρεσία [Zheng99].

- Ο οργανισμός ποιότητας υπηρεσίας (QOSFORUM) ορίζει την ποιότητα υπηρεσίας ως την ικανότητα ενός δικτυακού στοιχείου (π.χ. ενός δρομολογητή, ενός υπολογιστή, ακόμη και μιας εφαρμογής) να εξασφαλίσει ένα συγκεκριμένο επίπεδο εξυπηρέτησης, όσον αφορά την μετάδοση των δεδομένων [QoS99].

Η έννοια την ποιότητας υπηρεσίας καθώς και οι παράμετροι και τα μεγέθη με τα οποία σχετίζεται είναι διαφορετική ακόμη και στα διαφορετικά επίπεδα μιας δικτυακής εφαρμογής (σχήμα 2.1). Έτσι λοιπόν, σύμφωνα με το επίπεδο της εφαρμογής, η ποιότητα υπηρεσίας χαρακτηρίζεται ως:

- Ποιότητα υπηρεσίας χρήστη (User QoS): Ο χρήστης που βρίσκεται στο υψηλότερο επίπεδο μιας εφαρμογής αντιλαμβάνεται την ποιότητα υπηρεσίας με απλούς όρους. Έτσι, για έναν χρήστη η ποιότητα υπηρεσίας που του παρέχεται μπορεί να είναι χαμηλή, μεσαία και υψηλή.



Σχήμα 2.1: Η ποιότητα Υπηρεσίας (ΠοιΥπ) στα διάφορα επίπεδα μιας δικτυακής εφαρμογής

- Ποιότητα υπηρεσίας εφαρμογής (Application QoS): Σε αυτό το επίπεδο η ποιότητα υπηρεσίας σχετίζεται με παραμέτρους που αφορούν την εφαρμογή που χρησιμοποιείται.

- Ποιότητα υπηρεσίας στο επίπεδο μεταφοράς (Transport QoS) και επίπεδο δικτύου (Network QoS): Σε αυτά τα επίπεδα η ποιότητα υπηρεσίας παίρνει πιο συγκεκριμένη μορφή και αναφέρεται σε παραμέτρους του δικτύου όπως το εύρος ζώνης (bandwidth), την καθυστέρηση μετάδοσης (delay), την απόκλιση καθυστέρησης (jitter) και την απώλεια πακέτων (packet loss).
- Ποιότητα υπηρεσίας σε επίπεδο λειτουργικού συστήματος (System QoS): Σε αυτό το επίπεδο η ποιότητα υπηρεσίας σχετίζεται με παραμέτρους, όπως η προτεραιότητα και η διάρκεια των εκτελούμενων εργασιών.

Για λόγους συντομίας ο όρος Ποιότητα Υπηρεσίας θα αναφέρεται ως ΠοιΥπ, από εδώ και στο εξής.

2.2 Τρόπος λειτουργία του διαδικτύου σήμερα και η ανάγκη για παροχή ΠοιΥπ

2.2.1 Συμβατική ροή της πληροφορίας στο διαδίκτυο

Σε αυτό το σημείο είναι σκόπιμο να περιγράψει ο συμβατικός τρόπος λειτουργίας του διαδικτύου και οι λόγοι για τους οποίους η ΠοιΥπ που παρέχεται είναι αρκετές φορές χαμηλή.

Το διαδίκτυο αποτελείται από μια συλλογή δρομολογητών (routers) και διασυνδέσεις μετάδοσης (links). Οι δρομολογητές λαμβάνουν ένα πακέτο που φτάνει σε αυτούς, αποφασίζουν για τον επόμενο κόμβο και τοποθετούν το πακέτο στην ουρά εξόδου για τον επιλεγμένο κόμβο. Οι διασυνδέσεις μετάδοσης έχουν χαρακτηριστικά καθυστέρησης, εύρους ζώνης και αξιοπιστίας. Χαμηλή ποιότητα υπηρεσιών συνήθως προκύπτει όταν ένα τμήμα της κίνησης επιλέγει μια διασύνδεση το εύρος ζώνης μετάδοσης της οποίας, καθίσταται ανεπαρκές για μια εκτεταμένη χρονική περίοδο. Στην περίπτωση αυτή οι ουρές εξόδου του δρομολογητή, που σχετίζονται με την συγκεκριμένη διασύνδεση μετάδοσης, αρχίζουν να γεμίζουν προκαλώντας επιπλέον καθυστέρηση μεταφοράς. Όταν μια ή περισσότερες ουρές του δρομολογητή γεμίσουν, τότε ο δρομολογητής αναγκάζεται να απορρίψει πακέτα. Στη συνέχεια οι προσαρμοζόμενες στα δεδομένα του δικτύου ροές μειώνουν τους ρυθμούς αποστολής δεδομένων προκειμένου να ελαχιστοποιήσουν απώλειες λόγω συμφόρησης, ελαττώνοντας το διαθέσιμο για την εφαρμογή εύρος ζώνης.

Χαμηλή ποιότητα υπηρεσιών μπορεί επίσης να προκληθεί και με άλλους τρόπους. Αστάθεια στα πρωτόκολλα δρομολόγησης μπορεί να αναγκάσει τους δρομολογητές να αλλάξουν άμεσα την επιλογή του καλύτερου επόμενου μονοπατιού. Αυτό έχει ως αποτέλεσμα την αύξηση της κίνησης μέσα στα όρια μιας από άκρο-σε-άκρο ροής δεδομένων και επομένως την εισαγωγή σημαντικών επιπέδων απόκλισης καθυστέρησης και μια αυξημένη πιθανότητα παράδοσης μη ταξινομημένων πακέτων (reliability).

2.2.2 Λόγοι για τους οποίους τα IP δίκτυα δεν παρέχουν ΠοιΥπ

Οι λόγοι για τους οποίους δίκτυα με βάση το πρωτόκολλο του διαδικτύου (Internet Protocol ή IP) δεν υποστήριζαν έως σήμερα ΠοιΥπ είναι διάφοροι. Κατ' αρχήν η οικογένεια πρωτοκόλλων TCP/IP βασίστηκε στη ιδέα της δίκαιης και από κοινού

πρόσβασης σε όλους, χωρίς ειδική μεταχείριση σε κάποια μερίδα δεδομένων. Με εξαίρεση τα τελικά σημεία επικοινωνίας, καμία σύνδεση δεν διατηρείται σε οποιοδήποτε μέρος του δικτύου. Δεύτερον, οι δρομολογητές της προηγούμενης γενιάς (πολλοί από τους οποίους ακόμα βρίσκονται σε λειτουργία) χρησιμοποιούσαν την “πρώτο εισερχόμενο πρώτο εξερχόμενο” (First In First Out-FIFO) στρατηγική στις ουρές τους. Στις περιπτώσεις που περισσότερα πακέτα από όσα ο δρομολογητής μπορούσε να διαχειριστεί έφταναν σε αυτόν, η ουρά γέμιζε και τα αμέσως επόμενα πακέτα που έφταναν απορρίπτονταν.

Σαν ένας ακόμα παράγοντας που λειτούργησε ενάντια στη υποστήριξη ΠοιΥπ από τα IP δίκτυα, μπορεί να θεωρηθεί το γεγονός ότι η επικείμενη κατάρρευση του διαδικτύου λόγω αυξανόμενης κίνησης και FIFO δρομολογητών, αντιμετωπίστηκε με αλλαγές στο TCP πρωτόκολλο έτσι ώστε να μπορεί να προσαρμόσει τους ρυθμούς αποστολής του στη διαθέσιμη χωρητικότητα του δικτύου. Συγκεκριμένα, το TCP άρχισε να ρυθμίζει τους ρυθμούς αποστολής του βάσει των αφίξεων και των συχνοτήτων των βεβαιώσεων λήψης (acknowledgments) που έφταναν από τον παραλήπτη. Στις περιπτώσεις συμφόρησης του δικτύου, ο αποστολέας μείωνε το ρυθμό μετάδοσης πληροφορίας, ενώ όποτε το δίκτυο είχε διαθέσιμη χωρητικότητα ο αποστολέας αύξανε το ρυθμό αποστολής του.

Ένας ακόμη λόγος, είναι ότι το IP και η αρχιτεκτονική του διαδικτύου είναι βασισμένα στην απλή λογική της μεταφοράς των αυτοδύναμων πακέτων (datagrams) με διευθύνσεις πηγής και προορισμού (source, destination), χωρίς τη δημιουργία σύνδεσης μεταξύ αποστολέα και παραλήπτη. Το διαδίκτυο έχει ιστορικά χτιστεί πάνω στην ιδέα ενός “χαζού” δικτύου με “έξυπνα” άκρα (αποστολείς, παραλήπτες) [Saltzer84]. Υπάρχει όμως ένα τίμημα για αυτή την απλότητα. Ο λόγος για τον οποίο το IP είναι απλό, οφείλεται στο ότι δεν προσφέρει πολλές υπηρεσίες. Όλα τα IP πακέτα περιέχουν διεύθυνση προορισμού και αυτό επιτρέπει την ανεξάρτητη δρομολόγηση του κάθε πακέτου. Το IP μπορεί να σπάσει τα πακέτα δεδομένων των υψηλότερων επιπέδων μέσα στους δρομολογητές και να τα συναρμολογήσει στους παραλήπτες, πράγμα που επιτρέπει την διέλευση των πακέτων από δίκτυα διαφορετικού φυσικού μέσου. Όμως, το IP δεν προσφέρει αξιόπιστη μεταφορά δεδομένων. Οι δρομολογητές έχουν την δυνατότητα να απορρίπτουν κάποια πακέτα χωρίς να ειδοποιούν ούτε τον αποστολέα ούτε τον

παραλήπτη. Το IP στηρίζεται σε πρωτόκολλα υψηλότερου επιπέδου (π.χ. TCP), τα οποία είναι υπεύθυνα για την επαναμετάδοση πακέτων που έχουν απορριφθεί από τους δρομολογητές. Αυτοί οι μηχανισμοί αξιοπιστίας δεν μπορούν να εγγυηθούν ασφαλή μετάδοση δεδομένων, ούτε μπορούν να εγγυηθούν έγκαιρη διανομή, ή να προσφέρουν εγγυήσεις για την ΠοιΥπ που παρέχουν. Το IP προσφέρει μια υπηρεσία που ονομάζεται “καλύτερης προσπάθειας” (best-effort) [Lefelhocz96], δηλαδή δεν προσφέρει καμία εγγύηση για το πότε θα φτάσουν τα δεδομένα ή ποσά δεδομένα είναι σε θέση να παραδώσει. Απλά εγγυάται ότι θα χρησιμοποιήσει όλους τους πόρους που έχει διαθέσιμους για να εξυπηρετήσει τα πακέτα. Έτσι γεννήθηκε το ερώτημα, πως θα ήταν δυνατό να προστεθούν υπηρεσίες στο επίπεδο IP χωρίς να προστεθεί πολυπλοκότητα, που θα άλλαζε τα βασικά χαρακτηριστικά του;

Γενικά, στόχος της ΠοιΥπ είναι να παρέχει ένα επίπεδο προβλεψιμότητας και ελέγχου πέρα από την τρέχουσα IP υπηρεσία καλύτερης προσπάθειας. Επίσης, εστιάζει στην παροχή μιας προβλέψιμης υπηρεσίας κατά τη διάρκεια περιόδων συμφόρησης. Συγκεκριμένα στοχεύει στην:

- Ελαχιστοποίηση της καθυστέρησης μεταφοράς.
- Ελαχιστοποίηση της απόκλισης καθυστέρησης.
- Δυνατότητα δέσμευσης πόρων ώστε να ικανοποιείται ένας ελάχιστος ρυθμός μεταφοράς δεδομένων.

Αντίθετα με τις αμιγείς τεχνολογίες εικονικού κυκλώματος (virtual circuit) όπως το ATM (Asynchronous Transfer Mode) και το Frame Relay, το IP δεν δεσμεύει αυστηρά τους πόρους του δικτύου. Αυτό προσφέρει μια πιο αποδοτική χρήση του προσφερομένου εύρους ζώνης και φυσικά είναι πιο ευέλικτο. Άλλωστε καταστάσεις συμφόρησης του δικτύου δεν παρουσιάζονται συνεχώς, αλλά κυρίως σε περιόδους όπου στιγμιαίων εξάρσεων της κίνησης. Βέβαια σε δίκτυα όπου οι απαιτήσεις ξεπερνούν τις δυνατότητες τους, μία τέτοια κατάσταση είναι μόνιμο φαινόμενο. Το IP είναι βασισμένο στη φιλοσοφία μεταφοράς πακέτου και έτσι χρησιμοποιεί το διαθέσιμο εύρος ζώνης πιο αποδοτικά, αφού διαθέτει τους ελεύθερους πόρους όπου είναι αναγκαίο. Αυτό επιτρέπει στο IP να προσαρμόζεται στις εφαρμογές με διαφορετικές απαιτήσεις. Παρόλα αυτά μπορεί να προκύψουν προβλήματα εξυπηρέτησης. Η ικανότητα αντιμετώπισης αυτής της

αστάθειας καθορίζεται από τη συμφωνηθείσα ποιότητα εξυπηρέτησης που υπάρχει ανάμεσα στον παροχό και τον συνδρομητή, αλλά τέτοιες ικανότητες είναι σχετικά περιορισμένες σε IP δίκτυα.

Τέλος, ένας ακόμη λόγος για τον οποίο τα IP δίκτυα δεν υποστηρίζουν ΠοιΥπ είναι ότι, η αναβάθμιση του TCP πρωτοκόλλου για τις ανάγκες της ΠοιΥπ δεν ήταν απαραίτητη όσο κυριαρχούσαν εφαρμογές όπως το HTTP, το FTP, το e-mail κλπ. Οι εφαρμογές αυτές μπορούν να προσαρμόσουν τους ρυθμούς αποστολής δεδομένων τους σε οποιαδήποτε χωρητικότητα προσφέρει το δίκτυο.

2.2.3 Αναγκαιότητα παροχής ΠοιΥπ από το διαδίκτυο

Αποτέλεσμα της τεράστιας αύξησης του διαδικτύου, ήταν να εμφανιστούν οι αδυναμίες του πρωτοκόλλου IP. Η αύξηση του διαθέσιμου εύρους ζώνης για να αποφευχθούν οι κορεσμένες συνδέσεις του είναι η προφανής λύση. Αλλά το πρόβλημα είναι πιο περίπλοκο από ένα απλό ζήτημα χωρητικότητας. Η κυκλοφορία δεδομένων στο διαδίκτυο δεν αυξάνεται απλώς, αλλά αλλάζει και ως προς την φύση της. Υπάρχουν πολλοί νέοι τύποι δικτυακών δεδομένων, παραγόμενοι από διάφορες νέες εφαρμογές, οι οποίες διαφέρουν ως προς τις λειτουργικές τους απαιτήσεις.

Αρκετές από τη νέες εφαρμογές του διαδικτύου είναι πολυμεσικές (multimedia) και απαιτούν σημαντικό εύρος ζώνης. Άλλες έχουν ακριβείς απαιτήσεις συγχρονισμού, ή απαιτήσεις επικοινωνίας ενός-προς-πολλούς (one-to-many) ή πολλών-προς-πολλούς (many-to-many). Αυτού του τύπου οι εφαρμογές απαιτούν υπηρεσίες δικτύων πέρα από την απλή καλύτερη-προσπάθεια που το IP παρέχει σήμερα. Στην πραγματικότητα, απαιτούν από τα σημερινά “χαζά” IP δίκτυα να αποκτήσουν κάποια “νοημοσύνη”. Η παροχή υπηρεσιών τηλεφωνίας από το διαδίκτυο είναι μια από τις εφαρμογές που έχουν προσελκύσει το ενδιαφέρον εταιριών και ερευνητών. Η επιθυμία παροχής τηλεφωνικών υπηρεσιών περνάει αναπόφευκτα από τη σύγκλιση των βιομηχανιών τηλεφώνου και των βιομηχανιών του διαδικτύου. Αυτό είναι αρκετά ενδιαφέρον, μιας και οι σχεδιαστικές αρχές των τηλεφωνικών δικτύων είναι ακριβώς το αντίθετο από εκείνες των δικτύων IP. Το IP χρησιμοποιεί τη μεταγωγή πακέτων και παρέχει την υπηρεσία καλύτερης-προσπάθεια, ενώ τα τηλεφωνικά δίκτυα χρησιμοποιούν την μεταγωγή κυκλώματος παρέχοντας προσανατολισμένες προς τη σύνδεση υπηρεσίες. Το ενδιαφέρον λοιπόν για

την παροχή υπηρεσιών τηλεφωνίας από το διαδίκτυο δεν είναι τυχαίο. Μια διπλής κατεύθυνσης, σε πραγματικό χρόνο τηλεφωνική συνομιλία είναι πολύ απαιτητική εφαρμογή για ένα δίκτυο.

Ένα δίκτυο υπάρχει για να υποστηρίξει τις εφαρμογές. Με αυτό το δεδομένο, ένα δίκτυο μεταβάλλεται σύμφωνα με τις ανάγκες των εφαρμογών. Για την ακρίβεια, ένα δίκτυο μεταβάλλεται για να καλύψει τις ανάγκες των χρηστών. Οι δικτυακές εφαρμογές μπορούν να χαρακτηριστούν [QOS99] με βάση το πόσο προβλέψιμος είναι ο ρυθμός μετάδοσης δεδομένων (Πίνακας 2.1) ή με βάση τον βαθμό ανοχής τους στην καθυστέρηση (Πίνακας 2.2).

Πίνακας 2.1: Χαρακτηρισμός εφαρμογών βάσει της δυνατότητας πρόβλεψης του ρυθμού μετάδοσης δεδομένων

Τύπος ρυθμού μετάδοσης	Περιγραφή
Σταθερής ροής (Stream)	Προβλέψιμη παράδοση με έναν σχετικά σταθερό ρυθμό μετάδοσης (CBR). Σε μερικές από αυτές τις εφαρμογές (π.χ. video και ήχος) ο ρυθμός είναι κυμαινόμενος, αλλά θεωρείται σταθερός επειδή υπάρχουν γνωστά άνω όρια στην μετάδοση.
Εκρηκτικός (Burst)	Απροσδιόριστη παράδοση με έναν μεταβλητό ρυθμό μετάδοσης (VBR). Εφαρμογές όπως η μεταφορά αρχείων μπορεί να μετακινήσουν τέτοιο όγκο δεδομένων, ώστε να χρησιμοποιηθεί όλο το διαθέσιμο εύρος ζώνης (δεν υπάρχει άνω όριο στην μετάδοση).

Πίνακας 2.2: Χαρακτηρισμός εφαρμογών βάσει της ανοχής τους στην καθυστέρηση

Ανοχή στην καθυστέρηση	Τύπος εφαρμογής	Περιγραφή
Υψηλή ↑ ↓ Χαμηλή	Ασύγχρονη (Asynchronous)	Δεν υπάρχουν περιορισμοί στον χρόνο παράδοσης των δεδομένων.
	Σύγχρονη (Synchronous)	Η παράδοση των δεδομένων είναι ευαίσθητη στον χρόνο, αλλά υπάρχει ευελιξία στην καθυστέρηση.
	Αλληλοδραστική (Interactive)	Οι καθυστερήσεις μπορεί να γίνουν εμφανείς στον χρήστη/εφαρμογή, αλλά δεν επηρεάζουν την λειτουργία της εφαρμογής.
	Ισόχρονη (Isochronous)	Εξαρτώμενη από τον χρόνο παράδοσης, σε βαθμό που οι καθυστερήσεις να επηρεάζουν την σωστή λειτουργία την εφαρμογής.
	Επιχειρησιακά-κρίσιμη (mission-Critical)	Οι καθυστερήσεις στην παράδοση δεδομένων καθιστούν την λειτουργία της εφαρμογής αδύνατη.

Η σύγκλιση όλων των δικτύων δεδομένων με το διαδίκτυο φαίνεται σήμερα αναπόφευκτη. Για τα δίκτυα ραδιοφώνου και τηλεόρασης η σύγκλιση αυτή έχει ήδη αρχίσει, αλλά έχει πολύ δρόμο ακόμα. Αυτό που χρειάζεται καταρχάς είναι εύρος ζώνης. Μια άλλη απαίτηση της σύγκλισης αυτής, είναι η παροχή υπηρεσιών πολλαπλής διανομής δεδομένων (multicast). Η προοπτικές μιας τέτοιας σύγκλισης είναι εκπληκτικές και οι νέες εφαρμογές αμέτρητες. Η πολλαπλή διανομή δεδομένων θα επιτρέψει την αμφίπλευρη επικοινωνία των χρηστών με τους παροχείς υπηρεσιών και μεταξύ τους. Έτσι θα υπάρξουν εφαρμογές όπως: ομάδες συζήτησης, διαλογική τηλεμάθηση, παιχνίδια-επιδείξεις και στιγμιαίες έρευνες. Πριν όμως όλα αυτά πραγματοποιηθούν, υπάρχουν άλλες δικτυακές υπηρεσίες οι οποίες πρέπει να υλοποιηθούν, εκτός από την πολλαπλή διανομή δεδομένων. Αυτές είναι: η διπλής κατεύθυνσης και σε πραγματικό χρόνο μετάδοση δεδομένων.

Η αύξηση του εύρους ζώνης του διαδικτύου δεν λύνει όλα τα προβλήματα. Ας πάρουμε για παράδειγμα την παροχή τηλεφωνικών υπηρεσιών από το διαδίκτυο. Παρόλο που πρόκειται για μια πολυμεσική εφαρμογή (ήχος) οι απαιτήσεις της σε εύρος ζώνης είναι σχετικά μικρές (περίπου 8Kbps για κάθε κατεύθυνση). Άρα εδώ το πρόβλημα δεν είναι

το εύρος ζώνης αλλά οι καθυστερήσεις. Για την τηλεφωνία πάνω από το διαδίκτυο, που είναι μια πραγματικού χρόνου και διπλής κατεύθυνσης εφαρμογή, οι απαιτήσεις χρονισμού είναι πολύ πιο σημαντικές από ότι το εύρος ζώνης. Καθυστερήσεις άνω των 0.5 δευτερόλεπτων, καθιστούν την λειτουργία την τηλεφωνικής υπηρεσίας αδύνατη.

Η αύξηση του εύρους ζώνης θα βελτιώσει την ποιότητα υπηρεσίας στο διαδίκτυο και αυτό είναι ένα πρώτο σημαντικό βήμα, αλλά δεν είναι αρκετό για να λύσει το πρόβλημα καθυστερήσεων. Οι καθυστερήσεις δρομολόγησης και η απώλεια πακέτων οφείλονται, όπως έχουμε ήδη αναφέρει, στην παροδική συμφόρηση των δικτύων και στην εκρηκτική φύση της κυκλοφορίας.

Στα πλαίσια της διατριβής θα γίνει μια εκτενής παρουσίαση των των βασικών αρχών, των μηχανισμών και των αρχιτεκτονικών που προτείνονται για την παροχή ΠοιΥπ στα δίκτυα τηλεπικοινωνιών. Η απάντηση λοιπόν στο πρόβλημα παροχής ΠοιΥπ στο διαδίκτυο, θα δοθεί στις επόμενες παραγράφους.

2.3 Το γενικό πλαίσιο για την παροχή ΠοιΥπ

Πριν αναπτυχθούν διεξοδικά οι μηχανισμοί και οι αρχιτεκτονικές παροχής ΠοιΥπ στο διαδίκτυο, θα αναφερθούν κάποιες γενικές σχεδιαστικές αρχές και προδιαγραφές που πρέπει να ακολουθηθούν για την υποστήριξη της.

2.3.1 Γενικές αρχές για την παροχή ΠοιΥπ

Οι βασικότερες αρχές που πρέπει να τηρούνται για την παροχή ΠοιΥπ από ένα δίκτυο αναφέρονται παρακάτω:

- *Η αρχή της ολοκλήρωσης:* Η αρχή αυτή καθορίζει ότι η ΠοιΥπ πρέπει να είναι διαχειρίσιμη και προβλέψιμη στην υλοποίησή της σε όλα τα επίπεδα της αρχιτεκτονικής, ώστε να παρέχεται από άκρο-σε-άκρο [Kurose93].
- *Η αρχή του διαχωρισμού:* Η αρχή αυτή καθορίζει ότι η μεταφορά περιεχομένου, ο έλεγχος και η διαχείριση είναι ξεχωριστές δραστηριότητες. Οι τρεις αυτές δραστηριότητες πρέπει να είναι διακριτές σε ένα γενικό πλαίσιο ΠοιΥπ [Lazar92].
- *Η αρχή της διαφάνειας:* Η αρχή αυτή καθορίζει ότι οι εφαρμογές που χρησιμοποιούν ΠοιΥπ πρέπει να είναι απομονωμένες από τις λειτουργίες που υλοποιούν την ΠοιΥπ.
- *Η αρχή των διακριτών δραστηριοτήτων:* Η αρχή αυτή καθορίζει ότι οι λειτουργίες που υλοποιούν την ΠοιΥπ, είναι κατανεμημένες σε διάφορα τμήματα του συστήματος [Lazar92].
- *Η αρχή της απόδοσης:* Η αρχή αυτή καθορίζει ότι κάθε εφαρμογή που χρησιμοποιεί ΠοιΥπ πρέπει να έχει μια ελάχιστη απόδοση.

2.3.2 Προδιαγραφές για την παροχή ΠοιΥπ

Για την παροχή και διατήρηση της ΠοιΥπ τόσο στο ίδιο το δίκτυο όσο και στους τελικούς χρηστές, έχουν οριστεί μια σειρά από προδιαγραφές. Οι προδιαγραφές αυτές είναι διαφορετικές για κάθε επίπεδο του συστήματος και περιγράφονται παρακάτω:

- *Προδιαγραφή για συγχρονισμό της ροής:* Ορίζει τον βαθμό συγχρονισμού μεταξύ πολλών αλληλεξαρτημένων ροών.

- *Προδιαγραφή για την απόδοση των ροών:* Ορίζει τις ανάγκες του χρήστη σε απόδοση ροών.
- *Προδιαγραφή για το επίπεδο της υπηρεσίας:* Καθορίζει την από άκρο-σε-άκρο διαθεσιμότητα πόρων που απαιτείται.
- *Προδιαγραφή για την διαχείριση της ΠοιΥπ:* Καθορίζει τις μεταβολές στην παροχή της ΠοιΥπ που μπορεί να αντιμετωπίσει κάθε είδος κίνησης.
- *Προδιαγραφή για το κόστος της υπηρεσίας:* καθορίζει το ποσό που είναι διατεθειμένος να πληρώσει ένας χρήστης για κάποιο επίπεδο υπηρεσίας.

2.4 Μηχανισμοί παροχής ΠοιΥπ

Οι μηχανισμοί παροχής ΠοιΥπ επιλέγονται και διαμορφώνονται σύμφωνα με τις προδιαγραφές που θέτει ο χρήστης, την διαθεσιμότητα των πόρων και την πολιτική διαχείρισης των. Στη διαχείριση των πόρων, οι μηχανισμοί παροχής ποιότητας υπηρεσίας ταξινομούνται είτε ως στατικοί είτε ως δυναμικοί. Η στατική διαχείριση των πόρων ασχολείται με την δημιουργία της ροής δεδομένων και την από άκρο-σε-άκρο επαναδιαπραγμάτευσης της παρεχόμενης ΠοιΥπ και αναφέρεται ως προπαρασκευή της ΠοιΥπ. Η δυναμική διαχείριση των πόρων ασχολείται με την καθεαυτού μεταφορά των δεδομένων και αναφέρεται ως έλεγχος και διαχείριση ΠοιΥπ. Η διάκριση μεταξύ του ελέγχου και της διαχείρισης ΠοιΥπ βασίζεται στα διαφορετικά χρονικά πλαίσια, στα οποία οι δυο αυτοί μηχανισμοί λαμβάνουν χώρα. Συγκεκριμένα ο μηχανισμός έλεγχου ΠοιΥπ λειτουργεί σε πολύ στενότερα χρονικά πλαίσια από ότι ο μηχανισμός διαχείρισης. Σύμφωνα λοιπόν με όσα αναφέρθηκαν παραπάνω οι μηχανισμοί παροχής ΠοιΥπ διακρίνονται σε [Augrecoechea96]:

- Μηχανισμούς προπαρασκευής ΠοιΥπ
- Μηχανισμούς έλεγχου ΠοιΥπ
- Μηχανισμούς διαχείρισης ΠοιΥπ

Οι μηχανισμοί αυτοί περιγράφονται αναλυτικά στις επόμενες παραγράφους.

2.4.1 Μηχανισμοί προπαρασκευής ΠοιΥπ

Ο μηχανισμός προπαρασκευής ΠοιΥπ είναι υπεύθυνος για την από άκρο-σε-άκρο δημιουργία της ροής δεδομένων. Για την προπαρασκευή της ΠοιΥπ, ένα σύστημα θα πρέπει να διαθέτει τις παρακάτω λειτουργίες:

Αντιστοίχιση (Mapping): Εκτελεί τη λειτουργία της αυτόματης μετάφρασης των αντιπροσωπεύσεων της ΠοιΥπ, μεταξύ των διαφορετικών επιπέδων του συστήματος (π.χ λειτουργικό σύστημα, επίπεδο μεταφοράς, επίπεδο δίκτυο, κ.λπ.). Με αυτόν τον τρόπο, ο χρήστης δεν είναι αναγκασμένος να γνωρίζει λεπτομέρειες λειτουργίας των χαμηλότερων επιπέδων του συστήματος. Παραδείγματος χάριν, σε επίπεδο μεταφοράς δεδομένων η ΠοιΥπ μπορεί να εκφράσει τις απαιτήσεις ροής ως περιορισμούς στο μέσο και μέγιστο

εύρους ζώνης, στην απόκλιση καθυστέρησης, στις απώλειες και την καθυστέρηση μετάδοσης. Για να πραγματοποιηθεί όμως ο έλεγχος αποδοχής και η δέσμευση πόρων (οι όροι αυτοί εξηγούνται παρακάτω) οι απατήσεις αυτές θα πρέπει να μεταφραστούν σε κάτι που να έχει περισσότερο νόημα για το σύστημα του τελικού χρήστη.

Δοκιμή αποδοχής ή Έλεγχος αποδοχής (Admission testing, Admission control): Η λειτουργία αυτή είναι υπεύθυνη για την σύγκριση των πόρων, που απαιτούνται από μια αίτηση παροχής ΠοιΥπ, με τους διαθέσιμους πόρους του συστήματος. Η απόφαση εάν ένα νέο αίτημα θα γίνει αποδεκτό, εξαρτάται από την πολιτική διαχείρισης πόρων του συστήματος και από την διαθεσιμότητα τους.

Δέσμευση πόρων (Resource reservation): Με την λειτουργία αυτή πραγματοποιείται η δέσμευση των πόρων του συστήματος για την δημιουργία μιας από άκρο σε άκρο ροής δεδομένων. Το τελικό αποτέλεσμα αυτής της λειτουργίας, σε επίπεδο δικτύου, είναι η κατάλληλη διάρθρωση μιας σειράς δρομολογητών

2.4.2 Μηχανισμοί έλεγχου ΠοιΥπ

Οι μηχανισμοί ελέγχου ΠοιΥπ λειτουργούν σε ταχύτητες κοντά σε αυτές της μεταφοράς δεδομένων. Παρέχουν σε πραγματικό χρόνο έλεγχο της κυκλοφορίας των ροών, που δημιουργούνται κατά την φάση προπαρασκευής ΠοιΥπ. Οι θεμελιώδεις μηχανισμοί έλεγχου ΠοιΥπ περιλαμβάνουν τα εξής:

Διαμόρφωση ροής (Flow shaping): Ρυθμίζει τις ροές δεδομένων με βάση τις προδιαγραφές απόδοσης ροής που καθορίζει ο χρήστης. Ο μηχανισμός αυτός επιτρέπει την από άκρο σε άκρο δέσμευση πόρων και την διαμόρφωση της κυκλοφορίας, μέσα στο υπό διαχείριση δίκτυο [Ferrari90].

Σχεδιασμός ροής (Traffic Scheduling): Ο μηχανισμός αυτός διαχειρίζεται την αποστολή των ροών από τους τελικούς χρηστές και από το δίκτυο με έναν ενοποιημένο τρόπο [Zhang91]. Οι ροές σχεδιάζονται γενικά ανεξάρτητα στους τελικούς χρηστές του συστήματος, αλλά μπορεί να συναθροιστούν μέσα στο δίκτυο. Αυτό εξαρτάται από το επίπεδο υπηρεσίας και την πολιτική για το σχεδιασμό της ροής που υιοθετείται στο δίκτυο [Kurose93].

Επιτήρηση ροής (Traffic policing): Ο μηχανισμός αυτός ελέγχει αν η ΠοιΥπ που παρέχεται στον χρήστη από το σύστημα, είναι η ίδια με αυτήν που απαίτησε ο χρήστης. Η επιτήρηση ροής συνήθως χρησιμοποιείται για την ανίχνευση ροών δεδομένων που δεν συμπεριφέρονται σωστά.

Έλεγχος ροής (flow control): Ο μηχανισμός αυτός επιτρέπει στον αποστολέα να στείλει δεδομένα μέσα στα συμφωνηθέντα επίπεδα αποστολής, δεδομένου ότι οι πόροι του δικτύου έχουν δεσμευθεί εκ των προτέρων.

Συγχρονισμός ροής (flow synchronization): Ο μηχανισμός αυτός είναι απαραίτητος για τον έλεγχο της αλληλουχίας γεγονότων αλλά και τον ακριβή συγχρονισμό, που απαιτείται στις εφαρμογές πολυμέσων.

2.4.3 Μηχανισμός διαχείρισης ΠοιΥπ

Για να διατηρήσουν τα συμφωνηθέντα επίπεδα ΠοιΥπ, η δέσμευση απλώς των πόρων δεν είναι αρκετή. Η διαχείριση απαιτείται για να εξασφαλίσει ότι η ΠοιΥπ είναι και συνεχής. Η διαχείριση ΠοιΥπ είναι λειτουργικά παρόμοια με τον έλεγχο ΠοιΥπ, αλλά λειτουργεί σε ένα πιο αργό χρονικό πλαίσιο. Οι θεμελιώδεις μηχανισμοί διαχείρισης ΠοιΥπ περιλαμβάνουν τα εξής:

Παρακολούθηση ΠοιΥπ (Monitoring): Επιτρέπει σε κάθε επίπεδο του συστήματος να παρακολουθήσει την ΠοιΥπ που επιτυγχάνεται από τα χαμηλότερα επίπεδα. Η παρακολούθηση την ΠοιΥπ είναι ένα αναπόσπαστο τμήμα ενός συστήματος ανατροφοδότησης πληροφοριών, απαραίτητο για την διατήρηση της ΠοιΥπ στα συμφωνημένα επίπεδα, μετά την δέσμευση των πόρων.

Συντήρηση ΠοιΥπ (Maintenance): Ο μηχανισμός αυτός συγκρίνει την παρακολουθούμενη ΠοιΥπ με την αναμενόμενη και έπειτα, αν χρειάζεται, ασκεί διορθωτικές ενέργειες για να διασφαλίσει το συμφωνημένο επίπεδο ΠοιΥπ.

Υποβάθμιση ΠοιΥπ (Degradation): Ο μηχανισμός αυτός αποστέλλει μια ένδειξη στον χρήστη, ότι τα χαμηλότερα στρώματα έχουν αποτύχει να διατηρήσουν την ΠοιΥπ της ροής και τίποτα δεν μπορεί να γίνει περαιτέρω από το μηχανισμό συντήρησης ΠοιΥπ. Ως απάντηση προς μια τέτοια ένδειξη, ο χρήστης μπορεί να επιλέξει είτε να προσαρμοστεί

στο διαθέσιμο επίπεδο ΠοιΥπ, είτε να επαναδιαπραγματευτεί από άκρο-σε-άκρο ένα νέο μειωμένο επίπεδο υπηρεσίας.

Διαθεσιμότητα ΠοιΥπ (Availability): Επιτρέπει στην εφαρμογή διαχείρισης να διευκρινίζει το διάστημα στο οποίο μια ή περισσότερες παράμετροι ΠοιΥπ (π.χ. η καθυστέρηση, η απόκλιση καθυστέρησης, το εύρος ζώνης, οι απώλειες) θα ελέγχονται και η εφαρμογή θα ενημερώνεται για την απόδοση, μέσω ενός σήματος ΠοιΥπ [Campbell94].

2.5 Αρχιτεκτονικές παροχής ΠοιΥπ

Στις προηγούμενες παραγράφους αναφέρθηκαν οι βασικές θεωρητικές αρχές για την παροχή ΠοιΥπ σε ένα δίκτυο επικοινωνιών. Με βάση λοιπόν αυτές τις αρχές, έγινε μια προσπάθεια από διάφορες ομάδες ερευνητών να αναπτυχθούν οι αρχιτεκτονικές παροχής ΠοιΥπ στο διαδίκτυο. Το εγχείρημα αυτό δεν ήταν (και δεν είναι) καθόλου εύκολο. Η υπηρεσία “καλύτερης προσπάθειας” στην οποία στηρίχτηκε το διαδίκτυο βασίζεται στην ύπαρξη ενός απλού δικτύου με “έξυπνα” άκρα. Έτσι λοιπόν όλες οι αρχιτεκτονικές ή πρωτόκολλα που αναπτύχθηκαν, παρά τις διαφορετικές ιδέες πάνω στις οποίες στηρίχτηκαν, προσπάθησαν να διατηρήσουν αυτή την βασική αρχή.

Η βασική αιτία για την ανάπτυξη διαφορετικών αρχιτεκτονικών ήταν οι διαφορετικές απαιτήσεις που έχουν οι δικτυακές εφαρμογές σε ΠοιΥπ. Η ερευνά για την παροχή ΠοιΥπ κινήθηκε προς δυο κατευθύνσεις, έτσι προέκυψαν οι δυο παρακάτω βασικοί τύποι [QoSF99]:

- **Με Δέσμευση πόρων:** οι δικτυακοί πόροι κατανέμονται σύμφωνα με την αίτηση για παροχή ΠοιΥπ μιας εφαρμογής και ανάλογα με την πολιτική διαχείρισης του εύρους ζώνης.
- **Με Ανάθεση προτεραιότητας:** η δικτυακή κυκλοφορία κατηγοριοποιείται και οι δικτυακοί πόροι κατανέμονται σύμφωνα με τα κριτήρια της πολιτικής διαχείρισης εύρους ζώνης. Για να καταστεί δυνατή η παροχή ΠοιΥπ, τα δικτυακά στοιχεία δίνουν προτεραιότητα στις κατηγορίες που έχουν υψηλότερες απαιτήσεις.

Αυτοί οι τύποι παροχής ΠοιΥπ μπορεί να εφαρμοστούν σε ατομικές ροές εφαρμογών ή σε σύνολα ροών, οπότε προκύπτουν δύο άλλοι τρόποι με τους οποίους μπορεί χαρακτηριστεί η ΠοιΥπ:

- **Ανά ροή:** Μια ροή ορίζεται σαν ένα ατομικό, μονοκατευθυντικό ρεύμα δεδομένων μεταξύ δύο εφαρμογών (αποστολέας και παραλήπτης), που αναγνωρίζεται από μια πεντάδα στοιχείων (πρωτόκολλο μεταφοράς, διεύθυνση πηγής, αριθμός θύρας της πηγής, διεύθυνση προορισμού και αριθμός θύρας του προορισμού).

- **Ανά σύνολο ροών:** Ένα τέτοιο σύνολο είναι απλά δύο ή περισσότερες ροές. Τυπικά, οι ροές θα έχουν κάτι κοινό (π.χ. μία ή περισσότερες από τις παραμέτρους της πεντάδας στοιχείων, μια ετικέτα ή αριθμό προτεραιότητας, ή κάποια πληροφορία πιστοποίησης).

Οι τρεις σημαντικότερες αρχιτεκτονικές προτάσεις για παροχή ΠοιΥπ σε IP δίκτυα είναι οι εξής:

Ολοκληρωμένες Υπηρεσίες-Integrated Services (IntServ): Χρησιμοποιείται για τη δέσμευση πόρων σε σύνολα ροών. Το πρωτόκολλο δέσμευσης **ReSerVation Protocol (RSVP)** παρέχει την απαραίτητη σηματοδότηση για την δέσμευση των δικτυακών πόρων.

Διαφοροποιημένες Υπηρεσίες-Differentiated Services (DiffServ): Παρέχει ένα συνηθισμένο και απλό τρόπο για να κατηγοριοποιεί και να θέτει προτεραιότητες στα σύνολα της δικτυακής κυκλοφορίας (ροές).

Πολυ-πρωτόκολλο Μεταγωγής Ετικέτας-Multi Protocol Label Switching (MPLS): Παρέχει διαχείριση του εύρους ζώνης για σύνολα ροών μέσω ελέγχου δρομολόγησης, σύμφωνα με ετικέτες στις επικεφαλίδες των πακέτων.

Ο τρόπος λειτουργίας των παραπάνω αρχιτεκτονικών, τα πλεονεκτήματα και τα μειονεκτήματά τους, καθώς και οι ομοιότητες και οι διαφορές τους παρουσιάζονται αναλυτικά στην συνέχεια.

2.5.1 Αρχιτεκτονική Ολοκληρωμένων Υπηρεσιών

Ο οργανισμός Internet Engineering Task Force [IETF], ανταποκρινόμενος στην απαίτηση για παροχή ολοκληρωμένων υπηρεσιών στο διαδίκτυο, προχώρησε στην ανάπτυξη της αρχιτεκτονικής Ολοκληρωμένων Υπηρεσιών (Integrated Services (IntServ)) [IntServ]. Η αρχιτεκτονική αυτή χαρακτηρίζεται από μια ισχυρή δόση πολυπλοκότητας. Σχεδιάστηκε αρχικά για να παρέχει ένα σύνολο προεκτάσεων στο παραδοσιακό μοντέλο μετάδοσης “καλύτερης προσπάθειας”. Η αρχική της δομή είχε σκοπό να παρέχει κάποια ιδιαίτερη μεταχείριση σε ορισμένους τύπους κυκλοφορίας και

ένα μηχανισμό στις εφαρμογές, ώστε να έχουν τη δυνατότητα να επιλέξουν ανάμεσα σε πολλά επίπεδα υπηρεσιών για την μετάδοση της κυκλοφορία τους.

2.5.1.1 Κλάσεις Υπηρεσιών

Το πρώτο μέλημα της ομάδας εργασίας ανάπτυξης των ολοκληρωμένων υπηρεσιών, ήταν να ορίσει ένα νέο σύνολο υπηρεσιών. Οι κλάσεις υπηρεσιών λοιπόν που ορίστηκαν είναι οι ακόλουθες δυο:

- Υπηρεσία Ελεγχόμενου Φόρτου (Controlled Load Service-CLS) [Wroclawski97A].
- Υπηρεσία με Εγγύηση (Guaranteed Service-GS) [Shenker97].

Η υπηρεσία ελεγχόμενου φόρτου δεν παρέχει ποσοτικές εγγυήσεις αναφορικά με την ποιότητα της υπηρεσίας. Προσεγγίζει τη συμπεριφορά της υπηρεσίας καλύτερης προσπάθειας σε συνθήκες χαμηλού φόρτου δικτύου. Η βασική διαφορά μεταξύ των δύο υπηρεσιών είναι ότι η υπηρεσία ελεγχόμενου φόρτου δεν υποβαθμίζεται όσο ο φόρτος του δικτύου αυξάνει. Αντίθετα, παραμένει τόσο καλή όσο η υπηρεσία καλύτερης προσπάθειας σε ένα δίκτυο χαμηλού φόρτου.

Με την υπηρεσία αυτή, αναμένεται ότι ένα πολύ μεγάλο ποσοστό των μεταδιδόμενων πακέτων θα φθάσουν στον προορισμό τους με επιτυχία. Το ποσοστό των πακέτων που δε μεταδόθηκαν με επιτυχία πρέπει να προσεγγίζει το βασικό ρυθμό λανθασμένων πακέτων (basic error rate) του μέσου μετάδοσης. Επιπλέον, η καθυστέρηση διάβασης (transit delay) που συναντούν τα περισσότερα πακέτα, δεν πρέπει να ξεπερνά κατά πολύ την ελάχιστη καθυστέρηση μετάδοσης, οποιουδήποτε πακέτου που έχει μεταδοθεί με επιτυχία. Για να διασφαλιστεί η ισχύς των παραπάνω, θα πρέπει να παρέχεται μια εκτίμηση της κίνησης των δεδομένων που στέλνονται στο δίκτυο. Αυτή η εκτίμηση ονομάζεται Ένδειξη Προδιαγραφών Κίνησης (Traffic specifications-Tspec). Με αυτό τον τρόπο, η υπηρεσία εξασφαλίζει ότι η κίνηση που συμμορφώνεται με το Tspec θα εξυπηρετείται κατά τα συμφωνημένα. Αν όμως τα χαρακτηριστικά της κίνησης μεταβληθούν έξω από τα όρια του Tspec, η ροή θα αντιμετωπιστεί όπως αν ήταν σε ένα υπερφορτωμένο δίκτυο χωρίς κρατήσεις. Θα παρατηρηθούν δηλαδή, μεγάλες καθυστερήσεις ή ακόμα και απώλεια πακέτων.

Η υπηρεσία με εγγύηση εξασφαλίζει ένα συγκεκριμένο εύρος ζώνης, μια σταθερή καθυστέρηση κατά τη μεταφορά των δεδομένων από άκρο σε άκρο, ενώ εγγυάται ότι δε θα χάνονται πακέτα στις ουρές του συστήματος, με την προϋπόθεση ότι τα πακέτα ανήκουν σε ροή που συμμορφώνεται με τις προδιαγραφές κίνησης. Δεν αποσκοπεί στο να βελτιώσει την απόκλιση καθυστέρησης, δηλαδή τη διαφορά μεταξύ της ελάχιστης και της μέγιστης καθυστέρησης πακέτου, παρά μόνο τη μέγιστη καθυστέρηση. Γενικά, έχει ως στόχο να καλύψει εφαρμογές που έχουν αυστηρές απαιτήσεις για μεταφορά δεδομένων σε πραγματικό χρόνο. Τέτοιες είναι κατεξοχήν εφαρμογές που διακινούν ήχο ή και εικόνα.

Η υπηρεσία με εγγύηση προσεγγίζει το “ρευστό μοντέλο” (fluid model) υπηρεσίας, το οποίο ορίζει ότι υπηρεσία με ρυθμό εξυπηρέτησης R ισοδυναμεί με την υπηρεσία που παρέχεται από ένα καλώδιο εύρους ζώνης R , το οποίο βρίσκεται αποκλειστικά στη διάθεση του δέκτη. Η υπηρεσία αυτή βασίζεται ουσιαστικά στο γεγονός ότι η καθυστέρηση που ορίζεται από το “ρευστό μοντέλο” για μια ροή με προδιαγραφές κίνησης (r,b) του κάδου με κουπόνι (token bucket), η οποία εξυπηρετείται από μια γραμμή εύρους ζώνης R , φράσσεται από τον όρο b/R , όσο το $R \leq r$. Η υπηρεσία με εγγύηση, με ρυθμό εξυπηρέτησης R , προσεγγίζει αυτή τη συμπεριφορά.

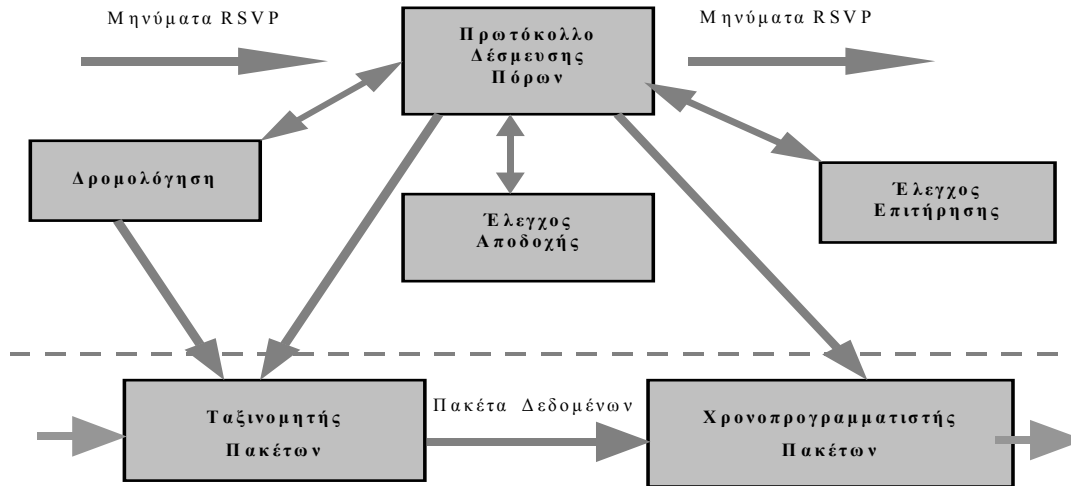
2.5.1.2 Το μοντέλο Ολοκληρωμένων Υπηρεσιών διαδικτύου

Για να υποστηριχθούν οι ολοκληρωμένες υπηρεσίες διαδικτύου, χρειάστηκε να οριστεί και να αναπτυχθεί ένα πλαίσιο στο οποίο θα κινούνται. Για το σκοπό αυτό έχουν οριστεί τέσσερα βασικά τμήματα στα οποία περιλαμβάνονται τα εξής: έλεγχος αποδοχής (admission control), έλεγχος επιτήρησης (policy control), δρομολόγηση και ταξινόμηση πακέτων (routing packet classification) και το πρωτόκολλο εγκατάστασης και δέσμευσης πόρων (Resource **ReSerVation Protocol-RSVP**).

Αρχικά να εξηγήσουμε τον όρο “ροή δεδομένων”. Με τον όρο αυτό εννοούμε μια ομάδα από διαδοχικά πακέτα που έχουν παραχθεί από την ίδια δραστηριότητα ενός χρήστη και υπόκεινται στην ίδια ΠοιΥπ. Επίσης θεωρείται ότι μια ροή δεδομένων είναι “simplex”, δηλαδή προέρχεται από μια μοναδική πηγή αλλά μπορεί να έχει N προορισμούς.

Οι ολοκληρωμένες υπηρεσίες διαδικτύου ορίζουν διαφορετικές κλάσεις υπηρεσιών. Η λειτουργία των δρομολογητών, που εξασφαλίζει διαφορετική ποιότητα υπηρεσιών,

ονομάζεται “έλεγχος κυκλοφορίας” (traffic control). Ο έλεγχος κυκλοφορίας με τη σειρά του υλοποιείται από τρία τμήματα του μοντέλου ολοκληρωμένων υπηρεσιών. Τα τμήματα αυτά είναι: ο χρονοπρογραμματιστής πακέτων (Packet Scheduler), ο κατηγοριοποιητής πακέτων (Packet Classifier) και ο έλεγχος αποδοχής (Admission Control). Το μοντέλο ολοκληρωμένων υπηρεσιών παρουσιάζεται στο σχήμα 2.2. Στην συνέχεια αναλύεται η λειτουργία κάθε τμήματος του μοντέλου.



Σχήμα 2.2: Το μοντέλο Ολοκληρωμένων Υπηρεσιών

Χρονοπρογραμματιστής πακέτων

Το λειτουργικό τμήμα του χρονοπρογραμματιστή πακέτων είναι υπεύθυνο για την προώθηση των διαφορετικών ροών, χρησιμοποιώντας για το σκοπό αυτό ένα σύνολο από ουρές και άλλους μηχανισμούς όπως χρονομετρητές (timers). Ο χρονοπρογραμματιστής πακέτων αναλαμβάνει να παρέχει στις ροές δεδομένων την ΠοιΥπ που έχουν ζητήσει. Πολυπλέκει πακέτα που ανήκουν σε διαφορετικές ροές, μαζί με εκείνα που ανήκουν στην υπηρεσία καλύτερης προσπάθειας, προς τις εξόδους του δρομολογητή. Ουσιαστικά αποτελεί τη λειτουργική μονάδα που υλοποιεί και παρέχει την ζητούμενη ΠοιΥπ. Οι λεπτομέρειες του αλγορίθμου χρονοπρογραμματισμού είναι διαφορετικές για κάθε μέσο εξόδου.

Ταξινομητής πακέτων

Κάθε πακέτο που εισέρχεται σε ένα δρομολογητή θα πρέπει να αντιστοιχεί σε κάποια κλάση υπηρεσίας, για να διευκολύνεται ο έλεγχος της κυκλοφορίας και ο μηχανισμός χρέωσης. Αυτή η αντιστοίχιση των πακέτων σε κλάσεις που θα λάβουν την ίδια ΠοιΥπ,

πραγματοποιείται από τον ταξινομητή πακέτων. Η κλάση στην οποία θα εισαχθεί κάποιο πακέτο προσδιορίζεται από τα περιεχόμενα της επικεφαλίδας και από έναν επιπλέον αριθμό (classification number), που προστίθεται σε αυτό. Μια κλάση μπορεί να αντιστοιχεί σε μια μεγάλη κατηγορία ροών, π.χ. όλες οι ροές με πληροφορία εικόνας, ή όλες οι ροές μιας συγκεκριμένης πηγής. Μπορεί όμως να περιέχει και μια μόνο ροή.

Έλεγχος αποδοχής

Εδώ υλοποιείται ο αλγόριθμος μέσω του οποίου ένας δρομολογητής αποφασίζει αν μια νέα ροή θα γίνει αποδεκτή ή όχι, χωρίς όμως να επιβαρύνει την ΠοιΥπ που παρέχεται σε άλλες υπάρχουσες ροές δεδομένων. Ο έλεγχος αποδοχής ενεργοποιείται σε κάθε κόμβο, κατά μήκος του μονοπατιού που συνδέει τα δύο άκρα, για να αποδεχθεί ή να απορρίψει την αίτηση για δέσμευση πόρων. Η αποδοχή των αιτήσεων είναι σε άμεση συμφωνία με το είδος της υπηρεσίας που έχει ζητηθεί και με βάση αυτή δέχεται ή όχι νέες αιτήσεις.

Έλεγχος επιτήρησης

Το λειτουργικό τμήμα του έλεγχου επιτήρησης είναι υπεύθυνο για τον έλεγχο των χρηστών που ζητούν κάποια συγκεκριμένη υπηρεσία. Στο τμήμα αυτό αποφασίζεται αν ο χρήστης έχει την άδεια να χρησιμοποιήσει την υπηρεσία την οποία ζητά ή όχι. Ο έλεγχος επιτήρησης παρουσιάζει ιδιαίτερο ενδιαφέρον στην περίπτωση εκείνη που υλοποιείται κάποιος μηχανισμός χρέωσης των χρηστών, αφού θα πρέπει να ενσωματώνει την πολιτική χρέωσης.

Δρομολόγηση

Το τμήμα της δρομολόγησης παρέχει σαφείς πληροφορίες για τη δρομολόγηση κάθε πακέτου. Έτσι, προσδιορίζεται η διεύθυνση του επόμενου κόμβου στον οποίο πρέπει να αποσταλεί, καθώς και η μονάδα εξόδου του δρομολογητή από όπου θα πρέπει να προωθηθεί κάθε πακέτο.

2.5.1.3 Το πρωτόκολλο RSVP

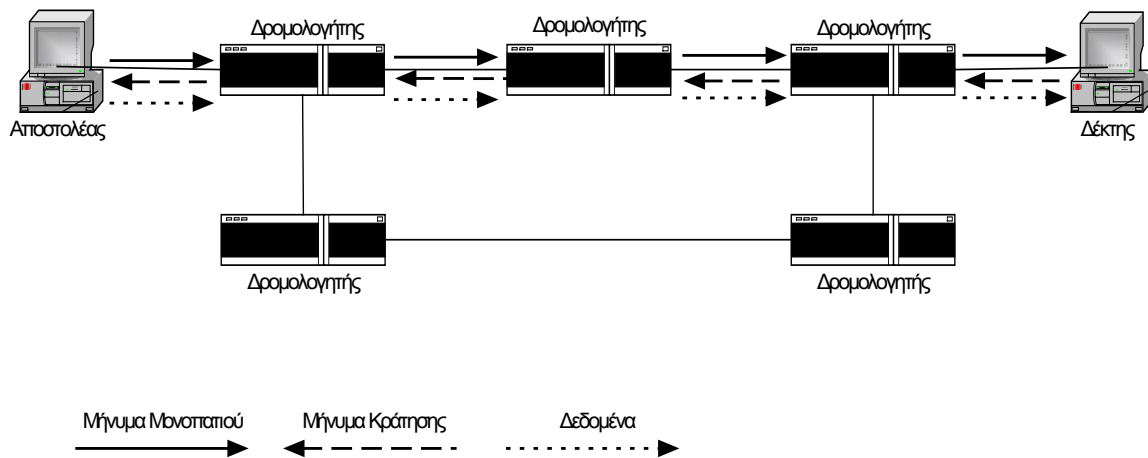
Το RSVP [Braden97] αποτελεί σήμερα μια ευρύτατα αποδεκτή λύση στο πρόβλημα παροχής ΠοιΥπ, σε IP δίκτυα. Παρέχει τη δυνατότητα δέσμευσης πόρων τόσο για εφαρμογές απλής διανομής δεδομένων (unicast) όσο και για εφαρμογές πολλαπλής διανομής δεδομένων (multicast). Υπάρχουν και άλλα πρωτόκολλα που έχουν σχεδιαστεί

για τον ίδιο σκοπό, αλλά το RSVP παρουσιάζει ξεχωριστή ευελιξία και επεκτασιμότητα, εξαιτίας των καινοτομιών που εφαρμόστηκαν κατά το σχεδιασμό του:

- Οι πόροι δεσμεύονται με δυναμικό τρόπο, ανάλογα με το πως μεταβάλλονται οι ανάγκες.
- Το RSVP δεσμεύει πόρους μόνο προς τη μία κατεύθυνση ενώ την πρωτοβουλία για τη δέσμευση έχει το άκρο εκείνο που πρόκειται να λάβει τα δεδομένα. Πρέπει να τονιστεί ότι το RSVP δεν είναι πρωτόκολλο δρομολόγησης, αλλά πρωτόκολλο σηματοδοσίας. Για την αποδοτικότερη λειτουργία θα πρέπει να υποστηρίζεται από ένα πρωτόκολλο δρομολόγησης, το οποίο θα μπορεί να λαμβάνει αποφάσεις δρομολόγησης με βάση τη ζητούμενη ΠοιΥπ.
- Το RSVP μπορεί να λειτουργήσει τόσο πάνω από IPv4 όσο και από IPv6.

Το RSVP σχεδιάστηκε από μια ομάδα εργασίας της IETF, με σκοπό να υποστηρίξει τα είδη υπηρεσιών που ορίστηκαν από το μοντέλο ολοκληρωμένων υπηρεσιών [Wroclawski97B]. Ωστόσο είναι απαραίτητος ένας λογικός διαχωρισμός ανάμεσα στις ολοκληρωμένες υπηρεσίες και το RSVP. Το RSVP σχεδιάστηκε για να χρησιμοποιείται από μια ποικιλία υπηρεσιών ελέγχου ΠοιΥπ, όπως και οι υπηρεσίες σχεδιάστηκαν για να χρησιμοποιούνται από μια ποικιλία μηχανισμών. Το RSVP δεν είναι ένα πρωτόκολλο δρομολόγησης αλλά πρωτόκολλο σηματοδοσίας. Δεν υπολογίζει από μόνο του διαδρομές αλλά χρησιμοποιεί τα υπάρχοντα IP πρωτόκολλα δρομολόγησης. Μεταφέρει αιτήσεις κρατήσεων πόρων, συμπεριλαμβανομένων των προδιαγραφών κυκλοφορίας (Traffic specs), των προδιαγραφών ΠοιΥπ (QoS specs) και της διαθεσιμότητας πόρων του δικτύου. Πραγματοποιεί κρατήσεις πόρων σε κάθε δρομολογητή και για αυτό πρέπει να είναι υλοποιημένο σε όλους τους δρομολογητές του μονοπατιού που θα δεσμευθεί. Επειδή το RSVP είναι πρωτόκολλο μονής κατεύθυνσης, οι δεσμεύσεις γίνονται μόνο προς τη μια κατεύθυνση. Αποκαθιστά και διατηρεί μια *χαλαρή κατάσταση* σε κάθε κόμβο του μονοπατιού που έχει δεσμευθεί. Αυτό επιτυγχάνεται με περιοδικά μηνύματα ανανέωσης που στέλνονται κάθε 30 δευτερόλεπτα, για τη διατήρηση της κατάστασης δέσμευσης του μονοπατιού. Σε περίπτωση απουσίας αυτών των μηνυμάτων η κατάσταση που έχει δημιουργηθεί καταργείται.

Η λειτουργία του RSVP παρουσιάζεται στο Σχήμα 2.3 και είναι η εξής: Ο αποστολέας στέλνει περιοδικά ένα μήνυμα μονοπατιού (path message), το οποίο περιγράφει τον τύπο της κυκλοφορίας που θα αποσταλεί και τους πόρους που είναι απαραίτητοι για να την υποστηρίξουν. Τα μηνύματα μονοπατιού χρησιμοποιούνται για να αποθηκεύουν πληροφορία σχετική με το μονοπάτι σε κάθε κόμβο. Ένας δέκτης που λαμβάνει το path message ανταποκρίνεται στέλνοντας ένα μήνυμα κράτησης (Resv message) προς τον αποστολέα, διαμέσου του ίδιου μονοπατιού που ακολούθησε και το path message. Επειδή ο δέκτης είναι υπεύθυνος για την απαίτηση των επιθυμητών υπηρεσιών, τα μηνύματα κράτησης προσδιορίζουν την επιθυμητή ΠοιΥπ και πραγματοποιούν τις δεσμεύσεις σε κάθε κόμβο του μονοπατιού (αν υποτεθεί ότι υπάρχουν διαθέσιμοι οι πόροι). Αφού το μήνυμα κράτησης φτάσει στον αποστολέα, αποκαθίσταται η δέσμευση από άκρο-σε-άκρο και ο αποστολέας αρχίζει να μεταδίδει τα δεδομένα.



Σχήμα 2.3: Λειτουργία του RSVP

2.5.2 Διαφοροποιημένες Υπηρεσίες

Η πρώτη προσπάθεια για παροχή ΠοιΥπ (ολοκληρωμένες υπηρεσίες με χρήση του RSVP) εμφάνισε αρκετά μειονεκτήματα, με κυριότερο αυτό της μη επεκτασιμότητας σε μεγάλα δίκτυα. Έτσι το μοντέλο αυτό δεν κατέστη κατάλληλο για εφαρμογή στο διαδίκτυο. Αποτέλεσμα ήταν να σχηματιστεί η ομάδα εργασίας για το μοντέλο των Διαφοροποιημένων Υπηρεσιών (Differentiated Services ή DiffServ) το Φεβρουάριο του 98 και έτσι το Δεκέμβριο του ίδιου χρόνου ήταν έτοιμο το τελικό κείμενο [Diffserv]. Ο

σκοπός της ομάδας εργασίας, ήταν αρχικά να ορίσει το πεδίο διαφοροποιημένων υπηρεσιών (DiffServ Field ή DS) στην επικεφαλίδα των IP πακέτων, αντικαθιστώντας το πεδίο τύπου υπηρεσίας (Type of Service, TOS) στο IPv4 ή το πεδίο κλάσης κίνησης (Traffic Class) στο IPv6. Αυτή τη στιγμή έχουν γίνει οι απαραίτητοι ορισμοί [Nichols98] του DS πεδίου και της αρχιτεκτονικής των διαφοροποιημένων υπηρεσιών, ωστόσο δεν έχουν ακόμη τυποποιηθεί οι διάφορες υπηρεσίες που μπορεί να προσφέρει το μοντέλο. Μελετώνται διάφορες προτάσεις (οι κυριότερες θα παρουσιαστούν στη συνέχεια), από τις οποίες θα προταθούν για προτυποποίηση αυτές που θα γνωρίσουν μεγάλη επιτυχία και εξάπλωση στην κοινότητα του διαδικτύου.

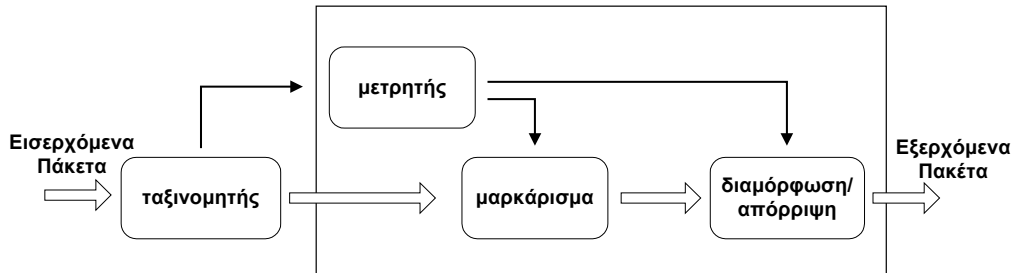
Οι Διαφοροποιημένες Υπηρεσίες (ΔΥ) χαρακτηρίζονται από το γεγονός ότι παρέχονται προς μια κατεύθυνση, είναι δηλαδή μονοκατευθυντήριες (uni-directional) και άρα ασυμμετρικές. Η αρχιτεκτονική ΔΥ μπορεί να χρησιμοποιηθεί μόνο για απλή μετάδοση και το μοντέλο δε μπορεί ακόμα να υποστηρίξει πολλαπλή μετάδοση. Η αρχιτεκτονική αυτή προσεγγίζει το πρόβλημα της επεκτασιμότητας χωρίζοντας την κυκλοφορία σε ένα αριθμό από ροές (aggregations), καθεμία με διαφορετική απαίτηση ΠοιΥπ. Δεν επιχειρεί να εγγυηθεί ένα επίπεδο υπηρεσίας (όπως το μοντέλο των ολοκληρωμένων υπηρεσιών) αλλά περισσότερο επιχειρεί μια σχετική διάταξη των ροών, έτσι ώστε μια συνολική ροή να τυχαίνει καλύτερης ή χειρότερης μεταχείρισης σε σχέση με άλλες.

Περίληπτικά η λειτουργία του μοντέλου έχει ως εξής: οι πελάτες αιτούνται ένα συγκεκριμένο επίπεδο υπηρεσίας μαρκάροντας το DS πεδίο του κάθε πακέτου με μια συγκεκριμένη τιμή. Η τιμή αυτή προσδιορίζει την ανά κόμβο συμπεριφορά του δικτύου (Per-Hop Behavior, PHB) ως προς το πακέτο. Οι τιμές του DS πεδίου είναι μέσα στα πλαίσια της συμφωνίας ανάμεσα στον παροχό και στον πελάτη (Service Level Agreement, SLA) και ορίζουν τις παραμέτρους του επιπέδου υπηρεσίας, όπως ο ρυθμός μετάδοσης, η προτεραιότητα μετάδοσης και απόρριψης, η εξυπηρέτηση στην ουρά κ.α. Στην συνέχεια παρουσιάζεται αναλυτικά η λειτουργία του μοντέλου ΔΥ.

2.5.2.1 Στοιχεία της αρχιτεκτονικής διαφοροποιημένων υπηρεσιών

Πριν η κυκλοφορία ενσωματωθεί σε μια συνολική ροή, τα πακέτα που ανήκουν σε αυτή τη ροή πρέπει να αναγνωριστούν. Για να προστατευθούν οι εγγυήσεις υπηρεσιών για κάθε συνολική ροή, τίθενται κάποιοι περιορισμοί στο ποσό της κυκλοφορίας που ο

χρήστης μπορεί να εισάγει στο δίκτυο. Τα στοιχεία της αρχιτεκτονικής ΔΥ, που έχουν την ευθύνη για αυτές τις λειτουργίες καλούνται ταξινόμηση και έλεγχος. Επίσης ένας δρομολογητής πρέπει να είναι ικανός να μετράει, να διαχειρίζεται και να απορρίπτει πακέτα μιας ροής. Η αρχιτεκτονική ΔΥ [Blake98] παρουσιάζεται στο σχήμα 2.4.



Σχήμα 2.4: Αρχιτεκτονική διαφοροποιημένων υπηρεσιών

Ταξινομητής

Ένας ταξινομητής εξετάζει κάθε εισερχόμενο πακέτο και προσδιορίζει σε ποια ροή ανήκει. Τα στοιχεία που λαμβάνονται υπόψη κατά τον προσδιορισμό είναι η IP διεύθυνση πηγής, η IP διεύθυνση προορισμού καθώς και οι αριθμοί θυρών της πηγής και του προορισμού. Ο ταξινομητής είναι επίσης σε θέση να προσδιορίσει την συνολική ροή στην οποία το πακέτο ανήκει, με βάση την τιμή του DS πεδίο του.

Μετρητής

Αφού μια ροή ταξινομηθεί, πρέπει να μετρηθεί η κατανάλωση πόρων από αυτήν. Για να προσδιοριστεί ότι η ροή δεν υπερβαίνει τα προσυμφωνημένα όρια κατανάλωσης πόρων, αρχικά πρέπει ο μετρητής να μετρήσει την ένταση της ροής για κάποια χρονική περίοδο. Ωστόσο, μόνο αυτό δεν είναι αρκετό για να διασφαλιστεί ότι η ροή συμμορφώνεται με τους όρους του συμβολαίου, αφού ο πελάτης μπορεί να στέλνει δεδομένα με ρυθμούς υψηλότερους από τον επιτρεπόμενο σε συγκεκριμένα χρονικά διαστήματα. (δηλαδή καταιγισμούς πακέτων-bursts). Έτσι, μια τυπική συμφωνία ΠοιΥπ θέτει περιορισμούς τόσο για το μέγεθος των καταιγισμών όσο και για το μέγιστο εύρος ζώνης μιας ροής. Συνεπώς, ο δρομολογητής πρέπει να μετράει το ρυθμό μετάδοσης της ροής όσο και το μέγεθος των καταιγισμών κυκλοφορίας. Η μέτρηση της κυκλοφορίας είναι επίσης σημαντική για τη χρέωση των χρηστών, ανάλογα με τις διάφορες υπηρεσίες που χρησιμοποιούν. Όταν διαφορετικές υπηρεσίες προσφέρονται πάνω από την ίδια φυσική σύνδεση, η ανάγκη για ανεξάρτητη χρέωση καθίσταται σημαντική. Σε μερικές

περιπτώσεις λαμβάνεται υπόψη στη χρέωση και ο προορισμός της κυκλοφορίας, αν οι αποστάσεις είναι πολύ μεγάλες.

Μαρκάρισμα

Η λειτουργία του μαρκαρίσματος είναι αυτή που θέτει την τιμή στο DS πεδίο του πακέτου. Το στοιχείο που εκτελεί αυτή την λειτουργία μπορεί να είναι προγραμματισμένο να μαρκάρει όλα τα πακέτα σε μια καθορισμένη τιμή ή σε διάφορες τιμές που αντιστοιχούν σε διαφορετικές ροές, ανάλογα με την κατάσταση του μετρητή. Συνήθως τα πακέτα μαρκάρονται σε τρεις διαφορετικές κατηγορίες (Three Color Marker) [Heinanen99]. Σε κάθε κατηγορία χαρακτηρίζεται από ένα χρώμα (πράσινο, κίτρινο και κόκκινο). Το μαρκάρισμα των πακέτων γίνεται με βάση τρεις παραμέτρους της κίνησης: Το Committed Information Rate (CIR), το Committed Burst Size (CBS), και το Excess Burst Size (EBS). Ένα πακέτο μαρκάρεται πράσινο αν η κίνηση δεν ξεπερνά το όριο του CBS, κίτρινο αν ξεπερνά το όριο CBS αλλά όχι το όριο του EBS και κόκκινη αν ξεπερνά και το όριο του EBS.

Διαμόρφωση

Όταν μια ροή περιέχει ένα καταιγισμό πακέτων, ένας δρομολογητής μπορεί να επιλέξει να επεξεργαστεί τα πακέτα με ένα πλήθος από τρόπους. Μια λύση είναι να τα επεξεργαστεί κανονικά αν βρίσκονται μέσα στα προκαθορισμένα όρια. Μια άλλη εναλλακτική λύση είναι να απορροφήσει τα πακέτα και να καθορίσει το ρυθμό μετάδοσής τους για μια μεγάλη χρονική περίοδο. Έτσι μπορεί να εξομαλύνει τη ροή ή ακόμα και να την εξαφανίσει. Η τελευταία λύση είναι να απορρίψει τα πακέτα που υπερβαίνουν ένα συγκεκριμένο κατώφλι. Αυτό το κατώφλι μπορεί να είναι ένα άνω όριο του αριθμού των πακέτων καταιγισμού, που ο δρομολογητής μπορεί να διατηρήσει σε οποιαδήποτε στιγμή. Ένα άλλο κατώφλι μπορεί να είναι το ποσό του χρόνου που τα πακέτα πρέπει να διατηρηθούν, χωρίς το περιεχόμενό τους να χάσει το ενδιαφέρον του.

Απόρριψη

Όταν μια ροή υπερβαίνει το συμφωνημένο ρυθμό ή ένας καταιγισμός υπερβαίνει ένα μέγιστο κατώφλι, ο δρομολογητής μπορεί να επιλέξει να απορρίψει ένα ή περισσότερα πακέτα της ροής. Όπως αναφέρθηκε, η απόρριψη πακέτων είναι μια κοινή πρακτική για

την αντιμετώπιση της συμφόρησης. Ανάλογα με την υπηρεσία, τα κριτήρια απόρριψης μπορεί να ποικίλουν. Όταν λαμβάνεται υπόψη και η ΠοιΥπ τότε η απόφαση για το αν ένα πακέτο πρέπει να απορριφθεί γίνεται αρκετά πολύπλοκη.

2.5.2.2 Διάκριση ανάμεσα στη λειτουργία των δρομολογητών κορμού και άκρων

Μια από τις κύριες αρχές της αρχιτεκτονικής ΔΥ, είναι η διάκριση ανάμεσα στον κορμό και στα άκρα ενός διαχειριστικού πεδίου (με τον όρο διαχειριστικό πεδίο εννοούμε ένα σύνολο από δρομολογητές και χρηστές που βρίσκονται υπό τον έλεγχο της ίδιας διαχειριστικής αρχής-Domain). Σε αντίθεση με την αρχιτεκτονική ολοκληρωμένων υπηρεσιών, που εκτελεί τις ίδιες διεργασίες σε κάθε δρομολογητή του μονοπατιού, η αρχιτεκτονική ΔΥ ωθεί τις περισσότερες λειτουργίες ταξινόμησης και ελέγχου στα άκρα του δικτύου και απλοποιεί τις λειτουργίες στον κορμό. Όταν ένας δρομολογητής στα άκρα λαμβάνει ένα πακέτο από έναν χρήστη ή από ένα γειτονικό διαχειριστικό πεδίο, συνήθως το ελέγχει και το ταξινομεί λεπτομερώς. Αντίθετα, στον κορμό του δικτύου δεν υπάρχει έλεγχος και η διαδικασία της ταξινόμησης περιορίζεται στην εξέταση του DS πεδίου για να αποφασιστεί η κατάλληλη ενέργεια προώθησης.

Το σύνολο των συσκευών που αποτελούν τα άκρα και τον κορμό του δικτύου συνήθως αναφέρεται ως πεδίο ΔΥ. Το πεδίο αυτό είναι ένα υποσύνολο του διαχειριστικού πεδίου. Αυτό οφείλεται στους εξής λόγους: Πρώτον, σε πολλές περιπτώσεις πολλά στοιχεία του δικτύου δεν έχουν την ικανότητα παροχής ΔΥ. Δεύτερον, αρκετοί χρηστές μπορεί να μην εμπνέουν εμπιστοσύνη όσον αφορά το μαρκάρισμα των πακέτων και έτσι δε μπορούν να θεωρηθούν μέρος του πεδίου.

Αν ο χρήστης δεν είναι αξιόπιστος, ο πρώτος δρομολογητής στο άκρο του πεδίου ΔΥ εκτελεί τις λειτουργίες ταξινόμησης και ελέγχου, που αφορούν όλη την κυκλοφορία που στέλνει ο χρήστης. Έτσι ο δρομολογητής θα υποδείξει σε ποια συνολική ροή ανήκει κάθε πακέτο θέτοντας την ανάλογη τιμή στο πεδίο DS.

Με τη χρήση του DS πεδίου, κάθε δρομολογητής κορμού μπορεί να μεταχειριστεί κάθε πακέτο που ανήκει σε μια συγκεκριμένη συνολική ροή με τον ίδιο τρόπο. Για παράδειγμα, ένα πακέτο μπορεί να μαρκαριστεί με τρόπο που να δείχνει ότι πρέπει να προωθηθεί όσο το δυνατό γρηγορότερα, διότι ανήκει σε μια εφαρμογή πραγματικού χρόνου. Μαρκάροντας το πακέτο με αυτό τον τρόπο κάθε δρομολογητής γνωρίζει ότι θα

πρέπει να προσπαθήσει να προωθήσει το πακέτο πριν να προωθήσει πακέτα που δεν είναι ευαίσθητα στην καθυστέρηση.

Ένα από τα οφέλη της αρχιτεκτονικής ΔΥ είναι ότι κάθε δρομολογητής κορμού βασίζεται αποκλειστικά στο DS πεδίο για να αποφασίσει τους κανόνες προώθησης ενός πακέτου. Ο δρομολογητής μπορεί να επεξεργαστεί πολλά πακέτα από διαφορετικές ροές με τον ίδιο τρόπο, όταν το DS πεδίο έχει την ίδια τιμή σε κάθε πακέτο. Με την ομαδοποίηση πακέτων με βάση την τιμή του DS πεδίου, η αρχιτεκτονική ΔΥ μπορεί να δημιουργήσει συνολικές ροές που είναι εύκολο να προσδιοριστούν.

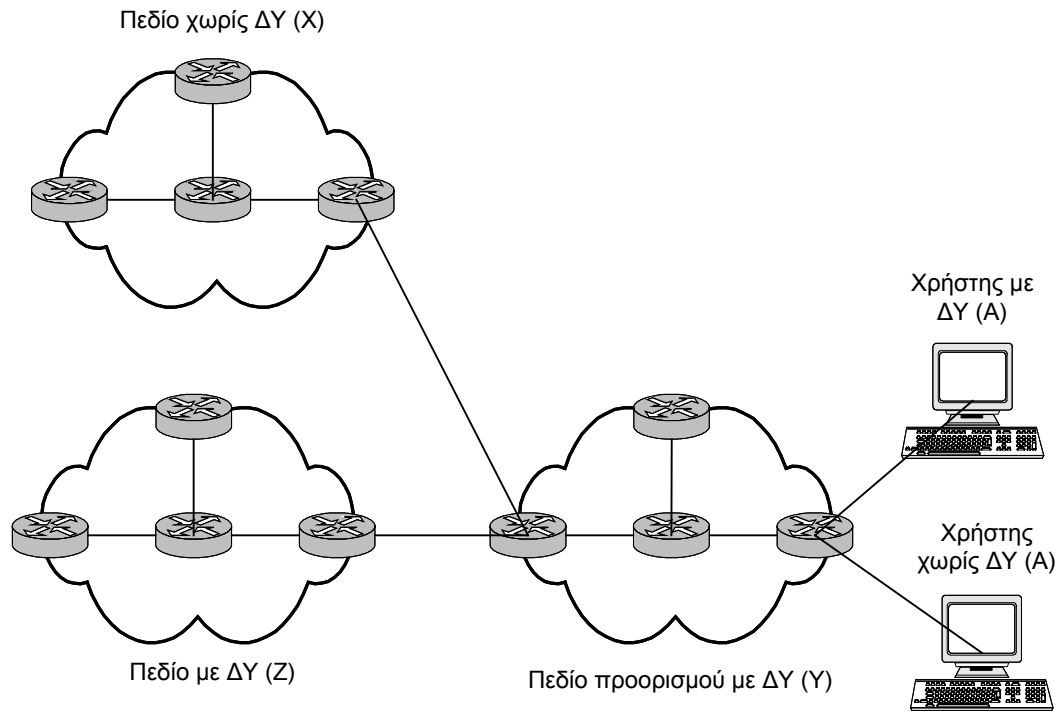
Για να υπάρχει κάποιο σημαντικό όφελος, πρέπει να τεθούν κάποιοι περιορισμοί στην ποσότητα κυκλοφορίας κάθε συνολικής ροής. Στην περίπτωση παραδείγματος χάριν, που όλοι οι χρήστες μαρκάρουν τα πακέτα τους ώστε να εξυπηρετηθούν κατά τον καλύτερο δυνατό τρόπο, με μικρή καθυστέρηση και μικρές απώλειες, αυτό έχει ως αποτέλεσμα όλες οι συνολικές ροές να εξυπηρετούνταν με την υπηρεσία καλύτερης προσπάθειας. Επομένως είναι αναγκαία η επιβεβαίωση από τον δρομολογητή στο άκρο του δικτύου, ότι ο χρήστης έχει το δικαίωμα να χρησιμοποιήσει το συγκεκριμένο μαρκάρισμα και ότι η εφαρμογή δεν υπερβαίνει το εύρος ζώνης που σχετίζεται με αυτό. Ο δρομολογητής είναι δυνατό να ξαναμαρκάρει το πακέτο, με βάση τις πολιτικές που ορίζονται από το διαχειριστή του δικτύου. Ωστόσο, είναι προτιμότερο το μαρκάρισμα των πακέτων να γίνεται από τον χρήστη, αφού αυτός γνωρίζει καλύτερα τις απαιτήσεις σε ΠοιΥπ της εφαρμογής του.

