

Πανεπιστήμιο Πατρών  
Πολυτεχνική Σχολή  
Τμήμα Μηχανικών Ηλεκτρονικών Υπολογιστών και Πληροφορικής

Διπλωματική Εργασία

# Αλγόριθμοι Επικοινωνίας σε Ad-hoc Κινητά Δίκτυα

Παναγιώτης Κόκκινος  
ΑΜ: 2125

Επιβλέπων: Χρήστος Ζαρολιάγκης

Πάτρα, Ιούλιος 2003

©Copyright Παναγιώτης Κόκκινος, Ιούλιος 2003

# Περιεχόμενα

Κατάλογος Σχημάτων	vi
Κατάλογος Πινάκων	xv
Πρόλογος	xvi
Ευχαριστίες	xix
<b>1 Εισαγωγή</b>	<b>1</b>
1.1 Γενικά . . . . .	1
1.2 Εφαρμογές των ad-hoc δικτύων . . . . .	2
1.3 Τεχνολογίες ασύρματης επικοινωνίας . . . . .	3
1.4 Συνεισφορά της διπλωματικής . . . . .	4
1.5 Οργάνωση της διπλωματικής . . . . .	5
<b>2 Χαρακτηριστικά των</b>	
<b>  πρωτοκόλλων δρομολόγησης για ad-hoc δίκτυα</b>	<b>7</b>
2.1 Γράφημα συνδεσιμότητας των πρωτοκόλλων δρομολόγησης . . .	7
2.2 Κατηγορίες των πρωτοκόλλων δρομολόγησης . . . . .	7
2.2.1 Γενικά . . . . .	7
2.2.2 Πρωτόκολλα δρομολόγησης που στηρίζονται στην εύρεση διαδρομών από κόμβους . . . . .	8
2.2.3 Πρωτόκολλα δρομολόγησης που στηρίζονται στην κίνηση των κόμβων . . . . .	10
2.3 Επιθυμητές ιδιότητες των πρωτοκόλλων δρομολόγησης . . . . .	11
2.4 Η σχέση με τα πρωτόκολλα δρομολόγησης για τα ενσύρματα δίκτυα . . . . .	11
<b>3 Πρωτόκολλα δρομολόγησης για ad-hoc δίκτυα</b>	<b>13</b>
3.1 Γενικά . . . . .	13
3.2 Dynamic Source Routing (DSR) . . . . .	13
3.2.1 Γενικά . . . . .	13

3.2.2	Το ασύρματο ad-hoc δίκτυο του DSR . . . . .	14
3.2.3	Διαδικασία Ανακάλυψης Διαδρομής (ΑΔ) . . . . .	15
3.2.4	Διαδικασία Συντήρησης Διαδρομής (ΣΔ) . . . . .	18
3.2.5	Βελτιστοποιήσεις και ειδικές περιπτώσεις για την διαδικασία ΑΔ . . . . .	20
3.2.6	Βελτιστοποιήσεις και ειδικές περιπτώσεις για την διαδικασία ΣΔ . . . . .	21
3.3	Zone Routing Protocol (ZRP) . . . . .	23
3.3.1	Γενικά . . . . .	23
3.3.2	Το ασύρματο ad-hoc δίκτυο του ZRP . . . . .	24
3.3.3	Intrazone Routing Protocol (IARP) . . . . .	26
3.3.4	Bordercast Resolution Protocol (BRP) . . . . .	29
3.3.5	Interzone Routing Protocol (IERP) . . . . .	31
3.4	Άλλα πρωτόκολλα δρομολόγησης . . . . .	33
3.4.1	Destination-Sequenced Distance-Vector (DSDV) . . . . .	33
3.4.2	Clusterhead Gateway Switch Routing (CGSR) . . . . .	34
3.4.3	Wireless Routing Protocol (WRP) . . . . .	35
3.4.4	Ad-hoc On Demand Distance Vector (AODV) . . . . .	36
3.4.5	Temporally Ordered Routing Algorithm (TORA) . . . . .	37
3.4.6	Associativity Based Routing (ABR) . . . . .	38
3.4.7	Support Routing Protocol (SRP) . . . . .	40
3.4.8	Παραλλαγές του Support Routing Protocol (SRP) . . . . .	41
<b>4</b>	<b>Υλοποίηση του Dynamic Source Routing (DSR)</b>	<b>43</b>
4.1	Γενικά . . . . .	43
4.2	Γενικό μοντέλο προσομοιωτή . . . . .	43
4.3	Δομές δεδομένων . . . . .	44
4.4	Τα πακέτα του πρωτοκόλλου . . . . .	48
4.5	Υλοποίηση του πρωτοκόλλου . . . . .	52
4.6	Εκτέλεση του πρωτοκόλλου . . . . .	56
4.7	Ειδικές περιπτώσεις και παραδοχές της υλοποίησης . . . . .	56
<b>5</b>	<b>Υλοποίηση του Zone Routing Protocol (ZRP)</b>	<b>59</b>
5.1	Γενικά . . . . .	59
5.2	Γενικό μοντέλο προσομοιωτή . . . . .	59
5.3	Δομές δεδομένων . . . . .	60
5.4	Τα πακέτα του πρωτοκόλλου . . . . .	65
5.5	Υλοποίηση του πρωτοκόλλου . . . . .	68
5.6	Εκτέλεση του πρωτοκόλλου . . . . .	74
5.7	Ειδικές περιπτώσεις και παραδοχές της υλοποίησης . . . . .	74

<b>6</b>	<b>Πειραματική αξιολόγηση</b>	<b>77</b>
6.1	Γενικά . . . . .	77
6.2	Περιβάλλον προσομοίωσης . . . . .	77
6.3	Διασύνδεση των υλοποιήσεων των πρωτοκόλλων με το περιβάλλον προσομοίωσης . . . . .	79
6.4	Παράμετροι και μετρούμενα μεγέθη πειραμάτων . . . . .	80
6.5	Dynamic Source Routing (DSR) . . . . .	81
6.5.1	Γενικά . . . . .	81
6.5.2	Μεταβολή ρυθμού κίνησης . . . . .	82
6.5.3	Μεταβολή πλήθους κινητών σταθμών . . . . .	86
6.5.4	Μεταβολή ρυθμού δημιουργίας πακέτων δεδομένων . . . . .	89
6.6	Zone Routing Protocol (ZRP) . . . . .	93
6.6.1	Γενικά . . . . .	93
6.6.2	Μεταβολή ρυθμού κίνησης . . . . .	94
6.6.3	Μεταβολή πλήθους κινητών σταθμών . . . . .	97
6.6.4	Μεταβολή ρυθμού δημιουργίας πακέτων δεδομένων . . . . .	100
6.7	Support Routing Protocol (SRP) . . . . .	101
6.8	Σύγκριση DSR, ZRP και SRP . . . . .	103
<b>7</b>	<b>Συνδυασμοί DSR και ZRP με το SRP</b>	<b>111</b>
7.1	Συνδυασμοί . . . . .	111
7.2	Παράμετροι και μετρούμενα μεγέθη πειραμάτων . . . . .	113
7.3	Πειράματα συνδυασμού DSR με το SRP . . . . .	113
7.4	Πειράματα συνδυασμού ZRP με το SRP . . . . .	119
<b>8</b>	<b>Συμπεράσματα</b>	<b>125</b>
<b>9</b>	<b>Μελλοντική εργασία</b>	<b>127</b>
<b>A'</b>	<b>Γραφικές παραστάσεις για το DSR</b>	<b>129</b>
A'.1	Ρυθμός κίνησης . . . . .	129
A'.2	Κινητοί σταθμοί . . . . .	133
A'.3	Ρυθμός δημιουργίας πακέτων δεδομένων . . . . .	137
<b>B'</b>	<b>Γραφικές παραστάσεις για το ZRP</b>	<b>141</b>
B'.1	Ρυθμός κίνησης . . . . .	141
B'.2	Κινητοί σταθμοί . . . . .	143

---

<b>Γ' Γραφικές παραστάσεις για το SRP</b>	<b>145</b>
Γ'.1 Σταθμοί υποστήριξης . . . . .	145
Γ'.2 Ρυθμός κίνησης . . . . .	148
Γ'.3 Κινητοί σταθμοί . . . . .	150
Γ'.4 Ρυθμός δημιουργίας πακέτων δεδομένων . . . . .	153
<b>Δ' Γραφικές παραστάσεις για τη σύγκριση DSR, ZRP και SRP</b>	<b>157</b>
Δ'.1 Κινητοί σταθμοί . . . . .	157
Δ'.2 Ρυθμός δημιουργίας πακέτων δεδομένων . . . . .	159
<b>Ε' Γραφικές παραστάσεις για το συνδυασμό DSR με το SRP</b>	<b>161</b>
Ε'.1 Ρυθμός κίνησης . . . . .	161
Ε'.2 Κινητοί σταθμοί . . . . .	166
Ε'.3 Ρυθμός δημιουργίας πακέτων δεδομένων . . . . .	173
<b>Ϝ' Γραφικές παραστάσεις για το συνδυασμό ZRP με το SRP</b>	<b>181</b>
Ϝ'.1 Ρυθμός κίνησης . . . . .	181
Ϝ'.2 Κινητοί σταθμοί . . . . .	183
Ϝ'.3 Ρυθμός δημιουργίας πακέτων δεδομένων . . . . .	185
<b>Ζ' Συντημήσεις και άλλες παρατηρήσεις</b>	<b>189</b>
<b>Βιβλιογραφία</b>	<b>193</b>

# Κατάλογος Σχημάτων

3.1	Γράφημα συνδεσιμότητας δικτύου για το DSR . . . . .	14
3.2	Περιορισμοί στην αποθήκευση ΠΔ στην ΜΠΑΔ . . . . .	20
3.3	Μεγάλο πλήθος ΑΠΔ . . . . .	22
3.4	Γράφημα συνδεσιμότητας δικτύου για το ZRP . . . . .	26
6.1	Ποσοστό Παραλαβής - Ρυθμός Κίνησης, για δισδιάστατα γραφήματα (DSR) . . . . .	82
6.2	Μέσος Αριθμός Βημάτων - Ρυθμός Κίνησης, για δισδιάστατα γραφήματα (DSR) . . . . .	83
6.3	Μέση Καθυστέρηση Λήψης - Ρυθμός Κίνησης, για δισδιάστατα γραφήματα (DSR) . . . . .	84
6.4	Ποσοστό Παραλαβής - Κινητοί Σταθμοί, για δισδιάστατα γραφήματα (DSR) . . . . .	86
6.5	Μέσος Αριθμός Βημάτων - Κινητοί Σταθμοί, για δισδιάστατα γραφήματα (DSR) . . . . .	87
6.6	Μέση Καθυστέρηση Λήψης - Κινητοί Σταθμοί, για δισδιάστατα γραφήματα (DSR) . . . . .	88
6.7	Ποσοστό Παραλαβής - Ρυθμός Δημιουργίας Πακέτων Δεδομένων, για δισδιάστατα γραφήματα (DSR) . . . . .	89
6.8	Μέσος Αριθμός Βημάτων - Ρυθμός Δημιουργίας Πακέτων Δεδομένων, για δισδιάστατα γραφήματα (DSR) . . . . .	90
6.9	Μέση Καθυστέρηση Λήψης - Ρυθμός Δημιουργίας Πακέτων Δεδομένων, για δισδιάστατα γραφήματα (DSR) . . . . .	91
6.10	Ποσοστό Παραλαβής - Ρυθμός Κίνησης, για δισδιάστατα γραφήματα (ZRP) . . . . .	95
6.11	Μέσος Αριθμός Βημάτων - Ρυθμός Κίνησης, για δισδιάστατα γραφήματα (ZRP) . . . . .	96
6.12	Μέση Καθυστέρηση Λήψης - Ρυθμός Κίνησης, για δισδιάστατα γραφήματα (ZRP) . . . . .	97
6.13	Ποσοστό Παραλαβής - Κινητοί Σταθμοί, για δισδιάστατα γραφήματα (ZRP) . . . . .	98

6.14 Μέσος Αριθμός Βημάτων - Κινητοί Σταθμοί, για δισδιάστατα γραφήματα (ZRP) . . . . .	99
6.15 Μέση Καθυστέρηση Λήψης - Κινητοί Σταθμοί, για δισδιάστατα γραφήματα (ZRP) . . . . .	100
6.16 Μέση Καθυστέρηση Λήψης - Σταθμοί Υποστήριξης, για δισδιάστατα γραφήματα (121) (SRP) . . . . .	101
6.17 Ποσοστό Παραλαβής - Ρυθμός Κίνησης, για δισδιάστατα γραφήματα (SRP) . . . . .	102
6.18 Μέση Καθυστέρηση Λήψης - Ρυθμός Κίνησης, για δισδιάστατα γραφήματα (SRP) . . . . .	103
6.19 Ποσοστό Παραλαβής - Ρυθμός Κίνησης, για δισδιάστατα γραφήματα (DSR, ZRP, SRP) . . . . .	104
6.20 Μέσος Αριθμός Βημάτων - Ρυθμός Κίνησης, για δισδιάστατα γραφήματα (DSR, ZRP) . . . . .	106
6.21 Μέση Καθυστέρηση Λήψης - Ρυθμός Κίνησης, για δισδιάστατα γραφήματα (DSR, ZRP, SRP) . . . . .	107
7.1 Ποσοστό Χρήσης DSR - Ρυθμός Κίνησης, για δισδιάστατα γραφήματα (συνδυασμός DSR - SRP) . . . . .	114
7.2 Ποσοστό Χρήσης SRP - Ρυθμός Κίνησης, για δισδιάστατα γραφήματα (συνδυασμός DSR - SRP) . . . . .	114
7.3 Ποσοστό Παραλαβής - Ρυθμός Κίνησης, για δισδιάστατα γραφήματα (συνδυασμός DSR - SRP, SRP) . . . . .	115
7.4 Ποσοστό Επιτυχούς Χρήσης DSR - Ρυθμός Κίνησης, για δισδιάστατα γραφήματα (συνδυασμός DSR - SRP) . . . . .	116
7.5 Ποσοστό Επιτυχούς Χρήσης SRP - Ρυθμός Κίνησης, για δισδιάστατα γραφήματα (συνδυασμός DSR - SRP) . . . . .	116
7.6 Μέση Καθυστέρηση Λήψης - Ρυθμός Κίνησης, για δισδιάστατα γραφήματα (συνδυασμός DSR - SRP, SRP) . . . . .	117
7.7 Μέση Καθυστέρηση Λήψης - Κινητοί Σταθμοί, για δισδιάστατα γραφήματα (συνδυασμός DSR - SRP, SRP) . . . . .	118
7.8 Ποσοστό Χρήσης ZRP - Κινητοί Σταθμοί, για δισδιάστατα γραφήματα (συνδυασμός ZRP - SRP) . . . . .	120
7.9 Ποσοστό Παραλαβής - Ρυθμός Κίνησης, για δισδιάστατα γραφήματα (συνδυασμός ZRP - SRP, SRP) . . . . .	121
7.10 Ποσοστό Παραλαβής - Κινητοί Σταθμοί, για δισδιάστατα γραφήματα (συνδυασμός ZRP - SRP, SRP) . . . . .	121
7.11 Ποσοστό Επιτυχούς Χρήσης ZRP - Ρυθμός Κίνησης, για δισδιάστατα γραφήματα (συνδυασμός ZRP - SRP) . . . . .	122
7.12 Μέση Καθυστέρηση Λήψης - Ρυθμός Κίνησης, για δισδιάστατα γραφήματα (συνδυασμός ZRP - SRP, SRP) . . . . .	123



7.13 Μέση Καθυστέρηση Λήψης - Κινητοί Σταθμοί, για δισδιάστατα γραφήματα (συνδυασμός ZRP - SRP, SRP) . . . . .	124
A'.1 Αριθμός Πακέτων Ελέγχου - Ρυθμός Κίνησης, για δισδιάστατα γραφήματα (DSR) . . . . .	129
A'.2 Χρήση Μνήμης - Ρυθμός Κίνησης, για δισδιάστατα γραφήματα (DSR) . . . . .	130
A'.3 Ποσοστό Παραλαβής - Ρυθμός Κίνησης, για τρισδιάστατα γραφήματα (DSR) . . . . .	130
A'.4 Μέσος Αριθμός Βημάτων - Ρυθμός Κίνησης, για τρισδιάστατα γραφήματα (DSR) . . . . .	131
A'.5 Μέση Καθυστέρηση Λήψης - Ρυθμός Κίνησης, για τρισδιάστατα γραφήματα (DSR) . . . . .	131
A'.6 Αριθμός Πακέτων Ελέγχου - Ρυθμός Κίνησης, για τρισδιάστατα γραφήματα (DSR) . . . . .	132
A'.7 Χρήση Μνήμης - Ρυθμός Κίνησης, για τρισδιάστατα γραφήματα (DSR) . . . . .	132
A'.8 Αριθμός Πακέτων Ελέγχου - Κινητοί Σταθμοί, για δισδιάστατα γραφήματα (DSR) . . . . .	133
A'.9 Χρήση Μνήμης - Κινητοί Σταθμοί, για δισδιάστατα γραφήματα (DSR) . . . . .	134
A'.10 Ποσοστό Παραλαβής - Κινητοί Σταθμοί, για τρισδιάστατα γραφήματα (DSR) . . . . .	134
A'.11 Μέσος Αριθμός Βημάτων - Κινητοί Σταθμοί, για τρισδιάστατα γραφήματα (DSR) . . . . .	135
A'.12 Μέση Καθυστέρηση Λήψης - Κινητοί Σταθμοί, για τρισδιάστατα γραφήματα (DSR) . . . . .	135
A'.13 Αριθμός Πακέτων Ελέγχου - Κινητοί Σταθμοί, για τρισδιάστατα γραφήματα (DSR) . . . . .	136
A'.14 Χρήση Μνήμης - Κινητοί Σταθμοί, για τρισδιάστατα γραφήματα (DSR) . . . . .	136
A'.15 Αριθμός Πακέτων Ελέγχου - Ρυθμός Δημιουργίας Πακέτων Δεδομένων, για δισδιάστατα γραφήματα (DSR) . . . . .	137
A'.16 Χρήση Μνήμης - Ρυθμός Δημιουργίας Πακέτων Δεδομένων, για δισδιάστατα γραφήματα (DSR) . . . . .	138
A'.17 Ποσοστό Παραλαβής - Ρυθμός Δημιουργίας Πακέτων Δεδομένων, για τρισδιάστατα γραφήματα (DSR) . . . . .	138
A'.18 Μέσος Αριθμός Βημάτων - Ρυθμός Δημιουργίας Πακέτων Δεδομένων, για τρισδιάστατα γραφήματα (DSR) . . . . .	139
A'.19 Μέση Καθυστέρηση Λήψης - Ρυθμός Δημιουργίας Πακέτων Δεδομένων, για τρισδιάστατα γραφήματα (DSR) . . . . .	139

A'.20	Αριθμός Πακέτων Ελέγχου - Ρυθμός Δημιουργίας Πακέτων Δεδομένων, για τρισδιάστατα γραφήματα (DSR) . . . . .	140
A'.21	Χρήση Μνήμης - Ρυθμός Δημιουργίας Πακέτων Δεδομένων, για τρισδιάστατα γραφήματα (DSR) . . . . .	140
B'.1	Αριθμός Πακέτων Ελέγχου - Ρυθμός Κίνησης, για δισδιάστατα γραφήματα (ZRP) . . . . .	141
B'.2	Χρήση Μνήμης - Ρυθμός Κίνησης, για δισδιάστατα γραφήματα (ZRP) . . . . .	142
B'.3	Αριθμός Πακέτων Ελέγχου - Κινητοί Σταθμοί, για δισδιάστατα γραφήματα (ZRP) . . . . .	143
B'.4	Χρήση Μνήμης - Κινητοί Σταθμοί, για δισδιάστατα γραφήματα (ZRP) . . . . .	144
Γ'.1	Μέση Καθυστέρηση Λήψης - Σταθμοί Υποστήριξης, για δισδιάστατα γραφήματα (529) (SRP) . . . . .	145
Γ'.2	Μέση Καθυστέρηση Λήψης - Σταθμοί Υποστήριξης, για δισδιάστατα γραφήματα (1024) (SRP) . . . . .	146
Γ'.3	Μέση Καθυστέρηση Λήψης - Σταθμοί Υποστήριξης, για τρισδιάστατα γραφήματα (125) (SRP) . . . . .	146
Γ'.4	Μέση Καθυστέρηση Λήψης - Σταθμοί Υποστήριξης, για τρισδιάστατα γραφήματα (512) (SRP) . . . . .	147
Γ'.5	Μέση Καθυστέρηση Λήψης - Σταθμοί Υποστήριξης, για τρισδιάστατα γραφήματα (1000) (SRP) . . . . .	147
Γ'.6	Ποσοστό Παραλαβής - Ρυθμός Κίνησης, για τρισδιάστατα γραφήματα (SRP) . . . . .	148
Γ'.7	Μέση Καθυστέρηση Λήψης - Ρυθμός Κίνησης, για τρισδιάστατα γραφήματα (SRP) . . . . .	149
Γ'.8	Ποσοστό Παραλαβής - Κινητοί Σταθμοί, για δισδιάστατα γραφήματα (SRP) . . . . .	150
Γ'.9	Μέση Καθυστέρηση Λήψης - Κινητοί Σταθμοί, για δισδιάστατα γραφήματα (SRP) . . . . .	151
Γ'.10	Ποσοστό Παραλαβής - Κινητοί Σταθμοί, για τρισδιάστατα γραφήματα (SRP) . . . . .	151
Γ'.11	Μέση Καθυστέρηση Λήψης - Κινητοί Σταθμοί, για τρισδιάστατα γραφήματα (SRP) . . . . .	152
Γ'.12	Ποσοστό Παραλαβής - Ρυθμός Δημιουργίας Πακέτων Δεδομένων, για δισδιάστατα γραφήματα (SRP) . . . . .	153
Γ'.13	Μέση Καθυστέρηση Λήψης - Ρυθμός Δημιουργίας Πακέτων Δεδομένων, για δισδιάστατα γραφήματα (SRP) . . . . .	154

Γ'.14 Ποσοστό Παραλαβής - Ρυθμός Δημιουργίας Πακέτων Δεδομένων, για τρισδιάστατα γραφήματα (SRP) . . . . .	154
Γ'.15 Μέση Καθυστέρηση Λήψης - Ρυθμός Δημιουργίας Πακέτων Δεδομένων, για τρισδιάστατα γραφήματα (SRP) . . . . .	155
Δ'.1 Ποσοστό Παραλαβής - Κινητοί Σταθμοί, για δισδιάστατα γραφήματα (DSR, ZRP, SRP) . . . . .	157
Δ'.2 Μέσος Αριθμός Βημάτων - Κινητοί Σταθμοί, για δισδιάστατα γραφήματα (DSR, ZRP) . . . . .	158
Δ'.3 Μέση Καθυστέρηση Λήψης - Κινητοί Σταθμοί, για δισδιάστατα γραφήματα (DSR, ZRP, SRP) . . . . .	158
Δ'.4 Ποσοστό Παραλαβής - Ρυθμός Δημιουργίας Πακέτων Δεδομένων, για δισδιάστατα γραφήματα (DSR, ZRP, SRP) . . . . .	159
Δ'.5 Μέσος Αριθμός Βημάτων - Ρυθμός Δημιουργίας Πακέτων Δεδομένων, για δισδιάστατα γραφήματα (DSR, ZRP) . . . . .	160
Δ'.6 Μέση Καθυστέρηση Λήψης - Ρυθμός Δημιουργίας Πακέτων Δεδομένων, για δισδιάστατα γραφήματα (DSR, ZRP, SRP) . . . . .	160
Ε'.1 Ποσοστό Παραλαβής - Ρυθμός Κίνησης, για δισδιάστατα γραφήματα (συνδυασμός DSR - SRP) . . . . .	161
Ε'.2 Μέση Καθυστέρηση Λήψης - Ρυθμός Κίνησης, για δισδιάστατα γραφήματα (συνδυασμός DSR - SRP) . . . . .	162
Ε'.3 Ποσοστό Παραλαβής - Ρυθμός Κίνησης, για τρισδιάστατα γραφήματα (συνδυασμός DSR - SRP) . . . . .	162
Ε'.4 Μέση Καθυστέρηση Λήψης - Ρυθμός Κίνησης, για τρισδιάστατα γραφήματα (συνδυασμός DSR - SRP) . . . . .	163
Ε'.5 Ποσοστό Χρήσης DSR - Ρυθμός Κίνησης, για τρισδιάστατα γραφήματα (συνδυασμός DSR - SRP) . . . . .	163
Ε'.6 Ποσοστό Χρήσης SRP - Ρυθμός Κίνησης, για τρισδιάστατα γραφήματα (συνδυασμός DSR - SRP) . . . . .	164
Ε'.7 Ποσοστό Επιτυχούς Χρήσης DSR - Ρυθμός Κίνησης, για τρισδιάστατα γραφήματα (συνδυασμός DSR - SRP) . . . . .	164
Ε'.8 Ποσοστό Επιτυχούς Χρήσης SRP - Ρυθμός Κίνησης, για τρισδιάστατα γραφήματα (συνδυασμός DSR - SRP) . . . . .	165
Ε'.9 Ποσοστό Παραλαβής - Κινητοί Σταθμοί, για δισδιάστατα γραφήματα (συνδυασμός DSR - SRP, SRP) . . . . .	166
Ε'.10 Ποσοστό Παραλαβής - Κινητοί Σταθμοί, για δισδιάστατα γραφήματα (συνδυασμός DSR - SRP) . . . . .	167
Ε'.11 Μέση Καθυστέρηση Λήψης - Κινητοί Σταθμοί, για δισδιάστατα γραφήματα (συνδυασμός DSR - SRP) . . . . .	167

E'.12 Ποσοστό Χρήσης DSR - Κινητοί Σταθμοί, για δισδιάστατα γραφήματα (συνδυασμός DSR - SRP) . . . . .	168
E'.13 Ποσοστό Χρήσης SRP - Κινητοί Σταθμοί, για δισδιάστατα γραφήματα (συνδυασμός DSR - SRP) . . . . .	168
E'.14 Ποσοστό Επιτυχούς Χρήσης DSR - Κινητοί Σταθμοί, για δισδιάστατα γραφήματα (συνδυασμός DSR - SRP) . . . . .	169
E'.15 Ποσοστό Επιτυχούς Χρήσης SRP - Κινητοί Σταθμοί, για δισδιάστατα γραφήματα (συνδυασμός DSR - SRP) . . . . .	169
E'.16 Ποσοστό Παραλαβής - Κινητοί Σταθμοί, για τρισδιάστατα γραφήματα (συνδυασμός DSR - SRP) . . . . .	170
E'.17 Μέση Καθυστέρηση Λήψης - Κινητοί Σταθμοί, για τρισδιάστατα γραφήματα (συνδυασμός DSR - SRP) . . . . .	170
E'.18 Ποσοστό Χρήσης DSR - Κινητοί Σταθμοί, για τρισδιάστατα γραφήματα (συνδυασμός DSR - SRP) . . . . .	171
E'.19 Ποσοστό Χρήσης SRP - Κινητοί Σταθμοί, για τρισδιάστατα γραφήματα (συνδυασμός DSR - SRP) . . . . .	171
E'.20 Ποσοστό Επιτυχούς Χρήσης DSR - Κινητοί Σταθμοί, για τρισδιάστατα γραφήματα (συνδυασμός DSR - SRP) . . . . .	172
E'.21 Ποσοστό Επιτυχούς Χρήσης SRP - Κινητοί Σταθμοί, για τρισδιάστατα γραφήματα (συνδυασμός DSR - SRP) . . . . .	172
E'.22 Ποσοστό Παραλαβής - Ρυθμός Δημιουργίας Πακέτων Δεδομένων, για δισδιάστατα γραφήματα (συνδυασμός DSR - SRP, SRP) . . . . .	173
E'.23 Μέση Καθυστέρηση Λήψης - Ρυθμός Δημιουργίας Πακέτων Δεδομένων, για δισδιάστατα γραφήματα (συνδυασμός DSR - SRP, SRP) . . . . .	174
E'.24 Ποσοστό Παραλαβής - Ρυθμός Δημιουργίας Πακέτων Δεδομένων, για δισδιάστατα γραφήματα (συνδυασμός DSR - SRP) . . . . .	174
E'.25 Μέση Καθυστέρηση Λήψης - Ρυθμός Δημιουργίας Πακέτων Δεδομένων, για δισδιάστατα γραφήματα (συνδυασμός DSR - SRP) . . . . .	175
E'.26 Ποσοστό Χρήσης DSR - Ρυθμός Δημιουργίας Πακέτων Δεδομένων, για δισδιάστατα γραφήματα (συνδυασμός DSR - SRP) . . . . .	175
E'.27 Ποσοστό Χρήσης SRP - Ρυθμός Δημιουργίας Πακέτων Δεδομένων, για δισδιάστατα γραφήματα (συνδυασμός DSR - SRP) . . . . .	176
E'.28 Ποσοστό Επιτυχούς Χρήσης DSR - Ρυθμός Δημιουργίας Πακέτων Δεδομένων, για δισδιάστατα γραφήματα (συνδυασμός DSR - SRP) . . . . .	176
E'.29 Ποσοστό Επιτυχούς Χρήσης SRP - Ρυθμός Δημιουργίας Πακέτων Δεδομένων, για δισδιάστατα γραφήματα (συνδυασμός DSR - SRP) . . . . .	177

Ε'.30 Ποσοστό Παραλαβής - Ρυθμός Δημιουργίας Πακέτων Δεδομένων, για τρισδιάστατα γραφήματα (συνδυασμός DSR - SRP) . . .	177
Ε'.31 Μέση Καθυστέρηση Λήψης - Ρυθμός Δημιουργίας Πακέτων Δεδομένων, για τρισδιάστατα γραφήματα (συνδυασμός DSR - SRP) . . . . .	178
Ε'.32 Ποσοστό Χρήσης DSR - Ρυθμός Δημιουργίας Πακέτων Δεδομένων, για τρισδιάστατα γραφήματα (συνδυασμός DSR - SRP)	178
Ε'.33 Ποσοστό Χρήσης SRP - Ρυθμός Δημιουργίας Πακέτων Δεδομένων, για τρισδιάστατα γραφήματα (συνδυασμός DSR - SRP)	179
Ε'.34 Ποσοστό Επιτυχούς Χρήσης DSR - Ρυθμός Δημιουργίας Πακέτων Δεδομένων, για τρισδιάστατα γραφήματα (συνδυασμός DSR - SRP) . . . . .	179
Ε'.35 Ποσοστό Επιτυχούς Χρήσης SRP - Ρυθμός Δημιουργίας Πακέτων Δεδομένων, για τρισδιάστατα γραφήματα (συνδυασμός DSR - SRP) . . . . .	180
ε'.1 Ποσοστό Χρήσης ZRP - Ρυθμός Κίνησης, για δισδιάστατα γραφήματα (συνδυασμός ZRP - SRP) . . . . .	181
ε'.2 Ποσοστό Χρήσης SRP - Ρυθμός Κίνησης, για δισδιάστατα γραφήματα (συνδυασμός ZRP - SRP) . . . . .	182
ε'.3 Ποσοστό Επιτυχούς Χρήσης SRP - Ρυθμός Κίνησης, για δισδιάστατα γραφήματα (συνδυασμός ZRP - SRP) . . . . .	182
ε'.4 Ποσοστό Χρήσης SRP - Κινητοί Σταθμοί, για δισδιάστατα γραφήματα (συνδυασμός ZRP - SRP) . . . . .	183
ε'.5 Ποσοστό Επιτυχούς Χρήσης ZRP - Κινητοί Σταθμοί, για δισδιάστατα γραφήματα (συνδυασμός ZRP - SRP) . . . . .	184
ε'.6 Ποσοστό Επιτυχούς Χρήσης SRP - Κινητοί Σταθμοί, για δισδιάστατα γραφήματα (συνδυασμός ZRP - SRP) . . . . .	184
ε'.7 Ποσοστό Παραλαβής - Ρυθμός Δημιουργίας Πακέτων Δεδομένων, για δισδιάστατα γραφήματα (συνδυασμός ZRP - SRP, SRP)	185
ε'.8 Μέση Καθυστέρηση Λήψης - Ρυθμός Δημιουργίας Πακέτων Δεδομένων, για δισδιάστατα γραφήματα (συνδυασμός ZRP - SRP, SRP) . . . . .	186
ε'.9 Ποσοστό Χρήσης ZRP - Ρυθμός Δημιουργίας Πακέτων Δεδομένων, για δισδιάστατα γραφήματα (συνδυασμός ZRP - SRP)	186
ε'.10 Ποσοστό Χρήσης SRP - Ρυθμός Δημιουργίας Πακέτων Δεδομένων, για δισδιάστατα γραφήματα (συνδυασμός ZRP - SRP)	187
ε'.11 Ποσοστό Επιτυχούς Χρήσης ZRP - Ρυθμός Δημιουργίας Πακέτων Δεδομένων, για δισδιάστατα γραφήματα (συνδυασμός ZRP - SRP) . . . . .	187

φ'.12 Ποσοστό Επιτυχούς Χρήσης SRP - Ρυθμός Δημιουργίας Πακέτων Δεδομένων, για δισδιάστατα γραφήματα (συνδυασμός ZRP - S- RP) . . . . .	188
---	-----

## Κατάλογος Πινάκων

3.1	Δομή μίας ΜΠΑΔ . . . . .	15
3.2	Δομή μίας ΑΙΔ . . . . .	16
3.3	Δομή μίας ΑΠΔ . . . . .	16
3.4	Δομή ενός πακέτου δεδομένων . . . . .	17
3.5	Δομή ενός ΛΔ . . . . .	19
3.6	Δομή μηνύματος που οι κόμβοι ανταλλάσσουν μεταξύ τους για το NDP . . . . .	27
3.7	Πεδία του Πίνακα Γειτόνων . . . . .	27
3.8	Δομή του Πακέτου Κατάστασης Ακμών . . . . .	28
3.9	Πεδία του Πίνακα Κατάστασης Ακμών . . . . .	28
3.10	Πεδία του πίνακα δρομολόγησης του IARP . . . . .	29
3.11	Δομή του Πακέτου Αίτησης Ανακάλυψης Διαδρομής . . . . .	30
3.12	Πεδία του Πίνακα Κάλυψης Αιτήσεων . . . . .	30
3.13	Πεδία του πίνακα δρομολόγησης του IERP . . . . .	31
3.14	Δομή του Ενσωματωμένου Πακέτου . . . . .	32





# Πρόλογος

Τα τελευταία χρόνια η αλματώδης τεχνολογική ανάπτυξη εισήγαγε στην καθημερινότητα μας την έννοια του κινητού υπολογισμού. Αυτή εμφανίζεται με την ύπαρξη μικρών αλλά πολύ ισχυρών υπολογιστικά συσκευών, οι οποίες μπορούν να επικοινωνούν ασύρματα μεταξύ τους ακόμα και εν κινήσει. Η κινητή (κυψελοειδής) τηλεφωνία είναι σήμερα το χαρακτηριστικότερο παράδειγμα εφαρμογής του κινητού υπολογισμού. Τα τελευταία χρόνια ο κινητός υπολογισμός κάνει την εμφάνιση του και στο χώρο των υπολογιστών, με τα ασύρματα δίκτυα δεδομένων.

Ένα από τα σημαντικότερα μοντέλα ασύρματων δικτύων δεδομένων είναι αυτό των ad-hoc δικτύων. Σε αυτό το μοντέλο δεν υπάρχει καμμία κεντρική, σταθερή υποδομή αλλά οι κινητοί σταθμοί, οι οποίοι είναι οι μόνοι υπεύθυνοι για την ασύρματη επικοινωνία τους, δημιουργούν ένα προσωρινό δίκτυο. Στα ad-hoc δίκτυα ιδιαίτερο ενδιαφέρον παρουσιάζει η περίπτωση που δεν είναι δυνατή η άμεση ασύρματη επικοινωνία κάθε κινητού σταθμού με κάθε άλλον. Στην περίπτωση αυτή για την επικοινωνία δύο κινητών σταθμών που δεν έχουν άμεση ασύρματη επαφή θα πρέπει κάποιος από τους υπόλοιπους να συνεργαστούν για να μεταφέρουν πακέτα δεδομένων μεταξύ των δύο αυτών σταθμών. Η εύρεση πρωτοκόλλων δρομολόγησης που κάνουν δυνατό αυτόν τον τρόπο επικοινωνίας και οργάνωσης των σταθμών, στα ad-hoc δίκτυα, είναι ένας πολύ σημαντικός τομέας έρευνας.

Στη διπλωματική αρχικά κάνουμε μία γενική παρουσίαση των πρωτοκόλλων δρομολόγησης για ad-hoc δίκτυα. Αναλυτικά εξετάζουμε δύο πρωτόκολλα δρομολόγησης, το Dynamic Source Routing (DSR) και το Zone Routing Protocol (ZRP), τα οποία και υλοποιήσαμε. Στη συνέχεια παρουσιάζουμε τα αποτελέσματα των πειραμάτων μας με αυτά τα πρωτόκολλα, τα οποία πραγματοποιήθηκαν με τη βοήθεια ενός περιβάλλοντος προσομοίωσης. Ακόμα παρουσιάζουμε τα αποτελέσματα των πειραμάτων μας και με ένα τρίτο πρωτόκολλο δρομολόγησης, το Support Routing Protocol (SRP). Με τα πειράματα αυτά μπορούμε και συγκρίνουμε τα τρία πρωτόκολλα. Τέλος υλοποιήσαμε και εκτελέσαμε πειράματα με συνδυασμούς των τριών αυτών πρωτοκόλλων, προσπαθώντας να βελτιώσουμε την συμπεριφορά τους. Παρουσιάζουμε τα

αποτελέσματα και αυτών των πειραμάτων.

Οι δυνατότητες εφαρμογής των ad-hoc δικτύων τόσο στη καθημερινή μας ζωή όσο και σε εξειδικευμένες περιπτώσεις είναι πάρα πολλές. Η ημέρα που θα γευτούμε στην πράξη τα οφέλη αυτών των εφαρμογών δεν είναι πολύ μακριά.

# Ευχαριστίες

Καταρχήν θέλω να ευχαριστήσω την οικογένεια μου, Χριστόφορο, Βασιλεία και Κώστα, για την αγάπη και την στήριξη τους.

Ευχαριστώ τον κ.Χρήστο Ζαρολιάγκη, επιβλέπων της διπλωματικής, που μου έδωσε την δυνατότητα να ασχοληθώ με ένα τόσο ενδιαφέρον θέμα όπως αυτό των πρωτοκόλλων δρομολόγησης για τα ad-hoc δίκτυα. Η καθοδήγηση του, οι παρατηρήσεις και οι συμβουλές του ήταν πολύ σημαντικές για την ολοκλήρωση της διπλωματικής. Παράλληλα θέλω να τον ευχαριστήσω για τις πολύτιμες γνώσεις που απέκτησα παρακολουθώντας όλα αυτά τα χρόνια τα μαθήματα που διδάσκει στη σχολή.

Ευχαριστώ τον κ.Ιωάννη Χατζηγιαννάκη, συνεπιβλέπων της διπλωματικής, με τον οποίον είχα μία στενή συνεργασία για τη συγγραφή της διπλωματικής. Άλλωστε μέρος της διπλωματικής αυτής έχει σαν βάση δικό του έργο. Ο κ.Χατζηγιαννάκης ήταν πάντα παρόν για να με βοηθάει και να με παροτρύνει. Γιάννη ευχαριστώ.

Τέλος να ευχαριστήσω τη συμφοιτήτρια μου Ελένη Καλτσά που με βοήθησε ιδιαίτερα στη φάση των πειραμάτων.

Με την εργασία αυτή κλείνει ο κύκλος των προπτυχιακών σπουδών μου στο τμήμα των Μηχανικών Ηλεκτρονικών Υπολογιστών και Πληροφορικής. Οι γνώσεις και οι εμπειρίες που απέκτησα μέσω των σπουδών μου στο τμήμα μου δίνουν ισχυρά εφόδια για το μέλλον.



# Κεφάλαιο 1

## Εισαγωγή

### 1.1 Γενικά

Τα τελευταία χρόνια η αλματώδης τεχνολογική ανάπτυξη εισήγαγε στην καθημερινότητα μας την έννοια του κινητού υπολογισμού. Αυτή εμφανίζεται με την ύπαρξη μικρών αλλά πολύ ισχυρών υπολογιστικά συσκευών, οι οποίες μπορούν να επικοινωνούν ασύρματα μεταξύ τους ακόμα και εν κινήσει. Η κινητή (κυψελοειδής) τηλεφωνία είναι σήμερα το χαρακτηριστικότερο παράδειγμα εφαρμογής του κινητού υπολογισμού. Τα τελευταία χρόνια ωστόσο ο κινητός υπολογισμός κάνει την εμφάνιση του και στο χώρο των υπολογιστών και συγκεκριμένα στα δίκτυα δεδομένων.

Σήμερα σχεδόν κάθε υπολογιστής είναι συνδεδεμένος στο διαδίκτυο ή σε κάποιο άλλο μικρότερο δίκτυο δεδομένων. Οι υπηρεσίες και οι δυνατότητες που τα δίκτυα δεδομένων προσφέρουν για επικοινωνία, συνεργασία, ενημέρωση, εκπαίδευση και διασκέδαση είναι άπειρες. Παράλληλα η εμφάνιση μικρών, φορητών υπολογιστικών συσκευών (π.χ. laptop, PDA) με απίστευτες δυνατότητες αλλά και οι εξελίξεις στον χώρο της ασύρματη ψηφιακής επικοινωνίας οδήγησαν στα ασύρματα δίκτυα δεδομένων.

Υπάρχουν δύο βασικά μοντέλα ασυρμάτων δικτύων δεδομένων, αλλά και γενικότερα κινητού υπολογισμού. Αυτά διαφέρουν κυρίως στον τρόπο με τον οποίο οι κινητοί σταθμοί, οι οποίοι τα αποτελούν, επικοινωνούν μεταξύ τους. Στο πρώτο μοντέλο υπάρχει μία σταθερή, κεντρική υποδομή μέσω της οποίας πραγματοποιείται η επικοινωνία μεταξύ των κινητών σταθμών. Η υποδομή αυτή χρησιμοποιείται και για την επικοινωνία με άλλα δίκτυα. Χαρακτηριστικό παράδειγμα χρήσης του μοντέλου αυτού είναι τα WLANs (Wireless LANs), με την χρήση σημείων πρόσβασης (Access Points). Στο δεύτερο μοντέλο δεν υπάρχει καμμία κεντρική, σταθερή υποδομή αλλά οι κινητοί σταθμοί, οι οποίοι είναι οι μόνοι υπεύθυνοι για την ασύρματη επικοινωνία τους, δημιουργούν ένα

προσωρινό δίκτυο. Τα δίκτυα δεδομένων τα οποία αντιστοιχούν σε αυτό το δεύτερο μοντέλο είναι γνωστά σαν ad-hoc δίκτυα.

Στα ad-hoc δίκτυα στην περίπτωση που κάθε κινητός σταθμός βρίσκεται μέσα στο εύρος μετάδοσης των υπολοίπων, τότε η επικοινωνία μεταξύ τους μπορεί να πραγματοποιηθεί άμεσα με την χρήση του ασύρματου μέσου. Ιδιαίτερο ενδιαφέρον ωστόσο παρουσιάζει η περίπτωση που δεν είναι δυνατή η άμεση ασύρματη επικοινωνία κάθε κινητού σταθμού με κάθε άλλον. Στην περίπτωση αυτή για την επικοινωνία δύο κινητών σταθμών που δεν έχουν άμεση ασύρματη επαφή θα πρέπει κάποιοι από τους υπόλοιπους να συνεργαστούν για να μεταφέρουν πακέτα δεδομένων μεταξύ των δύο αυτών σταθμών. Η εύρεση πρωτοκόλλων δρομολόγησης που κάνουν δυνατό αυτόν τον τρόπο επικοινωνίας και οργάνωσης των σταθμών, στα ad-hoc δίκτυα, είναι ένας πολύ σημαντικός τομέας έρευνας.

## 1.2 Εφαρμογές των ad-hoc δικτύων

Στα ad-hoc δίκτυα οι χρήστες μπορούν να χρησιμοποιούν τις συνηθισμένες δικτυακές εφαρμογές τους αποδεδειγμένοι από τις όποιες καλωδιακές ή κεντρικές σταθερές υποδομές (π.χ. ενσύρματα δίκτυα, κυψελοειδοί συστήματα). Αυτό είναι ιδιαίτερα χρήσιμο σε περιπτώσεις όπου δεν υπάρχει η απαραίτητη υποδομή όπως σε φτωχές περιοχές του πλανήτη, σε περιοχές όπου κάποιο γεγονός έχει καταστρέψει την όποια υποδομή ή σε περιοχές και καταστάσεις που η χρήση της είναι δύσκολη έως αδύνατη (π.χ. στην ύπαιθρο). Επομένως ad-hoc δίκτυα μπορεί να δημιουργηθούν π.χ. μεταξύ των κινητών σταθμών συνεργατών σε ένα συνέδριο ή μέσα σε ένα εργοστάσιο, μεταξύ των κινητών σταθμών των μελών ομάδων διάσωσης, μεταξύ των κινητών σταθμών στρατιωτών σε ένα πεδίο μάχης ή ακόμα και μεταξύ των κινητών σταθμών φίλων που βρίσκονται σε διαφορετικά οχήματα. Οι εφαρμογές είναι άπειρες.

Παρόλα αυτά ακόμα και στην περίπτωση που η απαραίτητη υποδομή είναι υπαρκτή η χρήση της για την επικοινωνία σε τοπικό επίπεδο μοιάζει να είναι μη αναγκαία, από την στιγμή που είναι δυνατή η ασύρματη επικοινωνία μεταξύ των διαφόρων σταθμών. Έτσι τα ad-hoc δίκτυα μπορεί να αποτελέσουν τον νέο τρόπο δικτύωσης στο σπίτι ή στον χώρο εργασίας.

Επιπλέον ένα ad-hoc δίκτυο μπορεί να συνδέεται με ένα άλλο σταθερό δίκτυο δεδομένων (π.χ. το διαδίκτυο), δίνοντας έτσι στους χρήστες του όλα τα πλεονεκτήματα ενός ad-hoc δικτύου (κίνηση, αποδέσμευση από καλώδια και σταθερές υποδομές) αλλά και τις υπηρεσίες και τις πληροφορίες που χρειάζονται από το σταθερό δίκτυο δεδομένων. Παράλληλα με τον τρόπο αυτό επεκτείνεται η εμβέλεια του σταθερού δικτύου δεδομένων αφού μπορούν να αποκτήσουν πρόσβαση σε αυτό και κινητοί σταθμοί που δεν έχουν άμεση ασύρματη επικοινωνία

ωνία με κάποιο σημείο πρόσβασης του (Access Point).

Τα ad-hoc δίκτυα στο μέλλον θα χρησιμοποιηθούν σε διάφορες ιδιαίτερα πρωτοποριακές εφαρμογές. Χαρακτηριστικό παράδειγμα μίας τέτοιας εφαρμογής είναι τα δίκτυα έξυπνης σκόνης, που είναι ένα είδος ad-hoc δικτύων. Τα δίκτυα αυτά αποτελούνται από μικρές υπολογιστικές μονάδες ειδικού σκοπού, οι οποίες διασπείρονται σε μεγάλες ποσότητες σε μία γεωγραφική περιοχή. Οι μονάδες αυτές έχουν την δυνατότητα να αντιλαμβάνονται κρίσιμα γεγονότα (π.χ. εκδήλωση πυρκαγιάς) ή να κάνουν διάφορες μετρήσεις (π.χ. κλιματολογικές μετρήσεις) και εν συνεχεία να επικοινωνούν με τις γειτονικές τους για την μεταφορά της πληροφορίας που συνέλλεξαν σε κάποια κεντρική μονάδα.

### 1.3 Τεχνολογίες ασύρματης επικοινωνίας

Γενικά για την επίτευξη της ασύρματης επικοινωνίας χρησιμοποιούνται ηλεκτρομαγνητικά σήματα, είτε υπερύθρων είτε ραδιοκυμάτων. Η χρήση ωστόσο των υπερύθρων σήμερα είναι σχετικά περιορισμένη λόγω της μικρής εμβέλειας των αντίστοιχων συσκευών αλλά και της απαίτησης για οπτική επαφή αυτών. Έτσι οι περισσότερες τεχνολογίες ασύρματης επικοινωνίας σήμερα στηρίζονται στα ραδιοκύματα. Τα αντίστοιχα σημαντικότερα πρότυπα ασύρματης επικοινωνίας, τα οποία χρησιμοποιούνται και στα ad-hoc δίκτυα, είναι το 802.11 και οι εξελίξεις του (802.11b, 802.11a, 802.11g) καθώς και το Bluetooth. Το πρότυπο 802.11 και οι εξελίξεις του έχουν μεγαλύτερο εύρος μετάδοσης και προσφέρουν μεγαλύτερες ταχύτητες από το Bluetooth. Το Bluetooth ωστόσο εξαιτίας κυρίως του χαμηλού κόστους του χρησιμοποιείται ιδιαίτερα για την ασύρματη επικοινωνία μικρών συσκευών όπως τα PDA. Στις τεχνολογίες αυτές η μετάδοση του σήματος μπορεί να γίνεται από σημείο σε σημείο ή από ένα σημείο σε όλους όσους βρίσκονται μέσα στο εύρος μετάδοσης του πομπού.

Οι τεχνολογίες αυτές και γενικότερα η μετάδοση δεδομένων μέσω του αέρα, παρουσιάζουν ορισμένα μειονεκτήματα σε σχέση με την μετάδοση μέσω των καλωδίων. Καταρχήν η ταχύτητα μετάδοσης δεδομένων πάνω από το ασύρματο μέσο είναι τουλάχιστον μία τάξης μεγέθους μικρότερη σε σχέση με την ταχύτητα στο ενσύρματο μέσο. Ακόμα ο ρυθμός των λαθών κατά την μετάδοση δεδομένων στο ασύρματο μέσο είναι αρκετά μεγαλύτερος σε σχέση με αυτόν του ενσύρματου μέσου. Αυτό οφείλεται στο γεγονός ότι το σήμα που μεταδίδεται μέσω του αέρα είναι περισσότερο ευάλωτο στις κλιματολογικές συνθήκες και σε άλλες πηγές θορύβου καθώς και στα εμπόδια (φυσικά ή τεχνητά) του περιβάλλοντα χώρου. Παράλληλα το πρωτόκολλο TCP, που χρησιμοποιείται και στα ad-hoc δίκτυα, αντιμετωπίζει τα αντίστοιχα χαμένα πακέτα δεδομένων σαν σημάδι συμφόρησης του δικτύου και όχι σαν αποτέλεσμα των αυξημένων λαθών δεδομένων. Αυτό έχει σαν αποτέλεσμα την ακόμα χαμηλότερη από-

δοση του ασύρματου μέσου όταν υπάρχουν πολλά λάθροι δεδομένων. Επιπλέον η ασφάλεια των δεδομένων που μεταφέρονται μέσω του ασύρματου μέσου είναι μικρότερη σε σχέση με την ασφάλεια των δεδομένων στο ενσύρματο μέσο. Προσπάθειες για την αύξηση της ασφάλειας με την χρήση κλειδιών ή άλλων μεθόδων (π.χ. WEP - Wired Equivalent Privacy) πέρα από το ότι είναι δύσκολες στην εφαρμογή τους, οδηγούν και στην περαιτέρω μείωση της απόδοσης του ασύρματου δικτύου δεδομένων. Τέλος η ασύρματη επικοινωνία μεταξύ δύο κινητών σταθμών δεν είναι πάντα αμφίδρομη, δηλαδή το γεγονός ότι ένας σταθμός επικοινωνεί ασύρματα με έναν άλλον δεν σημαίνει ότι ισχύει και το αντίστροφο. Αυτό μπορεί να οφείλεται στο διαφορετικό εύρος μετάδοσης των δύο σταθμών, στην διαφορετική ευαισθησία των αντίστοιχων δεκτών, σε παρεμβολές καθώς και σε άλλους παράγοντες.