2.5.2.3 Πολλαπλά διαχειριστικά πεδία

Μέχρι τώρα είδαμε πως τα πακέτα μαρκάρονται και επεξεργάζονται μέσα στα όρια ενός μόνο διαχειριστικού πεδίου. Το διαδίκτυο όμως αποτελείται από πολλά διαχειριστικά πεδία. Η λειτουργία της αρχιτεκτονικής ΔΥ, όταν ένα πακέτο διέρχεται μέσα από πολλά διαχειριστικά πεδία, παρουσιάζονται στην εικόνα 2.1.

Όταν ένα πακέτο φθάνει από ένα χρήστη (Α) χωρίς να του έχει ανατεθεί μια τιμή στο DS πεδίο, ο δρομολογητής στο άκρο του πεδίου που υποστηρίζει ΔΥ (Υ), επιλέγει να εντάξει το πακέτο σε κάποια συνολική ροή με βάση τους διαμορφωμένους κανόνες και πολιτικές. Επίσης, όταν ένα πακέτο από ένα γειτονικό πεδίο (Χ), το οποίο δεν

υποστηρίζει ΔΥ, φτάνει στον δρομολογητή άκρου του πεδίου Y, αυτός ο δρομολογητής μπορεί με τον ίδιο τρόπο να αναθέσει μια τιμή στο DS πεδίο του πακέτου.



Εικόνα 2.1: Πολλαπλά διαχειριστικά πεδία

Με τον όρο προφίλ κίνησης (traffic profile) στην αρχιτεκτονική ΔΥ εννοούμε μια περιγραφή των ιδιοτήτων μιας ροής κυκλοφορίας, όπως το ρυθμό μετάδοσης και το μέγεθος καταιγισμού. Στην περίπτωση που ένα πακέτο φτάνει από ένα χρήστη (B) που υποστηρίζει ΔΥ, επιβλέπεται με βάση το προφίλ που αντιστοιχεί σε αυτόν τον χρήστη. Ομοίως, αν ένα πακέτο φθάνει από ένα πεδίο (Z) που υποστηρίζει ΔΥ επιβλέπεται από το πεδίο προορισμού (Y) ως προς το προφίλ που σχετίζεται με το γειτονικό πεδίο (Z). Αν η κυκλοφορία ικανοποιεί το προφίλ, τα πακέτα μπορούν να προχωρήσουν στον κορμό του πεδίου προορισμού (Y). Αν ένα πακέτο δεν ικανοποιεί το προφίλ, ο δρομολογητής στο άκρο του (Y) μπορεί να τροποποιήσει, δηλαδή να ξαναμαρκάρει, την τιμή του DS πεδίου του πακέτου ή μπορεί να απορρίψει το πακέτο. Είναι επίσης δυνατό ο δρομολογητής αυτός να επιτρέψει στα πακέτα να περάσουν, αλλά να χρεώσει επιπλέον τον χρήστη (B) ή το διαχειριστικό πεδίο (Z). Συνήθως είναι πιο συνετό για τον χρήστη (B) ή το πεδίο (Z) να διαχειριστούν, να επαναμαρκάρουν ή να απορρίψουν από μόνοι τους πακέτα που είναι έξω από τα όρια του προφίλ κίνησης. Αυτό είναι πιο φυσικό, αφού

οι αποστολές της κυκλοφορίας μπορούν να διακρίνουν πιο καλά τις σημαντικές από τις λιγότερο σημαντικές ροές, από ότι ο δρομολογητής στο άκρο του δικτύου.

2.5.2.4 Συμβόλαια Υπηρεσιών

Στην προηγούμενη παράγραφο αναφέραμε ότι οι λειτουργίες ενός διαχειριστικού πεδίου καθορίζονται από προδιαμορφωμένους κανόνες και πολιτικές. Οι κανόνες και οι πολιτικές ορίζονται από ένα συμβόλαιο υπηρεσιών (Service Level Agreement-SLA), ανάμεσα στον πελάτη και στον προμηθευτή υπηρεσιών διαδικτύου (Internet Service Provider-ISP) και καθορίζουν το επίπεδο υπηρεσίας που θα παρέχεται στον πελάτη. Όταν λέμε “πελάτης” μπορεί να εννοούμε ένα οργανισμό χρηστών ή ακόμη και ένα άλλο διαχειριστικό πεδίο που υποστηρίζει ΔΥ. Ένα SLA μπορεί να περιλαμβάνει κανόνες ρύθμισης της κυκλοφορίας, που συνιστούν με τη σειρά τους μια συμφωνία ρύθμισης κυκλοφορίας (Traffic Conditioning Agreement-TCA). Το TCA προσδιορίζει λεπτομερείς παραμέτρους υπηρεσιών για κάθε επίπεδο υπηρεσίας, όπως:

1. Λεπτομερείς παραμέτρους απόδοσης υπηρεσίας, όπως ο αναμενόμενος ρυθμός μετάδοσης, η πιθανότητα απόρριψης και η καθυστέρηση.
2. Περιορισμούς στα σημεία εισόδου και εξόδου όπου η υπηρεσία παρέχεται, καθορίζοντας την εμβέλεια της υπηρεσίας.
3. Προφίλ κίνησης στα οποία πρέπει να επιμείνει η παρεχόμενη υπηρεσία, όπως παράμετροι του κάδου με κουπόνι (token-bucket).
4. Ρύθμιση της κυκλοφορίας που υπερβαίνει τα καθορισμένα προφίλ.
5. Παρεχόμενες υπηρεσίες μαρκαρίσματος.
6. Παρεχόμενες υπηρεσίες διαμόρφωσης της κίνησης.

Επιπρόσθετα, το SLA μπορεί να καθορίζει πιο γενικά χαρακτηριστικά υπηρεσιών, όπως:

1. Διαθεσιμότητα/Αξιοπιστία, που μπορεί να περιλαμβάνουν και τη συμπεριφορά στην περίπτωση όπου αποτυχίες έχουν σαν αποτέλεσμα την επαναδρομολόγηση της κίνησης.
2. Υπηρεσίες κρυπτογράφησης.
3. Περιορισμούς στη δρομολόγηση.
4. Μηχανισμούς πιστοποίησης.
5. Μηχανισμούς για την επίβλεψη και τον λογιστικό έλεγχο της υπηρεσίας.

6. Ευθύνες για τη θέση και τη λειτουργικότητα του εξοπλισμού, ενέργειες όταν ένα συμβόλαιο καταπατείται, δυνατότητες υποστήριξης.
7. Μηχανισμούς χρέωσης.

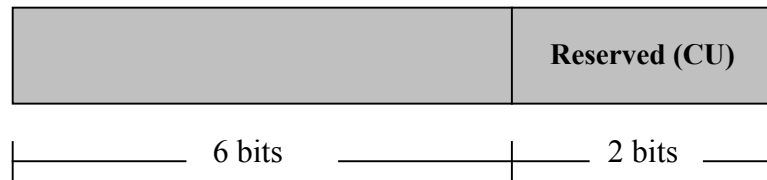
2.5.2.5 Το πεδίο DS και συμβατότητα με το πεδίο TOS

Ο αρχικός σκοπός της ομάδας εργασίας ΔΥ, ήταν να ορίσει το DS πεδίο στην επικεφαλίδα των IP πακέτων, αντικαθιστώντας το πεδίο Type of Service (TOS) [Almquist99] στην περίπτωση του πρωτοκόλλου IPv4, ή το πεδίο Traffic Class στην περίπτωση του πρωτοκόλλου IPv6. Θα επικεντρωθούμε στην περίπτωση του IPv4, τα ίδια περίπου όμως ισχύουν και για το IPv6.

Στην επικεφαλίδα κάθε πακέτου του IPv4, είχε δεσμευθεί μια οκτάδα από bits που αποτελούσε το λεγόμενο TOS byte. Από αυτά τα bits το ένα ήταν αχρησιμοποίητο, τρία χρησιμοποιούνταν για τον ορισμό προτεραιότητας και τα υπόλοιπα τέσσερα αποτελούσαν το πεδίο Type of Service. Η χρήση του TOS byte είχε σκοπό να παρέχει δύο δυνατότητες: πρώτον να δίνει σε ορισμένα πακέτα μεγαλύτερη προτεραιότητα σχετικά με κάποια άλλα, και δεύτερον να παρέχει στους δρομολογητές την κατάλληλη πληροφορία για τον τύπο της κυκλοφορίας. Έτσι οι δρομολογητές θα μπορούσαν να υπολογίζουν διαφορετικά μονοπάτια για κάθε τιμή του Type of Service. Ωστόσο, το TOS byte δε βρήκε μεγάλη εφαρμογή και δε γνώρισε μεγάλη διάδοση. Ο κύριος λόγος ήταν ότι το TOS byte ορίστηκε από την αρχή χωρίς μεγάλη σαφήνεια με αποτέλεσμα να χρησιμοποιείται με διαφορετικό τρόπο και να παίρνει διάφορες διερμηνείες. Επομένως, δεν ήταν κατάλληλο για την παροχή υπηρεσιών από άκρο-σε-άκρο. Έτσι αποφασίστηκε να αναιρεθούν οι σημασίες που του είχαν δοθεί. Η ομάδα εργασίας των ΔΥ θα ορίσει νέες σημασίες για το πεδίο αυτό και θα το ονομάσει πεδίο DS.

Στην εικόνα 2.2 παρουσιάζεται η μορφή του πεδίου DS, όπως έχει οριστεί από την ομάδα εργασίας των ΔΥ. Τα πρώτα 6 bits του πεδίου αποτελούν το Differentiated Services Code Point (DSCP) και χρησιμοποιούνται για να δηλώσουν την PHB (εξηγούνται αναλυτικά στην επόμενη παράγραφο) που θα έχει το πακέτο και κατά συνέπεια την υπηρεσία που θα του παρέχεται. Συνολικά μπορούν να υλοποιηθούν στο ίδιο δίκτυο μέχρι και 64 (2^6) διαφορετικές υπηρεσίες. Ένα μέρος του πεδίου των 6-bit χρησιμοποιείται για τον καθορισμό καθολικών από άκρο σε άκρο συμπεριφορών και ένα

άλλο είναι δεσμευμένο για τοπική ή πειραματική χρήση. Αυτό το μέρος του πεδίου επιτρέπει κάποιους πειραματισμούς με νέες συμπεριφορές και υπηρεσίες, καθώς και χρήση για υπηρεσίες και συμπεριφορές που απευθύνονται σε μια συγκεκριμένη αγορά. Τα τελευταία 2 bits του πεδίου DS είναι δεσμευμένα για μελλοντική χρήση και αγνοούνται κατά τη διαδικασία της επιλογής PHB.



CU – Currently Unused

Εικόνα 2.2: To DS Field

Υπάρχει ένα πλήθος από δρομολογητές που δεν ανήκουν σε πεδία που υποστηρίζουν ΔΥ και συνεχίζουν να ρυθμίζουν τη λειτουργία τους με βάση την τιμή του TOS byte στην επικεφαλίδα των IP πακέτων. Η μετατροπή ενός πεδίου που δεν υποστηρίζει ΔΥ σε πεδίο που υποστηρίζει ΔΥ είναι μη πρακτική και αρκετά δύσκολη. Έτσι για την παροχή ολοκληρωμένων από άκρο σε άκρο υπηρεσιών πρέπει να υπάρχει κάποια συμβατότητα ανάμεσα στο DSCP πεδίο και στο TOS byte. Για το σκοπό αυτό τα 3 bits προτεραιότητας του TOS byte λαμβάνονται υπόψη και διερμηνεύονται στο DSCP πεδίο, από τον δρομολογητή στο άκρο του πεδίου που υποστηρίζει ΔΥ. Μπορούν λοιπόν να υποστηριχθούν από το DSCP πεδίο, $8(2^3)$ επίπεδα προτεραιότητας.

2.5.2.6 Ανά κόμβο συμπεριφορές

Η ανά κόμβο συμπεριφορά (Per-Hop Behavior-PHB), σύμφωνα με την ορολογία των ΔΥ ορίζεται ως “η εξωτερικά παρατηρούμενη συμπεριφορά ενός κόμβου του πεδίου ΔΥ σε μια συγκεκριμένη συνολική ροή”. Αυτή η εξωτερική συμπεριφορά αναφέρεται σε παραμέτρους όπως ο ρυθμός απώλειας, η καθυστέρηση, η απόκλιση καθυστέρησης, κ.α. Η PHB είναι το μέσο με το οποίο ένας κόμβος κατανέμει τους πόρους του στις διάφορες συνολικές ροές και με τη βοήθεια του μέσου αυτού μπορούν να δημιουργηθούν ΔΥ.

Η συμπεριφορά σε κάθε κόμβο καθορίζεται από την ερμηνεία των 6 bits του DSCP πεδίου ως σύνολο, ενώ τα 2 αχρησιμοποίητα bits δεν λαμβάνονται υπόψη κατά την

επιλογή PHB. Ένας πίνακας αντιστοίχισης τιμών του DSCP πεδίου (code points) μπορεί να περιέχει τόσο 1→1 όσο και N→1 αντιστοιχήσεις, δηλαδή οπωσδήποτε κάθε τιμή του πεδίου πρέπει να αντιστοιχεί σε μια PHB και πολλές διαφορετικές τιμές του πεδίου μπορούν να αντιστοιχούν στην ίδια PHB.

Πακέτα τα οποία έχουν μια μη αναγνωρίσιμη τιμή στο πεδίο DS, πρέπει να προωθούνται στο δίκτυο με βάση κάποια συμφωνημένη συμπεριφορά (π.χ. “καλύτερης προσπάθειας”) και οι DS τιμές των πακέτων δε πρέπει να αλλάζουν. Αυτή η συμφωνημένη PHB οφείλει να είναι διαθέσιμη σε κάθε κόμβο και η προτεινόμενη DS τιμή που αντιστοιχεί σε αυτή τη συμπεριφορά είναι το "000000".

Υπάρχουν ήδη προτυποποιημένες PHBs που αντιστοιχούν σε κάποιες συγκεκριμένες τιμές του πεδίου DS. Ωστόσο, υπάρχει χώρος για περαιτέρω αντιστοιχήσεις. Οι παροχείς ΔΥ μπορούν να βασίζονται στις προτυποποιημένες PHBs για τη δημιουργία των διαφόρων υπηρεσιών.

Οι PHBs μπορούν να προσδιοριστούν με βάση την προτεραιότητα στην κατανάλωση πόρων (π.χ. εύρος ζώνης) σε σχέση με άλλες PHBs, ή με βάση τα σχετικά χαρακτηριστικά της κυκλοφορίας (π.χ. καθυστέρηση, απώλειες). Πολλές PHBs μπορούν να προσδιοριστούν ως μια ομάδα PHB. Συνήθως οι PHBs που ανήκουν σε μια ομάδα έχουν ένα κοινό περιορισμό, που μπορεί να αναφέρεται στον χρονοπρογραμματισμό των πακέτων ή στην πολιτική διαχείρισης των στοιχείων προσωρινής αποθήκευσης (buffers).

Η προσπάθεια για παροχή από άκρο σε άκρο υπηρεσιών προϋποθέτει την υποστήριξη των υπηρεσιών κατά μήκος πολλών διαχειριστικών πεδίων. Στα όρια των πεδίων προκύπτουν πολλά προβλήματα συμβατότητας, όπως:

- Οι υπηρεσίες που παρέχονται από ένα πεδίο μπορεί να είναι ασύμβατες με τις υπηρεσίες ενός γειτονικού πεδίου.
- Οι υπηρεσίες των δύο πεδίων μπορεί να είναι συμβατές, αλλά οι PHBs που χρησιμοποιούνται για την παροχή της υπηρεσίας μπορεί να διαφέρουν.
- Η PHB μπορεί να είναι ίδια, αλλά τη τιμή του DS πεδίου που χρησιμοποιείται για την απαίτηση της PHB μπορεί να διαφέρει.

- Η PHB και οι DS τιμές μπορεί να είναι ίδια, αλλά τα διαφορετικά μοντέλα παροχής και χρέωσης να έχουν σαν αποτέλεσμα διαφορετική υπηρεσία.

Η επίλυση αυτών των προβλημάτων απαιτεί τον ορισμό συμβατών υπηρεσιών και τη διαπραγμάτευση των DS τιμών που θα χρησιμοποιούνται για την απαίτηση των υπηρεσιών. Η διαδικασία αυτή απλοποιείται σημαντικά με την παροχή ενός συνόλου καθολικών υπηρεσιών. Οι προτεινόμενες υπηρεσίες παρουσιάζονται στην συνέχεια.

2.5.2.7 Προτεινόμενες υπηρεσίες για προτυποποίηση

Οι ενέργειες που μπορούν να πραγματοποιηθούν στα πακέτα που φτάνουν σε ένα δρομολογητή είναι αρκετές. Μπορούν να προωθηθούν ή να απορριφθούν, να καθυστερήσουν και να μετρηθούν (για σκοπούς χρέωσης και εξακρίβωσης της τήρησης των όρων του συμβολαίου κυκλοφορίας). Αυτές οι ενέργειες μπορούν να γίνουν σε πολλούς συνδυασμούς και σε πολλά επίπεδα διαφοροποίησης. Πρόθεση της αρχιτεκτονικής ΔΥ είναι να επιτρέψει σε έναν αριθμό από αυτούς του συνδυασμούς να γίνουν προτυποποιημένες συμπεριφορές. Αφού στο εμπόριο αναπτυχθούν προϊόντα που να υποστηρίζουν αυτές τις συμπεριφορές, οι παροχείς υπηρεσιών μπορούν να τις χρησιμοποιήσουν για να δημιουργήσουν μοναδικές υπηρεσίες. Διαφορετικά διαχειριστικά πεδία μπορούν να συνδυάσουν τις ίδιες συμπεριφορές με διαφορετικό τρόπο, ώστε να δημιουργήσουν τη δικιά τους ξεχωριστή υπηρεσία που θα παρέχει το επιθυμητό επίπεδο ΠοιΥπ. Επίσης κάθε πεδίο μπορεί να έχει ξεχωριστές πολιτικές για το πόση κυκλοφορία επιτρέπεται σε κάθε συνολική ροή. Τέλος τα μοντέλα χρέωσης θα διαφέρουν από πεδίο σε πεδίο. Μέχρι τώρα έχουν οριστεί δυο τύποι υπηρεσιών και τα αντίστοιχα PHBs:

- Προνομιακή Υπηρεσία-Premium Service
- Εγγυημένη Υπηρεσία-Assured Service

Οι υπηρεσίες δικτύου διακρίνονται σε δυο είδη, ανάλογα με τα χαρακτηριστικά της υπηρεσίας που καθορίζουν τη μεταφορά ενός πακέτου σε ένα ή περισσότερα μονοπάτια μέσα στο δίκτυο. υπάρχουν λοιπόν οι **ποιοτικές υπηρεσίες** (qualitative), όπου τα παραπάνω χαρακτηριστικά μπορούν να εκφραστούν συγκριτικά, με την έννοια ότι κάποια υπηρεσία θα έχει μεγαλύτερη προτεραιότητα σε σχέση με κάποια άλλη, και οι

ποσοτικές υπηρεσίες (quantitative), όπου τα χαρακτηριστικά μπορούν να εκφραστούν ποσοτικά, όπως π.χ. καθυστέρηση μετάδοσης, απόκλιση καθυστέρησης, ρυθμός μετάδοσης δεδομένων, απώλεια.

Προνομιακή Υπηρεσία

Πρόκειται για μια ποσοτική υπηρεσία που προσομοιώνει την παραδοσιακή υπηρεσία μισθωμένης γραμμής. Έτσι, εγγυάται την μετάδοση της κυκλοφορίας του χρήστη με ένα μέγιστο εύρος ζώνης, αμελητέα καθυστέρηση και πολύ μικρή πιθανότητα απόρριψης, μέχρι ένα προσυμφωνημένο ρυθμό. Πάνω από αυτό το ρυθμό η κυκλοφορία απορρίπτεται. Η υπηρεσία αυτή τυπικά προσφέρεται ανάμεσα σε δύο συγκεκριμένα σημεία και είναι κατάλληλη για πολλές εφαρμογές. Εξαιτίας όμως των υψηλών ποιοτικών εγγυήσεων, συνήθως χρεώνεται παραπάνω από εναλλακτικές υπηρεσίες και επομένως είναι συνετό να χρησιμοποιείται μόνο σε εφαρμογές που έχουν πραγματικά ανάγκη από αυτό τον τύπο υπηρεσίας. Έχει προταθεί η υλοποίηση της υπηρεσίας αυτής με χρήση μιας κατάλληλης PHB, της Expedited Forwarding (EF) PHB [Jacobson99]. Αυτό βέβαια δεν αποκλείει τη χρήση της EF PHB και για τη δημιουργία άλλων υπηρεσιών. Η συγκεκριμένη όμως PHB είναι κατάλληλη για την προνομιακή υπηρεσία, αφού μπορεί να χρησιμοποιηθεί για τη δημιουργία μιας υπηρεσίας με μικρές απώλειες, μικρή καθυστέρηση, χαμηλή απόκλιση καθυστέρησης και εγγυημένο εύρος ζώνης. Η EF PHB ορίζεται ως η συμπεριφορά προώθησης για μια συνολική ροή όπου ο ρυθμός αναχώρησης των πακέτων από οποιοδήποτε κόμβο, πρέπει να είναι τουλάχιστον ίσος ή να υπερβαίνει μια προσυμφωνημένη τιμή. Η EF κυκλοφορία θα πρέπει να λαμβάνει αυτό το ρυθμό ανεξάρτητα από την ένταση οποιασδήποτε άλλης κυκλοφορίας που προσπαθεί να διέλθει από τον κόμβο. Έτσι η EF PHB επιτρέπει σε εφαρμογές ευαίσθητες στην καθυστέρηση να λειτουργούν κανονικά ακόμα και όταν το δίκτυο βρίσκεται σε κατάσταση υψηλού φόρτου.

Εγγυημένη Υπηρεσία

Η εγγυημένη υπηρεσία προσφέρει χαλαρότερες εγγυήσεις από την προνομιακή υπηρεσία, αλλά θεωρείται κι αυτή ποσοτική υπηρεσία. Υπόσχεται τη μετάδοση κυκλοφορίας με ένα υψηλό βαθμό αξιοπιστίας και με μεταβλητή αλλά φραγμένη καθυστέρηση, μέχρι ένα συγκεκριμένο ρυθμό. Πάνω από αυτό το ρυθμό η κυκλοφορία

καθυστερεί σημαντικά ή απορρίπτεται. Η ανά κόμβο συμπεριφορά που χρησιμοποιείται για την υλοποίηση αυτής της υπηρεσίας καλείται Assured Forwarding (AF) [Heinanen99]. Σύμφωνα με τη συμπεριφορά αυτή, δίνεται στο χρήστη το δικαίωμα χρήσης ενός ελάχιστου εύρους ζώνης. Ωστόσο, ο χρήστης μπορεί να στέλνει επιπρόσθετη κυκλοφορία με ρυθμό που υπερβαίνει το ελάχιστο εύρος ζώνης. Τα πακέτα κάτω από αυτό το ελάχιστο κατώφλι εύρους ζώνης μαρκάρονται ως “εντός προφίλ” (in profile), ενώ τα πακέτα που υπερβαίνουν αυτό το κατώφλι μαρκάρονται ως “εκτός προφίλ” (out of profile). Σε περιόδους συμφόρησης, τα πακέτα της δεύτερης κατηγορίας έχουν μεγαλύτερη προτεραιότητα απόρριψης. Αυτός είναι και ο σκοπός αυτής της συμπεριφοράς, δηλαδή να διευκρινίσει μόνο πότε ένα πακέτο μπορεί να απορριφθεί και όχι να δώσει στο πακέτο μεγαλύτερη προτεραιότητα, όσον αφορά την προώθηση, σε σχέση με κάποιο άλλο. Η εγγυημένη υπηρεσία, είναι κατάλληλη για πολλές εφαρμογές. Η χρέωσή της είναι χαμηλότερη από αυτή της προνομιακής υπηρεσίας, εξαιτίας της μεταβλητής καθυστέρησης. Ωστόσο, λόγω του ορίου στη καθυστέρηση και του υψηλού βαθμού αξιοπιστίας χρεώνεται ακριβότερα από άλλες υπηρεσίες. Είναι εξαιρετικά κατάλληλη για εφαρμογές αναπαραγωγής video και ήχου, όπου ένα αρκετά μεγάλο εύρος ζώνης απαιτείται σε συνεχή βάση, αλλά η μη αλληλεπιδραστική φύση της κυκλοφορίας την καθιστά κατά κάποιο τρόπο ανεκτική στην καθυστέρηση.

2.5.2.8 Χρονοπρογραμματιστές (schedulers) κατάλληλοι για την παροχή ΔΥ

Ιστορικά, τα δίκτυα ήταν σχεδιασμένα να εξυπηρετούν προκαθορισμένα είδη κίνησης. Παραδείγματος χάριν, το τηλεφωνικό δίκτυο είναι σχεδιασμένο να μεταφέρει φωνή και το διαδίκτυο να μεταφέρει δεδομένα. Για την παροχή ΠοιΥπ όμως, ένα δίκτυο θα πρέπει να είναι σε θέση να προσφέρει ένα συγκεκριμένο εύρος ζώνης και να εγγυηθεί μια μέγιστη καθυστέρηση για κάθε σύνδεση. Αυτές οι δυο παράμετροι συνήθως δεν συσχετίζονται. Για παράδειγμα μια σύνδεση μεταφοράς δεδομένων video απαιτεί ένα συγκεκριμένο εύρος ζώνης, αλλά δεν θέτει αυστηρούς περιορισμούς στην καθυστέρηση μετάδοσης. Για την παροχή ΠοιΥπ τα δίκτυα χρησιμοποιούν μια σειρά από χρονοπρογραμματιστές οι οποίοι διαρθρώνονται κατάλληλα, ώστε να προωθούν την κίνηση παρέχοντας διαφορετική ποιότητα υπηρεσία σε διαφορετικές ροές κίνησης. Οι χρονοπρογραμματιστές που συνήθως χρησιμοποιούνται είναι οι παρακάτω.

- Priority Queuing-PQ (Εναπόθεση σε ουρά βάσει προτεραιοτήτων).
- Fair schedulers (Δίκαιος χρονοπρογραμματισμός).
- Class Based Queuing-CBQ (Εναπόθεση σε ουρά βάσει διαφορετικών κλάσεων κίνησης).

Στη συνέχεια παρουσιάζουμε τον τρόπο λειτουργίας των παραπάνω χρονοπρογραμματιστών δίνοντας ιδιαίτερη έμφαση στον CBQ, λόγο της χρήσης του στα πλαίσια της διατριβής.

Ο PQ είναι ο πλέον κατάλληλος χρονοπρογραμματιστής για την παροχή υπηρεσιών με έμφαση στην μικρή καθυστέρηση μετάδοσης. Βεβαιώνει ότι η πιο σημαντική κίνηση θα έχει τη γρηγορότερη μεταχείριση σε κάθε σημείο όπου χρησιμοποιείται. Είναι σχεδιασμένος για να δίνει αυστηρή προτεραιότητα στην πιο σημαντική κίνηση. Η κίνηση εισχωρεί σε μία από τέσσερις ουρές: υψηλής, μεσαίας, κανονικής ή χαμηλής προτεραιότητας. Όταν ο δρομολογητής είναι έτοιμος να στείλει ένα πακέτο, ελέγχει την ουρά υψηλής προτεραιότητας. Αν υπάρχει εκεί κάποιο πακέτο, το μεταδίδει. Αν όχι, ελέγχεται η μεσαίας προτεραιότητας ουρά και αν εκεί υπάρχει πακέτο το μεταδίδει. Αν όχι, ελέγχεται η ουρά κανονικής προτεραιότητας, και τέλος η χαμηλής προτεραιότητας. Η διαδικασία αυτή επαναλαμβάνεται για το επόμενο πακέτο. Αν η κίνηση στην ουρά υψηλότερης προτεραιότητας είναι μεγάλη, τότε οι άλλες ουρές αδρανούν. Ο αλγόριθμος παρέχει στις ουρές υψηλής προτεραιότητας ιδιαίτερη μεταχείριση έναντι των ουρών χαμηλής προτεραιότητας. Έτσι όμως οι χρονοπρογραμματιστές αυτοί δεν είναι σε θέση να εγγυηθούν ότι ένα ελάχιστο μέρος του εύρους ζώνης θα δοθεί στην κίνηση χαμηλής προτεραιότητας.

Οι “δίκαιοι χρονοπρογραμματιστές” [Nagle87] όπως ο Weighted Fair Queuing (WFQ) [Demers89] και ο Weighted Round Robin (WRR), χρησιμοποιούνται για να παράσχουν εγγυημένο εύρος ζώνης σε διαφορετικές ροές δεδομένων. Οι χρονοπρογραμματιστές αυτοί παρέχουν ένα συγκεκριμένο εύρος ζώνης σε κάθε σύνδεση, διατηρώντας ξεχωριστές ουρές για πακέτα από κάθε μία μεμονωμένη πηγή. Οι ουρές εξυπηρετούνται με έναν round-robin τρόπο, έτσι αποτρέπεται η αυθαίρετη αύξηση του εύρους ζώνης μίας πηγής, ή της καθυστέρησης άλλων πηγών. Ο WFQ μπορεί επίσης να παρέχει εγγυήσεις για το μέγιστη καθυστέρηση μετάδοσης [Parekh93, Parekh94] και φαινομενικά είναι η

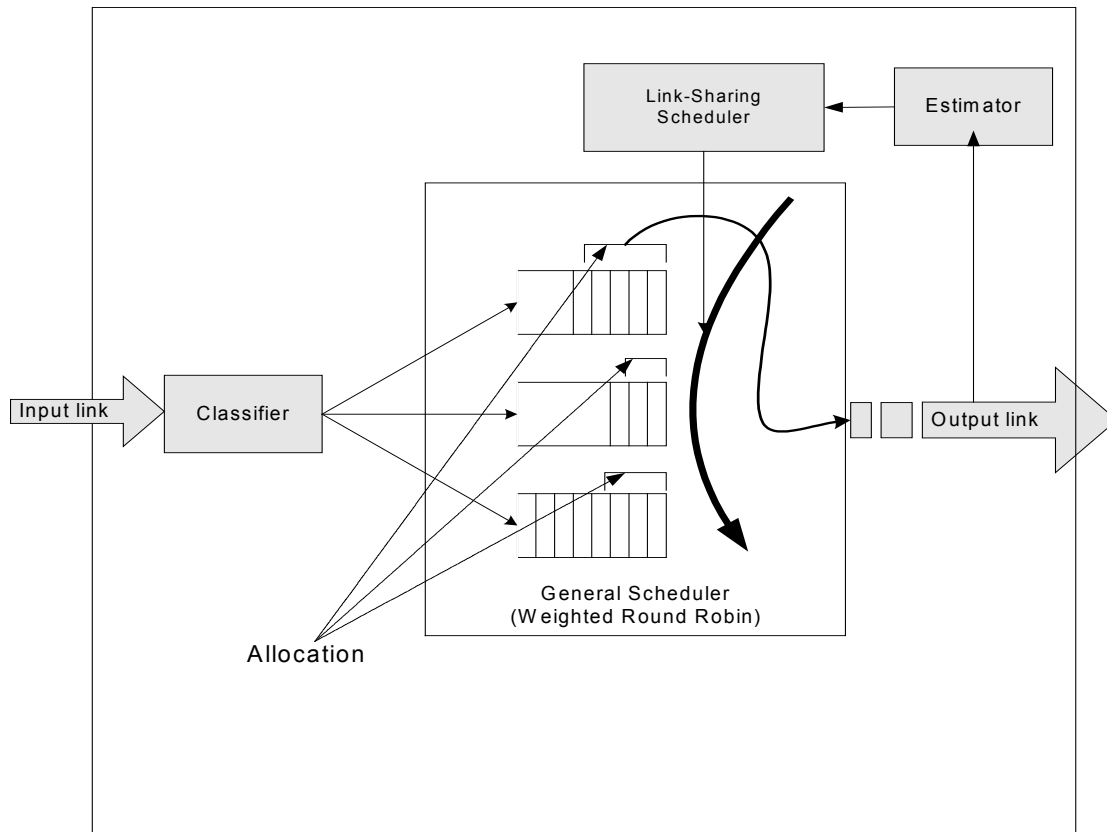
καλύτερη λύση για την παροχή εγγυήσεων τόσο για το εύρος ζώνης όσο και για την καθυστέρηση μετάδοσης. Όμως η καθυστέρηση μετάδοσης ελέγχεται μεταβάλλοντας το εύρος ζώνης που έχει οριστεί για μια σύνδεση. Επειδή η καθυστέρηση και το εύρος ζώνης δεν μπορούν να τροποποιηθούν ανεξάρτητα, το δίκτυο δεν είναι σε θέση να αποτρέψει μια σύνδεση μικρής καθυστέρησης από τα στείλει και ένα μεγάλο πλήθος δεδομένων, καταλαμβάνοντας μέρος από το εύρος ζώνης του δικτύου.

Οι PQ και οι δίκαιοι χρονοπρογραμματιστές δεν μπορούν να ικανοποιήσουν τις σύγχρονες απαιτήσεις για εγγυήσεις στην κατανομή του εύρους ζώνης και στην καθυστέρηση μετάδοσης. Ο χρονοπρογραμματιστής CBQ δημιουργήθηκε για να ικανοποιήσει ακριβώς αυτές τις απαιτήσεις. Στηρίζεται σε μια σειρά μηχανισμούς οι οποίοι συνδυάζουν τις ικανότητες των PQ και των δικαίων χρονοπρογραμματιστών. Στην επόμενη παράγραφο παρουσιάζεται αναλυτικά η λειτουργία του χρονοπρογραμματιστή CBQ[Risso01]

2.5.2.8.1 Ο χρονοπρογραμματιστής CBQ

Η αρχιτεκτονική του χρονοπρογραμματιστή CBQ παρουσιάζεται στο σχήμα 2.5. Στηρίζεται σε ένα γενικό “δίκαιο” χρονοπρογραμματιστή (General scheduler), ο οποίος ελέγχεται από έναν χρονοπρογραμματιστή διαμοιρασμού πόρων (link-Sharing Scheduler). Η εισερχόμενη κίνηση ταξινομείται (classifier) στην κατάλληλη ουρά, βάσει προκαθορισμένων κανόνων φιλτραρίσματος. Ο General scheduler (συνήθως ένας WRR) παίρνει τα πακέτα από την ουρά και τα προωθεί με τέτοιο τρόπο, ώστε κάθε κλάση εξυπηρέτησης να λαμβάνει τουλάχιστον το υποθετικό εύρος ζώνης που της αντιστοιχεί. Ο εκτιμητής (estimator) μετράει τον χρόνο προώθησης των πακέτων κάθε κλάσης και ελέγχει εάν αυτή έχει ξεπεράσει το προκαθορισμένο ρυθμό μετάδοσης πακέτων της (overlimit class). Ο link-Sharing Scheduler συνεργάζεται με τον εκτιμητή και κατανέμει το πλεονάζον εύρος ζώνης σύμφωνα με την δομή της κατανομής πόρων που υλοποιεί. Ο link-Sharing Scheduler αναστέλλει (για ορισμένο χρονικό διάστημα) την μετάδοση πακέτων από κλάσεις, οι οποίες έχουν υπερβεί τον ρυθμό μετάδοσης που τους έχει αποδοθεί. Ο χρόνος αναστολής υπολογίζεται με τέτοιο τρόπο, ώστε να υποχρεώσει την κλάση να συμμορφωθεί με τον ρυθμό μετάδοσης που της έχει δοθεί. Με απλά λόγια, μια κλάση, η μετάδοση πακέτων της οποίας έχει ανασταλεί, δεν είναι πλέον ενεργή και ο

WRR δεν τις παρέχει καμία υπηρεσία, μέχρι το τέλος της αναστολής. Ο Link-sharing scheduler λειτουργεί όπως ένα χρονοπρογραμματιστής PQ παρέχοντας εγγυήσεις για την καθυστέρηση μετάδοσης, δίνοντας όμως και ένα μέρος του εύρους ζώνης στις κλάσεις χαμηλής προτεραιότητας, ώστε αυτές να μην αδρανούν.



Σχήμα 2.5 Αρχιτεκτονική του χρονοπρογραμματιστή CBQ

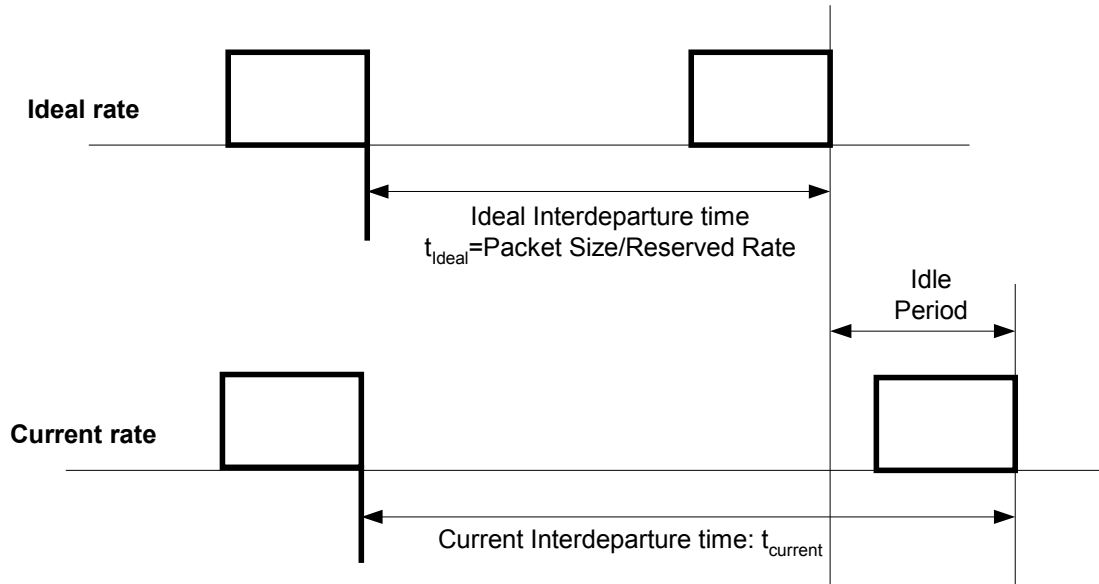
General scheduler

Ο General scheduler συνήθως αποτελείται από μια ομάδα από διαδοχικούς WRR χρονοπρογραμματιστές με διαφορετικές προτεραιότητες. Ένας WRR (ο οποίος σέβεται απόλυτα τις προτεραιότητες) μπορεί να αρχίσει να εξυπηρετεί της κλάσεις του, όταν οι κλάσεις υψηλότερης προτεραιότητας δεν έχουν αλλά πακέτα στην ουρά, ή όταν ο link-sharing scheduler δεν τους επιτρέπει να μεταδώσουν. Κάθε κλάση του WRR μπορεί να μεταδώσει ένα συγκεκριμένο αριθμό bytes σε κάθε γύρο (*allocation*) ο οποίος εξαρτάται από το εύρος ζώνης που έχει αποδοθεί σε κάθε κλάση. Το allocation αυξάνεται κατά μια συγκεκριμένη τιμή (*allotment*, διαφορετικό για κάθε κλάση) στο τέλος κάθε γύρου μετάδοσης. Το allotment εξαρτάται από το εγγυημένο εύρος ζώνης, που πρέπει να λάβει

η κλάση. Έτσι, κλάσεις με μεγάλο εύρος ζώνης μπορούν να μεταδώσουν περισσότερα δεδομένα σε κάθε γύρο. Το allocation κάθε κλάσης μειώνεται τόσο όσο ο αριθμός bytes που μεταδίδεται, κάθε φορά που η κλάση στέλνει ένα πακέτο. Μια κλάση μπορεί να μεταδώσει δεδομένα μόνο όταν το allocation της είναι θετικό. Το allocation μπορεί να πάρει αρνητικές τιμές, όμως δεν μπορεί να ξεπεράσει το allotment της κλάσης.

link-Sharing Scheduler

Ο link-Sharing Scheduler λαμβάνει υπόψη του διάφορες παραμέτρους, ώστε να ελέγχει τον τρόπο με τον οποίο προωθείται η κίνηση στο δίκτυο. Ο χρόνος αδρανείας Idle (σχήμα 2.6) είναι η διάφορα του τρέχοντος εσωτερικού χρόνου αποχώρησης inter-departure time ($IDT_{current}$) και του ιδεατού χρόνου αποχώρησης (Εξίσωση 2.1)



Σχήμα 2.6 Υπολογισμός της παραμέτρου αδρανείας (idle)

$$idle = IDT_{current} - IDT_{ideal} = IDT_{current} - \frac{p}{r \cdot f} \quad (2.1)$$

Όπου p είναι το μέγεθος του πακέτου που μεταδίδεται, r είναι το εύρος ζώνης της γραμμής και f είναι το κομμάτι εύρους ζώνης που έχει αποδοθεί στην συγκεκριμένη κλάση. Αν η κλάση υπερβαίνει τον προκαθορισμένο ρυθμό μετάδοσης τότε το $Idle > 0$, αλλιώς $idle < 0$.

Όμως η πιο σημαντική παράμετρος είναι το $avgidle$, το οποίο παρέχει το πραγματικό ρυθμό μετάδοσης της κίνησης. Υπολογίζεται με χρήση μιας εξίσωσης EWMA, με βάρος w (εξίσωση 2.2).

$$avgidle_{\pi+1} = (1-w) \cdot avgidle_{\pi} + w \cdot idle \quad (2.2)$$

Η μετάδοση αναστέλλεται για ένα συγκεκριμένο χρονικό διάστημα ($extradelat$) μόλις το $avgidle$ της κλάσης πάρει αρνητική τιμή. Έτσι, ο ρυθμός μετάδοσης της κλάσης συμμορφώνεται με το εύρος ζώνης που της έχει αποδοθεί. Το $extradelat$ υπολογίζεται με τέτοιο τρόπο (εξίσωση 2.3), ώστε η κλάση να διατηρήσει ένα σταθερό ρυθμό εκρηκτικότητας μετάδοσης m πακέτων, μετά το τέλος της αναστολής μετάδοσης. Έτσι λοιπόν, ο χρόνος αναστολής είναι:

$$\begin{aligned} avgidle_1 &= (1-w) \cdot avgidle_0 + w \cdot idle = 0 + w \cdot \left[extradelat + \frac{p}{r} \cdot \left(1 - \frac{1}{f} \right) \right] \\ avgidle_2 &= (1-w) \cdot avgidle_1 + w \cdot \frac{p}{r} \cdot \left(1 - \frac{1}{f} \right) \\ &\dots \\ avgidle_m &= (1-w) \cdot avgidle_{m-1} + w \cdot \frac{p}{r} \cdot \left(1 - \frac{1}{f} \right) = 0 \\ \Rightarrow extradelat &= \frac{p}{r} \cdot \left(\frac{1}{f} - 1 \right) \cdot \left(1 + \frac{1 - (1-w)^{m-1}}{w \cdot (1-w)^{m-1}} \right) \end{aligned} \quad (2.3)$$

Μεγαλύτερο $extradelat$ οδηγεί σε μεγαλύτερη εκρηκτικότητα, επιτρέποντας στην κλάση να στείλει περισσότερα πακέτα, πριν την αναστολή μετάδοσης. Η εκρηκτικότητα μετάδοσης πρέπει να κρατηθεί μικρή, ώστε να επιτρέπει ένα καλό επίπεδο πολυπλέξης μεταξύ συνδέσεων που χρησιμοποιούν την ίδια γραμμή. Μια κλάση υπό αναστολή θα στείλει τελικά ένα πακέτο μετά από χρόνο:

$$undertime = last_time + extradelat \quad (2.4)$$

όπου η παράμετρος $last_time$ είναι η χρονική στιγμή εξόδου του προηγούμενου πακέτου (που άνηκε σε αυτήν την κλάση) από τον χρονοπρογραμματιστή.

Τέλος, οι παράμετροι $minidle$ και $maxidle$ ορίζουν ένα άνω και ένα κάτω όριο για το $avgidle$, αντίστοιχα [Floyd95]. Το πρώτο αποτρέπει την τιμωρία μιας κλάσης για το

πλεονάζον εύρος ζώνης που πήρε, όταν οι άλλες κλάσεις ήταν σε αδράνεια. Το *maxidle* χρησιμοποιείται ώστε να αποτρέψει μια κλάση, που βρισκόταν σε αδράνεια, να στείλει δεδομένα για πολύ μεγάλο χρονικό διάστημα. Το *maxidle* μπορεί να υπολογιστεί με τρόπο παρόμοιο με τον υπολογισμό το *extradelay* (εξίσωση 2.5):

$$\begin{aligned}
 avgidle_1 &= (1-w) \cdot avgidle_0 + w \cdot offtime \geq \max idle \\
 avgidle_2 &= (1-w) \cdot \max idle + w \cdot \frac{p}{r} \left(1 - \frac{1}{f} \right) = 0 \\
 avgidle_M &= (1-w) \cdot avgidle_{M-1} + w \cdot \frac{p}{r} \left(1 - \frac{1}{f} \right) = 0 \\
 \max idle &= \frac{p}{r} \left(\frac{1}{f} - 1 \right) \left(\frac{1}{(1-w)^{M-1}} - 1 \right)
 \end{aligned} \tag{2.5}$$

Οι διαχειριστές δικτύων συνήθως δεν καθορίζουν τιμές για το *maxidle* και το *extradelay*. Αυτές οι παράμετροι αντικαθίστανται από τις παραμέτρους *maxburst* και *minburst* οι οποίες είναι ο μέγιστος αριθμός πακέτων που μπορεί να στείλει μια κλάση μετά από ένα μεγάλο διάστημα αδρανείας και ο μέγιστος αριθμός πακέτων που μπορεί να στείλει η κλάση σε σταθερή κατάσταση, αντίστοιχα. Το *maxidle* και το *extradelay* μπορούν εύκολα να εξαχθούν από τις παραμέτρους *maxburst* και *minburst*. Το *minburst* είναι η παράμετρος *m* στην εξίσωση 2.3 και το *maxburst* είναι η παράμετρος *M* στην εξίσωση 2.5.

2.5.3 Πολυ-πρωτόκολλο Μεταγωγής Ετικέτας

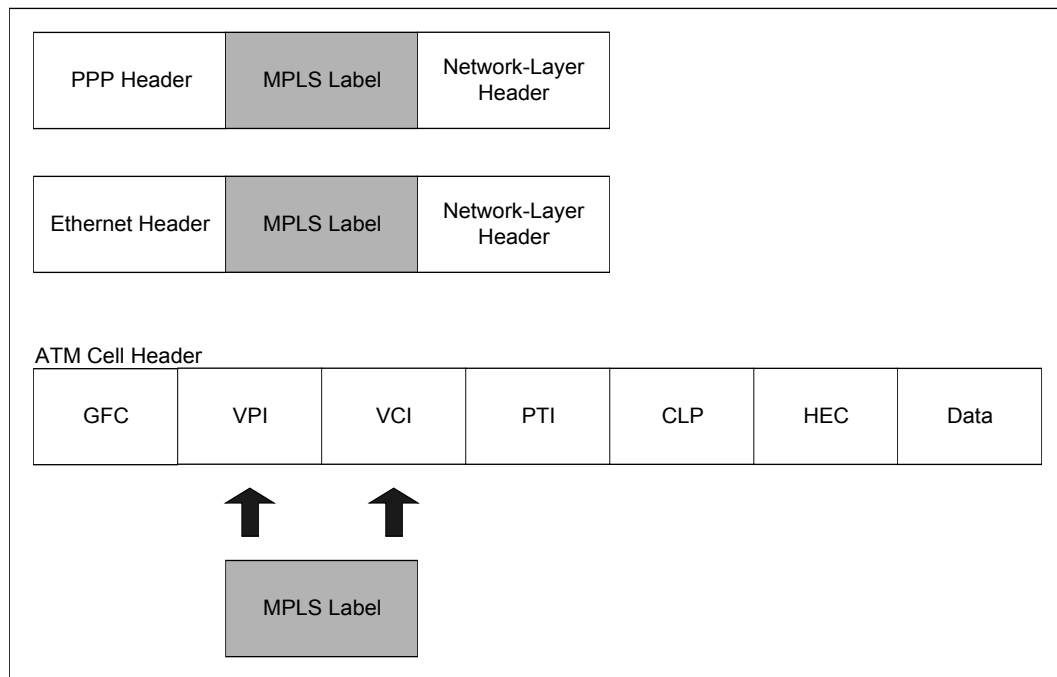
Ο τίτλος Πολυ-πρωτόκολλο Μεταγωγής Ετικέτας (Multi-Protocol Label Switching-MPLS) [MPLS] περιγράφει μια συνδυασμένη προσπάθεια στην IETF να ενοποιήσει τις καλύτερες από παρόμοιες έννοιες σε ένα τυποποιημένο πλαίσιο εργασίας και σε μια οικογένεια πρωτοκόλλων. Η τεχνολογία μεταγωγής ετικέτας αναμένεται να βελτιώσει το κόστος/απόδοση στο επιπέδου δρομολόγησης του δικτύου, να βελτιώσει την επεκτασιμότητα και να προσδώσει περισσότερη ευελιξία στην παροχή (νέων) υπηρεσιών δρομολόγησης (επιτρέποντας την προσθήκη νέων υπηρεσιών δρομολόγησης χωρίς κάποια αλλαγή στον τρόπο προώθησης).

Για την κατανόηση της έννοιας της μεταγωγής ετικέτας, πρέπει να εξεταστεί ο τρέχον τρόπος προώθησης που χρησιμοποιείται σήμερα. Η τρέχουσα μέθοδος για την δρομολόγηση και προώθηση αναφέρεται ως μέγιστη ταύτιση (longest match). Σύμφωνα με αυτή, ένας δρομολογητής αναφέρεται σε έναν πίνακα δρομολόγησης με προθέματα μεταβλητού μήκους και εγκαθιστά το “μακρύτερο”, ή το πιο συγκεκριμένο, πρόθεμα σαν προτίμηση για τους επακόλουθους μηχανισμούς προώθησης. Θεωρούμε ένα παράδειγμα όπου ένας δρομολογητής δέχεται ένα πακέτο που προορίζεται για την διεύθυνση 199.1.1.1. Υποθέτουμε ότι ο δρομολογητής έχει καταχωρήσεις στον πίνακα δρομολόγησης για τα 199.1.0.0/24 και 199.1.0.0/16. Υποθέτοντας ότι δεν υπάρχει έλεγχος διαχείρισης που θα μπορούσε να έρχεται σε αντίθεση με τη βασική συμπεριφορά της δυναμικής δρομολόγησης, ο δρομολογητής θα επιλέξει να προωθήσει το πακέτο σε ένα δρομολογητή από τον οποίο δέχθηκε μια ανακοίνωση για 199.1.0.0/24, επειδή είναι ένα πιο συγκεκριμένο πρόθεμα από το 199.1.0.0/16.

Είναι σημαντικό να σημειωθεί ότι ο χρόνος και οι πόροι που απαιτούνται για να πραγματοποιηθούν οι υπολογισμοί για τα μονοπάτια είναι ανάλογα με τον αριθμό των προθεμάτων και των πιθανών μονοπατιών. Σήμερα υπάρχει ένα πλήθος από προθέματα στο διαδίκτυο και συνεχίζει να υπάρχει μια αυξητική τάση. Ένα από τα αναμενόμενα αποτελέσματα της μεταγωγής ετικέτας, είναι η μείωση του χρόνου και των πόρων που απαιτούνται για να τους υπολογισμούς αυτούς. Με την μεταγωγή ετικέτας δεν υπάρχει πλέον η ανάγκη επιλογής του πιο συγκεκριμένου προθέματος (μέχρι ενός σημείου). Αυτό επιτυγχάνεται εισάγοντας μια ετικέτα (label) σταθερού μήκους, μεταξύ της επικεφαλίδας του επιπέδου δικτύου (επίπεδο 3) και της επικεφαλίδας του επιπέδου σύνδεσης (επίπεδο 2). Η ετικέτα αυτή μπορεί να χρησιμοποιηθεί για αποφάσεις σχετικές με το μονοπάτι και την προώθηση. Ο προσδιορισμός του καλύτερο μονοπατιού, βασιζόμενοι σε ένα σύνολο από τιμές σταθερού μήκους, είναι υπολογιστικά λιγότερο απαιτητικός, με αποτέλεσμα να απαιτεί και λιγότερο χρόνο.

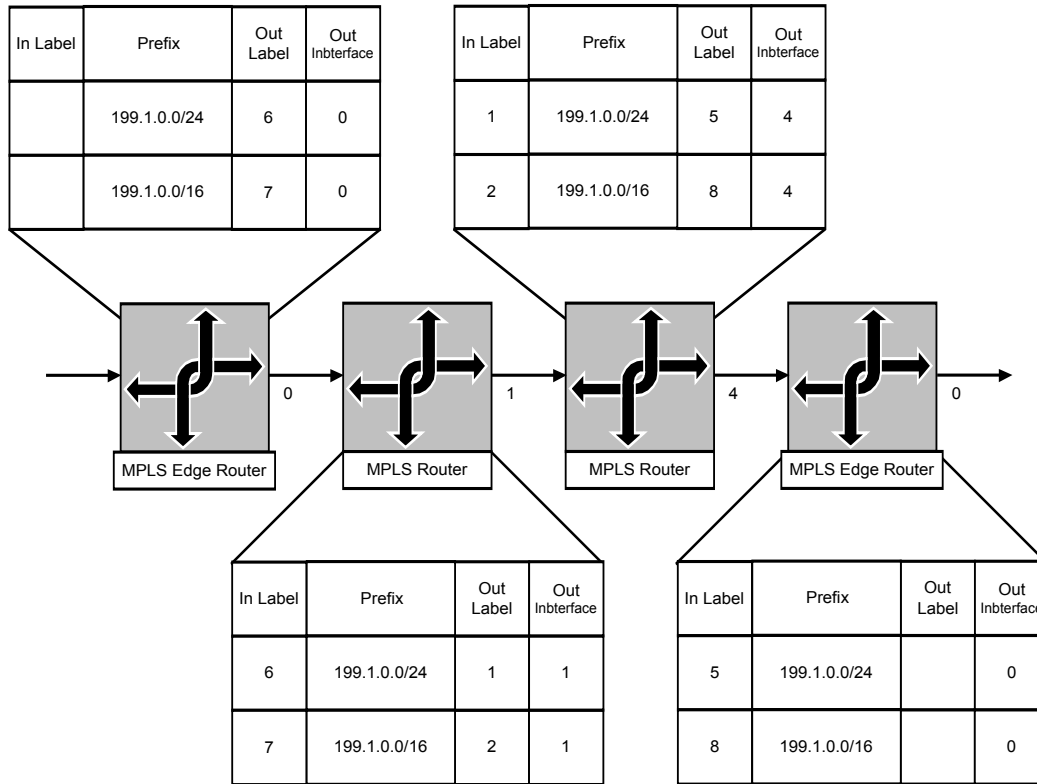
Το MPLS, μεταξύ σοβαρού και αστείου, αναφέρεται και ως πρωτόκολλο του επιπέδου 2.5 λόγω του ότι δεν είναι ούτε επίπεδο 3 ούτε επίπεδο 2, αλλά εισάγεται κάπου μεταξύ του επιπέδου δικτύου και του επιπέδου σύνδεσης. Το επίπεδο δικτύου θα μπορούσε να είναι οποιοδήποτε από τα διάφορα δίκτυα που χρησιμοποιούνται σήμερα, όπως IP, IPX,

AppleTalk κ.λπ. Έτσι εξηγείται και το *multiprotocol* στην ονομασία του MPLS. Ωστόσο, εξαιτίας της μεγάλης εξάπλωσης του διαδικτύου και των IP δικτύων, πρωταρχικό ενδιαφέρον για αυτή την τεχνολογία είναι να προσαρμόσει το IP πρωτόκολλο. Η διαδικασία της εισαγωγής ενός MPLS label σε πρωτόκολλα διαφόρων τύπων, παρουσιάζεται στην εικόνα 2.3.



Εικόνα 2.3: Εισαγωγή μιας MPLS ετικέτας

Οι ετικέτες κατανέμονται στο MPLS δίκτυο από ένα δυναμικό πρωτόκολλο κατανομής ετικετών, και περιορίζονται σε ένα πρόθεμα ή σε ένα σύνολο από προθέματα. Η σύνδεση προθέματος-ετικέτας γίνεται από έναν κόμβο στο άκρο του MPLS δικτύου (ένας δρομολογητής που αλληλεπιδρά με άλλους κόμβους που δεν υποστηρίζουν MPLS). Ο ακραίος MPLS κόμβος ανταλλάσσει πληροφορία για την δρομολόγηση με κόμβους που δεν υποστηρίζουν MPLS. Τοπικά συνδέει και ενοποιεί προθέματα που προέρχονται από το επίπεδο δικτύου σε MPLS ετικέτες και κατανέμει τις ετικέτες στους κόμβους του MPLS δικτύου [Rosen01]. Η διαδικασία αυτή παρουσιάζεται στην εικόνα 2.4.

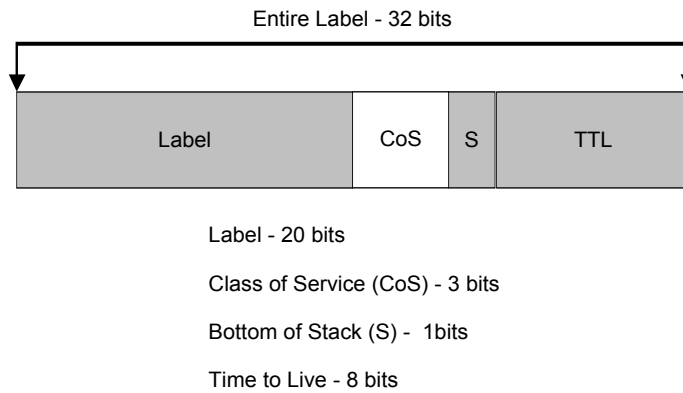


Εικόνα 2.4: Η διαδικασία μεταγωγής ετικέτας

Το MPLS δίνει πολλές δυνατότητες για παροχή ΠοιΥπ. Μια από αυτές είναι η απευθείας αντιστοίχιση των 3 bits του πεδίου προτεραιότητας, των επικεφαλίδων των IP πακέτων, σε ένα πεδίο της ετικέτας του MPLS. Το πεδίο αυτό ονομάζεται πεδίο κλάσης υπηρεσίας (Class of Service-CoS). Η MPLS ετικέτα και το πεδίο CoS παρουσιάζονται στην εικόνα 2.5. Καθώς τα IP πακέτα εισέρχονται σε ένα MPLS πεδίο, ο δρομολογητής στο άκρο του MPLS δικτύου είναι υπεύθυνος για την αντιστοίχιση των bits στην επικεφαλίδα του IP πακέτου, στο CoS πεδίο της MPLS επικεφαλίδας.

Από το ένα άκρο ενός MPLS δικτύου σε ένα άλλο μπορούν να υπάρχουν διάφορα παράλληλα μονοπάτια με διαφορετικά χαρακτηριστικά, για παράδειγμα με διαφορετικό εύρος ζώνης. Με την χρήση του πεδίου CoS, είναι δυνατό να επιλεχθούν διαφορετικά μονοπάτια και κάθε μονοπάτι να προσφέρει ένα ξεχωριστό χαρακτηριστικό. Η κυκλοφορία, που έχει ετικέτες με μεγαλύτερες τιμές CoS, μπορεί να προωθηθεί κατά μήκος ενός μονοπατιού υψηλότερης ταχύτητας και μικρότερης καθυστέρησης. Η κυκλοφορία που έχει ετικέτες με μικρότερες τιμές CoS μπορεί να προωθηθεί κατά μήκος ενός μονοπατιού χαμηλότερης ταχύτητας και μεγαλύτερης καθυστέρησης. Αυτό το

παράδειγμα δείχνει μια μέθοδο για την παροχή ΔΥ με χρήση του MPLS [Heinanen99]. Βεβαίως, υπάρχουν διαφορετικές προσεγγίσεις. Η παροχή ΠοιΥπ από το MPLS μπορεί να γίνει χωρίς να ληφθεί υπόψη το CoS, αλλά να βασιστεί σε άλλα κριτήρια, όπως η διεύθυνση της πηγής, η διεύθυνση του δέκτη ή χαρακτηριστικά της ροής. Ακόμη, μπορεί να είναι σε αντιστοιχία με RSVP μηνύματα [Awduche98].



Εικόνα 2.5: Τα CoS bits στην ετικέτα του MPLS

2.6 Η Ποιότητα Υπηρεσίας στα πλαίσια της διατριβής

Υπάρχουν πολλοί τρόποι με τους οποίους κάποιος μπορεί να αντιληφθεί την έννοια της ποιότητας υπηρεσίας στα δίκτυα επικοινωνιών. Επίσης υπάρχουν διάφοροι μηχανισμοί και αρχιτεκτονικές παροχής ΠοιΥπ, ώστε δεν είναι δυνατόν μια διατριβή να αναλύσει όλα αυτά τα θέματα, ή τουλάχιστον να ασχοληθεί με όλα στον ίδιο βαθμό. Στα πλαίσια της διατριβής η ΠοιΥπ αντιμετωπίζεται περισσότερο ως η ικανότητα ενός δικτυακού στοιχείου να εξασφαλίσει ένα συγκεκριμένο επίπεδο εξυπηρέτησης. Όσον αφορά το επίπεδο δικτυακής εφαρμογής (χρήστη, εφαρμογής, δικτύου, κτλ) η διατριβή εστιάζει στην ποιότητα υπηρεσίας σε επίπεδο δικτύου. Έτσι λοιπόν, οι παράμετροι ΠοιΥπ στα πλαίσια της διατριβής είναι η καθυστέρηση μετάδοσης, η απόκλιση καθυστέρησης, ο ρυθμός λαθών και το ωφέλιμο φορτίο.

Σε πρακτικό επίπεδο η παροχή ΠοιΥπ επιτυγχάνεται με την κατάλληλη διάρθρωση των δρομολογητών του δικτύου, ώστε να παρέχουν εγγυήσεις για την τιμές των παραπάνω παραμέτρων. Επίσης, σε ότι αφορά τους μηχανισμούς ποιότητας υπηρεσίας η διατριβή ασχολείται κυρίως με τους μηχανισμούς έλεγχου ΠοιΥπ και λιγότερο στους μηχανισμούς προπαρασκευής και διαχείρισης.

Το μοντέλο Ολοκληρωμένων Υπηρεσιών παρουσίασε αρκετές δυσκολίες στην εφαρμογή του, με κυριότερο την επεκτασιμότητα σε μεγάλα δίκτυα. Το πρόβλημα αυτό ήρθε να λύσει η αρχιτεκτονική ΔΥ. Η εφαρμογή διαχείρισης και παρακολούθησης ΠοιΥπ, που αναπτύχθηκε στα πλαίσια της διατριβής και θα παρουσιαστεί σε επόμενο κεφάλαιο, είναι βασισμένη στην αρχιτεκτονική ΔΥ. Στην αρχιτεκτονική αυτή είναι βασισμένες και όλες οι άλλες πρωτότυπες εφαρμογές που αναπτύχθηκαν, για την διεξαγωγή πειραμάτων.

2.7 Σύνοψη κεφαλαίου

Στο κεφαλαίο αυτό, έγινε μια προσπάθεια να προσδιοριστεί η έννοια της ποιότητας υπηρεσίας στο χώρο των δικτύων επικοινωνιών. Τονίστηκε το γεγονός, ότι η έννοια αυτή είναι συγκεχυμένη και η ερμηνεία που δίνεται εξαρτάται από τις διαφορετικές αντιλήψεις που υπάρχουν γύρω από το θέμα. Τέλος, δόθηκαν μια σειρά από ορισμούς της ΠοιΥπ, από διάφορους οργανισμούς του διαδικτύου.

Στην συνέχεια παρουσιάστηκε η ροή της πληροφορίας στο διαδίκτυο και οι λόγους για τους οποίους, η ποιότητα υπηρεσίας που παρέχεται είναι χαμηλή. Έπειτα, παρουσιάστηκαν οι λόγοι για τους οποίους το διαδίκτυο δεν παρέχει ΠοιΥπ. Η δυσκολία παροχής υπηρεσιών από το διαδίκτυο, εντοπίζεται στην ιδέα της “καλύτερης προσπάθειας”, πάνω στην οποία είναι δομημένο. Το αντάλλαγμα για την απλότητα που προσέδωσε η ιδέα αυτή, ήταν η έλλειψη εγγυήσεων στην παροχή υπηρεσιών. Η ανάγκη για την παροχή ΠοιΥπ από το διαδίκτυο, δημιουργήθηκε από τις νέες εφαρμογές. Παρουσιάζοντας τις νέες δικτυακές εφαρμογές απεδείχθη, ότι το πρόβλημα ΠοιΥπ στο διαδίκτυο σήμερα δεν σχετίζεται μόνο με το εύρος ζώνης, αλλά και τις αυστηρές απαιτήσεις που έχουν οι εφαρμογές αυτές, όσον αφορά την καθυστέρηση μετάδοσης. Οι νέες εφαρμογές δεν αύξησαν μόνο την κυκλοφορία δεδομένων στο διαδίκτυο, αλλά άλλαξαν και την φύση της κυκλοφορίας αυτής.

Κατόπιν, παρουσιάστηκαν οι γενικές αρχές, οι προδιαγραφές και οι μηχανισμοί που πρέπει να παρέχονται από ένα δίκτυο επικοινωνιών, για την υποστήριξη ΠοιΥπ. Η γνώση αυτού του θεωρητικού υποβάθρου, κρίθηκε απαραίτητη για την ευκολότερη κατανόηση των αρχιτεκτονικών παροχής ΠοιΥπ. Κλείνοντας αυτήν την ενότητα, παρουσιάστηκαν τις τρεις επικρατέστερες αρχιτεκτονικές παροχής ΠοιΥπ που είναι οι εξής: ολοκληρωμένες υπηρεσίες, διαφοροποιημένες υπηρεσίες και το Πολυ-πρωτόκολλο μεταγωγής ετικέτας.

2.8 Βιβλιογραφικές αναφορές

[Ferguson98] P. Ferguson and G. Huston, “Quality of Service: Delivering quality of service on the Internet and in corporate networks”, *J. Wiley & Sons, Inc.*, 1998, ISBN 0471243582.

[Zheng99] B. Zheng and M. Atiquzzammn “Traffic management of multimedia over ATM Networks”, *IEEE Communications magazine* Vol. 37, No. 1, January 1999, pp.38-42.

[QoS99] QoS Forum, “The need for QoS”, *White paper*, Διαθέσιμο στην ηλεκτρονική διεύθυνση: http://www-sop.inria.fr/rodeo/rserban/Files/Need_for_QoS-v4.pdf, 1999

[Saltzer84] D.R. Saltzer and D. Clark, “End to End Arguments in System Design”, *ACM Transactions in Computer Systems*, November 1984, pp. 277-288.

[Lefelhocz96] C. Lefelhocz, B. Lyles, S. Shenker and L. Zhang, “Congestion Control for Best-Effort Service: Why We Need a New Paradigm”, *IEEE Network Magazine*, Vol. 10, No. 1, Jan/Feb 1996.

[Kurose93] J.F. Kurose, “Open Issues and Challenges in Providing Quality of Service Guarantees in High Speed Networks”, *ACM Computer Communications Review* Vol. 23, No. 1, January 1993, pp. 6-15.

[Lazar92] A.A. Lazar, “A Real-time Control, Management, and Information Transport Architecture for Broadband Networks”, *Proc. International Zurich Seminar on Digital Communications*, 1992, pp. 281-295.

[Aurrecoechea96] C. Aurrecoechea and A. T. Campbell, “A Survey of QoS Architectures”, *Proc. IFIP International workshop on Quality of Service*, Paris, March, 1996.

[Ferrari90] D. Ferrari, and D.C. Verma, “A Scheme for Real-Time Channel Establishment in Wide-Area Networks”, *IEEE J. Select. Areas Commun.*, Vol. 8, No. 3, April 1990, pp. 368-379.

[Zhang91] H. Zhang and S. Keshav, “Comparison of Rate-Based Service Disciplines”, *Proc. ACM SIGCOMM*, 1991, pp. 113-121.

[Campbell94] A. Campbell, G. Coulson, and D. Hutchison, “A Quality of Service Architecture”, *ACM Computer Communications Review*, April 1994.

[QoS99] QoS Forum, “Qos Protocols and architectures”, *White paper*, Διαθέσιμο στην ηλεκτρονική διεύθυνση: <http://www.cs.ucsb.edu/~almeroth/classes/F01.201B/papers/qos.pdf>, 1999.

[IETF] Αποτελεί την ομάδα ανάπτυξης του διαδικτύου. Αποτελείται από εθελοντές από τον χώρο την βιομηχανίας δικτύων και των ακαδημαϊκό χώρο. Σκοπός τους είναι η δημιουργία νέων πρωτοκόλλων. Περισσότερες πληροφορίες στην ηλεκτρονική διεύθυνση: <http://www.ietf.org>.

[IntServ] Η πληροφορία για της Ολοκληρωμένες Υπηρεσίες όπως αναπτύχθηκαν από την ομάδα εργασίας της IETF “Integrated Services” είναι διαθέσιμη στις παρακάτω ηλεκτρονικές διευθύνσεις: <http://www.ietf.org/html.charters/intservcharter.html> και <http://www.ietf.org/ids.by.wg/intserv.html>.

[Wroclawski97A] J. Wroclawski, “Specification of the Controlled-Load Network Element Service”, *IETF RFC 2211*, September 1997.

[Shenker97] S. Shenker, C. Partridge and R. Guerin, “Specification of Guaranteed Quality of Service”, *IETF RFC 2212*, September 1997

[Braden97] R. Braden, L. Zhang, S. Berson, S. Herzog and S. Jamin, “Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification”, *IETF RFC 2205*, September 1997.

[Wroclawski97B] J. Wroclawski, “The Use of RSVP with IETF Integrated Services,” *IETF RFC 2210*, September 1997.

[Diffserv] Η πληροφορία για της Διαφοροποιημένες Υπηρεσίες όπως αναπτύχθηκαν από την ομάδα εργασίας της IETF “differentiated Services” είναι διαθέσιμη στις παρακάτω ηλεκτρονικές διευθύνσεις: <http://www.ietf.org/html.charters/diffserv-charter.html> και <http://www.ietf.org/ids.by.wg/diffserv.html>

- [Blake98] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss, “An Architecture for Differentiated Services”, *IETF RFC 2475*, December 1998.
- [Nichols98] K. Nichols, S. Blake, F. Baker, D. Black, “Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers”, *IETF RFC 2474*, December 1998.
- [Almquist99] P. Almquist, “Type of Service in the Internet Protocol Suite”, *IETF RFC 1349*, July 1992.
- [Heinanen99] J. Heinanen, T. Finland and R. Guerin “ A Single Rate Three Color Marker”, *IETF RFC 2697*, September 1999.
- [Jacobson99] V. Jacobson, K. Nichols and K. Poduri, “An Expedited Forwarding PHB”, *IETF RFC 2598*, June 1999.
- [Heinanen99] J. Heinanen, F. Baker, W. Weiss and J. Wroclawski, “Assured Forwarding PHB Group”, *IETF RFC 2597*, June 1999.
- [Nagle87] J. Nagle, “On packet switches with infinite storage”, *IEEE Transactions on Communications*, Vol. 35, No. 4, April 1987, pp. 435-438.
- [Demers89] A. Demers, S. Keshav, and S. Shenker, “Analysis and Simulation of A Fair Queuing Algorithm”, *Journal of Internetworking Research and Experience*, October 1990, pp. 3-26.
- [Parekh93] A. K. Parekh and R. G. Gallager, “A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Single-node Case”, *IEEE/ACM Transactions on Networking*, Vol.1, No. 3, June 1993, pp. 344–357.
- [Parekh94] A. K. Parekh and R. G. Gallager, “A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Multiple Node Case, *IEEE/ACM Transactions on Networking*, Vol. 2, No. 2, Apr. 1994, pp. 137–150.
- [Floyd95] S. Floyd, “Notes on Class Based Queuing: Setting Parameters” *Informal notes*, September 1995, διαθέσιμο στην ηλεκτρονική διεύθυνση: <http://www.icir.org/floyd/cbq.html>.

[Risso01] Fulvio Risso, “Decoupling Bandwidth and Delay Properties in Class Based Queuing”, *Proc 6th IEEE Symposium on Computers and Communications (ISCC 2001)*, Hammamet, Tunisia, July 2001, pp. 524-531

[MPLS] Η πληροφορία για το Πολυ-Πρωτόκολλο Μεταγωγής Ετικέτας όπως αναπτύχθηκε από την ομάδα εργασίας της IETF “ Multi-Protocol Label Switching” είναι διαθέσιμη στις παρακάτω ηλεκτρονικές διευθύνσεις:

<http://www.ietf.org/html.charters/mppls-charter.html>

και <http://www.ietf.org/ids.by.wg/diffserv.html>.

[Rosen01] E. Rosen, A. Viswanathan, R. Callon, “Multiprotocol Label Switching Architecture”, *IETF RFC3031*, January 2001.

[Heinanen99] J. Heinanen, “Differentiated Services in MPLS Networks”, *IETF draft*, June 1999.

[Awduche98] D. Awduche, D. Gan, T. Li, G. Swallow and V. Srinivasan, “Extensions to RSVP for Traffic Engineering”, *IETF draft*, August 1998.

ΚΕΦΑΛΑΙΟ 3

ΤΕΧΝΟΛΟΓΙΑ ΚΙΝΗΤΩΝ ΠΡΑΚΤΟΡΩΝ

3.1 Εισαγωγή

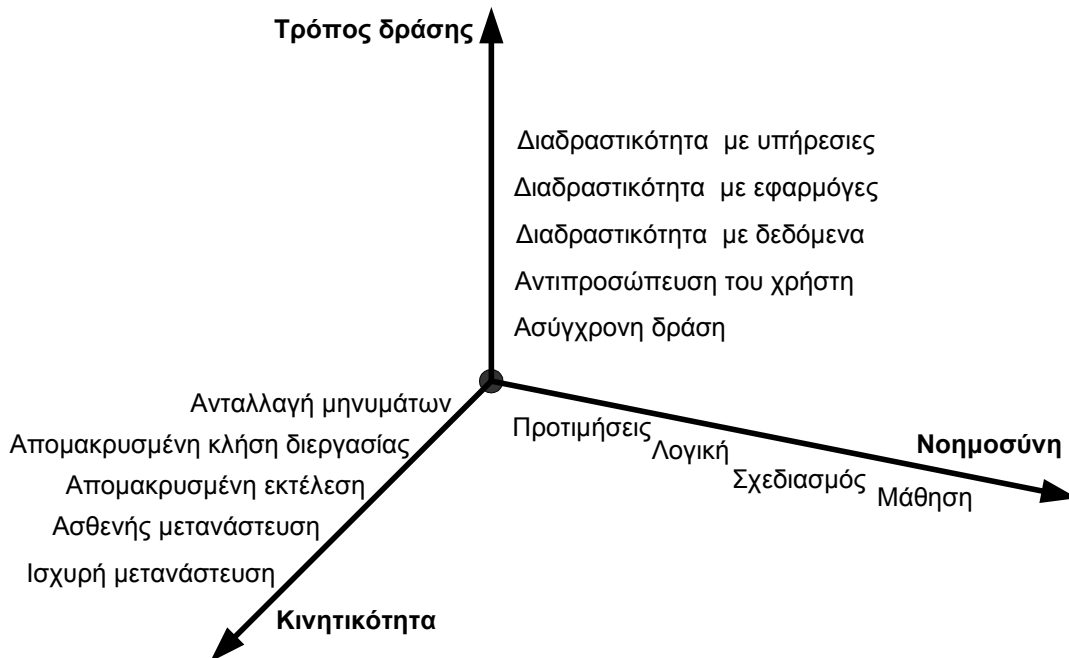
Ο όρος πράκτορας (agent) προέρχεται από την ελληνική λέξη “άγω”, που σημαίνει οδηγώ ή διευθύνω. Σήμερα ο όρος πράκτορας υποδηλώνει κάτι που παράγει, ή είναι ικανό να παράξει κάποιο αποτέλεσμα. Με βάση αυτόν τον ευρύ ορισμό, ο όρος πράκτορας μπορεί να χρησιμοποιηθεί για να χαρακτηρίσει τόσο αντικείμενα όσο και ανθρώπους (π.χ ταξιδιωτικός πράκτορας). Η έννοια του πράκτορα ως κάτι που οδηγεί η διευθύνει μια κατάσταση, ταιριάζει απόλυτα στις αντιλήψεις που κυριαρχούν σήμερα στο χώρο της επιστήμης των υπολογιστών. Σύμφωνα με αυτές, ο υπολογιστής πρέπει να πάψει να είναι ένα παθητικό αντικείμενο και να εξελιχθεί σε ένα πιο ενεργό. Θα πρέπει οι εργασίες να πάψουν να υπαγορεύονται στο υπολογιστή, αλλά να αναθέτονται. Με αυτό το σκεπτικό, δημιουργήθηκαν διάφορες προγραμματιστικές τεχνικές και προγράμματα, με σκοπό να προσδώσουν ένα πιο ενεργό ρόλο στους υπολογιστές. Έτσι λοιπόν γεννήθηκαν τα προγράμματα που σήμερα είναι γνωστά ως λογισμικοί πράκτορες (software agents).

Οι λογισμικοί πράκτορες μπορούν να ταξινομηθούν σε έναν τρισδιάστατο χώρο [Gilbert95], όπου οι τρεις διαστάσεις είναι οι: νοημοσύνη, τρόπος δράσης και κινητικότητα. Η ταξινόμηση αυτή παρουσιάζεται στο σχήμα 3.1.

Η πρώτη διάσταση η νοημοσύνη, έχει τις ρίζες στην ερευνά του τομέα της τεχνητής νοημοσύνης. Την δεκαετία του πενήντα ο Selfridge χρησιμοποίησε τον όρο πράκτορας για να χαρακτηρίσει ένα λογισμικό ρομπότ (soft robot), που θα “ζούσε” και θα δρούσε μέσα σε έναν υπολογιστή. Στόχος αυτής της προσπάθειας ήταν η δημιουργία ενός πράκτορα, που θα μπορούσε να εφαρμόσει τεχνικές συμβολικής τεχνητής νοημοσύνης και θα ήταν ικανός να εκτελέσει μια συγκεκριμένη διεργασία. Αυτοί οι ευφυείς πράκτορες μπορούν να ταξινομηθούν με βάση την ικανότητα να εκφράσουν προτιμήσεις

και την δυνατότητα να εκτελέσουν μια διεργασία, χρησιμοποιώντας λογική, σχεδιασμό και μάθηση [Shoam94].

Η δεύτερη διάσταση, ο τρόπος δράσης, είναι ο βαθμός αυτονομίας και εξουσίας που δίνεται στον πράκτορα. Μπορεί να μετρηθεί με βάση την φύση της αλληλεπίδρασης του πράκτορα με τα άλλα στοιχεία του συστήματος. Ένας πράκτορας θα πρέπει να είναι τουλάχιστον σε θέση να δράσει αυτόνομα. Ο βαθμός αυτονομίας ενισχύεται, αν ο πράκτορας είναι και κατά κάποιο τρόπο αντιπρόσωπος του χρήστη. Ένας πιο προηγμένος πράκτορας είναι σε θέση να αλληλεπιδράσει με τα δεδομένα, τις εφαρμογές και τις υπηρεσίες του συστήματος. Σύμφωνα με τις ικανότητες τους οι πράκτορες διαχωρίζονται σε αυτόνομους, συνεργαζόμενους και διαπραγματευόμενους [Chaib-Draa94].