## 1.4 Συνεισφορά της διπλωματικής

Στα πλαίσια της διπλωματικής ζητήθηκε η μελέτη, η κατανόηση και η κριτική περιγραφή της σχετικής έρευνας που αφορά τα πρωτόκολλα δρομολόγησης για ad-hoc κινητά δίκτυα. Παράλληλα ζητήθηκε η υλοποίηση των πρωτοκόλλων Dynamic Source Routing (DSR) και Zone Routing Protocol (ZRP), με στόχο την πειραματική αξιολόγηση τους με τη βοήθεια ενός περιβάλλοντος προσομοίωσης.

Στην διπλωματική αρχικά εξηγούμε το τι είναι ακριβώς τα ad-hoc δίκτυα, μιλώντας ακόμα για τις εφαρμογές τους αλλά και για τις τεχνολογίες που τα υποστηρίζουν. Στην συνέχεια περιγράφουμε τα χαρακτηριστικά των πρωτοκόλλων δρομολόγησης για ad-hoc δίκτυα, αναφερομένοι στις κατηγορίες στα οποία αυτά χωρίζονται αλλά στις ιδιότητες που είναι γενικά επιθυμητό αυτά να έχουν. Ακόμα περιγράφουμε εν συντομία τα σημαντικότερα πρωτόκολλα δρομολόγησης για ad-hoc δίκτυα. Αναλυτικά εξετάζουμε δύο πρωτόκολλα δρομολόγησης, το Dynamic Source Routing (DSR) και το Zone Routing Protocol (ZRP), τα οποία και υλοποιήσαμε. Περιγράφουμε βέβαια και τα σημαντικότερα χαρακτηριστικά αυτών των υλοποιήσεων. Στη συνέχεια παρουσιάζουμε τα αποτελέσματα των πειραμάτων μας με αυτά τα πρωτόκολλα, πειράματα τα οποία πραγματοποιήθηκαν με τη βοήθεια ενός περιβάλλοντος προσομοίωσης. Ακόμα παρουσιάζουμε τα αποτελέσματα των πειραμάτων μας και με ένα τρίτο πρωτόκολλο δρομολόγησης, το Support Routing Protocol (SRP). Με τα πειράματα αυτά μπορούμε και συγκρίνουμε τα τρία πρωτόκολλα. Τέλος υλοποιήσαμε και εκτελέσαμε πειράματα με συνδυασμούς των τριών αυτών πρωτοκόλλων, προσπαθώντας να βελτιώσουμε την συμπεριφορά τους. Παρουσιάζουμε τα αποτελέσματα και αυτών των πειραμάτων.



## 1.5 Οργάνωση της διπλωματικής

Η υπόλοιπη διπλωματική οργανώνεται ως εξής. Στο Κεφάλαιο 2 γίνεται αναφορά στα χαρακτηριστικά και στις επιθυμητές ιδιότητες των πρωτοκόλλων δρομολόγησης για τα ad-hoc δίκτυα. Στο Κεφάλαιο 3 παρουσιάζονται τα σημαντικότερα πρωτόκολλα δρομολόγησης. Στα Κεφάλαια 4 και 5 περιγράφονται οι υλοποιήσεις των πρωτοκόλλων DSR και ZRP. Τέλος στα Κεφάλαια 6, 7, 8 και 9 παρουσιάζονται τα αποτελέσματα των πειραμάτων μας καθώς και τα συμπεράσματα μας.



## Κεφάλαιο 2

# Χαρακτηριστικά των πρωτοκόλλων δρομολόγησης για ad-hoc δίκτυα

### 2.1 Γράφημα συνδεσιμότητας των πρωτοκόλλων δρομολόγησης

Για την περιγραφή της λειτουργίας των πρωτοκόλλων δρομολόγησης για τα ad-hoc δίκτυα αλλά και για την αξιολόγηση τους χρησιμοποιούνται βασικές έννοιες της θεωρίας των γράφων. Συγκεκριμένα ένα ad-hoc δίκτυο μοντελοποιείται σαν ένα γράφημα το οποίο αναπαριστά την συνδεσιμότητα του δικτύου. Οι κόμβοι του γραφήματος αυτού αναπαριστούν τους κινητούς σταθμούς, ενώ οι ακμές του αναπαριστούν την δυνατότητα της άμεσης ασύρματης επικοινωνίας μεταξύ των αντίστοιχων κινητών σταθμών. Καθώς οι σταθμοί κινούνται η τοπολογία του δικτύου αλλάζει μαζί και το αντίστοιχο γράφημα συνδεσιμότητας του.

### 2.2 Κατηγορίες των πρωτοκόλλων δρομολόγησης

#### 2.2.1 Γενικά

Τα πρωτόκολλα δρομολόγησης για ad-hoc δίκτυα συντελούν στην επίτευξη της επικοινωνίας μεταξύ κόμβων του δικτύου που δεν έχουν άμεση ασύρματη επικοινωνία. Αυτό γενικά επιτυγχάνεται με την χρήση άλλων κόμβων του δικτύου που μεταφέρουν τα πακέτα δεδομένων από τον αρχικό στον τελικό κόμβο.

Τα πρωτόκολλα αυτά μπορούν να χωριστούν σε δύο κατηγορίες, στα πρωτόκολλα που στηρίζονται στην εύρεση διαδρομών από κόμβους και στα πρωτόκολλα που στηρίζονται στην κίνηση των κόμβων. Τα περισσότερα πρωτόκολλα δρομολόγησης για ad-hoc δίκτυα λειτουργούν στο επίπεδο του δικτύου (network layer), ενώ άλλα λειτουργούν ή μπορούν να λειτουργήσουν και στο επίπεδο σύνδεσης δεδομένων (data link layer). Έτσι όλοι οι κόμβοι του δικτύου έχουν ένα μοναδικό προσδιοριστικό, μία IP ή μία MAC διεύθυνση ανάλογα.

### 2.2.2 Πρωτόκολλα δρομολόγησης που στηρίζονται στην εύρεση διαδρομών από κόμβους

Με την χρήση πρωτοκόλλων αυτής της κατηγορίας οι κόμβοι του δικτύου μπορούν να ανακαλύπτουν διαδρομές επικοινωνίας προς άλλους κόμβους με τους οποίους δεν έχουν άμεση ασύρματη επικοινωνία. Οι διαδρομές αυτές αποτελούνται από κόμβους όπου κάθε ένας μπορεί να επικοινωνήσει απευθείας ασύρματα με τον επόμενο κόμβο της διαδρομής. Τα πακέτα δεδομένων μεταφέρονται στη διαδρομή αυτή από τον έναν κόμβο στον επόμενο φτάνοντας έτσι από τον αρχικό στον τελικό κόμβο. Ένα τέτοιο πρωτόκολλο εκτελείται κατανεμημένα στους κόμβους του δικτύου. Δηλαδή οι κόμβοι μετέχουν ισότιμα στην λειτουργία του, δημιουργώντας, εξετάζοντας και προωθώντας πακέτα του πρωτοκόλλου ή πακέτα δεδομένων. Ακόμα επειδή η κίνηση των κόμβων έχει ως αποτέλεσμα την διάσπαση των διαδρομών και την δημιουργία νέων τα πρωτόκολλα θα πρέπει να φροντίζουν και για την συντήρηση των διαδρομών επικοινωνίας. Τα πρωτόκολλα αυτά χρησιμοποιούνται κυρίως στην περίπτωση που τα αντίστοιχα ad-hoc δίκτυα καλύπτουν μία μικρή περιοχή ή στην περίπτωση που τα δίκτυα αυτά είναι πολύ πυκνά, εξασφαλίζοντας έτσι την δυνατότητα εύρεσης διαδρομών μεταξύ απομακρυσμένων κόμβων στην πλειονότητα των περιπτώσεων.

Τα πρωτόκολλα δρομολόγησης αυτής της κατηγορίας μπορεί να ταξινομηθούν στα καθοδηγούμενα από πίνακα (proactive) και στα καθοδηγούμενα από απαιτήσεις αποστολής (reactive), χωρίς βέβαια να λείπουν και οι ενδιάμεσες λύσεις. Στα πρωτόκολλα καθοδηγούμενα από πίνακα κάθε κόμβος γνωρίζει μία τουλάχιστον διαδρομή προς κάθε άλλον κόμβο του δικτύου ακόμα και αν δεν χρειάζεται άμεσα αυτές τις διαδρομές για την μεταφορά πακέτων δεδομένων προς τους αντίστοιχους προορισμούς. Οι πληροφορίες αυτές δρομολόγησης διατηρούνται σε πίνακες σε κάθε κόμβο του δικτύου. Το πλεονέκτημα των πρωτοκόλλων αυτών είναι ότι όταν ένας κόμβος χρειαστεί να στείλει πακέτα δεδομένων σε κάποιον άλλον κόμβο του δικτύου τότε μία τουλάχιστον διαδρομή προς αυτόν είναι γνωστή και τα πακέτα δεδομένων μεταφέρονται χωρίς καμμία καθυστέρηση. Ωστόσο επειδή στα ad-hoc δίκτυα καθώς οι κόμβοι κινούνται

πολλές διαδρομές διασπώνται αλλά και νέες δημιουργούνται θα πρέπει οι κόμβοι συνεχώς να συντηρούν τις διαδρομές που γνωρίζουν. Για τον σκοπό αυτό στέλνονται από τους κόμβους του δικτύου επιπλέον πακέτα ελέγχου, τα οποία χρησιμοποιούνται για την συντήρηση των διαφόρων διαδρομών που οι κόμβοι γνωρίζουν ακόμα και αν αυτές δεν χρησιμοποιούνται. Το πλήθος αυτών των πακέτων ελέγχου είναι σε αναλογία με την κινητικότητα των κόμβων του δικτύου. Τα αποτελέσματα των πολλών πακέτων ελέγχου είναι η μείωση του εύρους ζώνης που διατίθεται για την μετάδοση των πακέτων δεδομένων, η συμφόρηση σε ενδιάμεσους κόμβους του δικτύου αλλά και η υπερβολική κατανάλωση των πόρων των κόμβων (ενεργειακών, υπολογιστικών και μνήμης). Επιπλέον είναι δυνατόν, λόγω της συνεχής κίνησης των κόμβων του δικτύου, σε έναν κόμβο να φτάσουν πακέτα ελέγχου τα οποία να μεταφέρουν πληροφορία η οποία δεν είναι πια έγκυρη. Αυτό θα έχει σαν πιθανό αποτέλεσμα την δημιουργία βρόγχων δρομολόγησης και μη έγκυρων διαδρομών. Από την άλλη πλευρά στα καθοδηγούμενα από απαιτήσεις αποστολής πρωτόκολλα δρομολόγησης ένας κόμβος ξεκινά μία διαδικασία ανακάλυψης διαδρομής προς κάποιον άλλον κόμβο του δικτύου όταν χρειάζεται να στείλει πακέτα δεδομένων προς αυτόν και μόνο τότε. Ακόμα από τη στιγμή που θα βρεθεί μία διαδρομή αυτή θα πρέπει να συντηρείται, αντιμετωπίζοντας έτσι πιθανές διασπάσεις της λόγω κίνησης των κόμβων του δικτύου. Η διαδρομή αυτή θα συντηρείται όσο ο κόμβος πηγή τη χρησιμοποιεί για την μεταφορά πακέτων δεδομένων ή μέχρι η συντήρηση της να γίνει αδύνατη λόγω π.χ. διάσπασης του δικτύου. Τα πρωτόκολλα αυτά χρησιμοποιούν λιγότερα πακέτα ελέγχου, ωστόσο υπάρχει καθυστέρηση στην έναρξη αποστολής πακέτων δεδομένων στον προορισμό μέχρι να ανακαλυφθεί μία διαδρομή προς αυτόν.

Ακόμα στα πρωτόκολλα δρομολόγησης αυτής της κατηγορίας η πληροφορία δρομολόγησης (οι διαδρομές) που το πρωτόκολλο συγκεντρώνει και αποθηκεύει σε κάθε κόμβο μπορεί να έχει δύο μορφές. Στην πρώτη περίπτωση το πρωτόκολλο αποθηκεύει σε κάθε κόμβο πληροφορία που προσδιορίζει τον γειτονικό κόμβο στον οποίον θα πρέπει να προωθηθούν τα πακέτα δεδομένων προς έναν συγκεκριμένο προορισμό (next hop routing). Στην δεύτερη περίπτωση το πρωτόκολλο αποθηκεύει σε κάθε κόμβο την πλήρη διαδρομή προς έναν συγκεκριμένο προορισμό και τα πακέτα δεδομένων που ξεκινούν από κάποιον κόμβο περιέχουν την πλήρη διαδρομή προς τον προορισμό (source routing). Το next hop routing πλεονεκτεί έναντι του source routing μιας και το μέγεθος των πακέτων ελέγχου αλλά και δεδομένων είναι μικρότερο όπως επίσης και το μέγεθος του πίνακα δρομολόγησης σε κάθε κόμβο.

Τέλος πολλά πρωτόκολλα δρομολόγησης αυτής της κατηγορίας θεωρούν πως σε κάθε περίπτωση που δύο κόμβοι επικοινωνούν ασύρματα απευθείας μπορούν να επικοινωνούν αμφίδρομα, θεώρηση που στην πραγματικότητα δεν είναι πάντα σωστή. Υπάρχουν ωστόσο και πρωτόκολλα δρομολόγησης που δεν

κάνουν αυτήν την θεώρηση, αποφεύγοντας να χρησιμοποιούν στις διαδρομές που ανακαλύπτουν ζευγάρια κόμβων που επικοινωνούν μονόδρομα. Ενώ υπάρχουν και πρωτόκολλα δρομολόγησης που ομοίως δεν κάνουν αυτήν την θεώρηση, αλλά ανακαλύπτουν ακόμα και διαδρομές που περιέχουν ζευγάρια κόμβων που επικοινωνούν μονόδρομα. Η λειτουργία αυτών των πρωτοκόλλων είναι σαφώς πιο περίπλοκη. Η μονόδρομη επικοινωνία μεταξύ δύο κόμβων είναι πολλές φορές ένδειξη της χαμηλής ποιότητας ασύρματης επικοινωνίας μεταξύ αυτών, επομένως η χρήση τέτοιων ζευγαριών κόμβων στις ανακαλυφθέντες διαδρομές επικοινωνίας περιέχει τον κίνδυνο γρήγορης διάσπασης αυτών και την ανάγκη εκκίνησης νέων διαδικασιών ανακάλυψης διαδρομής. Από την άλλη μεριά η χρήση τέτοιων ζευγαριών μπορεί να οδηγήσει στην εύρεση, από το πρωτόκολλο, πιο σύντομων διαδρομών ή ακόμα και διαδρομών προς έναν κόμβο στην περίπτωση που δεν υπάρχουν ανάλογες διαδρομές με μόνο ζευγάρια κόμβων που επικοινωνούν αμφίδρομα.

### 2.2.3 Πρωτόκολλα δρομολόγησης που στηρίζονται στην κίνηση των κόμβων

Στην περίπτωση που τα ad-hoc δίκτυα εκτείνονται σε μία μεγάλη περιοχή μπορεί να μην είναι δυνατή η εύρεση διαδρομών μεταξύ κόμβων που δεν έχουν άμεση ασύρματη επικοινωνία. Ακόμα στην περίπτωση που μία διαδρομή έχει μεγάλο μήκος τότε η κίνηση των κόμβων κάνει πολύ πιθανή την γρήγορη διάσπασή της. Λύση στα παραπάνω προβλήματα έρχονται να δώσουν τα πρωτόκολλα δρομολόγησης που στηρίζονται στη, τυχαία ή μη, κίνηση των κόμβων για τη μεταφορά των διαφόρων πακέτων. Δηλαδή δύο κόμβοι ανταλλάσσουν πακέτα όταν συναντώνται. Τα πρωτόκολλα αυτά χωρίζονται στα μη-εξαναγκαστικά, στα εξαναγκαστικά και στα ημί-εξαναγκαστικά πρωτόκολλα δρομολόγησης.

Η εκτέλεση των μη-εξαναγκαστικών πρωτοκόλλων δεν επιρεάζει την κίνηση των κόμβων του δικτύου, η οποία είναι τυχαία. Η μεταφορά πακέτων δεδομένων μεταξύ δύο κόμβων του δικτύου μπορεί να γίνει μόνο όταν αυτοί οι δύο κόμβοι βρεθούν τυχαία σε απόσταση που να επιτρέπει την άμεση ασύρματη επικοινωνία τους. Βέβαια μιας και η κίνηση των κόμβων είναι τυχαία είναι πολύ πιθανό δύο κόμβοι που θέλουν να επικοινωνήσουν να μην μπορέσουν.

Αντίθετα τα εξαναγκαστικά πρωτόκολλα απαιτούν για την λειτουργία τους όλοι οι κόμβοι του δικτύου να εκτελούν συγκεκριμένες κινήσεις. Με τον τρόπο αυτό εξασφαλίζεται ότι οι κόμβοι που θέλουν να επικοινωνήσουν κάποια στιγμή θα βρεθούν σε απόσταση που να επιτρέπει την άμεση ασύρματη επικοινωνία τους.

Τέλος τα ημί-εξαναγκαστικά πρωτόκολλα είναι η ενδιάμεση περίπτωση και απαιτούν μόνο ένα μέρος των κόμβων να κινούνται με ένα συγκεκριμένο τρόπο,

ενώ η κίνηση των υπολοίπων είναι τυχαία. Οι κόμβοι που κινούνται με ένα συγκεκριμένο τρόπο καλύπτουν γρήγορα ολόκληρη την περιοχή του δικτύου και αναλαμβάνουν να μεταφέρουν πακέτα δεδομένων μεταξύ κόμβων του δικτύου που κινούνται τυχαία και που δεν έχουν άμεση ασύρματη επικοινωνία. Η ομάδα των κόμβων οι οποίοι κινούνται με ένα συγκεκριμένο τρόπο ονομάζεται υποστήριξη, ενώ το κομμάτι του ημί-εξαναγκαστικού πρωτοκόλλου που καθορίζει τον τρόπο με τον οποίον τα μέλη της υποστήριξης θα κινούνται και θα επικοινωνούν ονομάζεται πρωτόκολλο διαχείρισης της υποστήριξης. Το άλλο κομμάτι του ημί-εξαναγκαστικού πρωτοκόλλου αφορά τους κόμβους που δεν ανήκουν στην υποστήριξη και στον τρόπο με τον οποίον αυτοί διαχειρίζονται τα πακέτα που θέλουν να στείλουν σε κάποιον άλλο κόμβο του δικτύου.

## 2.3 Επιθυμητές ιδιότητες των πρωτοκόλλων δρομολόγησης

Τα ad-hoc δίκτυα παρουσιάζουν ιδιαιτερότητες που θα πρέπει να λαμβάνονται υπόψη από τα αντίστοιχα πρωτόκολλα δρομολόγησης. Καταρχήν οι κόμβοι ενός ad-hoc δικτύου έχουν συνήθως περιορισμένες υπολογιστικές, ενεργειακές αλλά και αποθηκευτικές δυνατότητες, επομένως τα πρωτόκολλα δρομολόγησης θα πρέπει να χρειάζονται για την λειτουργία τους λίγους ή μικρούς υπολογισμούς, λίγες μεταδόσεις πακέτων ελέγχου αλλά και την ελάχιστη δυνατή αποθήκευση πληροφοριών δρομολόγησης σε κάθε κόμβο. Ακόμα η μονόδρομη επικοινωνία μεταξύ κόμβων του δικτύου θα πρέπει να λαμβάνεται υπόψη από τα πρωτόκολλα. Παράλληλα η συνεχής κίνηση των κόμβων και η δημιουργία αλλά και η διάσπαση διαδρομών θα πρέπει να αντιμετωπίζονται από το πρωτόκολλο με τρόπο που να μην δημιουργεί πολλά νέα πακέτα ελέγχου. Τα πολλά πακέτα ελέγχου καταναλώνουν το περιορισμένο ούτως ή άλλως εύρος ζώνης του ασύρματου μέσου. Τέλος θα πρέπει η εύρεση νέων διαδρομών, από τα αντίστοιχα πρωτόκολλα ύστερα από μία τοπολογική αλλαγή, να γίνεται σύντομα (χρόνος σύγκλισης) και σίγουρα πριν την επόμενη τοπολογική αλλαγή. Σε διαφορετική περίπτωση δεν θα είναι δυνατή ποτέ η οποιαδήποτε επικοινωνία.

## 2.4 Η σχέση με τα πρωτόκολλα δρομολόγησης για τα ενσύρματα δίκτυα

Τα πρωτόκολλα δρομολόγησης για τα ενσύρματα δίκτυα εκτελούν την ίδια βασική λειτουργία με τα πρωτόκολλα για τα ad-hoc δίκτυα, δηλαδή παρέχουν στους κόμβους του δικτύου την απαραίτητη βοήθεια για την δρομολόγηση των

πακέτων δεδομένων στον τελικό προορισμό τους.

Βέβαια τα πρωτόκολλα αυτά δεν έχουν σχεδιαστεί για το περιβάλλον των ad-hoc δικτύων. Συγκεκριμένα τα πρωτόκολλα αυτά καταναλώνουν για την λειτουργία τους πολλούς πόρους (υπολογιστικούς, ενεργειακούς και αποθηκευτικούς). Άλλωστε στα ενσύρματα δίκτυα μόνο ορισμένες ειδικές συσκευές (π.χ. routers) εκτελούν τα πρωτόκολλα αυτά, ενώ οι υπόλοιποι κόμβοι απλώς δρομολογούν τα πακέτα δεδομένων τους μέσω αυτών. Ακόμα τα πρωτόκολλα αυτά αντιδρούν με πλήθος πακέτων ελέγχου σε περίπτωση αλλαγής της τοπολογίας του δικτύου, ενώ ο χρόνος σύγκλισης μπορεί να είναι αρκετά μεγάλος. Παράλληλα το ασύρματο μέσο παρουσιάζει πολλές ιδιαιτερότητες και μειονεκτήματα σε σχέση με τον ενσύρματο μέσο, όπως την μονόδρομη επικοινωνία μεταξύ δύο κόμβων και το μικρό εύρος ζώνης.

Πολλά πρωτόκολλα δρομολόγησης για τα ad-hoc δίκτυα σχεδιάστηκαν έχοντας ως βάση τα αντίστοιχα πρωτόκολλα των ενσύρματων δικτύων, χρησιμοποιώντας κάποια χαρακτηριστικά τους και προσαρμόζοντάς αυτά στο περιβάλλον των ad-hoc δικτύων.



## Κεφάλαιο 3

# Πρωτόκολλα δρομολόγησης για ad-hoc δίκτυα

### 3.1 Γενικά

Ερευνητές από όλο το κόσμο έχουν προτείνει διάφορα πρωτόκολλα δρομολόγησης για τα ad-hoc δίκτυα. Ορισμένα μάλιστα από αυτά έχουν κατατεθεί για αξιολόγηση στην αντίστοιχη επιτροπή της IETF (Internet Engineering Task Force), MANET (Mobile Ad-hoc Networking). Παρακάτω περιγράφονται τα σημαντικότερα πρωτόκολλα δρομολόγησης για ad-hoc δίκτυα που έχουν προταθεί.

### 3.2 Dynamic Source Routing (DSR)

#### 3.2.1 Γενικά

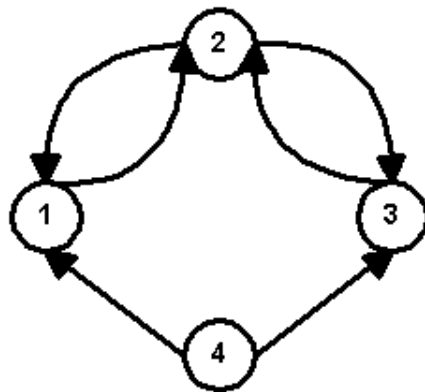
Το DSR είναι ένα πρωτόκολλο δρομολόγησης καθοδηγούμενο από απαιτήσεις αποστολής που βασίζεται στις διαδικασίες της ανακάλυψης και της συντήρησης διαδρομών και το οποίο χρησιμοποιεί source routing. Το DSR είναι ένα πρωτόκολλο που υλοποιείται στο επίπεδο δικτύου του μοντέλου αναφοράς δικτυακών πρωτοκόλλων ISO/OSI.

Μέσω της διαδικασίας Ανακάλυψης Διαδρομής (ΑΔ) ένας κόμβος του δικτύου, ο οποίος θέλει να στείλει πακέτα δεδομένων σε έναν προορισμό, μαθαίνει μία πλήρη διαδρομή (ΠΔ) προς τον προορισμό. Μέσω της διαδικασίας Συντήρησης Διαδρομής (ΣΔ) ενημερώνεται ένας κόμβος του δικτύου, ο οποίος χρησιμοποιεί μια ΠΔ για να στέλνει πακέτα δεδομένων σε έναν προορισμό, ότι κάποια ακμή αυτής της ΠΔ δεν λειτουργεί. Η αλλαγή αυτή στην τοπολογία του δικτύου προέρχεται κατά κύριο λόγο από την κίνηση και απομάκρυνση

των κόμβων στα άκρα της ακμής. Και οι δύο αυτές διαδικασίες λειτουργούν απολύτως δυναμικά, δηλαδή λειτουργούν είτε για την ανακάλυψη νέων ΠΔ όταν υπάρχει άμεση ανάγκη για επικοινωνία είτε για την συντήρηση ΠΔ που ήδη χρησιμοποιούνται για επικοινωνία.

### 3.2.2 Το ασύρματο ad-hoc δίκτυο του DSR

Στο δίκτυο στο οποίο λειτουργεί ο DSR η επικοινωνία των κόμβων γίνεται μόνο ασύρματα, χωρίς δηλαδή της χρήση οποιασδήποτε σταθερής καλωδιακής υποδομής. Οι γείτονες ενός κόμβου στο ασύρματο αυτό ad-hoc δίκτυο είναι όλοι εκείνοι με τους οποίους ο κόμβος μπορεί να επικοινωνήσει απευθείας ασύρματα, χωρίς την βοήθεια οποιουδήποτε τρίτου σταθμού. Προφανώς το πλήθος των γειτόνων ενός κόμβου εξαρτάται από το εύρος μετάδοσης του κόμβου αυτού. Επομένως στο γράφημα συνδεσιμότητας που αναπαριστά το δίκτυο στο οποίο λειτουργεί ο DSR, μία ακμή μεταξύ δύο κόμβων αναπαριστά ακριβώς την δυνατότητα τους για άμεση ασύρματη επικοινωνία. Το σχήμα 3.1 δείχνει έναν πιθανό γράφημα συνδεσιμότητας.



Σχήμα 3.1: Γράφημα συνδεσιμότητας δικτύου για το DSR

Κάθε κόμβος στο δίκτυο, στο οποίο λειτουργεί ο DSR, έχει ένα μοναδικό προσδιοριστικό π.χ. μία μοναδική IP διεύθυνση. Ακόμα κάθε κόμβος το δίκτυο λειτουργεί σε μία από δύο καταστάσεις (promiscuous ή όχι). Η διαφορά τους βρίσκεται στο κατά πόσο ο κόμβος διαβάζει ή όχι ένα πακέτο το οποίο δεν απευθύνεται σε αυτόν, αλλά το οποίο τύχει να ακούσει. Παράλληλα κάθε κόμβος του δικτύου μετέχει στο πρωτόκολλο DSR κάθε στιγμή, με την έννοια ότι θα πρέπει να προωθεί πακέτα για άλλους κόμβους όταν κάτι τέτοιο του ζητηθεί. Ακόμα κάθε κόμβος του ασύρματου ad-hoc δικτύου μπορεί να κινείται οποιαδήποτε στιγμή χωρίς καμία ειδοποίηση. Η κίνηση αυτή μπορεί να

είναι τυχαία και δεν εξαρτάται από την κίνηση ή την μη κίνηση άλλων κόμβων. Η κίνηση βέβαια αυτή των κόμβων μπορεί να έχει ως αποτέλεσμα την είσοδο ή την έξοδο ενός κόμβου από το δίκτυο ή ακόμα και την διάσπαση του δικτύου.

Γενικά η ασύρματη επικοινωνία μεταξύ δύο κόμβων δεν είναι πάντα δυνατή και προς τις δύο κατευθύνσεις. Δηλαδή το ότι ένας κόμβος μπορεί να επικοινωνήσει ασύρματα απευθείας με έναν άλλον κόμβο δεν σημαίνει απαραίτητα ότι ισχύει και το αντίστροφο. Έτσι στο γράφημα συνδεσιμότητας, που αναπαριστά το ασύρματο δίκτυο, δεν είναι όλες οι ακμές διπλής κατεύθυνσεως, υπάρχουν και ακμές μονής κατεύθυνσεως. Σε αντίθεση με πολλά άλλα πρωτόκολλα δρομολόγησης ο DSR μπορεί να δρομολογεί πακέτα πάνω από διαδρομές που περιέχουν ακμές μονής ή διπλής κατεύθυνσης. Η χρήση ακμών μονής κατεύθυνσης μπορεί να οδηγήσει στην ανακάλυψη από τον DSR πιο αποδοτικών διαδρομών ή ακόμα και διαδρομών προς έναν κόμβο στην περίπτωση που δεν υπάρχουν ανάλογες διαδρομές με μόνο ακμές διπλής κατεύθυνσης.

### 3.2.3 Διαδικασία Ανακάλυψης Διαδρομής (ΑΔ)

Η διαδικασία ΑΔ χρησιμοποιείται όταν ένας κόμβος θέλει να στείλει πακέτα δεδομένων σε έναν άλλο κόμβο του δικτύου. Ο κόμβος για να μπορέσει να στείλει τα δεδομένα αυτά θα πρέπει να βρει μία ΠΔ προς τον κόμβο - προορισμό.

Πριν ωστόσο ο κόμβος - αφετηρία ξεκινήσει την διαδικασία ΑΔ ψάχνει στην Μνήμη Προσωρινής Αποθήκευσης Διαδρομών (ΜΠΑΔ) του για μια τέτοια ΠΔ. Η ΜΠΑΔ είναι μία δομή που βρίσκεται σε κάθε κόμβο του δικτύου και χρησιμοποιείται για την αποθήκευση ΠΔ που ο κόμβος έχει μάθει είτε μέσω της διαδικασίας ΑΔ είτε διαβάζοντας την πληροφορία ΠΔ που περιέχεται σε άλλα πακέτα που ο κόμβος προωθεί ή που απλώς τύχει να ακούσει. Μια ΜΠΑΔ μπορεί να έχει την παρακάτω δομή:

ΠΔ 1
ΠΔ 2
...
ΠΔ n

Πίνακας 3.1: Δομή μίας ΜΠΑΔ

Μία ΜΠΑΔ μπορεί, δηλαδή, να είναι ένας πίνακας ΠΔ προς διάφορους προορισμούς.

Στην περίπτωση που ο κόμβος δεν βρει κάτι στην ΜΠΑΔ του, τότε ξεκινά την διαδικασία ΑΔ με σκοπό την ανακάλυψη μίας ΠΔ προς τον κόμβο - προορισμό. Κατά την διαδικασία της ΑΔ ο κόμβος - αφετηρία δημιουργεί ένα μήνυμα Αίτησης Διαδρομής (ΑΙΔ) το οποίο και στέλνει σε όλους του-

ς γειτονικούς του κόμβους (broadcast), δηλαδή σε όλους τους κόμβους που βρίσκονται μέσα στην εμβέλεια του.

Μια ΑΙΔ μπορεί να έχει την παρακάτω δομή:

Αφετηρία	Προορισμός
Αριθμός Αίτησης	Ενδιάμεσοι Κόμβοι

Πίνακας 3.2: Δομή μίας ΑΙΔ

Στην δομή αυτή η Αφετηρία είναι το προσδιοριστικό (π.χ. IP) του κόμβου - αφετηρία που δημιουργήσε και έστειλε για πρώτη φορά αυτό το μήνυμα και ο οποίος θέλει να μάθει, μέσω της διαδικασίας ΑΔ, μια ΠΔ προς τον κόμβο - προορισμό με προσδιοριστικό το Προορισμός. Κάθε τέτοιο μήνυμα έχει έναν Αριθμό Αίτησης ο οποίος τίθεται από τον κόμβο - αφετηρία. Αυτός ο Αριθμός Αίτησης σε συνδυασμό με την Αφετηρία προσδιορίζουν μοναδικά το συγκεκριμένο μήνυμα ΑΙΔ. Ακόμα κάθε ΑΙΔ περιέχει το πεδίο Ενδιάμεσοι Κόμβοι, όπου καταγράφονται με σειρά τα προσδιοριστικά όλων των ενδιάμεσων κόμβων που έλαβαν και στην συνέχεια προώθησαν αυτό το συγκεκριμένο μήνυμα ΑΙΔ.

Όταν ένας κόμβος του δικτύου λάβει μία ΑΙΔ αρχικά την ελέγχει για να διαπιστώσει μήπως έχει λάβει, σε μία προηγούμενη χρονική στιγμή, μία άλλη ΑΙΔ από τον ίδιο κόμβο - αφετηρία και με τον ίδιο Αριθμό Αίτησης. Ακόμα ελέγχει μήπως το δικό του μοναδικό προσδιοριστικό υπάρχει ανάμεσα στα προσδιοριστικά των κόμβων του πεδίου Ενδιάμεσοι Κόμβοι, ελέγχοντας έτσι την ύπαρξη κύκλου. Σε περίπτωση που κάποιος από αυτούς τους ελέγχους αποβεί θετικός τότε ο κόμβος απορρίπτει την αντίστοιχη ΑΙΔ. Αν οι έλεγχοι είναι όλοι αρνητικοί και ο κόμβος που έλαβε την ΑΙΔ είναι ο κόμβος - προορισμός τότε αυτός επιστρέφει στον κόμβο - αφετηρία μία ΑΠάντηση Διαδρομής (ΑΠΔ). Στην περίπτωση που ο κόμβος αυτός δεν είναι ο κόμβος - προορισμός τότε απλώς προσθέτει το προσδιοριστικό του στο πεδίο Ενδιάμεσοι Κόμβοι και προωθεί (broadcast) την ΑΙΔ στους γείτονες του.

Μια ΑΠΔ μπορεί να έχει την παρακάτω δομή:

Αφετηρία	Προορισμός
Αριθμός Αίτησης	ΠΔ
Επόμενος Κόμβος	ΠΔ Επιστροφής

Πίνακας 3.3: Δομή μίας ΑΠΔ

Στη δομή αυτή τα πεδία Αφετηρία και Προορισμός είναι τα προσδιοριστικά του κόμβου που έστειλε και του κόμβου που θα πρέπει τελικά να λάβει αυτή την ΑΠΔ. Ο Αριθμός Αίτησης έχει την ίδια τιμή με το αντίστοιχο πεδίο του μηνύματος ΑΙΔ που ο κόμβος - προορισμός έλαβε. Το πεδίο ΠΔ είναι η ΠΔ

που ανακαλύφθηκε από την διαδικασία ΑΔ και υπήρχε στο πεδίο Ενδιάμεσοι Κόμβοι της ΑΙΔ που ο κόμβος - προορισμός έλαβε. Ο Επόμενος Κόμβος είναι το προσδιοριστικό του κόμβου που θα πρέπει να λάβει το μήνυμα της ΑΠΔ στο επόμενο βήμα. Τέλος το πεδίο ΠΔ Επιστροφής είναι η ΠΔ που χρησιμοποιείται για την επιστροφή της ΑΠΔ στον κόμβο - αφετηρία.

Για την επιστροφή της ΑΠΔ στον κόμβο - αφετηρία, ο κόμβος - προορισμός αρχικά κοιτά στην ΜΠΑΔ του για μια κατάλληλη ΠΔ. Σε περίπτωση που βρει μια τέτοια ΠΔ την τοποθετεί στο πεδίο ΠΔ Επιστροφής της ΑΠΔ και στη συνέχεια στέλνει την ΑΠΔ στον Επόμενο Κόμβο με βάση αυτή την ΠΔ. Ομοίως κάθε ενδιάμεσος κόμβος που λαμβάνει την ΑΠΔ την προωθεί σύμφωνα με την ΠΔ Επιστροφής. Στην περίπτωση που μια τέτοια ΠΔ δεν βρεθεί στην ΜΠΑΔ τότε ο κόμβος - προορισμός ξεκινά μια καινούρια διαδικασία ΑΔ με στόχο την ανακάλυψη μια ΠΔ προς τον κόμβο - αφετηρία. Για την αποφυγή βέβαια ενός κύκλου από διαδικασίες ΑΔ ο κόμβος - προορισμός προσθέτει στο νέο μήνυμα ΑΙΔ την ΑΠΔ που ο ίδιος δημιούργησε. Σε περίπτωση που οι ακμές του δικτύου είναι διπλής κατεύθυνσης τότε ο κόμβος - προορισμός, αντί να κάνει κάποια από τις παραπάνω ενέργειες, μπορεί να συμπληρώσει το πεδίο ΠΔ Επιστροφής της ΑΠΔ αντιστρέφοντας απλώς την ΠΔ του πεδίου Ενδιάμεσοι Κόμβοι που υπήρχε στην ΑΙΔ που έλαβε. Στη συνέχεια ο κόμβος - προορισμός στέλνει την ΑΠΔ στον Επόμενο Κόμβο με βάση αυτή την αντεστραμμένη ΠΔ, ομοίως πράττει και κάθε ενδιάμεσος κόμβος που λαμβάνει αυτή την ΑΠΔ. Αν ένας κόμβος λάβει μία ΑΠΔ χωρίς αυτός να είναι ο Επόμενος Κόμβος τότε απλά απορρίπτει αυτή την ΑΠΔ.

Όταν τελικά η ΑΠΔ φτάσει στον κόμβο - αφετηρία, αυτός τοποθετεί την ΠΔ, που μόλις έλαβε, στην ΜΠΑΔ του. Στη συνέχεια στέλνει τα όποια πακέτα προς τον συγκεκριμένο κόμβο - προορισμό χρησιμοποιώντας αυτήν την ΠΔ, ενημερώνοντας ανάλογα την επικεφαλίδα του κάθε πακέτου. Κάθε ενδιάμεσος κόμβος προωθεί ένα πακέτο δεδομένων που λαμβάνει στον Επόμενο Κόμβο με βάση την ΠΔ που υπάρχει στην επικεφαλίδα του πακέτου. Αν ένας κόμβος λάβει ένα πακέτο δεδομένων χωρίς αυτός να είναι ο Επόμενος Κόμβος τότε απλά απορρίπτει το πακέτο αυτό.

Ένα πακέτο δεδομένων μπορεί να έχει την παρακάτω δομή:

Αφετηρία	Προορισμός
Επόμενος Κόμβος	ΠΔ
bit Επιβεβαίωσης	Δεδομένα

Πίνακας 3.4: Δομή ενός πακέτου δεδομένων

Στη δομή αυτή η Αφετηρία και ο Προορισμός είναι τα προσδιοριστικά του κόμβου που αρχικά έστειλε το πακέτο δεδομένων και του κόμβου που τελικά

θα πρέπει να λάβει αυτό το πακέτο δεδομένων. Το πεδίο Επόμενος Κόμβος είναι ο επόμενος ενδιάμεσος κόμβος που θα πρέπει να λάβει το πακέτο κατά την πορεία του προς τον προορισμό. Το πεδίο ΠΔ είναι η ΠΔ που χρησιμοποιείται για φτάσει το πακέτο δεδομένων στον προορισμό. Τέλος το bit Επιβεβαίωσης χρησιμοποιείται στην διαδικασία της ΣΔ.

Όταν ο κόμβος - αφετηρία δεν γνωρίζει μία ΠΔ προς έναν κόμβο - προορισμό, πέρα από το να ξεκινήσει την διαδικασία ΑΔ, αποθηκεύει το ή τα αντίστοιχα πακέτα δεδομένων σε μία προσωρινή μνήμη του, η οποία λέγεται Μνήμη Προσωρινής Αποθήκευσης Πακέτων (ΜΠΑΠ). Σε κάθε νέο πακέτο στην ΜΠΑΠ τοποθετείται και μια Χρονοσφραγίδα που δηλώνει την χρονική στιγμή που το πακέτο μπήκε σε αυτήν. Αν ύστερα από κάποιο χρονικό διάστημα δεν έχει βρεθεί ακόμα μια ΠΔ προς τον κόμβο - προορισμό του πακέτου τότε αυτό απορρίπτεται. Παράλληλα, μιας και η ΜΠΑΠ είναι πεπερασμένου μεγέθους, λειτουργεί και μία διαδικασία που φροντίζει να μην γεμίσει η ΜΠΑΠ (π.χ. FIFO). Προφανώς η διαδικασία αυτή δεν εξαρτάται από τις Χρονοσφραγίδες των πακέτων. Όταν τελικά φτάσει στον κόμβο - αφετηρία μια ΑΠΔ, αυτός αφαιρεί από την ΜΠΑΠ του τα αντίστοιχα πακέτα και τα στέλνει στον κόμβο - προορισμού τους χρησιμοποιώντας την ΠΔ της ΑΠΔ που έλαβε. Ακόμα κάθε κόμβος - αφετηρία ανά τακτά διαστήματα ξεκινά καινούριες διαδικασίες ΑΔ για πακέτα που παραμένουν στην ΜΠΑΠ του. Για την αποφυγή όμως εκκίνησης πολλών νέων διαδικασιών ΑΔ για τον ίδιο κόμβο - προορισμό, ο οποίος μπορεί να είναι για κάποιο λόγο (π.χ. διάσπαση δικτύου) μη προσιτός, ο κόμβος - αφετηρία διατηρεί έναν μηχανισμό exponential backoff. Ο μηχανισμός αυτός περιορίζει τον ρυθμό των νέων διαδικασιών ΑΔ για τον ίδιο κόμβο - προορισμό. Σε διαφορετική περίπτωση το πλήθος των διαφορετικών μηνυμάτων ΑΙΔ μπορεί να δημιουργούσε μεγάλη επιβάρυνση στο δίκτυο. Τέλος νέα πακέτα που πρέπει να σταλούν από τον κόμβο - αφετηρία προς τον ίδιο προορισμό αποθηκεύονται στην ΜΠΑΠ μέχρι να έρθει μια ΑΠΔ. Η εκκίνηση μίας νέας διαδικασίας ΑΔ για τον συγκεκριμένο κόμβο - προορισμό γίνεται πάλι μόνο όταν το επιτρέψει ο μηχανισμός του exponential backoff.

### 3.2.4 Διαδικασία Συντήρησης Διαδρομής (ΣΔ)

Η διαδικασία ΣΔ ενημερώνει έναν κόμβο του δικτύου που χρησιμοποιεί μια ΠΔ για να στέλνει πακέτα δεδομένων σε έναν κόμβο - προορισμό, ότι κάποια ακμή αυτής της ΠΔ δεν λειτουργεί. Η διαδικασία ΣΔ στηρίζεται σε έναν μηχανισμό αποστολής επιβεβαίωσης από έναν κόμβο που έλαβε ένα πακέτο δεδομένων προς τον κόμβο που του έστειλε αυτό το πακέτο δεδομένων. Συγκεκριμένα κάθε κόμβος που στέλνει ή απλώς προωθεί ένα πακέτο δεδομένων με βάση μια ΠΔ, περιμένει μετά την αποστολή του πακέτου την επιστροφή μιας επιβεβαίωσης από τον Επόμενο Κόμβο ότι το πακέτο έφτασε σε αυτόν. Μάλιστα το ίδιο

πακέτο δεδομένων ξαναστέλεται στον Επόμενο Κόμβο έναν μέγιστο αριθμό φορών μέχρι να έρθει αυτή η επιβεβαίωση λήψης.

Μια τέτοια επιβεβαίωση λήψης μπορεί να είναι έμμεση ή άμεση. Μια έμμεση επιβεβαίωση μπορεί να προέλθει όταν ο κόμβος που έστειλε ή προώθησε το πακέτο ακούσει τον Επόμενο Κόμβο, με βάση την ΠΔ, να προωθεί το ίδιο πακέτο δεδομένων. Αν ωστόσο μια τέτοια έμμεση επιβεβαίωση δεν είναι δυνατή π.χ. επειδή οι ακμές του ασύρματου δικτύου δεν είναι κατά ανάγκη διπλής κατεύθυνσης, ο κόμβος που στέλνει ή προωθεί ένα πακέτο δεδομένων μπορεί να θέσει το bit Επιβεβαίωσης στην επικεφαλίδα του πακέτου πριν το στείλει. Με τον τρόπο αυτό ο κόμβος ζητά από τον Επόμενο Κόμβο να του στείλει μία άμεση επιβεβαίωση λήψης του πακέτου (Acknowledgment). Αν η ακμή μεταξύ αυτών των δύο κόμβων είναι διπλής κατεύθυνσης τότε η επιβεβαίωση επιστρέφεται κατευθείαν στον κόμβο που έστειλε το πακέτο, σε διαφορετική περίπτωση η επιβεβαίωση μπορεί να μεταφερθεί μέσω μια διαφορετικής διαδρομής με πολλές ακμές. Ακόμα σε ορισμένες περιπτώσεις η επιβεβαίωση λήψης μπορεί να είναι μέρος του MAC πρωτοκόλλου (π.χ. IEEE 802.11).

Ένα κόμβος ξαναστέλνει ένα πακέτο δεδομένων στον Επόμενο Κόμβο, με βάση την ΠΔ, ένα μέγιστο αριθμό φορών μέχρι να λάβει μία επιβεβαίωση λήψης από αυτόν. Αν τελικά δεν έρθει μια τέτοια επιβεβαίωση ο κόμβος επιστρέφει ένα μήνυμα Λάθος Διαδρομής (ΛΔ) στον κόμβο - αφετηρία του πακέτου δεδομένων. Αυτό το μήνυμα ΛΔ περιέχει την ακμή του δικτύου πάνω από την οποία δεν ήταν δυνατό να προωθηθεί το πακέτο δεδομένων.

Ένα ΛΔ μπορεί να έχει την παρακάτω δομή:

Αφετηρία	Προορισμός
Επόμενος Κόμβος	ΠΔ Επιστροφής
Χαλασμένη Ακμή	-

Πίνακας 3.5: Δομή ενός ΛΔ

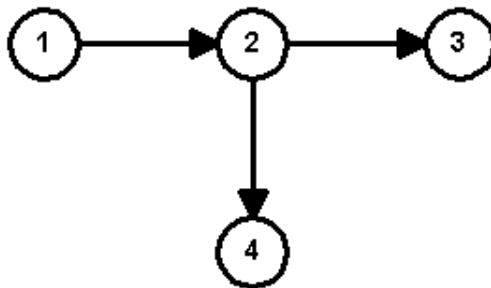
Στη δομή αυτή η Αφετηρία και ο Προορισμός είναι τα προσδιοριστικά του κόμβου που αρχικά έστειλε το ΛΔ και του κόμβου που τελικά θα πρέπει να λάβει αυτό το ΛΔ. Στη δομή αυτή το πεδίο Επόμενος Κόμβος είναι ο κόμβος που θα πρέπει να λάβει το μήνυμα αυτό, ενώ το πεδίο ΠΔ Επιστροφής είναι η ΠΔ που χρησιμοποιείται για την επιστροφή του μηνύματος ΛΔ στον κόμβο - αφετηρία που θέλουμε να ενημερωθεί για το πρόβλημα. Τέλος το πεδίο Χαλασμένη Ακμή δηλώνει την ακμή του δικτύου που δημιούργησε το πρόβλημα και είχε σαν αποτέλεσμα την μη μετάδοση ενός πακέτου δεδομένων σε έναν επόμενο κόμβο.

Όταν ο κόμβος - αφετηρία λάβει ένα τέτοιο μήνυμα τότε ενημερώνει κατάλληλα την ΜΠΑΔ του, αφαιρώντας όλες τις ΠΔ που περιέχουν την Χαλασμένη Ακ-

μή. Στην συνέχεια για την επαναμετάδοση των πακέτων δεδομένων ή για την μετάδοση νέων πακέτων στον ίδιο προορισμό ακολουθούνται τα γνωστά βήματα, δηλαδή πρώτα γίνεται αναζήτηση στην ΜΠΑΔ για μία άλλη ΠΔ και αν αυτή η αναζήτηση αποτύχει τότε και αν το επιτρέπει ο μηχανισμός του exponential backoff γίνεται η εκκίνηση μιας νέας διαδικασίας ΑΔ.

### 3.2.5 Βελτιστοποιήσεις και ειδικές περιπτώσεις για την διαδικασία ΑΔ

Ένας κόμβος που προωθεί ή απλώς τύχει να ακούσει ένα πακέτο δεδομένων ή ένα μήνυμα ΑΙΔ ή ΑΠΔ μπορεί να προσθέσει την ΠΔ, που αυτό περιέχει, στην ΜΠΑΔ του. Ο μόνος περιορισμός στην αποθήκευση τέτοιων διαδρομών είναι η πιθανή ύπαρξη στο δίκτυο ακμών μονής κατεύθυνσεως. Ένας κόμβος ο οποίος δεν γνωρίζει αν οι ακμές του δικτύου είναι ή όχι διπλής κατεύθυνσης δεν θα πρέπει να αποθηκεύει τέτοιες διαδρομές. Στο σχήμα 3.2 ο κόμβος 2 προωθεί πακέτα δεδομένων από τον 1 στον 3. Ο κόμβος 4 ακούγοντας ένα τέτοιο πακέτο δεδομένων μαθαίνει για την διαδρομή από τον κόμβο 2 στον κόμβο 3. Παρόλα αυτά δεν θα πρέπει να αποθηκεύσει την ΠΔ (4,2,3) στην ΜΠΑΔ του, μιας και η ακμή από τον κόμβο 2 στον κόμβο 4 είναι μονής κατεύθυνσης.



Σχήμα 3.2: Περιορισμοί στην αποθήκευση ΠΔ στην ΜΠΑΔ

Κάθε φορά που ένας κόμβος λαμβάνει μια ΑΙΔ για την οποία δεν είναι ο τελικός προορισμός μπορεί, πριν προωθήσει την ΑΙΔ, να ψάξει στην ΜΠΑΔ του για μία ΠΔ προς τον τελικό προορισμό. Σε περίπτωση που βρει μία τέτοια διαδρομή τότε επιστρέφει μία ΑΠΔ στον κόμβο - αφετηρία και δεν προωθεί την ΑΙΔ που έλαβε. Η ΠΔ που περιέχει αυτή η ΑΠΔ προκύπτει από την συνένωση δύο διαδρομών, της διαδρομής που ακολούθησε η ΑΙΔ μέχρι να φτάσει στον κόμβο αυτό και της διαδρομής προς τον τελικό προορισμό που ο κόμβος βρήκε στην ΜΠΑΔ του. Παρόλα αυτά κάθε τέτοιος κόμβος θα πρέπει, πριν επιστρέψει μία τέτοια ΑΠΔ, να ελέγξει αν η ΠΔ που περιέχεται στην ΑΠΔ έχει επαναλαμ-



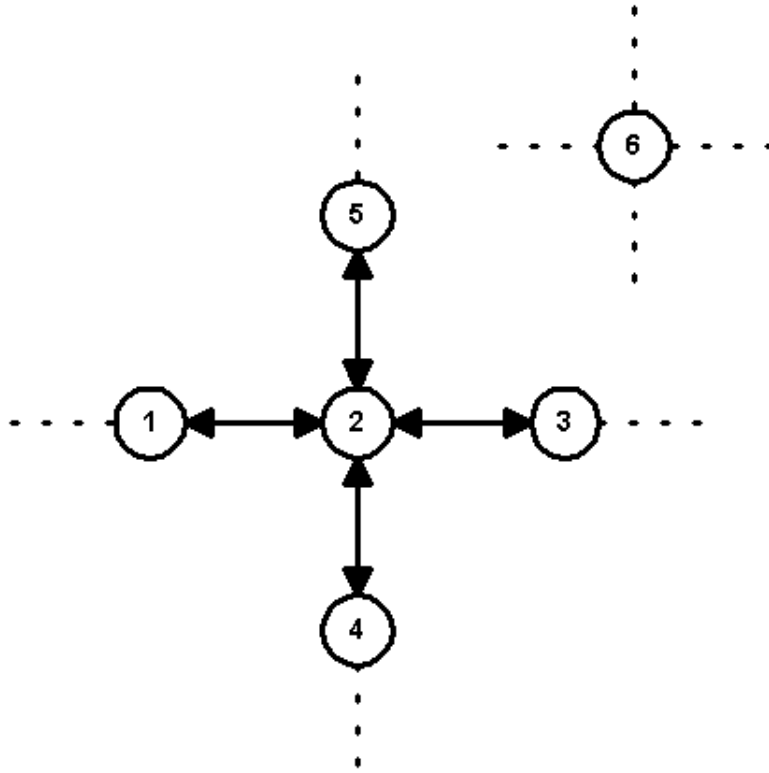
βανόμενους κόμβους. Σε περίπτωση που κάτι τέτοιο συμβεί ο κόμβος θα πρέπει να απορρίψει την ΑΙΔ, μην επιστρέφοντας μία ΑΠΔ και μην προωθώντας την ΑΙΔ.