Σχήμα 3.1: Ταξινόμηση των λογισμικών πρακτόρων

Η τρίτη διάσταση, η κινητικότητα, εμφανίστηκε την δεκαετία του ενενήντα και είναι αποτέλεσμα της ανάπτυξης των δικτύων υπολογιστών. Η μεγάλη ανάπτυξη των δικτύων υπολογιστών, αύξησε την απαίτηση για τεχνικές αξιοποίησης του τεράστιου πλήθους δεδομένων που προέκυψε. Αυτή η διάσταση των πρακτόρων προσδιορίζει την δυνατότητα μετακίνησης δεδομένων και υπολογιστικής δύναμης από κόμβο σε κόμβο, σε ένα δίκτυο υπολογιστών. Οι πράκτορες που έχουν την ικανότητα μετακίνησης

ονομάζονται κινητοί πράκτορες (mobile agents). Η κατηγορία αυτή θα παρουσιαστεί αναλυτικά σε επόμενη ενότητα.

Ένας λογισμικός πράκτορας δεν είναι υποχρεωτικό να εμφανίζει και τις τρεις ικανότητες που παρουσιάστηκαν παραπάνω. Υπάρχουν παραδείγματος χάριν, πράκτορες οι οποίοι δεν έχουν την ικανότητα να κινούνται. Αυτοί ονομάζονται στατικοί πράκτορες και εκτελούν μια συγκεκριμένη εργασία πάνω σε ένα μόνο υπολογιστή.

3.2 Μια ιστορική ανάδρομη - Από την μέθοδο ανταλλαγής μηνυμάτων στην τεχνολογία κινητών πρακτόρων

Η επικοινωνία μεταξύ των διαφορετικών τμημάτων ενός κατανεμημένου συστήματος μπορεί να υποστηριχθεί από διάφορες μεθόδους. Η πρώτη μέθοδος που προτάθηκε, ήταν η ανταλλαγή μηνυμάτων (message passing). Με αυτή τη μέθοδο, οι διάφορες διεργασίες του κατανεμημένου συστήματος επικοινωνούν με αποστολή και λήψη μηνυμάτων. Η ανταλλαγή αυτή μπορεί να είναι τόσο σύγχρονη όσο και ασύγχρονη. Η ασύγχρονη ανταλλαγή μηνυμάτων είναι πιο ευέλικτη και μπορεί να υποστηρίξει μια μεγάλη ποικιλία εφαρμογών. Η μέθοδος αυτή θεωρήθηκε αρχικά ελπιδοφόρα για την υλοποίηση κατανεμημένων συστημάτων, στην πορεία όμως αποδείχτηκε ότι η χρήση της ήταν αρκετά περίπλοκη.

Η εναλλακτική λύση που προτάθηκε, στηρίζεται στην ιδέα της απομακρυσμένης κλήσης διεργασίας (Remote Procedure Call-RPC). Με αυτή τη μέθοδο, η επικοινωνία επιτυγχάνεται με απευθείας κλήση της απομακρυσμένης διεργασίας και όχι με αποστολή και λήψη μηνυμάτων. Με την μέθοδο RPC οι διεργασίες πραγματοποιούνται με βάση το μοντέλο πελάτη/εξυπηρετητή (client/server). Οι πελάτες ζητούν την εκτέλεση μιας διεργασίας από τους εξυπηρετητές, αυτοί εκτελούν την ζητούμενη διεργασία και επιστρέφουν τα αποτελέσματα στους πελάτες. Η μέθοδος υποστηρίζει σύγχρονη και ασύγχρονη κλήση διεργασιών. Μεγάλο μέρος από την πολυπλοκότητα της επικοινωνίας εμπεριέχεται σε υπο-διεργασίες οι οποίες πραγματοποιούνται αυτόματα. Υπάρχουν παραδείγματα χάριν υπό-διεργασίες οι οποίες πραγματοποιούν την κωδικοποίηση/αποκωδικοποίηση των μηνυμάτων κλήσης διεργασιών σε ετερογενή συστήματα. Έτσι, η δημιουργία εφαρμογών με την χρήση της μεθόδου RPC είναι αρκετά πιο απλή, από ότι με την μέθοδο ανταλλαγής μηνυμάτων.

Είναι προφανές ότι για να λειτουργήσει ή μέθοδος RPC θα πρέπει η καλούμενη διεργασία, να είναι διαθέσιμη στο απομακρυσμένο μηχάνημα. Αυτή η απαίτηση περιορίζει την χρήση της μεθόδου σε μεγάλα κατανεμημένα συστήματα. Σε πολλές περιπτώσεις, είναι θεμιτή η αποστολή μιας ολόκληρης διεργασίας σε ένα απομακρυσμένο μηχάνημα (π.χ. ο πελάτης στέλνει ένα ειδικό φίλτρο σε μια απομακρυσμένη βάση δεδομένων, επιτυγχάνοντας την επεξεργασία των δεδομένων

τοπικά). Αυτού του είδους η λειτουργία παρέχεται από μια μέθοδο που ονομάζεται απομακρυσμένη αξιολόγηση (remote evaluation) [Stamos90A, Stamos90B] και είναι μια επέκταση της RPC μεθόδου. Με την απομακρυσμένη αξιολόγηση, δεν αποστέλλονται μόνο οι παράμετροι της καλούμενης διεργασίας, αλλά και ο κώδικας της διεργασίας. Η διεργασία εκτελείται στο απομακρυσμένο μηχάνημα και τα αποτελέσματα μεταδίδονται στο αρχικό μηχάνημα, μετά την ολοκλήρωσή της. Η διαδικασία αποστολής και λήψης κώδικα από ένα μηχάνημα σε ένα άλλο, είναι γενικά γνωστή ως απομακρυσμένη εκτέλεση (remote execution).

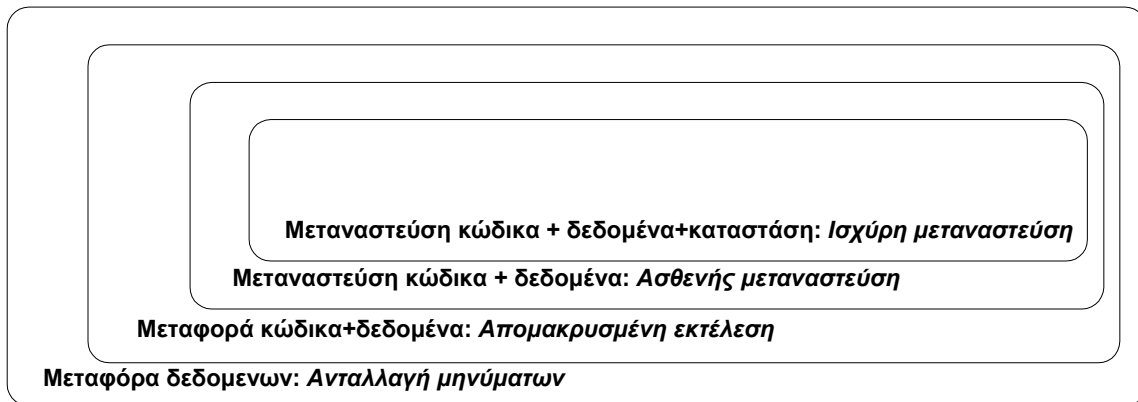
Η απομακρυσμένη εκτέλεση επιτρέπει μόνο την μεταφορά του κώδικα της διεργασίας και την λήψη των αποτελεσμάτων της. Η ιδέα των κινητών πρακτόρων (ΚΠ) επιτρέπει την μεταφορά ολόκληρων προγραμμάτων. Η λειτουργία των ΚΠ προϋποθέτει όχι μόνο την μεταφορά του κώδικα, αλλά και την κατάσταση του πράκτορα στο απομακρυσμένο μηχάνημα. Η κατάσταση ενός πράκτορα χωρίζεται στα δεδομένα και στο στάδιο εκτέλεσης της διεργασίας που εκτελείται. Η διαδικασία μεταφοράς ενός πράκτορα από ένα μηχάνημα σε ένα άλλο καλείται μετανάστευση (migration). Διακρίνονται δυο τύποι μετανάστευσης Η ασθενής μετανάστευση και η ισχυρή.

Στην ισχυρή μετανάστευση (strong migration), τόσο τα δεδομένα όσο και το στάδιο εκτέλεσης μεταδίδονται μαζί με τον κώδικα του πράκτορα στο απομακρυσμένο μηχάνημα. Μόλις ο πράκτορας φτάσει στον προορισμό του επαναδημιουργείται και συνεχίζει την εκτέλεση του από το σημείο που σταμάτησε πριν την μεταφορά. Η ιδέα αυτή είναι πολύ ελκυστική για του δημιουργούς κατανεμημένων εφαρμογών. Υπάρχει όμως ένα μειονέκτημα. Αρκετές φορές το μέγεθος της πληροφορίας, για την επαναδημιουργία του πράκτορα, είναι μεγάλο. Αυτό έχει ως αποτέλεσμα την αύξηση του χρόνου μεταφοράς και την δέσμευση εύρους ζώνης του δικτύου.

Αυτό το μειονέκτημα οδήγησε στην δημιουργία της λεγόμενης ασθενής μετανάστευσης (weak migration). Στην ασθενή μετανάστευση δεν μεταφέρεται το στάδιο εκτέλεσης του πράκτορα, αλλά μόνο τα δεδομένα. Αυτό έχει ως αποτέλεσμα την μείωση του μεγέθους του πράκτορα, με ανάλογη όμως μείωση των ικανοτήτων του. Περαιτέρω μείωση του μεγέθους μπορεί να επιτευχθεί με επιλογή, από τον προγραμματιστή, των δεδομένων που θα μεταφερθούν. Αν και η μέθοδος αυτή μειώνει το μέγεθος του πράκτορα, επιβαρύνει

τον προγραμματιστή με την επιλογή των δεδομένων που είναι απαραίτητα για την σωστή λειτουργία του μετά την μεταφορά.

Με μια πρώτη μάτια η ισχυρή μετανάστευση μοιάζει με την ιδέα την μετανάστευσης διεργασίας (process migration) [Smith88]. Όμως υπάρχει μια βασική διάφορα. Στην μετανάστευση διεργασίας η απόφαση για το πια διεργασία θα κινηθεί, σε πιο μηχάνημα και ποτέ, παίρνεται από το σύστημα. Ενώ, στην μετανάστευση πράκτορα οι αποφάσεις αυτές παίρνονται από τον ίδιο τον πράκτορα. Κατά συνέπεια όλες οι γλώσσες προγραμματισμού που χρησιμοποιούνται για την δημιουργία ΚΠ, πρέπει να περιέχουν μια εντολή μετακίνησης (move). Στο σχήμα 3.2 παρουσιάζονται οι βαθμοί κινητικότητας λογισμικών πρακτόρων, όπως παρουσιάστηκαν στις προηγούμενες παραγράφους.



Σχήμα 3.2: βαθμοί κινητικότητας λογισμικών πρακτόρων

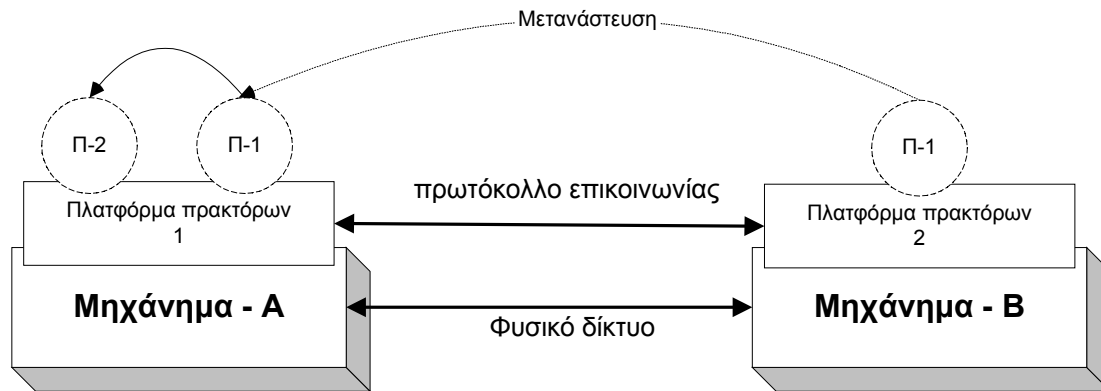
3.3 *Κινητοί πράκτορες*

Το όραμα ενός “ψηφιακού βοηθού” που θα μετακινείται μέσα στο δίκτυο για την εκτέλεση μιας συγκεκριμένης εργασίας εκ’ μέρους του χρήστη, αποτέλεσε το ερέθισμα για την ερευνά στην περιοχή των ΚΠ. Αυτή η απόδοση ανθρώπινων χαρακτηριστικών στους ΚΠ είναι λίγο παραπλανητική. Οι χρηστές μπορεί να νιώσουν άβολα με την ύπαρξη ενός “έξυπνου βοηθού” ο οποίος εκτελεί εργασίες, με έναν μη προβλέψιμο τρόπο [Shneiderman97]. Υπάρχει όμως πραγματικά ανάγκη για προγράμματα τα οποία προβλέψιμα και αξιόπιστα θα είναι ικανά να εκτελέσουν συγκεκριμένες εργασίες μέσα σε ένα δίκτυο υπολογιστών, χωρίς την επίβλεψη του χρήστη. Τέτοια προγράμματα θα μπορούν να εκτελέσουν μια εργασία ακόμα και όταν ο χρήστης δεν είναι συνδεδεμένος στο δίκτυο και να έχουν πρόσβαση σε μηχανήματα, στα οποία ο χρήστης δεν έχει λογαριασμό πρόσβασης. Με πιο απλά λόγια θα είναι ικανά να βοηθήσουν τον χρήστη σε ένα αχανές κατακευματισμένο δίκτυο, όπως το διαδίκτυο. Τα προγράμματα οι ικανότητες των οποίων μόλις περιγράφηκαν, είναι σήμερα γνωστά ως κινητοί πράκτορες.

Ένας πιο συγκεκριμένος ορισμός των ΚΠ, σύμφωνα με τον οργανισμό Object Management Group (OMG) είναι ο παρακάτω.

Ένας κινητός πράκτορας είναι ένα πρόγραμμα το οποίο δεν είναι δεσμευμένο στο σύστημα το οποίο εκτελείται αλλά έχει την ικανότητα να μετακινεί τον εαυτό του από ένα σύστημα σε ένα άλλο, σε ένα δίκτυο υπολογιστών [OMG01].

Στο σχήμα 3.3 παρουσιάζεται ένα απλό παράδειγμα της λειτουργία ΚΠ σε ένα κατακευματισμένο σύστημα. Σε αυτό το σημείο θα θεωρήσουμε το περιβάλλον εξυπηρέτησης του πράκτορα (πλατφόρμα) ως ένα μαύρο κουτί, που του παρέχει υπηρεσίες. Η ανάλυση των συστημάτων υποστήριξης πράκτορα, καθώς και οι υπηρεσίες οι οποίες παρέχουν γίνεται στην επόμενη ενότητα.



Σχήμα 3.3: Ένα παράδειγμα της λειτουργίας κινητών πρακτόρων

Στο σχήμα 3.3 παρουσιάζονται δυο μηχανήματα συνδεδεμένα σε ένα δίκτυο υπολογιστών. Σε κάθε μηχανήμα υπάρχει μια πλατφόρμα, η οποία είναι το περιβάλλον εκτέλεσης του πράκτορα. Οι πλατφόρμες παρέχουν μια σειρά από υπηρεσίες στους πράκτορες, που τους επιτρέπουν να μεταναστεύουν από ένα μηχανήμα στο άλλο, να επικοινωνούν μεταξύ τους, με το σύστημα που τους φιλοξενεί, κ.α.

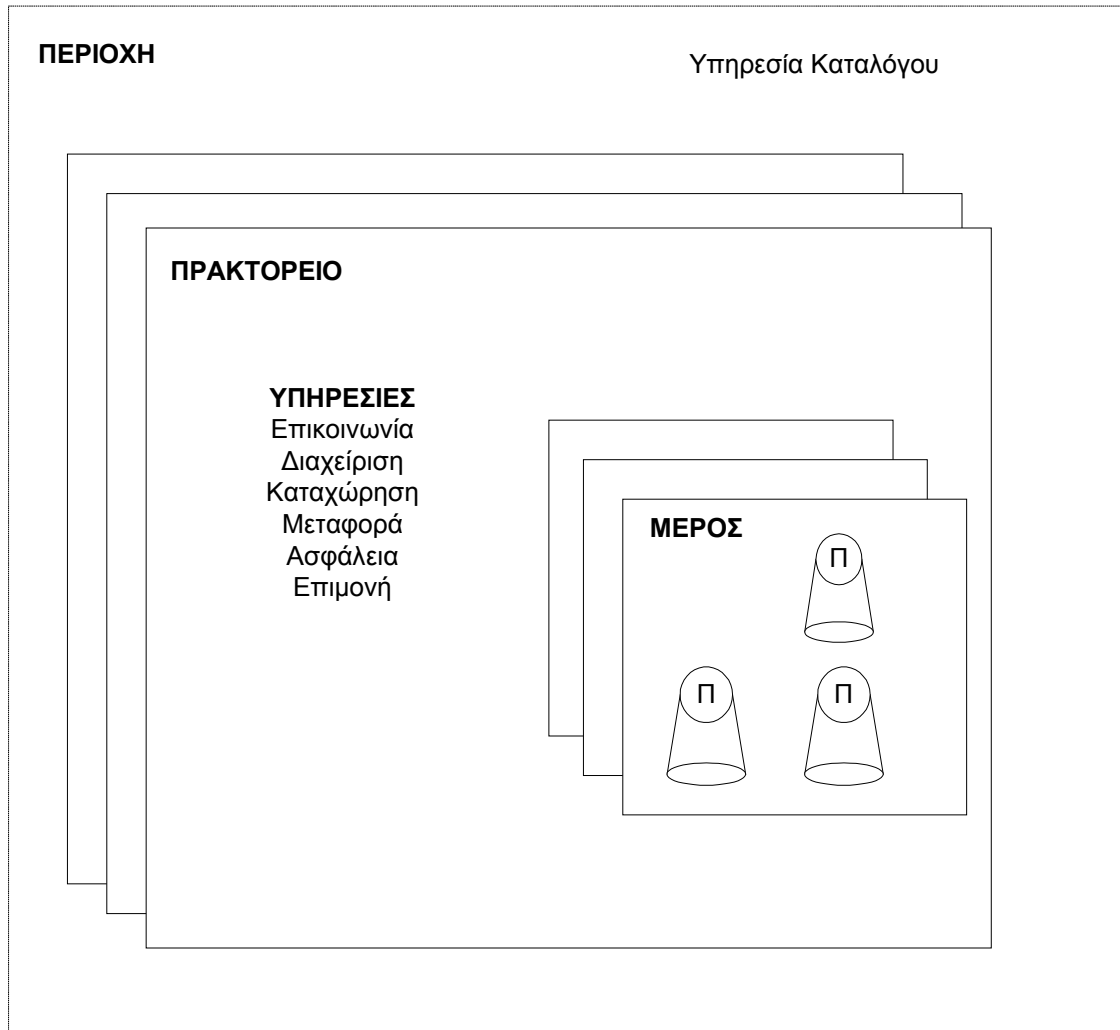
Στο μηχανήμα Β υπάρχει ένα πράκτορας Π-1, ο οποίος πρέπει να μεταναστεύσει στο μηχανήμα Α. Η πλατφορμα-2 επικοινωνεί με την πλατφορμα-2 και προετοιμάζουν την μετανάστευση του πράκτορα. Αφού ο πράκτορας μεταναστεύσει στον προορισμό του, έχει την δυνατότητα να αλληλεπιδράσει τοπικά με τον πράκτορα Π-2 ή και με το ίδιο το σύστημα του μηχανήματος.

3.4 Συστήματα υποστήριξης κινητών πρακτόρων - πλατφόρμες κινητών πρακτόρων

Η Τεχνολογία Κινητών Πρακτόρων (ΤΚΠ) είναι άρρηκτα συνδεδεμένη με την ύπαρξη συστημάτων υποστήριξης της λειτουργία τους. Αυτά τα συστήματα ονομάζονται Πλατφόρμες Κινητών Πρακτόρων ΠΚΠ (Mobile Agent Platforms, MAP). Ένας ΚΠ είναι αδύνατον να λειτουργήσει χωρίς την ύπαρξη ενός συστήματος, που θα παρέχει τις απαραίτητες υπηρεσίες για την εκτέλεση του κώδικα του.

Δυστυχώς, δεν υπάρχει κάποιο συγκεκριμένο πρότυπο βάσει του οποίου δημιουργούνται οι ΠΚΠ. Υπάρχουν όμως κάποια βασικά χαρακτηριστικά στην αρχιτεκτονική τους και κάποιες βασικές υπηρεσίες, οι οποίες πρέπει να παρέχονται από τα συστήματα αυτά, ώστε να είναι δυνατή η λειτουργία των πρακτόρων. Στο σχήμα 3.4 παρουσιάζεται η γενική αρχιτεκτονική μιας ΠΚΠ και οι υπηρεσίες οι οποίες υποστηρίζονται από αυτή. Πρέπει να τονίσουμε ότι η ορολογία που χρησιμοποιείται για τα δομικά στοιχεία της αρχιτεκτονικής και τις υπηρεσίες δεν είναι η ίδια σε όλες τις ΠΚΠ, αλλά η έννοιες όπως θα περιγράψουν απαντώνται στις περισσότερες από αυτές.

Το *πρακτορείο* είναι το βασικό δομικό στοιχείο μιας ΠΚΠ. Το στοιχείο αυτό παρέχει το περιβάλλον εκτέλεσης του πράκτορα, στο μηχάνημα που τον φιλοξενεί. Κάθε μηχάνημα ενός κατανεμημένου συστήματος ΚΠ πρέπει να διαθέτει τουλάχιστον ένα ενεργό πρακτορείο, ώστε να είναι δυνατή η εκτέλεση του πράκτορα. Ένα πρακτορείο μπορεί να περιέχει ένα ή περισσότερα *μέρη*. Τα μέρη δεν είναι τίποτα άλλο από μια λογική ομαδοποίηση των λειτουργιών, που παρέχονται από το συγκεκριμένο πρακτορείο. Παραδείγματος χάριν, μπορεί να υπάρχει ένα μέρος στο οποίο οι πράκτορες συναντιούνται και ανταλλάσσουν πληροφορίες. Συνήθως τα ονόματα που δίνονται στα μέρη, αντιστοιχούν στις λειτουργίες που αυτά παρέχουν. Τα πρακτορεία και τα μέρη τους είναι δυνατόν να περιέχονται σε μια περιοχή. Η ύπαρξη των περιοχών καθιστά δυνατή την διαχείριση των πρακτορειών και των μερών τους. Συνήθως οι περιοχές παρέχουν μια υπηρεσία



Σχήμα 3.4: Αρχιτεκτονική και υπηρεσίες μιας ΠΚΠ

κατάλογου ονομάτων για όλα τα πρακτορεία, τα μέρη αλλά και τους πράκτορες που εκτελούνται σε αυτά. Με πιο απλά λόγια, οι περιοχές διατηρούν πληροφορίες για τα υπόλοιπα στοιχεία που περιέχουν, διευκολύνοντας την επικοινωνία των στοιχείων αυτών μεταξύ τους.

3.4.1 Υπηρεσίες ΠΚΠ

Σε αυτήν την ενότητα παρουσιάζονται οι βασικές υπηρεσίες που πρέπει να παρέχει μια ΠΚΠ, για την υποστήριξη της λειτουργίας των πρακτόρων. Όπως παρουσιάζονται και στο Σχήμα 3.4, οι υπηρεσίες αυτές είναι οι παρακάτω:

- Υπηρεσία επικοινωνίας
- Υπηρεσία διαχείρισης

- Υπηρεσία καταχώρησης
- Υπηρεσία ασφάλειας
- Υπηρεσία μεταφοράς
- Υπηρεσία επιμονής
- Υπηρεσία καταλόγου

3.4.1.1 Υπηρεσία επικοινωνίας

Η υπηρεσία αυτή είναι υπεύθυνη για την επικοινωνία όλων των στοιχείων, που συνθέτουν ένα κατανεμημένο σύστημα ΚΠ. Είναι απαραίτητη για την μεταφορά των πρακτόρων κατά την μετανάστευση, την επικοινωνία μεταξύ πρακτόρων και την εγγραφή όλων των στοιχείων του συστήματος στην υπηρεσία κατάλογου.

Γενικά διακρίνονται δυο τύποι επικοινωνίας [Sloman87], οι οποίοι μπορούν να εφαρμοστούν και στην επικοινωνία μεταξύ πρακτόρων.

Σύγχρονη επικοινωνία: Σε αυτόν τον τύπο ένας διάυλος επικοινωνίας διατηρείται καθόλη την διάρκεια του διάλογου, μεταξύ των πρακτόρων

Ασύγχρονη επικοινωνία: Στην περίπτωση αυτή, ανταλλάσσονται μια σειρά από ανεξάρτητα μηνύματα και έτσι δεν υπάρχει η ανάγκη διατήρησης ενός σταθερού διαύλου επικοινωνίας.

Η βασική διάφορα μεταξύ των δυο τύπων, είναι ότι η σύγχρονη επικοινωνία είναι πολύ ευαίσθητη στα λάθη. Αν η σειρά των μηνυμάτων χαλάσει κατά κάποιο τρόπο, είναι πολύ δύσκολο να αποκατασταθεί η επικοινωνία. Στην ασύγχρονη επικοινωνία, η σειρά με την οποία λαμβάνουν τα μηνύματα οι πράκτορες δεν έχει επίδραση στην σωστή λειτουργία. Επιπλέον, με αυτόν τον τρόπο ένα πράκτορας μπορεί να επικοινωνεί ταυτόχρονα με πολλούς άλλους, αφού είναι σε θέση να επεξεργάζεται τα μηνύματα που λαμβάνει με οποιόν τρόπο και σειρά αυτός επιθυμεί.

Οι περισσότερες ΠΚΠ υποστηρίζουν περισσότερα από ένα πρωτόκολλα με τα οποία μπορούν να επικοινωνούν οι πράκτορες. Τα πιο συνηθισμένα είναι: απλές συνδέσεις (sockets), η Απομακρυσμένη Κλήση Μεθόδου (Remote Method Invocation RMI) και η αρχιτεκτονική της CORBA [CORBA].

3.4.1.2 Υπηρεσία διαχείρισης

Η υπηρεσίας διαχείρισης επιτρέπει στους χρηστές του συστήματος (ανθρώπους), να ελέγχουν και να παρακολουθούν την λειτουργία των πρακτόρων των πρακτορειών και των μερών. Οι κύριες λειτουργίες αυτής την υπηρεσίας είναι οι: δημιουργία, κατάργηση, κλωνοποίηση, αντιγραφή, αναστολή λειτουργίας και η επαναφορά λειτουργίας των πρακτόρων. Επιπλέον, παρέχει στην δυνατότητα παρακολούθησης των πρακτόρων και των μερών ενός πρακτορείου.

Κατά την διάρκεια λειτουργίας ενός πράκτορα, δηλαδή από την δημιουργία ως την κατάργηση, ο πράκτορας δεν είναι απαραίτητα σε ενεργή κατάσταση. Σε κάποιο σημείο της εκτέλεσης του πράκτορα ίσως κριθεί απαραίτητο από τον χρήστη, να διακόψει προσωρινά την εκτέλεση του. Όταν συμβεί αυτό, ο πράκτορας βρίσκεται σε κατάσταση αναστολής λειτουργίας (suspended). Ο χρήστης φυσικά είναι σε θέση να επαναφέρει τον πράκτορα στην κατάσταση που ήταν πριν την αναστολή. Η λειτουργία αυτή ονομάζεται επαναφορά (resume) πράκτορα. Η λειτουργία αναστολής δεν έχει καμία σχέση με την λειτουργία κατάργησης, που σημαίνει το τέλος λειτουργίας του πράκτορα ως ξεχωριστή διεργασία.

Δυο άλλες λειτουργίες που παρέχει η υπηρεσία διαχείρισης, είναι η αντιγραφή και κλωνοποίηση πράκτορα. Και με τις δυο αυτές λειτουργίες, δημιουργείται ένα ακριβές αντίγραφο του πράκτορα. Το ακριβές αυτό αντίγραφο δεν έχει μόνο την ίδια πληροφορία με τον πράκτορα αλλά και το ίδιο στάδιο εκτέλεσης. Η διάφορα μεταξύ των δυο αυτών λειτουργιών είναι ότι η κλωνοποίηση μπορεί να πραγματοποιηθεί μόνο στο ίδιο μέρος (πιο απλά στο ίδιο μηχάνημα), ενώ η αντιγραφή μπορεί να πραγματοποιηθεί και σε ένα απομακρυσμένο μέρος.

3.4.1.3 Υπηρεσία καταχώρησης

Κάθε πρακτορείο πρέπει να είναι σε θέση να γνωρίζει όλους τους πράκτορες και τα μέρη τα οποία περιέχει. Αυτό εξυπηρετεί τόσο του χρηστές του συστήματος, ώστε να μπορούν να επέμβουν εξωτερικά για την διαχείριση των πρακτόρων, αλλά και τους πράκτορες που επισκέπτονται το πρακτορείο και χρειάζονται πληροφορίες για να επικοινωνήσουν με άλλους πράκτορες, που βρίσκονται εκεί. Αυτή η υπηρεσία των πρακτορειών συνδέεται άμεσα με την υπηρεσία κατάλογου που παρέχουν οι περιοχές. Έτσι λοιπόν, κάθε νέος

πράκτορας που επισκέπτεται ένα πρακτορείο και κάθε νέο μέρος που δημιουργείται σε αυτό, καταχωρούνται.

3.4.1.4 Υπηρεσία μεταφοράς

Η υπηρεσία αυτή υποστηρίζει την διαδικασία μετανάστευσης πράκτορα από το ένα μηχάνημα στο άλλο. Μετά την μετανάστευση ο πράκτορας συνεχίζει την εκτέλεση του ακριβώς από το σημείο στο οποίο διακόπηκε. Η υπηρεσία μεταφοράς είναι υπεύθυνη για τον συντονισμό της μεταφοράς της πληροφορίας και του σταδίου εκτέλεσης του πράκτορα. Η πραγματική μεταφορά πραγματοποιείται από την υπηρεσία επικοινωνίας.

3.4.1.5 Υπηρεσία επιμονής

Η υπηρεσία επιμονής (persistence) δεν είναι μια από τις βασικές υπηρεσίες των ΠΚΠ, παρόλα αυτά παρέχεται από μερικές από αυτές. Επιτρέπει την αποθήκευση της πληροφορίας των πρακτόρων και των μερών ενός πρακτορείου, ώστε να είναι δυνατή η ανακατασκευή τους, σε περίπτωση που παρουσιαστεί κάποιο πρόβλημα στο σύστημα (π.χ. διακοπή ρεύματος). Η πληροφορίες αυτές αποθηκεύονται σε ένα αποθηκευτικό μέσο (αρχείο ή βάση δεδομένων) και ανακτώνται με την επαναλειτουργία του πρακτορείου. Φυσικά αποθηκεύεται και το στάδιο εκτέλεσης των πρακτόρων, ώστε η λειτουργία τους να συνεχιστεί από το σημείο που διακόπηκε.

3.4.1.6 Υπηρεσία καταλόγου

Όλα τα στοιχεία τα οποία συνθέτουν ένα κατανεμημένο σύστημα ΚΠ, οι πράκτορες, τα πρακτορεία, τα μέρη και οι περιοχές πρέπει να διαθέτουν ένα μοναδικό όνομα. Η μοναδικότητα του ονόματος καθιστά δυνατή την αναγνώριση τους. Η υπηρεσία κατάλογου είναι υπεύθυνη για την ανάθεση και αναγνώριση των μοναδικών αυτών ονομάτων. Η υπηρεσία καταλόγου περιοχής διατηρεί πληροφορίες για όλα τα στοιχεία που περιέχονται σε αυτή. Όταν ένα νέο πρακτορείο ή μέρος δημιουργείται, καταχωρείται αυτόματα από την υπηρεσία κατάλογου στη περιοχή που ανήκει. Τα πρακτορεία και τα μέρη είναι συνδεδεμένα με μια περιοχή για όλη τη διάρκεια ύπαρξής τους. Οι πράκτορες από την άλλη μεριά, είναι σε θέση να μετακινούνται από περιοχή σε περιοχή. Η υπηρεσία κατάλογου της κάθε περιοχής μπορεί να δώσει πληροφορίες για όλους τους πράκτορες που περιέχει, αφού και κάθε πράκτορας καταχωρείται αυτόματα στην νέα περιοχή μετά την μετακίνησή του. Επικοινωνώντας με την υπηρεσία κατάλογου τα

διάφορα στοιχεία του κατανεμημένου συστήματος αλλά και οι χρήστες, μπορούν να ανακαλύψουν την τρέχουσα θέση ενός πράκτορα με τον οποίο θέλουν να επικοινωνήσουν.

3.4.1.7 Υπηρεσία ασφάλειας

Το πρόβλημα της ασφάλειας είναι κοινό για όλα τα συστήματα υπολογιστών και σχετίζεται με την διατήρηση της μυστικότητας και της ακεραιότητας των δεδομένων ενός συστήματος, το οποίο εξυπηρετεί πολλούς χρήστες σε ένα δίκτυο υπολογιστών. Τα συστήματα υπολογιστών σήμερα παρέχουν μηχανισμούς που επιτρέπουν στους χρήστες να επέμβουν σε μέρη του συστήματος και δεδομένα στα οποία έχουν πρόσβαση, διασφαλίζοντας έτσι την ασφάλεια του. Στην περίπτωση όμως των κινητών πρακτόρων τα πράγματα είναι πιο πολύπλοκα.

Η ασφάλεια ενός συστήματος που επισκέπτεται ένας πράκτορας διασφαλίζεται μόνο αν η εκτέλεση του πράκτορα παρακολουθείται στενά και οι πρόσβαση σε τοπικούς πόρους είναι περιορισμένη. Αν δεν υπάρχει ο απαραίτητος βαθμός ασφάλειας, ένας πράκτορας μπορεί να αλλάξει δεδομένα στα οποία κανονικά δεν έχει πρόσβαση και να επέμβει στην εκτέλεση άλλων πρακτόρων και διεργασιών του συστήματος. Έτσι λοιπόν πριν γίνει αποδεκτή η εκτέλεση ενός πράκτορα σε ένα σύστημα, η υπηρεσία ασφάλειας διεξάγει μια σειρά από ελέγχους:

Πιστοποίηση (authentication): Με την πιστοποίηση ελέγχεται ότι ο πράκτορας έχει σταλεί από ένα “έμπιστο” μηχανήμα. Αυτό μπορεί να γίνει ζητώντας πληροφορίες πιστοποίησης από το μηχανήμα στο οποίο ο πράκτορας δημιουργήθηκε αρχικά, ή από το μηχανήμα στο οποίο ο πράκτορας βρισκόταν πριν την μετανάστευση, ή από κάποιο τρίτο μηχανήμα. Αν η πιστοποίηση του πράκτορα αποτύχει τότε ο πράκτορας δεν γίνεται δεκτός για εκτέλεση, ή στην καλύτερη περίπτωση η εκτέλεση του επιτρέπεται σε ένα πολύ περιορισμένο περιβάλλον.

Επαλήθευση (verification): Η επαλήθευση ελέγχει τον κώδικα το πράκτορα, ώστε να διασφαλίσει ότι δεν πρόκειται να εκτελέσει κάποια απαγορευμένη εργασία. Ωστόσο, κάποια μέρη του κώδικα δεν μπορούν να επαληθευτούν πριν την εκτέλεση τους, τέτοια είναι οι μεταβλητές που περνάνε σαν παράμετροι μιας μεθόδου. Για αυτό το λόγο, η διαδικασία της επαλήθευσης ελέγχει ότι ο πράκτορας δεν διαταράσσει το περιβάλλον

εκτέλεσης του. Αν το περιβάλλον εκτέλεσης παραμένει άθικτο, τότε ο πράκτορας δεν είναι σε θέση να εκτελέσει εργασίες πέρα από τις καθορισμένες. Μια τεχνική που ονομάζεται *proof-carrying code* [Necula97] επιτρέπει σε ένα μηχάνημα να καθορίσει αν ο κώδικας, από ένα “μη-έμπιστο” μηχάνημα, είναι ασφαλές να εκτελεστεί. Η τεχνική αυτή επισυνάπτει ένα ενδεικτικό ασφάλειας σε κάθε κομμάτι του προς μεταφορά κώδικα.

Εξουσιοδότηση (authentication): Η εξουσιοδότηση ελέγχει τις διεργασίες του πράκτορα κατά την εκτέλεση του. Στο στάδιο αυτό καταχωρούνται στον πράκτορα τα δικαιώματα πρόσβασης του στους πόρους και τα δεδομένα του συστήματος. Τα δικαιώματα αυτά καθορίζουν αν ο πράκτορας θα είναι σε θέση να διαβάσει, να γράφει και να τροποποιεί δεδομένα του συστήματος που τον φιλοξενεί.

Οι τρεις μηχανισμοί που αναφέραμε παραπάνω διασφαλίζουν, σε κάποιο βαθμό, την ασφάλεια του μηχανήματος που φιλοξενεί τον πράκτορα. Εκτός όμως από την ασφάλεια του μηχανήματος, υπάρχει και το θέμα της ασφάλειας των δεδομένων του πράκτορα, καθώς αυτός μετακινείται από το ένα μηχάνημα στο άλλο. Και για αυτήν την περίπτωση, υπάρχουν κάποιες τεχνικές ασφάλειας τις οποίες τα συστήματα κινητών πρακτόρων μπορούν να χρησιμοποιήσουν. Οι τεχνικές αυτές είναι:

Αλγόριθμοι κρυπτογράφησης: Αλγόριθμοι όπως ο PGP και άλλοι (Kaufman95) μπορούν να χρησιμοποιηθούν για να προστατεύσουν τα περιεχόμενα ενός κινητού πράκτορα κατά την μετακίνησή του.

Ψηφιακές υπογραφές: Με την τεχνική των ψηφιακών υπογράφων (Tanenbaum02), ο αποδέκτης ενός μηνύματος μπορεί να πιστοποιήσει την ταυτότητα του αποστολέα.

Η ασφάλεια είναι ένα πολύ μεγάλο θέμα και δεν σχετίζεται μόνο με την ΤΚΠ. Η εκτέλεση ενός κινητού πράκτορα δεν κρύβει περισσότερους κινδύνους για το μηχάνημα που τον εκτελεί από ότι ένα Applet της Java. Καθώς το διαδίκτυο εξελίσσεται, αναπτύσσονται νέοι μηχανισμοί ασφάλειας τόσο σε επίπεδο δικτύου όσο και σε επίπεδο εφαρμογών. Οι μηχανισμοί αυτοί σίγουρα θα συμπεριληφθούν στα συστήματα υποστήριξης πρακτόρων, παρέχοντας έναν μεγαλύτερο βαθμό ασφάλειας.

3.4.2 Τεχνολογία κινητών πρακτόρων και η γλώσσα προγραμματισμού Java

Μέχρι τώρα αναλύσαμε τα χαρακτηριστικά και τις υπηρεσίες που παρέχουν οι ΠΚΠ χωρίς να αναφερθούμε στους μηχανισμούς των προγραμματιστικών γλωσσών που υποστηρίζουν την υλοποίησή τους.

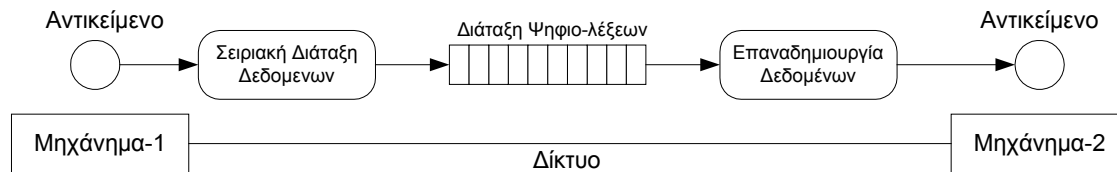
Η ανάπτυξη των ΠΚΠ και κατά συνέπεια της ΤΚΠ στηρίχθηκε κατά ένα μεγάλο μέρος στις διερμηνεύτηκες γλώσσες προγραμματισμού. Από όλες τις διερμηνεύτηκες γλώσσες, αυτή που είναι η καταλληλότερη για την δημιουργία ΚΠ είναι η Java [Gosling96]. Η Java είναι μια προγραμματιστική γλώσσα, διερμηνευτική, αντικειμενοστραφής και λόγω του ότι παρέχει ένα δικό της εικονικό περιβάλλον εκτέλεσης (Java Virtual Machine-JVM) [Lindholm99], είναι ανεξάρτητη από το λειτουργικό σύστημα του υπολογιστή που εκτελείται. Η ύπαρξη του εικονικού περιβάλλοντος εκτέλεσης επιτρέπει την εκτέλεση του πράκτορα σε ετερογενή συστήματα, γεγονός που ενισχύει τη χρήση της Java ως γλώσσα υλοποίησης ΚΠ. Υποστηρίζει σειριακή διάταξη των αντικειμένων (serialisation), χρησιμοποιώντας μια ανεξάρτητη από το λειτουργικό σύστημα αναπαράσταση τους. Το χαρακτηριστικό αυτό καθιστά δυνατή την μετάδοση ολόκληρων αντικειμένων, σε ένα δίκτυο υπολογιστών. Επιπλέον παρέχει μηχανισμούς ασφάλειας [Fritzing96], οι οποίοι μπορούν να χρησιμοποιηθούν για την ασφάλεια που πρέπει να παρέχεται από τις ΠΚΠ.

Τα παραπάνω χαρακτηριστικά, που δεν συναντώνται σε καμία άλλη γλώσσα προγραμματισμού, καθιστούν την Java ως την πλέον κατάλληλη για την δημιουργία ΚΠ. Έτσι εξηγείται και το γεγονός ότι οι συντριπτική πλειοψηφία των ΠΚΠ είναι υλοποιημένα σε αυτή την γλώσσα προγραμματισμού. Οι πλατφόρμες που συνήθως χρησιμοποιούνται είναι οι παρακάτω:

- Grasshopper της IKV ++ [Gras]
- Voyager της Objectspace [Voay]
- Concodia των Mitsubishi Electric Research Laboratories [Conc]
- Hive του MIT Media Lab [Hive]
- James της Siemens S.A [Jame]

3.4.2.1 Υποστήριξη της λειτουργίας κινητών πρακτόρων από μηχανισμούς της Java

Η κυριότερη λειτουργία ενός κινητού πράκτορα, που είναι η μετακίνηση του από μηχάνημα σε μηχάνημα, υποστηρίζεται στην Java από έναν μηχανισμό που ονομάζεται σειριακή διάταξη δεδομένων. Με τον μηχανισμό αυτό, ένα ενεργό αντικείμενο μετατρέπεται σε μια διάταξη από ψηφίο-λέξεις (bytes). Η διάταξη ψηφίο-λέξεων μπορεί να μεταδοθεί μέσω του δικτύου σε ένα άλλο μηχάνημα και να μετατραπεί εκεί στο αρχικό αντικείμενο. Η διαδικασία αυτή παρουσιάζεται στο σχήμα 3.5. Με τη σειριακή διάταξη μόνο τα δεδομένα του αντικειμένου μετατρέπονται σε ψηφίο-λέξεις και όχι το στάδιο εκτέλεσης του. Έτσι όταν το αντικείμενο επαναδημιουργηθεί στο απομακρυσμένο μηχάνημα, η εκτέλεση του (που είναι μια καινούργια διεργασία) πρέπει να ξεκινήσει από ένα προκαθορισμένο σημείο. Αυτό το σημείο μπορεί να είναι μια μέθοδος με ένα συγκεκριμένο όνομα.



Σχήμα 3.5: Μεταφορά αντικειμένου με μηχανισμούς της Java

Στην Java ο αρχικός κώδικας μια κλάσης μεταφράζεται αρχικά σε κώδικα ψηφίο-λέξεων, ο οποίος εκτελείται στην συνεχεία στο εικονικό περιβάλλον. Ο μεταφρασμένος κώδικας μπορεί να μεταφερθεί σε απομακρυσμένο μηχάνημα, όπως ένα αντικείμενο. Δηλαδή, όχι μόνο ενεργά αντικείμενα αλλά και ολόκληρες κλάσεις μπορούν να μεταφερθούν από ένα μηχάνημα στο άλλο. Η ικανότητα αυτή είναι πολύ χρήσιμη, γιατί έτσι ο κώδικας ενός πράκτορα δεν είναι απαραίτητο να υπάρχει εκ των προτέρων στο μηχάνημα προορισμού. Το περιβάλλον εκτέλεσης του πράκτορα (που είναι το ίδιο με το εικονικό περιβάλλον της Java) μπορεί να ζητήσει από το μηχάνημα που δημιουργήθηκε αρχικά ο πράκτορας, τον κώδικα της κλάσης του.

Η πρόσβαση σε αντικείμενα τα οποία εκτελούνται σε ένα απομακρυσμένο μηχάνημα μπορεί να επιτευχθεί με την λειτουργία απομακρυσμένης κλήσης μεθόδου (RMI), που υποστηρίζεται από την Java [JavaSoft98]. Έτσι μια μέθοδος του πράκτορα μπορεί να

κληθεί από το μηχάνημα στο οποίο αρχικά δημιουργήθηκε, ακόμα και αν αυτός έχει ήδη μεταναστεύσει σε κάποιο άλλο. Αυτός ο μηχανισμός της Java διευκολύνει την επικοινωνία των πρακτόρων με το σύστημα, αλλά και την επικοινωνία μεταξύ τους.

3.5 Πλεονεκτήματα και εφαρμογές της τεχνολογίας κινητών πρακτόρων

Το ενδιαφέρον για τους ΚΠ δεν παρακινείται από την τεχνολογία αυτή καθ' εαυτή, αλλά κυρίως από τα οφέλη της χρήσης της στην δημιουργία κατακεντρωμένων συστημάτων, σε σχέση με το κλασσικό μοντέλο πελάτη/εξυπηρετητή. Τα κυριότερα πλεονεκτήματα που προσφέρει η ΤΚΠ, όπως αυτά αναγνωρίστηκαν από διάφορες μελέτες [Lange99, Harrison95] είναι τα εξής:

- Μείωση του φορτίου του δικτύου
- Εκτός σύνδεσης εργασία
- Εύκολη αναβάθμιση λειτουργιών
- Παράλληλη επεξεργασία
- Μείωση των καθυστερήσεων λόγω δικτύου
- Λειτουργία σε ετερογενή περιβάλλοντα
- Εύκολη κλιμάκωση της εφαρμογής

Στην συνέχεια αναλύουμε τα πλεονεκτήματα αυτά και τον τρόπο με τον οποίο μπορούν να χρησιμοποιηθούν στις κατακεντρωμένες εφαρμογές, για να βελτιώσουν την λειτουργία τους.

Με το μοντέλο πελάτη/εξυπηρετητή εφαρμογές αναζήτησης και επεξεργασίας πληροφορίας είναι αναγκασμένες να συγκεντρώσουν μεγάλη ποσότητα ακατέργαστης πληροφορίας από απομακρυσμένα μηχανήματα, με σκοπό να την επεξεργαστούν και να αποδώσουν ένα συγκριτικά μικρό ποσοστό αποτελεσμάτων. Η μετακίνηση αυτή της ακατέργαστης πληροφορίας έχει ως αποτέλεσμα την αύξηση του φορτίου στο δίκτυο. Με την χρήση των ΚΠ δεν υπάρχει ανάγκη μεταφοράς της πληροφορίας. Οι πράκτορες επισκέπτονται τα απομακρυσμένα μηχανήματα και επεξεργάζονται την πληροφορία τοπικά. Μόνο τα αποτελέσματα της επεξεργασίας μεταφέρονται από το δίκτυο. Με αυτόν τον τρόπο μειώνεται το απαιτούμενο εύρος ζώνης. Το χαρακτηριστικό αυτό είναι

πολύ χρήσιμο σε εφαρμογές παρακολούθησης δεδομένων και γεγονότων. Έτσι, αν μια εφαρμογή παρακολουθεί τις τιμές κάποιων συγκεκριμένων μεταβλητών, ώστε αυτές να μην ξεπερνούν μια ανωτάτη ή κατώτατη τιμή, δεν χρειάζεται να μεταφέρονται περιοδικά όλα τα δεδομένα από το απομακρυσμένο μηχάνημα. Ένας πράκτορας μπορεί να ελέγχει την τιμή των μεταβλητών και να ειδοποιεί την εφαρμογή, όταν αυτές περάσουν ένα ανώτερο ή κατώτερο όριο.

Μερικές εφαρμογές απαιτούν επαναλαμβανόμενες αλληλεπιδράσεις μεταξύ απομακρυσμένων μηχανημάτων. Αυτό έχει ως αποτέλεσμα την διατήρηση μια μόνιμης σύνδεσης μεταξύ τους. Με τους ΚΠ η διατήρηση μόνιμης σύνδεσης δεν είναι απαραίτητη. Μετά την αποστολή του πράκτορα η σύνδεση μπορεί να διακοπεί, αφού ο πράκτορας μπορεί αυτόνομα και ασύγχρονα να ολοκληρώσει την διεργασία στο απομακρυσμένο μηχάνημα. Η παράδοση των αποτελεσμάτων της διεργασίας γίνεται μετά την επανασύνδεση των μηχανημάτων. Αυτό το χαρακτηριστικό είναι πολύ χρήσιμο όταν χρησιμοποιούνται φορητοί υπολογιστές (Laptops), οι οποίοι συνήθως είναι αναγκασμένοι να χρησιμοποιήσουν ένα μικρό εύρος ζώνης πάνω από αναξιόπιστες συνδέσεις.

Στις εφαρμογές πελάτη/εξυπηρετητή, οι εξυπηρετητές συνήθως παρέχουν μια σειρά από καθορισμένες λειτουργίες τις οποίες μπορούν να χρησιμοποιήσουν οι πελάτες, για την εκτέλεση μιας διεργασίας. Οι απαιτήσεις όμως των πελατών μπορεί να αλλάξουν και να μην καλύπτονται από τις παρεχόμενες λειτουργίες. Σε αυτή την περίπτωση οι λειτουργίες (το πρόγραμμα) του εξυπηρετητή θα έπρεπε να αναβαθμιστούν. Με χρήση των ΚΠ οι λειτουργίες, που εκτελούνται σε ένα απομακρυσμένο μηχάνημα, μεταφέρονται από τον ίδιο τον πράκτορα. Έτσι λοιπόν, για να αναβαθμιστούν οι λειτουργίες ενός κατανεμημένου συστήματος, αρκεί να αναβαθμιστεί ο κώδικας του πράκτορα.

Οι ΚΠ παρέχουν επίσης την δυνατότητα παράλληλης επεξεργασίας. Μπορούν να εκτελούνται ταυτόχρονα σε διάφορα μηχανήματα ενός κατανεμημένου συστήματος, για την περάτωση μιας διεργασίας. Η παράλληλη επεξεργασία έχει ως αποτέλεσμα την βελτίωση της απόδοσης της εφαρμογής.

Σε εφαρμογές πραγματικού χρόνου, για παράδειγμα στον απομακρυσμένο έλεγχος μια συσκευής σε μια γραμμή παραγωγής, η καθυστέρηση του δικτύου είναι ένας

απρόβλεπτος παράγοντας που μπορεί να επηρεάσει την ορθή λειτουργία. Με την χρήση ΚΠ έναν πράκτορα αποστέλλεται στην απομακρυσμένη συσκευή. Έτσι ο έλεγχος της γίνεται τοπικά χωρίς την αποστολή μηνυμάτων, που μπορεί να καθυστερήσουν στο δίκτυο.

Μια κατανεμημένη εφαρμογή συνήθως εκτελείται σε ετερογενή περιβάλλοντα, τόσο από άποψη συσκευών όσο και από άποψη λειτουργικών συστημάτων. Οι ΚΠ είναι ανεξάρτητοι από το λειτουργικό σύστημα του μηχανήματος στο οποίο εκτελούνται. Εξαρτώνται μόνο από περιβάλλον υποστήριξης του πράκτορα, που πρέπει να είναι εγκατεστημένο σε κάθε μηχανήμα του κατανεμημένου συστήματος.

Τα χαρακτηριστικά της ΤΚΠ όπως η τοπική επεξεργασία, η παράλληλη επεξεργασία, η εύκολη αναβάθμιση, επιτρέπουν την καλύτερη κλιμάκωση μιας κατανεμημένης εφαρμογής. Καθώς τα δίκτυα υπολογιστών γίνονται ολοένα και μεγαλύτερα, η δυνατότητα εύκολης κλιμάκωσης από ένα μικρότερο σε ένα μεγαλύτερο δίκτυο, είναι ένα από τα σπουδαιότερα πλεονεκτήματα της ΤΚΠ.

3.6 Μειονεκτήματα της τεχνολογία κινητών πρακτόρων

Στην προηγούμενη ενότητα παρουσιάσαμε τα πλεονεκτήματα της ΤΚΠ πρακτόρων σε σχέση με το μοντέλο πελάτη/εξυπηρετητή. Φυσικά αυτή η σχετικά νέα τεχνολογία δεν παρουσιάζει μόνο πλεονεκτήματα, αλλά και μειονεκτήματα.

Ένα χαρακτηριστικό της ΤΚΠ που μπορεί να θεωρηθεί ως μειονέκτημα είναι ότι για την λειτουργία της απαιτείται η ύπαρξη του περιβάλλοντος εκτέλεσης του πράκτορα, σε κάθε μηχανήμα του κατανεμημένου συστήματος. Όπως ήδη αναφέραμε, δεν υπάρχει ένα κοινό πρότυπο βάσει του οποίου να κατασκευάζονται οι ΠΚΠ. Έτσι πράκτορες οι οποίοι έχουν δημιουργηθεί σε μια πλατφόρμα δεν μπορούν να εκτελεστούν από μια άλλη. Ούτε φυσικά μπορούν να αλληλεπιδράσουν με πράκτορες και στοιχεία της. Το πρόβλημα αυτό εντοπίστηκε σχετικά νωρίς και έτσι προτάθηκε από την OMG (που είναι ο πιο σημαντικός οργανισμός στην δημιουργία πρότυπων, στην περιοχή των κινητών πρακτόρων) ένα πρότυπο που θα επέτρεπε την αλληλεπίδραση μεταξύ διαφορετικών ΠΚΠ. Το πρότυπο αυτό ονομάζεται Mobile Agent System Interoperability Facility (MASIF) [MASIF]. Δυστυχώς, το πρότυπο αυτό δεν έχει ακόμα υιοθετηθεί από αρκετές από τις υπάρχουσες ΠΚΠ, ώστε να δοθεί μια λύση στο πρόβλημα της αλληλεπίδρασης.

Το παράδειγμα των ΚΠ παρουσιάζει προβλήματα κυρίως στον τομέα της ασφάλειας. Τα μηχανήματα που φιλοξενούν ΚΠ είναι εκτεθειμένα στον κίνδυνο να δεχτούν έναν πράκτορα ο οποίος θα υποκλέψει, ή θα καταστρέψει ευαίσθητα δεδομένα του συστήματος, διαταράσσοντας έτσι την ομαλή λειτουργία. Ακόμη, ένας πράκτορας μπορεί θεμιτά ή αθέμιτα (λόγο λάθος προγραμματισμού) να δεσμεύσει πόρους του συστήματος και του δικτύου, μην επιτρέποντας σε άλλες διεργασίες να εκτελεστούν. Αλλά και οι ίδιοι οι πράκτορες πρέπει να προστατεύσουν την πληροφορία που μεταφέρουν, από το σύστημα το οποίο επισκέπτονται. Ήδη έχουμε αναφέρει μηχανισμούς για την προστασία τόσο των επισκεπτόμενων συστημάτων όσο και των πρακτόρων. Όμως, το πρόβλημα της ασφάλειας παραμένει ένα μεγάλο θέμα τόσο για την ΤΚΠ όσο και για τις υπόλοιπες δικτυακές εφαρμογές. Όσο περισσότερους και εγκυρότερους μηχανισμούς ασφάλειας προσφέρουν οι ΠΚΠ, τόσο περισσότερο θα εμπιστεύονται οι χρηστές εφαρμογές στηριγμένες στην ΤΚΠ.

3.7 Κινητοί πράκτορες στα πλαίσια της διατριβής

Στα πλαίσια της διατριβής, η ΤΚΠ αντιμετωπίζεται τόσο ως εργαλείο δημιουργίας κατανεμημένων εφαρμογών όσο και ως αντικείμενο μελέτης. Με χρήση της ΤΚΠ θα δημιουργηθούν μια σειρά από εφαρμογές, με σκοπό την διαχείριση και παρακολούθηση ενός IP δικτύου. Πέρα από την χρήση της ΤΚΠ για την δημιουργία εφαρμογών, προτείνονται και αξιολογούνται μια σειρά από μεθόδους, για την βελτίωση της απόδοσης της σε συστήματα διαχείρισης δικτύου.

Όλοι πράκτορες εκτελούνται σε πλατφόρμες υλοποιημένες στην γλώσσα προγραμματίσμου Java, κατά επέκταση και οι πράκτορες είναι υλοποιημένοι σε αυτήν την γλώσσα. Οι δυο πλατφόρμες που χρησιμοποιούνται για την δημιουργία των πρακτόρων είναι η Grasshopper και η Voyager. Και οι δυο ακολουθούν σε μεγάλο βαθμό το μοντέλο πλατφόρμας το οποίο παρουσιάστηκε σε αυτό το κεφάλαιο. Από τις δυο ΠΚΠ, μόνο η Grasshopper υποστηρίζει το πρότυπο MASIF. Έτσι, δεν είναι δυνατή η αλληλεπίδραση μεταξύ των στοιχείων τους. Για να μην υπάρχει όμως ο περιορισμός εκτέλεσης του κώδικα του πράκτορα σε μια μόνο πλατφόρμα, αναπτύχθηκε στα πλαίσια της διατριβής, μια Διεπαφή Προγραμματισμού Εφαρμογής ΔΠΕ (Application Programming Interface, API). Η ΔΠΕ επιτρέπει την εκτέλεση του πράκτορα και στις δυο πλατφόρμες, με ελάχιστες αλλαγές στον κώδικα του. Ο τρόπος με τον οποίο επιτυγχάνεται αυτό θα παρουσιαστεί αναλυτικά στο επόμενο κεφάλαιο.

Οι ΚΠ είναι υλοποιημένοι με την μέθοδο της ισχυρής μετανάστευσης, στην οποία μαζί με τα δεδομένα μεταφέρεται και το στάδιο εκτέλεσης τους. Στις εφαρμογές που θα παρουσιαστούν στα επόμενα κεφαλαία, οι ΚΠ επισκέπτονται περισσότερα από ένα μηχανήματα, για να ολοκληρώσουν την εργασία που τους έχει ανατεθεί. Για αυτόν τον τρόπο λειτουργίας, το στάδιο εκτέλεσης του πράκτορα είναι απαραίτητη πληροφορία.

Η διατριβή δεν ασχολείται σε μεγάλο βαθμό με θέματα ασφάλειας. Η ασφάλεια του συστήματος διασφαλίζεται με την ύπαρξη στατικών πρακτόρων. Οι στατικοί πράκτορες δεν μετακινούνται κατά την διάρκεια εκτέλεσης τους. Στα πλαίσια της διατριβής, οι ΚΠ δεν αλληλεπιδρούν απευθείας με το σύστημα το οποίο επισκέπτονται αλλά μέσω στατικών πρακτόρων οι οποίο εκτελούνται σε αυτό. Οι στατικοί πράκτορες δρουν ως διεπαφές, επιτρέποντας στους ΚΠ τους να έχουν πρόσβαση στις λειτουργίες και τα

δεδομένα του επισκεπτόμενου συστήματος. Η μη άμεση αλληλεπίδραση των ΚΠ με το επισκεπτόμενο σύστημα, μειώνει τα προβλήματα ασφάλειας. Επιπλέον, η χρήση των στατικών πρακτόρων οδηγεί σε μείωση του μεγέθους των ΚΠ. Οι πράκτορες δεν είναι αναγκασμένοι να μεταφέρουν τις μεθόδους αλληλεπίδρασης με το σύστημα, αφού αυτές προσφέρονται από τους στατικούς πράκτορες.

3.8 Σύνοψη κεφαλαίου

Στο κεφαλαίο αυτό παρουσιάστηκαν τα βασικά στοιχεία της ΤΚΠ. Αρχικά δόθηκε ένας ορισμός της έννοια των λογισμικών πρακτόρων και μια κατηγοριοποίηση τους σε έναν τρισδιάστατο χώρο, στον οποίο η κάθε διάσταση είναι μια από τις ικανότητες των λογισμικών πρακτόρων (νοημοσύνη, τρόπος, δράσης, κινητικότητα).

Ιδιαίτερο ενδιαφέρον δόθηκε σε μια κατηγορία λογισμικών πρακτόρων που ονομάζονται κινητοί πράκτορες, λόγω του ότι έχουν την ικανότητα να κινούνται από ένα το μηχάνημα στο άλλο, σε ένα δίκτυο υπολογιστών. Παρουσιάστηκε η εξέλιξη του τρόπου λειτουργίας των κατανεμημένων συστημάτων και πως η τεχνολογία οδηγήθηκε από την μέθοδο της απλής ανταλλαγής μηνυμάτων στους ΚΠ. Με αλλά λόγια περιγράφηκε πως από το κλασσικό μοντέλο πελάτη/εξυπηρετητή προέκυψε η ανάγκη για την δημιουργία της ΤΚΠ. Διακρίνονται δυο τρόποι με τους οποίους ένας πράκτορας μπορεί να μετακινηθεί σε ένα δίκτυο υπολογιστών, η ισχυρή μετανάστευση και η ασθενής μετανάστευση.

Στη συνέχεια παρουσιάστηκε η αρχιτεκτονική ενός συστήματος υποστήριξης κινητών πρακτόρων (Πλατφόρμα Κινητών Πρακτόρων-ΠΚΠ) και οι βασικές υπηρεσίες που πρέπει να παρέχονται, ώστε να είναι δυνατή η λειτουργία των πρακτόρων. Οι υπάρχοντες ΠΚΠ δεν ακολουθούν επακριβώς την αρχιτεκτονική η οποία παρουσιάστηκε, λόγω το ότι δεν υπάρχει ένα κοινό πρότυπο για την υλοποίησή τους. Όμως, η αρχιτεκτονική αυτή καλύπτει σε μεγάλο μέρος τον τρόπο λειτουργίας και τις υπηρεσίες που παρέχουν τα περισσότερες από αυτές.

Οι περισσότερες ΠΚΠ είναι υλοποιημένες στην γλώσσα προγραμματισμού Java. Έτσι, στην συνέχεια παρουσιάστηκαν τα χαρακτηριστικά και οι μηχανισμοί που παρέχονται από την Java και την καθιστούν την πλέον κατάλληλη για την ανάπτυξη ΚΠ.

Οι ΚΠ χρησιμοποιούνται για την δημιουργία κατανεμημένων εφαρμογών διότι παρουσιάζουν μια σειρά από πλεονεκτήματα, σε σχέση με το μοντέλο πελάτη/εξυπηρετητή. Μαζί με τα πλεονεκτήματα παρουσιάζεται και ο τρόπος με τον οποίο επωφελούνται οι κατανεμημένες εφαρμογές από την χρήση τους. Υπάρχουν φυσικά και κάποια μειονεκτήματα τα οποία παρουσιάζονται επίσης.

Τέλος παρουσιάζεται ο τρόπος με τον οποίο αντιμετωπίζονται οι ΚΠ στα πλαίσια της διατριβής και γίνεται μια αναφορά στις πλατφόρμες οι οποίες θα χρησιμοποιηθούν για την υλοποίησή τους.

3.9 Βιβλιογραφικές αναφορές

- [Gilbert95] D. Gilbert, et. al., “IBM Intelligent Agent Strategy”, *Technical Report, IBM Corporation*, Research Triangle Park, NC, USA, 1995.
- [Shoam94] Y. Shoam, and B. Thomas, “Agent-Oriented Programming”, *Encyclopedia of Computer Science and Technology*, Vol. 29, 1994, pp. 1-17.
- [Chaib-Draa94] B. Chaib-Draa, “Distributed Artificial Intelligence: An Overview”. *Encyclopedia of Computer Science and Technology*, Vol. 29, 1994, pp. 215-243.
- [Stamos90A] J. Stamos, and D. K. Gifford, “Remote Evaluation”, *ACM Trans. Programming Languages and Systems*, Vol. 12, No. 4, 1990, pp. 537-565.
- [Stamos90B] J. Stamos, and D. K. Gifford, “Implementing Remote Evaluation” *IEEE Trans. Software Engineering*, Vol. 16, No. 7, 1990, pp. 710-722.
- [Smith88] J. M. Smith, “A Survey of Process Migration Mechanisms”, *ACM Operating System Review*, Vol. 22, No. 3, 1988, pp. 28-40.
- [Shneiderman97] B. Shneiderman, “Direct Manipulation Versus Agents: Paths to Predictable, Controllable, and Comprehensible Interfaces,” *Software Agents*, J. M. Bradshaw, Editor, MIT Press, Cambridge, MA (1997), pp. 97-106.
- [OMG01] Object Management Group, “Mobile Agent Facility Specification”, 28 October 2001.
- [Sloman87] M. Sloman and J. Krammer , “distributed system and computer network”, *Prentice Hall International(UK)* , 1987, ISBN:0132158647.
- [Necula97] G. C. Vecula, “Proof-Carrying Code”, *Proc. ACM Symposium on the Principles of programming Languages*, January 1997, pp. 106-119.
- [CORBA] Common Object Request Broker Architecture, Πληροφορία διαθέσιμη στην ηλεκτρονική διεύθυνση: <http://www.omg.org/gettingstarted/corbafaq.htm>.
- [Kaufman95] C. Kaufman, R. Perlman, and M. Speciner, “Network Security: Private Communication in a Public World”, *Prentice-Hall PTR*, 1995, ISBN: 0130614661.

- [Tanenbaum02] A. S. Tanenbaum, “Computer Networks”, *Prentice Hall PTR*; 4th edition, 2002, ISBN: 0130661023.
- [Gosling00] J. Gosling, B. Joy and G. Steele, “The Java Language Speciation”, *Addison-Wesley Pub Co*, 2nd edition, 2000, ISBN: 0201310082.
- [Fritzinger96] J. S. Fritzinger and M. Mueller, “Java Security Technical report”, *Sun Microsystems Inc*, 1996, Διαθέσιμο στην ηλεκτρονική διεύθυνση: <http://java.sun.com/security/whitepaper.ps>.
- [Lindholm99] T. Lindholm and F. Yellin “The Java Virtual Machine Specification”, *Addison-Wesley Pub Co*, 2nd edition, 1999, ISBN: 0201432943.
- [JavaSoft98] JavaSoft, “Java RMI Specification”, 1998, Διαθέσιμο στην ηλεκτρονική διεύθυνση: <http://java.sun.com/products/jdk/1.2/docs/guide/rmi/spec/rmiTOC.doc.html>,
- [Gras] Η πληροφορία για την πλατφόρμα Grasshopper είναι διαθέσιμη στην ηλεκτρονική διεύθυνση: <http://www.grasshopper.de>.
- [Voya] Η πληροφορία για την πλατφόρμα Voyager είναι διαθέσιμη στην ηλεκτρονική διεύθυνση: <http://www.recursionsw.com/products/voyager/voyager.asp>.
- [Conc] Η πληροφορία για την πλατφόρμα Concordia είναι διαθέσιμη στην ηλεκτρονική διεύθυνση: <http://www.merl.com/projects/concordia/>.
- [Hive] Η πληροφορία για την πλατφόρμα Hive είναι διαθέσιμη στην ηλεκτρονική διεύθυνση: <http://xenia.media.mit.edu/~nelson/research/hive-asama99/asama-html/>.
- [Jame] Η πληροφορία για την πλατφόρμα James είναι διαθέσιμη στην ηλεκτρονική διεύθυνση: <http://james.dei.uc.pt/>
- [Lange99] D. Lange and M. Oshima, “Seven Good Reasons for Mobile Agents”, *Communications of the ACM*, Vol.42, No.3, pp. 88-89, March 1999.
- [Harrison95] C. G. Harrison, D. M. Chess and A. Kershenbaum, “Mobile Agents, Are they a good idea”, *Technical report, IBM Research Division, T.J.Watson Research Center*, March 1995.

[MASIF] D. Milojicic et al, “The OMG Mobile Agent System Interoperability Facility”,
OMG Document, Διαθέσιμο στην ηλεκτρονική διεύθυνση:
http://www.hpl.hp.com/personal/Dejan_Milojicic/ma4.pdf

ΚΕΦΑΛΑΙΟ 4

ΕΦΑΡΜΟΓΗ ΤΗΣ ΤΕΧΝΟΛΟΓΙΑΣ ΚΙΝΗΤΩΝ ΠΡΑΚΤΩΡΩΝ ΣΕ ΣΥΣΤΗΜΑΤΑ ΠΑΡΟΧΗΣ ΠΟΙΟΤΗΤΑΣ ΥΠΗΡΕΣΙΑΣ ΣΤΟ ΔΙΑΔΙΚΤΥΟ

4.1 Εισαγωγή

Τα δομή των τηλεπικοινωνιακών δικτύων γίνεται διαρκώς περισσότερο κατακεντρωμένη και ετερογενής, με διάφορες τεχνολογίες ενοποιημένες στο ίδιο δίκτυο (π.χ. ATM, IP, ADSL) και διάφορα δικτυακά εξαρτήματα, συνήθως από διαφορετικούς προμηθευτές. Ταυτόχρονα, η απαίτηση την παροχή ΠοιΥπ από τα δίκτυα επικοινωνιών γίνεται ολοένα και εντονότερη. Με αυτά τα δεδομένα, η αποτελεσματική διαχείριση αποκτά μεγαλύτερη σημασία. Τα σύγχρονα συστήματα διαχείρισης πρέπει να είναι αρκετά ευέλικτα και ευφυή, ώστε δυναμικά να διαρθρώνουν και να παρακολουθούν το υπό διαχείριση δίκτυο. Θα πρέπει να είναι σε θέση από την μια μεριά να πληροφορούν τους χρηστές για την ΠοιΥπ που τους παρέχεται και από την άλλη να επιτρέπουν στους διαχειριστές να διαχειρίζονται κατάλληλα τους πόρους του δικτύου.

Στο δεύτερο κεφάλαιο έγινε μια επισκόπηση των λειτουργιών που πρέπει να υποστηρίζονται από ένα δίκτυο για την παροχή ΠοιΥπ, καθώς και διάφορες προτεινόμενες αρχιτεκτονικές που μπορούν να χρησιμοποιηθούν για το σκοπό αυτό. Στο προηγούμενο κεφάλαιο παρουσιάστηκε αναλυτικά η ΤΚΠ. Τα δυο αυτά κεφαλαία περιέχουν το απαραίτητο θεωρητικό υπόβαθρο για την κατανόηση του συστήματος διαχείρισης δικτύου, που παρουσιάζεται σε αυτό το κεφάλαιο. Το σύστημα είναι σχεδιασμένο με βάση την αρχιτεκτονική ΔΥ και υλοποιημένο με χρήση της ΤΚΠ. Συγκεκριμένα το σύστημα παρέχει τις παρακάτω λειτουργίες:

- Διάρθρωση ενός δικτύου αποτελούμενου από ετερογενείς δρομολογητές για την παροχή ΠοιΥπ.
- Παρακολούθηση της παρεχόμενης ΠοιΥπ.

Η αρχιτεκτονική ΔΥ παρέχει ένα μεγαλύτερο βαθμό ευελιξίας στη διαχείριση ενός δικτύου, σε σχέση με τις άλλες προτεινόμενες αρχιτεκτονικές. Για τον λόγο αυτό έχει κερδίσει την γενική αποδοχή της κοινότητας του διαδικτύου. Η ΤΚΠ είναι μια σχετικά νέα τεχνολογία υλοποίησης καταναμεμημένων εφαρμογών. Οι λόγοι για τους οποίους τα συστήματα διαχειρίσεις δικτύου υιοθέτησαν την τεχνολογία αυτή, αναφέρονται στην επόμενη ενότητα.

Η αξιολόγηση της προτεινόμενου συστήματος διαχείρισης πραγματοποιείται βάσει πραγματικών μετρήσεων. Για το σκοπό αυτό, δημιουργήθηκε ένα πειραματικό IP δίκτυο ΔΥ.

Η εφαρμογή διαχείρισης που παρουσιάζεται, σχεδιάστηκε και υλοποιήθηκε στα πλαίσια του Ευρωπαϊκού ερευνητού προγράμματος MANTRIP-IST-10921.

4.2 Τεχνολογία κινητών πρακτόρων και συστήματα διαχείρισης δικτύου

Σήμερα, η διαχείριση δικτύων πραγματοποιείται με χρήση μιας σειράς πρωτοκόλλων, σχεδιασμένα για το σκοπό αυτό. Τα πιο γνωστά από αυτά είναι το Simple Network Management Protocol (SNMP) της IETF [Case90] και το Common Management Information System/Common Management Information Protocol CMIS/CMIP της ISO [CMIP]. Και τα δυο πρωτόκολλα ακολουθούν το κεντροποιημένο μοντέλο πελάτη/εξυπηρετητή. Σύμφωνα με το μοντέλο αυτό, οι πελάτες είναι τα υπό διαχείριση στοιχεία του δικτύου και ο εξυπηρετητής είναι το σύστημα διαχείρισης. Σε κάθε δικτυακό στοιχείο (δρομολογητές, μηχανήματα, κ.α) εκτελείται μια λογισμική εφαρμογή, που καλείται αντιπρόσωπος διαχείρισης. Ο αντιπρόσωπος είναι μια διεπαφή σε μια βάση διαχείρισης της πληροφορίας, του υπό διαχείριση στοιχείου. Η βάση αυτή ονομάζεται Management Information Base (MIB). Ο αντιπρόσωπος διαχείρισης είναι υπεύθυνος για την εκτέλεση μια σειρά διαχειριστικών εντολών, στο δικτυακό στοιχείο που βρίσκεται. Η διαχείριση του δικτύου πραγματοποιείται με ανταλλαγή μηνυμάτων μεταξύ του συστήματος διαχείρισης και των αντιπρόσωπων. Το κεντροποιημένο μοντέλο διαχείρισης δικτύου παρουσιάζεται στο σχήμα 4.1. Με την αύξηση του μεγέθους των δικτύων, το μοντέλο αυτό παρουσίασε μια σειρά από προβλήματα [Martin-Flatin99, Goldszmidt93]:

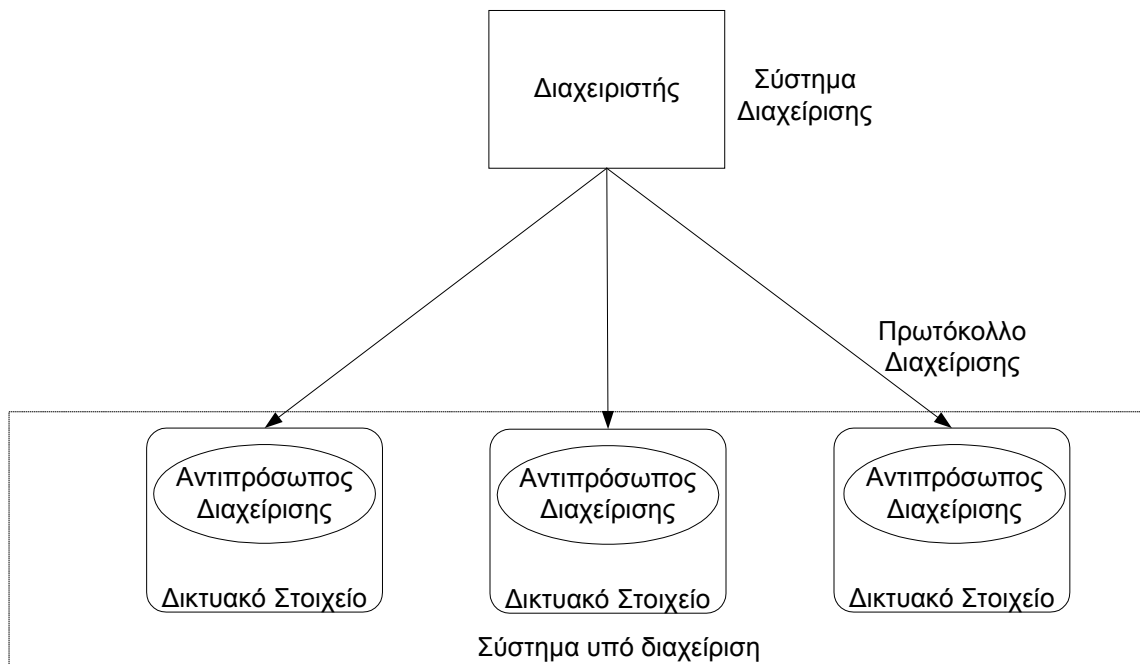
Η αναβάθμιση των υπηρεσιών που προσφέρει το σύστημα είναι πολύ δύσκολη: Το σύστημα διαχείρισης είναι σχεδιασμένο να προσφέρει προκαθορισμένες υπηρεσίες. Για να αλλάξουν οι υπάρχουσες, ή για να προστεθούν νέες πρέπει να αναβαθμιστούν όλα τα στοιχεία του συστήματος (διαχειριστής και αντιπρόσωποι).

Χρησιμοποιεί του πόρους του συστήματος αναποτελεσματικά: όλες οι υπολογιστικές διεργασίες πραγματοποιούνται από έναν κεντρικό διαχειριστή του συστήματος. Η νέα τάση στα δίκτυα υπολογιστών, είναι η μεταφορά της υπολογιστικής δύναμης στα δεδομένα και όχι αντίστροφα. Αυτή η λογική οδηγεί σε περισσότερο κατανεμημένα συστήματα.

Εξαρτάται από την διαθεσιμότητα του δικτύου: Η διαχείριση του δικτύου είναι αδύνατη χωρίς την ύπαρξη σταθερής επικοινωνίας μεταξύ του διαχειριστή και των αντιπρόσωπων.

Αυξάνει την κίνηση στο δίκτυο: Η συνεχής συλλογή πληροφοριών από τους αντιπρόσωπους αυξάνει την κίνηση στο δίκτυο. Η αύξηση αυτή είναι ιδιαίτερα σημαντική, καθώς το μέγεθος του υπό διαχείριση δικτύου αυξάνει.

Παρουσιάζει δυσκολία κλιμάκωσης: Η κλιμάκωση του μοντέλου αυτού είναι δύσκολη σε μεγαλύτερα δίκτυα, λόγω της αναποτελεσματικής χρήσης των υπολογιστικών πόρων και των πόρων του δικτύου.



Σχήμα 4.1: Κεντροποιημένη διαχείριση δικτύου

Τα προβλήματα που παρουσίασε το κεντροποιημένο μοντέλο διαχείρισης οδήγησε στην αναζήτηση εναλλακτικών λύσεων. Η ΤΚΠ είναι μια πολλά υποσχόμενη λύση στην διαχείριση δικτύων. Οι ΚΠ μπορούν αυτόνομα να μετακινηθούν από το ένα δικτυακό στοιχείο στο άλλο, ώστε να συλλέξουν και να επεξεργαστούν την πληροφορία τοπικά. Η χρήση των ΚΠ δεν περιορίζεται στην συλλογή και επεξεργασία πληροφοριών, αλλά επεκτείνεται και σε λειτουργίες διόρθωσης του δικτύου. Με την χρήση τους μπορούν να προκύψουν συστήματα διαχείρισης τα οποία θα είναι περισσότερο καταναμημένα και προσαρμόσιμα στις τρέχουσες απαιτήσεις. Στο προηγούμενο κεφαλαίο έγινε μια εκτενής

αναφορά των πλεονεκτημάτων της ΤΚΠ, έναντι του μοντέλου πελάτη/εξυπηρετητή. Στην συνέχεια παραθέτουμε τον τρόπο με τον οποίο μερικά από αυτά βρίσκουν εφαρμογή στα συστήματα διαχείρισης δικτύου [Bieszczad98]:

Εύκολη επέκταση της λειτουργίας: Για την επέκταση των λειτουργιών του συστήματος είναι αρκετή η αναβάθμιση της λειτουργίας ενός πράκτορα και όχι όλων των στοιχείων του συστήματος.