Η χρήση των ΠΔ που βρίσκονται στην ΜΠΑΔ διαφόρων κόμβων μπορεί να προκαλέσει έναν μεγάλο αριθμό από ΑΠΔ, από πολλούς από αυτούς, σε μία μόνο ΑΙΔ. Το πλήθος αυτό των ΑΠΔ μπορεί να προκαλέσει συμφόρηση στο δίκτυο, συγκρούσεις πακέτων και άσκοπη κατανάλωση πολύτιμου εύρους ζώνης του ασύρματου μέσου. Στο σχήμα 3.3 όλοι οι γείτονες του κόμβου 2 υποθέτουμε ότι έχουν στην ΜΠΑΔ τους μια ΠΔ προς τον κόμβο 6. Έτσι η αποστολή από τον κόμβο 2 μίας ΑΙΔ για τον κόμβο 6 μπορεί να προκαλέσει ένα πλήθος από ΑΠΔ από τους γείτονες του κόμβου 2. Κάθε μία τέτοια ΑΠΔ περιέχει μία ΠΔ, διαφορετικού μήκους, προς τον κόμβο 6. Για την αντιμετώπιση αυτού του φαινομένου κάθε κόμβος πριν στείλει μία ΑΠΔ μπορεί να καθυστερήσει την αποστολή της, για ένα χρονικό διάστημα που εξαρτάται από το μήκος της ΠΔ που επιστρέφει στην ΑΠΔ του. Αυτό έχει σαν αποτέλεσμα κόμβοι με μικρότερο μήκος ΠΔ στην ΑΠΔ τους να απαντούν στον κόμβο - αφετηρία πιο γρήγορα. Κατά την διάρκεια αυτής της καθυστέρησης ο κόμβος ακούει για πακέτα δεδομένων από τον κόμβο - αφετηρία προς τον κόμβο - προορισμό. Αν ο κόμβος ακούσει ένα τέτοιο πακέτο δεδομένων που η ΠΔ που περιέχει είναι μικρότερου ή ίσου μήκους με αυτή που ο κόμβος θέλει να στείλει στην ΑΠΔ του τότε δεν την στέλνει, μιας και διαπιστώνει ότι κόμβος - αφετηρία ήδη έλαβε μία άλλη ΑΠΔ με ΠΔ που είναι τουλάχιστον το ίδιο καλή με την δική του. Σε διαφορετική περίπτωση ο κόμβος μετά το τέλος της καθυστέρησης στέλνει την ΑΠΔ του στον κόμβο - αφετηρία.

Κάθε ΑΙΔ μπορεί να περιέχει το πεδίο Μέγιστος Αριθμός Ενδιάμεσων Κόμβων. Το πεδίο αυτό μπορεί να χρησιμοποιηθεί για να περιοριστεί ο αριθμός των ενδιάμεσων κόμβων που επιτρέπεται να προωθήσουν μία συγκεκριμένη ΑΙΔ. Κάθε κόμβος που λαμβάνει ένα μήνυμα ΑΙΔ, με αυτό το πεδίο, μειώνει κατά ένα τον αριθμό στο πεδίο πριν προωθήσει το μήνυμα. Αν ο αριθμός στο πεδίο αυτό γίνει 0 πριν το μήνυμα φτάσει στον κόμβο - προορισμό ή σε κάποιον ενδιάμεσο κόμβο που ξέρει μια ΠΔ προς τον κόμβο - προορισμό τότε η αντίστοιχη ΑΙΔ απορρίπτεται.

### 3.2.6 Βελτιστοποιήσεις και ειδικές περιπτώσεις για την διαδικασία ΣΔ

Όταν ένας κόμβος στείλει ένα μήνυμα ΛΔ στον κόμβο αφετηρία μπορεί να διασώσει και να μην απορρίψει το πακέτο δεδομένων το οποίο και προκάλεσε το ΛΔ. Συγκεκριμένα ο κόμβος αφού στείλει το μήνυμα ΛΔ στη συνέχεια φάχνει στην ΜΠΑΔ του για μία ΠΔ προς τον προορισμό του πακέτου δεδομένων. Στην



Σχήμα 3.3: Μεγάλο πλήθος ΑΠΔ

περίπτωση που μία τέτοια διαδρομή βρεθεί τότε αντικαθιστά την ΠΔ του πακέτου δεδομένων με την ΠΔ της ΜΠΑΔ του. Στην συνέχεια το πακέτο στέλνεται στον Επόμενο Κόμβο με βάση την νέα ΠΔ. Στην περίπτωση που δεν βρεθεί στην ΜΠΑΔ του κόμβου μία τέτοια ΠΔ το πακέτο δεδομένων απορρίπτεται. Ακόμα στην περίπτωση που ένα πακέτο δεδομένων διασωθεί με τον τρόπο που αναφέραμε θα πρέπει αυτό να δηλωθεί σε κάποιο πεδίο της επικεφαλίδας του, διαφορετικά συνεχόμενες διασώσεις του πακέτου από διάφορους ενδιαμέσους κόμβους μπορεί να προκαλέσουν έναν κύκλο δρομολόγησης, στέλνοντας το πακέτο αυτό από τον έναν σωτήρα του στον άλλον. Ένας άλλος τρόπος για την διάσωση του πακέτου δεδομένων είναι η μερική αντικατάσταση της ΠΔ του πακέτου με την ΠΔ που βρέθηκε στην ΜΠΑΔ του κόμβου. Συγκεκριμένα μπορούμε να αντικαταστήσουμε με την ΠΔ που βρέθηκε στην ΜΠΑΔ μόνο το κομμάτι, της αρχικής ΠΔ του πακέτου, από τον κόμβο αυτό μέχρι τον κόμβο - προορισμό. Στην περίπτωση αυτή θα πρέπει βέβαια να ελέγξουμε μήπως η νέα ΠΔ έχει επαναλαμβανόμενους κόμβους, αποφεύγοντας έτσι τον κίνδυνο των κύκλων δρομολόγησης. Ένα σοβαρό μειονέκτημα του δεύτερου τρόπου διάσω-

σης είναι ότι δεν μπορεί το πακέτο δεδομένων με την νέα ΠΔ να χρησιμοποιήσει κάποιους από τους κόμβους που προηγουμένως χρησιμοποίησε για να φτάσει μέχρι τον κόμβο - σωτήρα του. Δηλαδή δεν μπορεί να επιστρέψει πίσω σε κάποιους από τους κόμβους από τους οποίους πέρασε και στην συνέχεια να ακολουθήσει μία διαφορετική διαδρομή μέχρι τον προορισμό ("backtracking").

Κατά την διάρκεια μετάδοσης ενός πακέτου δεδομένων είναι δυνατόν ένας κόμβος να ακούσει το πακέτο αυτό ακόμα και αν δεν απευθύνεται σε αυτόν, δηλαδή χωρίς να είναι ο Επόμενος Κόμβος. Αν διαβάζοντας το πακέτο αυτό ο κόμβος διαπιστώσει ότι είναι ένας από τους κόμβους που υπάρχουν μέσα στην ΠΔ του πακέτου από τους οποίους δεν έχει περάσει ακόμα αυτό, τότε αυτό σημαίνει ότι οι κόμβοι αυτής της ΠΔ ανάμεσα στον κόμβο που προώθησε το πακέτο και στον κόμβο που το άκουσε δεν χρειάζονται για την δρομολόγηση αντίστοιχων πακέτων. Στην περίπτωση αυτή ο κόμβος που άκουσε αυτό το πακέτο δεδομένων μπορεί να επιστρέψει στον κόμβο - αφετηρία μία ΑΠΔ η οποία θα περιέχει την νέα ΠΔ χωρίς τους κόμβους που δεν χρειάζονται.

Όταν ένας κόμβος - αφετηρία λάβει ένα μήνυμα ΛΔ για ένα πακέτο δεδομένων που έστειλε, τότε στην επόμενη διαδικασία ΑΔ μπορεί να προσθέσει στην ΑΙΔ αυτό το μήνυμα ΛΔ. Έτσι όλοι οι γειτονικοί αλλά και άλλοι κόμβοι θα αφαιρέσουν από την ΜΠΑΔ τους τις ΠΔ που περιέχουν την χαλασμένη ακμή. Με τον τρόπο αυτό αποφεύγεται ο κίνδυνος να έχουμε από κάποιο κόμβο ΑΠΔ που περιέχει την χαλασμένη ακμή.

Υπάρχει ακόμα η δυνατότητα στην ΜΠΑΔ πέρα από ΠΔ να αποθηκεύουμε και πληροφορίες για χαλασμένες ακμές, συνοδευόμενες βέβαια από ένα χρόνο λήξης (timeout). Με το τρόπο αυτό θα αποφεύγεται η χρησιμοποίησή τους από τον κόμβο για ένα καθορισμένο χρονικό διάστημα. Η δυνατότητα αυτή είναι ιδιαίτερα χρήσιμη για ακμές που η λειτουργία τους δεν είναι συνεχής, δηλαδή για ακμές που άλλες φορές δουλεύουν και άλλες όχι. Έτσι προσθέτοντας στην ΜΠΑΔ πληροφορία για αυτές τις ακμές ο κόμβος αποφεύγει να τις χρησιμοποιήσει ακόμα και αν κάποια στιγμή δουλεύουν. Η δυσκολία στη μέθοδο αυτή έγκειται στον προσδιορισμό του κατάλληλου χρόνου λήξης.

## 3.3 Zone Routing Protocol (ZRP)

### 3.3.1 Γενικά

Το ZRP είναι ένα υβριδικό πρωτόκολλο δρομολόγησης, το οποίο περιλαμβάνει χαρακτηριστικά τόσο των πρωτοκόλλων καθοδηγούμενων από πίνακα όσο και των πρωτοκόλλων καθοδηγούμενων από απαιτήσεις αποστολής. Στο πρωτόκολλο αυτό κάθε κόμβος διατηρεί γνώση της τοπολογίας του δικτύου σε μία περιοχή γύρω από αυτόν (ζώνη δρομολόγησης). Η γνώση αυτή βο-

ηθάει τόσο στην ανακάλυψη νέων διαδρομών, όταν αυτές χρειαστούν για την μεταφορά πακέτων δεδομένων, όσο και στη συντήρηση των χρησιμοποιούμενων διαδρομών ύστερα από αλλαγές στην τοπολογία του δικτύου. Το ZRP είναι ειδικά σχεδιασμένο για ad-hoc δίκτυα με μεγάλο πλήθος από κόμβους, οι οποίοι έχουν αυξημένη κινητικότητα και μεγάλη διασπορά. Ακόμα μπορεί να χρησιμοποιήσει τόσο source routing όσο και next hop routing. Το ZRP είναι ένα πρωτόκολλο που υλοποιείται πάνω από το επίπεδο δικτύου του μοντέλου αναφοράς δικτυακών πρωτοκόλλων ISO/OSI.

Όταν μιλάμε για το πρωτόκολλο δρομολόγησης ZRP ουσιαστικά αναφερόμαστε σε μία οικογένεια πρωτοκόλλων που όλα μαζί συντελούν στην αξιόπιστη και αποτελεσματική μεταφορά δεδομένων μέσα σε ένα ασύρματο ad-hoc δίκτυο. Τα πρωτόκολλα αυτά είναι το Intrazone Routing Protocol (IARP), το Bordercast Resolution Protocol (BRP) και το Interzone Routing Protocol (IERP). Το IARP είναι το πρωτόκολλο δρομολόγησης καθοδηγούμενο από πίνακα, το οποίο επιτρέπει σε κάθε κόμβο να έχει γνώση της τοπολογίας της ζώνης δρομολόγησης (ΖΔ) του. Η γνώση αυτή χρησιμοποιείται από τα άλλα δύο πρωτόκολλα. Το πρωτόκολλο BRP αναλαμβάνει την αποτελεσματική μετάδοση μηνυμάτων του πρωτοκόλλου IERP. Το IERP είναι το πρωτόκολλο δρομολόγησης καθοδηγούμενο από απαιτήσεις αποστολής, το οποίο ανακαλύπτει διαδρομές μεταξύ κόμβων που δεν βρίσκονται μέσα στην ίδια ΖΔ και συντηρεί χρησιμοποιούμενες διαδρομές ύστερα από τοπολογικές αλλαγές.

### 3.3.2 Το ασύρματο ad-hoc δίκτυο του ZRP

Στο δίκτυο στο οποίο λειτουργεί ο ZRP η επικοινωνία των κόμβων γίνεται μόνο ασύρματα, χωρίς δηλαδή της χρήση οποιασδήποτε σταθερής καλωδιακής υποδομής. Οι γείτονες ενός κόμβου, στο ασύρματο αυτό ad-hoc δίκτυο, είναι όλοι εκείνοι με τους οποίους ο κόμβος αυτός μπορεί να επικοινωνήσει ασύρματα απευθείας, χωρίς την παρεμβολή οποιουδήποτε τρίτου σταθμού. Προφανώς το πλήθος των γειτόνων ενός κόμβου εξαρτάται από το εύρος μετάδοσης του κόμβου αυτού. Επομένως στο γράφημα συνδεσιμότητας που αναπαριστά το δίκτυο μία ακμή μεταξύ δύο κόμβων αναπαριστά ακριβώς την δυνατότητα τους για άμεση ασύρματη επικοινωνία.

Κύριο χαρακτηριστικό του ασύρματου ad-hoc δίκτυο του ZRP είναι οι ζώνες δρομολόγησης (ΖΔ) των κόμβων του. Συγκεκριμένα η ΖΔ ενός κόμβου ορίζεται σαν το σύνολο των κόμβων του δικτύου των οποίων η απόσταση (σε ακμές) από τον αρχικό κόμβο είναι μικρότερη (εσωτερικοί κόμβοι) ή ίση (περιφερειακοί κόμβοι) από μία τιμή που ονομάζεται ακτίνα ζώνης (AZ). Επομένως όσο μεγαλύτερο είναι το εύρος της μετάδοσης ενός κόμβου τόσο μεγαλύτερο είναι το πλήθος των κόμβων που ανήκουν στην ΖΔ του. Κάθε κόμβος του δικτύου έχει την δική του ΖΔ με αποτέλεσμα οι ΖΔ να επικαλύπτονται. Ακόμα κάθε

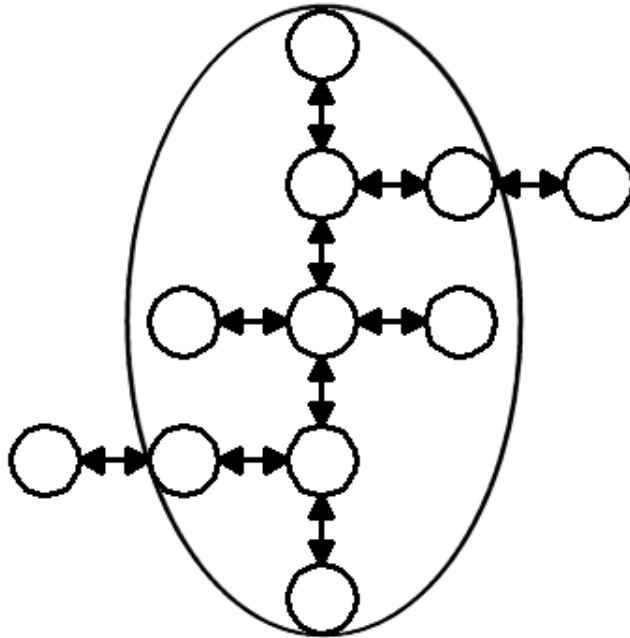
κόμβος του δικτύου διατηρεί (proactively) γνώση της τοπολογίας της ΖΔ του.

Η επιλογή της σωστής τιμής για την ΑΖ είναι πολύ σημαντική για την απόδοση του ΖΡΡ. Η επιλογή αυτή εξαρτάται από την κινητικότητα των κόμβων του δικτύου, από το πλήθος των αιτήσεων για ανακάλυψη νέων διαδρομών αλλά και από το κατά πόσο το δίκτυο είναι αραιό ή πυκνό. Συγκεκριμένα μεγαλύτερη ΑΖ επιλέγεται όταν σε ένα αραιό δίκτυο υπάρχουν πολλές αιτήσεις για νέες διαδρομές και/ή όταν οι περισσότεροι κόμβοι κινούνται πολύ αργά. Έτσι στην ακραία περίπτωση που όλοι οι κόμβοι είναι ακίνητοι τότε η ΑΖ πρέπει να είναι άπειρη, δηλαδή κάθε ΖΔ να περιλαμβάνει όλους τους κόμβους του δικτύου, έτσι ο ΖΡΡ μετατρέπεται σε ένα ουσιαστικά πρωτόκολλο καθοδηγούμενο από πίνακα. Από την άλλη μεριά μικρότερη ΑΖ επιλέγεται όταν σε ένα πυκνό δίκτυο υπάρχουν λίγες αιτήσεις για ανακάλυψη νέων διαδρομών και/ή όταν το δίκτυο αποτελείται από λίγους κόμβους οι οποίοι κινούνται γρήγορα μεταξύ τους. Στην ακραία περίπτωση επιλέγεται ΑΖ ίση με ένα και ο ΖΡΡ μετατρέπεται σε ένα ουσιαστικά πρωτόκολλο καθοδηγούμενο από απαιτήσεις αποστολής. Γενικά η ΑΖ που αντιστοιχεί σε κάθε κόμβο ενός δικτύου μπορεί να διαφέρει από περιοχή σε περιοχή του δικτύου ή ακόμα και από κόμβο σε κόμβο. Μπορεί ακόμα η τιμή αυτή να αλλάζει δυναμικά κατά την διάρκεια εκτέλεσης του πρωτοκόλλου ανάλογα με τις συνθήκες που επικρατούν στο δίκτυο.

Γενικά η ασύρματη επικοινωνία μεταξύ δύο κόμβων δεν είναι πάντα δυνατή και προς τις δύο κατευθύνσεις. Δηλαδή το ότι ένας κόμβος μπορεί να επικοινωνήσει ασύρματα απευθείας με έναν άλλον κόμβο δεν σημαίνει απαραίτητα ότι ισχύει και το αντίστροφο. Σε αντίθεση με πολλά άλλα πρωτόκολλα δρομολόγησης ο ΖΡΡ υποστηρίζει, τοπικά, τις ακμές μονής κατευθύνσεως. Συγκεκριμένα μία ακμή μονής κατευθύνσεως μπορεί να χρησιμοποιηθεί από τον ΖΡΡ εφόσον η ΖΔ της κορυφής της ακμής περιέχει την ουρά της ακμής και το αντίστροφο.

Κάθε κόμβος στο δίκτυο, στο οποίο λειτουργεί ο ΖΡΡ, έχει ένα μοναδικό προσδιοριστικό (π.χ. IP διεύθυνση). Το πρωτόκολλο ΖΡΡ αναγνωρίζει τους διάφορους κόμβους του δικτύου χρησιμοποιώντας αυτά τα προσδιοριστικά. Παράλληλα κάθε κόμβος του δικτύου μετέχει στο πρωτόκολλο ΖΡΡ κάθε στιγμή, με την έννοια ότι θα πρέπει να προωθεί πακέτα για άλλους κόμβους όταν κάτι τέτοιο του ζητηθεί. Ακόμα κάθε κόμβος μπορεί να κινείται οποιαδήποτε στιγμή χωρίς καμία ειδοποίηση. Η κίνηση αυτή μπορεί να είναι τυχαία και δεν εξαρτάται από την κίνηση ή την μη κίνηση άλλων κόμβων. Η κίνηση ακόμα αυτή των κόμβων μπορεί να έχει ως αποτέλεσμα την είσοδο ή την έξοδο ενός κόμβου από μία ΖΔ ή ακόμα και την διάσπαση του δικτύου.

Στο παρακάτω σχήμα φαίνεται ένα παράδειγμα ενός γράφου συνδεσιμότητας που αναπαριστά ένα δίκτυο στο οποίο εκτελείται το πρωτόκολλο ΖΡΡ. Σημειώνεται επίσης και μία ΖΔ γύρω από τον αντίστοιχο κόμβο, μια ΖΔ δεν έχει απαραίτητα κυκλικό σχήμα.



Σχήμα 3.4: Γράφημα συνδεσιμότητας δικτύου για το ZRP

### 3.3.3 Intrazone Routing Protocol (IARP)

Το IARP είναι το πρωτόκολλο δρομολόγησης καθοδηγούμενο από πίνακα που επιτρέπει σε κάθε κόμβο του δικτύου να έχει γνώση της τοπολογίας της ΖΔ του. Κάθε κόμβος δηλαδή, με την βοήθεια του IARP, βρίσκει διαδρομές προς όλους τους κόμβους που υπάρχουν μέσα στην ΖΔ του και μόνο για αυτούς. Έτσι αποφεύγεται, όταν χρειαστεί να σταλλεί ένα νέο πακέτο δεδομένων, η αναζήτηση μέσα στο δίκτυο για μια διαδρομή προς έναν κόμβο που βρίσκεται μέσα στην ΖΔ του αρχικού κόμβου. Ακόμα η γνώση αυτή βοηθάει στην αποδοτική μετάδοση μηνυμάτων μέσα στο δίκτυο για την ανακάλυψη νέων διαδρομών αλλά και στην συντήρηση ήδη γνωστών διαδρομών ύστερα από τοπολογικές αλλαγές.

Σαν IARP μπορεί να χρησιμοποιηθεί κάποιο από τα ήδη γνωστά πρωτόκολλα δρομολόγησης καθοδηγούμενα από πίνακα, περιορίζοντας όμως την ενημέρωση για τις όποιες τοπολογικές αλλαγές αφορούν έναν κόμβο στα μέλη της ΖΔ αυτού. Αυτό σημαίνει ότι οι οποιοσδήποτε τοπολογικές αλλαγές συμβαίνουν στο δίκτυο έχουν μόνο τοπική επίδραση και δεν επηρεάζουν κόμβους που βρίσκονται στην άλλη μεριά του δικτύου.

Για την λειτουργία του IARP κάθε κόμβος πρέπει να γνωρίζει ποιοί είναι οι γειτονικοί του κόμβοι και να ενημερώνεται για οποιοσδήποτε αλλαγές σε αυτούς. Η γνώση αυτή παρέχεται από ένα άλλο πρωτόκολλο, του MAC επιπέ-

δου, το Neighbor Discovery Protocol (NDP) το οποίο δεν ανήκει στο ZRP. Με βάση το NDP κάθε κόμβος περιοδικά εκπέμπει σε όλους τους γείτονες του ένα πακέτο το οποίο περιέχει το μοναδικό προσδιοριστικό του (π.χ. την IP διεύθυνση του), δηλώνοντας με τον τρόπο αυτό την παρουσία του. Κάθε κόμβος που λαμβάνει ένα τέτοιο πακέτο καταχωρεί την πληροφορία αυτή σε έναν εσωτερικό του πίνακα, τον οποίον και ελέγχει περιοδικά. Ο πίνακας αυτός ονομάζεται Πίνακας Γειτόνων (ΠΓ). Αν ένας γείτονας δεν έχει στείλει το πακέτο αυτό μετά από έναν συγκεκριμένο αριθμό ελέγχων του ΠΓ τότε θεωρείται ότι ο κόμβος αυτός δεν είναι πλέον γείτονας και αφαιρείται από τον ΠΓ. Ακόμα αν στον ΠΓ καταχωρηθεί ένας κόμβος ο οποίος δεν υπήρχε στον προηγούμενο έλεγχο του πίνακα τότε αυτός θεωρείται σαν ένας καινούριος γείτονας. Σε κάθε περίπτωση που διαπιστώνεται μία οποιαδήποτε αλλαγή στους γείτονες ενός κόμβου το NDP, που τρέχει στον κόμβο, ενημερώνει το IARP.

Τα μηνύματα που ανταλλάσσουν οι κόμβοι μεταξύ τους, για την λειτουργία του NDP πρωτοκόλλου, έχουν την παρακάτω δομή:

Προσδιοριστικό κόμβου
-----------------------

Πίνακας 3.6: Δομή μηνύματος που οι κόμβοι ανταλλάσσουν μεταξύ τους για το NDP

Ο Πίνακας Γειτόνων (ΠΓ) έχει τα παρακάτω πεδία:

ΠΚ	Καινούριος	Μετρητής Απουσίας
----	------------	-------------------

Πίνακας 3.7: Πεδία του Πίνακα Γειτόνων

Το πεδίο ΠΚ είναι το προσδιοριστικό ενός κόμβου γείτονα. Το πεδίο Καινούριος δηλώνει το αν ο αντίστοιχος κόμβος είναι ένας νέος γείτονας. Το πεδίο Μετρητής Απουσίας διατηρεί τον αριθμό των ελέγχων του ΠΓ που έχουν γίνει χωρίς εν τω μεταξύ ο αντίστοιχος κόμβος να στείλει ένα μήνυμα με το οποίο να δηλώνει την παρουσία του.