Μείωση του υπολογιστικού φόρτου του διαχειριστή: Με χρήση των ΚΠ, η επεξεργασία της πληροφορίας και η λήψη αποφάσεων δεν γίνεται κεντροποιημένα. Ένας πράκτορας μπορεί να επεξεργαστεί την πληροφορία τοπικά στο δικτυακό στοιχείο και να παραδώσει στον διαχειριστή μόνο τα αποτελέσματα της επεξεργασίας. Με αυτόν τον τρόπο, μέρος της υπολογιστικής δύναμης μεταφέρεται από τον διαχειριστή στα δικτυακά στοιχεία.

Μείωση της κίνησης στο δίκτυο: Με την χρήση ΚΠ δεν είναι απαραίτητη η μεταφορά της πληροφορίας στον διαχειριστή, αφού η επεξεργασία γίνεται τοπικά. Φυσικά υπάρχει μια επιβάρυνση στο δίκτυο λόγω της μετακίνησης του πράκτορα, αλλά τις περισσότερες φορές είναι αμελητέα σε σχέση με τον όγκο των δεδομένων που διαφορετικά θα έπρεπε να μετακινηθεί.

Ανεξαρτησία από την διαθεσιμότητα του δικτύου: Ένα πράκτορας μπορεί να συνεχιστεί τοπικά το έργο του ακόμα και όταν το δίκτυο δεν είναι διαθέσιμο. Μετά την αποκατάσταση του δικτύου, τα αποτελέσματα της διεργασίας μεταφέρονται στον διαχειριστή.

Οι παραπάνω λόγοι δικαιολογούν την τάση που υπάρχει σήμερα, για την χρήση της ΤΚΠ στην υλοποίηση συστημάτων της διαχείρισης δικτύων.

4.3 Σύστημα διαχείρισης δικτύου διαφοροποιημένων υπηρεσιών

4.3.1 Αρχιτεκτονική του Συστήματος

Η αρχιτεκτονική του προτεινομένου συστήματος διαχείρισης δικτύου ΔΥ, παρουσιάζεται στο σχήμα 4.2 [D22]. Το σύστημα χωρίζεται λειτουργικά σε τέσσερα επίπεδα. Το επίπεδο εφαρμογής, το επίπεδο υπηρεσιών, το επίπεδο προσαρμογής και το επίπεδο δικτύου.

Στο **επίπεδο εφαρμογής** βρίσκεται η διεπαφή που επιτρέπει την επικοινωνία των χρηστών με το σύστημα. Εδώ θα πρέπει να σημειωθεί ότι υπάρχουν δυο ομάδες χρηστών, οι απλοί χρηστές και οι διαχειριστές του συστήματος. Η επικοινωνία και των δυο ομάδων χρηστών με το σύστημα πραγματοποιείται μέσω γραφικού περιβάλλοντος. Με χρήση του περιβάλλοντος αυτού, οι διαχειριστές μπορούν να αλλάξουν δυναμικά τους παραμέτρους ΠοιΥπ του δικτύου (π.χ το μέγιστο εύρος ζώνης της κάθε κλάσης υπηρεσίας, το μέγεθος των ουρών αναμονής, το μέγεθος των ενταμιευτών κ.α). Ακόμη, μπορούν να αλλάξουν την περίοδο παρακολούθησης, να εγγράψουν και να διαγράψουν χρηστές, καθώς και τα σημεία πρόσβασης τους στο δίκτυο. Οι χρηστές μπορούν μέσω της εφαρμογής να ζητούν μια μονοκατευθυντήρια ή αμφίδρομη σύνδεση με συγκεκριμένη ΠοιΥπ, με άλλους χρηστές του συστήματος. Τόσο οι χρηστές όσο και οι διαχειριστές μπορούν, μέσω της εφαρμογής, να παρατηρήσουν την ΠοιΥπ που λαμβάνει η κίνηση στο δίκτυο. (καθυστέρηση, απόκλιση καθυστέρησης, εύρος ζώνης και απώλειες).

Το **επίπεδο υπηρεσιών** παρέχει όλες τις απαραίτητες λειτουργίες για την δημιουργία συνδέσεων με συγκεκριμένη ΠοιΥπ, καθώς επίσης και τις λειτουργίες παρακολούθησης της. Η ομάδα Parlay [Parlay], είναι μια ανοικτή κοινοπραξία που συγκροτήθηκε με σκοπό να αναπτύξει ανοικτές Διεπαφές Προγραμματισμού Εφαρμογής (ΔΠΕ), οι οποίες θα μπορούν να χρησιμοποιηθούν από τρίτους. Στα πλαίσια των εργασιών της, ανέπτυξε μια ΔΠΕ για την δημιουργία συνδέσεων σε ένα δίκτυο ΔΥ. Η ΔΠΕ αυτή υιοθετήθηκε από το σύστημα προκειμένου να επιτευχθεί η διάρθρωση των παραμέτρων ΠοιΥπ. Εκτός από την εφαρμογή των προδιαγραφών Parlay, υιοθετήθηκε επίσης η προσέγγιση που προτείνεται από το JAIN [JAIN]. Αυτή η ομάδα εργασίας υποστηρίζεται από την Sun

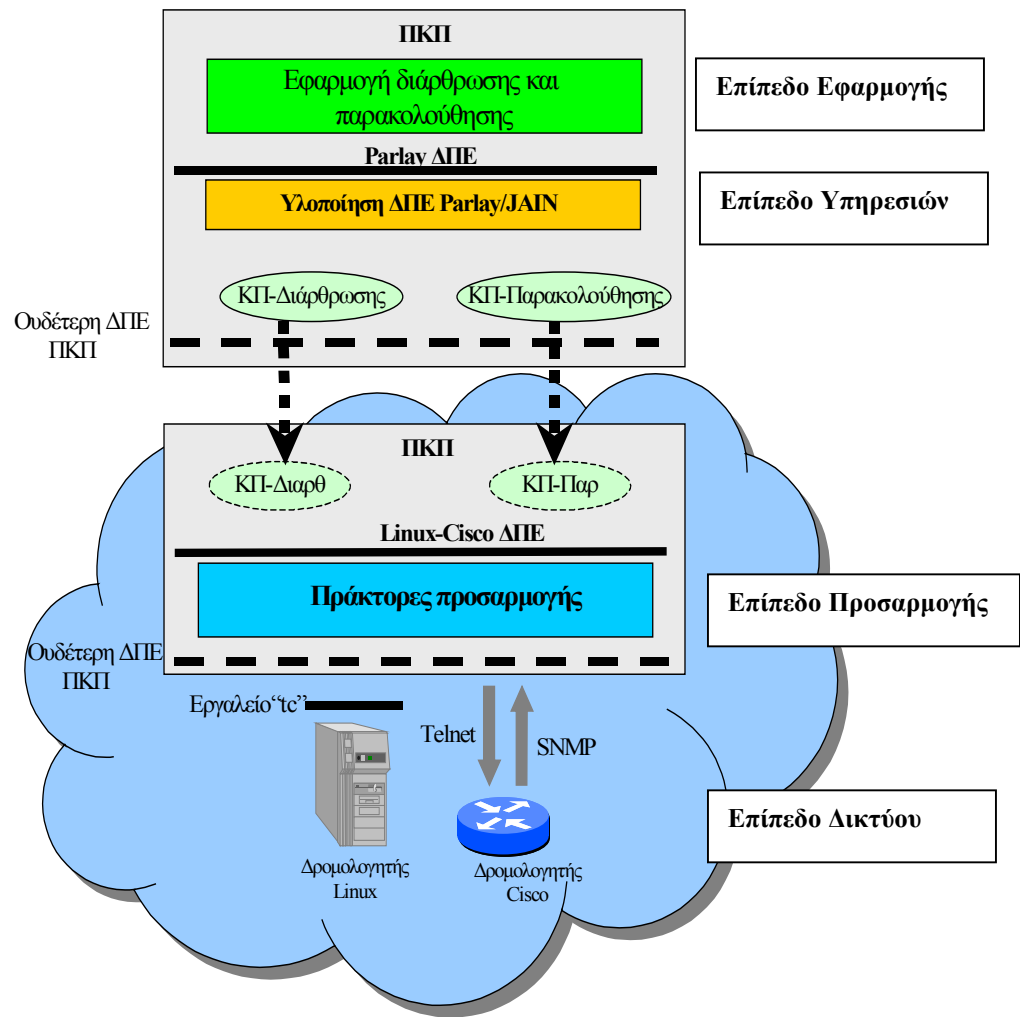
Microsystems και έχει έναν παρόμοιο στόχο με το Parlay, αλλά περισσότερο εστιασμένο στο να καθορίσει μια ανοικτή ΔΠΕ βασισμένη στην γλώσσα προγραμματισμού Java. Φυσικά υπήρξαν μερικές αλλαγές και προσθήκες στους υπάρχοντες ορισμούς των ΔΠΕ, χωρίς όμως να αλλάζει ριζικά η δομή τους.

Το σύστημα είναι ικανό να διαχειριστεί IP δίκτυα βασισμένα τόσο σε δρομολογητές τύπου Cisco, όσο και σε υπηρεσίες δρομολόγησης που παρέχονται από το λειτουργικό σύστημα Linux (δρομολογητές τύπου Linux). Το **επίπεδο προσαρμογής** είναι αρμόδιο για την απόκρυψη των λεπτομερειών των πρωτοκόλλων του υποκείμενου επιπέδου δικτύου, από το στρώμα υπηρεσιών. Στην πραγματικότητα, το επίπεδο αυτό αποτελείται από στατικούς πράκτορες οι οποίοι βρίσκονται στους κόμβους του δικτύου. Οι στατικοί πράκτορες δρουν ως διεπαφές μεταξύ των ΚΠ που είναι υπεύθυνοι για την διάρθρωση και παρακολούθηση των δρομολογητών και του δικτυακού στοιχείου.

Τέλος το **επίπεδο δικτύου** αποτελείται από τους δρομολογητές του δικτύου.

Το όλο σύστημα είναι στηριγμένο στην ΤΚΠ. Όπως παρουσιάζεται και στο σχήμα 4.2, όλες οι λειτουργίες τις εφαρμογής εκτελούνται μέσα στο περιβάλλον που παρέχει μια ΠΚΠ. Έτσι επιτυγχάνεται η ευκολότερη επικοινωνία των στοιχείων του συστήματος, αφού έναν πράκτορα επικοινωνεί ευκολότερα με έναν άλλον πράκτορα, από ότι με μια άλλη εφαρμογή. Όλα τα δομικά στοιχεία του συστήματος, εκτός των ΚΠ, μπορούν να θεωρηθούν ως στατικοί πράκτορες. Η επικοινωνία και αλληλεπίδραση του συστήματος με τους απομακρυσμένους δικτυακούς κόμβους γίνεται με χρήση ΚΠ. Υπάρχουν πράκτορες υπεύθυνοι για την διάρθρωση του δικτύου (ΚΠ-Διάρθρωσης) και πράκτορες υπεύθυνοι για την παρακολούθηση του δικτύου (ΚΠ-Παρακολούθησης).

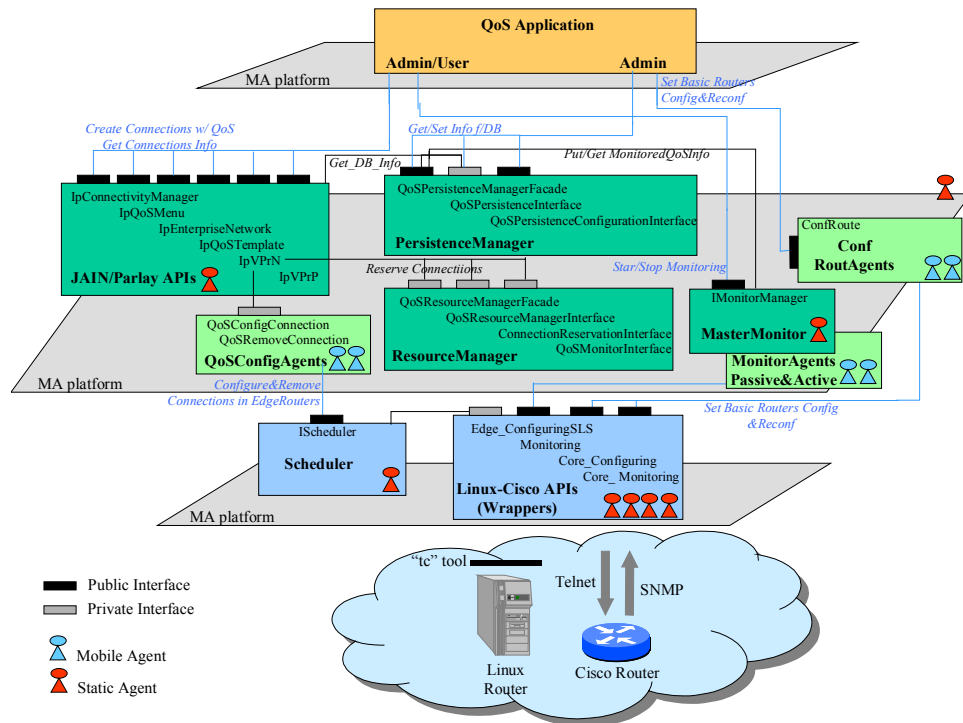
Ένα από τα μεγαλύτερα προβλήματα που αντιμετωπίζουν οι εφαρμογές κινητών πρακτόρων σήμερα, είναι η δέσμευση τους με μια συγκεκριμένη ΠΚΠ. Έτσι κρίθηκε σκόπιμη η ανάπτυξη μιας ουδέτερη ΔΠΕ ΠΚΠ για τις πλατφόρμες Grasshopper και Voyager, οι οποίες χρησιμοποιούνται συνήθως. Φυσικά η ουδέτερη ΔΠΕ μπορεί εύκολα να επεκταθεί ώστε να συμπεριληφθούν και άλλες ΠΚΠ. Η ΔΠΕ επιτρέπει στην εφαρμογή (και τους πράκτορες φυσικά) να εκτελείται σε δυο διαφορετικές ΠΚΠ, με ελάχιστη μεταβολή στον κώδικα



Σχήμα 4.2: Αρχιτεκτονική του συστήματος παροχής και παρακολούθησης ΠοιΥπ

4.3.2 Ανάλυση της λειτουργία του συστήματος

Στο σχήμα 4.3, παρουσιάζεται αναλυτικά η αρχιτεκτονική του συστήματος με όλα τα υποσυστήματα τις διεπαφές τους, κινητούς και στατικούς πράκτορες [D42].



Σχήμα 4.3: Δομικά στοιχεία της αρχιτεκτονική του συστήματος διάρθρωσης και παρακολούθησης ΠοιΥ

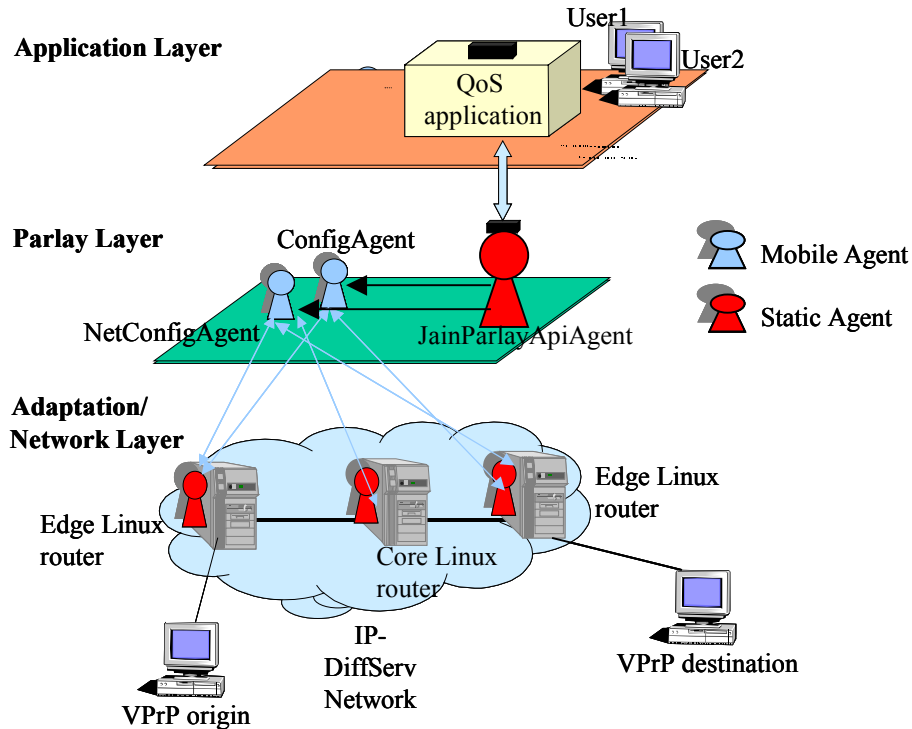
Το σύστημα χωρίζεται λειτουργικά σε μια σειρά από υποσυστήματα, κάθε ένα από τα οποία πραγματοποιεί μια συγκεκριμένη λειτουργία. Τα υποσυστήματα αυτά είναι:

- Υποσύστημα εφαρμογής
- Υποσύστημα διάρθρωσης ΠοιΥπ
- Υποσύστημα παρακολούθησης ΠοιΥΠ
- Υποσύστημα προσαρμογής

Όλο το σύστημα είναι υλοποιημένο με χρήση της γλώσσας προγραμματισμού Java.

4.3.2.1 Υποσύστημα διάρθρωσης

Το υποσύστημα διάρθρωσης βρίσκεται στο επίπεδο υπηρεσιών του συστήματος και είναι υπεύθυνο για την δημιουργία και λύση των συνδέσεων ΠοιΥπ. Η λειτουργία του υποσυστήματος αυτού είναι βασισμένη στον διαχειριστή συνδετικότητας της ΔΠΕ Parlay [Parlay2]. Η λειτουργία του υποσυστήματος με την χρήση των ΔΠΕ Parlay και των πρακτόρων παρουσιάζεται στο σχήμα 4.4.



Σχήμα 4.4: Λειτουργία του υποσυστήματος διάρθρωσης

Πριν παρουσιαστεί ο τρόπος λειτουργίας του υποσυστήματος διάρθρωσης, είναι απαραίτητο να εξηγηθεί η λογική με την οποία δημιουργούνται οι συνδέσεις, σύμφωνα με την ΔΠΕ Jain/Parlay. Στο δίκτυο του παροχέα (Enterprise Network) δημιουργούνται μια σειρά από εικονικά ιδιωτικά δίκτυα (Virtual Private Networks, VPNs). Μέσα στα VPPrPs ο χρήστης μπορεί να δημιουργήσει εικονικούς ιδιωτικούς δίαυλους (Virtual Private Pipes, VPPrPs). Οι δίαυλοι αυτοί είναι οι συνδέσεις που παρέχουν μια συγκεκριμένη ΠοιΥπ. Κάθε PHB της αρχιτεκτονικής ΔΥ, αναπαριστάται από μια φόρμα που περιέχει όλη την πληροφορία σχετική με αυτό (καθυστέρηση, μέγιστο και ελάχιστο εύρος ζώνης, κ.τ.λ). Η φόρμα αυτή ονομάζεται φόρμα ΠοιΥπ (QoSTemplate). Οι φόρμες ΠοιΥπ ορίζονται από τον διαχειριστή του συστήματος και αντιστοιχούν στις δεκατέσσερις διαφορετικές PHBs, που προσφέρει η αρχιτεκτονική ΔΥ.

Ο JainParlayApiAgent είναι ένας στατικός πράκτορας ο οποίος υλοποιεί τον διαχειριστή συνδετικότητας Parlay. Ένας διαχειριστής συνδετικότητας κατά Parlay αποτελείται από τρεις λογισμικές υπομονάδες: τον διαχειριστή συνδετικότητας (ParlayConnectivityManager), τον διαχειριστή οντοτήτων (QoSPersistenceManager) και τον διαχειριστή πόρων (QoSResourceManager) (βλέπε σχήμα 4.3).

Ο ParlayConnectivityManager ελέγχει την διαδικασία δημιουργία της σύνδεσης και είναι η υπομονάδα η οποία επικοινωνεί με το υποσύστημα εφαρμογής. Για την δημιουργία μια σύνδεσης, ο ParlayConnectivityManager χρησιμοποιεί τις δυο άλλες υπομονάδες τόσο για να δώσει πληροφορίες στο υποσύστημα εφαρμογής, όσο και για τον έλεγχο της διαδικασίας δημιουργίας συνδέσεως. Ο QoSPersistenceManager είναι υπεύθυνος για την αρχειοποίηση και διατήρηση της πληροφορίας που σχετίζεται με μια σύνδεση. Προσφέρει πρόσβαση στις φόρμες ΠοιΥπ, διατηρεί τις αλλαγές του χρήστη πάνω στην φόρμα και παρέχει πληροφορίες για την απόδοση της συγκεκριμένης σύνδεσης. Η πληροφορία για την απόδοση μια σύνδεσης είναι προϊόν παρακολούθησης της. Ο τρόπος με τον οποίο πραγματοποιείται η παρακολούθηση παρουσιάζεται στην αμέσως επόμενη ενότητα. Ο QoSResourceManager είναι υπεύθυνος για την διαχείριση της πληροφορίας που σχετίζεται με του πόρους του δικτύου. Διατηρεί μια συνολική εικόνα των διαθέσιμων πόρων. Πριν την δημιουργία μιας σύνδεσης και αφού έχει διαπιστωθεί ότι ο χρήστη την δικαιούται, ο QoSResourceManager παρέχει την πληροφορία για το αν υπάρχουν οι απαραίτητοι πόροι στον δίκτυο, για την επίτευξη της. Ο QoSResourceManager είναι λοιπόν υπεύθυνος για την λειτουργία αποδοχής κλήσης (admission control). Αν υπάρχουν οι απαιτούμενοι πόροι ο JainParlayApiAgent δημιουργεί τον ΚΠ διάρθρωσης σύνδεσης (QoSConfigureConnectionAgent), ο οποίος μεταναστεύει στον δρομολογητή που αποτελεί το σημείο πρόσβασης του χρήστη στο δίκτυο. Ο ΚΠ διάρθρωσης σύνδεσης αλληλεπιδρά με τον στατικό πράκτορα του επιπέδου προσαρμογής (την λειτουργία αυτών των πρακτόρων θα την αναλύσουμε στην ενότητα που ασχολείται με το υποσύστημα προσαρμογής), που βρίσκεται εκεί και διαρθρώνει κατάλληλα τον δρομολογητή. Για την λύση μια σύνδεσης, ο JainParlayApiAgent δημιουργεί τον ΚΠ λύσης σύνδεσης (QoSRemoveConnectionAgent), ο οποίος μεταναστεύει στον δρομολογητή που αποτελεί το σημείο πρόσβασης του χρήστη στο δίκτυο. Ο πράκτορας αυτός, αλληλεπιδρά με τον στατικό πράκτορα του επιπέδου προσαρμογής, που βρίσκεται εκεί και διαρθρώνει κατάλληλα τον δρομολογητή, για την λύση της σύνδεσης.

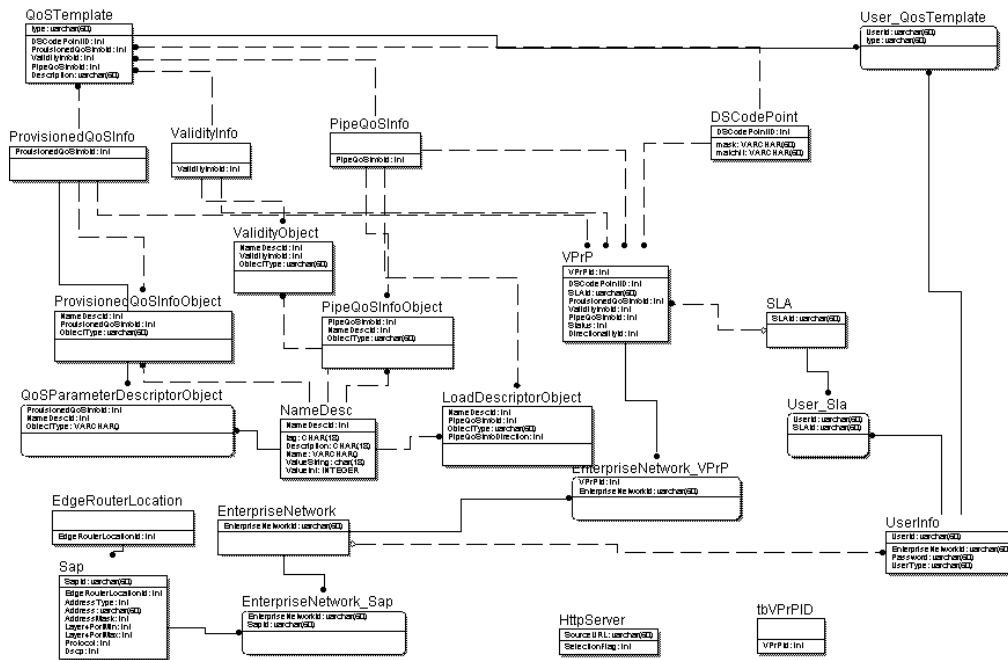
4.3.2.1.1 Προπαρασκευή του συστήματος πριν την δημιουργία συνδέσεων

Για την λειτουργία του υποσυστήματος διάρθρωσης είναι απαραίτητο να προηγηθεί μια διαδικασία προπαρασκευής της πληροφορίας που διαθέτει το υποσύστημα και μια αρχική διάρθρωση του υπό διαχείριση δικτύου. Σε ένα δίκτυο ΔΥ υπάρχει διάκριση μεταξύ της λειτουργίας των δρομολογητών κορμού (Core) και των δρομολογητών ακρών (Edge) του δικτύου. Βασική προϋπόθεση για την λειτουργία του συστήματος, είναι μια αρχική διάρθρωση όλων των δρομολογητών κορμού του δικτύου. Αυτό επιτυγχάνεται με την αποστολή των ΚΠ προπαρασκευής (NetConfigAgents), στους δρομολογητές αυτούς. Μετά την αρχική διόρθωση από τον NetConfigAgent, οι δρομολογητές κορμού είναι σε θέση να προωθήσουν κατάλληλα την κίνηση από τις διάφορες κλάσεις υπηρεσίας. Άρα για την δημιουργία μιας σύνδεσης στο δίκτυο ΔΥ, είναι αρκετή η διάρθρωση μόνο των ακραίων δρομολογητών. Οι ακραίοι δρομολογητές είναι τα σημεία πρόσβασης των χρηστών στο δίκτυο. Επίσης, για την λειτουργία του υποσυστήματος διάρθρωσης θα πρέπει, εκ των προτέρων, να έχουν εγκατασταθεί στους ακραίους δρομολογητές οι πράκτορες του επιπέδου προσαρμογής. Μόνον τότε είναι δυνατή η διάρθρωση τους, από τους πράκτορες διάρθρωσης (QoSConfigureConnectionAgent, QoSRemoveConnectionAgent).

Το σύστημα είναι εφοδιασμένο με μια βάση πληροφοριών (Το σύστημα διαχείρισης της βάσης είναι η MySQL [MySQL]) η οποία δεν χρησιμοποιείται μόνο από το υποσύστημα διάρθρωσης, αλλά και από άλλα υποσυστήματα για την αποθήκευση και άντληση πληροφοριών. Πριν την χρήση του υποσυστήματος διάρθρωσης θα πρέπει να αποθηκευτούν στη βάση οι απαραίτητες για την λειτουργία του πληροφορίες. Η διαδικασία αυτή πραγματοποιείται με την χρήση γραφικού περιβάλλοντος από τον διαχειριστή. Η πληροφορία αυτή περιλαμβάνει:

- Την πληροφορία για τις διάφορες φόρμες ΠοιΥπ (QoSTemplates).
- Την πληροφορία σχετική με του χρηστές του συστήματος (είδος χρήστη, δίκτυο στο οποίο ανήκει, τα συμβόλαια υπηρεσιών (SLA) του χρήστη).
- Την δομή του δικτύου (τους δρομολογητές, το εύρος ζώνης που προσφέρει ο καθένας και το εύρος ζώνης που δικαιούται κάθε κλάση υπηρεσίας).
- Τα σημεία πρόσβασης των χρηστών στο δίκτυο (Service access points-Sap)

Η βάση του συστήματος παρουσιάζεται στην εικόνα 4.1



Εικόνα 4.1: Η βάση πληροφοριών του συστήματος

4.3.2.1.2 Δημιουργία σύνδεσης με χρήση του υποσυστήματος διάρθρωσης

Στους επόμενους πίνακες παρουσιάζονται οι διεπαφές των λογισμικών υπομονάδων του υποσυστήματος διάρθρωσης.

Πίνακας 4.1: Διεπαφές την υπομονάδας ParlayConnectivityManager

Διεπαφή	Λειτουργία
IpConnectivityManager	Είναι το σημείο πρόσβασης προς το στρώμα υπηρεσίας. Μέσω αυτού είναι δυνατή η προσπέλαση των διεπαφών IpQoSMenu ή το IpEnterpriseNetwork.
IpQoSMenu	Παρέχει πρόσβαση σε ένα ή σε μια λίστα από IpQoSTemplates.
IpQoSTemplates	Κάθε κλάση υπηρεσίας που μπορεί να προσφερθεί στον χρήστη αναπαριστάται από μια μοναδική φόρμα (template). Η διεπαφή IpQoSTemplates επιτρέπει την άντληση/μετατροπή πληροφοριών που σχετίζονται με μια κλάση υπηρεσίας ή μια σύνδεση.
IpEnterpriseNetwork:	Επιτρέπει την άντληση πληροφοριών για το δίκτυο του παροχέα υπηρεσιών.
IpVPrN	Επιτρέπει τη δημιουργία/καταστροφή των εικονικών ιδιωτικών δικτύων. Επίσης, παρέχει πληροφορία για τα VPrP που έχουν

	δημιουργηθεί σε αυτά.
IpVPrP	Επιτρέπει την άντληση πληροφοριών που σχετίζονται με μια εικονική σύνδεση (VPrP).

Πίνακας 4.2: Διεπαφές την υπομονάδας *QoSPersistenceManager*

Διεπαφή	Λειτουργία
QoSPersistenceManagerFacade	Υπεύθυνη για την παροχή πρόσβασης σε αντικείμενα που υλοποιούν τις διεπαφές QoSPersistenceInterface και QoSPersistenceConfigurationInterface.
QoSPersistenceInterface	Υπεύθυνο να αρχικοποιεί, να δημιουργεί και να αλλάζει τη διατηρούμενη κατάσταση των αντικειμένων ΠοιΥπ.
QoSPersistenceConfigurationInterface	Υπεύθυνο για τη συμπλήρωση της βάσης με τα απαραίτητα δεδομένα.

Πίνακας 4.3: Διεπαφές την υπομονάδας *QoSResourceManager*

Διεπαφή	Λειτουργία
QoSResourceManagerFacade	Υπεύθυνη για την παροχή/πρόσβασης σε αντικείμενα που υλοποιούν το QoSResourceManagerInterface.
QoSResourceManagerInterface	Προσφέρει υπηρεσίες για την δέσμευση και απελευθέρωση των πόρων του δικτύου.
ConnectionReservationInterface	Αναπαριστά μια δέσμευσης σύνδεσης.

Στις εικόνες 4.2 και 4.3 παρουσιάζεται η αλληλουχία λειτουργιών που πραγματοποιούνται για την δημιουργία μια σύνδεσης, με χρήση του υποσυστήματος διάρθρωσης.

Εικόνα 4.2: Αρχικά ο χρήστης επιλέγει μια φόρμα ΠοιΥπ (QoSTemplate), η οποία συσχετίζεται με ένα από τα συμβόλαια υπηρεσιών (SLA) του [βήματα 1→4]. Στη συνέχεια, μεταβάλλει τις προκαθορισμένες τιμές την φόρμας ΠοιΥπ που έχει επιλέξει [βήματα 4→10]. Κάθε φόρμα ΠοιΥπ περιέχει τρία διαφορετικά μέρη πληροφορίας, στα οποία ο χρήστης μπορεί μερικώς να επέμβει. Ο χρήστης δεν μπορεί να αλλάξει στοιχεία της φόρμας που μεταβάλλουν την κλάση υπηρεσίας, που παρέχει η φόρμα αυτή. Έτσι μπορεί π.χ. να διαλέξει αν η σύνδεση θα είναι αμφίπλευρη και το μέγιστο και ελάχιστο εύρος ζώνης, αλλά δεν μπορεί να αλλάξει τις παραμέτρους που σχετίζονται με την

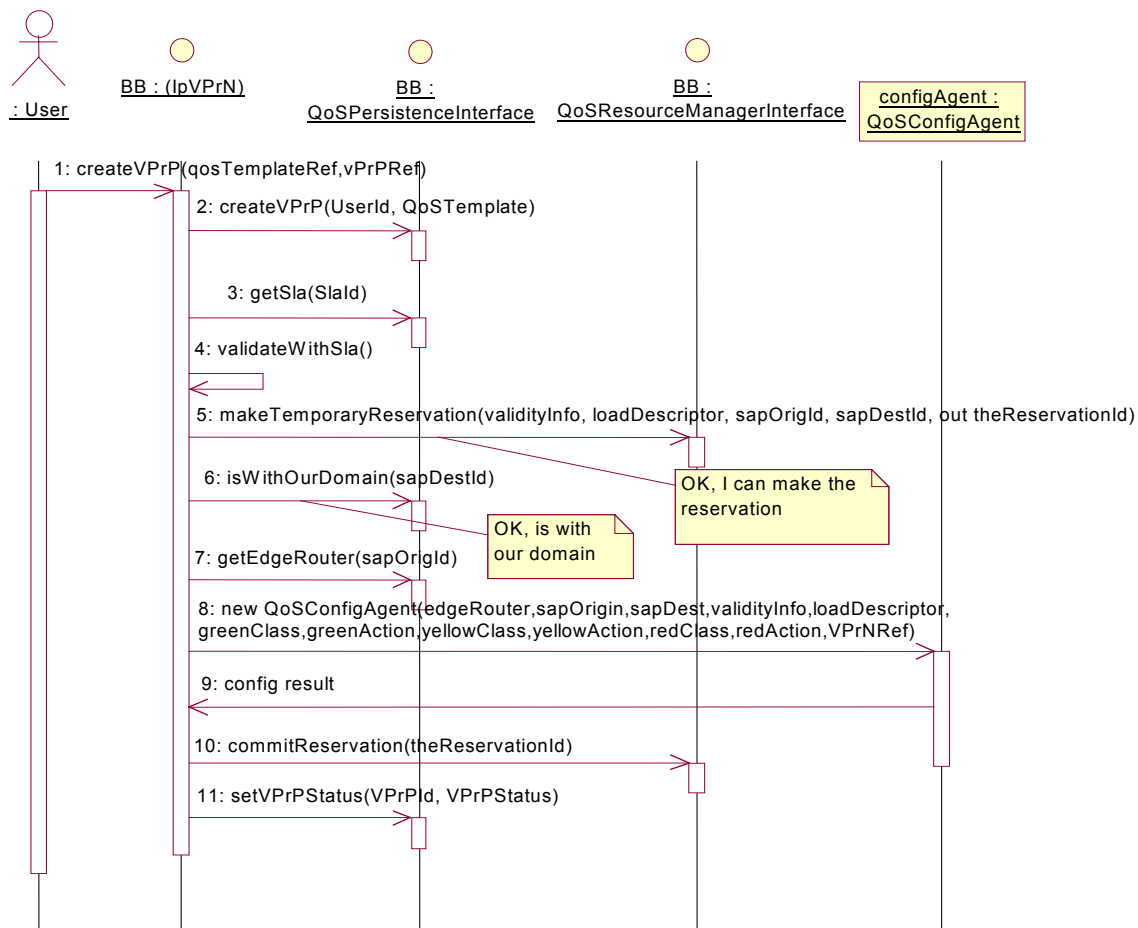
καθυστέρηση των πακέτων της συγκεκριμένης κλάσης ΠοιΥπ, βάσει της οποίας έχει δημιουργηθεί η σύνδεση. Τα τρία διαφορετικά μέρη πληροφορίας που περιέχει μια φόρμα ΠοιΥπ είναι: α) πληροφορία σχετική με το δίαυλο σύνδεσης (PipeQoSInfo), β) πληροφορία σχετική με τα χαρακτηριστικά μετάδοσης την σύνδεσης (ProvisionQoSInfo), όπως καθυστέρηση, απώλειες κ.α και γ) πληροφορία σχετική με τον χρόνο δημιουργίας και διάρκειας της σύνδεσης (ValidityInfo). Έπειτα από την διαμόρφωση της φόρμας, αντλείται η πληροφορία για το δίκτυο παροχέα του χρήστη, το εικονικό ιδιωτικό δίκτυο στο οποίο θα δημιουργηθεί η σύνδεση και δίνεται η εντολή για την δημιουργία της [βήματα 11→13].



Εικόνα 4.2 Διάγραμμα αλληλουχίας δημιουργίας συνδέσεως (α)

Εικόνα 4.3 (συνεχία από το βήμα 13 της εικόνας 4.2): Αρχικοποιείται η σύνδεση στο επίπεδο οντοτήτων. Δηλαδή, δημιουργείται ένα αντικείμενο με την πληροφορία για αυτή

την σύνδεση [βήματα 1→2]. Πραγματοποιείται ο έλεγχος για το αν το η αιτούμενη σύνδεση είναι σύμφωνη με το συμβόλαιο υπηρεσιών του χρήστη [βήματα 2->4]. Ελέγχεται αν υπάρχουν οι απαιτούμενοι πόροι στο δίκτυο για την δημιουργία της σύνδεσης και πραγματοποιείται μια αρχική τους δέσμευση. Η σύνδεση θεωρείται τώρα ως εκκρεμής [βήμα 5]. Ελέγχεται αν και οι δυο χρηστές της σύνδεσης ανήκουν στο ίδιο δίκτυο [βήμα 6]. Αναγνωρίζεται ο ακραίος δρομολογητής της σύνδεσης [βήμα 7]. Δημιουργείται ο ΚΠ διάρθρωσης και αποστέλλεται στον ακραίο δρομολογητή [βήμα 8]. Γίνεται γνωστό το αποτέλεσμα της διάρθρωσης [βήμα 9]. Επιβεβαιώνεται η δέσμευση των πόρων του δικτύου για αυτή την σύνδεση. Η σύνδεση είναι πλέον ενεργή [βήματα 10→11].



Εικόνα 4.3: Διάγραμμα αλληλουχίας δημιουργίας συνδέσεως (β)

4.3.2.2 Υποσύστημα Παρακολούθησης ΠοιΥπ

Το υποσύστημα Παρακολούθησης ΠοιΥπ βρίσκεται στο στρώμα υπηρεσιών του συστήματος και είναι υπεύθυνο για την παρακολούθηση της ΠοιΥπ, που λαμβάνουν οι

υπάρχουσες συνδέσεις. Χρησιμοποιούνται τεχνικές τόσο ενεργητικής όσο και παθητικής παρακολούθησης. Η υπηρεσία παρακολούθησης είναι μια βασική λειτουργία που πρέπει να παρέχει κάθε σύστημα παροχής ΠοιΥπ, ώστε να είναι σε θέση ο χρήστης να πιστοποιεί ότι η ΠοιΥπ που του παρέχεται είναι αυτή την οποία ζήτησε και για την οποία έχει πληρώσει.. Συγκεκριμένα, οι παράμετροι οι οποίοι παρακολουθούνται είναι:

Παθητική Παρακολούθηση

- Εύρος ζώνης
- Απώλεια πακέτων

Ενεργητική Παρακολούθηση

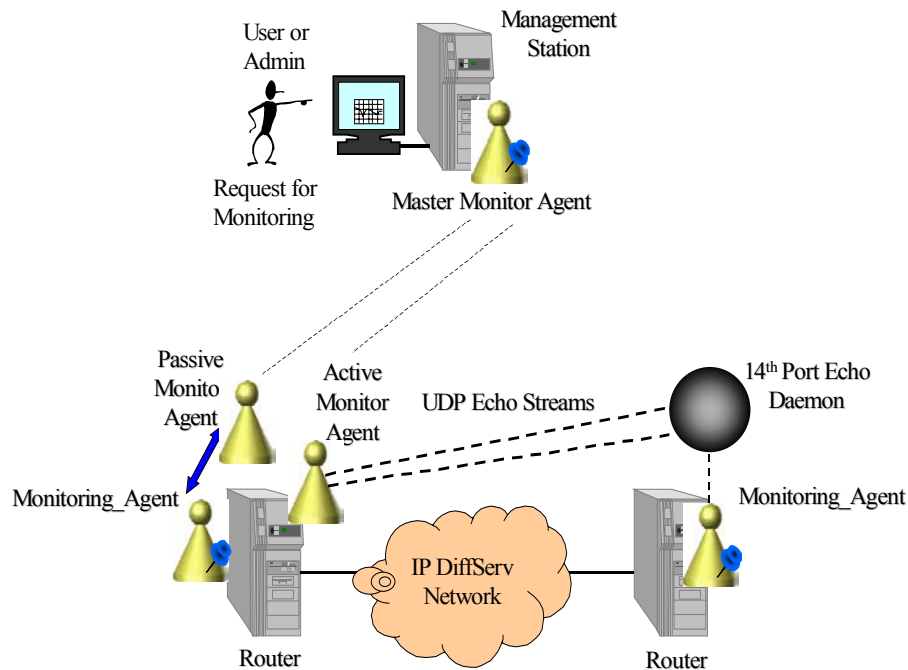
- Καθυστέρηση
- Απόκλιση καθυστέρησης

Η λειτουργία του συστήματος παρακολούθησης ΠοιΥπ παρουσιάζεται στο σχήμα 4.5. Ο πράκτορας διαχείρισης παρακολούθησης (MasterMonitorAgent) είναι υπεύθυνος για το συντονισμό του συστήματος παρακολούθησης, στα πλαίσια της δημιουργίας/τροποποίησης/τερματισμού μιας λειτουργίας παρακολούθησης, για ένα συγκεκριμένο VPrP. Ο MasterMonitorAgent θα δημιουργήσει τους ΚΠ με σκοπό την παθητική και ενεργητική παρακολούθηση.

Η παθητική παρακολούθηση χρησιμοποιείται από το υποσύστημα παρακολούθησης, προκειμένου να παρασχεθούν στοιχεία για τη εκμετάλλευση του εύρους ζώνης και την απώλεια πακέτων μιας σύνδεσης. Η διαδικασία παρακολούθησης καλείται “παθητική”, καθώς τα στοιχεία λαμβάνονται χωρίς την παρέμβασης της διαδικασίας παρακολούθησης στην πραγματική σύνδεση. Ο πράκτορας παθητικής παρακολούθησης (PassiveMonitoringAgent) μεταναστεύει στον ακραίο δρομολογητή της σύνδεσης και εξάγει τις ακατέργαστες πληροφορίες απωλειών και εύρους ζώνης. Η πληροφορίες αυτές παρέχονται από τον πράκτορα παρακολούθησης (Monitoring_Agent), ο οποίος είναι ένας πράκτορας του επιπέδου προσαρμογής.

Η ενεργητική παρακολούθηση χρησιμοποιείται από το σύστημα παρακολούθησης προκειμένου να παρασχεθούν δεδομένα για την καθυστέρηση πακέτων και την απόκλιση καθυστέρησης μιας σύνδεσης. Μια τέτοια διαδικασία ελέγχου καλείται ενεργητική, διότι

τα δεδομένα λαμβάνονται με την δημιουργία ενός μικρού ποσού κυκλοφορίας στο πραγματικό δίκτυο. Ο ActiveMonitoringAgent μεταναστεύει στον δρομολογητή εισόδου της σύνδεσης και ανταλλάσσει UDP πακέτα με τον Monitoring_Agent, ο οποίος βρίσκεται στον δρομολογητή εξόδου της σύνδεσης. Ο Monitoring_Agent υλοποιεί έναν δαίμονα που επιστρέφει τα πακέτα τα οποία δέχεται (Echo daemon). Με την ανταλλαγή UDP πακέτων, με διαφορετικό DSCP πεδίο, είναι δυνατή η ενεργητική παρακολούθηση των διαφορετικών κλάσεων υπηρεσίας.



Σχήμα 4.5: Η λειτουργία του συστήματος παρακολούθησης ΠοιΥπ

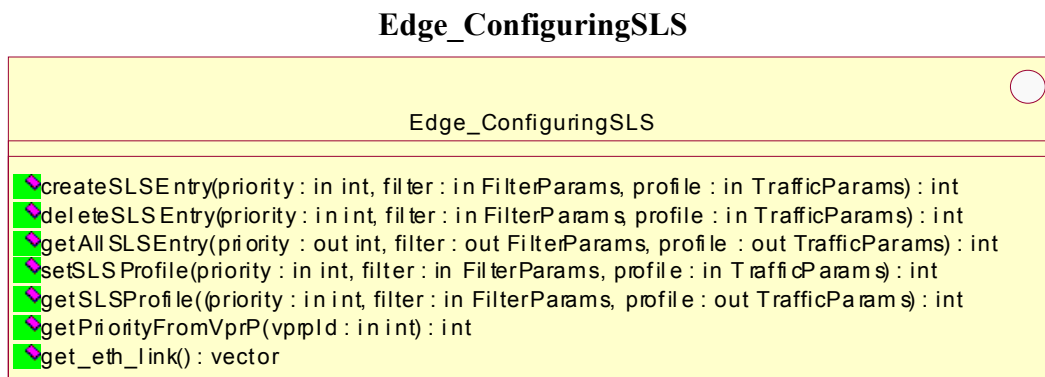
Τόσο ο PassiveMonitoringAgent όσο και ο ActiveMonitoringAgent μεταναστεύουν μετά το πέρας ενός ορισμένου χρονικού διαστήματος (ορίζεται από τον χρήστη) στο μηχάνημα που βρίσκεται MasterMonitorAgent και παραδίδουν την πληροφορία που συνέλεξαν. Ο MasterMonitorAgent αποθηκεύει την πληροφορία αυτήν στην κοινή βάση δεδομένων του συστήματος, χρησιμοποιώντας την διεπαφή QoSPersistenceInterface. Μέσω της ίδιας διεπαφής ο χρήστης είναι σε θέση να ελέγξει την απόδοση της σύνδεσης του, οποιαδήποτε στιγμή το επιθυμεί.

4.3.2.3 Υποσύστημα προσαρμογής

Το υποσύστημα προσαρμογής είναι αρμόδιο για την απόκρυψη των λεπτομερειών των πρωτοκόλλων του υποκείμενου επιπέδου δικτύου, από το επίπεδο υπηρεσιών. Το υποσύστημα αυτό παρουσιάζεται στο σχήμα 4.3. Αποτελείται από τους πράκτορες προσαρμογής (wrappers) και από τους χρονοπρογραμματιστές (schedules). Ήδη έχουμε αναφέρει ότι το σύστημα διαχείρισης είναι σε θέση να διαχειριστεί δίκτυα που αποτελούνται τόσο από δρομολογητές Linux όσο και από δρομολογητές Cisco. Με αυτό το δεδομένο και λόγω του ότι δεν υπάρχει ένα κοινός τρόπος διαχείρισης των δυο αυτών τύπων δρομολογητών, ήταν απαραίτητη η δημιουργία διεπαφών που θα επέτρεπαν στον ΚΠ να αλληλεπιδράσουν με τους δρομολογητές, ανεξάρτητα από το είδος τους. Οι πράκτορες προσαρμογής είναι στατικοί πράκτορες οι οποίο βρίσκονται στον δρομολογητή (η κοντά στον δρομολογητή στην περίπτωση Cisco) και δρουν ως διεπαφές μεταξύ των ΚΠ και των δρομολογητών. Εναλλακτικά, θα μπορούσαν να είχαν δημιουργηθεί διαφορετικοί ΚΠ για κάθε τύπο δρομολογητή. Αυτή η προσέγγιση απορρίφθηκε διότι θα οδηγούσε σε αρκετά πολύπλοκους πράκτορες με μεγάλο μέγεθος. Έτσι λοιπόν, όλες οι λειτουργίες αλληλεπίδρασης των ΚΠ με τους δρομολογητές γίνεται μέσω των πρακτόρων προσαρμογής.

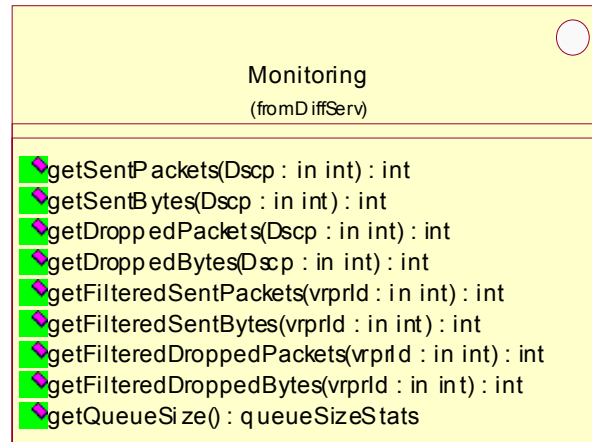
Οι χρονοπρογραμματιστές είναι οι λογισμικές υπομονάδες που εγείρουν τις διαδικασίες διάρθρωσης των δρομολογητών την χρονική στιγμή που έχει απαιτήσει ο χρήστης, στην αίτηση σύνδεσης. Η δημιουργία μιας σύνδεσης δεν ταυτίζεται χρονικά με την αίτηση του χρήστη.

Στην συνέχεια παρουσιάζονται οι διεπαφές του υποσυστήματος προσαρμογής (κάθε μια υλοποιείται από έναν σταθερό πράκτορα)



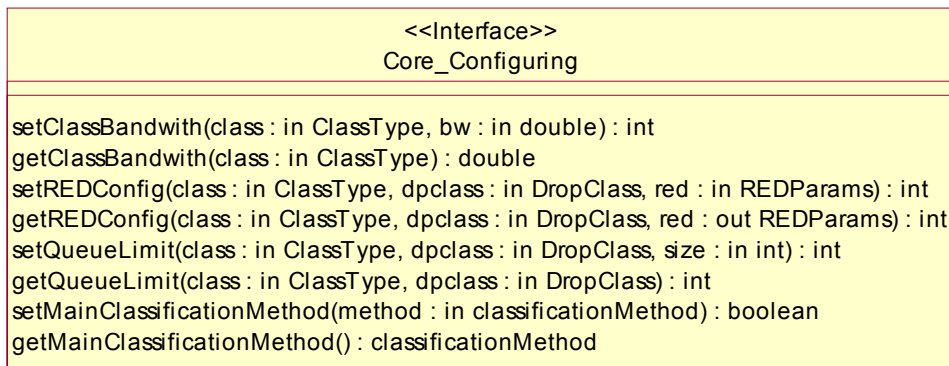
Η διεπαφή `Edge_ConfiguringSLS`, προσφέρει τις μεθόδους που είναι απαραίτητες για την διάρθρωση των ακραίων δρομολογητών (δημιουργία, τερματικός συνδέσεων).

Monitoring



Η διεπαφή `Monitoring` προσφέρει τις μεθόδους για την παθητική παρακολούθηση του δρομολογητή. Ο στατικός πράκτορας που την υλοποιεί, παρέχει επίσης τον δαίμονα επιστροφής πακέτων (`Echo daemon`), που χρησιμοποιείται για την ενεργητική παρακολούθηση.

Core_Configuring



Η διεπαφή `Core_Configuring` χρησιμοποιείται από τους πράκτορες προπαρασκευής για την αρχική διάρθρωση του δικτύου.

4.3.2.3.1 Λεπτομέρειες υλοποίησης του υποσυστήματος προσαρμογής

Ο συνήθης τρόπος διαχείρισης ενός δικτύου είναι με χρήση ενός εκ των πρωτοκόλλων διαχείρισης δικτύου (`SNMP` και `CMIS/CMIP`). Για την χρήση τους όμως είναι

απαραίτητη η ύπαρξη μια κοινής βάση διαχείρισης της πληροφορίας (Management Information Base-MIB). Δυστυχώς, στην περίπτωση των ΔΥ ούτε οι δρομολογητές Cisco [Cisco] ούτε οι δρομολογητές Linux παρέχουν μια τέτοια βάση.

Στην περίπτωση των δρομολογητών Cisco που υποστηρίζουν ΠοιΥπ (Cisco 7000, Cisco 7020) υλοποιείται μια MIB ΔΥ (Class-Based Quality of Service Management Information Base, Class-Based QoS MIB), η οποία όμως παρέχει μόνο λειτουργίες παρακολούθησης. Η διάθρωση των δρομολογητών επιτυγχάνεται με χρήση μιας διεπαφής γραμμής εντολών (Command Line Interface-CLI). Οι εντολές αποστέλλονται στον δρομολογητή μέσω μιας σύνδεσης Telnet. Δηλαδή, Στην περίπτωση των δρομολογητών Cisco, ο στατικός πράκτορας προσαρμογής βρίσκεται σε ένα μηχάνημα κοντά στον δρομολογητή και τον διαρθρώνει χρησιμοποιώντας το CLI. Η παρακολούθηση του δρομολογητή γίνεται με χρήση του πρωτοκόλλου SNMP, από τον πράκτορα προσαρμογής.

Στην περίπτωση των δρομολογητών Linux έχει υλοποιεί μια MIB ΔΥ η οποία όμως βρίσκεται ακόμα σε πειραματικό στάδιο [Kotsilieris02]. Ακόμη και αν η συγκεκριμένη MIB βρισκόταν σε τελική μορφή, δεν θα μπορούσε να υπάρξει ένας ενιαίος τρόπος διαχείρισης των δυο τύπων δρομολογητών, διότι η MIB των δρομολογητών Cisco έχει διαφορετικές προδιαγραφές. Παρόλα αυτά, το λειτουργικό σύστημα Linux έχει εγγενής ικανότητες δρομολόγησης. Με βάση τις ικανότητες αυτές αναπτύχθηκε ένα εργαλείο διαχείρισης δρομολογητών Linux ΔΥ. Το εργαλείο έλεγχου κίνησης (traffic control tool, tc) αναπτύχθηκε από τον Alexander Kuznetsov και αποτελεί σήμερα την μοναδική λύση για την διαχείριση δρομολογητών Linux ΔΥ.

Συνοψίζοντας, οι τεχνικές που χρησιμοποιούνται από το υποσύστημα προσαρμογής για την διάρθρωση και παρακολούθηση των δρομολογητών Cisco και Linux είναι οι παρακάτω:

Δρομολογητές Cisco

- Command line interface για διάρθρωση
- Πρωτόκολλο SNMP για παρακολούθηση

Δρομολογητές Linux

- Traffic control tool για διάρθρωση και παρακολούθηση

4.3.2.3.1.1 Διαχείριση δρομολογητών Cisco

4.3.2.3.1.1.1 Διάρθρωση

Η διεπαφή γραμμής εντολών που χρησιμοποιείται για την διάρθρωση των δρομολογητών Cisco ΔΥ, ονομάζεται Modular Quality of Service Command-Line Interface (Modular QoS CLI) [QoSCLI]. Η διάρθρωση ενός δρομολογητή περιλαμβάνει τρία βήματα:

- Τον ορισμό μια κλάσης κυκλοφορίας, με την χρήση μιας εντολής που ονομάζεται **class-map**.
- Την δημιουργία μια πολιτικής εξυπηρέτησης, συσχετίζοντας μια κλάση κυκλοφορίας με μια ή περισσότερες πολιτικές ΠοιΥπ. Η συσχέτιση πραγματοποιείται χρησιμοποιώντας την εντολή **policy-map**.
- Την συσχέτιση της πολιτικής εξυπηρέτησης με την θύρα εξόδου του δρομολογητή, με χρήση της εντολής **service-policy**.

Μια κλάση κυκλοφορίας περιλαμβάνει τρία στοιχεία, Ένα όνομα, μια σειρά εντολών αντιστοίχισης (match) και μια οδηγία για το πώς να αξιολογηθούν οι εντολές αντιστοίχισης. Οι εντολές αντιστοίχισης χρησιμοποιούνται για να προσδιορίσουν τα διάφορα κριτήρια ταξινόμησης των πακέτων. Τα πακέτα ελέγχονται εάν ταιριάζουν με τα κριτήρια που προσδιορίζονται στις εντολές αντιστοίχισης. Εάν ένα πακέτο ταιριάζει με τα προσδιορισμένα κριτήρια, τότε θεωρείται μέλος της κλάσης και διαβιβάζεται σύμφωνα με της προδιαγραφές ΠοιΥπ που έχουν οριστεί στη πολιτική εξυπηρέτησης. Τα πακέτα που αποτυγχάνουν να ικανοποιήσουν κάποιο από τα κριτήρια αντιστοίχισης, ταξινομούνται ως μέλη μιας προκαθορισμένης κλάσης για αυτό το είδος κίνησης.

Η εντολή policy-map χρησιμοποιείται για να συνδέσει μια κλάση κυκλοφορίας με μια ή περισσότερες πολιτικές ΠοιΥπ. Το αποτέλεσμα αυτής της ένωσης καλείται πολιτική εξυπηρέτησης. Μια πολιτική εξυπηρέτησης περιλαμβάνει τρία στοιχεία: ένα όνομα, μια κλάση κυκλοφορίας (που προσδιορίζεται με την εντολή κλάσης) και πολιτικές ΠοιΥπ. Ο σκοπός της πολιτικής εξυπηρέτησης είναι να συνδεθεί μια κλάση κυκλοφορίας με μια ή περισσότερες πολιτικές ΠοιΥπ. Το Modular QoS CLI δεν απαιτεί οι χρήστες απαραίτητως να συνδέουν μόνο μια κλάση κυκλοφορίας σε μια πολιτική εξυπηρέτησης.

Όταν τα πακέτα ταιριάζουν με περισσότερα από ένα κριτήρια αντιστοίχισης, οι πολλαπλές κλάσεις κυκλοφορίας μπορούν να είναι συνδεδεμένες με μια ενιαία πολιτική εξυπηρέτησης.

Η εντολή service-policy χρησιμοποιείται για να συνδέσει την πολιτική εξυπηρέτησης με μια θύρα εξόδου, του δρομολογητή. Επειδή τα στοιχεία της πολιτικής εξυπηρέτησης μπορούν να εφαρμοστούν στην είσοδο ή στην έξοδο των πακέτων, οι χρήστες πρέπει για να προσδιορίσουν εάν τα χαρακτηριστικά της πολιτικής εξυπηρέτησης πρέπει εφαρμόζονται στα εισερχόμενα ή στα εξερχόμενα πακέτα.

Οι στατικοί πράκτορες του υποσυστήματος προσαρμογής, που είναι υπεύθυνοι για την διάρθρωση του δικτύου, (Edge_ConfiguringSLS, Core_Configuring) παρέχουν μια διεπαφή στο Modular QoS CLI, που επιτρέπει την διάρθρωση των δρομολογητών Cisco.

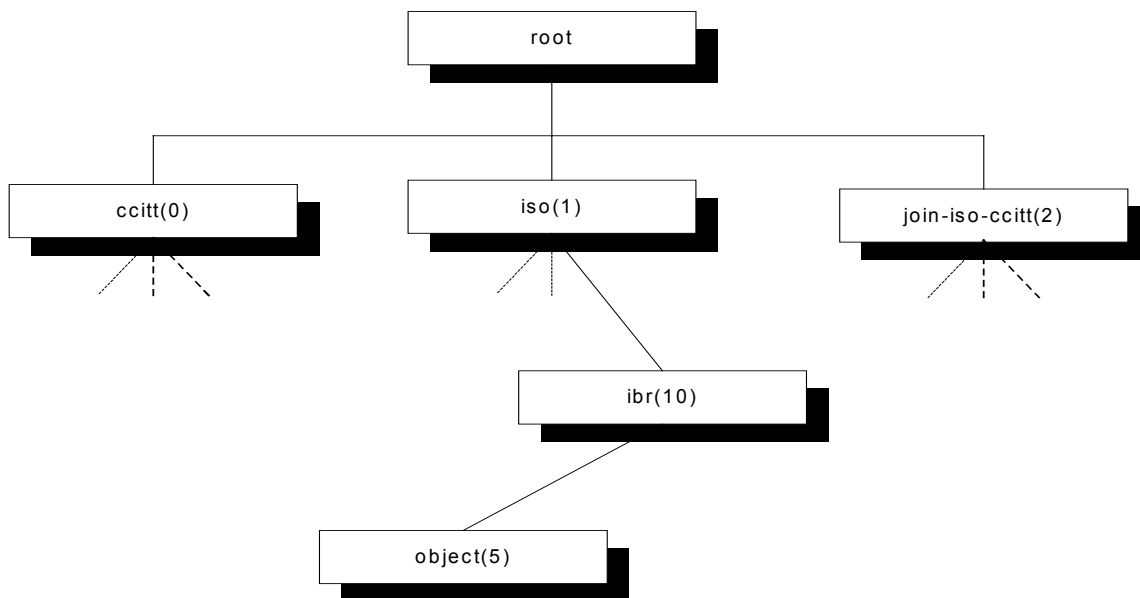
4.3.2.3.1.1.2 Παρακολούθηση

Η παρακολούθηση των δρομολογητών Cisco ΔΥ, πραγματοποιείται με χρήση του πρωτοκόλλου SNMP. Μια μικρή αναφορά στον τρόπο λειτουργία του SNMP έχει ήδη γίνει στην παράγραφο 4.2. Ένας διαχειριστής επικοινωνεί με τους αντιπρόσωπους διαχείρισης, που βρίσκονται στο δικτυακό στοιχείο (βλέπε σχήμα 4.1). Οι αντιπρόσωποι διαχείρισης παρέχουν μια διεπαφή στην MIB του δικτυακού στοιχείου. Στην περίπτωση των δρομολογητών Cisco ΔΥ η MIB (Class-Based QoS MIB) αποτελείται στην πραγματικότητα από δυο MIBs τις: CISCO-CLASS-BASED-QOS-MIB και την CISCO-CLASS-BASED-QOS-CAPABILITY-MIB.

Μια MIB είναι ένας ακριβής ορισμός της πληροφορίας που είναι προσιτή μέσω ενός πρωτοκόλλου διαχείρισης δικτύου. Χρησιμοποιώντας μια ιεραρχημένη και δομημένη μορφή, η MIB ορίζει την πληροφορία διαχείρισης δικτύου που είναι διαθέσιμη από μια συσκευή. Κάθε συσκευή πρέπει να χρησιμοποιεί την μορφή που ορίζεται από τη MIB για την απεικόνιση της πληροφορίας, ώστε να είναι συμβατή με το πρωτόκολλο διαχείρισης δικτύου.

Όλες οι MIB ακολουθούν μια δεντρική μορφή. Αυτή η προτυποποιημένη δεντρική μορφή είναι ορισμένη από δυο οργανισμούς τους: International Telegraph and Telephone Consultative Committee (CCITT) και International Electrotechnical Committee (ISO).

Μια MIB δεν είναι τίποτα παραπάνω από ένα αρχείο ASCII, μέσα στο οποίο ορίζονται μια σειρά από αντικείμενα διαχείρισης μια δικτυακής συσκευής. Ο ορισμός των αντικειμένων γίνεται με χρήση μια γλώσσας, που ονομάζεται Abstract Syntax Notation 1 (ASN.1) [ASN97]. Κάθε αντικείμενο μιας MIB αναγνωρίζεται από ένα αναγνωριστικό αντικειμένου (Object Identifier, OID), το οποίο εντοπίζει ένα αντικείμενο στο δέντρο της MIB. Ένας OID είναι συνήθως μια σειρά από ακέραιους αριθμούς χωρισμένους με τελείες. Σε κάθε κλάδο του δέντρου μιας MIB ανατίθεται ένας ακέραιος αριθμός, καθώς επίσης και ένα όνομα. Στο σχήμα 4.6 παρουσιάζεται η προτυποποιημένη δεντρική μορφή μια MIB, με τους ακέραιους αριθμούς και τα ονόματα που δίνονται σε κάθε κλάδο του δέντρου.



Σχήμα 4.6: Δεντρική μορφή μιας MIB

Ξεκινώντας πάντα από την ρίζα του δέντρου (root), ο OID του αντικειμένου (object) είναι ο εξής: iso.ibr...object ή με χρήση των ακεραίων αριθμών είναι: 1.10....5.

Χρησιμοποιώντας τους ορισμούς των αντικειμένων που υπάρχουν στην MIB, οι αντιπρόσωποι διαχείρισης μπορούν να δημιουργήσουν στιγμιότυπα αυτών των αντικειμένων. Οι διαχειριστές χρησιμοποιούν τα στιγμιότυπα για την παρακολούθηση και διάθρωση των δικτυακών συσκευών. Το πρωτόκολλο SNMP επιτρέπει την επικοινωνία μεταξύ των διαχειριστών και των αντιπρόσωπων διαχείρισης. Η επικοινωνία γίνεται με χρήση μονάδων δεδομένων πρωτοκόλλου (Protocol Data Units-PDU), οι

οποίες είναι ενθυλακωμένες σε πακέτα UDP. Το SNMP χρησιμοποιεί το User Datagram Protocol σαν πρωτόκολλο επίπεδου μεταφοράς. Υπάρχουν τέσσερις διακριτές λειτουργίες οι οποίες μπορούν να πραγματοποιηθούν με χρήση του SNMP πρωτοκόλλου:

- Ο διαχειριστής μπορεί στείλει ένα μήνυμα **GET** στον αντιπρόσωπο διαχείρισης για να συλλέξει την τιμή κάποιας μεταβλητής του υπό διαχείριση αντικειμένου.
- Ο διαχειριστής μπορεί στείλει ένα μήνυμα **GET-NEXT** στον αντιπρόσωπο διαχείρισης για να συλλέξει την ίδια πληροφορία για το επόμενο αντικείμενο από το δέντρο αντικειμένων της συσκευής.
- Ο διαχειριστής μπορεί στείλει ένα μήνυμα **SET** στον αντιπρόσωπο διαχείρισης για να καθορίσει την τιμή κάποιας μεταβλητής του υπό διαχείριση αντικειμένου.
- Ο πράκτορας διαχείρισης μπορεί ασύγχρονα να στείλει ένα μήνυμα **TRAP** στον διαχειριστή, πληροφορώντας τον για κάποιο γεγονός που συνέβη στην υπό διαχείριση συσκευή.

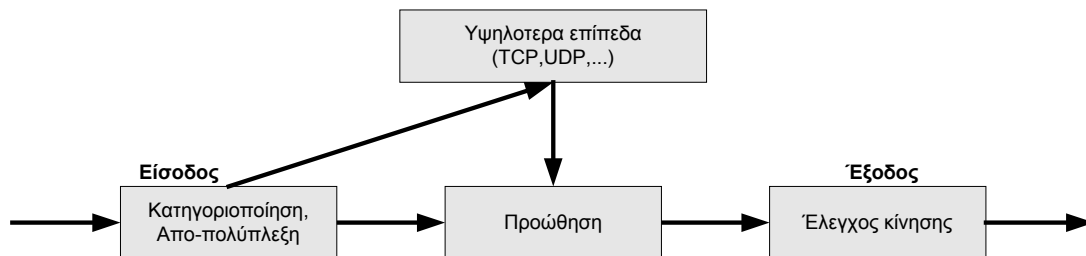
Η Class-Based QoS MIB, που υλοποιείται από τους δρομολογητές Cisco, επιτρέπει μόνο την χρήση μηνυμάτων GET και GET-NET για την παρακολούθηση τους. Ο στατικός πράκτορας του υποσυστήματος προσαρμογής (**Monitoring, CORE_ Monitoring**), συλλέγει την πληροφορία χρησιμοποιώντας SNMP μηνύματα, έπειτα από εντολή ενός ΚΠ παρακολούθησης.

4.3.2.3.1.2 Διαχείριση δρομολογητών Linux για την παροχή διαφοροποιημένων υπηρεσιών

Το Linux είναι ένα λειτουργικό σύστημα “ανοιχτού πηγαίου κώδικα” (open source code). Αρχικά δημιουργήθηκε από τον Linus Torvalds, στις αρχές του 1990, και αποτέλεσε το λειτουργικό σύστημα για προσωπικούς Η/Υ, σχεδιασμένο με βάση την φιλοσοφία του συστήματος UNIX. Το σύστημα αυτό υποστηρίζει εγγενώς την επεξεργασία, δρομολόγηση, και προώθηση IP πακέτων. Έτσι, ένας υπολογιστής εφοδιασμένος με το λειτουργικό σύστημα Linux και με δυο ή περισσότερες κάρτες δικτύου, μπορεί να λειτουργήσει ως δρομολογητής IP δικτύων.

Η διαδικασία με την οποία επεξεργάζεται τα IP πακέτα ένας δρομολογητής Linux παρουσιάζεται στο σχήμα 4.7. Αποτελείται από τρία τμήματα. Αρχικά τα εισερχόμενα

πακέτα κατηγοριοποιούνται και πραγματοποιείται η αποπολύπλεξη των ροών, που απευθύνονται στα υψηλότερα επίπεδα. Τα πακέτα που δεν απευθύνονται σε υψηλότερα επίπεδα (και αυτά που δημιουργούνται από τα επίπεδα αυτά) προωθούνται στο δίκτυο. Η προώθηση περιλαμβάνει την επιλογή του επόμενου κόμβου και της κάρτας δικτύου του δρομολογητή, που είναι κατάλληλη για την αποστολή του πακέτου στον κόμβο αυτόν. Τέλος, για κάθε κάρτα δικτύου υλοποιείται στον πυρήνα του λειτουργικού (kernel) ένα τμήμα έλεγχου εξόδου της κίνησης. Το τμήμα έλεγχου εξόδου αποφασίζει αν τα πακέτα θα τοποθετηθούν σε κάποια ουρά ή αν θα απορριφθούν (αυτό μπορεί να συμβεί αν η ουρά έχει ξεπεράσει κάποιο ανώτατο όριο μήκους ή αν ο ρυθμός άφιξης των πακέτων είναι πολύ μεγάλος).



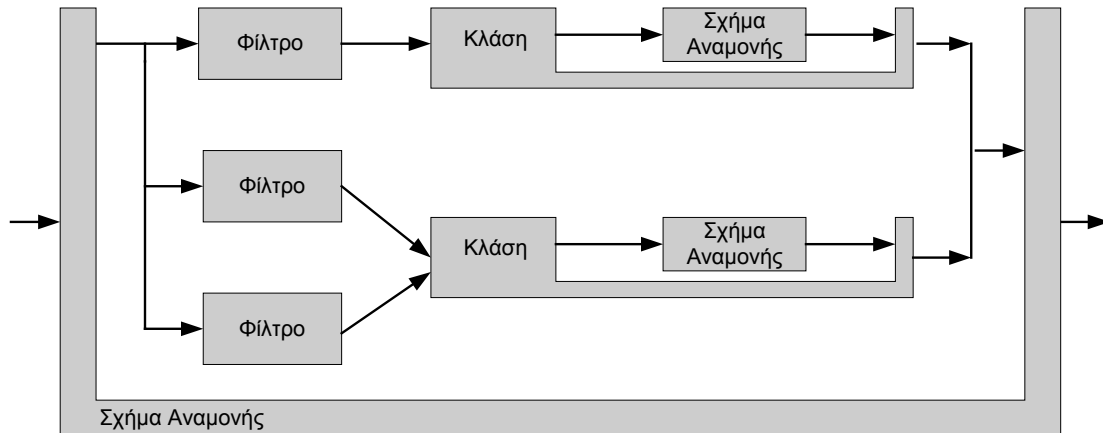
Σχήμα 4.7: Επεξεργασία IP πακέτων από δρομολογητή Linux

Επίσης, αποφασίζει με ποια σειρά θα αποσταλούν τα πακέτα (δίνοντας προτεραιότητα σε ορισμένες ροές) και καθυστερεί την μετάδοση ορισμένων πακέτων, περιορίζοντας έτσι τον ρυθμό μετάδοσης εξερχόμενων πακέτων. Όλες οι παραπάνω λειτουργίες ενός δρομολογητή Linux μπορούν να ελεγχθούν με την χρήση ειδικών προγραμμάτων, που έχουν δημιουργηθεί για το σκοπό αυτό. Τέτοια προγράμματα είναι το *ip* το *ipchains* και το *route*. Ο Alexey Kuznetsov δημιούργησε ένα εργαλείο το οποίο ονομάζεται traffic control (tc), ειδικά για την διαχείριση δρομολογητών Linux, που έχουν την δυνατότητα παροχής ΔΥ [Almesberger99].

Ο λειτουργία έλεγχου κίνησης σε ένα δρομολογητή Linux πραγματοποιείται στο τμήμα εξόδου. Στο τμήμα εισόδου πραγματοποιείται κατά βάση η λειτουργία της κατηγοριοποίησης και απο-πολύπλεξης. Έτσι για την παροχή ΔΥ από έναν δρομολογητή Linux, θα πρέπει να διαρθρωθούν τα τμήματα έλεγχου κίνησης που βρίσκονται στην έξοδο. Το τμήμα έλεγχου κίνησης στη έξοδο ενός δρομολογητή Linux ΔΥ αποτελείται από τα παρακάτω συστατικά στοιχεία:

- Σχήματα αναμονής (queuing disciplines)
- Κλάσεις (classes)
- Φίλτρα (filters)

Κάθε συσκευή δικτύου ενός δρομολογητή Linux είναι συσχετισμένη με ένα σχήμα αναμονής. Το σχήμα αναμονής ελέγχει τον τρόπο με τον οποίο θα εξυπηρετηθούν τα πακέτα, τα οποία βρίσκονται στην ουρά πακέτων της συσκευής. Ένα απλό σχήμα αναμονής μπορεί να αποτελείται από μια και μόνο ουρά, η οποία εξυπηρετεί τα πακέτα με βάση την σειρά που έχουν τοποθετηθεί σε αυτήν (enqueued). Πιο περίπλοκα σχήματα αναμονής μπορεί να χρησιμοποιήσουν φίλτρα για να κατηγοριοποιήσουν τα πακέτα σε κλάσεις και να εξυπηρετήσουν κάθε κλάση διαφορετικά, δίνοντας προτεραιότητα σε κάποιες κλάσεις. Οι κλάσεις με την σειρά τους μπορούν να χρησιμοποιήσουν κάποιο άλλο σχήμα αναμονής για να εξυπηρετήσουν τα πακέτα. Το σχήμα αναμονής της κλάσης μπορεί να χρησιμοποιεί άλλες κλάσεις οι οποίες έχουν τα δικά τους σχήματα αναμονής κ.ο.κ. Ένα τέτοιο σχήμα αναμονής παρουσιάζεται στο σχήμα 4.8.



Σχήμα 4.8: Ένα απλό σχήμα αναμονής με πολλαπλές κλάσεις

Ένα εξωτερικό σχήμα αναμονής, χρησιμοποιώντας δυο φίλτρα, κατευθύνει τα IP πακέτα σε δυο κλάσεις. Κάθε κλάση έχει το δικό της σχήμα αναμονής. Όπως φαίνεται και στο σχήμα 4.8 σε μια κλάση μπορεί να οδηγούνται πακέτα από περισσότερα του ενός φίλτρα.

Στον ακόλουθο πίνακα παρατίθενται τα κυριότερα σχήματα αναμονής, που υλοποιούνται στον πυρήνα του Linux, με μια σύντομη περιγραφή της λειτουργίας τους.

Πίνακας 4.4: Σχήματα αναμονής στον πυρήνα του Linux

Σχήμα Αναμονής	Περιγραφή
First-In-First-Out (FIFO)	Ο αλγόριθμος First-In-First-Out αποθηκεύει τα πακέτα όταν το δίκτυο βρίσκεται σε κατάσταση συμφόρησης και τα προωθεί σύμφωνα τη σειρά άφιξης, όταν το δίκτυο δεν είναι πια σε κατάσταση συμφόρησης.
Random Early Drop (RED)	Μία ουρά με διαχείριση απόρριψης των πακέτων σύμφωνα με τον αλγόριθμο RED [Floyd93]. Ο Αλγόριθμος αυτός χρησιμοποιεί συνθήκες τυχαιότητας, για να εξασφαλίσει ότι όλες οι συνδέσεις θα υποστούν απώλειες με τον ίδιο ρυθμό.
Token Bucket Filter (TBF)	Μια ουρά με διαμόρφωση κίνησης σύμφωνα με τον αλγόριθμο Token Bucket [Bernett02]. Ο αλγόριθμος αυτός είναι ένας μηχανισμός έλεγχου ροής που προσθέτει κουπόνια, ανά περιοδικά χρονικά διαστήματα, σε ένα κάδο και επιτρέπει στα πακέτα να μεταδίδονται από τον αποστολέα μόνο όταν ο κάδος περιέχει τόσα κουπόνια όσα και το μέγεθος του πακέτου.
Generalised Red (GRED)	Αποτελεί μια γενίκευση του αλγόριθμου RED.
Strict Priority (PRIO)	Είναι σχεδιασμένη για να δίνει αυστηρή προτεραιότητα στην πιο σημαντική κίνηση. Η κίνηση εισχωρεί σε μία από τέσσερις ουρές: υψηλής, μεσαίας, κανονικής ή χαμηλής προτεραιότητας. Όταν ο δρομολογητής είναι έτοιμος να μεταδώσει ένα πακέτο, ψάχνει στην ουρά υψηλής προτεραιότητας. Αν υπάρχει εκεί κάποιο πακέτο, το μεταδίδει. Αν όχι, ελέγχεται η ουρά μεσαίας προτεραιότητας ουρά, και αν εκεί υπάρχει πακέτο το μεταδίδει. Αν όχι, ψάχνει την ουρά κανονικής προτεραιότητας, και τέλος την ουρά χαμηλής προτεραιότητας. Η διαδικασία αυτή επαναλαμβάνεται για το επόμενο πακέτο. Αν η κίνηση στην ουρά υψηλότερης προτεραιότητας είναι μεγάλη, τότε οι άλλες ουρές αδρανούν.
Class-Based Queuing (CBQ)	Διαχειρίζεται τις ουρές αναμονής στους δρομολογητές του δικτύου χρησιμοποιώντας κλάσεις κυκλοφορίας και διαφορετικά επίπεδα προτεραιοτήτων (βλ. Παρ 2.7.2.8.1).
DS marking (DSMARK)	Μαρκάρει το πεδίο DSCP σε IP πακέτα.

4.3.2.3.1.3 Το πρόγραμμα tc

Το πρόγραμμα tc εκτελείται στον χώρο χρήστη και επικοινωνεί με το τμήμα έλεγχου κίνησης του πυρήνα του Linux, με σκοπό την διάρθρωση του τμήματος αυτού. Η επικοινωνία επιτυγχάνεται με χρήση των netlink sockets [Dhandapani99]. Τα netlink sockets είναι ένας μηχανισμός επικοινωνίας που επιτρέπει σε διεργασίες που εκτελούνται στον χώρο του χρήστη, να επικοινωνήσουν με τμήματα του πυρήνα. Η γενική σύνταξη του εργαλείου είναι η ακόλουθη:

```
tc [ OPTIONS ] OBJECT { COMMAND | help }
```

όπου:

```
OBJECT := { qdisk | class | filter }
```

```
OPTION := { -s [tatistics] | -d [etails] | -r [aw] }
```

```
COMMAND := { add | del | replace | change | get | ls }
```

Για την διαχείριση ενός σχήματος αναμονής (qdisk) η σύνταξη του εργαλείου είναι η εξής:

```
tc qdisk [ COMMAND ] dev DEVICEID [ handle QHANDLE ]
[ root | parent CLASSID ]
[estimator INTERNAL TIME_CONSTANT ]
[ (QDISK_KIND ) [ help| OPTIONS ] ]
```

ή

```
tc qdisk show [ dev DEVICEID ]
```

όπου:

QDISK_KIND: καθορίζει το είδος του σχήματος αναμονής π.χ. CBQ, TBF, PFIFO κ.λ.π.

DEVICEID: είναι το αναγνωριστικό της δικτυακής συσκευής (π.χ. eth0, eth1).

QHANDLE: είναι το αναγνωριστικό του σχήματος αναμονής.

CLASSID: είναι το αναγνωριστικό της κλάσης στην οποία προσάπτεται το σχήμα αναμονής.

ROOT: είναι το σχήμα αναμονής που αποτελεί την ρίζα για όλα τα αλλά.

Estimator: εκτιμά το εύρος ζώνης που λαμβάνει κάθε κλάση μέσα σε ένα καθορισμένο χρονικό διάστημα (INTERVAL), για να καθορίσει αν η κλάση λαμβάνει το εύρος που της αναλογεί.

Οι παράμετροι (OPTIONS) καθορίζουν τον τρόπο λειτουργίας του σχήματος αναμονής και είναι διαφορετικοί για κάθε σχήμα.