Σύμφωνα με το IARP κάθε φορά που υπάρχει μία αλλαγή στους γείτονες ενός κόμβου, αυτός αναλαμβάνει να ενημερώσει όλους τους κόμβους της ΖΔ του για αυτή την αλλαγή. Η ενημέρωση αυτή γίνεται με ένα Πακέτο Κατάστασης Ακμών (ΠΑΚΑ), το οποίο και προωθείται σε όλα τα μέλη της ΖΔ του. Με τον τρόπο αυτό κάθε κόμβος λαμβάνει από κάθε άλλο κόμβο που ανήκει στην ΖΔ του ένα τουλάχιστον ΠΑΚΑ. Από αυτά ο κόμβος υπολογίζει διαδρομές προς όλους τους κόμβους της ΖΔ του. Ένα ΠΑΚΑ έχει την παρακάτω δομή:

Το πεδίο ΠΚ πηγής είναι το προσδιοριστικό του κόμβου που έστειλε το πακέτο, ενώ τα πεδία ΠΚ γείτονα  $i$  είναι τα προσδιοριστικά των γειτόνων του

ΠΚ πηγής	Αριθμός Σειράς	Ακτίνα Ζώνης	TTL
ΠΚ γείτονα 1	...	...	ΠΚ γείτονα n

Πίνακας 3.8: Δομή του Πακέτου Κατάστασης Ακμών

κόμβου αυτού. Το πεδίο Αριθμός Σειράς σε συνδυασμό με το ΠΚ πηγής προσδιορίζουν μοναδικά αυτό το Πακέτο Κατάστασης Ακμών (ΠΑΚΑ). Το πεδίο Ακτίνα Ζώνης είναι η ΑΖ του κόμβου που έστειλε το πακέτο. Το πεδίο TTL (time-to-live) περιορίζει την διάδοση ενός ΠΑΚΑ μέσα στην ΖΔ του κόμβου που το έστειλε, προσδιορίζοντας τον αριθμό των ενδιάμεσων κόμβων από όπου αυτό το πακέτο μπορεί να προωθηθεί μέχρι να απορριφθεί. Κάθε κόμβος πριν προωθήσει ένα ΠΑΚΑ πρώτα μειώνει κατά ένα την τιμή του TTL πεδίου του, αν το TTL γίνει ίσο με το 0 τότε το πακέτο απορρίπτεται. Με τον τρόπο αυτό οποιεσδήποτε τοπολογικές αλλαγές συμβαίνουν στο δίκτυο έχουν μόνο τοπική επίδραση και δεν επηρεάζουν κόμβους που βρίσκονται στην άλλη μεριά του δικτύου.

Όταν το IARP που τρέχει σε έναν κόμβο ενημερωθεί από το NDP για μία αλλαγή στους γείτονες του κόμβου ή όταν ο κόμβος λάβει ένα νέο ΠΑΚΑ τότε ενημερώνονται ο Πίνακας Κατάστασης Ακμών (ΠΙΚΑ) και ο πίνακας δρομολόγησης του κόμβου ανάλογα. Στην συνέχεια το IARP ειδοποιεί το IERP για αυτές τις αλλαγές και προωθεί ένα ΠΑΚΑ σε όλους τους γειτονικούς του κόμβους. Το πακέτο αυτό είναι είτε το αντίστοιχο πακέτο που έλαβε είτε το πακέτο που δημιούργησε ο κόμβος μετά την ενημέρωσή του από το NDP.

Ο ΠΙΚΑ ενός κόμβου διατηρεί πληροφορίες για τους γείτονες όλων των κόμβων της ΖΔ του αλλά και για τους γείτονες του ιδίου. Ο ΠΙΚΑ έχει τα παρακάτω πεδία:

ΠΚ πηγής	Ακτίνα Ζώνης	Ώρα Εγγραφής
ΠΚ γείτονα 1	...	ΠΚ γείτονα n

Πίνακας 3.9: Πεδία του Πίνακα Κατάστασης Ακμών

Τα πεδία ΠΚ γείτονα i είναι τα προσδιοριστικά των γειτόνων του κόμβου με προσδιοριστικό ΠΚ πηγής. Το πεδίο Ακτίνα Ζώνης είναι η ΑΖ του κόμβου αυτού. Το πεδίο Ώρα Εγγραφής περιέχει την ώρα που έγινε η αντίστοιχη εγγραφή στον ΠΙΚΑ.

Ο πίνακας δρομολόγησης του IARP περιέχει συντομότερες διαδρομές προς όλους τους κόμβους της ΖΔ του αντίστοιχου κόμβου. Ο πίνακας αυτός έχει τα παρακάτω πεδία:

Το πεδίο ΠΚ προορισμού είναι το προσδιοριστικό του κόμβου προορισμού και τα πεδία ΠΚ i είναι τα προσδιοριστικά των ενδιάμεσων κόμβων της διαδρομής



ΠΚ προορισμού	ΠΚ 1	...	ΠΚ n
---------------	------	-----	------

Πίνακας 3.10: Πεδία του πίνακα δρομολόγησης του IARP

από τον αρχικό κόμβο μέχρι τον προορισμό.

Μία ακόμα λειτουργία του IARP είναι ο περιοδικός έλεγχος του ΠΚΑ για εγγραφές που ο χρόνος από την Ώρα Εγγραφής τους έχει ξεπεράσει κάποιο συγκεκριμένο όριο. Στην περίπτωση αυτή οι αντίστοιχες εγγραφές αφαιρούνται από τον ΠΚΑ, ο πίνακας δρομολόγησης του IARP ξαναυπολογίζεται και το IERP ενημερώνεται για τις αλλαγές στον πίνακα δρομολόγησης.

### 3.3.4 Bordercast Resolution Protocol (BRP)

Το BRP είναι μία υπηρεσία μετάδοσης μηνυμάτων, η οποία χρησιμοποιείται από το IERP για την αποστολή αιτήσεων ανακάλυψης διαδρομής. Η μετάδοση των μηνυμάτων αυτών γίνεται πιο αποδοτικά από τον κλασικό τρόπο μετάδοσης που βασίζεται στην εκπομπή του ίδιου μηνύματος σε όλους του γείτονες αδιακρίτως και στην επανάληψη του βήματος αυτού από όλους τους κόμβους που λαμβάνουν το μήνυμα (broadcast). Με βάση το BRP κάθε κόμβος προωθεί την αίτηση ανακάλυψης διαδρομής, που δημιουργεί ή που λαμβάνει, σε ορισμένους μόνο από τους γείτονες του. Συγκεκριμένα το BRP κατευθύνει την αίτηση μακριά από τον κόμβο πηγή, προς περιοχές του δικτύου που ακόμα δεν έχουν καλυφθεί. Ένας κόμβος θεωρείται καλυμμένος αν ανήκει στην ΖΔ ενός άλλου κόμβου ο οποίος έχει ήδη λάβει την αίτηση.

Κάθε κόμβος που θέλει να στείλει μια αίτηση ανακάλυψης διαδρομής, χρησιμοποιώντας το BRP, βασίζεται στην γνώση της τοπολογίας της ΖΔ του, η οποία παρέχεται από το IARP. Με την γνώση αυτή ο κόμβος κατασκευάζει ένα multicast - bordercast δέντρο με κορυφή τον ίδιο και φύλλα τους κόμβους που βρίσκονται στην περιφέρεια της ΖΔ του (peripheral nodes) και οι οποίοι δεν είναι καλυμμένοι. Στην συνέχεια ο κόμβος προωθεί την αίτηση ανακάλυψης διαδρομής στους γείτονες του στο δέντρο αυτό. Το BRP χρησιμοποιεί ειδικούς μηχανισμούς ελέγχου αιτήσεων και δομές που συντελούν ώστε οι αιτήσεις ανακάλυψης διαδρομής να μην πηγαίνουν σε ήδη καλυμμένες περιοχές του δικτύου. Έτσι το BRP πετυχαίνει την μετάδοση των αιτήσεων ανακάλυψης διαδρομής με λιγότερα μηνύματα από ότι οι κλασικοί broadcast μηχανισμοί.

Το πρωτόκολλο BRP επικοινωνεί άμεσα ή έμμεσα με τρία άλλα πρωτόκολλα. Συγκεκριμένα έχει πρόσβαση στις δομές του πρωτοκόλλου IARP, δηλαδή στον ΠΚΑ και στον πίνακα δρομολόγησης, από όπου αποκτά γνώση της τοπολογίας της ΖΔ του κόμβου στον οποίον τρέχει. Ακόμα επικοινωνεί με το επίπεδο του δικτύου (π.χ. IP) για την λήψη ή αποστολή Πακέτων Αιτήσεων Ανακάλυψης Διαδρομής (ΠΑΑΔ). Το πρωτόκολλο BRP επικοινωνεί και με το πρωτόκολλο

λο IERP με το οποίο ανταλλάσσει την πληροφορία της αίτησης ανακάλυψης διαδρομής που αφορά το IERP και η οποία υπάρχει μέσα σε κάθε ΠΑΑΔ.

Ένα ΠΑΑΔ έχει την παρακάτω δομή:

ΠΚ πηγής	ΠΚ προορισμού	Αριθμός Αίτησης
ΠΚ προηγούμενου	Ενσωματωμένο Πακέτο	-

Πίνακας 3.11: Δομή του Πακέτου Αίτησης Ανακάλυψης Διαδρομής

Τα πεδία ΠΚ πηγής και ΠΚ προορισμού προσδιορίζουν την πηγή και τον προορισμό αντίστοιχα της αίτησης. Το πεδίο Αριθμός Αίτησης είναι ένας διαδοχικός αριθμός ο οποίος μαζί με το ΠΚ πηγής προσδιορίζουν μοναδικά στο δίκτυο την συγκεκριμένη αίτηση. Το πεδίο ΠΚ προηγούμενου προσδιορίζει τον τελευταίο κόμβο που προώθησε αυτό το ΠΑΑΔ. Το Ενσωματωμένο Πακέτο (ΕΠ) περιέχει την πληροφορία της αίτησης ανακάλυψης διαδρομής που αφορά το IERP.

Όταν ένας κόμβος λάβει ένα ΠΑΑΔ τότε αρχικά καταγράφει την συγκεκριμένη αίτηση σε έναν πίνακα ο οποίος ονομάζεται Πίνακας Κάλυψης Αιτήσεων (ΠΚΑ). Παράλληλα στον ίδιο πίνακα σημειώνει όλους τους κόμβους της ΖΔ του, που ανήκουν και στην ΖΔ του ΠΚ προηγούμενου σαν καλυμμένους για την συγκεκριμένη βέβαια αίτηση. Στην συνέχεια ο κόμβος εξετάζει το κατά πόσο ο ίδιος είναι γείτονας με τον ΠΚ προηγούμενο στο multicast-bordercast δέντρο του. Αν είναι πράγματι γείτονας τότε παραδίδει το Ενσωματωμένο Πακέτο (ΕΠ) του ΠΑΑΔ στο IERP. Σε διαφορετική περίπτωση, αν δεν είναι γείτονας, ο κόμβος αυτός σημειώνει όλους τους κόμβους της ΖΔ του σαν καλυμμένους και απορρίπτει το πακέτο. Το γεγονός ότι ο κόμβος έλαβε ένα ΠΑΑΔ παρόλο που ο ίδιος δεν είναι γείτονας με τον ΠΚ προηγούμενο στο multicast-bordercast δέντρο του σημαίνει ότι η ΖΔ του κόμβου καλύπτεται από τις ΖΔ άλλων κόμβων που μεταδίδουν αυτό το ΠΑΑΔ.

Για την καταγραφή των καλυμμένων κόμβων αλλά και για την καταγραφή των αντίστοιχων αιτήσεων που έχουν ληφθεί κάθε κόμβος διατηρεί έναν Πίνακα Κάλυψης Αιτήσεων (ΠΚΑ). Αυτός ο πίνακας έχει τα παρακάτω πεδία:

ΠΚ πηγής	Αριθμός Αίτησης	Γράφημα Κάλυψης
----------	-----------------	-----------------

Πίνακας 3.12: Πεδία του Πίνακα Κάλυψης Αιτήσεων

Το πεδίο ΠΚ πηγής προσδιορίζει τον κόμβο που δημιούργησε την αντίστοιχη αίτηση. Το πεδίο Αριθμός Αίτησης είναι ένας διαδοχικός αριθμός ο οποίος μαζί με το ΠΚ πηγής προσδιορίζουν μοναδικά στο δίκτυο την συγκεκριμένη αίτηση. Το πεδίο Γράφημα Κάλυψης περιγράφει την τοπολογία της ΖΔ του κόμβου και

επιπλέον διατηρεί πληροφορία για το κατά πόσο κάθε μέλος - κόμβος της ΖΔ έχει καλυφθεί από την συγκεκριμένη αίτηση.

Όταν το πρωτόκολλο BRP λάβει από το IERP ένα ΕΠ τότε αρχικά ο κόμβος κατασκευάζει το multicast-bordercast δέντρο του. Για την κατασκευή αυτού του δέντρου χρησιμοποιούνται οι πληροφορίες που υπάρχουν στον ΠΚΑ και αφορούν την αντίστοιχη αίτηση. Στην συνέχεια ο κόμβος μεταδίδει το αντίστοιχο ΠΑΑΔ στους γείτονες του στο δέντρο που κατασκεύασε. Τέλος σημειώνει στον ΠΚΑ όλους τους κόμβους της ΖΔ του σαν καλυμμένους για την συγκεκριμένη αίτηση.

### 3.3.5 Interzone Routing Protocol (IERP)

Το IERP είναι το πρωτόκολλο δρομολόγησης καθοδηγούμενο από απαιτήσεις αποστολής της οικογένειας πρωτοκόλλων ZRP, το οποίο χρησιμοποιείται για την ανακάλυψη διαδρομών μεταξύ κόμβων που δεν βρίσκονται στην ίδια ΖΔ αλλά και για την συντήρηση ήδη γνωστών διαδρομών ύστερα από τοπολογικές αλλαγές. Το IERP χρησιμοποιεί τα δεδομένα και τις υπηρεσίες των άλλων δύο πρωτοκόλλων της οικογένειας (IARP, BRP).

Σαν IERP μπορεί να χρησιμοποιηθεί κάποιο από τα ήδη γνωστά πρωτόκολλα δρομολόγησης καθοδηγούμενα από απαιτήσεις αποστολής, το οποίο όμως θα πρέπει να τροποποιηθεί έτσι ώστε να εκμεταλλευτεί τις υπηρεσίες του BRP αλλά και την γνώση της τοπολογίας της ΖΔ που παρέχεται από το πρωτόκολλο IARP.

Το επίπεδο δικτύου είναι εκείνο που ειδοποιεί το IERP για την ανάγκη εκκίνησης μίας νέας διαδικασίας ανακάλυψης διαδρομής για κάποιον προορισμό όταν δεν βρει κάποια κατάλληλη διαδρομή στο πίνακα δρομολόγησης του IERP. Τότε το IERP ετοιμάζει την αντίστοιχη αίτηση ανακάλυψης διαδρομής και την στέλνει στο BRP για να την μεταδώσει. Αυτή η αίτηση ανακάλυψης διαδρομής δεν είναι άλλη από το Ενσωματωμένο Πακέτο ΕΠ του ΠΑΑΔ.

Ο πίνακας δρομολόγησης του IERP έχει τα παρακάτω πεδία:

ΠΚ προορισμού	ΠΚ 1	...	ΠΚ n
---------------	------	-----	------

Πίνακας 3.13: Πεδία του πίνακα δρομολόγησης του IERP

Το πεδίο ΠΚ προορισμού είναι το προσδιοριστικό του κόμβου προορισμού και τα πεδία ΠΚ i είναι τα προσδιοριστικά των ενδιαμέσων κόμβων της διαδρομής από τον αρχικό κόμβο μέχρι τον προορισμό.

Ενώ το ΕΠ έχει την παρακάτω δομή:

Τα πεδία ΠΚ πηγής και ΠΚ προορισμού προσδιορίζουν την πηγή και τον προορισμό αντίστοιχα της αίτησης. Το πεδίο Αριθμός Αίτησης είναι ένας δι-αδοχικός αριθμός ο οποίος μαζί με το ΠΚ πηγής προσδιορίζουν μοναδικά στο

ΠΚ πηγής	ΠΚ προορισμού	Αριθμός Αίτησης
ΠΚ 1	...	ΠΚ n

Πίνακας 3.14: Δομή του Ενσωματωμένου Πακέτου

δίκτυο την συγκεκριμένη αίτηση. Τέλος τα πεδία ΠΚ  $i$  είναι τα προσδιοριστικά των ενδιάμεσων κόμβων της διαδρομής που μέχρι στιγμής έχει ανακαλυφθεί καθώς η συγκεκριμένη αίτηση ανακάλυψης διαδρομής κινείται στο δίκτυο.

Το IERP χρησιμοποιεί το πρωτόκολλο BRP για την αποδοτική μετάδοση των αιτήσεων ανακάλυψης διαδρομής μέσα στο δίκτυο, ανταλλάσσοντας με αυτό το ΕΠ. Όταν το IERP λάβει από το BRP το ΕΠ καταρχήν ελέγχει αν υπάρχει στον πίνακα δρομολόγησης του αντίστοιχη διαδρομή προς τον προορισμό. Στην περίπτωση που δεν βρεθεί μία διαδρομή προς τον προορισμό τότε η διαδρομή του ΕΠ συμπληρώνεται με το προσδιοριστικό του κόμβου και στην συνέχεια το ΕΠ επιστρέφεται στο BRP για να το μεταδώσει. Στην περίπτωση που βρεθεί μία διαδρομή προς τον τελικό προορισμό, στον πίνακα δρομολόγησης του IERP, τότε δημιουργείται η αντίστοιχη απάντηση αίτησης διαδρομής. Σε αυτήν τοποθετείται η συνολική διαδρομή από τον κόμβο πηγή στον κόμβο προορισμό η οποία προήλθε από την συνένωση της διαδρομής που υπήρχε στο ΕΠ με την διαδρομή που βρέθηκε στον πίνακα δρομολόγησης του IERP. Αυτή η απάντηση αίτησης διαδρομής στέλνεται μέσω του επιπέδου δικτύου στον κόμβο πηγή ακολουθώντας την αντίστροφη διαδρομή από αυτήν που ακολούθησε η αντίστοιχη αίτηση ανακάλυψης διαδρομής. Η δομή μίας απάντησης αίτησης διαδρομής είναι πανομοιότυπη με αυτή του ΕΠ.

Όταν το πρωτόκολλο IERP λάβει μία απάντηση αίτησης διαδρομής από το επίπεδο δικτύου αρχικά αποθηκεύει στον πίνακα δρομολόγησης του τη διαδρομή προς τον κόμβο προορισμό που υπάρχει μέσα στην απάντηση. Στην συνέχεια αν ο κόμβος που έλαβε την απάντηση δεν είναι ο κόμβος πηγή τότε αυτή προωθείται προς τον κόμβο πηγή μέσω πάλι του επιπέδου δικτύου.

Όταν το πρωτόκολλο IERP ειδοποιηθεί από το πρωτόκολλο IARP για αλλαγή στην ΖΔ του κόμβου τότε χρησιμοποιεί την πληροφορία που παρέχεται από τον πίνακα δρομολόγησης του IARP με σκοπό την συντήρηση των ήδη γνωστών διαδρομών του πίνακα δρομολόγησης του IERP. Αυτό γίνεται επειδή η κίνηση των κόμβων μπορεί να έχει ως αποτέλεσμα κάποιες διαδρομές να είναι πλέον άκυρες. Η συντήρηση αυτή μπορεί να έχει σαν αποτέλεσμα την μείωση ή την αύξηση του μήκους των αντίστοιχων διαδρομών. Με την συντήρηση τέλος αυτή είναι δυνατόν να βρεθούν συντομότερες διαδρομές προς τους διάφορους προορισμούς αλλά και να αποκατασταθεί η εγκυρότητα ορισμένων διαδρομών.

Παρόμοια λειτουργία με το IERP πραγματοποιεί και το επίπεδο δικτύου, το οποίο μεταξύ των άλλων είναι υπεύθυνο για την μετάδοση των πακέτων

δεδομένων στον προορισμό τους. Συγκεκριμένα όταν το επίπεδο δικτύου ενός κόμβου λάβει ένα πακέτο δεδομένων ελέγχει την διαδρομή που υπάρχει μέσα σε αυτό και το ανακατευθύνει, αν χρειάζεται, σε μία συντομότερη ή απλώς έγκυρη διαδρομή προς τον τελικό προορισμό. Για την λειτουργία του αυτή το επίπεδο δικτύου χρησιμοποιεί την γνώση της τοπολογίας της ΖΔ του κόμβου.

## 3.4 Άλλα πρωτόκολλα δρομολόγησης

### 3.4.1 Destination-Sequenced Distance-Vector (DSDV)

Το DSDV είναι ένα πρωτόκολλο δρομολόγησης καθοδηγούμενο από πίνακα που βασίζεται στον αλγόριθμο δρομολόγησης Bellman-Ford, αποφεύγοντας την δημιουργία βρόχων δρομολόγησης. Κάθε κόμβος διατηρεί έναν πίνακα δρομολόγησης, όπου κάθε εγγραφή περιέχει τον επόμενο γειτονικό κόμβο προς έναν κόμβο προορισμού, τον αντίστοιχο αριθμό ενδιάμεσων κόμβων καθώς και έναν αριθμό ακολουθίας, ο οποίος ανατίθεται από τον κόμβο προορισμού. Κάθε κόμβος μεταδίδει περιοδικά στους γείτονες του τις αναπροσαρμογές του πίνακα δρομολόγησης του, οι οποίες προκλήθηκαν από τοπολογικές αλλαγές. Συγκεκριμένα κάθε εγγραφή σε ένα πακέτο ανανέωσης περιέχει έναν κόμβο προορισμού, τον αριθμό των ενδιάμεσων κόμβων και τον αριθμό ακολουθίας. Επειδή οι τοπολογικές αλλαγές μπορεί να προκαλέσουν μεγάλη κυκλοφορία πακέτων ανανέωσης στο δίκτυο, οι αναπροσαρμογές που μεταδίδονται από κάθε κόμβο μπορεί να έχουν δύο μορφές. Η πρώτη μορφή είναι γνωστή σαν πλήρη ενημέρωση, όπου ο κόμβος μεταδίδει ολόκληρο τον πίνακα δρομολόγησης του, πιθανός σε πολλά πακέτα. Η δεύτερη μορφή περιέχει μόνο τις αλλαγές μετά την τελευταία πλήρη ενημέρωση και μεταδίδεται σε ένα πακέτο, το επαυξητικό πακέτο. Όταν η τοπολογία του δικτύου αλλάζει γρήγορα οι πλήρεις ενημερώσεις είναι πιο συχνές. Κάθε κόμβος που λαμβάνει αυτά τα πακέτα ανανεώνει τον πίνακα δρομολόγησης του χρησιμοποιώντας τους αριθμούς ακολουθίας, έτσι ώστε να ξεχωρίζει τις παλιές πληροφορίες από τις νέες και να αποφεύγει την δημιουργία βρόχων δρομολόγησης. Συγκεκριμένα κάθε φορά για έναν κόμβο προορισμού επιλέγεται η πληροφορία με τον πιο πρόσφατο αριθμό ακολουθίας, ενώ σε περίπτωση ισότητας επιλέγεται εκείνη με την μικρότερη απόσταση από τον προορισμό. Σε περίπτωση που ένας κόμβος διαπιστώσει πως ένας κόμβος προορισμού δεν είναι πια προσιτός αναθέτει στην αντίστοιχη εγγραφή τον αμέσως επόμενο αριθμό ακολουθίας και θέτει τον αριθμό των ενδιάμεσων κόμβων στο άπειρο. Τέλος κάθε κόμβος διατηρεί έναν ξεχωριστό πίνακα με τον οποίον υπολογίζει τον μέσο χρόνο που χρειάζεται για να γίνει βέλτιστη η πληροφορία για κάποιον κόμβο προορισμού. Με βάση αυτούς τους χρόνους οι κόμβοι επιβραδύνουν την μετάδοση πακέτων ανανέωσης, μειώνοντας με τον

τρόπο αυτό τον αριθμό των πακέτων αναπροσαρμογής που μεταδίδονται στο δίκτυο.

### 3.4.2 Clusterhead Gateway Switch Routing (CGSR)

Το CGSR είναι ένα πρωτόκολλο δρομολόγησης που βασίζεται στην ομαδοποίηση των κόμβων σε σύνολα (clusters), όπου σε κάθε σύνολο εκλέγεται ένας κόμβος σαν αρχηγός. Η συχνή εκτέλεση ενός κατανεμημένου αλγορίθμου εκλογής αρχηγού, σε περίπτωση συχνής αλλαγής της σύστασης της ομάδας, μπορεί να προκαλέσει μείωση της απόδοσης του πρωτοκόλλου δρομολόγησης καθώς οι σταθμοί θα είναι απασχολημένοι περισσότερο με την επιλογή των αρχηγών παρά με την προώθηση των πακέτων. Για το λόγο αυτό το CGSR χρησιμοποιεί τον αλγόριθμο της ελάχιστης αλλαγής ομάδων Least Cluster Change (LCC), όπου ο κατανεμημένος αλγόριθμος εκλογής αρχηγού καλείται μόνο όταν έρχονται σε επαφή δύο αρχηγοί ή όταν ένας κόμβος χάσει επαφή με όλους τους αρχηγούς των υπολοίπων ομάδων.

Το CGSR βασίζεται στο πρωτόκολλο DSDV για την δρομολόγηση εντός των ομάδων. Παρόλα αυτά το τροποποιεί εφαρμόζοντας μία ιεραρχική προσέγγιση δρομολόγησης αρχηγών - πυλών για την δρομολόγηση από τον κόμβο πηγή στον κόμβο προορισμού. Οι κόμβοι πύλες βρίσκονται εντός της εμβέλειας επικοινωνίας δύο ή περισσότερων αρχηγών. Ένα πακέτο που στέλνεται από κάποιον κόμβο σε έναν άλλον κόμβο που ανήκει σε διαφορετική ομάδα αρχικά προωθείται στον αρχηγό της ομάδας του και στην συνέχεια από τον αρχηγό αυτόν προωθείται σε μία γειτονική ομάδα μέσω μίας πύλης που ενώνει τις δύο αυτές ομάδες. Η διαδικασία αυτή επαναλαμβάνεται έως ότου το πακέτο φτάσει στον αρχηγό της ομάδας του κόμβου προορισμού, ο οποίος και το παραδίδει σε αυτόν.

Για την λειτουργία του πρωτοκόλλου κάθε κόμβος διατηρεί δύο πίνακες, τον πίνακα μελών και τον πίνακα δρομολόγησης. Στον πίνακα μελών περιέχεται ο αρχηγός κάθε κόμβου του δικτύου. Αυτοί οι πίνακες μεταδίδονται από κάθε κόμβο περιοδικά με την χρήση του DSDV πρωτοκόλλου. Όταν ένας κόμβος λάβει έναν τέτοιο πίνακα από ένα γείτονα του ανανεώνει κατάλληλα τον δικό του πίνακα μελών. Ο πίνακας δρομολόγησης περιέχει τους επόμενους γειτονικούς κόμβους προς διάφορους προορισμούς. Όταν ένας κόμβος λάβει ένα πακέτο θα συμβουλευθεί τον πίνακα μελών του και τον πίνακα δρομολόγησης του για να καθορίσει τον πλησιέστερο αρχηγό ομάδας κατά μήκος της διαδρομής προς τον κόμβο προορισμού. Έπειτα θα ελέγξει τον πίνακα δρομολόγησης του για να καθορίσει το επόμενο βήμα που θα πραγματοποιήσει για να φτάσει το πακέτο στον επιλεγμένο αρχηγό, ο οποίος εντέλει το προωθεί στον κόμβο προορισμού.

### 3.4.3 Wireless Routing Protocol (WRP)

Το WRP είναι ένα πρωτόκολλο δρομολόγησης καθοδηγούμενο από πίνακα που βασίζεται, για την δρομολόγηση των πακέτων δεδομένων, στην ύπαρξη ενός αριθμού πινάκων σε κάθε κόμβο του δικτύου. Οι πίνακες αυτοί είναι οι εξής: ο πίνακας αποστάσεων, ο πίνακας δρομολόγησης, ο πίνακας κόστους συνδέσεων και ο πίνακας λίστας επαναμετάδοσης πακέτων ανανέωσης. Ο πίνακας αποστάσεων περιέχει την απόσταση (π.χ. τον αριθμό των ενδιαμέσων κόμβων) προς κάθε κόμβο προορισμού μέσω κάθε γείτονα καθώς και τον αντίστοιχο κόμβο που προηγείται του κόμβου προορισμού στη διαδρομή. Ο πίνακας δρομολόγησης περιέχει την μικρότερη απόσταση προς κάθε κόμβο προορισμού, τον επόμενο γειτονικό κόμβο της αντίστοιχης διαδρομής, τον κόμβο που προηγείται του κόμβου προορισμού στη διαδρομή και μία ετικέτα που δηλώνει αν αυτή η διαδρομή είναι σωστή, αν περιέχει έναν βρόχο ή αν δεν είναι πια έγκυρη. Η αποθήκευση σε κάθε εγγραφή του πίνακα δρομολόγησης του επόμενου κόμβου και του κόμβου που προηγείται του κόμβου προορισμού βοηθά στον εντοπισμό βρόχων δρομολόγησης. Ο πίνακας κόστους συνδέσεων περιέχει το κόστος της άμεσης ασύρματης σύνδεσης με κάθε γείτονα καθώς και τον αριθμό των καθορισμένων περιόδων που έχουν περάσει από τότε που το τελευταίο πακέτο λήφθηκε από τον κάθε γείτονα. Ο πίνακας λίστας επαναμετάδοσης πακέτων ανανέωσης περιέχει πληροφορίες για το ποιοι γείτονες δεν έχουν επιβεβαιώσει τα πακέτα ανανέωσης που έχουν σταλλεί από τον παρόντα κόμβο, έτσι ώστε αυτά να ξανασταλλούν.

Σε περίπτωση αλλαγής των γειτόνων ή λήψης ενός πακέτου ανανέωσης ένας κόμβος υπολογίζει τις νέες διαδρομές που προκύπτουν και ανανεώνει τον πίνακα απόστασης. Οι νέες διαδρομές μεταδίδονται πίσω στους αρχικούς κόμβους έτσι ώστε αυτοί να ενημερώσουν τους πίνακες του αναλόγως. Σε περίπτωση που κάποιες από αυτές τις διαδρομές είναι καλύτερες από αυτές του πίνακα δρομολόγησης ή ορισμένες διαδρομές του πίνακα δρομολόγησης δεν ισχύουν πια, τότε ανανεώνεται κατάλληλα και ο πίνακας. Παράλληλα ο κόμβος στέλνει και ένα νέο πακέτο ανανέωσης στους γείτονες του, το οποίο και καταγράφεται στον πίνακα λίστας επαναμετάδοσης πακέτων ανανέωσης. Κάθε γείτονας στον οποίον απευθύνεται αυτό το πακέτο και ο οποίος το λαμβάνει θα πρέπει να επιστρέψει μία επιβεβαίωση της λήψης του. Με την λήψη μίας τέτοιας επιβεβαίωσης ο κόμβος ανανεώνει κατάλληλα τον πίνακα λίστας επαναμετάδοσης πακέτων ανανέωσης.

Οι σταθμοί μαθαίνουν για την ύπαρξη των γειτόνων τους από την λήψη πακέτων ανανέωσης ή άλλων πακέτων. Αν ένας κόμβος δεν στείλει κάποιο πακέτο μέσα σε μία καθορισμένη περίοδο θα πρέπει να στείλει ένα πακέτο ύπαρξης για να δηλώσει στους γείτονες του την παρουσία του. Σε διαφορετική περίπτωση θεωρείται από τους γείτονες του ότι πια δεν υπάρχει άμεση ασύρματη

σύνδεση με αυτόν. Ακόμα όταν ένας κόμβος αντιληφθεί έναν νέο γείτονα στέλνει σε αυτόν ένα αντίγραφο των πληροφοριών του πίνακα δρομολόγησης του.

Κάθε φορά που ένας κόμβος αντιλαμβάνεται μία αλλαγή στην τοπολογία του δικτύου ελέγχει για την συνέπεια τους τις πληροφορίες που έχει λάβει από όλους του γείτονες του. Με τον τρόπο αυτό αποφεύγεται η δημιουργία βρόχων δρομολόγησης και επιτυγχάνεται συντομότερη σύγκλιση στις διαδρομές.

### 3.4.4 Ad-hoc On Demand Distance Vector (AODV)

Το AODV είναι ένα πρωτόκολλο δρομολόγησης καθοδηγούμενο από απαιτήσεις αποστολής που βασίζεται στο DSDV, το οποίο και βελτιώνει ελαχιστοποιώντας τον αριθμό των απαραίτητων πακέτων ελέγχου για την εύρεση και την συντήρηση διαδρομών. Στο πρωτόκολλο αυτό κάθε κόμβος του δικτύου, πέρα από την διεύθυνση του, διατηρεί έναν αριθμό ακολουθίας, ο οποίος αυξάνεται όταν ο κόμβος μάθει για μία τοπολογική αλλαγή στην γειτονιά του. Με την χρήση αυτού του αριθμού ακολουθίας εξασφαλίζεται ότι όλες οι διαδρομές δεν περιέχουν βρόχους και ότι διαθέτουν τις πιο πρόσφατες πληροφορίες δρομολόγησης. Το AODV μπορεί να προσφέρει unicast, multicast και broadcast δυνατότητες επικοινωνίας στους κόμβους του δικτύου.

Για την εύρεση unicast διαδρομών χρησιμοποιείται μία διαδικασία ανακάλυψης διαδρομών. Ένας κόμβος πηγή που θέλει να επικοινωνήσει με έναν κόμβο προορισμού και δεν γνωρίζει μία διαδρομή προς αυτόν, εκπέμπει στους γείτονες του ένα πακέτο αίτησης διαδρομής. Το πακέτο αυτό μεταδίδεται ομοίως από κόμβο σε κόμβο μέχρι να φτάσει στον κόμβο προορισμού ή σε κάποιον ενδιαμέσο κόμβο που γνωρίζει μία πρόσφατη διαδρομή προς τον προορισμό. Ένα πακέτο αίτησης διαδρομής περιλαμβάνει την διεύθυνση και τον αριθμό ακολουθίας του κόμβου πηγής και του κόμβου προορισμού. Ακόμα περιλαμβάνει και ένα μοναδικό προσδιοριστικό για την συγκεκριμένη αίτηση, το οποίο μαζί με την διεύθυνση του κόμβου πηγής προσδιορίζουν μοναδικά το πακέτο αυτό στο δίκτυο. Κάθε ενδιαμέσος κόμβος που λαμβάνει ένα πακέτο αίτησης διαδρομής καταγράφει αυτό το μοναδικό προσδιοριστικό καθώς και την διεύθυνση του κόμβου πηγής, με τον τρόπο αυτό αν λάβει αργότερα ξανά ένα αντίγραφο του ίδιου πακέτου το απορρίπτει. Ακόμα κάθε κόμβος που λαμβάνει ένα πακέτο αίτησης διαδρομής καταγράφει στον πίνακα δρομολόγησης του την διεύθυνση του γείτονα από τον οποίον λήφθηκε το πρώτο αντίγραφο του πακέτου, με τον τρόπο αυτό δημιουργείται μία αντίστροφη διαδρομή προς τον κόμβο πηγή.

Όταν το πακέτο αίτησης διαδρομής φτάσει στον κόμβο προορισμού ή σε έναν ενδιαμέσο κόμβο που γνωρίζει μία πρόσφατη διαδρομή προς τον προορισμό, τότε ο κόμβος αυτός επιστρέφει στον γείτονα, από τον οποίον έλαβε το πακέτο, ένα πακέτο απάντησης διαδρομής. Οι ενδιαμέσοι κόμβοι μπορούν να απαντή-



σουν σε ένα πακέτο αίτησης διαδρομής μόνο αν έχουν μία διαδρομή προς τον κόμβο προορισμού της οποίας ο αντίστοιχος αριθμός ακολουθίας του προορισμού είναι μεγαλύτερος ή ίσος με αυτόν που περιέχεται στο πακέτο αίτησης που λαμβάνουν. Με τον τρόπο αυτό αποφεύγεται η δημιουργία βρόχων δρομολόγησης μιας και διασφαλίζεται ότι η διαδρομή που ο ενδιαμέσος κόμβος γνωρίζει δεν είναι τόσο παλιά ώστε να περιέχει έναν προηγούμενο ενδιαμέσο κόμβο της διαδρομής που το πακέτο αίτησης έχει καθορίσει. Καθώς το πακέτο απάντησης διαδρομής επιστρέφει στον κόμβο πηγή ακολουθώντας την αντίστροφη διαδρομή κάθε ενδιαμέσος κόμβος καταγράφει στον πίνακα δρομολόγησης του τον γειτονικό κόμβο από τον οποίον έλαβε το πακέτο αυτό, με τον τρόπο αυτό σχηματίζεται μία ενεργή διαδρομή προς τον προορισμό. Παράλληλα αν ένας κόμβος δεν χρησιμοποιήσει μία διαδρομή του πίνακα δρομολόγησης του μέσα σε κάποια χρονική περίοδο τότε αυτή διαγράφεται από τον πίνακα δρομολόγησης. Το AODV υποστηρίζει την χρήση μόνο αμφίδρομων συνδέσεων μεταξύ κόμβων, για αυτό άλλωστε το πακέτο απάντησης διαδρομής μπορεί να κινείται κατά μήκος της διαδρομής που έχει δημιουργήσει ένα πακέτο αίτησης διαδρομής.

Για την συντήρηση των διαδρομών ύστερα από τοπολογικές αλλαγές ακολουθούνται διαφορετικά βήματα ανάλογα με το ποιοι κόμβοι της διαδρομής κινούνται. Αν κινηθεί ο κόμβος πηγής τότε αυτός εκκινεί μία νέα διαδικασία ανακάλυψης διαδρομής. Αν κινηθεί ένας ενδιαμέσος κόμβος ή ο κόμβος προορισμού τότε ο πιο πάνω γείτονας του αντιλαμβάνεται το γεγονός και ενημερώνει με ένα πακέτο αποτυχίας κάθε έναν από τους πιο πάνω γείτονες του που χρησιμοποιούν την σύνδεση που πια δεν υπάρχει. Με τον τρόπο αυτό το πακέτο αποτυχίας μπορεί να φτάσει μέχρι τον κόμβο (ή τους κόμβους) πηγής, ο οποίος αν εξακολουθεί να χρειάζεται μία διαδρομή προς τον κόμβο προορισμού μπορεί να εκκινήσει μία νέα διαδικασία ανακάλυψης διαδρομής.

Για την λειτουργία του πρωτοκόλλου είναι απαραίτητο κάθε κόμβος να διατηρεί γνώση των γειτόνων του. Ένας κόμβος μπορεί να αποκτήσει αυτή την γνώση από τα πακέτα δεδομένων και ελέγχου που λαμβάνει. Σε περίπτωση που ένας κόμβος δεν χρειαστεί να στείλει κανένα πακέτο δεδομένων ή ελέγχου για κάποια καθορισμένη περίοδο τότε για να δηλώσει την παρουσία του στέλνει σε όλους τους γειτονικούς του κόμβους ένα πακέτο γνωριμίας. Αν ένας κόμβος δεν λάβει κανένα είδους πακέτου από κάποιον γείτονα του για την καθορισμένη χρονική περίοδο τότε θεωρεί ότι αυτός δεν βρίσκεται πια εντός της εμβέλειας επικοινωνίας του.

### 3.4.5 Temporally Ordered Routing Algorithm (TORA)

Το TORA είναι ένα πρωτόκολλο δρομολόγησης καθοδηγούμενο από απαιτήσεις αποστολής που βασίζεται στη μέθοδο της αντιστροφής συνδέσεων. Είναι

ιδιαίτερα προσαρμοστικό και η βασική ιδέα του είναι ο περιορισμός των πακέτων ελέγχου σε ένα μικρό σύνολο κόμβων κοντά στην περιοχή όπου έγινε η αλλαγή στην τοπολογία του δικτύου. Για το σκοπό αυτό οι κόμβοι διατηρούν πληροφορίες για τις διαδρομές σε σχέση με τους γειτονικούς κόμβους. Το TORA βρίσκει πολλαπλές διαδρομές προς κάποιον κόμβο προορισμού, οι οποίες μπορεί να μην είναι βέλτιστες αλλά δεν περιέχουν κανέναν βρόχο. Το TORA βασίζεται στις διαδικασίες της ανακάλυψης διαδρομών, της συντήρησης διαδρομών και της διαγραφής διαδρομών.

Σε κάθε κόμβο αντιστοιχεί μία μετρική ύψους, αυτή χρησιμοποιείται κατά την ανακάλυψη και τη συντήρηση διαδρομών για τη δημιουργία ενός κατευθυνόμενου άκυκλου γραφήματος με κέντρο τον κόμβο προορισμού. Οι συνδέσεις μεταξύ των κόμβων του δικτύου χαρακτηρίζονται με μία κατεύθυνση (πάνω ή κάτω), σύμφωνα με την μετρική ύψους που έχουν οι γειτονικοί κόμβοι. Κατά την διαδικασία ανακάλυψης διαδρομών προς κάποιον κόμβο προορισμού διαμορφώνονται, με την αποστολή των κατάλληλων πακέτων ελέγχου, τα ύψοι των κόμβων με τέτοιο τρόπο ώστε να δημιουργηθεί το κατευθυνόμενο άκυκλο γράφημα. Όταν οι κόμβοι κινούνται κάποιες διαδρομές του κατευθυνόμενου άκυκλου γραφήματος καταστρέφονται και η διαδικασία της συντήρησης διαδρομών μπαίνει σε λειτουργία για την αποκατάσταση του κατευθυνόμενου άκυκλου γραφήματος προς τον ίδιο κόμβο προορισμού. Όταν ένας κόμβος βρεθεί χωρίς καμμία προς τα κάτω σύνδεση, λόγω σφάλματος, τότε υπολογίζει για τον εαυτό του μία νέα μετρική ύψους, την οποία και μεταδίδει στους γείτονες του. Η νέα μετρική ύψους εξαρτάται από το λογικό χρόνο της αποτυχίας της τελευταίας προς τα κάτω σύνδεσης του κόμβου, επομένως ο συγχρονισμός των κόμβων είναι πολύ σημαντικός για το πρωτόκολλο. Λαμβάνοντας τη νέα μετρική ύψους ορισμένες συνδέσεις αλλάζουν κατεύθυνση. Ακόμα όταν ένας κόμβος αντιληφθεί διάσπαση του δικτύου εκκινεί τη διαδικασία διαγραφής μη έγκυρων διαδρομών, πλημμυρίζοντας το δίκτυο με ένα πακέτο διαγραφής. Το ίδιο συμβαίνει και κατά τη διαδικασία της συντήρησης διαδρομών.

Στο TORA μπορούν να συμβούν διακυμάνσεις στην απόδοση όταν υπάρχει μία μαζική παραγωγή πακέτων ελέγχου από τους κόμβους του δικτύου. Αυτό μπορεί να γίνει όταν πολλοί κόμβοι ταυτόχρονα διαπιστώσουν διασπάσεις στο δίκτυο με αποτέλεσμα την εκκίνηση διαδικασιών διαγραφής διαδρομών και προσπάθειες ανακάλυψης νέων. Παρόλα αυτά αυτές οι διακυμάνσεις είναι προσωρινές και εν τέλει η μαζική παραγωγή πακέτων ελέγχου σταματά και οι διαδρομές συγκλίνουν.

### 3.4.6 Associativity Based Routing (ABR)

Το ABR είναι ένα πρωτόκολλο δρομολόγησης καθοδηγούμενο από απαιτήσεις αποστολής που ορίζει την μετρική του βαθμού σχετικής σταθερότητας για την

εύρεση των πιο σταθερών διαδρομών στο δίκτυο. Κάθε κόμβος εκπέμπει περιοδικά ένα πακέτο γνωριμίας που δηλώνει την παρουσία του. Κάθε γείτονας που λαμβάνει αυτό το πακέτο αυξάνει τον βαθμό σχετικής σταθερότητας, που διατηρεί για τον αντίστοιχο κόμβο, κατά μία μονάδα. Ο βαθμός αυτός μηδενίζεται όταν ο γειτονικός κόμβος ή ο ίδιος ο κόμβος κινηθεί εκτός εμβέλειας επικοινωνίας. Με την μετρική αυτή υποδηλώνεται η σταθερότητα της σύνδεσης με έναν γειτονικό κόμβο, όχι απαραίτητα και η ακινησία του. Συγκεκριμένα ένας υψηλός βαθμός σχετικής σταθερότητας υποδηλώνει χαμηλή κινητικότητα του γειτονικού κόμβου, ενώ ένας χαμηλός βαθμός υποδηλώνει υψηλή κινητικότητα. Το ABR βασίζεται στις διαδικασίες της ανακάλυψης διαδρομής, της συντήρησης διαδρομής και της διαγραφής διαδρομής.

Κατά την διαδικασία της ανακάλυψης διαδρομής ο κόμβος πηγή εκπέμπει ένα πακέτο ερώτησης και στην συνέχεια αναμένει για ένα πακέτο απάντησης, το οποίο περιέχει τη διαδρομή που αναζητά. Ένας κόμβος που λαμβάνει ένα πακέτο ερώτησης και οποίος δεν είναι ούτε ο κόμβος προορισμού ούτε γνωρίζει μία διαδρομή προς τον προορισμό, προσθέτει στο πακέτο την διεύθυνση του, τους βαθμούς σχετικής σταθερότητας των γειτόνων του καθώς και ορισμένες άλλες μετρικές ποιότητας και εκπέμπει το πακέτο. Ο κόμβος που θα παραλάβει αυτό το πακέτο διαγράφει όλους του βαθμούς σχετικής σταθερότητας που αφορούν κόμβους που δεν περιλαμβάνονται στην διαδρομή που το πακέτο περιέχει. Με τον τρόπο αυτό ο κόμβος προορισμού μπορεί, αφού λάβει πολλά πακέτα ερώτησης από τον ίδιο κόμβο πηγή, να επιλέξει την πιο σταθερή διαδρομή με βάση τους βαθμούς σχετικής σταθερότητας των κόμβων της. Σε περίπτωση που ορισμένες διαδρομές είναι το ίδιο σταθερές επιλέγεται εκείνη με το μικρότερο αριθμό κόμβων. Ο κόμβος προορισμού αφού επιλέξει την σταθερότερη διαδρομή στην συνέχεια επιστρέφει στον κόμβο πηγή ένα πακέτο απάντησης, ακολουθώντας την αντίστροφη διαδρομή. Με τον τρόπο αυτό μόνο ένα πακέτο απάντησης επιστρέφεται στον κόμβο πηγή. Ακόμα οι ενδιάμεσοι κόμβοι που λαμβάνουν και προωθούν αυτό το πακέτο απάντησης ανανεώνουν τους πίνακες δρομολόγησης τους με την διαδρομή που αυτό περιέχει.

Η διαδικασία της συντήρησης διαδρομής αποτελείται από διαφορετικά βήματα ανάλογα με το ποιοι κόμβοι της διαδρομής κινούνται. Η κίνηση του κόμβου πηγή έχει ως αποτέλεσμα την εκκίνηση, από τον ίδιο, μίας νέας διαδικασίας ανακάλυψης διαδρομής. Όταν κινηθεί ο κόμβος προορισμού ο αμέσως προηγούμενος κόμβος της διαδρομής διαγράφει από τον πίνακα δρομολόγησης του το μέρος αυτής που δεν λειτουργεί. Ακόμα ο κόμβος αυτός εκκινεί μία περιορισμένου εύρους διαδικασία ανακάλυψης διαδρομής προς τον κόμβο προορισμού. Αν αυτός ο ενδιάμεσος κόμβος δεν λάβει ένα πακέτο απάντησης μετά από μία καθορισμένη περίοδο τότε ενημερώνει τον παραπάνω κόμβο της διαδρομής για το πρόβλημα. Η διαδικασία αυτή επαναλαμβάνεται για τόσους ενδιάμεσους κόμβους της διαδρομής όσο είναι το μισό μήκος αυτής. Αν δεν βρεθεί

από κάποιον ενδιαμέσο κόμβο μία διαδρομή προς τον προορισμό, τότε ο μέσος αυτός κόμβος που ειδοποιείται για το πρόβλημα ενημερώνει αμέσως τον κόμβο πηγή. Ο κόμβος πηγή στην συνέχεια εκκινεί μία νέα διαδικασία ανακάλυψης διαδρομής. Η ίδια διαδικασία ακολουθείται και όταν ένας ενδιαμέσος κόμβος της διαδρομής κινηθεί. Η μόνη διαφορά είναι ότι στην περίπτωση αυτή ο αμέσως επόμενος κόμβος της διαδρομής στέλνει ένα πακέτο διαγραφής προς τον κόμβο προορισμού έτσι ώστε να διαγραφούν οι μη έγκυρες πια διαδρομές. Ακόμα το πρωτόκολλο μπορεί και επιλύει προβλήματα που προκύπτουν από πολλές ταυτόχρονες διαδικασίες ανακάλυψης και συντήρησης διαδρομής από τον ίδιο κόμβο πηγή προς τον ίδιο κόμβο προορισμού.

Όταν δεν υπάρχει ανάγκη για την συντήρηση μίας διαδρομής ο κόμβος πηγή εκπέμπει ένα πακέτο διαγραφής έτσι ώστε όλοι οι κόμβοι της διαδρομής να ενημερωθούν και να διαγράψουν την διαδρομή από τους πίνακες δρομολόγησης τους. Το πακέτο εκπέμπεται σε όλους τους γείτονες μιας και μπορεί η διαδρομή να μην είναι πια έγκυρη ύστερα από μία σειρά διαδικασιών συντήρησης διαδρομής από τους ενδιαμέσους κόμβους.

### 3.4.7 Support Routing Protocol (SRP)

Το SRP είναι ένα ημί-εξαναγκαστικό πρωτόκολλο δρομολόγησης. Βασίζεται σε ένα σύνολο κόμβων (υποστήριξη) για την μεταφορά των πακέτων δεδομένων μεταξύ κόμβων του δικτύου που κινούνται τυχαία και που δεν έχουν άμεση ασύρματη επικοινωνία. Τα μέλη της υποστήριξης κινούνται με ένα συγκεκριμένο τρόπο, καλύπτοντας γρήγορα ολόκληρη την περιοχή του δικτύου. Όταν ένας κόμβος του δικτύου θέλει να στείλει κάποια πακέτα δεδομένων σε έναν άλλο κόμβο του δικτύου, προωθεί τα πακέτα αυτά σε όποιο μέλος της υποστήριξης βρεθεί πρώτο μέσα στο εύρος μετάδοσης του. Μέχρι να συμβεί βέβαια αυτό τα πακέτα που εκκρεμούν διατηρούνται αποθηκευμένα στο κόμβο. Επίσης ο κόμβος αυτός παραλαμβάνει, από όποιο μέλος της υποστήριξης βρεθεί μέσα στο εύρος μετάδοσης του, όσα πακέτα δεδομένων απευθύνονται σε αυτόν καθώς και επιβεβαιώσεις λήψης πακέτων που έστειλε. Ακόμα όταν δύο κόμβοι που ανήκουν στην υποστήριξη έρθουν κοντά ανταλλάσσουν τα διάφορα πακέτα δεδομένων αλλά και τις επιβεβαιώσεις λήψης που έχουν να παραδώσουν.