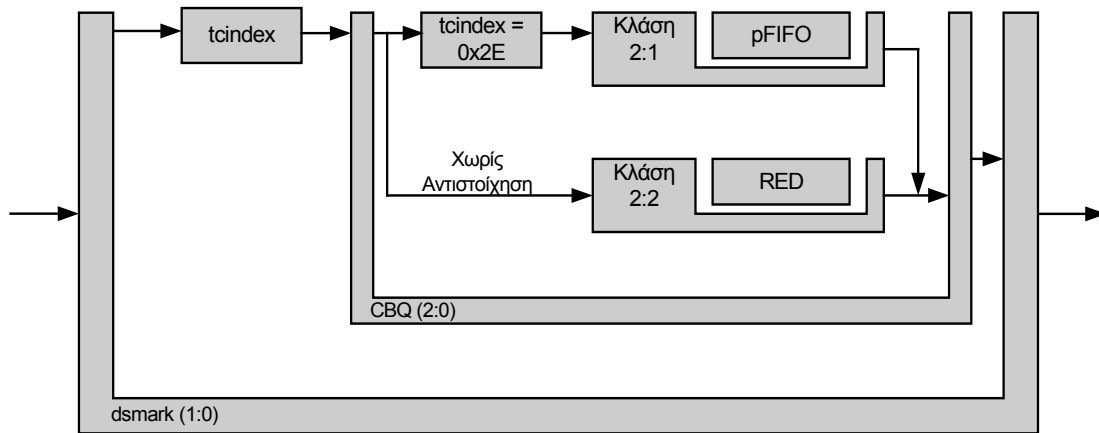
Οι εντολές (COMMAND) καθορίζουν αν το σχήμα αναμονής θα προστεθεί (add) στην δικτυακή συσκευή, θα αφαιρεθεί (del) ή θα αντικατασταθεί (replace).

Στο επόμενο παράδειγμα παραθέτουμε τις αλληπαλλήλες κλήσεις εντολών του tc για την διάρθρωση μιας συσκευής δικτύου (eth0), ώστε να λειτουργεί ως κόμβος κορμού ΔΥ. Ο κόμβος υποστηρίζει την EF PHB και την BE PHB.

1. tc qdisc add dev eth0 handle 1:0 root dsmark indices 64 set_tc_index
2. tc filter add dev eth0 parent 1:0 protocol ip prio 1 tcindex mask 0xfc shift 2
3. tc qdisc add dev eth0 parent 1:0 handle 2:0 cbq bandwidth 10 Mbit allot 1514 cell 8 avpkt 1000 mpu 64
4. tc class add dev eth0 parent 2:0 classid 2:1 cbq bandwidth 10Mbit rate 1500Kbit avpkt 1000 prio 1 bounded isolated allot 1514 weight 1 maxburst 10 defmap 1
5. tc qdisc add dev eth0 parent 2:1 pfifo limit 5
6. tc filter add dev eth0 parent 2:0 protocol ip prio 1 handle 2x0e tcindex classid 2:1 pass_on
7. tc class add dev eth0 parent 2:0 classid 2:2 cbq bandwidth 10Mbit rate 5Mbit avpkt 1000 prio allot 1514 weight 1 maxburst 21 borrow
8. tc qdisc add dev eth0 parent 2:2 red limit 60KB min 15KB max 45KB burst 20 avpkt 1000 bandwidth 10Mbit probability 0.4
9. tc filter add dev eth0 parent 2:0 protocol ip prio 2 handle 0 tcindex mask 0 classid 2:2 pass_on

Η γραμμή 1 θέτει ως σχήμα αναμονής στην ρίζα της συσκευής eth0 το σχήμα dsmark. Το dsmark αποθηκεύει την τιμή του TOS byte των πακέτων σε μια μεταβλητή του πυρήνα, η οποία ονομάζεται tcindex. Η γραμμή 2 προσάπτει ένα φίλτρο στον σχήμα της ρίζας, το οποίο απομονώνει τα έξι πιο σημαντικά ψηφία του TOS byte και τα μεταθέτει κατά δυο προς τα δεξιά. Έτσι, απομονώνεται η καθαρή τιμή του DSCP. Η γραμμή 3 δημιουργεί ένα σχήμα αναμονής τύπου CBQ ως παιδί (child) του σχήματος ρίζας. Στο σχήμα CBQ ορίζονται δυο κλάσεις με αναγνωριστικά 2:1 και 2:2. Η κλάση 2.1 (γραμμή 4) λαμβάνει εγγυημένο εύρος ζώνης μέχρι 1.5 Mbps και ως σχήμα αναμονής της κλάσης ορίζεται το PFIFO (Packet FIFO). Στην κλάση 2:2 (γραμμή 5) ορίζεται ένα ελάχιστο εύρος ζώνης 5 Mbps (γραμμή 7), όμως μπορεί να δανείζεται (borrow) από το συνολικό εύρος ζώνης,

εφόσον αυτό είναι διαθέσιμο. Το σχήμα αναμονής για αυτήν την κλάση είναι το RED. Έτσι, η κλάση 2:2 μπορεί να δανειστεί μέχρι και 3.5 Mbps, ώστε να μη επηρεάζει την κλάση 2:1, της οποίας το εύρος ζώνης είναι εγγυημένο (isolated). Το φίλτρο που ορίζεται στην γραμμή 6 κατατάσσει τα πακέτα με τιμή DSCP ίση με το δεκαεξαδικό 2E (τιμή του DSCP για την EF PHB) στην κλάση 2:1. Το φίλτρο που ορίζεται στην γραμμή 9 κατατάσσει όλα τα υπόλοιπα πακέτα στην κλάση 2:2. Η κλάση 2:1 θα εξυπηρετείται πάντα πρώτη, σε περίπτωση που υπάρχουν πακέτα στην ουρά, λόγο του ότι έχει μεγαλύτερη προτεραιότητα (prio 1) από την κλάση 2:2 (prio 8). Το αποτέλεσμα της παραπάνω διάρθρωσης στο τμήμα έλεγχου κίνησης του πυρήνα παρουσιάζεται στο σχήμα 4.9.



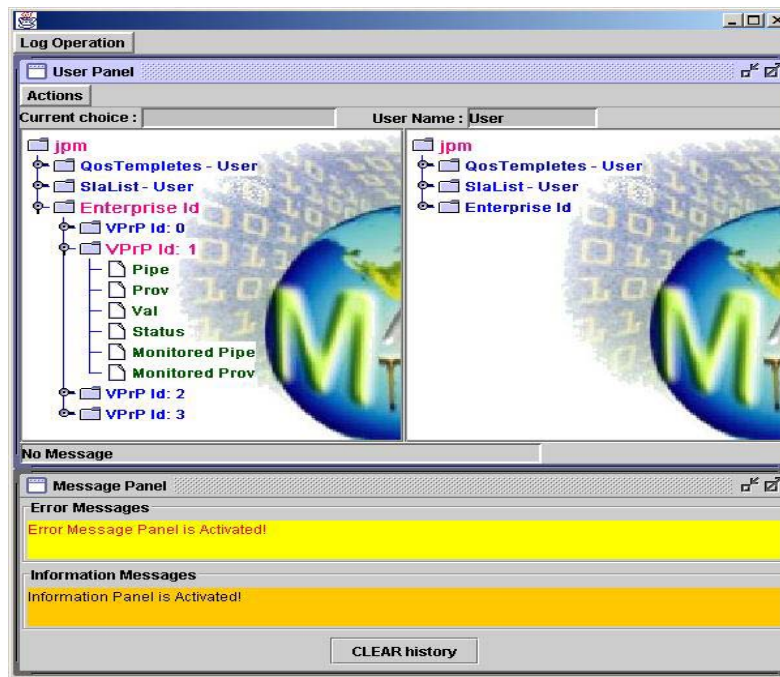
Σχήμα 4.9: Διάρθρωση δρομολογητή για παροχή EF PHB και EB PHB με χρήση του εργαλείου tc.

Οι πράκτορες του υποσυστήματος προσαρμογής, που είναι υπεύθυνοι για την διάρθρωση των δρομολογητών Linux, παρέχουν μια διεπαφή στο εργαλείο tc. Με βάση τις εντολές, που μεταφέρουν σε αυτούς οι ΚΠ διάρθρωσης, εκτελούν μια σειρά από εντολές tc στον δρομολογητή, επιτυγχάνοντας έτσι την διάρθρωση του.

Η εκτέλεση της εντολής tc qdisk show [dev DEVICEID] παρέχει πληροφορία για τον τρόπο με τον οποίο έχει αντιμετωπίσει τα πακέτα, των διάφορων κλάσεων υπηρεσίας, η συγκεκριμένη συσκευή. Ο πράκτορας του υποσυστήματος προσαρμογής, που είναι υπεύθυνος για την παθητική παρακολούθηση, φιλτράρει την πληροφορία αυτή και παραδίδει στον ΚΠ παρακολούθησης το εύρος ζώνης και τις απώλειες πακέτων, για κάθε κλάση υπηρεσίας που εξυπηρετείται από την δικτυακή συσκευή.

4.3.3 Υποσύστημα εφαρμογής

Το υποσύστημα αυτό αποτελείται από μια σειρά από γραφικές διεπαφές χρήστη (Graphical User Interfaces-GUI), μέσω των οποίων οι διαχειριστές και οι χρήστες έχουν πρόσβαση στις λειτουργίες του συστήματος. Το γραφικό περιβάλλον που παρέχεται σε έναν απλό χρήστη παρουσιάζεται στην εικόνα 4.4.



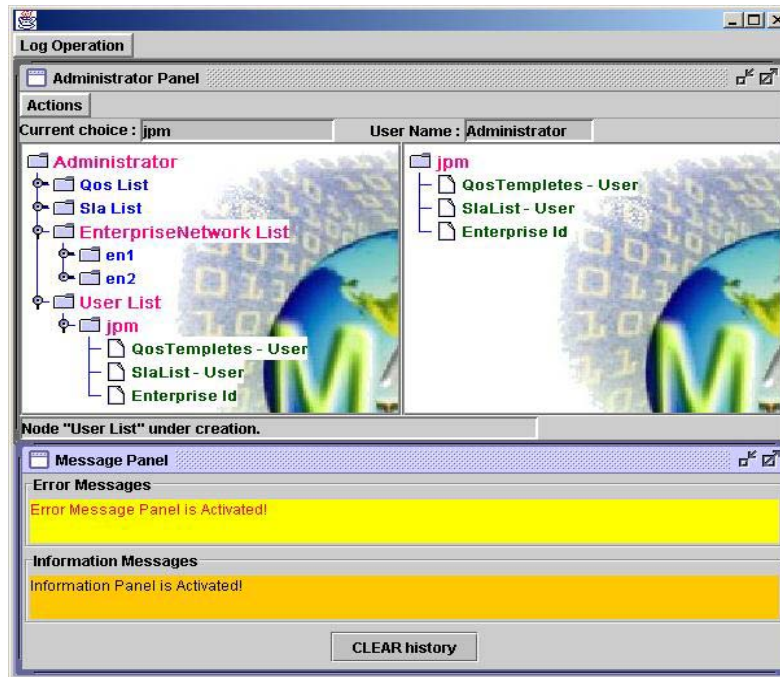
Εικόνα 4.4: Γραφικό περιβάλλον χρήστη

Μεσώ του γραφικού περιβάλλοντος ο χρήστη μπορεί:

- Να έχει πρόσβαση στις κλάσεις ΠοιΥπ, που έχουν οριστεί από τον διαχειριστή.
- Να μεταβάλει συγκεκριμένες παραμέτρους (μόνο τα πεδία που του επιτρέπεται να αλλάξει ως απλός χρήστης).
- Να επιλέξει ένα από τα SLA που διαθέτει.
- Να δημιουργήσει VPrPs.
- Να παρακολουθήσει την κατάσταση των VPrPs που δημιούργησε.
- Να αποδεσμεύσει τα VPrPs του.

Στην εικόνα 4.5 παρουσιάζεται το γραφικό περιβάλλον, που είναι διαθέσιμο στους διαχειριστές του συστήματος. Οι διαχειριστές μπορούν να εκτελέσουν όλες τις λειτουργίες ενός απλού χρήστη. Επιπλέον, έχουν την δυνατότητα να:

- Διαχειρισθούν την βάση δεδομένων προσθέτοντας/αφαιρώντας/μετατρέποντας τις κλάσεις και φόρμες ΠοιΥπ, τα SLAs, τα σημεία πρόσβασης των χρηστών και τους δρομολογητές του δικτύου.
- Δημιουργήσουν λογαριασμούς νέων χρηστών.
- Διαγράψουν χρηστές.
- Παρακολουθήσουν την κατάσταση του υπό διαχείριση δικτύου.



Εικόνα 4.5: Γραφικό περιβάλλον διαχειριστή

Στην εικόνα 4.6 παρουσιάζεται το γραφικό περιβάλλον που επιτρέπει την δημιουργία ενός SLA από τον διαχειριστή. Σε ένα SLA ορίζεται η χρονική περίοδος της σύνδεσης και το είδος εξυπηρέτησης, που δικαιούται η κίνηση που παράγεται από τον χρήστη. Η κίνηση μαρκάρεται σε τρία διαφορετικά προφίλ, ανάλογα με το εαν έχει ξεπεράσει κάποιο συγκεκριμένο ρυθμό μετάδοσης (Three Color Marker-TCM). Ο τρόπος λειτουργίας του TCM ειχει ήδη περιγραφει στην παραγραφο 2.7.2.1. Ανάλογα με προφίλ, στο οποίο ανήκει η κίνηση, λαμβάνει διαφορετική εξυπηρέτηση (PHB, καθυστέρηση, απόκλιση καθυστέρησης, απώλειες)

Create new QoS

Description Info | Pipe Info | Provision Info | Validity Info | State of New QoS

Information of new QoS

Description information

Name: MyQos
Description: Medium Qos
dscp: 110001

Pipe Information

Directionality: Bi

Forward Load

Min: 5000
Max: 300000
Mean: 150000
Burst: 200
Measur.int: 10000
Band.share: 500
band.Weight: 500

Reverse Load

Min: 5000
Max: 300000
Mean: 150000
Burst: 200
Measur.int: 10000
Band.share: 500
band.Weight: 500

Provision Information

Excess Green Load Action

GreenAct: TRANSMIT
GreenTemp: AF11

Jitter Info

Min: 500
Max: 1000
Mean: 600
Measur.Per: 10000
Priority: 0

Excess Yellow Load Action

YellowAct: TRANSMIT
YellowTemp: AF11

Delay Info

Min: 0
Max: 5000
Mean: 4000
Measur.Per: 10000
Priority: 0

Excess Red Load Action

RedAct: DROP
RedTemp: AF13

Loss Info

Min: 50
Max: 1000000
Mean: 100
Measur.Per: 10000
Priority: 0

Validity Information

Valid From: 2002-02-04 23:00:00
Valid Period(sec): 1000
Daily valid from(HH:MM): 10:00
Daily valid period(sec): 6000
Available Days: Tue/Fri/
Available Months: May/Jun/Aug/Sep/

CREATE CANCEL

Εικόνα 4.6: Δημιουργία SLA

Στην εικόνα 4.7 παρουσιάζεται το γραφικό περιβάλλον που επιτρέπει στον χρήστη να δημιουργήσει ένα VPrP. Με χρήση αυτού του γραφικού περιβάλλοντος, ο χρήστης μπορεί να δημιουργήσει μια VPrP σύνδεση καθορίζοντας τις παρακάτω παραμέτρους:

- Την κλάση υπηρεσίας που επιθυμεί. Η εφαρμογή παρέχει τέσσερις διαφορετικές κλάσεις υπηρεσίας, χωρίς αυτό να αποκλείει την δυνατότητα δημιουργίας άλλων κλάσεων Ξεκινώντας από την καλύτερη, αυτές είναι: η Χρυσή (Gold), η Αργυρή (Silver), η Χάλκινη (Bronze) και η καλύτερης προσπάθειας (Best effort).. Σε κάθε κλάση η κίνηση κατατάσσεται σε τρία διαφορετικά προφίλ με χρήση ενός Three Colour Marker. Στην εικόνα παρατηρούμε ότι για την χρυσή κλάση υπηρεσίας, η πράσινη κίνηση εξυπηρετείται με βάση το PHB EF, η κιτρίνη με βάση το PHB AF11 και η κόκκινη με βάση το BE PHB.

- Τα σημεία πρόσβασης (SAPs) πηγής και προορισμού.
- Την Κατευθυντικότητα της σύνδεσης (μονοκατευθυντήρια –unidirectional και αμφίδρομη- bidirectional).
- Το εγγυημένο εύρος ζώνης που επιθυμεί.
- Τον χρόνο δημιουργίας και χρονικό διάστημα της σύνδεσης.

Στο παράδειγμα της εικόνας 4.7, ένας χρήστης με SLA JPMSla ζητά μια αμφίδρομη σύνδεση. μεταξύ δυο σημείων (Sap1 και Sap2). Η σύνδεση αυτή θα πραγματοποιείται κάθε Τρίτη για δυο ώρες, από τις 10 μμ μέχρι τις 12 μμ, ξεκινώντας από τις 01/12/2002, για όλον τον χρόνο.

Create a new VPrP

<i>Sla ID</i>	JPMSla
<i>Originating Sap</i>	Sap1
<i>Destination Sap</i>	Sap2
<i>Directionality</i>	Bi

Forward	Reverse
<i>Class Of Service</i> BE	<i>Class Of Service</i> GOLD
<i>Min Bandwidth</i> 5000	<i>Min Bandwidth</i> 5000
<i>Mean Bandwidth</i> 150000	<i>Mean Bandwidth</i> 150000
<i>Maximum Band...</i> 300000	<i>Maximum Band...</i> 300000
<i>Burst Size</i> 200	<i>Burst Size</i> 200

Forward	Reverse
<i>Green Templ</i> BE <i>Green Act</i> TRANSMIT	<i>Green Templ</i> EF <i>Green Act</i> TRANSMIT
<i>Yellow Templ</i> BE <i>Yellow Act</i> TRANSMIT	<i>Yellow Templ</i> AF11 <i>Yellow Act</i> REMARK
<i>Red Templ</i> BE <i>Red Act</i> TRANSMIT	<i>Red Templ</i> BE <i>Red Action</i> DROP

<i>Valid From</i>	12-01-2002 00:00:00
<i>Validity Period (sec)</i>	259200
<i>Daily Valid From (HH:MM)</i>	10:00
<i>Daily Validity Period (sec)</i>	7200

☐ Mondays ☒ Tuesdays ☐ Wednesdays ☐ Thursdays ☐ Fridays ☐ Saturdays ☐ Sundays	☒ January ☒ March ☒ May ☒ July ☒ September ☒ November	☒ February ☒ April ☒ June ☒ August ☒ October ☒ December
---	--	--

Εικόνα 4.7: Δημιουργία VPrP

Ο χρήστης επιθυμεί να στέλνει κίνηση με κλάση υπηρεσίας BE και να δέχεται κίνηση με την χρυσή κλάση υπηρεσίας. Για την χρυσή κλάση υπηρεσίας, η κίνηση μεταξύ των τιμών του ελάχιστου εύρους ζώνης (Min bandwidth) και μέσου εύρους ζώνης (Mean bandwidth), χαρακτηρίζεται ως πράσινη. Η κίνηση μεταξύ των τιμών μέσου εύρους ζώνης και μέγιστου εύρους ζώνης (maximum bandwidth), χαρακτηρίζεται ως κιτρινή και αυτή που ξεπερνά το μέγιστο εύρος ζώνης, χαρακτηρίζεται ως κόκκινη.

Στην εικόνα 4.8 παρουσιάζεται το γραφικό περιβάλλον, μέσα από το οποίο ο χρήστης μπορεί να παρακολουθήσει την κατάσταση μιας σύνδεσης του. Συγκεκριμένα, η εικόνα παρουσιάζει τις παραμέτρους παθητικής παρακολούθησης (εύρος ζώνης και απώλειες), ενώ κατά αντίστοιχο τρόπο παρουσιάζονται και οι παράμετροι ενεργητικής παρακολούθησης (καθυστέρηση και απόκλιση καθυστέρησης).

The screenshot displays a 'Monitoring' window with the following sections:

- Service Origin:** Sap1
- Service Destination:** Sap2
- Directionality:** Bi
- Passive / Active:** The 'Passive' tab is selected.
- Passive Monitoring:**
 - Bandwidth Parameters:**

	FW	RV
Min	5000	5000
Max	300000	300000
Mean	150000	150000
Burst	200	200
 - Loss Parameters:**

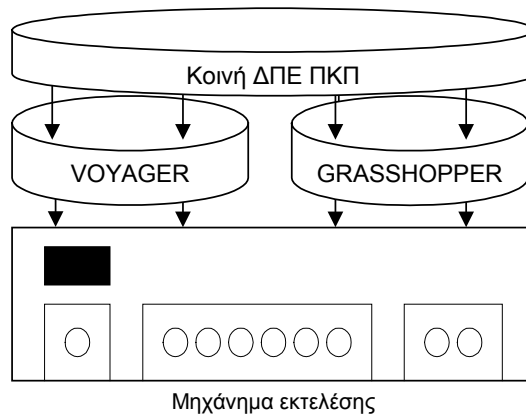
	FW	RV
Min	10	50
Max	1000000	1000000
Mean	20	100
- Monitoring Handler:**
 - Parameters:**
 - Polling period(ms): 10000
 - Report period(ms): 10000
 - Buttons: Reset, Modify
 - Actions:**
 - Buttons: Start, Stop

Εικόνα 4.8: Παθητική παρακολούθηση σύνδεσης

Ο χρήστης είναι σε θέση να ξεκινήσει και να σταματήσει την παρακολούθηση οποία στιγμή επιθυμεί. Επίσης μπορεί να ορίσει το χρονικό διάστημα ερωταπόκρισης (polling period), της παρακολούθησης.

4.3.4 Κοινή διεπαφή - ΠΚΠ (Common MA API)

Ένα από τα μεγαλύτερα προβλήματα που αντιμετωπίζουν οι εφαρμογές ΚΠ, είναι η δέσμευση της λειτουργίας τους με μια συγκεκριμένη ΠΚΠ. Στο τρίτο κεφαλαίο αναφέρθηκε ότι η OMG έχει δημιουργήσει το πρότυπο MASIF, το οποίο επιτρέπει την αλληλεπίδραση μεταξύ πρακτόρων διαφορετικών πλατφόρμων. Δυστυχώς το πρότυπο αυτό σήμερα υποστηρίζεται μόνο από μια ΠΚΠ (grasshopper). Οι δυο πλατφόρμες που κυριαρχούν στην δημιουργία εφαρμογών ΚΠ είναι η Grasshopper και η Voyager. Η αλληλεπίδραση πρακτόρων των δυο αυτών ΠΚΠ είναι σήμερα αδύνατη. Για να περιοριστεί την δέσμευση της εφαρμογής διαχείρισης δικτύου με μια από τις δυο ΠΚΠ, αναπτύχθηκε μια ΔΠΕ, που επιτρέπει την εκτέλεση των πρακτόρων και στις δυο πλατφόρμες, με ελάχιστη μεταβολή στον κώδικα. Αυτό φυσικά δεν σημαίνει ότι στοιχεία των δυο πλατφόρμων μπορούν να επικοινωνούν μεταξύ τους. Η κοινή ΔΠΕ ΠΚΠ δρα ως μια ξεχωριστή πλατφόρμα πάνω από τις ΠΚΠ, όπως φαίνεται και στο σχήμα 4.9. Με χρήση της ΔΠΕ, ο χρήστης είναι σε θέση να επιλέξει ποια πλατφόρμα επιθυμεί να χρησιμοποιήσει για την λειτουργία του συστήματος.



Σχήμα 4.9 λειτουργία Κοινής ΔΠΕ ΠΚΠ

Τόσο η πλατφόρμα Grasshopper όσο και η Voyager είναι υλοποιημένες στην γλώσσα προγραμματισμού Java. Έτσι η ΔΠΕ αποτελείται από δυο κλάσεις της Java, μια για κάθε πλατφόρμα. Η επιλογή, σε ποια πλατφόρμα θα εκτελεστεί ο πράκτορας, γίνεται με επιλογή μιας από τις δυο κλάσεις ως μητρική κλάση του πράκτορα. Το μόνο σημείο του κώδικα που πρέπει να μετατραπεί, για να μπορέσει ο πράκτορας να εκτελεστεί σε διαφορετική πλατφόρμα, είναι η μητρική κλάση.

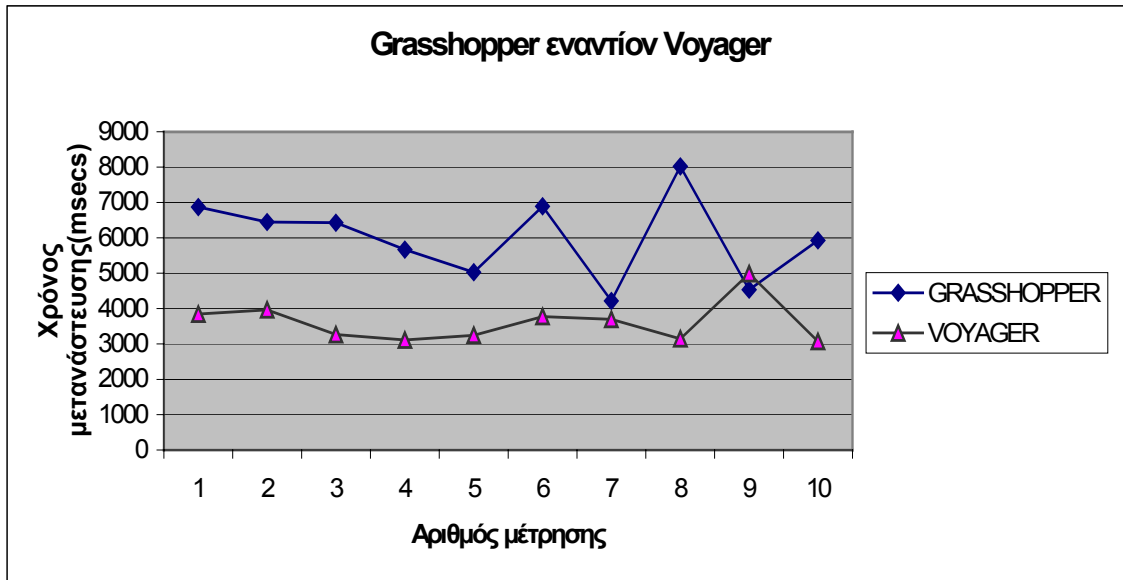
Στον επόμενο πίνακα παρουσιάζονται οι μέθοδοι που περιέχονται στις δυο κλάσεις, που αποτελούν την κοινή ΔΠΕ.

Μέθοδος	Περιγραφή
InitMe (AgentName, Host, Port)	Η μέθοδος αυτή αρχικοποιεί τον πράκτορα.
MoveMe(Destination, Port)	Μετακινεί τον πράκτορα σε ένα απομακρυσμένο μηχάνημα. Η πληροφορία που είναι απαραίτητη για την μετακίνηση ενός πράκτορα, δεν περιορίζεται μόνο το όνομα του μηχανήματος, αλλά περιλαμβάνει και η θύρα (Port) στην οποία “άκούει” η πλατφόρμα.
live()	Είναι η μέθοδος η οποία καλείται αυτόματα μετά την δημιουργία και μετά από κάθε μετακίνηση του πράκτορα.
setSecurity(Read, Write, Exec)	Η μέθοδος αυτή καθορίζει τα δικαιώματα του πράκτορα (ανάγνωση αρχείου, έγγραφη αρχείου, εκτέλεση εφαρμογής), στο μηχάνημα που εκτελείται.
SetPersistence(AgentName, Port)	Αυτή η μέθοδος καθορίζει αν ο πράκτορας είναι επίμονος. Αν δηλαδή ανακατασκευάζεται μετά από ένα σφάλμα του συστήματος, ώστε να συνεχίσει την λειτουργία από εκεί που σταμάτησε.
CreateRemoteProxy(Location, AgentName, Port, InterfaceName)	Δημιουργεί ένα αντιπρόσωπο ενός απομακρυσμένου πράκτορα. Με αυτήν την μέθοδο είναι δυνατή η απομακρυσμένη κλήση μεθόδου στον πράκτορα.
CreateProxy(AgentName, Port, InterfaceName)	Δημιουργεί ένα αντιπρόσωπο ενός πράκτορα που βρίσκεται στο ίδιο μηχάνημα. Αυτή η μέθοδος επιτρέπει την κλήση μεθόδου ενός πράκτορα.

Οι λειτουργίες που υλοποιούνται από τις παραπάνω μεθόδους είναι αρκετές για να υποστηρίξουν μια εφαρμογή ΚΠ.

Στο διάγραμμα 4.1, συγκρίνεται ο χρόνος μετανάστευσης ΚΠ, για τις δυο πλατφόρμες. Ο πράκτορας είναι υλοποιημένος χρησιμοποιώντας την κοινή διεπαφή και επισκέπτεται τρία διαφορετικά μηχανήματα, χωρίς να εκτελεί κάποια εργασία σε αυτά. Παρατηρούμε ότι η ΠΚΠ Voyager παρουσιάζει καλύτερους χρόνους μετανάστευσης. Όμως η δημιουργία πρακτόρων με χρήση της πλατφόρμας Grasshopper είναι αρκετά πιο εύκολη. Επίσης, η εν λόγω πλατφόρμα είναι η μόνη που ακολουθεί όλα τα πρότυπα του Foundation for Intelligent and Physical Agents (FIPA) [FIPA].

Ένα άλλο σημείο που αξίζει να παρατηρηθεί, είναι ότι οι χρόνοι μεταναστεύσεις δεν είναι σταθεροί για τις δέκα μετρήσεις που πραγματοποιήθηκαν, παρόλο που ο πράκτορας πραγματοποιεί το ίδιο ταξίδι. Αυτό οφείλεται στο ότι, ο χρόνος μετανάστευσης ενός πράκτορα εξαρτάται από την κατάσταση του υποκείμενου δικτύου, η οποία δεν είναι πάντα η ίδια.



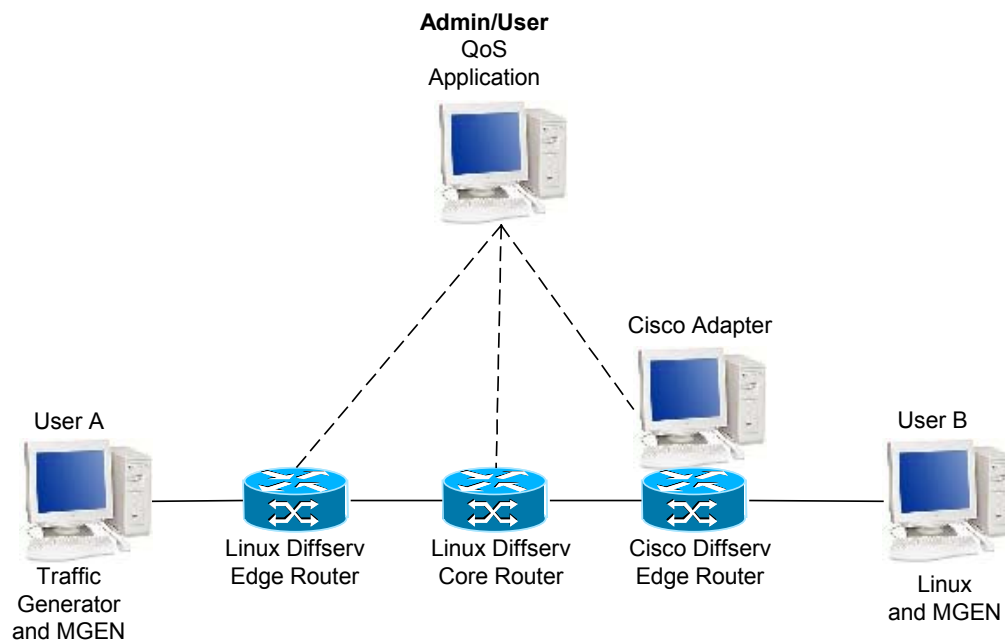
Διάγραμμα 4.1: σύγκριση χρόνου μετανάστευσης ΠΚΠ Grasshopper και Voyager

4.4 Πειραματική αξιολόγηση του συστήματος διαχείρισης

Στην ενότητα αυτή, το σύστημα διαχείρισης που παρουσιάστηκε, αξιολογείται ως προς την ικανότητα του να παρέχει την απαιτούμενη ΠοιΥπ. Το σύστημα θα πρέπει να είναι σε θέση να προσφέρει εγγυήσεις για το εύρος ζώνης και την καθυστέρηση μετάδοσης κάθε σύνδεσης, σύμφωνα με την κλάση υπηρεσίας που χρησιμοποιείται. Έτσι οι παράμετροι που λαμβάνονται υπόψη για την αξιολόγηση του συστήματος είναι:

- Η κατανομή του εύρους ζώνης
- Η καθυστέρηση μετάδοσης

Η αξιολόγηση του συστήματος στηρίζεται σε πραγματικές μετρήσεις. Το πειραματικό IP δίκτυο Δ.Υ (IP DiffSeRv network) που χρησιμοποιήθηκε για τον σκοπό αυτό παρουσιάζεται στο σχήμα 4.10.



Σχήμα 4.10: Πειραματικό IP δίκτυο ΔΥ

Το δίκτυο αποτελείται από τρεις δρομολογητές. Ο χρήστης Α (User A) είναι συνδεδεμένος με έναν ακραίο δρομολογητή τύπου Linux (Linux DiffServ Edge Router). Ο χρήστης Β (User B) είναι συνδεδεμένος με έναν ακραίο δρομολογητή τύπου Cisco (Cisco DiffServ Edge Router). Οι δυο ακραίοι δρομολογητές συνδέονται με έναν δρομολογητή κορμού, τύπου Linux (Linux DiffServ Core Router).

Στη συνέχεια παρουσιάζονται τα ιδιαίτερα χαρακτηρίστηκα των τμημάτων του πειραματικού περιβάλλοντος

Edge and Core DifServ Linux Routers:

- ❑ Mandrake 8.0 + JDK1.3.1
- ❑ Grasshopper 2.1 MA Platform+ Common API
- ❑ TC Tool
- ❑ Linux wrapper + scheduler

Administrator + QoS Application:

- ❑ Mandrake 8.0 + JDK1.3.1
- ❑ Grasshopper 2.1 MA Platform + Common API
- ❑ MySQL database
- ❑ QoS application

Edge DifServ Cisco Router:

- ❑ Router Cisco 7200
- ❑ IOS V12.2-4.T3

Cisco Adapter:

- ❑ Mandrake 8.0 + JSDK1.3.1
- ❑ Cisco wrapper + scheduler
- ❑ Grasshopper 2.1 MA Platform+ Common API

User A and User B:

- ❑ Mandrake 8.0
- ❑ MGEN v3.2a1

Στην περίπτωση των δρομολογητών Linux το απαραίτητο λογισμικό για την λειτουργία του συστήματος (εργαλείο tc, η ΠΚΠ Grasshopper και η κοινή διεπαφή ΠΚΠ, οι πράκτορες του υποσυστήματος προσαρμογής και όλοι οι κινητοί πράκτορες που θα επισκεφτούν τους δρομολογητές), εκτελούνται στο μηχάνημα του δρομολογητή. Στην περίπτωση του δρομολογητή Cisco, το λογισμικό αυτό εκτελείται σε ένα μηχάνημα που βρίσκεται κοντά στον δρομολογητή.

Το σύστημα διαχείρισης (QoSApplication) είναι σε θέση να διαρθρώσει τους δρομολογητές του δικτύου, ώστε να παρέχουν συγκεκριμένη ΠοιΥπ στην κίνηση που παράγεται από τους δυο χρηστές του συστήματος. Για την παραγωγή της κίνησης χρησιμοποιείται το εργαλείο MGEN [MGEN], το οποίο μια γεννήτρια κίνησης με δυνατότητα μαρκαρίσματος του DSCP πεδίου των πακέτων, ώστε να, αντιστοιχούν σε διαφορετικές κλάσεις υπηρεσίας.

Αφού ολοκληρωθεί το προπαρασκευαστικό στάδιο το οποίο περιλαμβάνει:

- Την εισαγωγή των απαραίτητων στοιχείων στην βάση δεδομένων του συστήματος.
- Την αρχική διάθρωση των δρομολογητών με αποστολή των πρακτόρων προπαρασκευής (NetConfigAgents).
- Την αποστολή των πρακτόρων του υποσυστήματος προσαρμογής (wrapper+scheduler) στους δρομολογητές.

Είναι πλέον δυνατή η διάθρωση του δικτύου, με χρήση του συστήματος.

4.4.1 Κατανομή εύρους ζώνης

Βασική ικανότητα ενός συστήματος παροχής ΠοιΥπ είναι η εγγυημένη κατανομή του εύρους ζώνης του δίκτυο, στις διαφορετικές ροές κίνησης. Το εύρος ζώνης (bandwidth) ορίζεται ως το πόσο δεδομένων που μεταδίδονται από το δίκτυο μέσα σε ένα συγκεκριμένο χρονικό διάστημα (εξίσωση 4.1) και εκφράζεται σε bits per second (bps).

$$\text{Bandwidth} = \frac{\text{transmitted data}}{\text{time}} \frac{\text{bits}}{\text{seconds}} \quad (4.1)$$

Η κατανομή αυτή γίνεται με χρήση διάφορων χρονοπρογραμματιστών, οι σημαντικότεροι από τους οποίους έχουν παρουσιαστεί στην παράγραφο 2.7.2.8. Ο χρονοπρογραμματιστής που χρησιμοποιείται από την εφαρμογή είναι ο Class based Queuing (CBQ), η λειτουργία του οποίου έχει παρουσιαστεί αναλυτικά στην παράγραφο 2.7.2.8.1. Περιληπτικά αναφέρουμε ότι σύμφωνα με τους μηχανισμούς του CBQ, οι διαφορετικές κλάσεις υπηρεσίας λαμβάνουν εγγυημένα ένα τμήμα από το συνολικό εύρος ζώνης της γραμμής, διατηρώντας την ικανότητα να το υπερβούν δανειζόμενες από το εύρος ζώνης άλλων κλάσεων, με την προϋπόθεση ότι αυτό δεν χρησιμοποιείται από

αυτές. Τόσο οι δρομολογητές Linux όσο και δρομολογητές CISCO υποστηρίζουν την δρομολόγηση με βάση τον χρονοπρογραμματιστή CBQ.

Για την διεξαγωγή των πειραμάτων, το πειραματικό δίκτυο (συνολικού εύρους ζώνης 10 Mbps) διαρθρώθηκε με χρήση του συστήματος διάθρωσης, έτσι ώστε να παρέχει εγγυημένο εύρος ζώνης σε τρεις διαφορετικές κλάσεις κίνησης με τα παρακάτω χαρακτηριστικά.

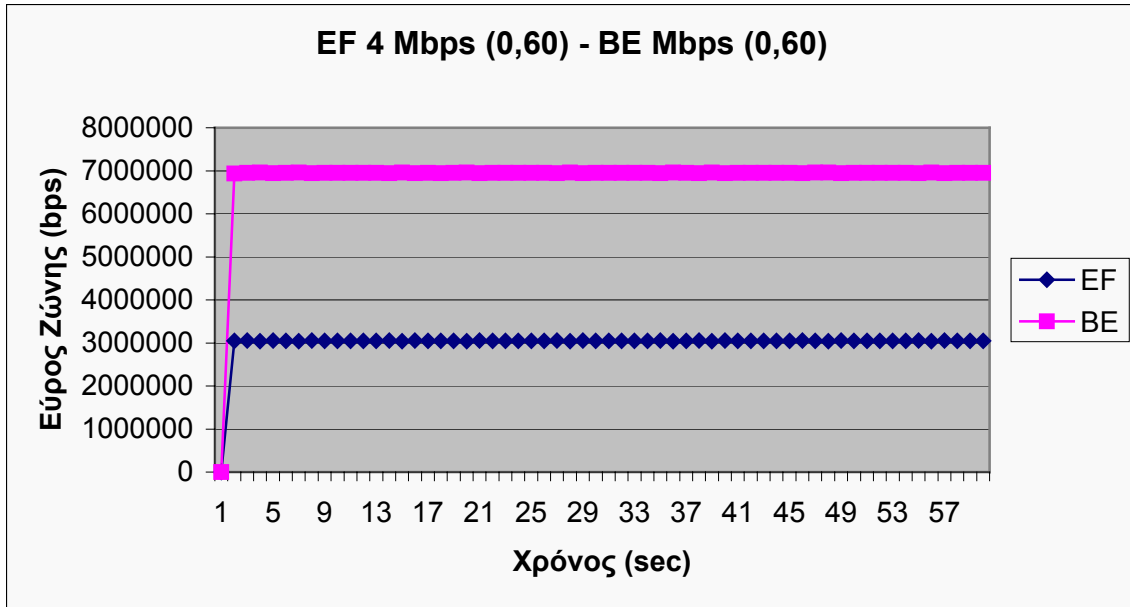
- EF με 3 Mbps εύρος ζώνης (isolated) περιορισμένο, δηλαδή χωρίς την δυνατότητα δανεισμού εύρους ζώνης από άλλες κλάσεις.
- AF11 με 2 Mbps με δυνατότητα δανεισμού από άλλες κλάσεις.
- AF31 με 5 Mbps με δυνατότητα δανεισμού από άλλες κλάσεις.

Η παραγωγή της κίνησης γίνεται με χρήση του εργαλείου MGEN, με κατεύθυνση από τον χρήστη Α προς τον χρήστη Β. Για την μέτρηση του εύρους ζώνης χρησιμοποιήθηκε το υποσύστημα παρακολούθησης, με περίοδο παρακολούθησης 1 sec.

Στην συνέχεια παρατίθενται τα αποτελέσματα μιας σειράς πειραμάτων, στα οποία αξιολογείται ο τρόπος με τον οποίο το σύστημα κατανέμει το εύρος ζώνης του δικτύου, στις διαφορετικές κλάσεις κίνησης, κάτω από διαφορετικές συνθήκες κίνησης. Σε αυτό το σημείο πρέπει να τονίσουμε ότι κατά την διεξαγωγή των πειραμάτων η επιπλέον κίνηση κάθε ροής, απορρίπτεται και δεν επαναμαρκάρεται, σε διαφορετική κλάση υπηρεσίας. Με αυτόν τον τρόπο, δεν υπάρχει διαρροή κίνησης από την μια κλάση υπηρεσίες στην άλλη.

Πείραμα 1

Κλάση υπηρεσίας	Παραγόμενη κίνηση(Mbps)	Χρόνος μετάδοσης(sec)
EF	4	0 → 60
BE	8	0 → 60

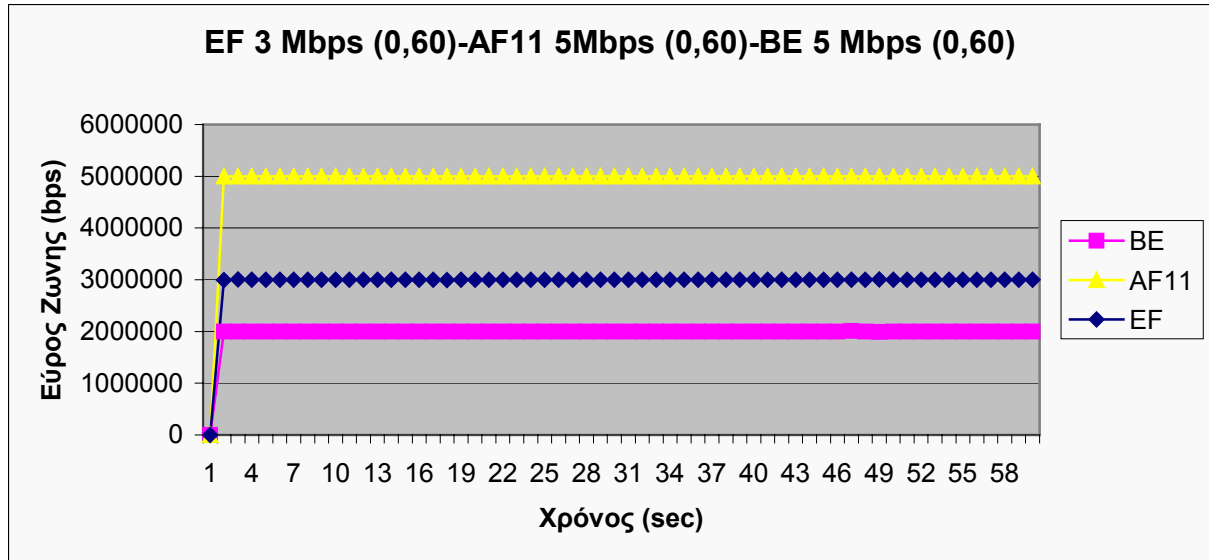


Διάγραμμα 4.2: Κατανομή εύρους ζώνης για κίνηση EF 4 Mbps (0 → 60) sec και BE 8 Mbps (0 → 60) sec

Η κίνηση EF λαμβάνει το 3 Mbps που τις αντιστοιχούν και το επιπλέον 1 Mbps απορρίπτεται. Η κίνηση BE 8 Mbps (0 → 60) sec λαμβάνει τα υπόλοιπα 7 Mbps του δίκτυο και 1 Mbps αυτής της κίνησης, επίσης απορρίπτεται.

Πείραμα 2

Κλάση υπηρεσίας	Παραγόμενη κίνηση(Mbps)	Χρόνος μετάδοσης(sec)
EF	3	0 → 60
AF11	5	0 → 60
BE	5	0 → 60

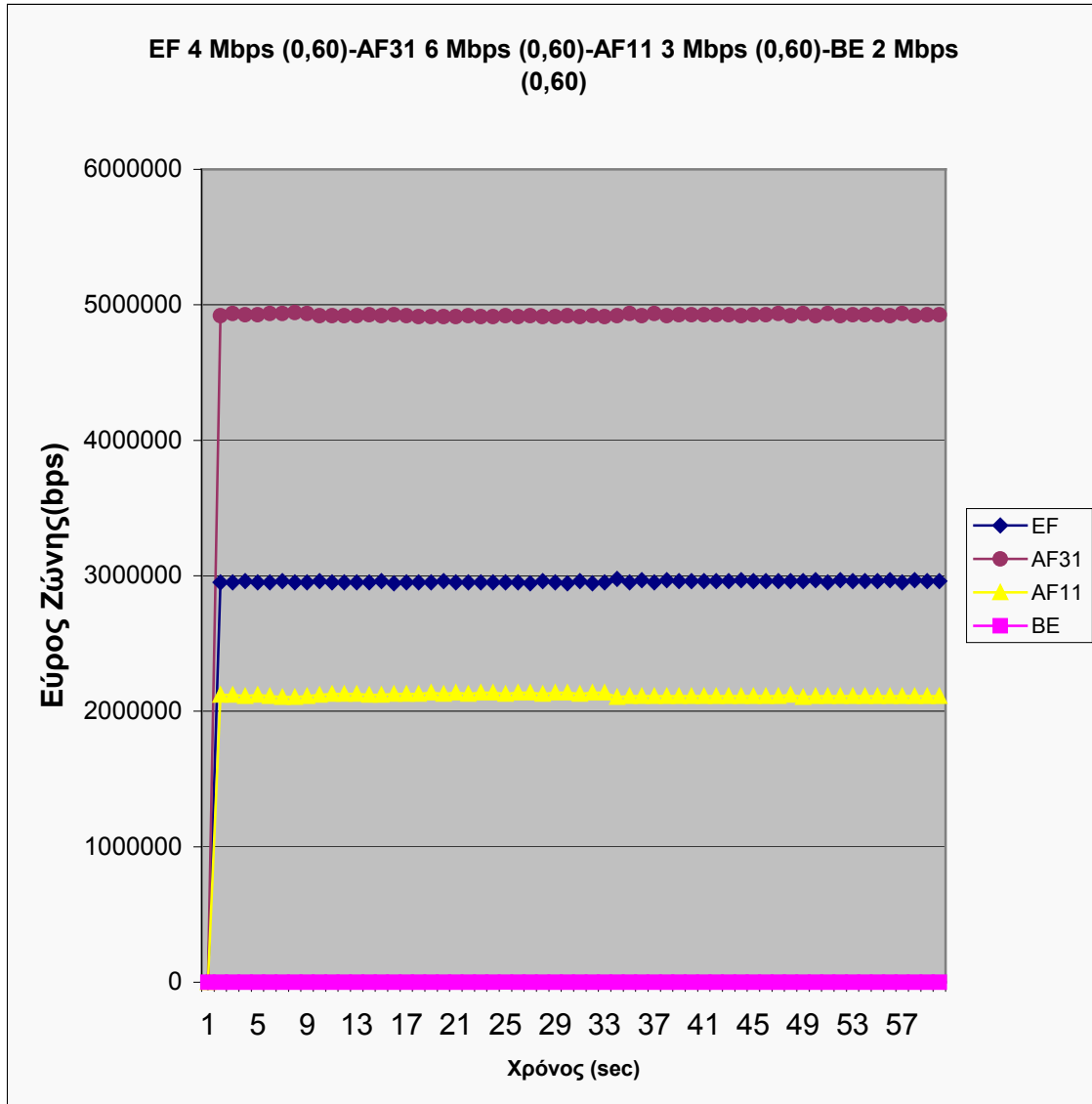


Διάγραμμα 4.3: Κατανομή εύρους ζώνης για κίνηση EF 3 Mbps (0 → 60) sec, AF11 5 Mbps (0 → 60) sec και BE 8 Mbps (0 → 60) sec

Η κίνηση EF λαμβάνει τα 3 Mbps που της αντιστοιχούν. Η κίνηση AF11 λαμβάνει 5 Mbps δανειζόμενη 3 Mbps από το περίσσιο εύρος ζώνης του δικτύου, ως κίνηση με υψηλότερη προτεραιότητα από την BE κίνηση, που δεν έχει καμία προτεραιότητα. Η BE κίνηση λαμβάνει τα 2 Mbps που περίσσεψαν από το συνολικό εύρος ζώνης του δικτύου. 2 Mbps αυτής της κίνησης απορρίπτονται.

Πείραμα 3

Κλάση υπηρεσίας	Παραγόμενη κίνηση(Mbps)	Χρόνος μετάδοσης(sec)
EF	3	0 → 60
AF31	6	0 → 60
AF11	3	0 → 60
BE	2	0 → 60

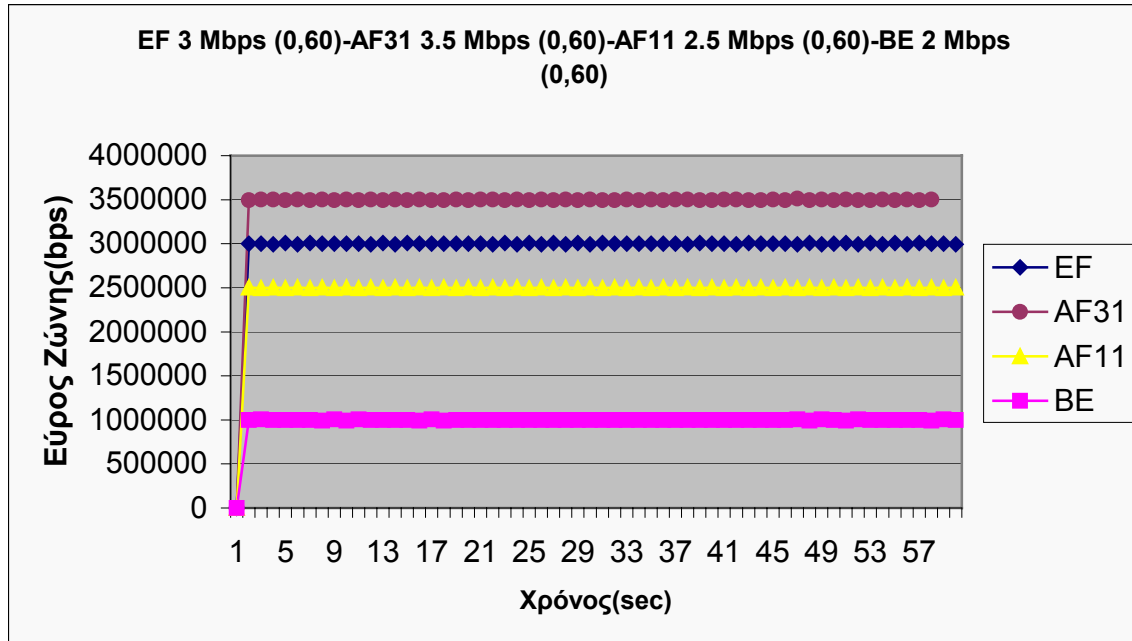


Διάγραμμα 4.4: Κατανομή εύρους ζώνης για κίνηση EF 4 Mbps (0 → 60) sec, AF31 6 Mbps (0 → 60) sec, AF11 3 Mbps (0 → 60) sec και BE 2 Mbps (0 → 60) sec

Η κίνηση EF λαμβάνει 3 Mbps, ενώ 1 Mbps αυτής της κίνησης απορρίπτεται. Η κίνηση AF31 λαμβάνει τα 5 Mbps που της αντιστοιχούν, ενώ 1 Mbps αυτής της κίνησης απορρίπτεται. Η κίνηση AF11 λαμβάνει 2 Mbps, ενώ 1 Mbps αυτής της κίνησης απορρίπτεται. Η BE κίνηση δεν λαμβάνει κανένα μέρος από το εύρος ζώνης του δικτύου. Το συνολικό εύρος ζώνης του δικτύου υπερκαλύφθηκε από τις απαιτήσεις των κλάσεων AF, έτσι δεν αλλά υπήρχαν περιθώρια δανεισμού. Η επιπλέον κίνηση αυτών των κλάσεων απορρίφθηκε, ενώ η BE κίνηση δεν κατέβαλε καθόλου πόρους.

Πείραμα 4

Κλάση υπηρεσίας	Παραγόμενη κίνηση(Mbps)	Χρόνος μετάδοσης(sec)
EF	3	0 → 60
AF31	3.5	0 → 60
AF11	2.5	0 → 60
BE	2	0 → 60

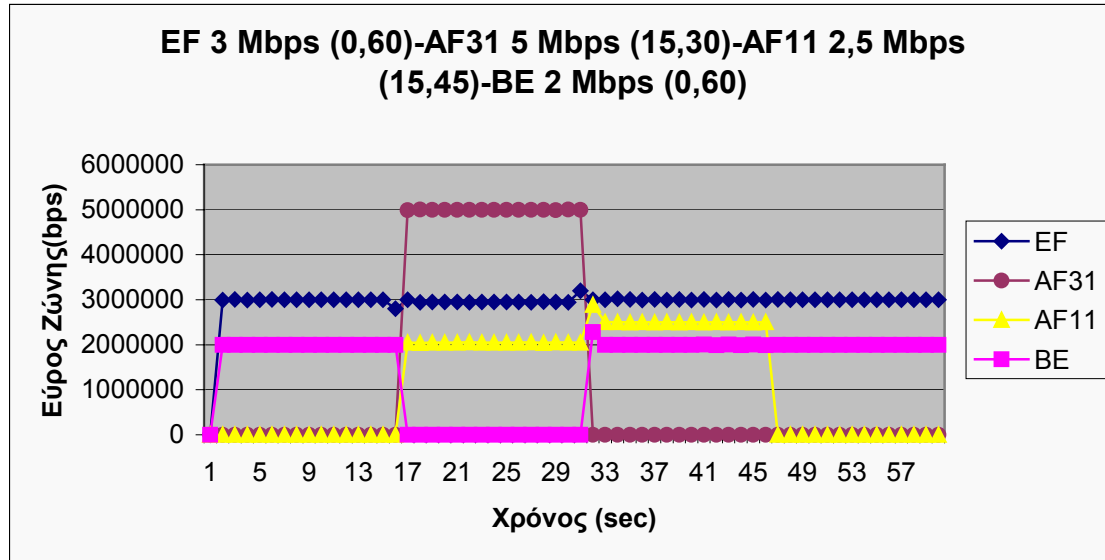


Διάγραμμα 4.5: Κατανομή εύρους ζώνης για κίνηση EF 3 Mbps (0 → 60) sec, AF31 3.5 Mbps (0 → 60) sec, AF11 2.5 Mbps (0 → 60) sec και BE 2 Mbps (0 → 60) sec

Η κίνηση EF λαμβάνει τα 3 Mbps που της αντιστοιχούν. Η κίνηση AF31 καταλαμβάνει 3.5 Mbps από τα 5 Mbps που της αντιστοιχούν. Η κίνηση AF11 καταλαμβάνει 2.5 Mbps δανειζόμενη 0.5 Mbps από το πλεονάζον 1.5 Mbps του δικτύου. Η BE κίνηση λαμβάνει το 1 Mbps, που περίσσεψε στο δίκτυο μετά την ικανοποίηση των απαιτήσεων των άλλων κλάσεων, ενώ 1 Mbps αυτής της κίνησης απορρίπτεται.

Πείραμα 5

Κλάση υπηρεσίας	Παραγόμενη κίνηση(Mbps)	Χρόνος μετάδοσης(sec)
EF	3	0 → 60
AF31	5	15 → 60
AF11	2.5	15 → 45
BE	2	0 → 60



Διάγραμμα 4.6: Κατανομή εύρους ζώνης για κίνηση EF 3 Mbps (0 → 60) sec, AF31 5 Mbps (15 → 30) sec, AF11 2.5 Mbps (15 → 45) sec και BE 2 Mbps (0 → 60) sec

Η κίνηση EF λαμβάνει τα 3 Mbps που της αντιστοιχούν, καθόλη την χρονική διάρκεια του πειράματος (0→60) sec. Η BE δεν λαμβάνει τα 2Mbps που απαιτεί μόνο κατά το χρονικό διάστημα (15→30), όπου τα 7 Mbps του εύρους ζώνης του δικτύου καταλαμβάνονται από τις κλάσεις AF. Η κίνηση AF31 εξυπηρετείται σύμφωνα με τις απαιτήσεις της για το χρονικό διάστημα (15→30). Η κίνηση AF11 λαμβάνει 2 Mbps το χρονικό διάστημα (15→30), ενώ παίρνει το επιπλέον 0.5 Mbps που αιτείται μόνο αφού σταματήσει η AF31 κίνηση και απελευθερωθούν οι πόροι του δικτύου. Αξίζει να παρατηρήσουμε ότι δημιουργούνται μικρές ακμές στο εύρος ζώνης που καταλαμβάνει η EF κίνηση, τις χρονικές στιγμές που η κίνηση AF31 αλλάζει απότομα (15 sec, 30). Οι ακμές αυτές κατά πάσα πιθανότητα, οφείλονται στην αδράνεια του δικτύου να προσαρμόσει στα νέα δεδομένα της κίνησης.

Με βάση τα πειραματικά αποτελέσματα, το σύστημα, με χρήση του χρονοπρογραμματιστή CBQ, παρέχει το απαιτούμενο εύρος ζώνης στις διαφορετικές ροές κίνησης παρουσιάζοντας αμελητέες αποκλίσεις από τις επιθυμητές τιμές.

4.4.2 Καθυστέρηση μετάδοσης

Εκτός από την κατανομή του εύρους ζώνης, ένα σύστημα παροχής ΠοιΥπ θα πρέπει να είναι σε θέση να παρέχει εγγυήσεις για την καθυστέρηση μετάδοσης των δεδομένων, που ανήκουν σε διαφορετικές κλάσεις υπηρεσίας.

Η μονόδρομη καθυστέρηση μετάδοσης. (one-way delay) ορίζεται ως ο χρόνος που μεσολαβεί από την αποστολή του πρώτου bit ενός πακέτου από την πηγή, έως την χρονική στιγμή που το τελευταίο bit αυτού του πακέτου λαμβάνεται από τον προορισμό [Almes99]. Έστω ότι T_{SRC} η χρονική στιγμή που το πρώτο bit ενός πακέτου μεταδόθηκε από την πηγή και T_{DST} η χρονική στιγμή που το τελευταίο bit του πακέτου αυτού φτάνει στον προορισμό, τότε η καθυστέρηση μετάδοσης dT δίνεται από την παρακάτω εξίσωση (4.2)

$$dT = T_{DST} - T_{SRC} \quad (4.2)$$

Ένα πακέτο, το οποίο διέρχεται από έναν δρομολογητή, μπορεί να υποστεί μια καθυστέρηση μετάδοσης για διάφορους λόγους. Η καθυστέρηση αυτή οφείλεται κυρίως στην χρόνο αναμονής του πακέτου, έως ότου αυτό επιλέγει από το χρονοπρογραμματιστή και στον πραγματικό χρόνο μετάδοσης από το φυσικό μέσο. Υπάρχουν και άλλοι λόγοι καθυστέρησης που οφείλονται στον χρόνο που χρειάζεται ο δρομολογητής για να επεξεργαστεί την επικεφαλίδα του πακέτου, ώστε να το δρομολογήσει κατάλληλα.

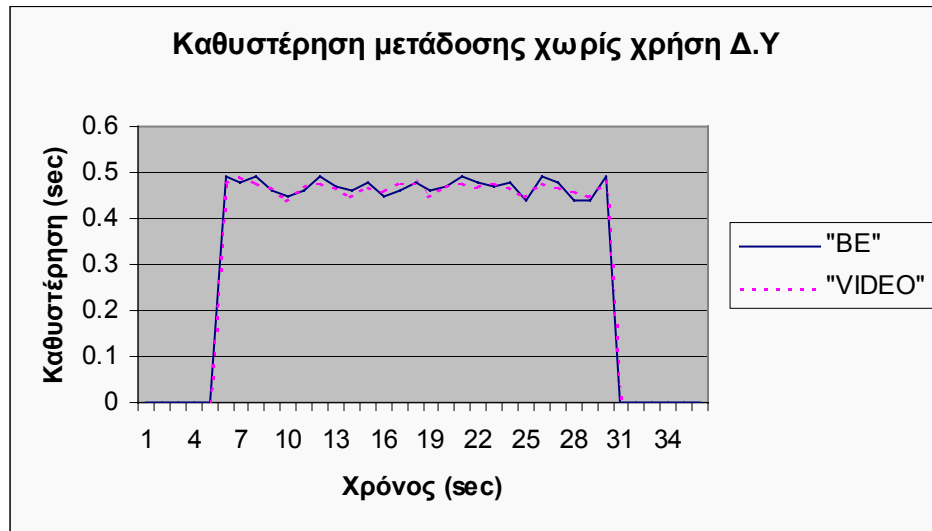
Η EF PHB σχεδιάστηκε ώστε να παρέχει μια υπηρεσία με μικρή καθυστέρηση μετάδοσης σε εφαρμογές με αυστηρές απαιτήσεις στη καθυστέρηση μετάδοσης των δεδομένων (π.χ video). Η πειραματική διαδικασία έλεγχου της καθυστέρησης μετάδοσης, που παρέχεται από το σύστημα, περιλαμβάνει την αποστολή ενός σήματος video χωρίς την χρήση ΔΥ και με την χρήση ΔΥ. Στο πρώτο πείραμα το σήμα video μεταδίδεται χωρίς τα πακέτα να μαρκάρονται σε κάποια κλάση υπηρεσίας, Ενώ στο δεύτερο πείραμα, τα πακέτα μαρκάρονται ώστε να ανήκουν στην EF PHB. Και στις δυο περιπτώσεις αποστέλλεται συγχρόνως BE κίνηση, ώστε να είναι δυνατή η σύγκριση της καθυστέρησης μετάδοσης για τις δυο διαφορετικές ροές κίνησης.

Το Πειραματικό IP δίκτυο ΔΥ είναι το ίδιο με αυτό των πειραμάτων κατανομής του εύρους ζώνης και παρουσιάζεται στο σχήμα 4.10. Η κίνηση μεταδίδεται από τον χρήστη Α προς τον χρήστη Β. Τα χαρακτηριστικά της κίνησης καθώς και οι παράμετροι του χρονοπρογραμματιστή CBQ παρουσιάζονται στον παρακάτω πίνακα.

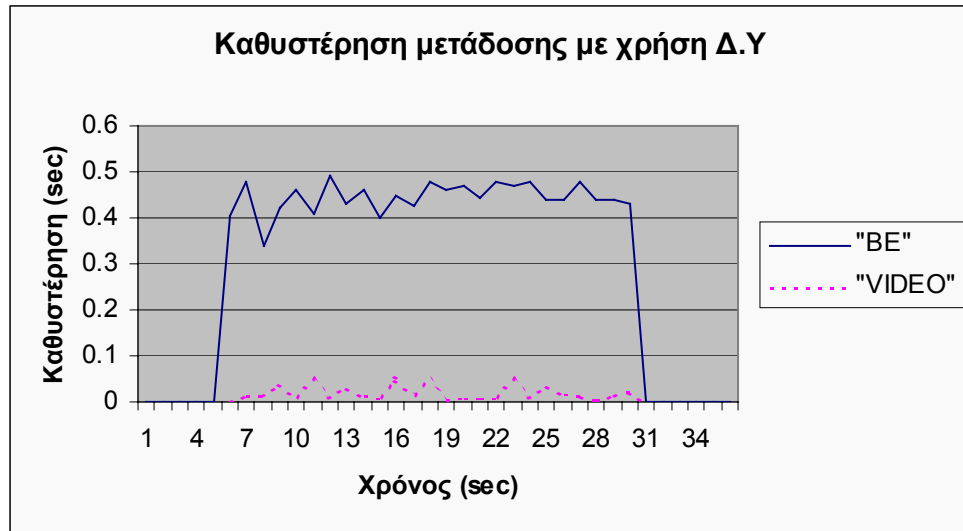
Πίνακας 4.5: Χαρακτηριστικά της κίνησης και οι παράμετροι του χρονοπρογραμματιστή CBQ

Ροή	Χαρακτηριστικά ροής	DSCP	CBQ παράμετροι
1	Video: 384kbit/s; Avg Packet size=800Bytes	EF	Queue size=3.2 kB; Rate=390 kbit/s
2	MGEN: rate=1Mbit/s; packet size=1kB	BE	Queue size=60kB; Rate=500 kbit/s

Για την μέτρηση της καθυστέρησης μετάδοσης χρησιμοποιήθηκε το υποσύστημα παρακολούθησης, με περίοδο παρακολούθησης 1 sec. Η καθυστέρηση μετάδοσης, με χρήση και χωρίς χρήση Δ.Υ, παρουσιάζεται στα διαγράμματα 4.7 και 4.8 αντίστοιχα.



Διάγραμμα 4.7: Καθυστέρηση μετάδοσης χωρίς χρήση ΔΥ



Διάγραμμα 4.8: Καθυστέρηση μετάδοσης με χρήση ΔΥ

Χωρίς χρήση Δ.Υ (διάγραμμα 4.7), η καθυστέρηση μετάδοσης όλης την κίνησης (BE και video) είναι της τάξης του 0,5 sec. Η καθυστέρηση αυτή οφείλεται στην αναμονή των πακέτων στην ουρά μέχρι την μετάδοση του, καθώς και στην καθυστέρηση μετάδοσης από το φυσικό μέσο. Η ελάττωση της καθυστέρησης αυτής στην κίνηση video είναι εμφανέστατη, όταν αυτή αντιμετωπίζεται ως EF κίνηση. Στην περίπτωση αυτή, η καθυστέρηση μετάδοσης οφείλεται εξ ολόκληρου στον πραγματικό χρόνο μετάδοσης από το φυσικό μέσο, εφόσον δεν υπάρχει χρόνος αναμονής στην ουρά (εξυπηρετούνται αμέσως). Η καθυστέρηση αυτής είναι της τάξης του 50 msec. Στους παραπάνω χρόνους συμπεριλαμβάνεται και η καθυστέρηση επεξεργασίας των πακέτων από τον δρομολογητή, για την κατάλληλη δρομολόγηση τους

4.5 Σύνοψη κεφαλαίου - συμπεράσματα

Στο κεφαλαίο αυτό παρουσιάστηκε και αξιολογήθηκε ένα κατακεντρωμένο σύστημα διαχείρισης δικτύου. Το σύστημα είναι ικανό να προσφέρει υπηρεσίες διάρθρωσης και παρακολούθησης ενός IP δικτύου ΔΥ. Για την παροχή των υπηρεσιών αυτών δεν ακολουθείται ο συνηθισμένος τρόπος διαχείρισης, στον οποίο μια κεντρική μονάδα έλεγχου επικοινωνεί μέσω ενός πρωτοκόλλου (συνήθως το SNMP) με αντιπρόσωπους διαχείρισης, που βρίσκονται στους κόμβους του δικτύου.

Η υλοποίηση του συστήματος είναι στηριγμένη στην ΤΚΠ. Η χρήση της τεχνολογία αυτής στην υλοποίηση συστημάτων διαχείρισης δικτύων προσδίδει μια σειρά από πλεονέκτημα, έναντι παρομοίων συστημάτων που ακολουθούν το κεντροποιημένο μοντέλο υλοποίησης. Τα σημαντικότερα από αυτά είναι η μείωση της κίνησης που παράγεται στο δίκτυο και η εύκολη αναβάθμιση του συστήματος, με προσθήκη νέων υπηρεσιών. Όλες οι λειτουργίες διάρθρωσης και παρακολούθησης πραγματοποιούνται με χρήση κινητών και στατικών πρακτόρων. Οι ΚΠ μεταφέρουν τις εντολές διάρθρωσης και παρακολούθησης στους κόμβους του δικτύου. Επίσης, μεταφέρουν τα αποτελέσματα της εκτέλεσης των εντολών στην κεντρική μονάδα του συστήματος, μέσω της οποίας παρουσιάζονται στον χρήστη. Οι στατικοί πράκτορες που βρίσκονται στους κόμβους του δικτύου (δρομολογητές), δρουν ως διεπαφές μεταξύ των ΚΠ και του λογισμικού του κόμβου. Δηλαδή, όλες οι αλληλεπιδράσεις μεταξύ των ΚΠ και των κόμβων πραγματοποιούνται διάμεσου των στατικών πρακτόρων. Εναλλακτικά, οι ΚΠ θα μπορούσαν να μεταφέρουν την απαραίτητη πληροφορία για την αλληλεπίδραση με το λογισμικό των κόμβων. Αυτή η προσέγγιση θα οδηγούσε σε κινητούς πράκτορες με μεγάλο μέγεθος, που θα είχε αρνητική επίδραση στον χρόνο μετανάστευσης τους.

Το σύστημα είναι σε θέση να διαχειριστεί δρομολογητές τύπου Linux και τύπου Cisco. Δυστυχώς δεν υπάρχει μια κοινή MIB ΔΥ, που να υποστηρίζεται και από του δυο τύπους. Η ύπαρξη μια τέτοιας MIB θα επέτρεπε την διαχείριση των δυο τύπων δρομολογητών, με ένα ενιαίο τρόπο. Το πρόβλημα αυτό αντιμετωπίστηκε με την δημιουργία ενός υποσυστήματος προσαρμογής, το οποίο αποκρύπτει τις λεπτομέρειες διαχείρισης των δρομολογητών από τους ΚΠ του συστήματος. Η λειτουργία του υποσυστήματος αυτού στηρίζεται στην ύπαρξη μιας σειράς στατικών πρακτόρων, που

διαθέτουν όλη την απαραίτητη πληροφορία για την παρακολούθηση και διάρθρωση των δρομολογητών.

Ένα πρόβλημα που αντιμετωπίζουν όλες οι εφαρμογές κινητών πρακτόρων, είναι η εξάρτηση της λειτουργία τους από μια και μόνο ΠΚΠ. Το πρόβλημα αυτό ξεπεράστηκε με την δημιουργία μια κοινής ΔΠΕ ΠΚΠ, για τις δυο ΠΚΠ που συνήθως χρησιμοποιούνται (Grasshopper και Voyager). Η χρήση της κοινής ΔΠΕ επιτρέπει την εκτέλεση των πρακτόρων του συστήματος και στις δυο πλατφόρμες, με ελάχιστη μεταβολή στον κώδικα τους.

Για την διάρθρωση των δρομολογητών υλοποιήθηκε η ΔΠΕ Jain/Parlay. Σύμφωνα με την λογική της ΔΠΕ Jain/Parlay, ένα χρήστης είναι σε θέση να δημιουργήσει εικονικούς ιδιωτικούς δίαυλους (Virtual Private Pipes, VPrP) με άλλους χρηστές του συστήματος. Οι δίαυλοι αυτοί είναι συνδέσεις που παρέχουν μια συγκεκριμένη ΠοιΥπ, στην επικοινωνία των χρηστών. Επίσης, ο χρήστης είναι σε θέση να παρακολουθήσει την ΠοιΥπ που του προσφέρει το σύστημα, ώστε να μπορεί να την συγκρίνει με αυτήν που απαιτήσε. Η παρακολούθηση της παρεχόμενης ΠοιΥπ έχει ιδιαίτερη σημασία στα συστήματα παροχής ΠοιΥπ, διότι ο χρήστης πρέπει να είναι σε θέση να γνωρίζει αν η ΠοιΥπ που του παρέχεται συμφωνεί με το συμβόλαιο υπηρεσιών (SLA), για το οποίο έχει πληρώσει.

Τα πειραματικά αποτελέσματα αξιολόγησης της εφαρμογής απέδειξαν ότι το σύστημα κατανέμει το εύρος ζώνης του δικτύου, στις διαφορετικές ροές κίνησης, παρουσιάζοντας αμελητέες αποκλίσεις από τις επιθυμητές τιμές. Η συλλογή των αποτελεσμάτων έγινε με χρήση του υποσυστήματος παρακολούθησης. Η συμφωνία των τιμών αυτών με την διάρθρωση του δικτύου αποδεικνύει ότι και το σύστημα παρακολούθησης δεν παρουσιάζει σφάλματα στον τρόπο λειτουργίας. Ένα σημείο το οποίο αξίζει να αναφέρουμε είναι ότι δημιουργούνται μικρές ακμές στο εύρος ζώνης που καταλαμβάνει μια κίνηση υψηλής προτεραιότητας όταν αυξομειώνεται απότομα το εύρος ζώνης μιας κίνησης χαμηλής προτεραιότητας. Η μόνο λογική εξήγηση για το γεγονός αυτό, είναι η προς στιγμήν αδράνεια του δικτύου να προσαρμοστεί με τα νέα δεδομένα της κίνησης.

Τα πειραματικά αποτελέσματα ήταν ικανοποιητικά και στην περίπτωση μέτρησης της καθυστέρησης μετάδοσης. Το προτεινόμενο σύστημα διαχείρισης, είναι σε θέση να

προσφέρει εγγυήσεις για την καθυστέρηση μετάδοσης, δεδομένων που ανήκουν σε διαφορετικές κλάσεις υπηρεσίας. Με κατάλληλη διάρθρωση των δρομολογητών του δικτύου, η παραγόμενη BE κίνηση παρουσίασε μια καθυστέρηση μετάδοσης της τάξης των 0,5 sec, ενώ EF κίνηση που μεταδόθηκε ταυτόχρονα παρουσίασε καθυστέρηση της τάξης των 50 msec. Και σε αυτήν την περίπτωση το σύστημα παρακολούθησης έδωσε αποτελέσματα σύμφωνα με τις αναμενόμενες τιμές, αποδεικνύοντας την σωστή λειτουργία του.

4.6 Βιβλιογραφικές αναφορές

[Case90] J. Case, M. Fedora, M. Schoffstall, J. Davin, “A Simple Network Management Protocol (SNMP)”, *IETF RFC 1157*, May 1990.

[CMIP] ISO/IEC 9596, Information Technology, Open Systems Interconnection, "Common Management Information Protocol (CMIP)", Part 1: Specification, Geneva, Switzerland, 1991.

[Goldszmidt93] G. Goldszmidt, “On Distributed System Management,” *Distributed Computing and Communications Lab*, Computer Science Department, Columbia University, 1993, Διαθέσιμο στη ηλεκτρονική διεύθυνση: <http://netman.cit.buffalo.edu/Papers.html>.

[Martin-Flatin99] J.P. Martin-Flatin and S. Znaty, “Two Taxonomies of Distributed Network Management Paradigms”, *Network and Systems Management: Emerging Trends and Future Challenges*, Plenum Press, New York, USA, 1999.

[Bieszczad98] A. Bieszczad, B. Pagurek, and T. White, “Mobile Agents for Network Management”, *IEEE Commun. Survey*, Vol. 1, No. 1, 1998.

[D22] MANTRIP – IST-10921 project deliverable D2.2, “Functional Specifications and Validation Scenarios”, January 2001.

[D42] MANTRIP – IST-10921 project deliverable D2.2, “management tools detailed design”, April 2002.

[Parlay] The Parlay Group, Η πληροφορία είναι διαθέσιμη στην ηλεκτρονική διεύθυνση: <http://www.parlay.org/>

[JAIN] The JAIN Initiative, Η πληροφορία είναι διαθέσιμη στην ηλεκτρονική διεύθυνση: <http://java.sun.com/products/jain>

[Parlay2] Parlay API's 2.1, Connectivity Manager APIs – Version 1.1, Διαθέσιμο στην ηλεκτρονική διεύθυνση: <http://www.parlay.org/specs/>

[MySQL] Διαθέσιμο στην ηλεκτρονική διεύθυνση: <http://www.mysql.com/>

[Cisco] Η πληροφορία είναι διαθέσιμη στην ηλεκτρονική διεύθυνση:
<http://www.cisco.com>

[Almesberger99] W. Almesberger, J.H. Salim and A. Kuznetsov, “Differentiated Services on Linux”, *IETF-draft*, March 12, 1999

[Kotsilieris02] T. Kotsilieris*, P. Zikos, E. Vlachos, S. Kalogeropoulos, A. Michalas, G. Karetos and V. Loumos, “A Prototype SNMP Management Framework for DiffServ Linux Routers, its Implementation and Performance”, *Proc Net-Con '02*, October 2002, pp. 405-416

[QoSCLI] Διαθέσιμο στην ηλεκτρονική διεύθυνση:
<http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/120newft/120limit/120xe/120xe5/mqc/mcli.htm>

[ASN97] Telecommunication Standardization Sector of ITU, “Information Technology – Abstract Syntax Notation One (ASN.1): Specification of basic notation,” December 1997

[Floyd93] S. Floyd and V. Jacobson, “Random Early Detection Gateways For Congestion Avoidance”, *IEEE/ACM Transactions on Networking*, Vol.1, No 4, August 1993, pp. 397-413.

[Bernet02] Y. Bernet, S. Blake D. Grossman and A. Smith, “An Informal Management Model for Diffserv Routers”, IETF RFC 3290, May 2002

[Almesberger99] W. Almesberger, J.H. Salim and A. Kuznetsov, “Differentiated Services on Linux”, *IETF-draft*, March 12, 1999

[Dhandapani99] G. Dhandapani and A. Sundaresan, “Netlink Sockets – Overview”, Information and Telecommunications Technology Center, Department of Electrical Engineering & Computer Science, The University of Kansas, διαθέσιμο στην ηλεκτρονική διεύθυνση <http://qos.ittc.ukans.edu/netlink/html/index.html>.

[FIPA] Η πληροφορία είναι διαθέσιμη στην ηλεκτρονική διεύθυνση:
<http://www.fipa.org/>.

[MGEN] The Multi-Generator (MGEN) Toolset, Naval Research Laboratory (NRL), Διαθέσιμο στην ηλεκτρονική διεύθυνση: <http://manimac.itd.nrl.navy.mil/MGEN>.

[Almes99] G. Almes S, Kalidindi and M. Zekauskas, “A One-way Delay Metric for IPPM”, *IETF RFC2679*, September 1999.

ΚΕΦΑΛΑΙΟ 5

ΣΥΓΚΡΙΣΗ ΤΗΣ ΤΕΧΝΟΛΟΓΙΑ ΚΙΝΗΤΩΝ ΠΡΑΚΤΟΡΩΝ ΜΕ ΤΟ ΜΟΝΤΕΛΟ ΠΕΛΑΤΗ/ΕΞΥΠΗΡΕΤΗΤΗ

5.1 Εισαγωγή

Στο προηγούμενο κεφάλαιο παρουσιάστηκε ένα ολοκληρωμένο σύστημα διαχείρισης ενός IP δικτύου ΔΥ, υλοποιημένο με χρήση της ΤΚΠ. Στο κεφάλαιο αυτό θα παρουσιαστεί μια σύγκριση της απόδοσης του συστήματος αυτού με μια εναλλακτική υλοποίηση του, με βάση το μοντέλο πελάτη/εξυπηρετητή [Kalogeropoulos03]. Συγκεκριμένα θα συγκριθεί ο χρόνος που χρειάζεται κάθε μια από τις δυο προσεγγίσεις υλοποίησης, για να πραγματοποιήσει την αρχική διάρθρωση του δικτύου (δρομολογητές κορμού). Στις παραγράφους 4.2 και 4.3 παρουσιάστηκαν τα προβλήματα που παρουσιάζει το μοντέλο πελάτη/εξυπηρετητή και οι βελτιώσεις που μπορεί να επιφέρει η ΤΚΠ, σε ένα σύστημα διαχείρισης δικτύου. Αναμφίβολα, οι ΚΠ παρουσιάζουν μια σειρά πλεονεκτημάτων έναντι του μοντέλου πελάτη/εξυπηρετητή, όσον αφορά την χρήση του δικτύου, την ευκολία αναβάθμισης του συστήματος και την κλιμάκωση του σε μεγαλύτερα δίκτυα. Όμως, παρόλο που αρκετοί ερευνητές μελέτησαν την εφαρμογή της ΤΚΠ σε διάφορα συστήματα [Carzaniga97, Straier97], λίγοι ασχολήθηκαν με την επίδραση που έχει η τεχνολογία αυτή στη απόδοση του συστήματος [Ismail99, Kotsilieris02].

Στις επόμενες παραγράφους θα αποδείξουμε ότι η χρήση της ΤΚΠ δεν επιφέρει πάντα βελτιώσεις στην απόδοση ενός συστήματος. Στην πραγματικότητα αυτό επιτυγχάνεται μόνο υπό ορισμένες προϋποθέσεις, που σχετίζονται με την πολυπλοκότητα του προς εκτέλεση έργου. Για να είναι δυνατή η σύγκριση των δυο προσεγγίσεων, με βάση την πολυπλοκότητα του προς εκτέλεση έργου, οι δυο εφαρμογές υλοποιήθηκαν με τέτοιο τρόπο ώστε να επιτρέπουν την εκτέλεση διεργασιών με διαφορετική και αυξανόμενη πολυπλοκότητα. Αυτό έχει ως αποτέλεσμα κάποιες μικρές επεμβάσεις στην λειτουργία του συστήματος, που παρουσιάστηκε στο κεφάλαιο 4. Οι αλλαγές αυτές επισημαίνονται

στην συνεχεία και διαχωρίζονται από την πραγματική λειτουργία του αρχικού συστήματος διαχείρισης.

Η σύγκριση των δυο εναλλακτικών προσεγγίσεων διαχείρισης δικτύου που παρουσιάζεται, βασίζεται σε πραγματικές μετρήσεις. Για τον σκοπό αυτό, εγκαταστάθηκε ένα πειραματικό IP δίκτυο ΔΥ, αποτελούμενο από δρομολογητές τύπου Linux. Ο τρόπος διάρθρωσης των δρομολογητών αυτού του τύπου, για την παροχή ΔΥ, έχει ήδη παρουσιαστεί στην παράγραφο 4.4.3.1.2. Στα πλαίσια του κεφαλαίου αυτού, θα δοθούν περισσότερες πληροφορίες για την αρχική διάρθρωση των δρομολογητών κορμού ενός δικτύου ΔΥ.

5.2 Διάρθρωση δρομολογητών κορμού ενός δικτύου ΔΥ αποτελούμενο από δρομολογητές Linux.

Οι δρομολογητές κορμού ενός δικτύου ΔΥ θα πρέπει να είναι διαρθρωμένοι με τέτοιο τρόπο, ώστε να εξυπηρετούν την εισερχόμενη κίνηση ανάλογα με το DSCP πεδίο των πακέτων. Για αυτό τον λόγο, είναι απαραίτητη η προπαρασκευή των δρομολογητών αυτών, ώστε να εκτελούν αυτήν την λειτουργία (κατηγοριοποίηση της κίνησης και ανάλογη εξυπηρέτηση). Στους δρομολογητές Linux η διάρθρωση αυτή επιτυγχάνεται με την χρήση ενός εργαλείου, το οποίο ονομάζεται traffic control tool [Stanic01] (tc tool). Η λειτουργία του tc έχει ήδη περιγράφει στην παράγραφο 4.4.3.1.2. Επαναληπτικά αναφέρουμε, ότι η διάρθρωση με χρήση του εργαλείου tc πραγματοποιείται με την εκτέλεση μια σειράς εντολών στο περιβάλλον του χρήστη.

Η σειρά εντολών που εκτελούνται για την ανάγνωση του DSCP πεδίου ενός πακέτου και την ένταξη του σε μια από τις 14 PHBs των ΔΥ, παρουσιάζεται παρακάτω.

```
$TC filter add dev $DEV parent 1:0 protocol ip prio 1 u32 divisor 1
$TC filter add dev $DEV parent 1:0 prio 1 u32 match ip tos 0x28 0xfc classid 1:11
$TC filter add dev $DEV parent 1:0 prio 1 u32 match ip tos 0x30 0xfc classid 1:12
$TC filter add dev $DEV parent 1:0 prio 1 u32 match ip tos 0x38 0xfc classid 1:13
$TC filter add dev $DEV parent 1:0 prio 1 u32 match ip tos 0x48 0xfc classid 1:21
$TC filter add dev $DEV parent 1:0 prio 1 u32 match ip tos 0x50 0xfc classid 1:22
$TC filter add dev $DEV parent 1:0 prio 1 u32 match ip tos 0x58 0xfc classid 1:23
$TC filter add dev $DEV parent 1:0 prio 1 u32 match ip tos 0x68 0xfc classid 1:31
$TC filter add dev $DEV parent 1:0 prio 1 u32 match ip tos 0x70 0xfc classid 1:32
$TC filter add dev $DEV parent 1:0 prio 1 u32 match ip tos 0x78 0xfc classid 1:33
$TC filter add dev $DEV parent 1:0 prio 1 u32 match ip tos 0x88 0xfc classid 1:41
$TC filter add dev $DEV parent 1:0 prio 1 u32 match ip tos 0x90 0xfc classid 1:42
$TC filter add dev $DEV parent 1:0 prio 1 u32 match ip tos 0x98 0xfc classid 1:43
$TC filter add dev $DEV parent 1:0 prio 1 u32 match ip tos 0xb8 0xfc classid 1:51
```

Στον πίνακα 5.1 παρουσιάζεται η τιμή του DSCP πεδίου, των πακέτων, που εισέρχονται σε έναν δρομολογητή κορμού ΔΥ και η αντιστοίχηση τους σε μια PHB. Οι τιμές των πεδίων είναι σύμφωνες με τον ορισμό του DSCP πεδίου [Nichols98].

Η εξαγωγή του DSCP πεδίου από το TOS byte γίνεται ως εξής:

$$DSCP = (TOS \& \text{mask}) \gg \text{shift} \quad (5.1)$$

Όπου mask= 0xfc και Shift (μετατόπιση) κατά 2 προς τα δεξιά.

Για παράδειγμα εάν ένα πακέτο φτάσει στον δρομολογητή με τιμή TOS= 0xb8, τότε η τιμή του DSCP πεδίου είναι:

```
TOS      0xb8    10111000
mask     0xfc    10111000
shift    2       00101110 (0x2e, DSCP για την PHB EF)
```

Σύμφωνα με τον πίνακα 5.1 το αναγνωριστικό κλάσης που αντιστοιχεί σε αυτήν την PHB είναι το 1:51.

Πίνακας 5.1: Τιμή του DSCP πεδίου και αντιστοίχιση σε PHB

TOS (Δεκαδικός)	Δεκα- εξαδικός	Δυαδικός	DSCP (Δεκαδικός)	Δεκα- εξαδικός	Δυαδικός	Κλάση	PHB
40	0x28	00101000	10	0x0a	00001010	1:11	AF 11
48	0x30	00110000	12	0x0c	00001100	1:12	AF 12
56	0x38	00111000	14	0x0e	00001110	1:13	AF 13
72	0x48	01001000	18	0x12	00010010	1:21	AF 21
80	0x50	01010000	20	0x14	00010100	1:22	AF 22
88	0x58	01011000	22	0x16	00010110	1:23	AF 23
104	0x68	01101000	26	0x1a	00011010	1:31	AF 31
112	0x70	01110000	28	0x1c	00011100	1:32	AF 32
120	0x78	01111000	30	0x1e	00011110	1:33	AF 33
136	0x88	10001000	34	0x22	00100010	1:41	AF 41
144	0x90	10010000	36	0x24	00100100	1:42	AF 42
152	0x98	10011000	38	0x26	00100110	1:43	AF 43
184	0xb8	10111000	46	0x2e	00101110	1:51	EF
0	0	00000000	0	0	00000000	1:61	BE

Το επόμενο βήμα, μετά την κατηγοριοποίηση των πακέτων, είναι να οριστεί ο τρόπος με τον οποίο εξυπηρετούνται τα πακέτα που ανήκουν σε διαφορετικές ροές. Οι παρακάτω εντολές tc καθορίζουν τον τρόπο με τον οποίο εξυπηρετούν τα πακέτα οι διάφορες PHBs, που υλοποιούνται από τον δρομολογητή κορμού.

```
#CBQ qdisc
$TC qdisc add dev $DEV parent 1:0 handle 2:0 cbq bandwidth 10Mbit cell 8 avpkt 1000 mpu
64

#AF1
$TC class add dev $DEV parent 2:0 classid 2:1 cbq bandwidth 10Mbit \
rate 1Mbit avpkt 1000 prio 5 allot 1514 weight 1.5Mbit maxburst 21 defmap 0 borrow
$TC filter add dev $DEV parent 2:0 protocol ip prio 1 handle 1 tcindex classid 2:1
```

```

$TC qdisc add dev $DEV parent 2:1 gred setup DPs 3 default 2 prio
$TC qdisc change dev $DEV parent 2:1 gred limit 600K min 150K max 450K burst 200 \
avpkt 1000 bandwidth 10Mbit DP 1 probability 0.02 prio 2

$TC qdisc change dev $DEV parent 2:1 gred limit 600K min 150K max 450K burst 200 \
avpkt 1000 bandwidth 10Mbit DP 2 probability 0.04 prio 3
$TC qdisc change dev $DEV parent 2:1 gred limit 600K min 150K max 450K burst 200 \
avpkt 1000 bandwidth 10Mbit DP 3 probability 0.06 prio 4

#AF2
$TC class add dev $DEV parent 2:0 classid 2:2 cbq bandwidth 10Mbit \
rate 3Mbit avpkt 1000 prio 4 allot 1514 weight 1.5Mbit maxburst 21 defmap 0 borrow
$TC filter add dev $DEV parent 2:0 protocol ip prio 1 handle 2 tcindex classid 2:2
$TC qdisc add dev $DEV parent 2:2 gred setup DPs 3 default 2 prio
$TC qdisc change dev $DEV parent 2:2 gred limit 600K min 150K max 450K burst 200 \
avpkt 1000 bandwidth 10Mbit DP 1 probability 0.02 prio 2
$TC qdisc change dev $DEV parent 2:2 gred limit 600K min 150K max 450K burst 200 \
avpkt 1000 bandwidth 10Mbit DP 2 probability 0.04 prio 3
$TC qdisc change dev $DEV parent 2:2 gred limit 600K min 150K max 450K burst 200 \
avpkt 1000 bandwidth 10Mbit DP 3 probability 0.06 prio 4

#AF3
$TC class add dev $DEV parent 2:0 classid 2:3 cbq bandwidth 10Mbit \
rate 1Mbit avpkt 1000 prio 3 allot 1514 weight 1.5Mbit maxburst 21 defmap 0 borrow
$TC filter add dev $DEV parent 2:0 protocol ip prio 1 handle 3 tcindex classid 2:3
$TC qdisc add dev $DEV parent 2:3 gred setup DPs 3 default 2 prio
$TC qdisc change dev $DEV parent 2:3 gred limit 600K min 150K max 450K burst 200 \
avpkt 1000 bandwidth 10Mbit DP 1 probability 0.02 prio 2
$TC qdisc change dev $DEV parent 2:3 gred limit 600K min 150K max 450K burst 200 \
avpkt 1000 bandwidth 10Mbit DP 2 probability 0.04 prio 3
$TC qdisc change dev $DEV parent 2:3 gred limit 600K min 150K max 450K burst 200 \
avpkt 1000 bandwidth 10Mbit DP 3 probability 0.06 prio 4

#AF4
$TC class add dev $DEV parent 2:0 classid 2:4 cbq bandwidth 10Mbit \
rate 3Mbit avpkt 1000 prio 2 allot 1514 weight 1.5Mbit maxburst 21 defmap 0 borrow
$TC filter add dev $DEV parent 2:0 protocol ip prio 1 handle 4 tcindex classid 2:4
$TC qdisc add dev $DEV parent 2:4 gred setup DPs 3 default 2 prio
$TC qdisc change dev $DEV parent 2:4 gred limit 600K min 150K max 450K burst 200 \
avpkt 1000 bandwidth 10Mbit DP 1 probability 0.02 prio 2
$TC qdisc change dev $DEV parent 2:4 gred limit 600K min 150K max 450K burst 200 \
avpkt 1000 bandwidth 10Mbit DP 2 probability 0.04 prio 3
$TC qdisc change dev $DEV parent 2:4 gred limit 600K min 150K max 450K burst 200 \
avpkt 1000 bandwidth 10Mbit DP 3 probability 0.06 prio 4

#EF
$TC class add dev $DEV parent 2:0 classid 2:5 cbq bandwidth 10Mbit \
rate 1Mbit avpkt 1000 prio 1 allot 1514 weight 1.5Mbit maxburst 21 defmap 0
$TC filter add dev $DEV parent 2:0 protocol ip prio 1 handle 5 tcindex classid 2:5

```



```
$TC qdisc add dev $DEV parent 2:5 pfifo limit 5
```

```
#BE
```

```
$TC class add dev $DEV parent 2:0 classid 2:6 cbq bandwidth 10Mbit \
```

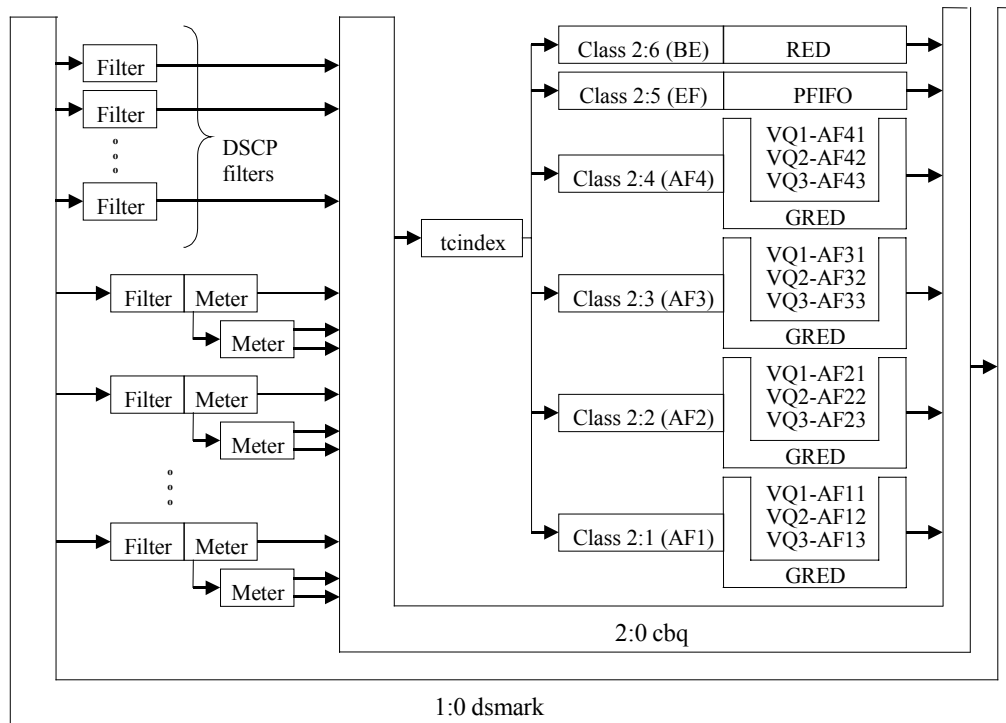
```
rate 1Mbit avpkt 1000 prio 7 allot 1514 weight 1.5Mbit maxburst 21 defmap 0
```

```
$TC filter add dev $DEV parent 2:0 protocol ip prio 1 handle 6 tcindex classid 2:6
```

```
$TC qdisc add dev $DEV parent 2:6 red limit 600000 min 150000 max 450000 burst 200 \
```

```
avpkt 1000 bandwidth 10Mbit probability 0.4
```

Με την εκτέλεση των παραπάνω εντολών όλοι οι δρομολογητές κορμού έχουν την διάρθρωση που παρουσιάζεται στο σχήμα 5.1.



Σχήμα 5.1: Σχήμα αναμονής του δρομολογητή κορμού

Η διάρθρωση του δρομολογητή για την εξυπηρέτηση της κίνησης από διαφορετικές ροές, επιτυγχάνεται με την προσθήκη (εντολή `tc add`) μιας σειράς από φίλτρα, κλάσεις και σχημάτων αναμονής, στο τμήμα εξόδου του. Η διάρθρωση αυτή πραγματοποιείται σταδιακά. Αρχικά διαρθρώνεται ένα εξωτερικό σχήμα αναμονής (`dsmark`—πρώτη γραμμή εντολών) και στην συνέχεια ορίζονται οι διάφορες PHBs.

5.3 Τρόπος λειτουργίας των δυο υλοποιήσεων με διαφορετική πολυπλοκότητα του προς εκτέλεση έργου

Οι δυο εναλλακτικές προσεγγίσεις διάρθρωσης των δρομολογητών θα πρέπει να είναι σε θέση να εκτελέσουν τις παραπάνω εντολές tc, στον δρομολογητή. Σύμφωνα με όσα αναφέρθηκαν στην προηγούμενη παράγραφο η διάρθρωση πραγματοποιείται σταδιακά. Η σταδιακή διάρθρωση επιτρέπει τον ορισμό τεσσάρων διαφορετικών τρόπων λειτουργίας, με αυξανόμενη πολυπλοκότητα, του προς εκτέλεση έργου.

Στον τον πρώτο τρόπο λειτουργίας διαρθρώνεται μόνο το εξωτερικό σχήμα αναμονής. Η tc εντολή που εκτελείται είναι η παρακάτω:

```
#CBQ Qdisc
$TC qdisc add dev $DEV parent 1:0 handle 2:0 cbq bandwidth 10Mbit cell 8 avpkt 1000 mpu
64
```

Ο δεύτερος τρόπος λειτουργίας επιτρέπει την διάρθρωση μιας PHB (6 tc εντολές για τις AF PHBs και 3 tc εντολές για τις PHBs EF και BE). Παραδείγματος χάριν, για την διάρθρωση του δρομολογητή, ώστε να εξυπηρετεί πακέτα την PHBAF3, θα πρέπει να εκτελεστούν σε αυτόν, οι παρακάτω έξι tc εντολές:

```
#AF3
$TC class add dev $DEV parent 2:0 classid 2:3 cbq bandwidth 10Mbit \
rate 1Mbit avpkt 1000 prio 3 allot 1514 weight 1.5Mbit maxburst 21 defmap 0 borrow
$TC filter add dev $DEV parent 2:0 protocol ip prio 1 handle 3 tcindex classid 2:3
$TC qdisc add dev $DEV parent 2:3 gred setup DPs 3 default 2 grio
$TC qdisc change dev $DEV parent 2:3 gred limit 600K min 150K max 450K burst 200 \
avpkt 1000 bandwidth 10Mbit DP 1 probability 0.02 prio 2
$TC qdisc change dev $DEV parent 2:3 gred limit 600K min 150K max 450K burst 200 \
avpkt 1000 bandwidth 10Mbit DP 2 probability 0.04 prio 3
$TC qdisc change dev $DEV parent 2:3 gred limit 600K min 150K max 450K burst 200 \
avpkt 1000 bandwidth 10Mbit DP 3 probability 0.06 prio 4
```

Ενώ, για την διάρθρωση της κλάσης υπηρεσίας EF, πρέπει να εκτελεστούν οι παρακάτω τρεις tc εντολές:

```
#EF
$TC class add dev $DEV parent 2:0 classid 2:5 cbq bandwidth 10Mbit \
rate 1Mbit avpkt 1000 prio 1 allot 1514 weight 1.5Mbit maxburst 21 defmap 0
$TC filter add dev $DEV parent 2:0 protocol ip prio 1 handle 5 tcindex classid 2:5
$TC qdisc add dev $DEV parent 2:5 pfifo limit 5
```

Ο τρίτος τρόπος λειτουργίας επιτρέπει την διάθρωση των 14 PHBs της αρχιτεκτονικής Δ.Υ και του αρχικού σχήματος αναμονής (33 εντολές tc, βλ. παρ. 5.2).

Στους τρεις πρώτους τρόπους λειτουργία που περιγράφηκαν, τα πιθανόν λάθη που μπορεί να προκύψουν, κατά την εκτέλεση μιας εντολής διάρθρωσης, δεν διορθώνονται. Η διαδικασία συνεχίζεται με τις υπόλοιπες εντολές και τα τυχόν λάθη απλώς αναφέρονται στο τέλος της διαδικασίας. Ο τέταρτος και τελευταίος τρόπος λειτουργία επιτρέπει την διάρθρωση των 14 κλάσεων υπηρεσίας, όπως και ο τρίτος τρόπος λειτουργίας, αλλά αυτή την φορά με δυνατότητα διόρθωσης λαθών. Σε αυτόν τον τρόπο λειτουργία εάν κατά την εκτέλεση μιας εντολή προκύψει κάποιο σφάλμα, η εφαρμογή είναι σε θέση να το διορθώσει προτού συνεχίσει με την εκτέλεση των επόμενων εντολών. Φυσικά η εφαρμογή είναι σε θέση να διορθώσει μόνο λάθη τα οποία σχετίζονται με την διάρθρωση του δρομολογητή. Αν μια εντολή προσπαθήσει να ορίσει ένα σχήμα αναμονής ή ένα φίλτρο (εντολές tc add) το οποίο έχει ήδη οριστεί από μια προηγούμενη διάρθρωση του δρομολογητή, τότε προκύπτει ένα λάθος και η απάντηση του δρομολογητή σε αυτήν την αίτηση είναι: RTNETLINK: file already exists. Στον τελευταίο τρόπο λειτουργία αυτό το πρόβλημα αντιμετωπίζεται με την χρήση μια εντολής tc change αντί της εντολής tc add, για το συγκεκριμένο σχήμα αναμονής ή φίλτρο.

Οι τρεις πρώτοι τρόποι λειτουργίας εκτελούν μια σειρά εντολών διόρθωσης στους δρομολογητές, χωρίς να εμπεριέχουν κάποια λογική αντιμετώπισης λαθών. Η αύξηση των προς εκτέλεσης εντολών οδηγεί στην αύξηση της πολυπλοκότητας του προς εκτέλεσης έργου, από τον έναν τρόπο λειτουργία στον άλλο. Ο τελευταίος τρόπος λειτουργίας είναι έχει τον μεγαλύτερο βαθμό πολυπλοκότητας, αφού εκτός από τον μέγιστο αριθμό εντολών εμπεριέχει και την λογική αντιμετώπισης λαθών.

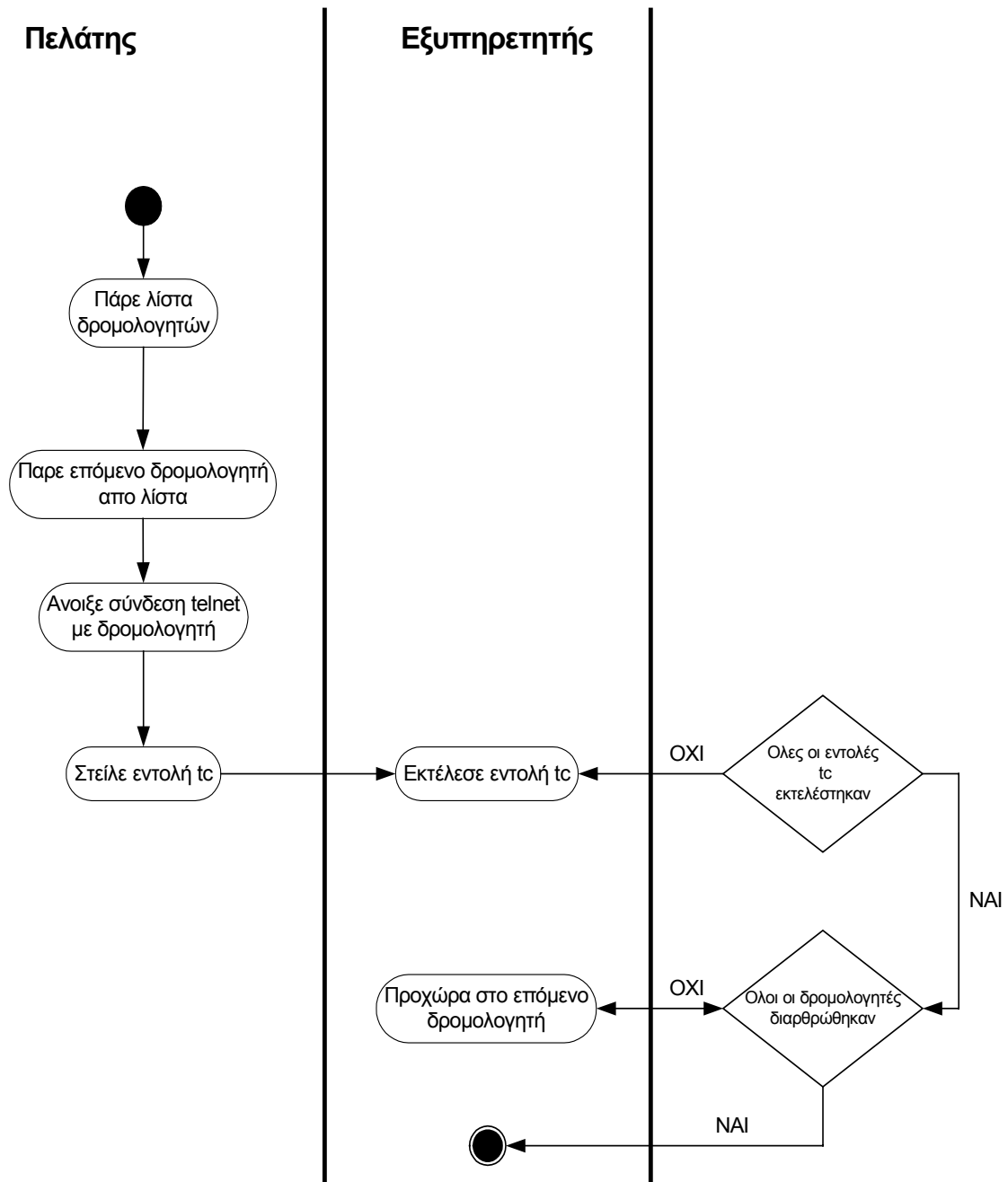
Σε αυτό το σημείο πρέπει να τονιστεί, ότι οι τέσσερις διαφορετικοί τρόποι λειτουργία δημιουργήθηκαν για αν επιτραπεί η διερεύνηση της επίδρασης της πολυπλοκότητας του προς εκτέλεση έργου, στις δυο διαφορετικές προσεγγίσεις υλοποιήσεις του συστήματος διάρθρωσης. Ο σωστός τρόπος λειτουργίας, που ακολουθείται από το σύστημα διαχείρισης δικτύου, που παρουσιάστηκε στο κεφαλαίο 4, είναι ο τελευταίος που επιτρέπει την διάρθρωση όλων των PHBs με διόρθωση λαθών.

5.3.1 Λειτουργία της εφαρμογής με βάση το μοντέλο πελάτη/εξυπηρετητή

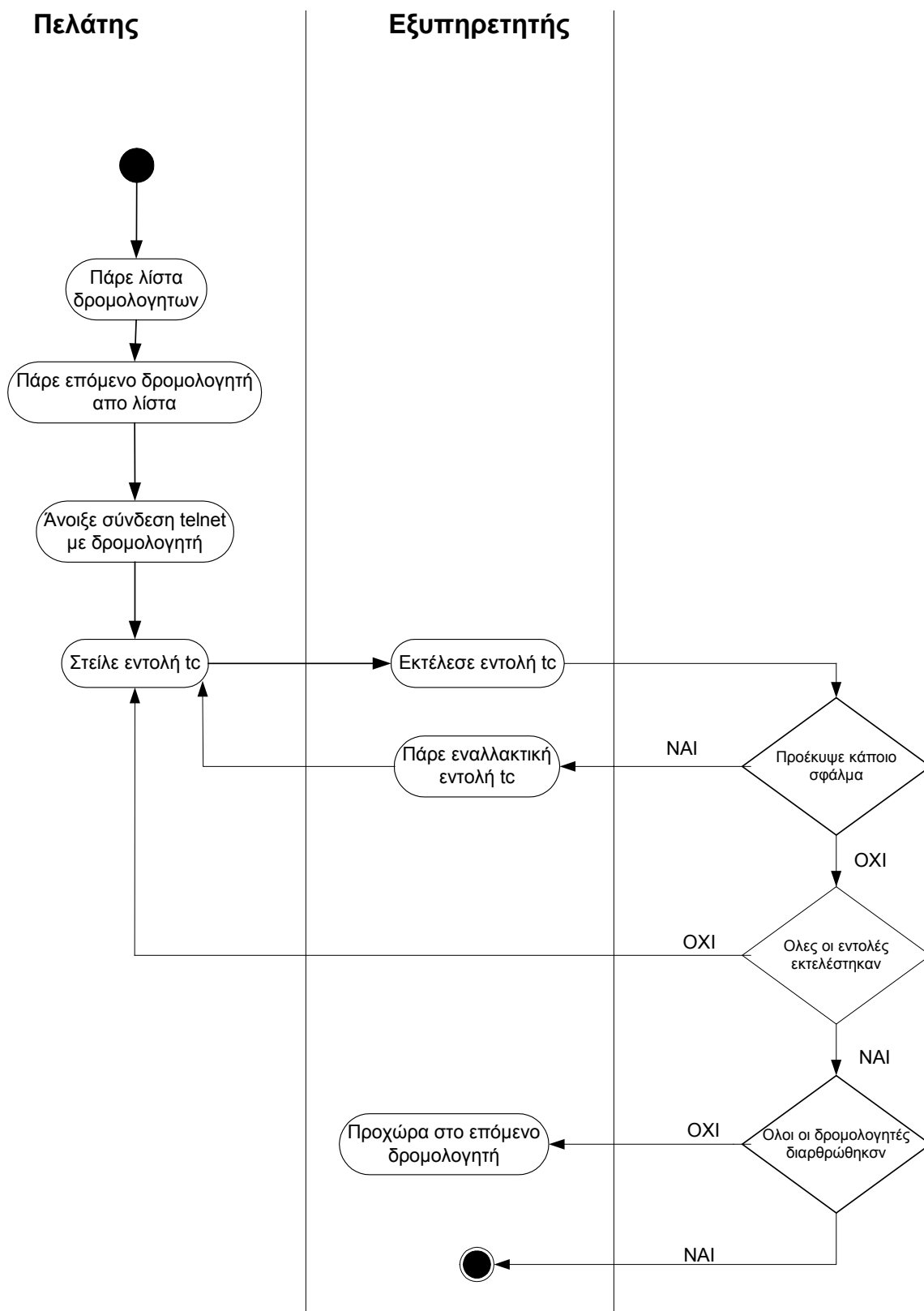
Ο όρος πελάτης/εξυπηρετητής (client/server) χρησιμοποιήθηκε για πρώτη φορά την δεκαετία του 1980 και αναφερόταν σε προσωπικούς υπολογιστές συνδεδεμένους σε ένα δίκτυο υπολογιστών. Στην γενική του μορφή η αρχιτεκτονική πελάτη/εξυπηρετητή είναι ένα μοντέλο δημιουργίας κατανεμημένων συστημάτων. Ως πελάτης θεωρείται το μηχάνημα το οποίο αιτείται την χρήση μια υπηρεσίας, που παρέχεται από ένα άλλο μηχάνημα το οποίο ονομάζεται εξυπηρετητής [Edelstein94, Schussel95]. Για την επικοινωνία των πελατών με τους εξυπηρετητές συνήθως χρησιμοποιείται η τεχνική της απομακρυσμένης κλήσης μεθόδου (Remote procedure call). Ένας πελάτης μπορεί να επικοινωνήσει με μια σειρά από εξυπηρετητές για την εκτέλεση μιας και μόνο εργασίας, ενώ ένας εξυπηρετητής μπορεί να προσφέρει υπηρεσίες σε πολλούς διαφορετικούς πελάτες.

Η εφαρμογή διάθρωσης δρομολογητών κορμού, με βάση το μοντέλο πελάτη/εξυπηρετητή, δεν είναι μια τυπική εφαρμογή αυτού του μοντέλου. Υπάρχει μόνο ένας πελάτης οποίος αιτείται την εκτέλεση μιας σειράς τε εντολών από τους εξυπηρετητές, που στην περίπτωση αυτή είναι οι δρομολογητές. Οι επικοινωνία πραγματοποιείται με χρήση μίας σύνδεσης Telnet [Postel93]. Ο πελάτης, που είναι η κεντρική μονάδα διάθρωσης, ανοίγει μια σειρά από συνδέσεις τύπου Telnet με τους δρομολογητές και αποστέλλει σε αυτούς τις τε εντολές προς εκτέλεση. Οι δρομολογητές, μέσω της ίδιας σύνδεσης, αποστέλλουν στο πελάτη το αποτέλεσμα της εκτέλεσης των εντολών, το οποίο χρησιμοποιείται ανάλογα με τον τρόπο λειτουργία που έχει επιλέγει. Στους τρεις πρώτους τρόπους αποθηκεύεται και παρουσιάζεται στον χρήστη μετά το τέλος όλης της διαδικασίας, ανεξάρτητα από την ύπαρξη λαθών. Στον τελευταίο τρόπο λειτουργίας αποστέλλεται μια νέα εντολή που διορθώνει το σφάλμα που προέκυψε κατά την διαδικασία. Η εφαρμογή είναι εξολοκλήρου υλοποιημένη σε Java. Η επίτευξη των συνδέσεων Telnet πραγματοποιείται με χρήση μια κλάσης Java, που προσφέρει υπηρεσίες δημιουργία και τερματισμού μια σύνδεσης Telnet, καθώς και την αποστολή και λήψη δεδομένων από αυτή την σύνδεση. Στα επόμενα σχήματα 5.2 και 5.3 παρουσιάζονται τα διαγράμματα λειτουργίας της εφαρμογής, χωρίς δυνατότητα διόρθωσης λαθών και με δυνατότητα διόρθωσης λαθών, αντίστοιχα.

Η διαδικασία διάρθρωσης με χρήση του μοντέλου αυτού είναι πλήρως κεντροποιημένη. Όλη η λογική του συστήματος βρίσκεται στο μηχανήμα του πελάτη (στην περίπτωση αυτή ο πελάτης είναι ο διαχειριστής του δικτύου). Ο πελάτης γνωρίζει την αλληλουχία εκτέλεσης των εντολών και τις εναλλακτικές εντολές που πρέπει να εκτελεστούν σε περίπτωση λάθους. Ο εξυπηρετητής απλώς εκτελεί τις εντολές που του αποστέλλονται και πληροφορεί για το αποτέλεσμα εκτέλεσης.



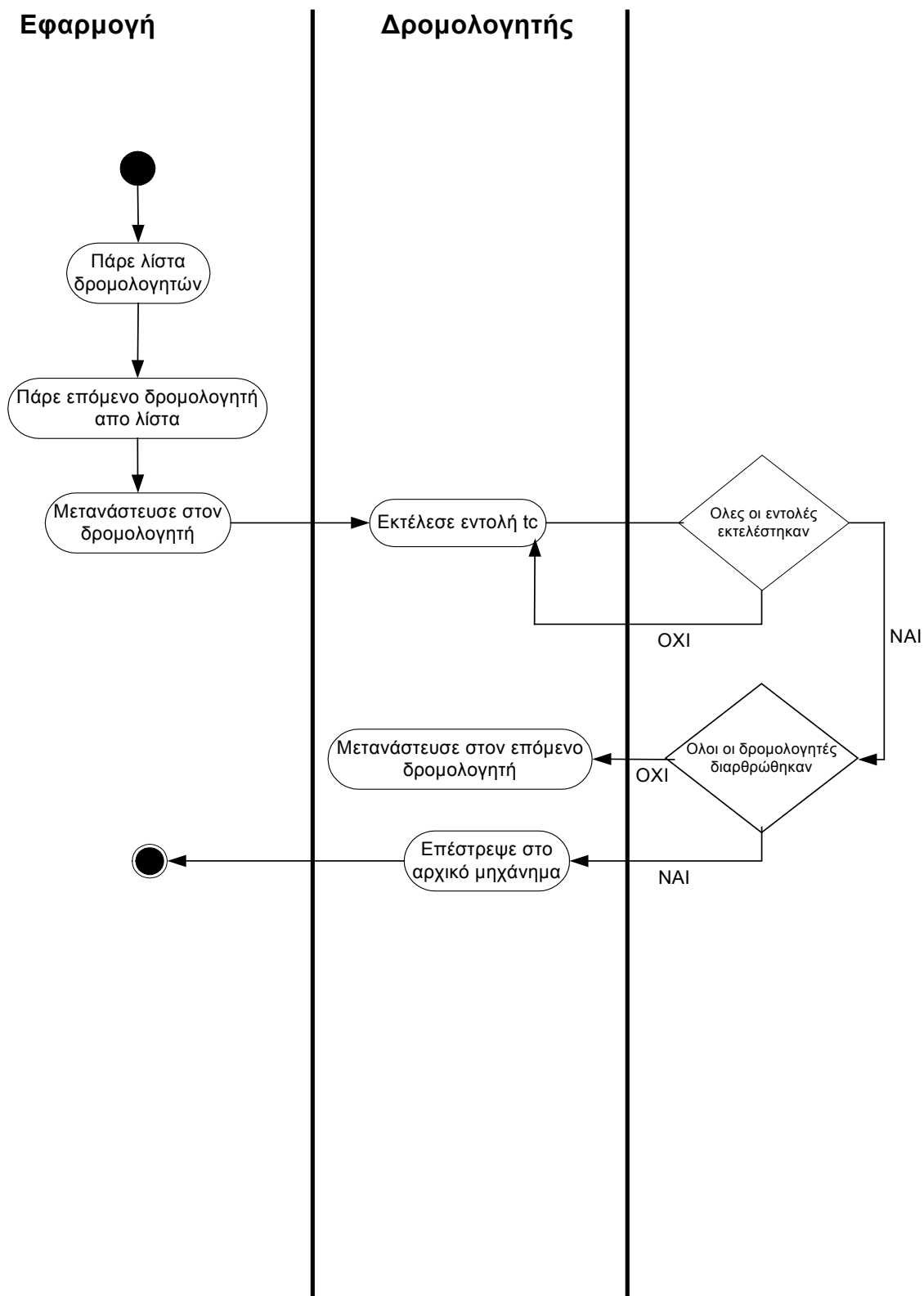
Σχήμα 5.2: Διάγραμμα λειτουργίας εφαρμογής πελάτη/εξυπηρετητή χωρίς διόρθωση λαθών



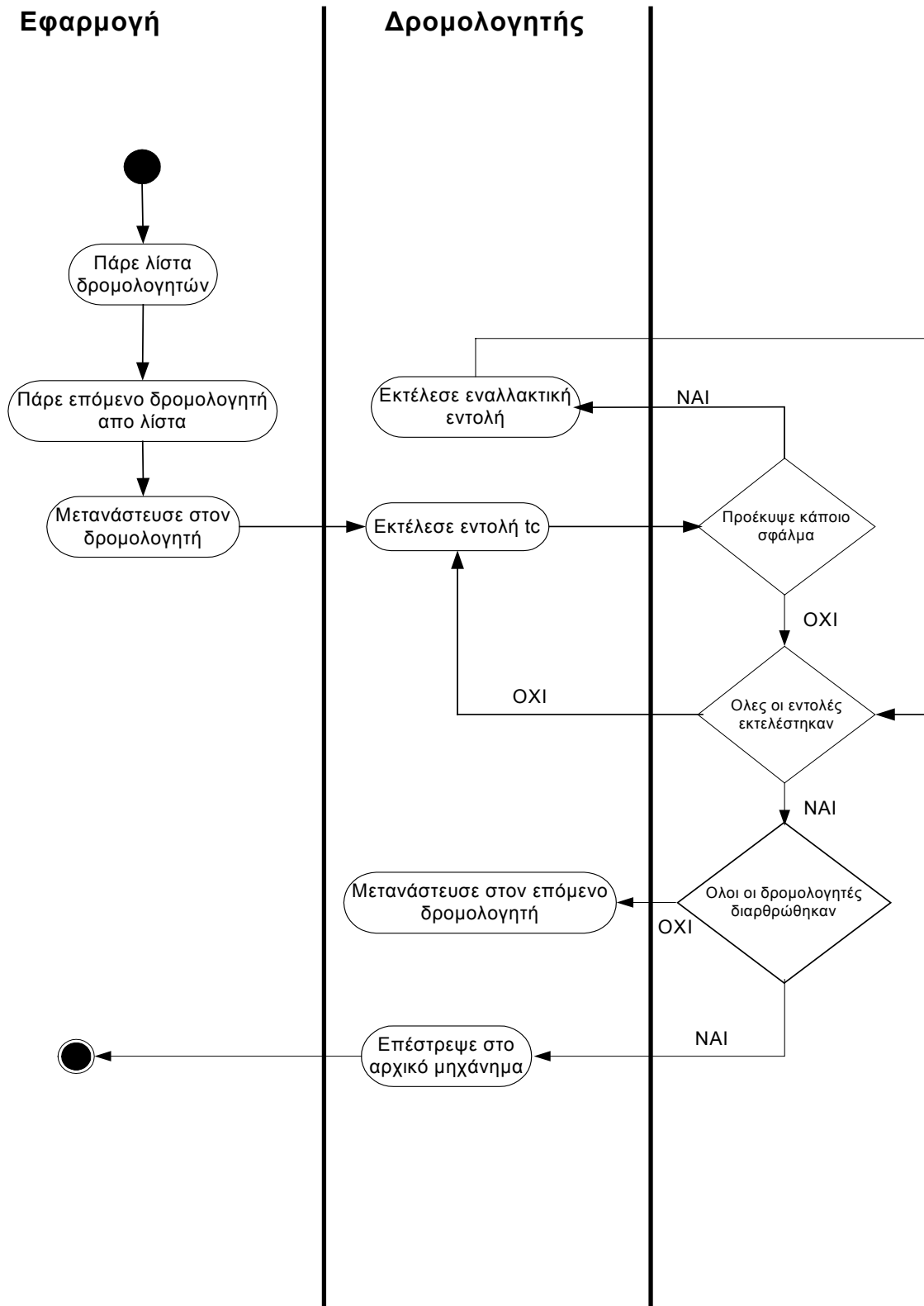
Σχήμα 5.3: Διάγραμμα λειτουργίας εφαρμογής πελάτη/εξυπηρετητή με διόρθωση λαθών

5.3.2 Λειτουργία της εφαρμογής με χρήση της ΤΚΠ

Στην περίπτωση των ΚΠ η διαδικασία διάρθρωσης πραγματοποιείται αποκεντροποιημένα. Ένα κινητός πράκτορας επισκέπτεται διαδοχικά όλους τους δρομολογητές και του διαρθρώνει κατάλληλα. Ο πράκτορας αυτός διαθέτει την λογική για την εκτέλεση των εντολών και την διόρθωση των λαθών. Η εφαρμογή δρα απλώς ως διεπαφή μεταξύ του χρήστη και του ΚΠ. Μετά την ανάθεση των προς διάρθρωση δρομολογητών, ο πράκτορας δρα αυτόνομα και επιστέφει στο αρχικό μηχάνημα μετά το πέρας της διαδικασίας, όπου πληροφορεί τον χρήστη για το αποτέλεσμα της. Έτσι λοιπόν, σε αντίθεση με το την προσέγγιση του μοντέλου πελάτη/εξυπηρετητή, με την χρήση των ΚΠ δεν υπάρχει ανάγκη απομακρυσμένης επικοινωνίας με το μηχάνημα της εφαρμογής για την λήψη των εντολών. Για την υποστήριξη της λειτουργίας του ΚΠ χρησιμοποιήθηκε η ΠΚΠ Grasshopper. Το σύστημα είναι υλοποιημένο με χρήση της γλώσσας προγραμματισμού Java. Στα επόμενα σχήματα 5.4 και 5.5 παρουσιάζονται τα διαγράμματα λειτουργίας της εφαρμογής κινητών πρακτόρων, χωρίς δυνατότητα διόρθωσης λαθών και με δυνατότητα διόρθωσης λαθών, αντίστοιχα.



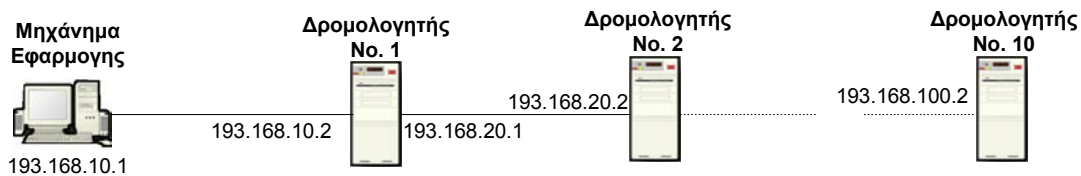
Σχήμα 5.4: Διάγραμμα λειτουργίας εφαρμογής κινητών πρακτόρων χωρίς διόρθωση λαθών



Σχήμα 5.5: Διάγραμμα λειτουργίας εφαρμογής κινητών πρακτόρων με διόρθωση λαθών

5.4 Πειραματική σύγκριση

Η σύγκριση των δυο προσεγγίσεων εστιάστηκε στον χρόνο εκτέλεσης που χρειάζεται κάθε μια, για την διάρθρωση ενός συγκεκριμένου αριθμού δρομολογητών. Τα πειράματα πραγματοποιήθηκαν για διαφορετικό αριθμό δρομολογητών, ώστε να είναι δυνατή η διερεύνηση της επίδρασης του αριθμού αυτών στον χρόνο εκτέλεσης της εφαρμογής. Ένας άλλος παράγοντας που λήφθηκε υπόψη στην σύγκριση, είναι το πόσο της κίνησης που δημιουργήθηκε στο δίκτυο από κάθε εφαρμογή, για την πραγματοποίηση της διάρθρωσης. Για την λήψη των μετρήσεων χρησιμοποιήθηκε ένα πειραματικό IP δίκτυο το οποίο απεικονίζεται στον σχήμα 5.6



Σχήμα 5.6: πειραματικό IP δίκτυο

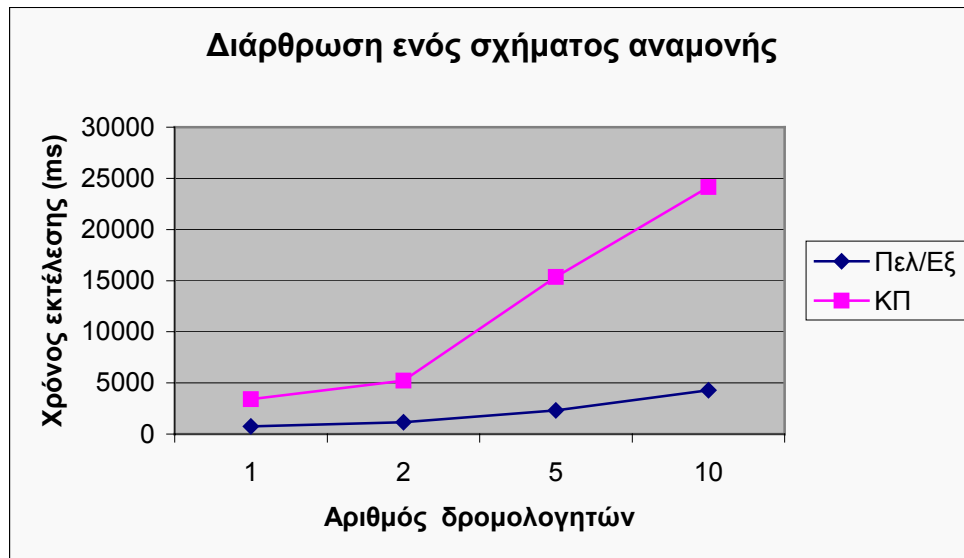
Το πειρακτικό δίκτυο αποτελείται από δέκα δρομολογητές τύπου Linux εφοδιασμένους με το εργαλείο tc. Συνδεδεμένο στο δίκτυο αυτό υπάρχει και το κεντρικό μηχάνημα στο οποίο εκτελούνται οι δυο εφαρμογές, εφοδιασμένο με το λειτουργικό σύστημα Windows 2000. Το εύρος ζώνης του δικτύου είναι 10 MBps. Για την μέτρηση του συνολικού χρόνου εκτέλεσης χρησιμοποιηθεί η βιβλιοθήκη Date της Java, η οποία παρέχει τον τρέχον χρόνο του μηχανήματος στο οποίο εκτελείται. Με χρήση της βιβλιοθήκης αυτής μετρήθηκε για κάθε πείραμα, ο χρόνος εκκίνησης της εφαρμογής και ο χρόνος τερματισμού. Έτσι, ο συνολικός χρόνος προέκυψε από την αφαίρεση αυτών των δυο μετρήσεων (εξίσωση 5.2). Για την μέτρηση του ποσού της κίνησης, που δημιουργείται από της δυο εφαρμογές στο δικτύου χρησιμοποιήθηκε το εργαλείο windump [windump]

$$T_{\text{εκτέλεσης}} = T_{\text{τερματισμού}} - T_{\text{εκκίνησης}} \quad (5.2)$$

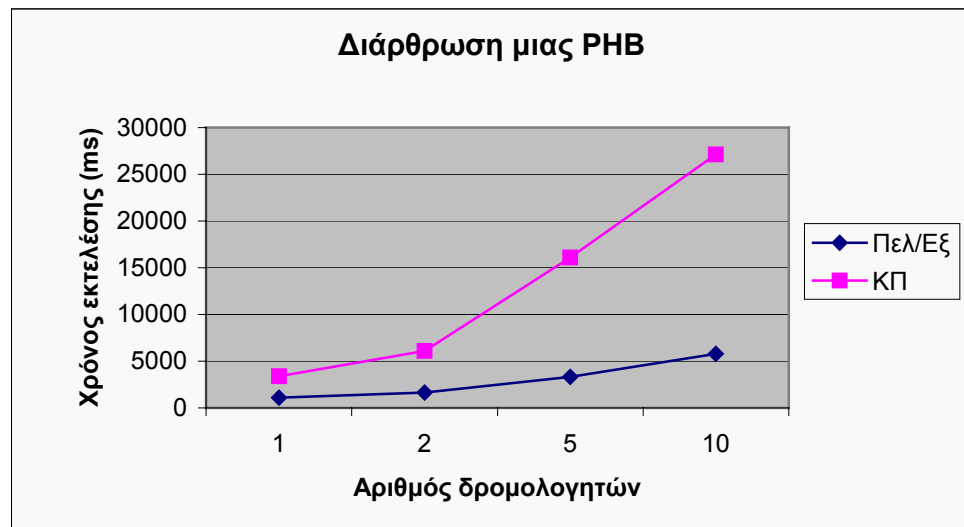
Η πειραματική διαδικασία περιλαμβάνει την εκτέλεση των τεσσάρων διαφορετικών τρόπων λειτουργίας, των δυο εφαρμογών, σε ένα αυξανόμενο αριθμό δρομολογητών. Συγκεκριμένα κάθε τρόπος λειτουργίας χρησιμοποιείται για την διάρθρωση ενός, δυο, πέντε και δέκα δρομολογητών. Έτσι, είναι δυνατόν να διερευνηθεί η επίδραση στη

πολυπλοκότητας του προς εκτέλεση έργου, στον χρόνο εκτέλεσης, καθώς επίσης και η επίδραση του αριθμού δρομολογητών στον χρόνο αυτό.

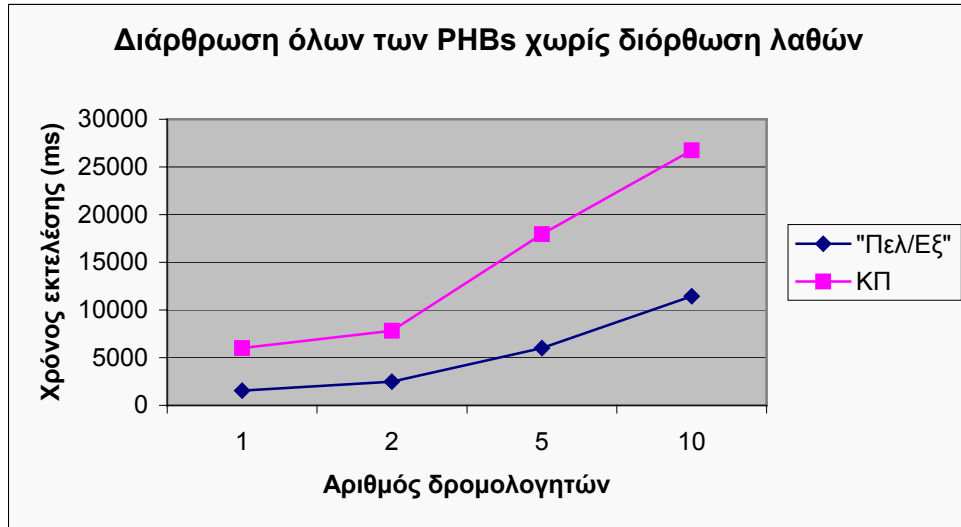
Στα παρακάτω διαγράμματα παρουσιάζεται ο χρόνος εκτέλεσης που απαιτείται από τις δυο εφαρμογές, για την διάθρωση ενός συγκεκριμένου αριθμού δρομολογητών. Κάθε διάγραμμα παρουσιάζει τον χρόνο εκτέλεσης για κάθε έναν από τους τέσσερις διαφορετικούς τρόπους λειτουργία, για έναν διαφορετικό αριθμό δρομολογητών.



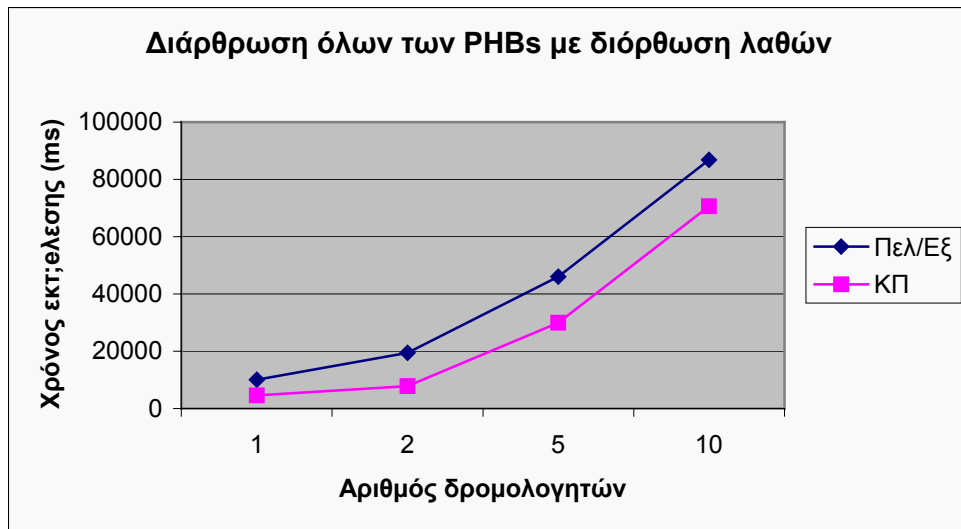
Διάγραμμα 5.1: Χρόνος εκτέλεσης για την διάρθρωση ενός σχήματος αναμονής



Διάγραμμα 5.2: Χρόνος εκτέλεσης για την διάρθρωση μιας PHB



Διάγραμμα 5.3: Χρόνος εκτέλεσης για την διάρθρωση όλων των PHBs χωρίς διόρθωση λαθών



Διάγραμμα 5.4: Χρόνος εκτέλεσης για την διάρθρωση όλων των PHBs χωρίς διόρθωση λαθών

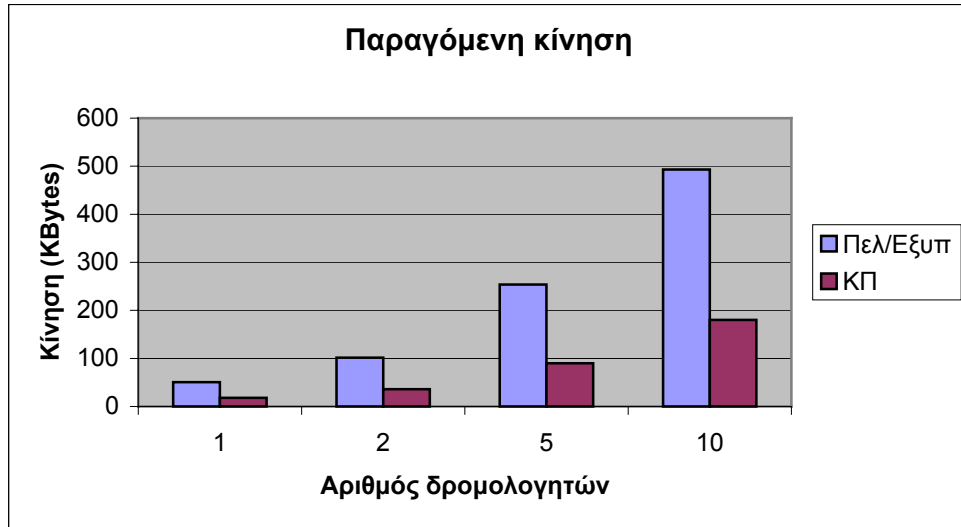
Για την αξιολόγηση της απόδοσης των δυο εφαρμογών, στην περίπτωση διάρθρωσης με διόρθωση λαθών, ήταν απαραίτητη η ύπαρξη σφαλμάτων κατά την διαδικασία διάρθρωσης. Αυτό επιτεύχθηκε με την διεξαγωγή του πειράματος αυτού (διάγραμμα 5.4), σε ένα δίκτυο που είχε ήδη υποστεί την διαδικασία προπαρασκευής των δρομολογητών του.

Με βάση τα πειραματικά αποτελέσματα, η εφαρμογή πελάτη/εξυπηρετητή παρουσιάζει σαφώς καλύτερους χρόνους εκτέλεσης από την προσέγγιση των ΚΠ, για τους τρεις πρώτους τρόπους λειτουργίας (διαγράμματα 5.1, 5.2 και 5.3). Η εφαρμογή με βάση την

ΤΚΠ παρουσιάζει καλύτερους χρόνους εκτέλεσης μόνο στην περίπτωση της διάρθρωσης με δυνατότητα διόρθωσης λαθών. Όσο η πολυπλοκότητα του προς εκτέλεση έργου παραμένει μικρή η εφαρμογή πελάτη/εξυπηρετητή έχει καλύτερη απόδοση από αυτή των ΚΠ. Οι ΚΠ παρουσιάζουν καλύτερα αποτελέσματα μόνο όταν το προς εκτέλεση έργο εμπεριέχει την λογική διόρθωσης λαθών, που αυξάνει σημαντικά την πολυπλοκότητα του. Μπορούμε επίσης να παρατηρήσουμε ότι για του τρεις πρώτους τρόπους λειτουργίας, όπου υπερέχει η εφαρμογή πελάτη/εξυπηρετητή, η διάφορα στον χρόνο εκτέλεσης μεταξύ των δυο προσεγγίσεων μειώνεται, καθώς αυξάνει η πολυπλοκότητα του προς εκτέλεση έργου.

Η αύξηση των δρομολογητών του δικτύου αυξάνει όπως ήταν αναμενόμενο τον χρόνο εκτέλεσης της εφαρμογής. Ένα σημείο το οποίο αξίζει να επισημανθεί, είναι ότι η αύξηση των δρομολογητών έχει μεγαλύτερη επίδραση στον χρόνο εκτέλεσης της εφαρμογής ΚΠ. Ο χρόνος εκτέλεσης της εφαρμογής αυτής αυξάνεται απότομα, όταν οι δρομολογητές αυξάνονται από δυο σε πέντε και από πέντε σε δέκα. Αυτή η αύξηση, στην περίπτωση των ΚΠ οφείλεται στην αύξηση του χρόνου μετανάστευσης, που απαιτείται για να επισκεφτεί ο πράκτορας όλους τους δρομολογητές. Στην περίπτωση της εφαρμογής πελάτη/εξυπηρετητή, δεν υπάρχει απότομη αύξηση του χρόνου εκτέλεσης με την αύξηση των δρομολογητών. Η αύξηση αυτή είναι ανάλογη με την αύξηση του αριθμού μηνυμάτων που ανταλλάσσονται μεταξύ του πελάτη και του εξυπηρετητή, για την διάθρωση του δρομολογητή.

Πέρα από τον χρόνο εκτέλεσης μιας εφαρμογής, ένα άλλος σημαντικός παράγοντας που αποτελεί κριτήριο εφαρμοσιμότητάς της, είναι ο τρόπος με τον οποίο χρησιμοποιεί το εύρος ζώνης του δικτύου. Το πόσο της κίνησης που παράγουν οι δυο εφαρμογές στο δίκτυο παρουσιάζεται στο ιστόγραμμα 5.5. Η μέτρηση της κίνησης πραγματοποιήθηκε μόνο για τον τελευταίο τρόπο λειτουργίας. Σε αυτόν τον τρόπο λειτουργίας, το πόσο της κίνησης που παράγεται από τις δυο εφαρμογές μεγιστοποιείται. Επίσης, η διάφορα στον χρόνο εκτέλεσης σε όλες τις άλλες περιπτώσεις δεν αφήνει περιθώρια αμφισβήτησης της υπεροχής του μοντέλου πελάτη/εξυπηρετητή.



Ιστόγραμμα 5.1: Παραγόμενη κίνηση από τις δυο εφαρμογές

Σύμφωνα με τα πειραματικά δεδομένα που παρουσιάζονται στο παραπάνω διάγραμμα, η εφαρμογή ΚΠ χρησιμοποιεί αποτελεσματικότερα το εύρος ζώνης του δικτύου για την διάρθρωση των δρομολογητών, από ότι η εφαρμογή με βάση το μοντέλο πελάτη/εξυπηρετητή. Επίσης παρατηρούμε ότι η το πόσο της παραγόμενης κίνησης αυξάνει αναλογικά με τον αριθμό των δρομολογητών, όπως ήταν αναμενόμενο. Η μεγαλύτερη παραγωγή κίνησης στο δίκτυο από την εφαρμογή πελάτη/εξυπηρετητή, οφείλεται στην ανταλλαγή μεγάλου αριθμού μηνυμάτων μεταξύ της εφαρμογής και του δρομολογητή, ώστε να επιτευχθεί η διάρθρωση του τελευταίου. Ενώ στην περίπτωση των ΚΠ η κίνηση που παράγεται οφείλεται αποκλειστικά στη μετανάστευση του πράκτορα, από τον έναν δρομολογητή στον άλλο. Για την συγκεκριμένη υλοποίηση η μετανάστευση του πράκτορα απαιτεί ένα εύρος ζώνης της τάξεως των 18KBytes.

5.5 Σύνοψη κεφαλαίου – συμπεράσματα

Στο κεφάλαιο αυτό παρουσιάστηκε μια σύγκριση της ΤΚΠ με το μοντέλο πελάτη/εξυπηρετητή. Συγκρίθηκε η κεντροποιημένη διαχείριση που εκφράζεται από το μοντέλο πελάτη/εξυπηρετητή με την αποκεντροποιημένη διαχείριση που εκφράζεται από την ΤΚΠ. Για τον λόγο αυτό, δημιουργήθηκαν δυο εφαρμογές, μια για κάθε προσέγγιση, που εκτελούν την αρχική διάρθρωση ενός IP δικτύου ΔΥ, αποτελούμενο από δρομολογητές Linux. Η σύγκριση στηρίχθηκε στο χρόνο εκτέλεσης που απαιτεί κάθε μια από τις δυο προσεγγίσεις για την εκτέλεση μιας σειράς διεργασιών, διαφορετικής πολυπλοκότητας. Η διαφορετική πολυπλοκότητα του προς εκτέλεση έργου, επέτρεψε την διερεύνηση της επίδραση που έχει στην απόδοση των εφαρμογών η λογική που εμπεριέχεται στο έργο που καλούνται να εκτελέσουν. Ένας άλλο παράγοντας ο οποίος ερευνήθηκε, ήταν ο τρόπος με το οποίο χρησιμοποιούν το εύρος ζώνης του δικτύου οι εφαρμογές. Ένα πειραματικό IP δίκτυο αποτελούμενο από δρομολογητές τύπου Linux, χρησιμοποιήθηκε για την διεξαγωγή των πειραμάτων.

Με βάση τα πειρακτικά αποτελέσματα, η εφαρμογή πελάτη/εξυπηρετητή παρουσιάζει πολύ καλύτερους χρόνους εκτέλεσης, όταν το προς εκτέλεση έργο είναι αρκετά απλό (πρώτος και δεύτερος τρόπος λειτουργίας). Η εφαρμογή πελάτη/εξυπηρετητή παρουσιάζει σε αυτή την περίπτωση χρόνους εκτέλεσης που είναι έως και 82% καλύτεροι από αυτούς της αντίστοιχης εφαρμογής με βάση την ΤΚΠ. Η διάφορα αυτή μειώνεται στο 58%, όταν η πολυπλοκότητα του προς εκτέλεση έργου αυξάνει (τρίτος τρόπος λειτουργίας). Η εφαρμογή ΚΠ παρουσιάζει χρόνους εκτέλεσης κατά ένα ποσοστό 20% καλύτερους από αυτούς της εφαρμογής με βάση το μοντέλο πελάτη/εξυπηρετητή, όταν το προς εκτέλεση έργο αυξάνει αρκετά την πολυπλοκότητα του εμπεριέχοντας λογική για διόρθωση λαθών (τέταρτος τρόπος λειτουργίας). Σε αυτήν την τελευταία περίπτωση, το εύρος ζώνης που χρησιμοποιείται από την εφαρμογή ΚΠ είναι το 1/3 από αυτό που χρησιμοποιείται από την εφαρμογή πελάτη/εξυπηρετητή.

Η σύγκριση των δυο μοντέλων απέδειξε ότι η απάντηση στην ερώτηση πιο από τα δυο είναι καταλληλότερο για την δημιουργία μιας κατανεμημένης εφαρμογής, εξαρτάται από την πολυπλοκότητα του προς εκτέλεση έργου. Όταν το διεργασία που πρέπει να εκτελεστεί, σε έναν απομακρυσμένο μηχανήμα της κατανεμημένης εφαρμογής, δεν

περιέχει κάποιο είδος λογικής, αλλά είναι μια απλή εκτέλεση μιας σειράς εντολών, τότε καταλληλότερο μοντέλο είναι αυτό του πελάτη/εξυπηρετητή. Στο κεντροποιημένο αυτό μοντέλο όλη η λογική της εφαρμογής διατηρείται σε ένα κεντρικό μηχανήμα που ελέγχει όλη την διαδικασία. Όσο η επικοινωνία με τα απομακρυσμένα μηχανήματα είναι περιορισμένη στην απλή ανάθεση των εντολών, ο χρόνος εκτέλεσης που προκύπτει είναι μικρός. Όταν το προς εκτέλεση έργο περιλαμβάνει την λήψη κάποιων αποφάσεων, για την περαιτέρω εκτέλεση του, τότε η επικοινωνία με το κεντρικό μηχανήμα αυξάνεται οδηγώντας σε μεγαλύτερους χρόνους εκτέλεσης. Η αύξηση της απομακρυσμένης επικοινωνίας οδηγεί επίσης σε ανάλογη αύξηση του ποσού κίνησης, που παράγεται από την εφαρμογή στο δίκτυο.

Όταν διεργασία που πρέπει να εκτελεστεί εμπεριέχει κάποιο είδος λογικής, που αυξάνει την πολυπλοκότητα της, τότε καταλληλότερο μοντέλο είναι αυτό των ΚΠ. Όσο το έργο είναι σχετικά απλό ο χρόνος μετανάστευσης του πράκτορα έχει αρνητική επίδραση στο χρόνο εκτέλεσης της εφαρμογής. Το μειονέκτημα αυτό για την εκτέλεση απλών διεργασιών σε απομακρυσμένα μηχανήματα μετατρέπεται σε πλεονέκτημα, όταν η πολυπλοκότητα αυξάνει. Τότε, ο πράκτορας περιέχοντας στο κώδικα του όλη την απαραίτητη λογική για την εκτέλεση της διεργασίας δεν είναι αναγκασμένος να επικοινωνεί με κάποια κεντρική μονάδα, όπως στην περίπτωση του πελάτη/εξυπηρετητή. Έτσι εξηγείται η καλύτερη απόδοση του μοντέλου αυτού σε πολύπλοκες διεργασίες και η καλύτερη χρήση του εύρους ζώνης του δικτύου.

5.6 Βιβλιογραφικές Αναφορές

[Kalogeropoulos03] S. Kalogeropoulos, T. Kotsilieris, G.Karetsos, A. Michalas, V.Loumos and E.Kayafas, “A Performance Evaluation of Mobile Agent technology in comparison to the Client/Server paradigm for QoS Configuration in DiffServ Domains”, *IAESTED International Conference on Artificial Intelligence and Applications (AIA 2003)*, Austria, February 2003.

[Carzaniga97] A. Carzaniga, B. P. Picco, and G. Vigna, “Designing Distributed Applications with Mobile Code Paradigms”, *Proc 19th International Conference on Software Engineering (ICSE'97)*, Boston USA, 1997, pp 22-32.

[Straier97] M. Straier and M. Schwehm, “A Performance Model for Mobile Agent Systems”, *Proc. Distributed Processing Techniques and Applications DPPTA'97*. Vol. II, Las Vegas 1997, pp. 1132-1140.

[Ismail99] L. Ismail and D. Hagimont, “A Performance Evaluation of the Mobile Agent Paradigm”, *Proc. Fourteenth Conference on Object-Oriented Programming, Systems, Languages and Applications (OOPSLA)*, Denver, November 1999, pp.306-313.

[Kotsilieris02] T. Kotsilieris, S. Kalogeropoulos, A. Michalas, P. Zikos, E. Vlachos, G. Karetsos and V. Loumos, “Performance Evaluation of Mobile Agent Technology for the Configuration and Auditing of SLSS in DiffServ Enabled IP Networks”, *Submitted for publication in the Journal of Telecommunication Systems, Modeling, Analysis, Design and Management*, (under review process), 2002.

[Stanic01] M P. Stanic, “tc – traffic control, Linux QoS control tool”, *Work in progress*, Διαθέσιμο στην ηλεκτρονική διεύθυνση <http://www.rns-nis.co.yu/~mps/linux-tc.ps.gz>

[Nichols98] K. Nichols, S. Blake, F. Baker and D. Black, “Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers”, *IETF RFC 2474*, December 1998

[Edelstein94] H. Edelstein, “Unraveling Client/Server Architecture”, *DBMS* May 1994, pp 34-40

[Schussel95] G Schussel,. “Client/Server Past, Present, and Future”, 1995

Διαθέσιμο στην ηλεκτρονική διεύθυνση: <http://www.dciexpo.com/geos>.

[Postel93] J. Postel and J. Reynolds, “TELNET PROTOCOL SPECIFICATION”, *IETF RFC854*, May 1983.

[windump] Διαθέσιμο στην ηλεκτρονική διεύθυνση: <http://windump.polito.it>.

ΚΕΦΑΛΑΙΟ 6

ΤΕΧΝΙΚΕΣ ΒΕΛΤΙΩΣΗΣ ΤΗΣ ΑΠΟΔΟΣΗΣ ΚΑΤΑΝΕΜΗΜΕΝΩΝ ΣΥΣΤΗΜΑΤΩΝ ΚΙΝΗΤΩΝ ΠΡΑΚΤΟΡΩΝ

6.1 Εισαγωγή

Τα αποτελέσματα της έρευνας στην περιοχή της ΤΚΠ, έχουν επιδείξει τη χρησιμότητα αυτού του παραδείγματος στην κατασκευή μιας πλειάδας κατανεμημένων εφαρμογών και συστημάτων. Τα τελευταία χρόνια, έχουμε παραστεί μάρτυρες ενός ενθουσιώδους ενδιαφέροντος για τη ΤΚΠ και μια πληθώρα ερευνητικών δραστηριοτήτων, που αφορούν την τεχνολογία αυτή, έχουν πραγματοποιηθεί. Το ενδιαφέρον για την ΤΚΠ παρακινήθηκε κυρίως από τα πλεονεκτήματα που παρέχει η χρήση της, σε ένα δίκτυο με περιορισμένο εύρος ζώνης. [Dag95, Gray96A]. Παρ' όλη αυτή την προσπάθεια, ένας αριθμός ζητημάτων που αφορούν κυρίως τη βελτιστοποίηση της απόδοσης εφαρμογών ΚΠ παραμένουν ανοιχτά. Στο προηγούμενο κεφάλαιο αποδείχθηκε ότι η απόδοση ενός συστήματος ΚΠ είναι υποδεέστερη από αυτή ενός συστήματος υλοποιημένου με βάση το μοντέλο πελάτη/εξυπηρετητή, για την εκτέλεση μιας απλής διεργασίας. Αρκετά τεχνικής και μη φύσεως εμπόδια πρέπει να υπερπηδηθούν προτού η ΤΚΠ υιοθετηθεί ευρέως στον τομέα των κατανεμημένων εφαρμογών. Μια σημαντική παράμετρος σχετική με για την εφαρμοσιμότητα και την “επιβίωση” μιας εφαρμογής και κυρίως μίας που κατανέμεται μέσω του διαδικτύου, είναι η απόδοση.

Ένας ΚΠ κατά την διάρκεια την λειτουργίας του συνήθως επισκέπτεται μια σειρά από μηχανήματα-στόχους, για την εκτέλεση ενός συγκεκριμένου έργου. Η βελτίωση της απόδοσης ενός συστήματος ΚΠ ανάγεται σε ένα πρόβλημα ανεύρεσης της βέλτιστης διαδρομής, που πρέπει να ακολουθήσει ο πράκτορας. Ο σχεδιασμός εκτέλεσης έργου είναι ένα θέμα το οποίο έχει ερευνηθεί διεξοδικά στον τομέα την τεχνητής νοημοσύνης [Etioni92, Sacerdoti75, Fikes72]. Γενικά, ο σχεδιασμός ορίζεται ως η επιλογή μια αλληλουχίας διεργασιών που έχουν ως αποτέλεσμα την επίτευξη ενός συγκεκριμένου έργου, με το ελάχιστο κόστος. Αρκετές μέθοδοι έχουν προταθεί για τον σχεδιασμό

διεργασιών που διεξάγονται σε στατικά περιβάλλοντα, όπου οι συνθήκες δεν μεταβάλλονται κατά την διάρκεια εκτέλεσης [Fikes71, Sacerdoti77]. Δυστυχώς, οι μέθοδοι αυτοί δεν μπορούν να εφαρμοστούν στην περίπτωση των ΚΠ, οι οποίο εκτελούνται σε ένα περιβάλλον το οποίο μεταβάλλεται δυναμικά. Κατά την διάρκεια εκτέλεσης ενός πράκτορα, τόσο η κατάσταση του υποκείμενου δικτύου όσο και των μηχανημάτων-στόχων δεν παραμένει σταθερή. Το πρόβλημα βελτίωσης της απόδοσης συστημάτων ΚΠ παρουσιάζει μεγαλύτερη αναλογία με το πολύ γνωστό πρόβλημα περιοδεύοντος πωλητή (Traveling Salesman Problem, TSP) [TSP]. Το πρόβλημα αυτό έχει μελετηθεί αρκετά, όμως λόγω μιας μικρής διαφοροποίησης οι προτεινόμενες λύσεις του δεν μπορούν να χρησιμοποιηθούν στην περίπτωση βελτίωσης της απόδοσης εφαρμογών ΚΠ.

Σε κάθε πρόβλημα ανεύρεσης βέλτιστης διαδρομής μεταξύ ενός συγκεκριμένου αριθμού σημείων, υπάρχουν μια σειρά από κόστη μετακίνησης βάσει των οποίων αναζητείται η βέλτιστη διαδρομή. Στη περίπτωση του TSP παραδείγματος χάριν, τα κόστη αυτά είναι η απόσταση μεταξύ των πόλεων. Οι παράμετροι οι οποίοι επηρεάζουν την απόδοση ενός ΚΠ είναι οι συνθήκες του υποκείμενου δικτύου (καθυστέρηση, διαθέσιμο εύρος ζώνης, απώλειες) και ο φόρτος εργασίας της Κεντρικής Μονάδας Επεξεργασίας (ΚΜΕ) των μηχανημάτων-στόχων. Η κατάσταση του υποκείμενου δικτύου επηρεάζει τον χρόνο μετανάστευσης του πράκτορα, ενώ ο φόρτος εργασίας της ΚΜΕ επηρεάζει τον χρόνο εκτέλεσης του. Οι δυο παραπάνω παράμετροι αποτελούν τα κόστη, βάσει των οποίων θα αναζητηθεί η βέλτιστη διαδρομή στην περίπτωση των ΚΠ.

Πέρα από την εύρεση της βέλτιστης διαδρομής, μια λύση που εύκολα έρχεται στο μυαλό και που αφορά τη βελτίωση της απόδοσης μιας εφαρμογής ΚΠ είναι η ακόλουθη: “Σε μια εφαρμογή που βασίζεται σε ΚΠ αντί να χρησιμοποιείται μονάχα ένας ΚΠ για την εκτέλεση μιας συγκεκριμένης εργασίας, να χρησιμοποιηθούν πολλοί ΚΠ που συνεργάζονται για να φέρουν σε πέρας την ίδια εργασία, εάν η φύση της εργασίας επιτρέπει κάτι τέτοιο”. Η προσέγγιση αυτή του προβλήματος επιφέρει μια ερώτηση: “Εάν η βελτίωση της απόδοσης με την κατάρτιση μιας εργασίας σε πολλούς πράκτορες θεωρηθεί δεδομένη, τότε ποιος είναι ο αριθμός των πρακτόρων που βελτιστοποιεί την απόδοση της εφαρμογής;”. Μια βεβαιωμένη απάντηση στην ερώτηση αυτή είναι να

χρησιμοποιηθεί ένας πράκτορας για κάθε μηχανήμα-στόχο. Δυστυχώς, η χρήση ΚΠ δεν είναι ανέξοδη. Η μετανάστευση κώδικα επιφέρει επιπλέον κίνηση στο δίκτυο. Επιπρόσθετα, οι ΚΠ απορροφούν μεγάλη ποσότητα υπολογιστικών πόρων από τον κόμβο που τους φιλοξενεί.

Οι δυο λοιπόν βασικές ιδέες για την βελτίωση της απόδοσης μια εφαρμογής ΚΠ, είναι η ανεύρεση της βέλτιστης διαδρομής και η κατάτμηση μιας εργασίας σε πολλούς πράκτορες. Η βασική διάφορα στην εφαρμογή τους, είναι ότι στην πρώτη περίπτωση δεν υπάρχει ανάγκη αλλαγής στον σχεδιασμό, ενώ δεύτερη περίπτωση ο σχεδιασμός αλλάζει, ώστε μια σειρά από πράκτορες να συνεργάζονται για την εκτέλεση μιας διεργασίας.

Στο κεφαλαίο αυτό προτείνονται και αξιολογούνται διάφοροι μηχανισμοί βελτίωσης την απόδοσης εφαρμογών ΚΠ. Συγκεκριμένα, θα παρουσιαστούν τρεις τρόποι (μηχανισμοί) βελτίωσης της απόδοσης. Οι δυο πρώτοι στηρίζονται στην ιδέα της βέλτιστης διαδρομής και ο τρίτος στην ιδέα της κατάτμησης εργασίας. Η αξιολογή όλων των μεθόδων βασίζεται σε πραγματικές μετρήσεις. Για αυτόν τον σκοπό δημιουργήθηκε, για κάθε έναν από αυτούς, μια πειραματική κατανεμημένη εφαρμογή διαχείρισης δικτύου.

6.2 Καθορισμός του προβλήματος

Σκοπός όλων των μηχανισμών, που θα παρουσιαστούν σε αυτό το κεφαλαίο, είναι η βελτίωση του συνολικού χρόνου εκτέλεσης μιας κατανεμημένης εφαρμογής ΚΠ, όπου μια διεργασία πρέπει να εκτελεστεί σε έναν προκαθορισμένο αριθμό μηχανημάτων-στόχων. Ο Κ. Moizumi στην διδακτορική του διατριβή με τίτλο “Προβλήματα σχεδιασμού Κινητών Πρακτόρων” [Moizumi98] μελέτησε ένα παρόμοιο πρόβλημα το οποίο ονόμασε πρόβλημα περιοδεύοντος πράκτορα. Η ερευνά αυτή εστιάστηκε κυρίως σε κατανεμημένες εφαρμογές αναζήτησης δεδομένων. Σε αυτήν την περίπτωση, ο πράκτορας δεν ήταν υποχρεωμένος να επισκεφτεί όλα τα μηχανήματα-στόχους. Η διεργασία ολοκληρωνόταν με την ανεύρεση των ζητούμενων δεδομένων. Επιπλέον, η συνάρτηση που καθόριζε την διαδρομή του πράκτορα ήταν στηριγμένη σε μια πιθανότητα ύπαρξης των προς αναζήτηση δεδομένων σε κάθε μηχανήμα.

Το πρόβλημα το οποίο προσπαθούμε να επιλύσουμε είναι περισσότερο προσανατολισμένο σε εφαρμογές διαχείρισης δικτύου. Σε αυτού του τύπου τις εφαρμογές, μια διεργασία πρέπει να εκτελεστεί σε όλους τους κόμβους του δικτύου. Άρα, η διεργασία σε αυτήν την περίπτωση ολοκληρώνεται μόνο μετά την επίσκεψη όλων των μηχανημάτων-στόχων. Το πρόβλημα λοιπόν ορίζεται ως εξής: “Δοθέντος ενός αριθμού μηχανημάτων-στόχων, ποια είναι η βέλτιστη διαδρομή την οποία πρέπει να ακολουθήσει ο πράκτορας, ώστε να επιστεφτεί όλους τους στόχους και να εκτελέσει μια συγκεκριμένη διεργασία σε αυτούς;” Η βασική διάφορα του προβλήματος αυτού με το TSP, είναι ότι στην περίπτωση του δεύτερου ο πωλητής επιστρέφει στο σημείο από το οποίο ξεκίνησε.

Η εύρεσης της βέλτιστης διαδρομής βασίζεται στην ελαχιστοποίηση του κόστους που εκφράζεται, από μαθηματική άποψη, από τη συνάρτηση [Kotsilieris02]:

$$E = \left\{ \sum_{i=1}^n c(s(i), a(i)) \right\} \quad (6.1)$$

όπου:

- $S(i)$ είναι η κατάσταση στον κόμβο i
- $a(i)$ είναι η ενέργεια που συμβαίνει στον κόμβο i .

- $c(s(i); a(i))$ καθορίζει το άμεσο κόστος για την ενέργεια που συμβαίνει στον κόμβο i

Τα πιθανά σύνολα καταστάσεων είναι οι συνθήκες του υποκείμενου δικτύου και ενδεχομένως οι θέσεις άλλων ΚΠ, εφόσον υπάρχει ανάγκη να επικοινωνήσουν ή να συνεργαστούν μεταξύ τους. Το σύνολο ενεργειών περιλαμβάνει τη μετανάστευση σε κάποιο συγκεκριμένο υπολογιστή (στοιχείο του δικτύου), καθώς επίσης και τις διάφορες ενέργειες ανίχνευσης του δικτύου, οι οποίες λαμβάνουν ενεργά τα στατιστικά στοιχεία του.

Το αποτέλεσμα του σχεδιασμού θα πρέπει να είναι μια ακολουθία της μορφής.
 $\langle a(1), a(2), \dots, a(n) \rangle$.

6.3 Βελτίωση της απόδοσης με αναζήτηση και ανάθεση της βέλτιστης διαδρομής

Το πρόβλημα βελτίωσης της απόδοσης μιας εφαρμογής ΚΠ, χωρίς αλλαγή στον σχεδιασμό, ανάγεται σε ένα πρόβλημα ανεύρεσης βέλτιστης διαδρομής. Η επίλυση ενός τέτοιου προβλήματος στηρίζεται σε μια σειρά από κόστη μετακίνησης. Για να αναπτυχθούν μηχανισμοί βελτίωσης της απόδοσης μιας εφαρμογής ΚΠ, πρέπει πρώτα να προσδιοριστούν επακριβώς οι παράμετροι που επηρεάζουν την απόδοση της. Οι παράμετροι αυτοί σχετίζονται με την μετανάστευση και το χρόνο εκτέλεσης του πράκτορα στο μηχανήμα-στόχο. Εφαρμογές που βασίζονται σε ΚΠ υπόκεινται σε συμφορήσεις απόδοσης εξαιτίας ανεπαρκών δικτυακών ή υπολογιστικών πόρων. Οι παράμετροι αυτοί συνοψίζονται ως εξής:

- Παράμετροι υποκείμενου δικτύου
- Παράμετροι του μηχανήματος που φιλοξενεί τον πράκτορα

Παράμετροι Υποκείμενου Δικτύου

Ο συνολικός χρόνος εκτέλεσης μιας εφαρμογής ΚΠ σχετίζεται στενά με το χρόνο μετανάστευσης των πρακτόρων. Ο χρόνος μετανάστευσης εξαρτάται από την κατάσταση του υποκείμενου δικτύου. Οι παράμετροι δικτύου που επηρεάζουν το χρόνο μετανάστευσης ενός πράκτορα είναι η καθυστέρηση δικτύου, το διαθέσιμο εύρος ζώνης και ο ρυθμός απώλειας πακέτων.

Παράμετροι μηχανήματος που φιλοξενεί τον πράκτορα

Οι ΚΠ είναι υπομονάδες λογισμικού οι οποίες παράγουν υψηλά φορτία στα μηχανήματα που εκτελούνται. Για λόγους μεταφερσιμότητας οι ΚΠ υλοποιούνται συνήθως σε Java, μια γλώσσα που εκτελείται σχετικά αργά. Το φόρτο εργασίας στο εξυπηρετούμενο μηχανήμα (το οποίο μερικές φορές παράγεται και από άλλους πράκτορες που φιλοξενούνται σε αυτό) επηρεάζει το χρόνο εκτέλεσης του πράκτορα. Στις περισσότερες περιπτώσεις, υπάρχει μόνο μια ενεργή ΠΚΠ σε κάθε μηχανήμα που εκτελείται πάνω από μια JVM, υπεύθυνη όχι μόνο για τη “φιλοξενία” των πρακτόρων, αλλά και για τη δημιουργία και την εκκίνηση νέων πρακτόρων. Είναι ξεκάθαρο ότι ο χρόνος εκτέλεσης ενός ΚΠ στο μηχανήμα-στόχο, εξαρτάται από το φόρτο εργασίας της Κεντρικής Μονάδας Επεξεργασίας ΚΜΕ (Central Process Unit-CPU) του μηχανήματος.

Μια άλλη παράμετρος που επηρεάζει την απόδοση του πράκτορα, η οποία όμως σχετίζεται άμεσα με τον σχεδιασμό του, είναι το μέγεθος του. Το μέγεθος του πράκτορα επηρεάζει την απόδοση, αφού όσο μεγαλύτερο είναι τόσο πιο δύσκολη γίνεται η μετανάστευση. Καθώς ο πράκτορας μεταναστεύει από το ένα μηχάνημα-στόχο στο άλλο μεταφέρει δεδομένα σχετικά με την κατάσταση του, που περιέχουν πληροφορίες από προηγούμενες εκτελέσεις της εργασίας. Καθώς μεγαλώνει ο αριθμός των επισκεφθέντων μηχανημάτων, αυξάνει και το μέγεθος του, κάνοντας την μετανάστευση δυσκολότερη. Μια πιθανή λύση σε αυτό το πρόβλημα είναι να προγραμματιστεί ο πράκτορας ώστε να επισκέπτεται έναν σταθερό αριθμό μηχανημάτων, διατηρώντας έτσι το μέγεθός του μικρό. Μια άλλη λύση είναι να προγραμματιστεί ώστε να στέλνει όλα τα δεδομένα που έχει συλλέξει στη βάση του και κατόπιν να συνεχίζει με τα εναπομείναντα μηχανήματα [Rubinstein99].

Οι παράμετροι λοιπόν που επηρεάζουν άμεσα την απόδοση μιας εφαρμογής ΚΠ είναι η κατάσταση του υποκείμενου δικτύου και ο φόρτος εργασίας του μηχανήματος-στόχου. Για να μπορέσουμε να διερευνήσουμε το κατά πόσο επηρεάζει την απόδοση κάθε μια από αυτές τις παραμέτρους χωριστά, αναπτύχθηκαν διαφορετικοί μηχανισμοί βελτίωσης της απόδοσης για κάθε παράμετρο.

6.3.1 Βελτίωση της απόδοσης βάσει του φόρτου εργασίας των μηχανημάτων-στόχων.

Οι ΚΠ δεν διαφέρουν σε τίποτα από ένα κοινό πρόγραμμα που εκτελείται σε έναν υπολογιστή, από την στιγμή που μεταναστεύσουν σε αυτόν. Όπως είναι γνωστό ένα μηχάνημα που εκτελεί μια σειρά προγραμμάτων παράλληλα, διαθέτει μόνο ένα μέρος της ΚΜΕ του σε κάθε ένα από αυτά. Είναι λοιπόν φυσικό ο φόρτος εργασίας της ΚΜΕ να επηρεάζει τον χρόνο εκτέλεσης ενός κινητού πράκτορα. Για να προσδιοριστεί ο βαθμός στον οποίο επηρεάζει την απόδοση ενός συστήματος ΚΠ ο φόρτος εργασίας την ΚΜΕ, αναπτύχθηκε ένας αλγόριθμος ανεύρεσης της βέλτιστης διαδρομής, με βάση τον φόρτο της ΚΜΕ των μηχανημάτων-στόχων.

Ο τρόπος με τον οποίο μεταβάλλεται ο φόρτος της ΚΜΕ ενός μηχανήματος είναι εντελώς τυχαίος. Ένα μηχάνημα μπορεί να βρίσκεται σε αδρανή κατάσταση (0% χρήση της ΚΜΕ) για μεγάλο χρονικό διάστημα και ξαφνικά να ξεκινήσει μια διεργασία που να

απαιτεί το 100% της ΚΜΕ, χωρίς να μπορεί να προσδιοριστεί ο χρόνος που θα διαρκέσει. Η αρχική μελέτη απέδειξε ότι ένας αλγόριθμος οποίος στηρίζεται μόνο στην τρέχουσα τιμή του φόρτου της ΚΜΕ, δεν είναι λειτουργικός. Η τυχαιότητα με την οποία μεταβάλλεται ο φόρτος εργασίας την ΚΜΕ δεν επιτρέπει τον ακριβή προσδιορισμό των περιόδων όπου αυτή βρίσκεται σε κατάσταση αδρανείας. Ακόμη και αν η εκτέλεση του πράκτορα ξεκινήσει μέσα σε μια τέτοια περίοδο, δεν υπάρχει τρόπος να αποκλειστεί η εκτέλεση μιας άλλης διεργασίας ταυτόχρονα με αυτόν. Έπρεπε λοιπόν να βρεθεί ένας τρόπος να αυξηθεί η πιθανότητα εκτέλεσης του πράκτορα σε ένα χρονικό διάστημα όπου η ΚΜΕ βρίσκεται σε κατάσταση αδράνειας. Ο προτεινόμενος αλγόριθμος στηρίζεται όχι μόνο στην τρέχουσα τιμή της ΚΜΕ αλλά και στις παρελθούσες τιμές αυτής. Η βασική ιδέα του αλγόριθμου είναι ότι σε ένα κατανεμημένο σύστημα υπολογιστών ένας υπολογιστής ο οποίος έχει παρουσιάσει πρόσφατα στο παρελθόν μεγάλο φόρτο εργασίας, παρουσιάζει αυξημένη πιθανότητα αν την τρέχουσα στιγμή βρίσκεται σε αδράνεια να την διατηρήσει για κάποιο χρονικό διάστημα, χωρίς αυτό να μπορεί να προσδιοριστεί.

Έστω n ο αριθμός των μηχανημάτων-στόχων, $Load_{current}(i)$ είναι η τρέχουσα τιμή του φόρτου εργασίας και $M(i)$ είναι ένας μετρητής όπου αυξάνει κατά μια μονάδα κάθε φορά που ο φόρτος εργασίας είναι μικρότερος από ένα κατώφλι (συγκεκριμένα 50%), όπου $i=1\dots n$, οι πιθανοί στόχοι. Ο αλγόριθμος που παρέχει το επόμενο μηχανήμα που θα επισκεφθεί ο πράκτορας, βάσει του φόρτου εργασίας των μηχανημάτων-στόχων παρουσιάζεται στο ακόλουθο ψευδοπρόγραμμα.

```
THRESHOLD=50;
```

```
 $M(i)=0$ ;
```

```
Int[] potentials = new int[n]
```

```
For ( $i=0$ ;  $i<n$ ;  $i++$ ) {
```

```
    If ( $Load_{current}(i) < THRESHOLD$ ) {
```

```
         $M(i) = M(i) + 1$ ;
```

```
        potentials[i] =  $M(i)$ 
```

```
    }
```

```
}
```

```
int min=0;
```

```
for (i=0;i<n;i++){
    if (potentials[i] != 0){
        min = i;
        break;
    }
}
for (i=min +1;i<n;i++){
    if ((potentials[i] < potentials[min]) && (potentials[i] != 0))
        min = i;
}
return min;
```

Ο παραπάνω αλγόριθμος είναι εμπνευσμένος από τον αλγόριθμο Least Recently Used (LRU), που χρησιμοποιείται για την αλλαγή σελίδων αποθηκευμένων στην μνήμη (paging).

6.3.1.1 Πειραματικό σύστημα διαχείρισης δικτύου

Για να ελεγχθεί η λειτουργικότητα του αλγόριθμου βελτίωσης της απόδοσης και κατ' επέκταση η επίδραση του φόρτου της ΚΜΕ σε αυτήν, υλοποιήθηκε ένα απλό σύστημα διαχείρισης δικτύου. Το σύστημα χρησιμοποιεί έναν ΚΠ για την εκτέλεση μιας σειράς SNMP εντολών, στους κόμβους του δικτύου. Η αρχιτεκτονική του συστήματος παρουσιάζεται στο σχήμα 6.1.

Το σύστημα αποτελείται από έναν κεντρικό υπολογιστή (το διαχειριστή) που δημιουργεί τον ΚΠ και εκτελεί τον αλγόριθμο βελτίωσης της απόδοσης και από Αντικείμενα Παρακολούθησης ΑΠ (monitoring Objects) του φόρτου της ΚΜΕ, που εκτελούνται στα μηχανήματα-στόχους. Στα μηχανήματα-στόχους εκτελείται επίσης ένας SNMP αντιπρόσωπος που παρέχει πρόσβαση στα δικτυακά δεδομένα που διατηρεί η διαχειριζόμενη συσκευή δικτύου. Οι ΚΠ της εφαρμογής είναι εφοδιασμένοι με την έκδοση 3.1 του AdventNet SNMP API [Adventnet] με το οποίο επικοινωνούν με τον SNMP αντιπρόσωπο, για να αντλήσουν τις απαραίτητες πληροφορίες.

Η ΠΚΠ που χρησιμοποιήθηκε για την υλοποίηση του συστήματος είναι η Voyager ORB της Object space. Η γλώσσα υλοποίησης της πλατφόρμας αυτής είναι η Java και κατά

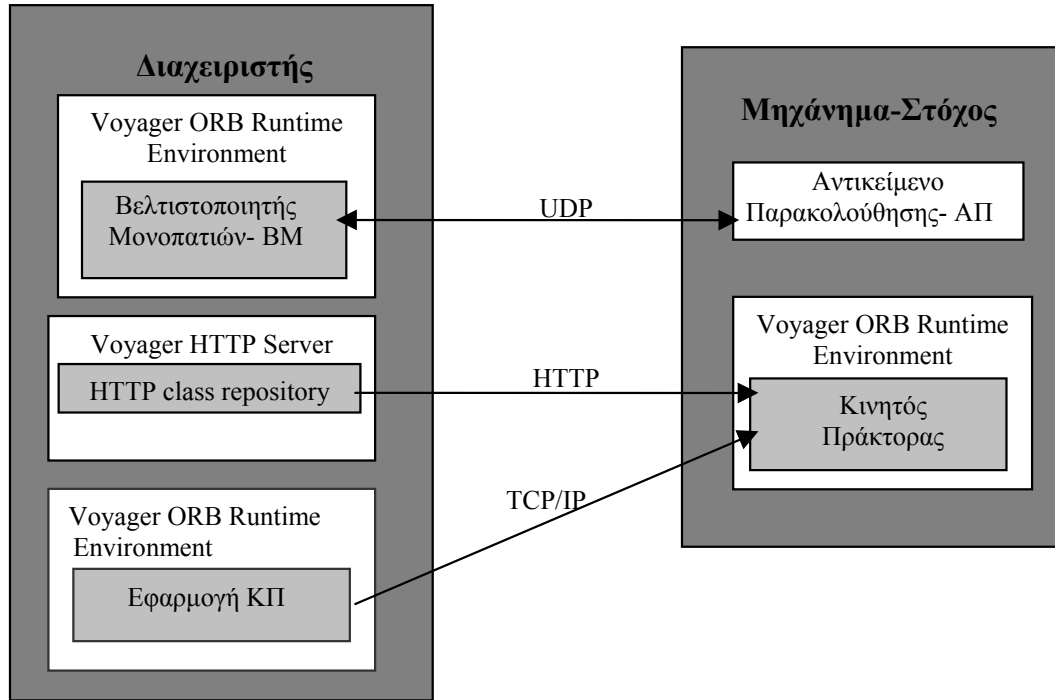
συνέπεια η γλώσσα αυτή χρησιμοποιήθηκε για την υλοποίηση όλων των υπο-οντοτήτων του συστήματος. Η ΠΚΠ εκτελείται σε όλα τα μηχανήματα του συστήματος και παρέχει το περιβάλλον εκτέλεσης των πρακτόρων. Σύμφωνα με την αρχιτεκτονική της πλατφόρμας αυτής, οι Java κλάσεις των ΚΠ διατηρούνται σε μια αποθήκη κλάσεων (class repository). Τα μηχανήματα-στόχοι έχουν πρόσβαση στις κλάσεις αυτές μέσω ενός HTTP εξυπηρετητή, που παρέχεται από την πλατφόρμα (Voyager HTTP server). Στην συγκεκριμένη εφαρμογή ο εξυπηρετητής αυτός εκτελείται στο κεντρικό μηχάνημα, στο οποίο βρίσκεται και η εφαρμογή διαχείρισης δικτύου.

Στο κεντρικό μηχάνημα εκτελούνται οι παρακάτω υπο-οντότητες:

- Το Voyager ORB runtime environment
- Ο HTTP Voyager Εξυπηρετητής
- Ο Βελτιστοποιητής Μονοπατιών BM
- η Εφαρμογή Διαχείρισης Δικτύων

Σε κάθε μηχανήμα-στόχο εκτελούνται οι ακόλουθες υπο-οντότητες:

- Το Voyager ORB runtime environment
- Το Αντικείμενο Παρακολούθησης
- Ο κινητός αντιπρόσωπος της Εφαρμογής Διαχείρισης Δικτύων



Σχήμα 6.1: Αρχιτεκτονική του συστήματος βελτίωσης της απόδοσης με παρακολούθηση του φόρτου της ΚΜΕ

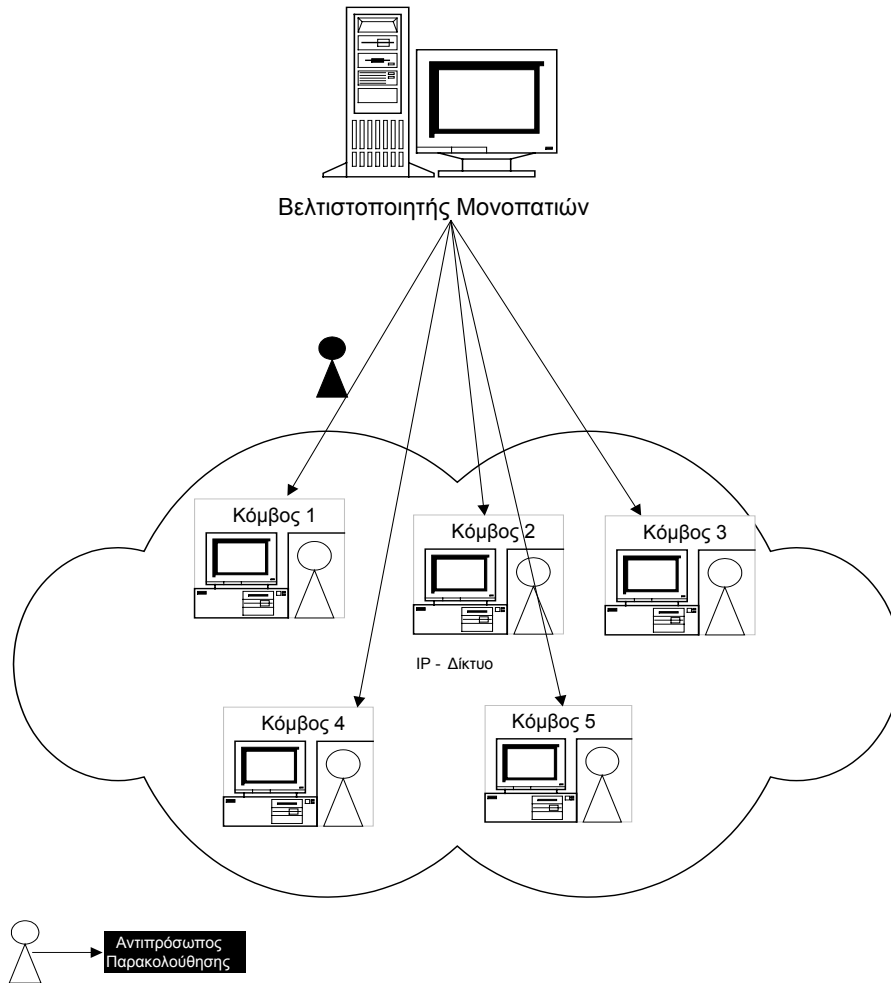
Ο BM υλοποιεί τον αλγόριθμο βελτιστοποίησης. Δέχεται αιτήσεις βελτιστοποίησης από την εφαρμογή ΚΠ. Οι αιτήσεις αυτές περιέχουν μια σειρά από μηχανήματα-στόχους. Ο BM επικοινωνεί με τα ΑΠ που βρίσκονται στα μηχανήματα-στόχους και λαμβάνει τις τρέχουσες τιμές του φόρτου της ΚΜΕ. Η συχνότητα δειγματοληψίας των τιμών είναι 1 sec. Η επικοινωνία μεταξύ του BM και των μηχανημάτων-στόχων πραγματοποιείται με χρήση του πρωτοκόλλου UDP. Ο BM υπολογίζει την βέλτιστη διαδρομή και παραδίδει αυτήν την πληροφορία στην εφαρμογή ΚΠ, η οποία εν συνεχεία δημιουργεί τον ΚΠ. Φυσικά, η κατάσταση της ΚΜΕ των μηχανημάτων-στόχων δεν παραμένει σταθερή καθόλη την διάρκεια του ταξιδιού του πράκτορα. Ο BM υπολογίζει διαρκώς την βέλτιστη διαδρομή με βάση τις μεταβαλλόμενες τιμές της ΚΜΕ των στόχων. Η πληροφορία αυτή μεταδίδεται σε όλα τα ΑΠ. Με αυτόν τον τρόπο, ο πράκτορας δεν χρειάζεται να επικοινωνεί με τον διαχειριστή για να πληροφορηθεί για τον επόμενο στόχο. Η πληροφορία αυτή παρέχεται από το ΑΠ του κόμβου στον οποίο βρίσκεται. Τα ΑΠ λοιπόν, εκτός από το να παρέχουν την τρέχουσα τιμή του φόρτου της ΚΜΕ στον BM, είναι υπεύθυνα να πληροφορούν τον ΚΠ για τον επόμενο στόχο που πρέπει να

επισκεφθεί. Έτσι, επιτυγχάνεται η γρήγορη πληροφόρηση του ΚΠ για τον επόμενο στόχο του. Η συνεχής ανταλλαγή μηνυμάτων και η εκτέλεση του αλγορίθμου έχει το μειονέκτημα της δημιουργίας μιας επιπλέον κίνησης δεδομένων στο υπό διαχείριση δίκτυο και την αύξηση του φόρτου εργασίας του μηχανήματος που εκτελεί την ΒΜ. Μια εναλλακτική προσέγγιση είναι να θεωρηθεί ότι ο φόρτος εργασίας της ΚΜΕ των στόχων παραμένει σταθερός, καθόλη την διάρκεια του ταξιδιού του πράκτορα. Με βάση αυτήν την θεώρηση ο υπολογισμός της βέλτιστης διαδρομής πραγματοποιείται μόνο μια φορά (στην αρχή). Η εφαρμογή παρέχει αυτόν τον απλούστερο τρόπο λειτουργίας, η μελέτη του οποίου όμως απέδειξε ότι δεν επιφέρει σημαντική βελτίωση στην απόδοση.

Η παρακολούθηση των πραγματικών τιμών του φόρτου της ΚΜΕ από τα ΑΠ με χρήση της γλώσσας προγραμματισμού Java δεν είναι κάτι απλό. Για την απόκτηση του τοπικού φόρτου εργασίας της ΚΜΕ, χρησιμοποιήθηκε η διεπαφή JNI (Java Native Interface) [JNI] που επιτρέπει την κλήση προγραμμάτων υλοποιημένων στην γλώσσα προγραμματισμού C από προγράμματα της Java. Οι τιμές της τοπικής ΚΜΕ λαμβάνονται με την χρήση ενός προγράμματος υλοποιημένο σε C, το οποίο συνεργάζεται με το ΑΠ, το οποίο είναι υλοποιημένο σε Java, με χρήση της JNI.

6.3.1.2 Πειραματικά δεδομένα

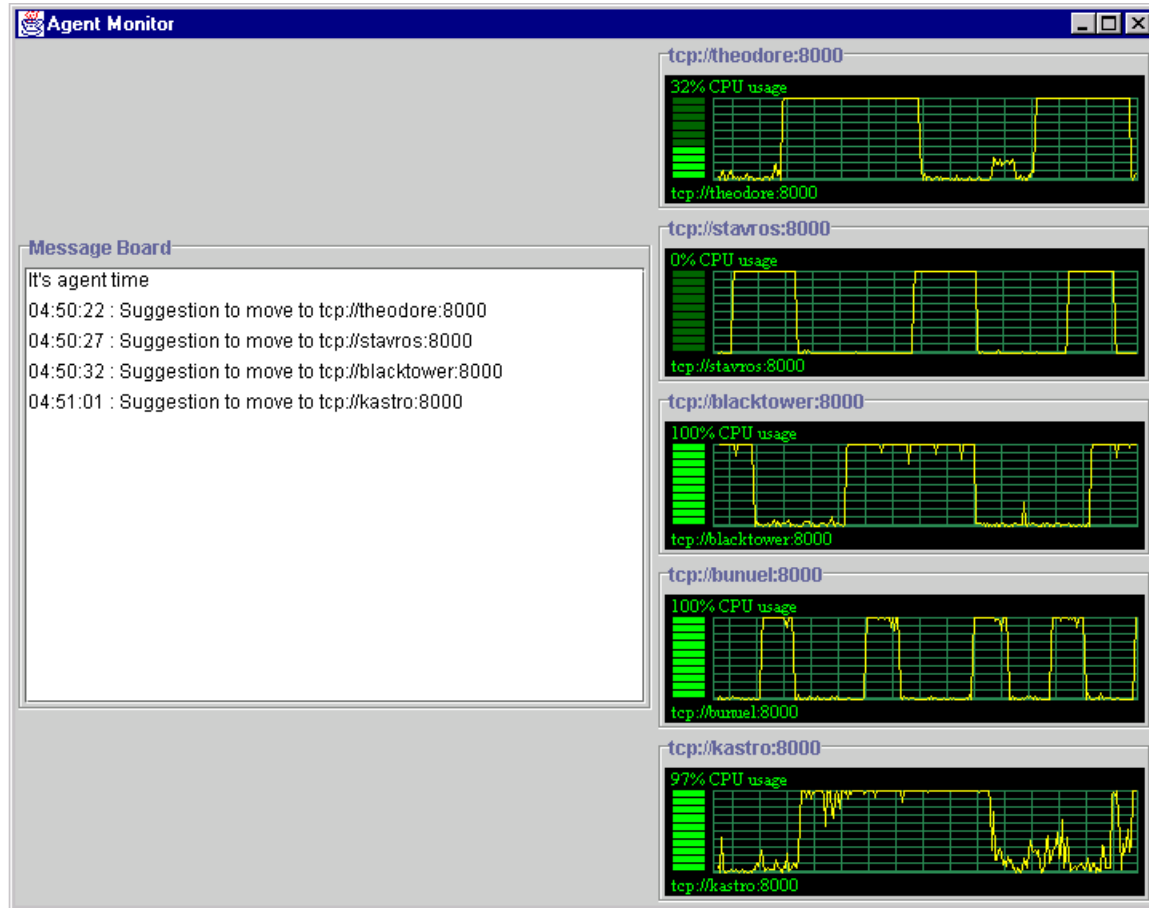
Για να αξιολογηθεί η βελτίωση που επιφέρει η χρήση του αλγόριθμου, χρησιμοποιήθηκε το πειραματικό IP δίκτυο που παρουσιάζεται στο σχήμα 6.2. Το δίκτυο αποτελείται από πέντε μηχανήματα-στόχους στα οποία πραγματοποιούνται μια σειρά από SNMP αιτήσεις και ένα κεντρικό μηχανήμα το οποίο εκτελείται η εφαρμογή διαχείρισης δικτύου με τον αλγόριθμο βελτίωσης. Η διάρθρωση του πειραματικού δικτύου είναι η ακόλουθη: τρεις υπολογιστές με ΚΜΕ 700MHz και μνήμη 256 MB RAM και τρεις υπολογιστές με ΚΜΕ 300MHz και 256 MB RAM, συνδεδεμένους σε ένα τοπικό Ethernet LAN 10Mbps. Το λειτουργικό σύστημα όλων τα μηχανημάτων, είναι το Linux Redhat v6.2 [Redhat].



Σχήμα 6.2: Πειραματικό IP δίκτυο με πέντε μηχανήματα-στόχους

Για να είναι δυνατή η εξαγωγή συμπερασμάτων όσων, αφορά την αποτελεσματικότητα του αλγόριθμου, τα μηχανήματα-στόχοι έπρεπε να παρουσιάζουν κάποια μεταβολή του φόρτου εργασίας της ΚΜΕ. Για αυτό τον λόγο, δημιουργήθηκε ένα πρόγραμμα Java η εκτέλεση του οποίου έχει ως αποτέλεσμα, οι αδρανείς καταστάσεις των ΚΜΕ των μηχανών προορισμού να ακολουθούν την κανονική, ομοιόμορφη και την εκθετική κατανομή αντίστοιχα, με μέση τιμή 50 ms και μέγιστη 100 ms. Στη μη αδρανή κατάσταση εκτελείται μια διεργασία η οποία καταλαμβάνει το 100% της ΚΜΕ του κάθε μηχανήματος. Σε όλες τις περιπτώσεις η διάρκεια για την οποία το φορτίο της ΚΜΕ ξεπερνά το 50% είναι 10 ms. Το γραφικό περιβάλλον της εφαρμογής, όπου

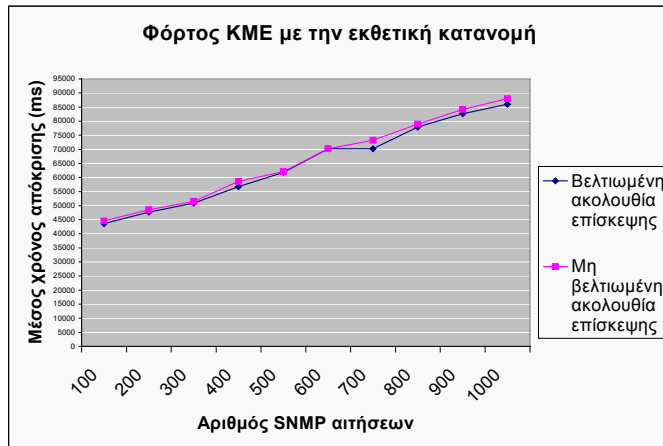
παρουσιάζονται οι τιμές των ΚΜΕ των υπό-παρακολούθηση στόχων (αριστερά) και η δρομολόγηση με βάση τον αλγόριθμο (δεξιά), παρουσιάζεται στο σχήμα 6.3.



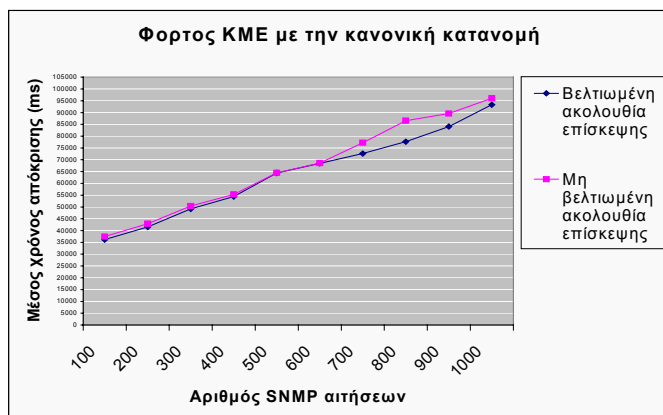
Σχήμα 6.3: Το γραφικό περιβάλλον της εφαρμογής διαχείρισης δικτύου

Για κάθε είδος προσομοίωσης του φορτίου της ΚΜΕ (κανονική, ομοιόμορφη και εκθετική κατανομή) εκτελέστηκαν 100 πειράματα με τον ΚΠ να εκτελεί ένα μεταβαλλόμενο αριθμό SNMP αιτήσεων, στα μηχανήματα-στόχους. Ο αριθμός των SNMP αιτήσεων μεταβαλλόταν από 100 έως 1000, με ένα διάστημα κλιμακούμενο κατά 100. Όλα τα πειράματα εκτελέστηκαν τόσο με χρήση όσο και χωρίς χρήση του αλγόριθμου βελτίωσης, ώστε να είναι δυνατή η σύγκριση. Σε όλα τα πειράματα θεωρήσαμε ότι οι δικτυακές παράμετροι (καθυστέρηση μετάδοσης, απόκλιση καθυστέρησης και απώλειες) διατηρούνται σταθερές κατά μήκος του δικτύου.

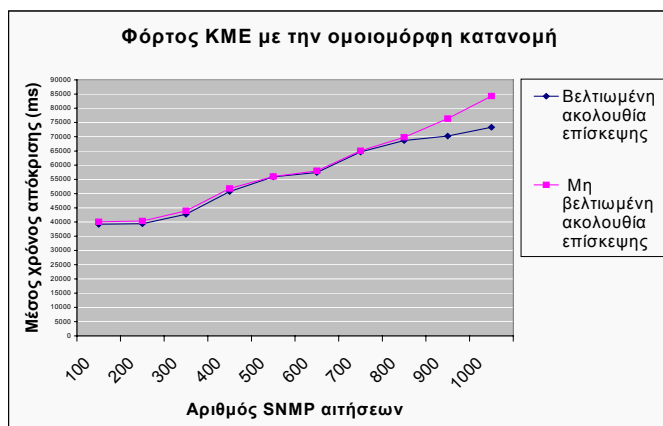
Στη συνέχεια παρουσιάζονται τα διαγράμματα και οι πίνακες με τους μέσους χρόνους απόκρισης των πειραμάτων, για κάθε κατανομή φορτίου της ΚΜΕ, με και χωρίς βελτίωση της ακολουθίας επίσκεψης του ΚΠ.



Αιτήσεις	Χρόνος απόκρισης με Βελτίωση (ms)	Χρόνος απόκρισης χωρίς βελτίωση (ms)
100	36194.22	37483.14
200	41593.05	42916.81
300	49217	50405.22
400	54407.19	55310.84
500	64391.84	64513.68
600	68506	68590.27
700	72638.38	77268.92
800	77665.81	86601.7
900	84081.95	89598
1000	93338.54	96080.19



Αιτήσεις	Χρόνος απόκρισης με βελτίωση (ms)	Χρόνος απόκρισης χωρίς βελτίωση (ms)
100	43598.24	44557.13
200	47716.26	48568.34
300	50861.84	51493.24
400	56735.76	58582.34
500	61769.34	62067.08
600	70182.29	70200.16
700	70218.39	73202.08
800	77965.79	78949.74
900	82616.18	84178.74
1000	86009.82	87972.29



Αιτήσεις	Χρόνος απόκρισης με βελτίωση (ms)	Χρόνος απόκρισης χωρίς βελτίωση (ms)
100	39201	40089.45
200	39417.45	40357.09
300	42745.09	43930.36
400	50718.36	51777.27
500	55892.36	55995.64
600	57423.45	57988.91
700	64648.36	65083.09
800	68673.18	69778.82
900	70250.82	76340.64
1000	73370	84288.55

Το αρχικό συμπέρασμα που μπορεί να εξαχθεί με την παρατήρηση των γραφημάτων, είναι ότι η χρήση του αλγόριθμου βελτίωσης με βάση το φορτίο της ΚΜΕ, βελτιώνει την απόδοση της εφαρμογής διαχείρισης δικτύων. Επίσης οι κλίσεις των καμπυλών και στις τρεις περιπτώσεις προσομοίωσης ΚΜΕ έχουν μια συγκρίσιμη τιμή. Ένα άλλο σημείο που αξίζει για να αναφερθεί, είναι ότι η βελτίωση απόδοσης συσχετίζεται στενά με τον αριθμό των SNMP αιτήσεων, που ο ΚΠ εκτελεί σε κάθε απομακρυσμένο κόμβο και το χρόνο που η ΚΜΕ του απομακρυσμένου κόμβου είναι αδρανής. Βλέπουμε ότι για τις μικρές και μεγάλες διεργασίες (από 100 έως 400 και από 700 έως 1000 αιτήσεις) η απόδοση είναι καλύτερη απ' ότι στις μέσου μεγέθους διεργασίες (από 500 έως 600 αιτήσεις). Επίσης εξάγεται το συμπέρασμα ότι υπάρχει βελτίωση περίπου 6% για τις μικρές διεργασίες και περίπου 9% για τις μεγάλες. Αυτό μπορεί να εξηγηθεί ως εξής. Από τα παραπάνω διαγράμματα και πίνακες μπορούμε εύκολα να συμπεράνουμε ότι η αρχικοποίηση ενός κινητού αντιπροσώπου σε έναν κόμβο SNMP είναι κατά μέσο όρο 31000 ms. Επίσης η εκτέλεση μιας στοιχειώδους διεργασίας, που αποτελείται από 100 SNMP αιτήσεις, όταν η ΚΜΕ βρίσκεται σε κατάσταση αδράνειας είναι 5400 ms. Κατά συνέπεια οι μικρές στοιχειώδεις εργασίες μπορούν να διεκπεραιωθούν μέσα στο χρονικό διάστημα στο οποίο η ΚΜΕ είναι αδρανής. Οι μέσου μεγέθους διεργασίες χρειάζονται συνήθως για την εκτέλεσή τους ένα πλήρη κύκλο του φορτίου της ΚΜΕ (μια αδρανής ημιπερίοδος και μια φορτωμένη) κι έτσι η βελτιστοποίηση της ακολουθίας επίσκεψης δεν βοηθά πολύ σε αυτήν την περίπτωση. Τελικά, οι μεγάλες διεργασίες εκτελούν κατά τη διάρκεια μιας αδρανούς ημιπεριόδου της ΚΜΕ, μιας φορτωμένης και τελικά άλλης μια αδρανούς.

6.3.2 Βελτίωση της απόδοσης βάσει της καθυστέρησης μετάδοσης του δικτύου

Στην προηγούμενη ενότητα παρουσιάστηκε και εφαρμόστηκε ένας αλγόριθμος ανεύρεσης της βέλτιστης διαδρομής, που πρέπει να ακολουθήσει ένα πράκτορας, με βάση τον φόρτο εργασίας της ΚΜΕ του μηχανήματος-στόχου. Μια άλλη παράμετρος που επηρεάζει την απόδοση ενός ΚΠ, κατά την διάρκεια του ταξιδιού του, είναι η κατάσταση του υποκείμενου δικτύου.

Σε αυτήν την ενότητα θα παρουσιαστεί ένας αλγόριθμος ανεύρεσης της βέλτιστης διαδρομής, με βάση την καθυστέρηση μετάδοσης του δίκτυο. Παρόλο που η κατάσταση

των συνδέσεων ενός δίκτυο χαρακτηρίζεται από τρεις παραμέτρους (καθυστέρηση, απόκλιση καθυστέρησης και απώλειες), ο αλγόριθμος στηρίζεται μόνο σε μια από αυτές. Η παρακολούθηση όλων των παραμέτρων ενός δικτύου και η ανάπτυξη ενός αλγόριθμου που θα υπολόγιζε την βέλτιστη διαδρομή βάσει όλων αυτών, θα οδηγούσε σε μια πολύ χρονοβόρα διαδικασία με αντίθετα από τα επιθυμητά αποτελέσματα. Έτσι, επιλέχθηκε ως παράμετρος ανεύρεσης της βέλτιστης διαδρομής η καθυστέρηση μετάδοσης, η οποία όπως θα αποδειχθεί στην συνέχεια επηρεάζει άμεσα τον χρόνο μετανάστευσης ενός πράκτορα.

Σε αντίθεση με τον την περίπτωση ανεύρεσης βέλτιστης διαδρομής, με βάση τον φόρτο εργασία της ΚΜΕ, το πρόβλημα τώρα μπορεί να αντιμετωπιστεί ως ένα πρόβλημα γράφου με βάρη, ή πιο σωστά ως μια παραλλαγή του TSP. Η ανάγκη δημιουργίας ενός αλγόριθμου βελτιστοποίησης που θα ελάμβανε υπόψη του και τις παρελθούσες τιμές της ΚΜΕ δεν επέτρεπε την αντιμετώπιση του προβλήματος ως ένα πρόβλημα γράφου με βάρη.

Πριν παρουσιαστεί ο αλγόριθμος ανεύρεσης βέλτιστης διαδρομής, με βάση την καθυστέρηση μετάδοσης, πρέπει να δοθούν οι απαραίτητες πληροφορίες για την θεωρία γραφών με βάρη.

6.3.2.1 Βασικά της θεωρίας γραφών με βάρη

Σε ένα γράφο με βάρη κάθε ακμή, που συνδέει δυο κόμβους του γράφου, σχετίζεται με ένα βάρος ή κόστος μετακίνησης. Στην περίπτωση του προβλήματος που μελετάται οι κόμβοι αντιπροσωπεύονται από τα μηχανήματα-στόχους, ενώ τα βάρη των ακμών είναι η καθυστέρηση μετάδοσης των συνδέσεων. Τα δυο προβλήματα τα οποία συνήθως απαντώνται στους γράφους με βάρη[Sedgewick84] είναι τα εξής:

- Εύρεση ενός τρόπου σύνδεσης όλων των κόμβων του γράφου με το ελάχιστο κόστος.
- Εύρεση του μονοπατιού με το ελάχιστο κόστος μεταξύ δυο κόμβων του γράφου.

Το πρώτο ονομάζεται πρόβλημα εύρεσης **ελάχιστου εκτεινόμενου δέντρου (Minimum Spanning Tree-MST)** και το δεύτερο ονομάζεται πρόβλημα **εύρεσης μικρότερου**

μονοπατιού (shortest path). Ένα άλλο πρόβλημα που επίσης σχετίζεται με τους γράφους με βάρη είναι το γνωστό **TSP**.

6.3.2.1.1 Ελάχιστο εκτεινόμενο δέντρο

Ένα *ελάχιστο εκτεινόμενο δέντρο*, ενός γράφου με βάρη, είναι ένα σύνολο από ακμές που ενώνουν όλους τους βρόχους, με τέτοιο τρόπο ώστε το άθροισμα των βαρών να είναι τουλάχιστον τόσο μικρό όσο το άθροισμα των βαρών κάθε αλλού συνόλου ακμών που συνδέουν όλους του βρόχους. Υπάρχουν περισσότερα από ένα MST σε ένα γράφο.

Ο Prim το 1957 [Prim57] παρουσίασε ένα αλγόριθμο για την εύρεση του MST, ενός γράφου με βάρη. Η βασική ιδέα στην οποία στηρίζεται ο αλγόριθμος του Prim είναι η δημιουργία ενός MST ξεκινώντας από έναν αυθαίρετο κόμβο, προσθέτοντας συνεχώς ακμές ελάχιστου κόστους μέχρι να συνδεθούν όλοι οι κόμβοι του γράφου. Συγκεκριμένα ο τρόπος λειτουργίας του αλγόριθμου μπορεί να περιγραφεί ως εξής:

1. Έστω U το σύνολο των κόμβων που ανήκουν σε ένα MST T . Το U αρχικοποιείται ως σύνολο $\{1\}$. Έστω $V-U$ ένα άλλο σύνολο κόμβων που δεν ανήκουν στο MST το οποίο αρχικοποιείται ως $\{2,3,4,\dots,n\}$. Το T αρχικοποιείται ως ένα άδειο σύνολο ακμών $\{\}$.
2. Έστω e η ακμή με το με το ελάχιστο κόστος που συνδέει έναν κόμβο του U με έναν κόμβο του $V-U$.
3. Η ακμή e προστίθεται στο T .
4. Έστω w ο ακραίος κόμβος του $V-U$ της ακμής e .
5. Ο κόμβος w προστίθεται στο U .
6. Ο κόμβος w αφαιρείται από το $V-U$.
7. Αν το σύνολο $V-U$ δεν είναι άδειο η διαδικασία επαναλαμβάνεται από το βήμα 1 αλλιώς το ζητούμενο MST είναι το T .

Απόδειξη :

Έστω ότι το T είναι ένα MST ενός γράφου G και ας υποθέσουμε ότι υπάρχει ένα άλλο MST S στον γράφο, τέτοιο ώστε το άθροισμα των βαρών του S $W(S)$ να είναι μικρότερο από αυτό του T $W(T)$ [$W(S) < W(T)$]. Θα πρέπει να υπάρχει μια ακμή ελάχιστου κόστους e στο T , η οποία δεν υπάρχει στο S . Προστίθεται η ακμή e στο S . Έτσι, δημιουργείται ένας κύκλος C που περιέχει την ακμή e . Ο C περιέχει και μια ακμή e' η οποία δεν

περιέχεται στο T , διαφορετικά το T θα περιείχε επίσης έναν κύκλο πράγμα που εξ ορισμού του MST δεν ισχύει. Αν αντικατασταθεί η ακμή e' στο S με την ακμή e προκύπτει ένα εκτεινόμενο δέντρο S' στο οποίο το κόστος e είναι μικρότερο ή ίσο από το κόστος e' . Οι ακμές e και e' πρέπει να συμπίπτουν σε ένα κοινό κόμβο. Σύμφωνα με τον αλγόριθμο, αν το κόστος της ακμής e' ήταν μικρότερο από αυτό της e θα είχε προτιμηθεί αυτή στην δημιουργία του MST T . Το S' είναι τώρα μια ακμή κοντύτερα στο T από ότι το S και ισχύει $W(S') \leq W(S)$. Επαναλαμβάνοντας την προηγούμενη αντικατάσταση προκύπτει ότι $S'=T$ και $W(T) \leq W(S)$, συμπέρασμα που είναι αντίθετο με την αρχική θεώρηση ($W(S) < W(T)$). Άρα, ο αλγόριθμος του Prim παρέχει ένα MST του γράφου G .

6.3.2.1.2 Μικρότερο μονοπάτι

Το πρόβλημα *μικρότερου μονοπατιού* ορίζεται ως η εύρεση του μονοπατιού, σε έναν γράφο με βάρη, που συνδέει δυο κόμβους x και y με το μικρότερο άθροισμα βαρών ακμών από όλα τα αλλά μονοπάτια αυτού του τύπου. Ένας καλός αλγόριθμος για την εύρεση του μικρότερου μονοπατιού είναι αυτός του Dijkstra [Dijkstra59]. Με την ίδια ευκολία που μπορεί να βρεθεί το μικρότερο μονοπάτι από έναν κόμβο σε έναν άλλο, μπορούν να βρεθούν τα μικρότερα μονοπάτια από έναν κόμβο προς όλους τους άλλους κόμβους του γράφου. Για τον λόγο αυτό, το πρόβλημα μερικές φορές αναφέρεται ως πρόβλημα εύρεση μικρότερων μονοπατιών μοναδικής-πηγής (single-source shortest paths).

Έστω ένας γράφος $G=(V,E)$ όπου V είναι το σύνολο των κόμβων και E είναι το σύνολο των ακμών.

Ο αλγόριθμος του Dijkstra χρησιμοποιεί δυο σύνολα από κόμβους:

- Το S είναι το σύνολο των κόμβων των οποίων το μικτότερο μονοπάτι από την πηγή έχει ήδη καθοριστεί.
- Το $S-V$ είναι το σύνολο των εναπομεινάντων κόμβων.

Επίσης διατηρεί τα:

- d είναι ο πίνακας με την εκτίμηση μικρότερου μονοπατιού προς κάθε κόμβο.
- π ένας πίνακας με τους προκατόχους κάθε κόμβου.

Ο βασικός τρόπος λειτουργίας του αλγόριθμου είναι ο εξής:

- Αρχικοποιούνται τα d και π για κάθε κόμβο j του V :
 - $d_j = \infty$
 - $\pi_j = \text{nil}$
- Το S ορίζεται ως ένα άδειο σύνολο.
- Όσο υπάρχουν ακόμα κόμβοι στο $V-S$:
 - i. Οι κόμβοι του $V-S$ ταξινομούνται με βάση την τρέχουσα καλύτερη εκτίμηση της απόστασης του από την πηγή.
 - ii. Ο κοντινότερος κόμβος του $V-S$ ο u τοποθετείται στο S .
 - iii. Οι εναπομείναντες κόμβοι του $V-S$ υπόκεινται σε μια διαδικασία χαλάρωσης (relaxation).

Η διαδικασία χαλάρωσης ενημερώνει τα βάρη όλων των κόμβων v που συνδέονται με ένα κόμβο u , έτσι ώστε κάθε κόμβος να μην έχει κανένα προκάτοχο ($\pi[v] = \text{nil}$) και οι εκτιμήσεις κόστους κάθε κόμβου από την πηγή ($d[v]$) να είναι ίσες με το άπειρο, εκτός από του κόμβου πηγή του οποίου η εκτίμηση κόστους είναι ίση με μηδέν ($d[s] = 0$).

Απόδειξη

Λήμμα 1: Τα μικτότερα μονοπάτια αποτελούνται από μικτότερα μονοπάτια.

Απόδειξη: αν υπήρχε ένα μικτότερο μονοπάτι από οποιοδήποτε υπό-μονοπάτι τότε αντικατάσταση αυτού του μονοπατιού θα έκανε όλα το μονοπάτι μικτότερο.

Λήμμα 2: αν $s \rightarrow \dots u \rightarrow v$ είναι το μικτότερο μονοπάτι από το s στο v , τότε μετά την προσθήκη του u στο S και την χαλάρωση (u, v, w) , $d[v] = \delta(s, v)$ και το $d[v]$ δεν μπορεί να αλλάξει μετέπειτα.

Απόδειξη: Κάθε στιγμή $d[v] \geq \delta(s, v)$ αν $d[v] = \delta(s, v)$, τότε το $d[v]$ δεν θα αλλαχθεί ποτέ από την υπό-ρουτίνα χαλάρωσης.

Χρησιμοποιώντας το λήμμα 2:

Έστω ότι ο u είναι ο πρώτος κόμβος που προστίθεται στο S για τον οποίο ισχύει $d[u] \neq \delta(s, u)$.

Ο u δεν μπορεί να είναι ο s γιατί $d[s] = 0$. Πρέπει να υπάρχει ένα μονοπάτι από τον s στον u , αλλιώς το $d[u]$ θα είναι ίσο με το άπειρο. Αφού υπάρχει μονοπάτι θα υπάρχει και

μικτότερο μονοπάτι. Όταν ο κόμβος x προστέθηκε στο S το $d[x]=\delta(s,x)$, αφού υποθέσαμε ότι ο u ήταν ο πρώτος κόμβος για τον οποίο δεν ισχύει αυτή η ισότητα. Η ακμή $x \rightarrow y$ έχει ήδη υποστεί την διαδικασία χαλάρωσης άρα $d[y]=\delta(s,y) \leq \delta(s,u) \leq d[u]$. Όμως όταν επιλέχθηκε το u τόσο αυτός ο κόμβος αυτός όσο και ο y άνηκαν στο $S-V$ άρα $d[u] \leq d[y]$, διαφορετικά θα είχε επιλέγει το y . Άρα οι παραπάνω ανισότητες πρέπει να είναι ισότητες $d[y]=\delta(s,y)=\delta(s,u)=d[u]$, αυτό έρχεται σε αντίθεση με την αρχική υπόθεση $d[u] \neq \delta(s,u)$. Έτσι, δια της εις άτοπον απαγωγής αποδεικνύεται ότι ο αλγόριθμος του Dijkstra παρέχει το μικτότερο μονοπάτι.

6.3.2.1.3 Το πρόβλημα περιοδεύοντος πωλητή (Traveling Salesman Problem-TSP)

Το TSP ορίζεται ως η εύρεση ενός μονοπατιού σε ένα γράφο με βάρη που ξεκινά και τελειώνει στον ίδιο κόμβο, περιλαμβάνει κάθε άλλο κόμβο μόνο μια φορά, και ελαχιστοποιεί το συνολικό κόστος των ακμών. Η έρευνα πάνω στο θέμα ξεκίνησε ουσιαστικά το 1856 με τη διατύπωση του αλγορίθμου Hamilton το 1856. Από τότε το θέμα έχει ερευνηθεί διεξοδικά, καθώς το TSP βρίσκει εφαρμογή σε πλειάδα περιπτώσεων της καθημερινότητας.

Όπως έχει ήδη αναφερθεί, η διάφορα του προβλήματος αυτού με το πρόβλημα το οποίο μελετάται, είναι ότι στην περίπτωση του TSP ο πωλητής επιστέφει στο σημείο από το οποίο ξεκίνησε. Ο αλγόριθμος εύρεσης βέλτιστης διαδρομής με βάση την καθυστέρηση μετάδοσης του δικτύου, που παρουσιάζεται στην επόμενη παράγραφο, αποτελεί μια προσέγγιση του TSP.

6.3.2.2 Αλγόριθμος εύρεσης της βέλτιστης διαδρομής με βάση την καθυστέρηση μετάδοσης του δικτύου

Ο προτεινόμενος αλγόριθμος ανεύρεσης της βέλτιστης διαδρομής είναι μια προσέγγιση του TSP που εγγυάται την εύρεση ενός μονοπατιού, ο χρόνος ταξιδιού του οποίου θα είναι το πολύ διπλάσιος από το χρόνο ταξιδιού του βέλτιστου μονοπατιού. Η είσοδος για τον αλγόριθμο αποτελείται από το σύνολο των προς επίσκεψη προορισμών, την καθυστέρηση μεταξύ των συνδέσεων και το σημείο εκκίνησης του ΚΠ. Τα βήματα του αλγορίθμου είναι τα εξής [Kalogeropoulos03]:

1. Εισαγωγή ενός πλήρως συνδεδεμένου γράφου με βάρη, όπου τα βάρη είναι οι τιμές της καθυστέρησης των συνδέσεων.
2. Υπολογισμός ενός MST χρησιμοποιώντας τον αλγόριθμο του Prim.
3. Υπολογισμός μια διάβασης από την αφετηρία προς όλους του άλλους κόμβους του MST (η τελευταία επιστροφή στην αφετηρία αγνοείται).
4. Διαγραφή των επαναλαμβανόμενων κόμβων (π.χ. εάν ο κατάλογος είναι 1,3,4,3,7,5,7,2 διαγράφοντας του επαναλαμβανόμενους κόμβους προκύπτει ο 1,3,4,7,5,2).
4. Χρησιμοποιώντας τον αλγόριθμο εύρεσης μικρότερου μονοπατιού του Dijkstra (για όλα τα ζεύγη) επιλέγεται το μικρότερο μονοπάτι για να διαβούμε την διαδρομή που προέκυψε από το βήμα 4 (έτσι για το παραπάνω παράδειγμα θα χρησιμοποιούσαμε το μικρότερο μονοπάτι μεταξύ των ζευγαριών $\{1,3\}$ $\{3,4\}$ $\{4,7\}$ $\{7,5\}$ $\{5,2\}$).

Αν οι τιμές των βαρών είναι θετικές και συμμετρικές $c_{ij} = c_{ji}$ και για κάθε τριάδα κόμβων (i,j,k) ισχύει η τριγωνική ανισότητα $c_{ik} \leq c_{ij} + c_{jk}$, τότε η διαδρομή που υπολογίζεται από τον παραπάνω αλγόριθμο είναι το πολύ διπλάσια από την βέλτιστη διαδρομή.

Απόδειξη:

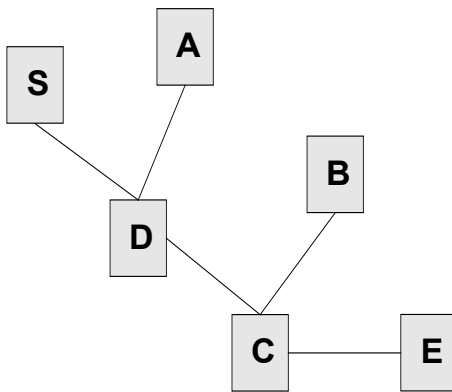
Έστω T το MST του γράφου. Το κόστος του T είναι μικρότερο ή ίσο από την βέλτιστη ζητούμενη διαδρομή. Ο υπολογισμός μιας διάβασης προς όλους τους κόμβους του MST έχει ως αποτέλεσμα την δημιουργία ενός κύκλου του Euler, έστω P . Το κόστος του P είναι διπλάσιο από αυτό του T . Στο 4 βήμα του αλγόριθμου αποκόπτονται οι επαναλαμβανόμενοι κόμβοι. Έτσι, αν ο κύκλος P έχει μια ακολουθία την μορφής $i, l, \dots, j, i, k$ τότε μετά το 4 βήμα προκύπτει μια ακολουθία της μορφής $i, l, \dots, j, k$. Έτσι λοιπόν, το κόστος μετάβασης από τον κόμβο j στον κόμβο k πριν την αποκοπή του κόμβου i ήταν $c_{ji} + c_{ik}$ ενώ μετά την αποκοπή είναι c_{jk} . Σύμφωνα με την τριγωνική ανισότητα ισχύει ότι $c_{ik} \leq c_{ji} + c_{ik}$. Άρα το 4 βήμα του αλγόριθμου δεν αυξάνει το κόστος του κύκλου του Euler και κατά συνέπεια η υπολογιζόμενη διαδρομή είναι το πολύ διπλάσια από την βέλτιστη. Το 5 βήμα μπορεί αν οδηγήσει μόνο σε περαιτέρω ελάττωση του κόστους της διαδρομής.

Στην συνέχεια δίνεται ένα παράδειγμα λειτουργίας του αλγόριθμου.

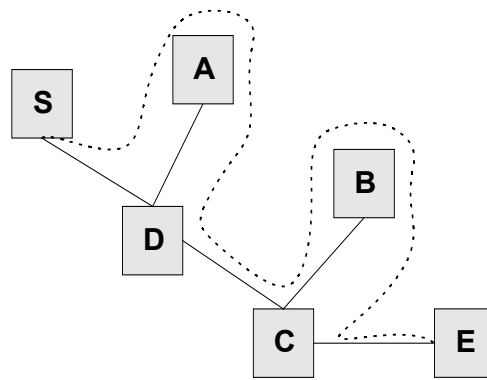
Βήμα 1: Γράφος με βάρη και σημείο εκκίνησης το S

	S	A	B	C	D	E
S		100	110	80	30	130
A	100		150	120	70	170
B	110	150		40	80	80
C	80	120	40		50	50
D	30	70	80	50		100
E	130	170	80	50	100	

Βήμα 2: MST

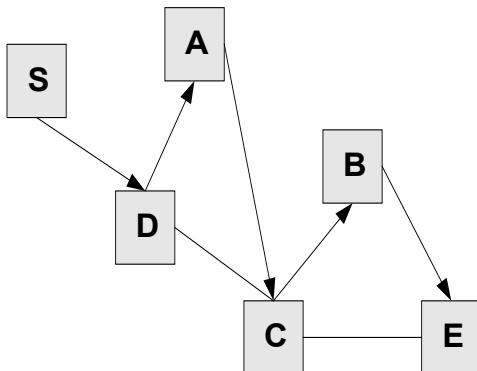


Βήμα 3 διαδρομή: **SDADCBCE**



Βήμα 4: αποκοπή επαναλαμβανόμενων κόμβων: **SDADCBCE** → **SDACBE**

Βήμα 5: Εύρεση μικρότερου μονοπατιού: **S** → **D** → **A** → **C** → **B** → **E**

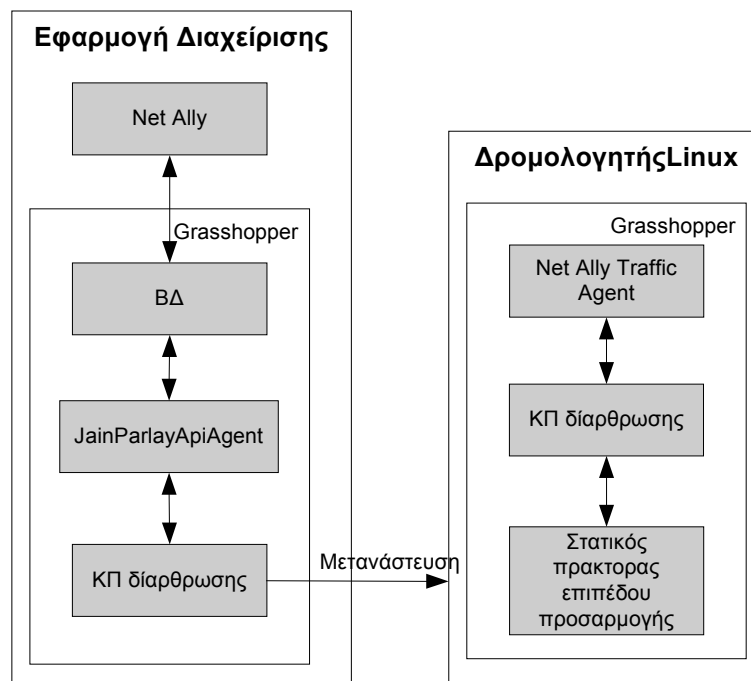


6.3.2.3 Πειραματική εφαρμογή

Στο κεφαλαίο 4 παρουσιάστηκε ένα ολοκληρωμένο σύστημα διάρθρωσης και παρακολούθησης ενός IP δικτύου ΔΥ. Η λειτουργία του συστήματος επεκτάθηκε με την δημιουργία μια οντότητας Βελτιστοποίησης Διαδρομής (ΒΔ), που υλοποιεί τον

παραπάνω αλγόριθμο. Ο ΚΠ υπεύθυνος για την διάρθρωση των δρομολογητών (QoSConfigureConnectionAgent) δέχεται εντολές από τον ΒΔ, όσον αφορά την διαδρομή που θα ακολουθήσει για να διαρθρώσει μια σειρά από δρομολογητές. Η αρχιτεκτονική του συστήματος, μετά την αναβάθμιση του, παρουσιάζεται στο σχήμα 6.4.

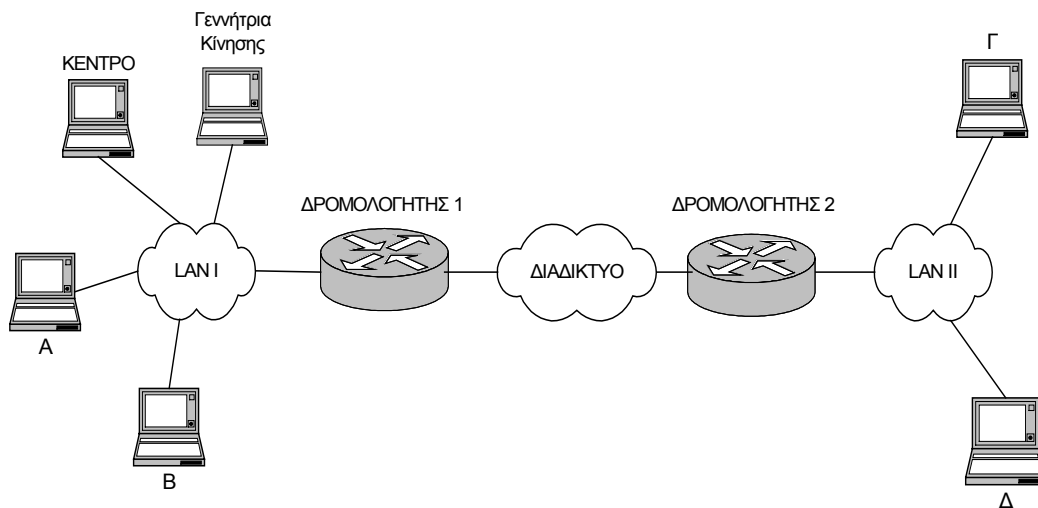
Ο JainParlayApiAgent που δημιουργεί τον ΚΠ διάρθρωσης τροφοδοτεί τον ΒΔ με τους δρομολογητές στόχους. Ο ΒΔ εκτελεί τον αλγόριθμο και επιστέφει στον JainParlayApiAgent την βελτιωμένη διαδρομή. Κατόπιν, δημιουργείται ο ΚΠ διάρθρωσης με χρήση αυτής της πληροφορίας. Για την εκτέλεση των τε εντολών στους δρομολογητές, ο ΚΠ διάρθρωσης επικοινωνεί με την στατικό πράκτορα επιπέδου προσαρμογής, ο οποίος εκτελείται στον δρομολογητή. Η καθυστέρηση μετάδοσης μεταξύ των μηχανημάτων προορισμού, παρακολουθείται χρησιμοποιώντας την πλατφόρμα Net Ally των Viola Networks [Viola]. Το Net Ally είναι ένα κατανεμημένο σύστημα παρακολούθησης απόδοσης δικτύων βασισμένο σε ένα κέντρο δοκιμών και διάφορους Αντιπροσώπους Κίνησης (Net Ally traffic agents).



Σχήμα 6.4: Αρχιτεκτονική του αναβαθμισμένου συστήματος διάρθρωσης

6.3.2.4 Πειραματικά Δεδομένα

Η τοπολογία του δικτύου, που χρησιμοποιήθηκε για να αξιολογηθεί η βελτίωση στην απόδοση που επιφέρει η χρήση του αλγόριθμου, παρουσιάζεται στο σχήμα 6.5. Το δίκτυο αποτελείται από δυο 10Mbps Ethernet LANs. Το τοπικό LAN I περιλαμβάνει τέσσερις κόμβους. Οι κόμβοι Α και Β είναι δρομολογητές Linux, στους οποίους εκτελούνται οι ΠΚΠ, ο στατικός πράκτορας επιπέδου προσαρμογής και οι Αντιπροσώπους Κίνησης Net Ally. Ο κόμβος KENTPO είναι εξοπλισμένος με το Net Ally και την εφαρμογή διαχείρισης που εκτελεί τον ΒΔ. Ο κόμβος Παραγωγής Κίνησης (ΠΚ - Traffic Generation) είναι ένας τερματικός σταθμός Linux εφοδιασμένος με λογισμικό παραγωγής κίνησης [Iperf]. Η γεννήτρια κίνησης θα χρησιμοποιηθεί για την δημιουργία τεχνητής καθυστέρηση στον κόμβο Β. Το τοπικό LAN II περιέχει μόνο δύο Linux δρομολογητές, τους κόμβους Γ και Δ. Οι κόμβοι συνδέονται στο LANs με χρήση hubs. Η σύνδεση με το διαδίκτυο πραγματοποιείται μέσω με μιας σύνδεσης 128Kbps.



Σχήμα 6.5: Τοπολογία πειραματικού δικτύου

Το σενάριο της πειραματικής διαδικασίας περιλαμβάνει την διάρθρωση των τεσσάρων δρομολογητών με και χωρίς την χρήση του ΒΔ. Για λόγους αξιολόγησης, εκτελέστηκαν δυο πειράματα με διαφορετικές δικτυακές συνθήκες.

Στο πρώτο πείραμα καμία κίνηση δεν παρήχθη από τη Γεννήτρια Κίνησης. Χρησιμοποιώντας το Net Ally οι μετρηθήσες καθυστερήσεις μεταξύ των κόμβων παρουσιάζονται στον πίνακα 6.1 :

Πίνακας 6.1: Καθυστέρηση μετάδοσης μεταξύ των κόμβων του δικτύου

Συνδέσεις	Καθυστέρηση (ms)
A $\leftrightarrow$ B	8
A $\leftrightarrow$ Γ	722
A $\leftrightarrow$ Δ	725
B $\leftrightarrow$ Γ	725
B $\leftrightarrow$ Δ	724
Γ $\leftrightarrow$ Δ	11

Αφετηρία ήταν ο κόμβος A. Το μονοπάτι που προτάθηκε από τη οντότητα BM ήταν το A $\rightarrow$ B $\rightarrow$ Γ $\rightarrow$ Δ και ο συνολικός χρόνος ταξιδιού, συμπεριλαμβανομένης της εκτέλεσης της στοιχειώδους εργασίας σε κάθε κόμβο, ήταν 5.4 sec. Ο συνολικός χρόνος εκτέλεσης της στοιχειώδους εργασίας αυξάνεται σε 9.7 sec, όταν δεν χρησιμοποιηθεί ο BM. Σε αυτήν την περίπτωση ο αντιπρόσωπος καθοδηγήθηκε να επισκεφτεί τους κόμβους A, Δ, B και Γ με αυτή τη σειρά. Αξίζει να σημειωθεί ότι το βέλτιστο μονοπάτι σε αυτό το παράδειγμα είναι το A $\rightarrow$ B $\rightarrow$ Δ $\rightarrow$ Γ, αλλά ο συνολικός χρόνος ταξιδιού ακολουθώντας αυτό το μονοπάτι είναι ίδιος με αυτόν του μονοπατιού που προτάθηκε από τον BΔ. Στον πίνακα 6.2 παρουσιάζονται συγκεντρωτικά τα αποτελέσματα:

Πίνακας 6.2: Αποτελέσματα ταξιδιού με και χωρίς βελτιστοποίηση

Τρόπος Λειτουργίας	Μονοπάτι	Απαιτούμενος Χρόνος
Με BΔ	A $\rightarrow$ B $\rightarrow$ Γ $\rightarrow$ Δ	5.4 sec
Χωρίς BΔ	A $\rightarrow$ Δ $\rightarrow$ B $\rightarrow$ Γ	9.7 sec

Για την διεξαγωγή του δεύτερου πειράματος, τεχνητή καθυστέρηση προστέθηκε στον κόμβο B. Η Γεννήτρια κίνησης παρήγαγε 10Mbps κίνησης προς τον κόμβο αυτό. Χρησιμοποιώντας το Net Ally οι μετρηθείσες καθυστερήσεις μεταξύ των κόμβων ήταν:

Πίνακας 6.3: Καθυστέρηση μετάδοσης μεταξύ των κόμβων του δικτύου με τεχνητή καθυστέρηση στον κόμβο B

Συνδέσεις	Καθυστέρηση (ms)
-----------	------------------

$A \leftrightarrow B$	1512
$A \leftrightarrow \Gamma$	716
$A \leftrightarrow \Delta$	729
$B \leftrightarrow \Gamma$	2234
$B \leftrightarrow \Delta$	732
$\Gamma \leftrightarrow \Delta$	11

Η διαδρομή που προτάθηκε από τον ΒΔ ήταν η $A \rightarrow \Gamma \rightarrow \Delta \rightarrow B$ (που σε αυτήν την περίπτωση είναι το βέλτιστο μονοπάτι) και ο συνολικός χρόνος ταξιδιού συμπεριλαμβανομένης της εκτέλεσης της εργασίας σε κάθε κόμβο ήταν 6.8 sec. Ο συνολικός χρόνος εκτέλεσης χωρίς χρήση του ΒΔ που χρησιμοποιεί το μονοπάτι $A \rightarrow B \rightarrow \Gamma \rightarrow \Delta$ ως επισκεπτόμενη ακολουθία, αυξήθηκε σε 12.4 sec.

Πίνακας 6.4 : Αποτελέσματα ταξιδιού με και χωρίς βελτιστοποίηση

Τρόπος Λειτουργίας	Μονοπάτι	Απαιτούμενος Χρόνος
Με ΒΔ	$A \rightarrow \Gamma \rightarrow \Delta \rightarrow B$	6.8 sec
Χωρίς ΒΔ	$A \rightarrow B \rightarrow \Gamma \rightarrow \Delta$	12.4 sec

Η πειραματική διαδικασία απέδειξε ότι η καθυστέρηση μετάδοσης επηρεάζει την συνολική απόδοση ενός συστήματος ΚΠ, στο οποίο ένας ΚΠ επισκέπτεται μια σειρά μηχανημάτων. Η χρήση του αλγόριθμου επιφέρει μια βελτίωση της απόδοσης κατά 45%. Η βελτιωμένη απόδοση είναι αποτέλεσμα της ελάττωσης του χρόνου μετανάστευσης του πράκτορα. Η βελτίωση αυτή είναι η μέγιστη που μπορεί να επιτευχθεί, αφού η βελτιωμένη διαδρομή συγκρίνεται με το χειρότερο σενάριο επίσκεψης των κόμβων. Στην περίπτωση μη χρήσης του ΒΔ, ο πράκτορας αναγκάζεται να περάσει δυο φορές από την αργή σύνδεση των 128Kbps. Το τίμημα για αυτήν την βελτίωση είναι η συνεχής παρακολούθηση της καθυστέρησης των κόμβων του δικτύου, η οποία παράγει μια επιπλέον κίνηση στο δίκτυο. Ένα σημείο το οποίο αξίζει να τονιστεί, είναι ότι μικρές διαφορές στην καθυστέρηση μετάδοσης δεν επηρεάζουν σε μεγάλο βαθμό τον συνολικό χρόνο μετανάστευσης. Στο πρώτο πείραμα ο ΒΔ δεν παρέχει την βέλτιστη διαδρομή. Η διάφορα όμως μεταξύ της βέλτιστης διαδρομής και της προτεινόμενης ήταν μόνο 1 msec. Αυτό έχει ως αποτέλεσμα, να μην υπάρχει διάφορα στον συνολικό χρόνο εκτέλεσης του πράκτορα στις δυο περιπτώσεις.

6.4 Βελτίωση της απόδοσης με κατάτμηση μιας εργασίας σε πολλούς πράκτορες.

Στις προηγούμενες παραγράφους παρουσιάστηκαν δυο μέθοδοι βελτίωσης της απόδοσης μιας εφαρμογής ΚΠ, με έλεγχο της σειράς επισκέψεων, προκειμένου να ελαχιστοποιηθεί ο αναμενόμενος χρόνος που χρειάζεται ένας πράκτορας για να ολοκληρώσει μια εργασία. Ως εκ τούτου, ελαττώνοντας τον χρόνο εκτέλεσης των ΚΠ, βελτιώνεται ο συνολικός χρόνος εκτέλεσης της εφαρμογής. Αυτές οι προτεινόμενες λύσεις προσπαθούν να βελτιώσουν την απόδοση από μια τεχνική σκοπιά. Οι προτεινόμενες μέθοδοι στοχεύουν στη βελτίωση της απόδοσης του συστήματος ΚΠ, βελτιώνοντας την απόδοση κάθε μεμονωμένου πράκτορα (δηλαδή, μειώνοντας το χρόνο μετανάστευσης ή εκτέλεσης). Το μειονέκτημα αυτών των μεθόδων είναι ότι η βελτιστοποίηση της σειράς επισκέψεων ενός πράκτορα εξαρτάται από τις συνθήκες του υποκείμενου δικτύου και των μηχανημάτων-στόχων. Έτσι, πρέπει να παρακολουθείται ένας αριθμός παραμέτρων απόφασης, όπως η καθυστέρηση δικτύου, το διαθέσιμο εύρος ζώνης, ο ρυθμός απώλειας πακέτων και το φόρτο της ΚΜΕ των μηχανημάτων-στόχων.

Ένας άλλος τρόπος βελτίωση της απόδοσης, ενός συστήματος βασισμένου σε ΚΠ, είναι αλλάζοντας τη σχεδίαση. Πρόκειται για μια απλή ιδέα: Διαμοιρασμός της εργασίας που είχε ανατεθεί σε έναν πράκτορα σε περισσότερους. Η βελτίωση της απόδοσης είναι αποτέλεσμα της παράλληλης επεξεργασίας [Muthukrishnan99] και της μείωσης του χρόνου μετανάστευσης του πράκτορα. Το πρώτο επιχείρημα είναι προφανές. Διαμοιράζοντας την εργασία ενός πράκτορα σε αρκετούς που δρουν αυτόνομα, ο ολικός χρόνος εκτέλεσης πιθανότατα θα μειωθεί. Χρησιμοποιώντας περισσότερους του ενός πράκτορες για την περάτωση μιας εργασίας, καθένας τους πρέπει να επισκεφτεί ένα σταθερό αριθμό μηχανημάτων-στόχων διατηρώντας το μέγεθός του μικρό, διευκολύνοντας έτσι τη διαδικασία μετανάστευσης. Το μόνο μειονέκτημα, το οποίο και θα διερευνηθεί στην συνέχεια, είναι ότι η προσέγγιση αυτή έχει μεγαλύτερες απαιτήσεις σε εύρος ζώνης.

Αρκετοί ορισμοί έχουν προταθεί για τον όρο **Σύστημα Πολλαπλών Πρακτόρων ΣΠΠ (MAS–Multi Agent System)**. Σύμφωνα με το [Durfée89] ένα ΣΠΠ είναι ένα χαλαρά συνδεδεμένο δίκτυο οντοτήτων επίλυσης προβλημάτων, οι οποίες συνεργάζονται για την

εύρεση απαντήσεων σε προβλήματα που ξεπερνούν τις ατομικές δυνατότητες γνώσης κάθε επιμέρους οντότητας. Πιο πρόσφατα, δόθηκε στον όρο ΣΠΠ ένα γενικότερο νόημα το οποίο χρησιμοποιείται πλέον για όλα τα είδη συστημάτων αποτελούμενων από πολλαπλά αυτόνομα συστατικά, που έχουν τα εξής χαρακτηριστικά [Jennings98]:

- Κάθε πράκτορας έχει ανεπαρκείς δυνατότητες επίλυσης ενός προβλήματος
- Δεν υπάρχει καθολικός έλεγχος του συστήματος
- Τα δεδομένα είναι κεντροποιημένα
- Ο υπολογισμός γίνεται ασύγχρονα

Με βάση την θεώρηση ότι η κατάτμηση εργασίας σε περισσότερους του ενός πράκτορες οδηγεί σε βελτίωση της απόδοσης, προκύπτει το εξής ερώτημα “ποιος είναι ο αριθμός πρακτόρων που βελτιστοποιεί την απόδοση του ΣΠΠ;”. Η έρευνα λοιπόν επικεντρώνεται στην εύρεση του αριθμού πρακτόρων που οδηγεί στη βέλτιστη απόδοση. Προτείνεται μια μεθοδολογία βελτιστοποίησης και αξιολογείται με χρήση ενός πειραματικού ΣΠΠ, σχεδιασμένο και υλοποιημένο για τις ανάγκες της έρευνας.

6.4.1 Μεθοδολογία Βελτιστοποίησης του αριθμού ΚΠ ενός ΣΠΠ

Οι παράμετροι που επηρεάζουν την απόδοση ενός ΣΠΠ είναι οι παρακάτω:

- Ο χρόνος δημιουργίας των πρακτόρων
- Ο χρόνος μετανάστευσης των πρακτόρων
- Ο χρόνος εκτέλεσης των πρακτόρων στο μηχανήμα-στόχο

Αν η εργασία ενός ΚΠ, ο οποίος πρέπει να επισκεφτεί ένα σταθερό αριθμό μηχανημάτων-στόχων, διαμοιραστεί σε αρκετούς πράκτορες, τότε οι προκύπτοντες πράκτορες δε θα επισκεφτούν όλοι τον ίδιο αριθμό μηχανημάτων. Εφόσον όλοι οι πράκτορες δρουν ασύγχρονα, εκτελώντας την ίδια εργασία σε διαφορετικά μηχανήματα-στόχους, ο ολικός χρόνος εκτέλεσης της εργασίας πρέπει να υπολογιστεί βάσει του πράκτορα με τον μεγαλύτερο χρόνο εκτέλεσης. Η κατάσταση του υποκείμενου δικτύου και των μηχανημάτων-στόχων δε λαμβάνονται υπόψη, όσον αφορά τον χρόνο μετανάστευσης και εκτέλεσης των πρακτόρων. Υποθέτουμε ότι η κατάσταση αυτή είναι ίδια για όλους τους πράκτορες και δεν επηρεάζει το χρόνο εκτέλεσης. Βάσει αυτής της υπόθεσης, ο πράκτορας με το μεγαλύτερο χρόνο εκτέλεσης είναι αυτός που θα

επισκεφτεί περισσότερα μηχανήματα από τους υπόλοιπους. Η ανάθεση των μηχανημάτων-στόχων στους πράκτορες, πρέπει να γίνει κατά τέτοιο τρόπο, ώστε να ελαχιστοποιείται ο ολικός χρόνος εκτέλεσης της εργασίας. Αυτό μπορεί να γίνει μόνο μέσω της διανομής των μηχανημάτων-στόχων στους διαθέσιμους πράκτορες, έτσι ώστε να ελαχιστοποιείται ο μέγιστος αριθμός μηχανημάτων-στόχων που ανατίθενται σε έναν αυτούς. Έχοντας αυτή την αρχή ως βάση, ο μέγιστος αριθμός μηχανημάτων που θα επισκεφτεί ένας πράκτορας ενός ΣΠΠ, μπορεί να εξαχθεί ως εξής:

Αν ο συνολικός αριθμός των μηχανημάτων-στόχων για μια δεδομένη εργασία είναι 1, τότε ο μέγιστος αριθμός μηχανημάτων n που ένας πράκτορας ενός ΣΠΠ αποτελούμενο από r πράκτορες θα επισκεφθεί είναι:

$$n = \begin{cases} 1 \text{ div } r & \text{αν } 1 \bmod r = 0 \\ (1 \text{ div } r) + 1 & \text{αν } 1 \bmod r \neq 0 \end{cases} \quad (6.2)$$

Έστω ότι ο αριθμός των μηχανημάτων στόχων για μια δεδομένη εργασία είναι ίσος με 10 και ο προτεινόμενος αριθμός πρακτόρων είναι ίσος με 5. Η πράξη $10 \bmod 5 = 0$ και καθένας από τους πράκτορες θα επισκεφθεί $10 \text{ div } 5 = 2$ μηχανήματα, για να ολοκληρωθεί η εργασία. Τώρα, εάν ο αριθμός των προτεινόμενων πρακτόρων ισούται με 3, τότε $10 \text{ div } 3 = 3$ και $10 \bmod 3 = 1$. Έτσι, καθένας από τους πράκτορες θα επισκεφθεί 3 μηχανήματα και ένας από αυτούς πρέπει επισκεφθεί ένα περισσότερο, προκειμένου να ολοκληρωθεί η εργασία. Ως εκ τούτου, στη δεύτερη περίπτωση ο μέγιστος αριθμός μηχανημάτων, που ένας πράκτορας του ΣΠΠ θα επισκεφθεί ισούται με $3 + 1 = 4$.

Ο ολικός χρόνος εκτέλεσης μιας εργασίας εκτελούμενης από μια εφαρμογή πολλαπλών πρακτόρων εκφράζεται από την ακόλουθη εξίσωση.

$$T(r) = \sum_{i=1}^n \{ M(i) + E(i) \} + C(r) \quad (6.3)$$

Όπου:

T: ολικός χρόνος εκτέλεσης μιας εργασίας

n: μέγιστος αριθμός μηχανημάτων-στόχων

M: χρόνος μετανάστευσης του πρακτόρων

C: χρόνος δημιουργίας των πρακτόρων

E: χρόνος εκτέλεσης στα μηχανήματα-στόχους

r: αριθμός των πρακτόρων

Ο βέλτιστος αριθμός των πρακτόρων, που χρησιμοποιούνται σε μια εφαρμογή πολλαπλών πρακτόρων, είναι εκείνος που ελαχιστοποιεί το αποτέλεσμα της εξίσωσης 6.3 [Kalogeropoulos02].

Με την αύξηση του αριθμού των πρακτόρων του συστήματος, σύμφωνα με την σχέση 6.2 ο αριθμός των μηχανημάτων-στόχων, που ένας ή περισσότεροι πράκτορες του συστήματος πρέπει να επισκεφθούν, μειώνεται. Αυτό οδηγεί σε μια μικρότερη τιμή για το πρώτο μέλος της σχέσης 6.3, το οποίο εξαρτάται από τον αριθμό των μηχανημάτων-στόχων. Από την άλλη μεριά, το δεύτερο μέλος της σχέσης, που εξαρτάται από το χρόνο δημιουργίας των πρακτόρων, αυξάνεται. Η επίδραση των δύο αυτών μελών στον ολικό χρόνο εκτέλεσης μπορεί να διερευνηθεί μόνο μέσω μετρήσεων σε μια πραγματική εφαρμογή. Γι' αυτό το λόγο σχεδιάστηκε και υλοποιήθηκε ένα ΣΠΠ, το οποίο παρουσιάζεται στην συνέχεια.

6.4.2 Σχεδίαση και υλοποίηση του πειραματικού ΣΠΠ

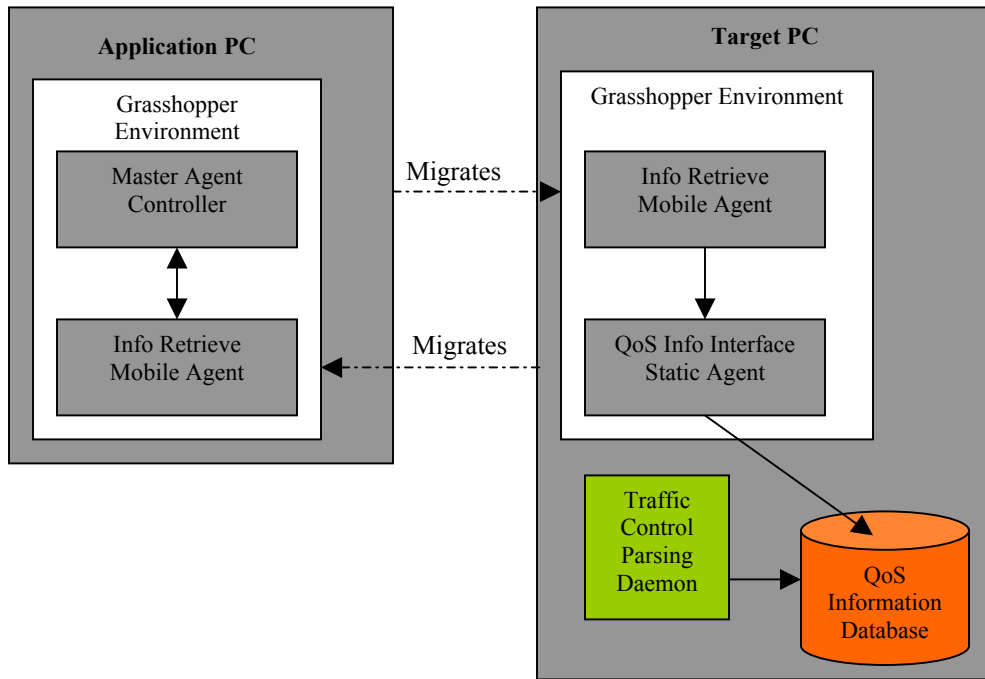
Μια απ' τις πολλά υποσχόμενες εφαρμογές ΚΠ είναι η συλλογή πληροφοριών [Rus96]. Αυτό το είδος εφαρμογών, περιλαμβάνει πρόσβαση σε τεράστια ποσά δεδομένων δια μέσου ενός δικτύου. Αντί να μεταδίδονται δεδομένα δια μέσω του δικτύου, ένας πράκτορας μεταναστεύει στο μηχανήμα όπου βρίσκεται η βάση δεδομένων, εκτελεί την εργασία του εκεί και κατόπιν επιστρέφει στο αρχικό μηχανήμα μεταφέροντας ένα αποτέλεσμα. Η χρήση ΚΠ σε συστήματα συλλογής πληροφοριών μπορεί να είναι αποτελεσματικότερη, αν η συλλογή δεδομένων που αποτελεί την καλύτερη απάντηση σε μια ερώτηση, είναι το σύνολο των δεδομένων που αποκτούνται από κατανεμημένες ετερογενείς πηγές πληροφοριών [Oates95].

Προκειμένου να δοκιμαστεί η επίδραση του αριθμού των πρακτόρων στην απόδοση μιας εφαρμογής βασισμένης σε πράκτορες, σχεδιάστηκε και υλοποιήθηκε ένα Σύστημα Συλλογής Πληροφοριών (Information Gathering System-IGS). Το σύστημα παρέχει στατιστική πληροφορία ΠοιΥπ, για ένα δίκτυο ΔΥ βασισμένο σε δρομολογητές τύπου Linux.

Προκειμένου να αποκτηθεί μια ξεκάθαρη εικόνα της κίνησης ενός δίκτυο ΔΥ, το οποίο αποτελείται από έναν αριθμό δρομολογητών, μια εφαρμογή στατιστικής παρακολούθησης πρέπει να ανακτά τα πακέτα που έχουν αποσταλεί (sent) και απορριφθεί (dropped), ανά κλάση υπηρεσίας, για ολόκληρο το δίκτυο. Επιπρόσθετα, η πληροφορία αυτή πρέπει να παρέχεται για διαφορετικές ώρες της ημέρας. Η τεχνολογία ΚΠ ταιριάζει απόλυτα στις απαιτήσεις εφαρμογών αυτού του είδους, διότι ο πράκτορας μπορεί να μεταναστεύσει στους δρομολογητές και να ανακτήσει πληροφορία ανά κλάση υπηρεσίας για διαφορετικές ώρες της ημέρας. Εξαιτίας του μεγάλου ποσού της προς ανάκτηση πληροφορίας, η προσέγγιση με ΚΠ είναι λιγότερο απαιτητική σε εύρος ζώνης απ' ό,τι το παράδειγμα πελάτη/εξυπηρετητή. Ένα άλλο πλεονέκτημα είναι ότι οι ΚΠ μπορούν να επεξεργαστούν την ανακτηθείσα πληροφορία κατά τη μετακίνησή τους μεταξύ των δρομολογητών, προτού επιστρέψουν στο αρχικό μηχάνημα με το αποτέλεσμα. Σημαντικό στοιχείο είναι ότι το έργο της επίσκεψης των δρομολογητών ενός δικτύου ΔΥ και της συλλογής της απαιτούμενης πληροφορίας μπορεί να πραγματοποιηθεί είτε με τη χρήση ενός πράκτορα είτε με διαμοιρασμό και ανάθεση σε διάφορους πράκτορες. Η τελευταία αυτή παρατήρηση κάνει την προτεινόμενη εφαρμογή κατάλληλη για τις ανάγκες του πειράματος. Το σχήμα 6.6 καταδεικνύει την αρχιτεκτονική του Συστήματος Συλλογής Στατιστικής Πληροφορίας ΠοιΥπ.

Η ΠΚΠ που χρησιμοποιείται είναι η Grasshopper, η οποία είναι υλοποιημένη με χρήση της γλώσσας προγραμματισμού Java. Ως εκ τούτου, το περιβάλλον του συστήματος είναι το Java Virtual Machine (JVM).

Η υπομονάδα Master controller agent είναι υπεύθυνη για την δημιουργία και εντολοδότηση των υπομονάδων Info Retrieve Mobile Agents, καθώς και την επεξεργασία και αναφορά των αποτελεσμάτων. Ο αριθμός των χρησιμοποιούμενων πρακτόρων από το ΣΠΠ εισάγεται ως όρισμα. Ο Master Controller Agent πρέπει να δημιουργήσει τους πράκτορες και να αναθέσει σε καθένα από αυτούς τα μηχανήματα-στόχους. Η ανάθεση βασίζεται στη μέθοδο που περιγράφηκε στην παράγραφο 6.4.1, η οποία ελαχιστοποιεί το μέγιστο αριθμό μηχανημάτων που επισκέπτονται οι πράκτορες. Όταν όλοι οι πράκτορες έχουν επιστρέψει, ο Master Controller Agent πρέπει να συνοψίσει και να αναφέρει τα αποτελέσματα στο χρήστη.



Σχήμα 6.6: Αρχιτεκτονική και υπομονάδες Συστήματος Συλλογής Στατιστικής Πληροφορίας ΠοιΥπ

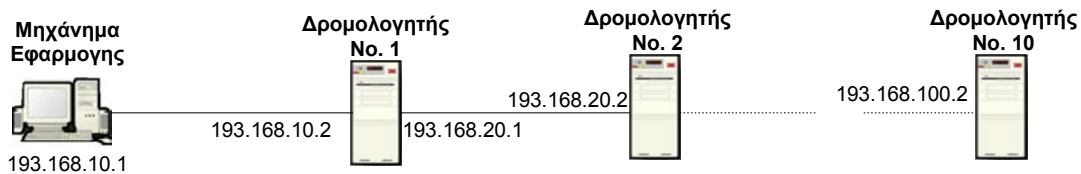
Οι υπομονάδες info Retrieve mobile agents μεταναστεύουν στα μηχανήματα-στόχους, σύμφωνα με τη λίστα στόχων που ανατίθεται σ' αυτούς απ' τον controller. Όντας στο στόχο, διασυνδέονται με την υπομονάδα QoS Info Interface Static Agent που βρίσκεται εκεί, ανακτώντας τα πακέτα που έχουν αποσταλεί και απορριφθεί, ανά κλάση υπηρεσίας, για μια συγκεκριμένη ώρα της ημέρας. Οι ΚΠ προ-επεξεργάζονται τα δεδομένα, αθροίζοντάς τα καθώς μετακινούνται μεταξύ των μηχανημάτων-στόχων. Κατ' αυτό τον τρόπο, επιστρέφουν στον κόμβο-ελεγκτή με ένα μόνο αποτέλεσμα που αφορά τον αριθμό πακέτων που έχουν αποσταλεί και απορριφθεί ανά κλάση υπηρεσίας, για όλους τους δρομολογητές που έχουν επισκεφθεί.

Η υπομονάδα QoS Info Interface Static Agent ανακτά τη σχετιζόμενη με την ΠοιΥπ πληροφορία από μια βάση δεδομένων MySQL, που εδρεύει γι' αυτό το σκοπό στο μηχανήμα-στόχο. Η υπομονάδα Traffic Control Parsing Daemon είναι υπεύθυνη για την εισαγωγή των πακέτων που έχουν αποσταλεί και απορριφθεί ανά κλάση υπηρεσίας στη βάση δεδομένων, κάθε μία ώρα.

Η ανάκτηση των πακέτων που έχουν αποσταλεί και απορριφθεί από του δρομολογητές Linux γίνεται με χρήση του εργαλείου `tc`, η λειτουργία του οποίου έχει παρουσιαστεί αναλυτικά στην παράγραφο 4.4.3.1.2. Η υπομονάδα Traffic Control Parsing Daemon αναλύει (parses) την παρεχόμενη από την εντολή `tc -s class show` πληροφορία, προκειμένου να πάρει τα πακέτα που έχουν αποσταλεί και απορριφθεί ανά κλάση υπηρεσίας, αποθηκεύοντάς τα μετά στην βάση δεδομένων.

6.4.3 Πειραματικά αποτελέσματα

Προκειμένου να δοκιμαστεί και να εκτιμηθεί η συμπεριφορά του συστήματος, όταν ο αριθμός των χρησιμοποιούμενων πρακτόρων αλλάζει, δημιουργήθηκε μια πλατφόρμα δοκιμών με δρομολογητές ΔΥ τύπου Linux. Οι δοκιμές που πραγματοποιήθηκαν επικεντρώθηκαν όχι μόνο στο χρόνο εκτέλεσης του ΣΠΠ, αλλά και στους δαπανούμενους πόρους δικτύου για διαφορετικό αριθμό πρακτόρων. Η πλατφόρμα δοκιμών παρουσιάζεται στο σχήμα 6.7.



Σχήμα 6.7: Η πλατφόρμα δοκιμών του Συστήματος Συλλογής Στατιστικής Πληροφορίας ΠοιΥπ

Η πλατφόρμα δοκιμών αποτελείται από 11 μηχανήματα: Δέκα μηχανήματα-στόχους και ένα μηχανήμα που φιλοξενεί την υπομονάδα Master Agent Controller. Τα μηχανήματα-στόχοι είναι δρομολογητές Linux, ενώ η υπομονάδα Master Agent Controller εκτελείται σε ένα μηχανήμα με λειτουργικό σύστημα MS Windows 2000. Τα χαρακτηριστικά των μηχανημάτων παρουσιάζονται στον Πίνακα 6.5.

Πίνακας 6.5: Χαρακτηριστικά των μηχανημάτων της πλατφόρμας δοκιμών

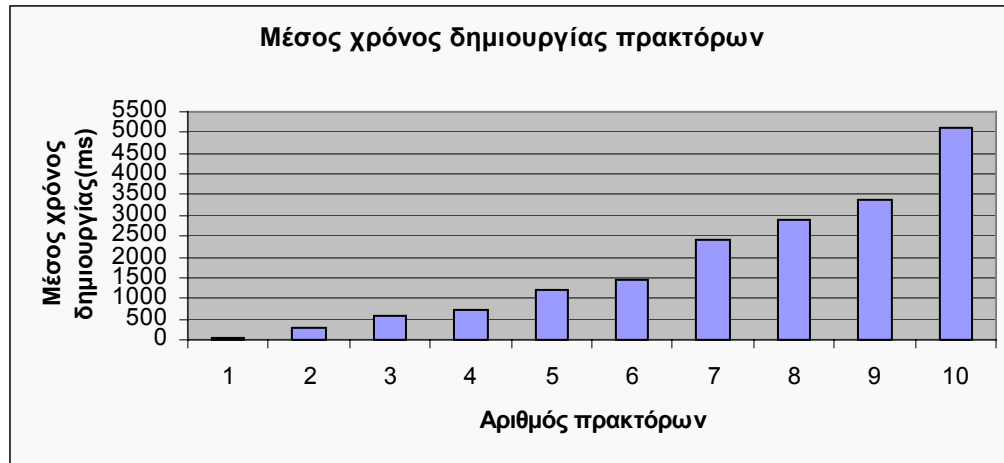
Τύπος	KME	Μνήμη	Δίσκος	ΛΣ
Μηχάνημα-στόχος	Pentium IV 1.7 GHz	128M	20G	Linux (2.4.1 kernel version)
ΣΠΠ	Pentium IV 1.7 GHz	1G	20G	Windows 2000 Server

Το σενάριο του πειράματος απαιτεί την συλλογή της πληροφορίας από δέκα δρομολογητές, χρησιμοποιώντας διαφορετικό αριθμό ΚΠ για την ολοκλήρωση της

εργασίας. Το πείραμα ξεκινάει με ένα μόνο ΚΠ και συνεχίζει αυξάνοντας τον αριθμό των πρακτόρων κατά ένα, μέχρι να αποστέλλεται ένας πράκτορας ανά μηχανήμα-στόχο. Έτσι, μπορεί να παρακολουθηθεί η επίδραση του αριθμού των ΚΠ στο χρόνο εκτέλεσης.

Εφόσον οι δοκιμές πραγματοποιούνται σε ένα πειραματικό IP δίκτυο με εύρους ζώνης 10Mbps, αποτελούμενο από ταυτόσημα μηχανήματα, μπορούν να γίνουν οι ακόλουθες υποθέσεις. Ο μέσος χρόνος μετανάστευσης και εκτέλεσης των πρακτόρων, από το ένα μηχανήμα-στόχο στο άλλο, είναι σταθερός. Η υπόθεση που αφορά το χρόνο μετανάστευσης είναι στην προκειμένη περίπτωση έγκυρη, επειδή τα πρόσθετα δεδομένα που μεταφέρονται από τον πρακτόρων, καθώς μετακινείται από το ένα μηχανήμα-στόχο στο άλλο, είναι μόλις 10 bytes. Αυτό επιτυγχάνεται με την άθροιση της συλλεγόμενης πληροφορίας όπως περιγράφηκε στην προηγούμενη παράγραφο. Ο χρόνος εκτέλεσης ενός πράκτορα στα απομακρυσμένα μηχανήματα εξαρτάται από τον τύπο και το τρέχον φόρτο εργασίας της ΚΜΕ του μηχανήματος. Αφού όλα τα μηχανήματα-στόχοι είναι ταυτόσημα και ο φόρτος εργασίας πλησιάζει το μηδέν, ο μέσος χρόνος απομακρυσμένης εκτέλεσης είναι σταθερός. Βάσει πραγματικών μετρήσεων ο μέσος χρόνος μετανάστευσης του πράκτορα από ένα μηχανήμα σε ένα άλλο ισούται με 1450 ms. Ο χρόνος απομακρυσμένης εκτέλεσης ισούται με 20 ms και ο χρόνος εκτέλεσης του master agent controller, μετά την επιστροφή όλων των πρακτόρων, ισούται με 80 ms.

Ο ολικός χρόνος εκτέλεσης της εργασίας, δίνεται από τη σχέση 6.3, που διατυπώθηκε στην παράγραφο 6.4.1. Οι υποθέσεις ότι οι χρόνοι μετανάστευσης και εκτέλεσης των πρακτόρων είναι σταθεροί διευκολύνουν τον υπολογισμό του ολικού χρόνου εκτέλεσης. Η μόνη παράμετρος που πρέπει να μετρηθεί είναι ο μέσος χρόνος δημιουργίας των πρακτόρων. Ο χρόνος αυτός, για διαφορετικό αριθμό πρακτόρων παρουσιάζεται στο ιστόγραμμα 6.1.



Ιστόγραμμα 6.1. Μέσος χρόνος δημιουργίας πρακτόρων

Όπως αναμενόταν, ο χρόνος δημιουργίας αυξάνεται σημαντικά, καθώς ο αριθμός των πρακτόρων αυξάνει. Ο μέσος χρόνος δημιουργίας υπολογίστηκε χρησιμοποιώντας έναν αριθμό μετρήσεων που πάρθηκαν στο μηχανήμα που φιλοξενούσε τον master agent controller. Για αριθμό πρακτόρων μεγαλύτερο του πέντε, η τυπική απόκλιση πλησιάζει τα 250 ms.

Το αποτέλεσμα της σχέσης 6.3, για την εργασίας της συλλογής και αναφοράς των στατιστικών στοιχείων παρουσιάζεται στον πίνακα 6.6. Ο μέγιστος αριθμός μηχανημάτων-στόχων προκύπτει από τη σχέση 6.2. Ο ολικός αριθμός μηχανημάτων-στόχων ισούται με 10. Στη συγκεκριμένη εργασία, οι πράκτορες πρέπει να επιστρέψουν και να αναφέρουν στον ελεγκτή πράκτορα. Αυτός ο χρόνος μετανάστευσης πρέπει να αθροιστεί στον ολικό χρόνο μετανάστευσης. Ο χρόνος εκτέλεσης του ελεγκτή πράκτορα, μετά την επιστροφή όλων των πρακτόρων, πρέπει επίσης να προστεθεί στο χρόνο εκτέλεσης.

Προκειμένου να αποδειχθεί ότι η προτεινόμενη μεθοδολογία παρέχει τον ολικό χρόνο εκτέλεσης της εργασίας για ένα διαφορετικό αριθμό χρησιμοποιούμενων ΚΠ, μετράται επίσης ο πραγματικός χρόνος εκτέλεσης και παρουσιάζεται στον πίνακα 6.6. Φυσικά, εξαιτίας των παραδοχών που έγιναν θα υπάρχει μια απόκλιση μεταξύ των πραγματικών και των υπολογισμένων τιμών. Εάν η απόκλιση αυτή είναι μεγάλη τότε ο βέλτιστος αριθμός ΚΠ για την εκτέλεση της εργασίας δεν μπορεί να εξαχθεί βάσει των υπολογισμένων τιμών.

Πίνακας 6.5: Υπολογισμένος και πραγματικός χρόνος εκτέλεσης για διαφορετικό αριθμό πρακτόρων

Αρ. ΚΠ	Μέγιστος αριθμός στόχων (n)	Χρόνος μετανάστευσης (ms)	Χρόνος εκτέλεσης (ms)	Χρόνος δημιουργίας (ms)	Υπολογισμένος χρόνος (ms)	Πραγματικός χρόνος (ms)
1	10	$10 \times 1450 + 1450$	$10 \times 20 + 80$	40	16270	16938
2	5	$5 \times 1450 + 1450$	$5 \times 20 + 80$	310	9190	9228
3	4	$4 \times 1450 + 1450$	$4 \times 20 + 80$	600	8010	7455
4	3	$3 \times 1450 + 1450$	$3 \times 20 + 80$	740	6680	6397
5	2	$2 \times 1450 + 1450$	$2 \times 20 + 80$	1200	5670	5504
6	2	$2 \times 1450 + 1450$	$2 \times 20 + 80$	1450	5920	5418
7	2	$2 \times 1450 + 1450$	$2 \times 20 + 80$	2400	6870	5743
8	2	$2 \times 1450 + 1450$	$2 \times 20 + 80$	2900	7370	6467
9	2	$2 \times 1450 + 1450$	$2 \times 20 + 80$	3400	7870	6934
10	1	$1 \times 1450 + 1450$	$1 \times 20 + 80$	5100	8100	7630

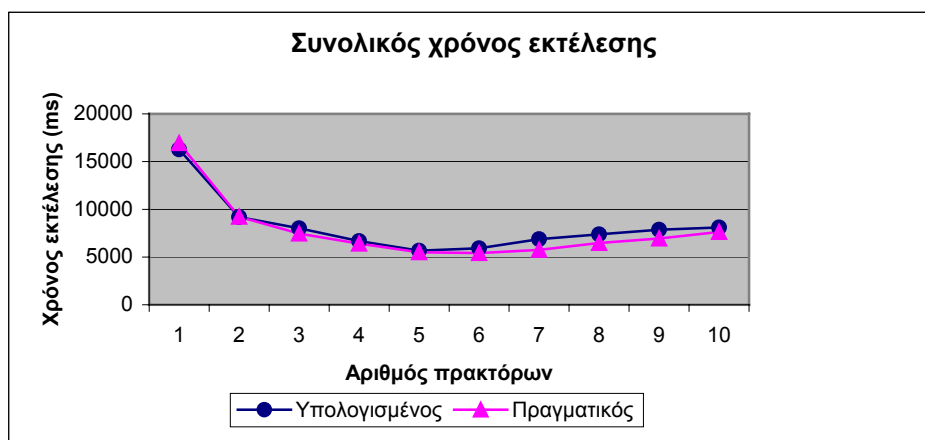
Ο υπολογισμένος και ο πραγματικός χρόνος εκτέλεσης για διαφορετικό αριθμό πρακτόρων παρουσιάζονται στο διάγραμμα 6.1

Όπως είχε προβλεφθεί, υπάρχει μια μικρή απόκλιση μεταξύ πραγματικών και υπολογισμένων τιμών, που προκαλείται από τις παραδοχές που έγιναν και την τυπική απόκλιση του μέσου χρόνου δημιουργίας των πρακτόρων. Ένα σημείο που αξίζει να τονιστεί είναι ότι βάσει των υπολογισμένων τιμών ο αριθμός των πρακτόρων που βελτιστοποιεί το χρόνο εκτέλεσης είναι πέντε και βάσει των πραγματικών τιμών ο αριθμός αυτός είναι έξι. Η διάφορα του πραγματικού χρόνου εκτέλεσης για 5 και 6 πράκτορες, είναι πολύ μικρή και βρίσκεται μέσα στην ακτίνα της τυπικής απόκλισης του χρόνου δημιουργίας. Ως εκ τούτου, το αποτέλεσμα αυτό θεωρείται φυσιολογικό και η εξαγωγή του βέλτιστου αριθμού πρακτόρων μπορεί να βασιστεί στις υπολογισμένες τιμές.

Η μικρή απόκλιση μεταξύ πραγματικών και υπολογισμένων τιμών, αποδεικνύει ότι οι παραδοχές που έγιναν για τη μετανάστευση και την εκτέλεση των πρακτόρων είναι σωστές. Το συμπέρασμα αυτό είναι πολύ σημαντικό για την εφαρμοσιμότητα της προτεινόμενης μεθοδολογίας. Αν το μέγεθος του πράκτορα δε μεταβάλλεται σημαντικά καθώς αυτός μεταναστεύει από το ένα μηχανήμα στο άλλο. Ο χρόνος μετανάστευσης εξαρτάται μόνο από την κατάσταση του υποκείμενου δικτύου και ο μέσος χρόνος μετανάστευσης των πρακτόρων μπορεί να θεωρηθεί σταθερός. Η ίδια θεώρηση μπορεί να γίνει και για το χρόνο εκτέλεσης του πράκτορα στα μηχανήματα-στόχους, ο οποίος εξαρτάται από τον τύπο του μηχανήματος και τον φόρτο εργασίας της ΚΜΕ. Στην

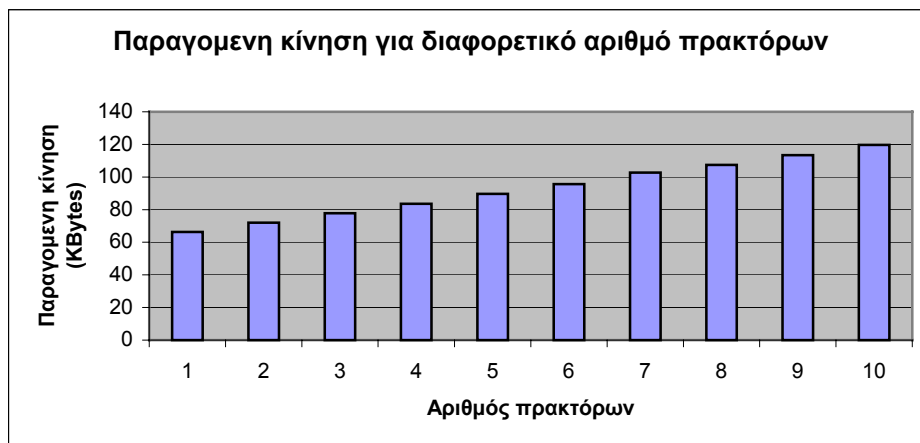
περίπτωση αυτή, η μετρούμενη παράμετρος για την εφαρμογή της προτεινόμενης μεθοδολογίας περιορίζεται μονάχα στο χρόνο δημιουργίας των πρακτόρων, διευκολύνοντας έτσι κατά πολύ τη διαδικασία υπολογισμού.

Βάσει των υπολογισμένων τιμών, ο χρόνος εκτέλεσης της εργασίας μειώνεται σημαντικά όσο ο αριθμός των πρακτόρων παραμένει μικρότερος ή ίσος του πέντε, που είναι ο βέλτιστος αριθμός πρακτόρων. Από το σημείο αυτό και μετά, ο χρόνος εκτέλεσης αυξάνεται. Αυτό μπορεί εξηγηθεί εύκολα με βάση το γεγονός ότι, για έναν αριθμό πρακτόρων μεταξύ του 5 και του 9, ο μέγιστος αριθμός στόχων που ανατίθενται σε έναν ή περισσότερους πράκτορες του συστήματος είναι ο ίδιος και ισούται με 2. Έτσι, οι χρόνοι μετανάστευσης και εκτέλεσης ενός πράκτορα με το μέγιστο αριθμό στόχων είναι οι ίδιοι, για έναν αριθμό πρακτόρων μεταξύ των δύο αυτών τιμών. Από την άλλη μεριά, καθώς αυξάνει ο αριθμός των χρησιμοποιούμενων πρακτόρων αυξάνει σημαντικά ο χρόνος δημιουργίας, με αποτέλεσμα μεγαλύτερους χρόνους εκτέλεσης της εργασίας. Στη περίπτωση που χρησιμοποιείται ένας πράκτορας για κάθε μηχανήμα-στόχο (δημιουργούνται 10 πράκτορες) ο χρόνος μετανάστευσης του πράκτορα με το μέγιστο αριθμό μηχανημάτων-στόχων ελαττώνεται. Ωστόσο, ο χρόνος δημιουργίας των πρακτόρων στο σημείο αυτό είναι πολύ μεγάλος. Έτσι, παρόλη τη βελτίωση του χρόνου μετανάστευσης, ο ολικός χρόνος εκτέλεσης είναι χειρότερος από την περίπτωση όπου ο αριθμός των πρακτόρων βρίσκεται μεταξύ των τιμών 5 και 9.



Γράφημα 6.1: Υπολογισμένος και Πραγματικός χρόνος εκτέλεσης για διαφορετικό αριθμό πρακτόρων

Τα πειραματικά αποτελέσματα δείχνουν ότι η χρήση του βέλτιστου αριθμού ΚΠ μπορεί να επιφέρει βελτίωση την απόδοση ενός ΣΠΠ έως και 75%. Η απόδοση της εφαρμογής αυξάνεται σημαντικά ακόμα και όταν ο αριθμός των πρακτόρων είναι μικρότερος του βέλτιστου. Η απόδοση της εφαρμογής με χρήση ενός πράκτορα, αυξάνεται σχεδόν κατά 40% ακόμα και όταν χρησιμοποιηθούν μονάχα δύο πράκτορες αντί ενός. Μια άλλη σημαντική παράμετρος στις κατανεμημένες εφαρμογές είναι η αξιοποίηση του εύρους ζώνης του δικτύου. Έτσι, χωρίς να θυσιάζεται μεγάλο μέρος της απόδοσης, μπορεί να χρησιμοποιηθεί ένας μικρότερος του βέλτιστου αριθμός πρακτόρων, προκειμένου να μειωθεί το ποσό της παραγόμενης κίνησης. Η απόφαση αυτή, όμως, πρέπει να παρθεί σύμφωνα με τους διαθέσιμους πόρους του δικτύου. Στο ιστόγραμμα 6.2, παρουσιάζεται η παραγόμενη κίνηση για διαφορετικό αριθμό πρακτόρων. Η επιπρόσθετη κίνηση είναι αποτέλεσμα του αρχικού μεγέθους του πράκτορα, που στην περίπτωση αυτή είναι 6.2 Kbytes. Η χρήση από μια εφαρμογή βασισμένη σε ΚΠ ενός υψηλού αριθμού πρακτόρων μεγάλου αρχικού μεγέθους, μπορεί να οδηγήσει σε σημαντική κατανάλωση πόρων δικτύου.



Ιστόγραμμα 6.2 Παραγόμενη κίνηση για διαφορετικό αριθμό πρακτόρων

6.5 Σύνοψη κεφαλαίου - Συμπεράσματα

Στο κεφαλαίο αυτό παρουσιάστηκαν τρεις διαφορετικοί τρόποι (μηχανισμοί) βελτίωσης της απόδοσης μιας εφαρμογής στηριγμένης στην ΤΚΠ. Ακολουθήθηκαν δυο διαφορετικές προσεγγίσεις του προβλήματος. Στη πρώτη προσέγγιση η βελτίωση επιτυγχάνεται με ανεύρεση και ανάθεση στον πράκτορα της βέλτιστης διαδρομής, με βάση μια παράμετρο που επηρεάζει την απόδοση του. Στην δεύτερη προσέγγιση η βελτίωση είναι αποτέλεσμα της κατάτμησης μιας εργασίας σε πολλούς πράκτορες, οι οποίοι δρουν ταυτόχρονα για την επίτευξη της.

Οι παράμετροι που επηρεάζουν την απόδοση ενός ΚΠ είναι η κατάσταση του υποκείμενου δικτύου και η κατάσταση των μηχανημάτων-στόχων. Για την ανεύρεση της βέλτιστης διαδρομής με βάση τις παραπάνω παραμέτρους, υλοποιήθηκαν και αξιολογήθηκαν πειραματικά δυο αλγόριθμοι. Κάθε αλγόριθμος υπολογίζει την βέλτιστη διαδρομή με βάση μια από τις δυο παραπάνω παραμέτρους, επιτρέποντας την διερεύνηση της επίδρασης στην απόδοση της κάθε παραμέτρου ξεχωριστά.

Ο πρώτος αλγόριθμος υπολογίζει την βέλτιστη διαδρομή, που πρέπει να ακολουθήσει ένα πράκτορας, με βάση τον φόρτο εργασίας της ΚΜΕ των μηχανημάτων-στόχων. Η αρχική μελέτη απέδειξε ότι ένας αλγόριθμος στηριγμένος μόνο στις τρέχουσες τιμές της ΚΜΕ δεν είναι λειτουργικός. Έτσι, ο αλγόριθμος που τελικά προτάθηκε χρησιμοποιεί και τις παρελθούσες τιμές της ΚΜΕ. Το πειραματικό στάδιο απέδειξε ότι με χρήση του αλγόριθμου μπορεί να επιτευχθεί μια αύξηση της απόδοσης του συστήματος κατά 10%. Η αύξηση αυτή όμως επιτυγχάνεται μόνο υπό ορισμένες προϋποθέσεις, που σχετίζονται με την διάρκεια του εκτελούμενου έργου. Γενικά μπορεί να λεχθεί ότι, η κατάσταση της ΚΜΕ των μηχανημάτων στόχων έχει μια σχετικά μικρή επίδραση στην απόδοση μιας εφαρμογής ΚΠ.

Ο δεύτερος αλγόριθμος υπολογίζει την βέλτιστη διαδρομή με βάση την καθυστέρηση μετάδοσης των συνδέσεων. Στη περίπτωση αυτή το πρόβλημα αντιμετωπίζεται ως ένα πρόβλημα γράφου με βάρη. Ο προτεινόμενος αλγόριθμος είναι μια προσέγγιση του TSP. Τα πειραματικά αποτελέσματα απέδειξαν ότι η χρήση του αλγόριθμου αυξάνει την απόδοση του συστήματος κατά 45%. Η κατάσταση του υποκείμενου δίκτυο επηρεάζει σε

πολύ μεγαλύτερο βαθμό την απόδοση μια εφαρμογής ΚΠ, από ότι η κατάσταση των μηχανημάτων στόχων.

Οι μέθοδοι ανεύρεσης της βέλτιστης διαδρομής με παρακολούθηση παραμέτρων έχουν το μειονέκτημα της δημιουργίας προσθετικής κίνησης στο δίκτυο, ως αποτέλεσμα της συνεχούς παρακολούθησης.

Με τους παραπάνω μηχανισμούς επιχειρείται η βελτίωση της απόδοσης με παρακολούθηση κάποιας παραμέτρου και ανεύρεση της βελτίωσης διαδρομής. Στην δεύτερη προσέγγιση του προβλήματος, η βελτίωση επιτυγχάνεται με κατάτμηση της εργασίας σε πολλούς πράκτορες. Έτσι το πρόβλημα ανάγεται στην ανεύρεση του αριθμού των πρακτόρων που βελτιστοποιούν την απόδοση. Προτείνεται λοιπόν ένας αλγόριθμος ανεύρεσης του βέλτιστου αριθμού πρακτόρων, που πρέπει να χρησιμοποιηθούν από ένα ΣΠΠ. Τα πειραματικά αποτελέσματα σε αυτή την περίπτωση απέδειξαν ότι η χρήση του βέλτιστου αριθμού πρακτόρων έχει ως αποτέλεσμα την αύξηση της απόδοσης έως και 75%. Το τίμημα για αυτήν την βελτίωση, είναι η αύξηση του εύρους ζώνη του δικτύου, που χρησιμοποιείται από την εφαρμογή καθώς ο αριθμός των πρακτόρων αυξάνει.

Και οι τρεις προτεινόμενοι μηχανισμοί επιφέρουν κάποια βελτίωση στην απόδοση μιας εφαρμογής ΚΠ. Μεγαλύτερη βελτίωση επιτυγχάνεται με την κατάτμηση εργασίας σε πολλούς πράκτορες. Η λύση αυτή αν και είναι η καλύτερη δεν είναι εφαρμόσιμη σε όλες τις περιπτώσεις, διότι απαιτεί αλλαγή στον σχεδιασμό της εφαρμογής και τον τρόπο λειτουργία των πρακτόρων. Η κατάτμηση μιας διεργασίας σε πολλούς πράκτορες δεν είναι δυνατή μόνο όταν ο ΚΠ πρέπει να έχει πλήρη γνώση όλης της διεργασίας, που σε μια κατανεμημένη εφαρμογή είναι σπάνιο. Γενικά, η αλλαγή σχεδιασμού για κατάτμηση μιας εργασίας, είναι δυνατή στο 99% των εφαρμογών. Το μειονέκτημα της αύξησης του απαιτούμενου εύρους ζώνης, με την αύξηση των χρησιμοποιούμενων πρακτόρων μπορεί να ξεπεραστεί χρησιμοποιώντας έναν αριθμό πρακτόρων μικρότερο από το βέλτιστο. Οι πειραματικές μετρήσεις έδειξαν ότι η κατάτμηση μιας εργασίας σε δυο μόνο πράκτορες επιφέρει μια αύξηση της απόδοσης κατά 40%.

6.6 Βιβλιογραφικές Αναφορές

[Dag95] J. Dag, R. Renesse and F. B. Schneider, “Operating systems support for mobile agents” *Proc. Fifth IEEE Workshop on Hot Topics in Operating Systems*, Wahsington, USA, May 1995, pp. 42- 45.

[Gray96A] R. S. Gray, “Gray.Agent Tcl: A flexible and secure mobile agent system”, *Proc. 4th Annual Tcl/Tk Workshop*, California, July 1996. pp 9-23.

[Sacerdoti75] E.Sacerdoti, “The nonlinear nature of plans”, *Proc. Fourth International Joint Conference on Artificial Intelligence*, 1975.

[Etioni92] O. Etioni, S. Hanks, D. Draper, N. Lesh, and M. Williamson, “An approach to planing with incomplete information” *Proc 3rd International Conference on Principles of Knowledge Representation and Reasoning*, 1992.

[Fikes72] R.E. Fikes, P.E. Hart, and N.J. Nilsson. “Learning and executing generalized robot plans”, *Artificial Intelligence*, 1972.

[Sacerdoti77] E D. Sacerdoti, “A Structure for Plans and Behavior”, *American Elsevier*, New York, 1977.

[Fikes71] R. E. Fikes and N. J. Nilsson, “Strips: A new approach to the application of theorem proving to problem solving”, *Artificial Intelligence* 2, 1971.

[TSP] N. L. Biggs, E. K. LLoyd, and R. J. Wilson, “Graph Theory 1736-1936”, Clarendon Press, Oxford, 1976.

[Moizumi98] K. Moizumi, “The mobile agent planning problem”, PhD thesis, Thayer School of Engineering, Dartmouth College, November 1998.

[Rubinstein99] M.G. Rubinstein and O. C. M. B. Duarte, “Analyzing Mobile Agent Scalability in Network Management”, *Proc IEEE Latin American Network Operations and Management Symposium (LANOMS'99)*, Rio de Janeiro, Brazil, December 1999.

[Kotsilieris02] T. Kotsilieris, A. Michalas, S. Kalogeropoulos, G. Karetsos , M. Sidi and V. Loumos,, “Performance optimization of Network Management Applications based on

Mobile Agents”, *Accepted for publication at the Informatik Forum Journal Special Issue on Mobile Agent Technology*, 2002.

[Adventnet] Διαθέσιμο στην ηλεκτρονική διεύθυνση: www.adventnet.com

[JNI] Η πληροφορία είναι διαθέσιμη στη ηλεκτρονική διεύθυνση: <http://java.sun.com/products/jdk/1.2/docs/guide/jni>.

[Redhat] Διαθέσιμο στην ηλεκτρονική διεύθυνση www.redhat.com.

[Sedgewick84] R. Sedgewick, “Algorithms”, Addison-Wesley publishing company, August 1984, ISBN 0201066726.

[Prim57] R.C. Prim, “Shortest connection networks and some generalizations”, *Bell Systems Technology Journal*, Vol. 36, 1957, pp 1389-1401.

[Dijkstra59] E. W. Dijkstra, “A note on two problems in connexion with graphs,” *Numerische Mathematik*, 1959.

[Kalogeropoulos03] S. Kalogeropoulos, T. Kotsilieris, M. Sidi, H. Gazit and J. Tuch, “A proposed methodology to improve the performance of agent-based applications through the identification of the optimum visited sequence”, *IAESTED International Conference on Artificial Intelligence and Applications (AIA 2003)*, Austria, February 2003.

[Viola] Η πληροφορία είναι διαθέσιμη στη ηλεκτρονική διεύθυνση: <http://www.violanetworks.com/>

[Iperf] Διαθέσιμο στην ηλεκτρονική διεύθυνση <http://dast.nlanr.net/Projects/Iperf/>

[Muthukrishnan99] M. Muthukrishnan and T. B. Suresh, “A Multi-Agent Approach to Distributed Computing”, *Autonomous Agents*, Seattle Washington USA, May 1999.

[Durfee89] E.H. Durfee, V.R. Lesser and D.D. Corkill, “Trends in Cooperative Distributed Problem Solving”, *IEEE Transactions on Knowledge and Data Engineering*, March 1989, pp. 63-83.

[Jennings98] N.R. Jennings, K. Sycara, and M. Wooldridge, “A Roadmap of Agent Research and Development”, *Autonomous Agents and Multi-Agent Systems Journal*, Vol. 1, Is. 1, 1998, pp. 7-38.

[Rus96] D. Rus, R. Gray and D. Kotz “Autonomous and Adaptive Agents that Gather Information”, *AAAI '96 International Workshop on intelligent Adaptive Agents*, August 1996.

[Kalogeropoulos02] S. Kalogeropoulos, G. Karetsos, A. Anagnostopoulos, T. Kotsilieris, A. Michalas and V. Loumos “A Methodology for Improving the Performance of Agent-Based Applications Through the Identification of the Optimal Number of Mobile Agents”, *Submitted for publication at the Autonomous Agents and Multi-Agent Systems*, 02.

[Oates95] T. Oates, M.V. Nagendra Prasad, V.R. Lesser and K.S. Decker, “A Distributed Problem Solving Approach to Cooperative Information Gathering”, *AAAI Spring Symposium*, Stanford CA, March 1995.

ΚΕΦΑΛΑΙΟ 7

ΣΥΜΠΕΡΑΣΜΑΤΑ

Το κεφάλαιο αυτό συνοψίζει τα βασικά συμπεράσματα της διατριβής. Παρουσιάζεται η συνεισφορά την εργασίας στην ερευνά σχετικά με την παροχή ΠοιΥπ στο διαδίκτυο και την ΤΚΠ. Επιπρόσθετα, παρατίθενται τα σημεία της διατριβής που αποτελούν έναυσμα για μελλοντική ερευνά.

7.1 Καινοτομικά στοιχεία της διατριβής

Η συνεισφορά της διατριβής εντοπίζεται σε δυο κύριους ερευνητικούς άξονες: (α) Την ανάπτυξη και αξιολόγηση συστημάτων παροχής ΠοιΥπ, σε IP δίκτυα, με χρήση της ΤΚΠ και (β) Την χρήση αξιολόγηση και βελτίωση της απόδοσης της ΤΚΠ, σε συστήματα διαχείρισης δικτύου.

Όσον αφορά την ανάπτυξη συστημάτων διαχείρισης ΠοιΥπ, η αρχιτεκτονική η οποία φαίνεται να υπερισχύει είναι αυτή των ΔΥ. Οι αρχιτεκτονική ολοκληρωμένων υπηρεσιών παρουσίασε δυσκολία επέκτασης σε μεγάλα δίκτυα. Από την άλλη μεριά, η αρχιτεκτονική ΔΥ παρέχει ένα μεγαλύτερο βαθμό ευελιξίας στη διαχείριση ενός δικτύου, με εγγυήσεις στην ΠοιΥπ και έχει κερδίσει την γενική αποδοχή της κοινότητας του διαδικτύου, ως η πλέον κατάλληλη για τον σκοπό αυτό. Το κεντροποιημένο μοντέλο διαχείρισης (πελάτη/εξυπηρετητή) παρουσίασε μια σειρά από προβλήματα με την αύξηση του μεγέθους των δικτύων, με κυριότερα την ακατάλληλη χρήσης του εύρους ζώνης του δικτύου και την δυσκολία κλιμάκωσης σε μεγαλύτερα δίκτυα. Η ΤΚΠ είναι μια εναλλακτική λύση στην υλοποίηση συστημάτων διαχείρισης δικτύου, η οποία παρουσιάζει πολλά πλεονεκτήματα έναντι του μοντέλου πελάτη/εξυπηρετητή.

Στο πλαίσιο της διατριβής σχεδιάστηκε, υλοποιήθηκε και αξιολογήθηκε ένα ολοκληρωμένο σύστημα παροχής και παρακολούθησης ΠοιΥπ, με βάση την αρχιτεκτονική ΔΥ και χρήση της ΤΚΠ. Η προτεινόμενη αρχιτεκτονική δίνει λύση σε μια σειρά από προβλήματα σχεδίασης υλοποίησης ενός τέτοιου συστήματος.

Το πρώτο και σημαντικότερο πρόβλημα είναι η διαχείριση ενός δικτύου που αποτελείται από διαφορετικού τύπου δρομολογητές (Cisco και Linux), αφού δυστυχώς δεν υπάρχει μια κοινή MIB ΔΥ που να υποστηρίζεται και από του δυο τύπους. Το πρόβλημα αυτό ξεπεράστηκε με την δημιουργία ενός υποσυστήματος εφαρμογής, το οποίο αποκρύπτει τις λεπτομέρειες διαχείρισης των δρομολογητών από τους κινητούς πράκτορες του συστήματος. Η λειτουργία του υποσυστήματος αυτού στηρίζεται στην ύπαρξη μιας σειράς στατικών πρακτόρων που διαθέτουν όλη την απαραίτητη πληροφορία για την παρακολούθηση και διάρθρωση των δρομολογητών.

Ένα δεύτερο πρόβλημα που αντιμετωπίζουν όλες οι εφαρμογές κινητών πρακτόρων είναι η εξάρτηση της λειτουργία τους από μια και μόνο ΠΚΠ. Λύση στο πρόβλημα αυτό δόθηκε, με την δημιουργία μια κοινής ΔΠΕ ΠΚΠ για τις δυο ΠΚΠ που συνήθως χρησιμοποιούνται (Grasshopper και Voyager). Η χρήση της κοινής ΔΠΕ επιτρέπει την εκτέλεση των πρακτόρων του συστήματος και στις δυο πλατφόρμες με ελάχιστη μεταβολή στον κώδικα τους. Η σύγκριση του χρόνου μετανάστευσης των δυο πλατφόρμων απέδειξε ότι η πλατφόρμα Voyager υπερέχει σε αυτόν τον τομέα έναντι της Grasshopper. Όμως η δημιουργία πρακτόρων με χρήση της πλατφόρμας Grasshopper είναι αρκετά πιο εύκολη. Επίσης, η εν λόγω πλατφόρμα είναι η μόνη που ακολουθεί όλα τα πρότυπα του Foundation for Intelligent and Physical Agents (FIPA), καθώς επίσης και πρότυπο MASIF το οποίο επιτρέπει την αλληλεπίδραση μεταξύ πρακτόρων διαφορετικών πλατφόρμων (δεν υποστηρίζεται από την Voyager). Η ΔΠΕ μπορεί εύκολα να επεκταθεί ώστε να περιλαμβάνει και άλλες ΠΚΠ.

Ιδιαίτερη σημασία σε μια εφαρμογή ΚΠ έχει να διατηρηθεί το μέγεθος των ΚΠ μικρό, ώστε να διευκολύνεται η μετανάστευση. Μικρότεροι χρόνοι μετανάστευσης οδηγούν σε καλύτερη απόδοση της εφαρμογής. Όλες οι λειτουργίες διάρθρωσης και παρακολούθησης πραγματοποιούνται με χρήση κινητών και στατικών πρακτόρων. Οι ΚΠ μεταφέρουν τις εντολές διάρθρωσης και παρακολούθησης στους κόμβους του δικτύου. Επίσης, μεταφέρουν τα αποτελέσματα της εκτέλεσης των εντολών στην κεντρική μονάδα του συστήματος, μέσω της οποίας παρουσιάζονται στον χρήστη. Οι στατικοί πράκτορες που βρίσκονται στους κόμβους του δικτύου (δρομολογητές) δρουν ως διεπαφές μεταξύ των κινητών πρακτόρων και του λογισμικού του κόμβου. Δηλαδή,

όλες οι αλληλεπιδράσεις μεταξύ των ΚΠ και των κόμβων γίνονται διάμεσου των στατικών πρακτόρων. Εναλλακτικά, οι ΚΠ θα μπορούσαν να μεταφέρουν την απαραίτητη πληροφορία για την αλληλεπίδραση με το λογισμικό των κόμβων. Αυτή η προσέγγιση θα οδηγούσε σε κινητούς πράκτορες με μεγάλο μέγεθος, που θα είχε αρνητική επίδραση στον χρόνο μετανάστευσης τους. Με την χρήση των στατικών πρακτόρων εξασφαλίστηκε το μικρό μέγεθος των ΚΠ.

Η αξιολόγηση της εφαρμογής έχει ιδιαίτερη βαρύτητα διότι βασίζεται σε πραγματικές μετρήσεις. Όσον αφορά την κατανομή του εύρους ζώνης, τα πειραματικά αποτελέσματα έδειξαν ότι το σύστημα κατανέμει το εύρος ζώνης του δικτύου στις διαφορετικές ροές κίνησης, παρουσιάζοντας αμελητέες αποκλίσεις από τις επιθυμητές τιμές. Η συλλογή των αποτελεσμάτων έγινε με χρήση του υποσυστήματος παρακολούθησης. Η συμφωνία των τιμών αυτών με την διάρθρωση του δικτύου αποδεικνύει ότι και το σύστημα παρακολούθησης δεν παρουσιάζει σφάλματα στον τρόπο λειτουργίας. Ένα σημείο το οποίο αξίζει να αναφέρουμε, είναι ότι δημιουργούνται μικρές ακμές στο εύρος ζώνης που καταλαμβάνει μια κίνηση υψηλής προτεραιότητας, όταν αυξομειώνεται απότομα το εύρος ζώνης μιας κίνησης χαμηλής προτεραιότητας. Η μόνη λογική εξήγηση για το γεγονός αυτό, είναι η προς στιγμήν αδράνεια του δικτύου να προσαρμοστεί στα νέα δεδομένα της κίνησης. Το προτεινόμενο σύστημα διαχείρισης είναι επίσης σε θέση να προσφέρει εγγυήσεις για την καθυστέρηση μετάδοσης δεδομένων, που ανήκουν σε διαφορετικές κλάσεις υπηρεσίας. Με κατάλληλη διάρθρωση των δρομολογητών του δικτύου, η παραγόμενη ΒΕ κίνηση παρουσίασε μια καθυστέρηση μετάδοσης της τάξης των 0,5 sec, ενώ ΕΦ κίνηση που μεταδόθηκε ταυτόχρονα παρουσίασε καθυστέρηση της τάξης των 50 msec. Και σε αυτήν την περίπτωση το σύστημα παρακολούθησης έδωσε αποτελέσματα σύμφωνα με τις αναμενόμενες τιμές, αποδεικνύοντας την σωστή λειτουργία του. Μπορούμε λοιπόν να εξάγουμε ως γενικό συμπέρασμα, ότι η αρχιτεκτονική ΔΥ είναι σε θέση να προσφέρει εγγυήσεις στη ΠοιΥπ που λαμβάνει η κίνηση σε ένα IP δίκτυο.

Στο πέμπτο κεφάλαιο παρουσιάστηκε μια σύγκριση της ΤΚΠ με το μοντέλο πελάτη/εξυπηρετητή. Δηλαδή, συγκρίθηκε η κεντροποιημένη διαχείριση που εκφράζεται από το μοντέλο πελάτη/εξυπηρετητή, με την αποκεντροποιημένη διαχείριση που

εκφράζεται από την ΤΚΠ. Η σύγκριση βασίστηκε στο χρόνο εκτέλεσης που απαιτεί κάθε μια από τις δυο προσεγγίσεις για την εκτέλεση μιας σειράς διεργασιών διαφορετικής πολυπλοκότητας. Η διαφορετική πολυπλοκότητα του προς εκτέλεση έργου, επέτρεψε την διερεύνηση της επίδραση που έχει στην απόδοση των εφαρμογών η λογική που εμπεριέχεται στο έργο που καλούνται να εκτελέσουν. Ένας άλλος παράγοντας ο οποίος ερευνήθηκε ήταν ο τρόπος με το οποίο χρησιμοποιούν το εύρος ζώνης του δικτύου οι εφαρμογές. Ένα πειραματικό IP δίκτυο δρομολογητών Linux, χρησιμοποιήθηκε για την διεξαγωγή των πειραμάτων.

Με βάση τα πειραματικά αποτελέσματα, η εφαρμογή πελάτη/εξυπηρετητή παρουσιάζει πολύ καλύτερους χρόνους εκτέλεσης, όταν το προς εκτέλεση έργο είναι αρκετά απλό. Η εφαρμογή πελάτη/εξυπηρετητή παρουσιάζει σε αυτή την περίπτωση χρόνους εκτέλεσης που είναι έως και 82% καλύτεροι από αυτούς της αντίστοιχης εφαρμογής με βάση την ΤΚΠ. Η διάφορα αυτή μειώνεται στο 58% όταν η πολυπλοκότητα του προς εκτέλεση έργου αυξάνει. Η εφαρμογή με βάση την ΤΚΠ παρουσιάζει χρόνους εκτέλεσης κατά ένα ποσοστό 20% καλύτερους από αυτούς της εφαρμογής με βάση το μοντέλο πελάτη/εξυπηρετητή, όταν το προς εκτέλεση έργο αυξάνει αρκετά την πολυπλοκότητα του, εμπεριέχοντας λογική για διόρθωση λαθών. Σε αυτήν την τελευταία περίπτωση, το εύρος ζώνης που χρησιμοποιείται από την εφαρμογή ΚΠ είναι το 1/3 αυτού που χρησιμοποιείται από την εφαρμογή πελάτη/εξυπηρετητή.

Η σύγκριση των δυο μοντέλων απέδειξε ότι η απάντηση στην ερώτηση πιο από τα δυο είναι καταλληλότερο για την δημιουργία μιας κατανεμημένη εφαρμογής, εξαρτάται από την πολυπλοκότητα του προς εκτέλεση έργου. Όταν η προς εκτέλεση διεργασία δεν περιέχει κάποιο είδος λογικής, αλλά είναι μια απλή εκτέλεση μιας σειράς εντολών, τότε καταλληλότερο μοντέλο είναι αυτό του πελάτη/εξυπηρετητή. Στο κεντροποιημένο αυτό μοντέλο όλη η λογική της εφαρμογής βρίσκεται συγκεντρωμένη σε ένα κεντρικό μηχάνημα που ελέγχει όλη την διαδικασία. Όσο η επικοινωνία με τα απομακρυσμένα μηχανήματα είναι περιορισμένη στην απλή ανάθεση των εντολών, ο χρόνος εκτέλεσης που προκύπτει είναι μικρός. Όταν το προς εκτέλεση έργο περιλαμβάνει την λήψη κάποιων αποφάσεων για την περαιτέρω εκτέλεση του, τότε η επικοινωνία με το κεντρικό μηχάνημα αυξάνεται οδηγώντας σε μεγαλύτερους χρόνους εκτέλεσης. Η αύξηση της

απομακρυσμένης επικοινωνίας οδηγεί επίσης σε ανάλογη αύξηση του ποσού κίνησης που παράγεται από την εφαρμογή στο δίκτυο.

Όταν η προς εκτέλεση διεργασία περιέχει κάποιο είδος λογικής, που αυξάνει την πολυπλοκότητα της, τότε καταλληλότερο μοντέλο είναι αυτό των ΚΠ. Όσο το έργο είναι σχετικά απλό, ο χρόνος μετανάστευσης του πράκτορα έχει αρνητική επίδραση στο χρόνο εκτέλεσης της εφαρμογής. Το μειονέκτημα αυτό για την εκτέλεση απλών διεργασιών σε απομακρυσμένα μηχανήματα μετατρέπεται σε πλεονέκτημα όταν η πολυπλοκότητα αυξάνει. Τότε, ο πράκτορας μεταφέροντας στον κώδικα του όλη την απαραίτητη λογική για την εκτέλεση της διεργασίας δεν είναι αναγκασμένος να επικοινωνεί με κάποια κεντρική μονάδα, όπως στην περίπτωση του πελάτη/εξυπηρετητή. Έτσι εξηγείται η καλύτερη απόδοση του μοντέλου αυτού σε πολύπλοκες διεργασίες και η καλύτερη χρήση του εύρους ζώνης του δικτύου.

Στο έκτο κεφαλαίο παρουσιάστηκαν τρεις διαφορετικές μέθοδοι (μηχανισμοί) βελτίωσης της απόδοσης συστημάτων διαχειρίσεις δικτύου υλοποιημένων με χρήση της ΤΚΠ. Ακολουθήθηκαν δυο διαφορετικές προσεγγίσεις του προβλήματος. Στη πρώτη προσέγγιση η βελτίωση επιτυγχάνεται με ανεύρεση και ανάθεση στον πράκτορα της βέλτιστης διαδρομής, με βάση μια παράμετρο που επηρεάζει την απόδοση του. Στην δεύτερη προσέγγιση η βελτίωση είναι αποτέλεσμα της κατάτμησης μιας εργασίας σε πολλούς πράκτορες οι οποίοι δρουν ταυτόχρονα για την επίτευξη της.

Αναπτύχθηκαν δυο αλγόριθμοι ανεύρεσης της βέλτιστης διαδρομής. Κάθε αλγόριθμος υπολογίζει την βέλτιστη διαδρομή βάσει μιας μόνο παραμέτρου που επηρεάζει την απόδοση ενός ΚΠ, επιτρέποντας έτσι την διερεύνηση της επίδρασης κάθε παραμέτρου στην απόδοση, ξεχωριστά.

Ο πρώτος αλγόριθμος υπολογίζει την βέλτιστη διαδρομή, που πρέπει να ακολουθήσει ένα πράκτορας, με βάση τον φόρτο εργασίας της ΚΜΕ των μηχανημάτων-στόχων. Η αρχική μελέτη απέδειξε ότι ένας αλγόριθμος βασισμένος μόνο στις τρέχουσες τιμές της ΚΜΕ δεν ήταν λειτουργικός. Έτσι, ο αλγόριθμος που τελικά προτάθηκε χρησιμοποιεί και τις παρελθούσες τιμές της ΚΜΕ. Το πειραματικό στάδιο απέδειξε ότι με χρήση του αλγόριθμου μπορεί να επιτευχθεί μια αύξηση της απόδοσης του συστήματος κατά 10%. Η αύξηση αυτή όμως επιτυγχάνεται μόνο υπό κάποιες προϋποθέσεις, που σχετίζονται με

την διάρκεια του εκτελούμενου έργου. Γενικά μπορεί να λεχθεί ότι, η κατάσταση της ΚΜΕ των μηχανημάτων στόχων έχει μια μικρή επίδραση στην απόδοση μιας εφαρμογής ΚΠ.

Ο δεύτερος αλγόριθμος υπολογίζει την βέλτιστη διαδρομή με βάση την καθυστέρηση μετάδοσης των συνδέσεων. Στη περίπτωση αυτή το πρόβλημα αντιμετωπίζεται ως ένα πρόβλημα γράφου με βάρη. Ο προτεινόμενος αλγόριθμος είναι μια προσέγγιση του TSP. Τα πειραματικά αποτελέσματα απέδειξαν ότι η χρήση του αλγόριθμου αυξάνει την απόδοση του συστήματος έως και 45%. Η κατάσταση του υποκείμενου δίκτυο επηρεάζει σε πολύ μεγαλύτερο βαθμό την απόδοση μια εφαρμογής ΚΠ από ότι η κατάσταση των μηχανημάτων στόχων.

Οι μέθοδοι ανεύρεσης της βέλτιστης διαδρομής με παρακολούθηση παραμέτρων έχουν το μειονέκτημα της δημιουργίας προσθετικής κίνησης στο δίκτυο ως αποτέλεσμα της συνεχούς παρακολούθησης.

Με τους παραπάνω μηχανισμούς επιχειρείται η βελτίωση της απόδοσης με παρακολούθηση κάποιας παραμέτρου και ανεύρεση της βελτίωσης διαδρομής. Στην δεύτερη προσέγγιση του προβλήματος, η βελτίωση επιτυγχάνεται με κατάτμηση της εργασίας σε πολλούς πράκτορες. Προτείνεται μια μεθοδολογία ανεύρεσης του αριθμού πρακτόρων, που βελτιστοποιούν την απόδοση ενός ΣΠΠ. Τα πειραματικά αποτελέσματα σε αυτή την περίπτωση απέδειξαν ότι η χρήση του βέλτιστου αριθμού πρακτόρων έχει ως αποτέλεσμα την αύξηση της απόδοσης έως και 75%. Το τίμημα για αυτήν την αύξηση είναι η αύξηση του εύρους ζώνης του δικτύου, που χρησιμοποιείται από την εφαρμογή καθώς ο αριθμός των πρακτόρων αυξάνει.

Και οι τρεις προτεινόμενοι μηχανισμοί επιτυγχάνουν την βελτίωση της απόδοσης μιας εφαρμογής ΚΠ. Μεγαλύτερη βελτίωση, επιτυγχάνεται με την κατάτμηση εργασίας σε πολλούς πράκτορες. Η λύση αυτή αν και είναι η καλύτερη, δεν είναι εφαρμόσιμη σε όλες τις περιπτώσεις, διότι απαιτεί αλλαγή στον σχεδιασμό της εφαρμογής και τον τρόπο λειτουργία των πρακτόρων. Η κατάτμηση μιας εργασίας σε πολλούς πράκτορες δεν είναι δυνατή μόνο όταν ο ΚΠ πρέπει να έχει πλήρη γνώση όλης της διεργασίας, που σε μια κατανομημένη εφαρμογή είναι σπάνιο.. Το μειονέκτημα της αύξησης του απαιτούμενου εύρους ζώνης μπορεί να ξεπεραστεί χρησιμοποιώντας έναν αριθμό πρακτόρων μικρότερο

από το βέλτιστο. Οι πειραματικές μετρήσεις έδειξαν ότι η κατάτμηση μιας εργασίας σε δυο μόνο πράκτορες επιφέρει μια αύξηση της απόδοσης κατά 40%.

7.2 Θέματα προς μελλοντική διερεύνηση

Τα αποτελέσματα της διατριβής αναφορικά με την επαλήθευση των μηχανισμών παροχής ΠοιΥπ, την σύγκριση της ΤΚΠ με το μοντέλο πελάτη/εξυπηρετητή και τις τεχνικές βελτίωσης της απόδοσης εφαρμογών ΚΠ, είναι αυτοτελή και ολοκληρωμένα. Παρόλα αυτά το ερευνητικό εύρος της εργασίας και τα αποτελέσματα αυτής, μπορούν να αποτελέσουν το έναυσμα για περαιτέρω διερεύνηση. Τα σημαντικότερα θέματα προς μελλοντική διερεύνηση παρουσιάζονται στην συνέχεια.

Τα πειραματικά αποτελέσματα της διατριβής επαλήθευσαν την δυνατότητα της αρχιτεκτονικής ΔΥ να παράσχει εγγυήσεις στην ΠοιΥπ, που λαμβάνει η κίνηση σε ένα IP δίκτυο. Όμως, το πειραματικό IP δίκτυο που χρησιμοποιήθηκε αποτελεί μόνο ένα διαχειριστικό πεδίο (domain), με περιορισμένο αριθμό κόμβων (δρομολογητών). Όπως είναι γνωστό, η επικοινωνία στο διαδίκτυο επιτυγχάνεται με την μεταφορά πακέτων μέσω πολλαπλών διαχειριστικών πεδίων που αποτελούνται από μεγάλο αριθμό δρομολογητών. Ένα σημαντικό θέμα προς μελλοντική διερεύνηση, είναι η επαλήθευση των πειραματικών αποτελεσμάτων σε ένα IP δίκτυο αποτελούμενο από πολλαπλά διαχειριστικά πεδία. Δυστυχώς ένα τέτοιο εγχείρημα απαιτεί αρκετό δικτυακό εξοπλισμό. Επομένως είναι προτιμότερο να πραγματοποιηθεί με χρήση κάποιου εξειδικευμένου εργαλείου προσομοίωσης (π.χ. το Network Simulator- NS), καθώς πειράματα με πραγματικό εξοπλισμό είναι δύσκολο να πραγματοποιηθούν σε τέτοιες κλίμακες.

Η εφαρμογή των ΔΥ σε ένα δίκτυο το οποίο αποτελείται από πολλά διαφορετικά διαχειριστικά πεδία, όπως το διαδίκτυο, δεν είναι απλή υπόθεση. Πρέπει πρώτα να δοθεί λύση σε μια σειρά από προβλήματα τα οποία σχετίζονται με την αντιμετώπιση της κίνησης καθώς αυτή διέρχεται από διαφορετικά διαχειριστικά πεδία. Όπως ήδη έχουμε αναφέρει, δεν έχουν ακόμη τυποποιηθεί οι διάφορες υπηρεσίες που μπορεί να προσφέρει η αρχιτεκτονική ΔΥ. Έτσι, η ΠοιΥπ που λαμβάνει μια κίνηση που ανήκει σε μια συγκεκριμένη υπηρεσία, μπορεί να είναι διαφορετική καθώς διέρχεται από διάφορα διαχειριστικά πεδία. Ακόμη, τα διαχειριστικά πεδία μπορεί να μαρκάρουν διαφορετικά την πακέτα που ανήκουν στην ίδια κλάση υπηρεσίας. Υπαρχει ακόμη και η περίπτωση, οι απαραίτητοι δικτυακοί πόροι για την από άκρο-σε-άκρο (end-to-end) μετάδοση της

κίνησης μέσω πολλαπλών διαχειριστικών πεδίων, να μην είναι διαθέσιμοι. Όλα τα παραπάνω ζητήματα απασχολούν σήμερα την κοινότητα του διαδικτύου και αποτελούν σημαντικά θέματα προς μελλοντική διερεύνηση. Το προτεινόμενο σύστημα παροχής ΠοιΥπ μπορεί εύκολα να αναβαθμίσει τις λειτουργίες του, λόγω του ότι στηρίζεται στην ΤΚΠ. Μια μελλοντική υλοποίηση είναι δυνατόν να συμπεριλαμβάνει μηχανισμούς έλεγχου και προσαρμογής της ΠοιΥπ, που λαμβάνει η κίνηση καθώς διέρχεται από διαφορετικά διαχειριστικά πεδία.

Στα πλαίσια της διατριβής παρουσιάστηκαν δυο αλγόριθμοι ανεύρεσης της βέλτιστης διαδρομής, που πρέπει να ακολουθήσει ένας ΚΠ, ο οποίος επισκέπτεται μια σειρά μηχανημάτων. Η δημιουργία δυο διαφορετικών αλγόριθμων έγινε με σκοπό την διερεύνηση της επίδρασης στη απόδοση ενός συστήματος ΚΠ, των παραμέτρων που την επηρεάζουν (φόρτος ΚΜΕ, καθυστέρηση μετάδοσης) χωριστά. Παρόλο που η κατάσταση των συνδέσεων ενός δίκτυο χαρακτηρίζεται από τρεις παραμέτρους (καθυστέρηση, απόκλιση καθυστέρησης και απώλειες), ο δεύτερος αλγόριθμος στηρίζεται μόνο σε μια από αυτές, την καθυστέρηση μετάδοσης. Η επιλογή αυτή έγινε με βάση την θεώρηση ότι η παρακολούθηση όλων των παραμέτρων ενός δικτύου και η ανάπτυξη ενός αλγόριθμου που θα υπολόγιζε την βέλτιστη διαδρομή βάσει όλων αυτών, θα οδηγούσε σε μια χρονοβόρα διαδικασία με αντίθετα από τα επιθυμητά αποτελέσματα. Το επόμενο βήμα λοιπόν, είναι η δημιουργία ενός αλγόριθμου ο οποίος θα υπολογίζει την βέλτιστη διαδρομή βάσει όλων των παραμέτρων οι οποίοι επηρεάζουν την απόδοση ενός ΚΠ. Η πειραματική αξιολόγηση αυτού του αλγόριθμου, θα αποδείξει εάν η παρακολούθηση και επεξεργασία όλων των παραμέτρων οδηγεί σε περαιτέρω βελτίωση της απόδοσης, ή σε μια χρονοβόρα διαδικασία με μείωση την απόδοσης.

Η τεχνική βελτίωσης την απόδοσης μιας εφαρμογής ΚΠ με τα καλύτερα αποτελέσματα είναι με διαμοιρασμό της εργασίας που είχε ανατεθεί σε έναν πράκτορα σε περισσότερους. Η πειραματική μελέτη απέδειξε ότι η αύξηση του αριθμού των πρακτόρων οδηγεί σε μια μεγάλη αύξηση του χρόνου δημιουργίας τους. Εάν με κάποιο τρόπο μπορούσε να μειωθεί αυτός ο χρόνος, τότε θα προέκυπτε ένας μεγαλύτερος βέλτιστος αριθμός πρακτόρων με μεγαλύτερη αύξηση της απόδοσης (μείωση των μηχανημάτων-στόχων με ανάλογη μείωση στον χρόνο μετανάστευσης). Μια ιδέα προς

αυτήν την κατεύθυνση είναι η χρήση της ικανότητας κλωνοποίησης των πρακτόρων. Με χρήση της λειτουργίας αυτής, το κεντρικό μηχάνημα δεν θα είναι αναγκασμένο να δημιουργήσει όλους τους πράκτορες. Η αύξηση του συνολικού χρόνου δημιουργίας είναι αποτέλεσμα της παράλληλης λειτουργίας όλων των πρακτόρων στο αρχικό μηχάνημα. Με εφαρμογή της κλωνοποίησης πράκτορα, θα δημιουργούνται αρχικά μια σειρά από πράκτορες οι οποίοι θα κλωνοποιούνται μετά την μετανάστευση τους σε κάποιο μηχάνημα στόχο. Έτσι ο βέλτιστος αριθμός πρακτόρων θα επιτυγχάνεται με κλωνοποίηση των αρχικών, μειώνοντας τον χρόνο δημιουργίας.

ΠΑΡΑΡΤΗΜΑ Α

ΑΚΡΩΝΥΜΙΑ ΚΑΙ ΑΠΟΔΟΣΕΙΣ ΞΕΝΙΚΩΝ ΌΡΩΝ

Ακρωνύμιο	Ξενικός Όρος	Ελληνικός Όρος
	Router	Δρομολογητής
	end-to-end	άκρο-σε-άκρο
	Client/Server	Πελάτη/Εξυπηρετητή
	QOSFORUM	Οργανισμός Ποιότητας Υπηρεσίας
	Bandwidth	Εύρος ζώνης
	Delay	Καθυστέρηση μετάδοσης
	Jitter	Απόκλιση καθυστέρησης
	Packet loss	Απώλεια πακέτων
	Links	Διασυνδέσεις μετάδοσης
	Datagrams	Αυτοδύναμα πακέτα
	best-effort service	Υπηρεσία καλύτερης προσπάθειας”
	Stream	Σταθερής ροής
	Multicast	Πολλαπλή διανομή δεδομένων
	Transit delay	Καθυστέρηση διάβασης
	Fluid model	Ρευστό μοντέλο
	Admission control	Έλεγχος αποδοχής
	Policy control	Έλεγχος επιτήρησης
	Routing packet classification	Δρομολόγηση και ταξινόμηση πακέτων
	Packet Scheduler	Χρονοπρογραμματιστής Πακέτων
	Packet Classifier	Κατηγοριοποιητής Πακέτων
	Unicast	Απλή διανομή δεδομένων
	Domain	Διαχειριστικό πεδίο

	Traffic profile	Προφίλ κίνησης
	token-bucket	Κάδος με κουπόνι
	Buffer	Στοιχείο προσωρινής αποθήκευσης
	Longest match	Μέγιστη ταύτιση
	Agent	Πράκτορας
	Software agents	Λογισμικοί πράκτορες
	Remote evaluation	Απομακρυσμένη αξιολόγηση
	Migration	Μετανάστευση
	QosPersistenceManager	διαχειριστής οντοτήτων
	ParlayConectivityManager	Διαχειριστής συνδετικότητας Parlay
	QosResourceManager	Διαχειριστής πόρων
	NetConfigAgent	Κινητός πράκτορας Προπαρασκευής
	QosTemplate	Φόρμα ΠοιΥπ
	Wrappers	Πράκτορες προσαρμογής
	Queuing disciplines	Σχήμα αναμονής
	Common MA API	Κοινή διεπαφή-ΠΚΠ
AF	Assured Forwarding	
ASN.1	Abstract Syntax Notation 1	
ATM	Asynchronous Transfer Mode	Ασύγχρονος Τρόπος Μεταφοράς
CBQ	Class Based Queuing	
CBS	Committed Burst Size	
CIR	Committed Information Rate	
CLI	Command Line Interface	Διεπαφή γραμμής εντολών
CLS	Controlled Load Service-	Υπηρεσία Ελεγχόμενου Φόρτου
CMIS/CMIP	Common Management Information System/Common Management Information Protocol	
CoS	Class of Service	Κλάσης υπηρεσίας
DS	DiffServ Field	Πεδίο διαφοροποιημένων

		υπηρεσιών
DSCP	Differentiated Services Code Point	
EBS	Excess Burst Size	
EF	Expedited Forwarding	
FIFO	First In First Out	Πρώτο εισερχόμενο πρώτο εξερχόμενο
GS	Guaranteed Service	Υπηρεσία με εγγύηση
GUI	Graphical User Interfaces	Γραφική διεπαφή χρήστη
IGS	Information Gathering System	Σύστημα Συλλογής Πληροφοριών
IntServ	Integrated Services	Ολοκληρωμένες Υπηρεσίες
IP	Internet Protocol	Πρωτόκολλο Διαδικτύου
ISO	International standard Organization	Διεθνής Οργανισμός Πρότυπων
ISP	Internet Service Provider	Προμηθευτής Υπηρεσιών Διαδικτύου
JNI	Java Native Interface	
JVM	Java Virtual Machine	
LRU	Least Recently Used	
MASIF	Mobile Agent System Interoperability Facility	
MIB	Management Information Base	Βάση Διαχείρισης Πληροφορίας
MPLS	Multi Protocol Label Switching	Πολυ-πρωτόκολλο Μεταγωγής Ετικέτας
MST	Minimum Spanning Tree	Ελάχιστο Εκτεινόμενο Δέντρο
OMG	Object Management Group	
PDU	Protocol Data Units	Μονάδες Δεδομένων Πρωτοκόλλου
PFIFO	Packet FIFO	
PHB	Per-Hop Behavior	Ανά κόμβο συμπεριφορά
PQ	Priority Queuing	
RED	Random Early Drop	
RMI	Remote Method Invocation	Απομακρυσμένη Κλήση

		Μεθόδου
RPC	Remote Procedure Call	Απομακρυσμένη Κλήση Διεργασίας
RSVP	(Resource ReSerVation Protocol	Πρωτόκολλο Εγκατάστασης και Δέσμευσης Πόρων
Sap	Service access points	Σημεία πρόσβασης των χρηστών
shortest path		Μικρότερο μονοπάτι
SLA	Service Level Agreement	Συμβόλαιο Υπηρεσιών
SNMP	Simple Network Management Protocol	
TBF	Token Bucket Filter	
tc	traffic control tool	
TCM	Three Color Marker	
TOS	Type of Service	Πεδίο Τύπου Υπηρεσίας
TSP	Traveling Salesman Problem	Πρόβλημα Περιοδεύοντος Πωλητή
Tspec)	Traffic specifications	Ένδειξη Προδιαγραφών Κίνησης
VoIP	Voice over IP	
VPrP	Virtural Private Pipe	Εικονικός Ιδιωτικός Δίαυλος
WFQ	Weighted Fair Queuing	
WRR	Weighted Round Robin	
ΑΠ		Αντικείμενα Παρακολούθησης
ΒΔ		Οντότητας Βελτιστοποίησης Διαδρομής
BM		Βελτιστοποιητής Μονοπατιών
ΔΠΕ	Application Programming Interface	Διεπαφή Προγραμματισμού Εφαρμογής
ΔΥ	Differentiated Services	Διαφοροποιημένες Υπηρεσίες
KME	Central Process Unit	Κεντρικής Μονάδας Επεξεργασίας
ΚΠ	Mobile Agent	Κινητός Πρακτορας
ΠΚΠ	Mobile Agent Platform	Πλατφόρμα Κινητών Πρακτόρων

ΠοιΥπ	Quality of Service	Ποιότητα υπηρεσίας
ΣΠΠ	Multi Agent System	Σύστημα Πολλαπλών Πρακτόρων
ΤΚΠ	Mobile Agent Technology	Τεχνολογία Κινητών Πρακτόρων