Δύο πρωτόκολλα διαχείρισης της υποστήριξης για το SRP είναι το πρωτόκολλο του φιδιού (snake) και το πρωτόκολλο των δρομέων (runners). Η βασική ιδέα του πρωτοκόλλου του φιδιού είναι ότι οι κόμβοι της υποστήριξης είναι πάντα γειτονικοί ανά ζεύγη δημιουργώντας μία αλυσίδα και κινούνται συντονισμένα ακολουθώντας το κόμβο που βρίσκεται στη κεφαλή της αλυσίδας. Ο κόμβος αυτός εκτελεί έναν τυχαίο περίπατο, καλύπτοντας γρήγορα ολόκληρη τη περιοχή του δικτύου. Στο πρωτόκολλο των δρομέων αντίθετα υπάρχει πλήρης έλλειψη συντονισμού και κάθε κόμβος της υποστήριξης εκτελεί έναν ανεξάρτη-

το τυχαίο περίπατο που καλύπτει ολόκληρη τη περιοχή του δικτύου.

### 3.4.8 Παραλλαγές του Support Routing Protocol (SRP)

Από το SRP έχουν προκύψει διάφορα άλλα ημί-εξαναγκαστικά πρωτόκολλα. Τα πρωτόκολλα αυτά είναι κατάλληλα για δίκτυα με κάποια ιδιαίτερα χαρακτηριστικά. Το Hierarchical Support Routing Protocol (HSRP) μπορεί να χρησιμοποιηθεί σε δίκτυα όπου πολλοί κόμβοι είναι συγκεντρωμένοι σε υποδίκτυα (χαμηλό επίπεδο ιεραρχίας) και επικοινωνούν με την χρήση ενός περιορισμένου αριθμού σημείων πρόσβασης σε μία αφαιρή αλλά γρήγορη προϋπάρχουσα υποδομή (υψηλό επίπεδο ιεραρχίας). Η υποστήριξη στο πρωτόκολλο αυτό μεταφέρει πακέτα μεταξύ του χαμηλού και του υψηλού επιπέδου. Το Adaptive Support Routing Protocol (ASRP) μπορεί να χρησιμοποιηθεί σε δίκτυα στα οποία η περιοχή που αυτά καλύπτουν αλλάζει κατά την εκτέλεση του πρωτοκόλλου. Στο πρωτόκολλο αυτό ο αριθμός των μελών της υποστήριξης προσαρμόζεται συνεχώς στις αλλαγές του δικτύου. Τέλος υπάρχει και ο συνδυασμός αυτών των πρωτοκόλλων, το Hierarchical Adaptive Support Routing Protocol (HASRP) που μπορεί να χρησιμοποιηθεί σε αντίστοιχα δίκτυα.



## Κεφάλαιο 4

# Υλοποίηση του Dynamic Source Routing (DSR)

### 4.1 Γενικά

Το DSR πρωτόκολλο υλοποιήθηκε με την γλώσσα προγραμματισμού C++ και με την χρήση της βιβλιοθήκης LEDA. Κατά την υλοποίηση έγινε η υπόθεση, η οποία ήταν βέβαια πραγματική, της ύπαρξης ενός προσομοιωτή.

### 4.2 Γενικό μοντέλο προσομοιωτή

Ο προσομοιωτής αναλαμβάνει όλες τις λειτουργίες που βρίσκονται κάτω από το επίπεδο του πρωτοκόλλου DSR αλλά και άλλες. Είναι αυτός που γνωρίζει κάθε στιγμή την τοπολογία του δικτύου, αλλά και αυτός που φροντίζει για την τυχαία κίνηση των κόμβων του δικτύου. Ο προσομοιωτής είναι ακόμα εκείνος που δίνει σε κάθε κόμβο του δικτύου ένα μοναδικό προσδιοριστικό. Παράλληλα χειρίζεται την ανταλλαγή των μηνυμάτων, του πρωτοκόλλου, μεταξύ των γειτονικών κόμβων του δικτύου (broadcast). Συγκεκριμένα για κάθε κόμβο διατηρείται μία δομή όπου και αποθηκεύονται τα μηνύματα τα οποία στέλνονται σε αυτόν από τους γείτονες του. Ο προσομοιωτής αφαιρεί και δίνει σε έναν κόμβο, όταν αυτός το ζητήσει, το επόμενο μήνυμα που υπάρχει στην δομή αυτή. Τα μηνύματα αυτά είναι τα πακέτα του πρωτοκόλλου και τα οποία ανταλλάσσονται μεταξύ των κόμβων με την μορφή αλφαριθμητικών. Αυτά βέβαια τα αλφαριθμητικά έχουν μία συγκεκριμένη δομή ανάλογα με το είδος του πακέτου που αναπαριστούν. Τέλος ο προσομοιωτής διατηρεί ένα ρολόι, το οποίο ξεκινά όταν αρχίσει η προσομοίωση και η τιμή του οποίου επιστρέφεται σε έναν κόμβο όταν αυτός την ζητήσει.

Το γενικό μοντέλο του προσομοιωτή:

```

class simulator {
    private:
        int time;

    public:
        simulator();
        bool receive(int &, char *, bool);
        int present_time();
        int getMyID();
        bool send(int, char *);
};

```

### 4.3 Δομές δεδομένων

Κάθε κόμβος του δικτύου διατηρεί ένα σύνολο από δομές δεδομένων οι οποίες χρησιμοποιούνται κατά την εκτέλεση του πρωτοκόλλου. Συγκεκριμένα οι υπάρχουσες τέσσερις τέτοιες δομές, αυτές είναι ο `send_buffer`, ο `after_send_buffer`, η `route_cache` και η `rreq_cache`.

Ο `send_buffer` είναι η Μνήμη Προσωρινής Αποθήκευσης Πακέτων (ΜΠΑΠ). Η μνήμη αυτή υλοποιείται ως εξής:

```

class send_buffer {
    private:
        list<data_pck> s_buffer;
        dictionary<int, int> dest_timestamp;
        dictionary<int, int> rreq_tries;

        int exp_backoff_sb(int, int);
        int search_dest_id(int);

    public:
        void add_dpck(data_pck, int);
        list<data_pck> remove_dpck(int);
        list<int> update_timestamps(int);
};

```



Σε αυτή την υλοποίηση της ΜΠΑΠ όλα τα πακέτα αποθηκεύονται σε έναν κοινό καταχωρητή (`s_buffer`), ο οποίος χρησιμοποιείται σαν μία FIFO ουρά. Από εκεί και πέρα τα πακέτα δεδομένων αντιμετωπίζονται ομαδικά με βάση τον κόμβο - προορισμού τους και όχι ατομικά. Έτσι για τους διάφορους κόμβους - προορισμού, για τους οποίους υπάρχει τουλάχιστον ένα αντίστοιχο πακέτο δεδομένων στην ΜΠΑΠ, χρησιμοποιούνται δύο δομές. Στην πρώτη (`req_tries`) αποθηκεύεται ο αριθμός των ΑΙΔ που έχουν γίνει για κάθε τέτοιον κόμβο - προορισμού και για τις οποίες δεν έχει έρθει καμία ΑΠΔ. Στην δεύτερη (`dest_timestamp`) αποθηκεύεται η χρονική στιγμή που θα πρέπει να σταλεί η επόμενη ΑΙΔ για κάθε κόμβο - προορισμού.

Η συνάρτηση `exp_backoff_sb` χρησιμοποιείται για τον υπολογισμό της χρονικής στιγμής που θα πρέπει να σταλεί μία νέα ΑΙΔ για έναν κόμβο - προορισμού. Η συνάρτηση αυτή ουσιαστικά υλοποιεί τον `binary exponential backoff` αλγόριθμο. Με βάση αυτόν το χρονικό διάστημα μέχρι την επόμενη αποστολή της ΑΙΔ για τον ίδιο κόμβο - προορισμού είναι μεταξύ 0 και  $2^r - 1$ , όπου  $r$  είναι ο αριθμός των ΑΙΔ που έχουν σταλεί για τον συγκεκριμένο κόμβο - προορισμού χωρίς να έχει επιστρέψει καμία ΑΠΔ. Στην υλοποίηση το χρονικό αυτό διάστημα πολλαπλασιάζεται με μία σταθερά, η τιμή της οποίας ισούται με το πλήθος των κόμβων του δικτύου. Επιλέγουμε αυτή την τιμή επειδή μία ΑΙΔ και η αντίστοιχη ΑΠΔ στην χειρότερη περίπτωση θα διανύσουν συνολικά μήκος, σε ακμές, ίσο με δύο φορές το πλήθος των κόμβων του δικτύου (μείον δύο). Θεωρούμε ότι στον προσομοιωτή σε κάθε αύξηση του ρολογιού ένα πακέτο (δεδομένων και μη) κινείται κατά μία ακμή.

Οι συναρτήσεις `add_dpck` και `remove_dpck` χειρίζονται την προσθήκη και την αφαίρεση, αντίστοιχα, πακέτων δεδομένων από την ΜΠΑΠ. Κάθε φορά βέβαια δημιουργούνται, αφαιρούνται ή ενημερώνονται κατάλληλα οι εγγραφές στις αντίστοιχες δομές της ΜΠΑΠ. Η χρησιμοποίηση του κοινού καταχωρητή, των πακέτων δεδομένων, σαν μία FIFO ουρά γίνεται από την συνάρτηση `add_dpck`.

Η συνάρτηση `update_timestamps` επιστρέφει τα προσδιοριστικά των κόμβων - προορισμού, για τους οποίους θα πρέπει να σταλεί μία καινούρια ΑΙΔ. Για την εύρεση αυτών των κόμβων γίνεται σύγκριση κάθε τιμής της αντίστοιχης δομής της ΜΠΑΠ (`dest_timestamp`), για κάθε κόμβο - προορισμού, με τον παρόντα χρόνο, ο οποίος επιστρέφεται από τον προσομοιωτή. Παράλληλα η συνάρτηση αυτή αφαιρεί από την ΜΠΑΠ και απορρίπτει τα πακέτα δεδομένων στους κόμβους - προορισμού των οποίων έχουν σταλεί ΑΙΔ που ξεπερνούν έναν μέγιστο αριθμό, χωρίς να έχει επιστρέψει καμία ΑΠΔ. Σε κάθε περίπτωση αφαιρούνται ή ενημερώνονται κατάλληλα οι εγγραφές στις αντίστοιχες δομές της ΜΠΑΠ. Τέλος με την χρήση ενός μέγιστου αριθμού ΑΙΔ για κάθε κόμβο - προορισμού αποφεύγεται η ανάγκη χρήσης χρονοσφραγίδων που να σηματοδοτούν την απόρριψη των αντίστοιχων πακέτων δεδομένων τα οποία παραμένουν ακόμα στην ΜΠΑΠ.

Ο `after_send_buffer` είναι η μνήμη στην οποία αποθηκεύονται τα πακέτα δεδομένων, αφού σταλούν στον επόμενο κόμβο, περιμένοντας για την επιβεβαίωση λήψης τους `acknowledgment`. Η μνήμη αυτή υλοποιείται ως εξής:

```
class after_send_buffer {
    private:
        list<data_pck> s_buffer_a;
        dictionary<int, int> next_timestamp_a;
        dictionary<int, int> dpck_tries_a;

        int exp_backoff_sb_a(int, int);
        int dpck_next(data_pck);
        int search_dpck_id_a(int, int);
        int search_next_id_a(int);

    public:
        int add_dpck_a(data_pck, int);
        void remove_dpck_a(int, int, int);
        list<data_pck> error_dpck();
        list<data_pck> update_timestamps_a(int);
};
```

Η μνήμη αυτή έχει παρόμοιες δομές (`s_buffer_a`, `dpck_tries_a`, `next_timestamp_a`) αλλά και συναρτήσεις με αυτές της ΜΠΑΠ. Ωστόσο τα πακέτα δεδομένων, στην μνήμη αυτή, δεν ομαδοποιούνται με βάση τον κόμβο - προορισμού τους αλλά με βάση τον κόμβο, τον επόμενο κόμβο, ο οποίος θα πρέπει να επιβεβαιώσει την λήψη τους. Ακόμα μία επιβεβαίωση από έναν επόμενο κόμβο δεν προκαλεί την αφαίρεση όλων των αντίστοιχων πακέτων δεδομένων από την μνήμη αλλά μόνο του πακέτου στο οποίο απευθύνεται η επιβεβαίωση αυτή.

Η συνάρτηση `exp_backoff_sb_a` υπολογίζει την χρονική στιγμή που πακέτα δεδομένων της μνήμης θα ξανασταλούν σε έναν επόμενο κόμβο. Γίνεται πάλι χρήση του `binary exponential backoff` αλγορίθμου. Στην υλοποίηση το χρονικό διάστημα που υπολογίζεται από τον αλγόριθμο πολλαπλασιάζεται με μία σταθερά, η τιμή της οποίας ισούται με δύο. Επιλέγουμε αυτή την τιμή επειδή ένα πακέτο δεδομένων και η αντίστοιχη, άμεση και όχι έμμεση, επιβεβαίωση θα διανύσουν συνολικά μήκος, σε ακμές, ίσο με δύο. Θεωρούμε ότι στον προσομοιωτή σε κάθε αύξηση του ρολογιού ένα πακέτο (δεδομένων και μη) κινείται κατά μία ακμή.

Οι συναρτήσεις `add_dpck_a` και `remove_dpck_a` χειρίζονται την προσθήκη και την αφαίρεση, αντίστοιχα, πακέτων δεδομένων από την μνήμη. Κάθε φορά

βέβαια δημιουργούνται, αφαιρούνται ή ενημερώνονται κατάλληλα οι εγγραφές στις αντίστοιχες δομές της μνήμης. Η χρησιμοποίηση του κοινού καταχωρητή, των πακέτων δεδομένων, σαν μία FIFO ουρά γίνεται από την συνάρτηση `ad_dpck_a`.

Η συνάρτηση `error_dpck` αφαιρεί από την μνήμη και επιστρέφει τα πακέτα δεδομένων για τα οποία έχει ξεπεραστεί ο μέγιστος αριθμός φορών που έχουν ξανασταλεί σε κάποιον επόμενο κόμβο, χωρίς να έχει επιστρέψει μία επιβεβαίωση. Τα πακέτα αυτά χρησιμοποιούνται από την υλοποίηση του πρωτοκόλλου για την δημιουργία των μηνυμάτων Λάθους Διαδρομής (ΛΔ). Σε κάθε περίπτωση αφαιρούνται και οι εγγραφές στις αντίστοιχες δομές της μνήμης.

Η συνάρτηση `update_timestamps_a` επιστρέφει τα πακέτα δεδομένων της μνήμης που θα πρέπει να ξανασταλούν στον επόμενο κόμβο τους. Για την εύρεση αυτών των πακέτων γίνεται σύγκριση κάθε τιμής της αντίστοιχης δομής της μνήμης (`next_timestamp_a`), για κάθε επόμενο κόμβο, με τον παρόντα χρόνο, ο οποίος επιστρέφεται από τον προσομοιωτή. Για κάθε βέβαια επόμενο κόμβο που ο παραπάνω έλεγχος είναι θετικός, πέρα από το να στέλνονται τα αντίστοιχα πακέτα δεδομένων, ανανεώνονται κατάλληλα και οι εγγραφές στις δομές της μνήμης.

Η `route_cache` είναι Μνήμη Προσωρινής Αποθήκευσης Διαδρομών (ΜΠΑΔ). Η μνήμη αυτή υλοποιείται ως εξής:

```
class route_cache {  
  
    private:  
        list < list <int> > rc_source_route;  
  
    public:  
        int add_source_route ( list <int> );  
        int remove_source_routes ( int , int );  
        list <int> search_source_route ( int );  
  
};
```

Η ΜΠΑΔ υλοποιείται σαν ένας πίνακας από ΠΔ. Η υλοποίηση αυτή της ΜΠΑΔ έχει μία συνάρτηση εισαγωγής νέων ΠΔ, η οποία φροντίζει ώστε στην ΜΠΑΔ να περιέχονται μοναδικές ΠΔ. Ακόμα έχει μία συνάρτηση διαγραφής, από την ΜΠΑΔ, ΠΔ που περιέχουν μια συγκεκριμένη ακμή και η οποία χρησιμοποιείται όταν ο κόμβος λάβει ένα μήνυμα ΛΔ. Τέλος έχει και μία συνάρτηση αναζήτησης ΠΔ, στην ΜΠΑΔ, με βάση τον κόμβο - προορισμού, η οποία και επιστρέφει την πρώτη τέτοια ΠΔ που θα βρει.

Η `rreq_cache` (route request cache) είναι η μνήμη στην οποία αποθηκεύονται οι ΑΙΔ που ο κόμβος έχει λάβει. Ένας κόμβος που λαμβάνει μία ΑΙΔ την οποία την έχει λάβει ξανά στο παρελθόν θα πρέπει να την απορρίψει. Η μνήμη αυτή υλοποιείται ως εξής:

```
class rreq_cache {  
    private:  
        dictionary<int, list<int>> rreq_received;  
  
    public:  
        void add_rreq_id(int, int);  
        int search_rreq_id(int, int);  
};
```

Κάθε ΑΙΔ προσδιορίζεται μοναδικά από τον κόμβο που την δημιούργησε και την έστειλε αρχικά και από έναν αριθμό (`request_id`), ο οποίος είναι μοναδικός για όλες τις ΑΙΔ που δημιουργούνται από τον συγκεκριμένο κόμβο. Η μνήμη αυτή υλοποιείται σαν ένα σύνολο από διανύσματα κάθε ένα από τα οποία αντιστοιχεί σε κόμβο από τον οποίον έχει ληφθεί μία τουλάχιστον ΑΙΔ. Κάθε διάνυσμα περιέχει τα `request_id` των ΑΙΔ που έχουν ληφθεί και προέρχονται από τον συγκεκριμένο κόμβο. Τέλος η υλοποίηση της μνήμης αυτής περιλαμβάνει μία συνάρτηση εισαγωγής, στην μνήμη, των μοναδικών προσδιοριστικών μίας ΑΙΔ και μία συνάρτηση αναζήτησης στην μνήμη μίας ΑΙΔ με βάση αυτά τα προσδιοριστικά.

## 4.4 Τα πακέτα του πρωτοκόλλου

Για την λειτουργία του πρωτοκόλλου DSR ανταλλάσσονται μεταξύ των κόμβων διάφορων ειδών πακέτα. Αυτά τα πακέτα είναι οι ΑΙΔ, οι ΑΠΔ, τα ΛΔ, τα πακέτα δεδομένων και οι επιβεβαιώσεις (acknowledgments).

Μία ΑΙΔ υλοποιείται ως εξής:

```
class route_request_pck {  
  
    public:  
        int start_node;  
        int end_node;  
        int request_id;  
        int hop_limit;  
        int broken_edge_start_node;  
        int broken_edge_end_node;  
        list<int> rreq_source_route;  
  
        route_request_pck(int, int, int, int, int, int);  
        route_request_pck(char *);  
        void rreq_convert_to_text(char *);  
  
};
```

Το `start_node` είναι το προσδιοριστικό του κόμβου που δημιουργεί και στέλνει την ΑΙΔ. Το `end_node` είναι το προσδιοριστικό του κόμβου για τον οποίον αυτή ΑΙΔ αναζητεί μία ΠΔ. Το `request_id` είναι ένας αριθμός ο οποίος δίνεται από τον `start_node` και ο οποίος είναι μοναδικός για όλες τις ΑΙΔ από τον κόμβο αυτό. Το `hop_limit` δηλώνει τον μέγιστο αριθμό ενδιάμεσων κόμβων από τους οποίους αυτή η ΑΙΔ μπορεί να διαδωθεί. Αν το πεδίο αυτό αρχικοποιηθεί με την τιμή -1 τότε η αντίστοιχη ΑΙΔ δεν έχει κανένα περιορισμό στην διάδοση της. Οι `broken_edge_start_node` και `broken_edge_end_node` θα μπορούσαν να χρησιμοποιηθούν για την παράλληλη διάδοση ενός ΔΔ, παρόλα αυτά δεν χρησιμοποιούνται στην παρούσα υλοποίηση. Τέλος η `rreq_source_route` είναι η διαδρομή που έχει ανακαλυφθεί μέχρι στιγμής από αυτή την ΑΙΔ.

Η υλοποίηση της ΑΙΔ πέρα από την κανονική συνάρτηση εγκατάστασης (constructor) έχει και μία δεύτερη συνάρτηση εγκατάστασης. Αυτή παίρνει σαν είσοδο ένα μήνυμα που ο κόμβος έλαβε από έναν γείτονα του και το οποίο αντιστοιχεί σε μία ΑΙΔ. Η συνάρτηση αναλύει, αποκωδικοποιεί κατά κάποιον τρόπο, το μήνυμα αυτό και αρχικοποιεί τις αντίστοιχες μεταβλητές της ΑΙΔ. Τέλος υλοποιείται η συνάρτηση `rreq_convert_to_text` η οποία μετατρέπει, κωδικοποιεί, την πληροφορία μίας ΑΙΔ σε ένα μήνυμα. Αυτό είναι το μήνυμα που αποστέλεται από έναν κόμβο σε όλους τους γειτονικούς του, μέσω πάντα του προσομοιωτή. Ανάλογες συναρτήσεις υλοποιούνται και για τα υπόλοιπα πακέτα.

Μία ΑΠΔ υλοποιείται ως εξής:

```

class route_reply_pck {
    public:
        int start_node;
        int end_node;
        int request_id;
        int next_node_pos;
        int dest_node;
        list<int> reply_source_route;
        list<int> rrepl_source_route;

        route_reply_pck(int, int, int, int, list<int>,
                        list<int>);
        route_reply_pck(char *);
        void rrepl_convert_to_text(char *);
};

```

Το `start_node` είναι το προσδιοριστικό του κόμβου που δημιουργεί και στέλνει την ΑΠΔ. Το `end_node` είναι το προσδιοριστικό του κόμβου στον οποίον αυτή η ΑΠΔ απευθύνεται. Το `request_id` είναι το `request_id` της αντίστοιχης ΑΙΔ που προκάλεσε αυτή την ΑΠΔ από τον κόμβο. Η `reply_source_route` είναι η ΠΔ που χρησιμοποιείται από αυτή την ΑΠΔ για την επιστροφή της στον `end_node`. Η `rrepl_source_route` είναι η ΠΔ η οποία ανακαλύφθηκε από την αντίστοιχη ΑΙΔ. Το `next_node_pos` χρησιμοποιείται σε σχέση με την `reply_source_route` για να δηλώσει τον κόμβο που θα πρέπει στην συνέχεια να λάβει αυτή την ΑΠΔ. Το `dest_node` δηλώνει τον κόμβο για τον οποίο αυτή η ΑΠΔ περιέχει μία ΠΔ.

Ένα πακέτο ΛΔ υλοποιείται ως εξής:

```

class route_error_pck {

    public:
        int start_node;
        int end_node;
        int broken_edge_start_node;
        int broken_edge_end_node;
        int next_node_pos;
        list<int> error_source_route;

        route_error_pck(int, int, int, int, list<int>);
        route_error_pck(char *);
        void rerr_convert_to_text(char *);

};

```

Το `start_node` είναι το προσδιοριστικό του κόμβου που δημιουργεί και στέλνει το συγκεκριμένο πακέτο ΛΔ. Το `end_node` είναι το προσδιοριστικό του κόμβου στον οποίο αυτό το ΛΔ απευθύνεται. Οι `broken_edge_start_node` και `broken_edge_end_node` δηλώνουν αντίστοιχα την κορυφή και την ουρά της αμής του δικτύου που φαίνεται να μην δουλεύει. Η `error_source_route` είναι η ΠΔ που χρησιμοποιείται από αυτό το πακέτο ΛΔ για να φτάσει στον `end_node`. Το `next_node_pos` χρησιμοποιείται σε σχέση με την `error_source_route` για να δηλώσει τον κόμβο που θα πρέπει στην συνέχεια να λάβει αυτό το πακέτο ΛΔ.

Ένα πακέτο δεδομένων υλοποιείται ως εξής:

```

class data_pck {

    public:
        int start_node;
        int end_node;
        int next_node_pos;
        int dpck_id;
        list<int> data_source_route;
        bool dpck_saved;
        bool ack_bit;

        data_pck();
        data_pck(int, int, int, bool, bool, list<int>);
        data_pck(char *);
        void dpck_convert_to_text(char *);

};

```

Το `start_node` είναι το προσδιοριστικό του κόμβου που δημιουργεί και στέλνει το πακέτο δεδομένων. Το `end_node` είναι το προσδιοριστικό του κόμβου στον οποίον πρέπει τελικά να φτάσει αυτό το πακέτο δεδομένων. Το `dpck_id` είναι ένας αριθμός ο οποίος δίνεται από τον `start_node` και ο οποίος είναι μοναδικός για όλα τα πακέτα δεδομένων που ξεκινούν από τον κόμβο αυτό. Η `data_source_route` είναι η ΠΔ που χρησιμοποιείται από αυτό το πακέτο δεδομένων για να φτάσει στον `end_node`. Το `next_node_pos` χρησιμοποιείται σε σχέση με την `data_source_route` για να δηλώσει τον κόμβο που θα πρέπει στην συνέχεια να λάβει αυτό το πακέτο δεδομένων. Η μεταβλητή `dpck_saved` δηλώνει αν το συγκεκριμένο πακέτο δεδομένων έχει διασωθεί από κάποιον κόμβο του δικτύου. Τέλος η μεταβλητή `ack_bit` δηλώνει το αν ένας ενδιαμέσος κόμβος με την λήψη αυτού του πακέτου θα πρέπει να στείλει μία επιβεβαίωση λήψης στον κόμβο από τον οποίον και το έλαβε.

Μία επιβεβαίωση `acknowledgment` υλοποιείται ως εξής:

```
class ack_data_pck {
    public:
        int start_node;
        int end_node;
        int dpck_id;
        int init_start_node;

        ack_data_pck(int , int , int , int);
        ack_data_pck(char *);
        void apck_convert_to_text(char *);
};
```

Το `start_node` είναι το προσδιοριστικό του κόμβου που δημιουργεί και στέλνει αυτή την επιβεβαίωση. Το `end_node` είναι το προσδιοριστικό του κόμβου στον οποίον αυτή η επιβεβαίωση απευθύνεται. Το `init_start_node` είναι το προσδιοριστικό του κόμβου που δημιούργησε και έστειλε το πακέτο δεδομένων που προκάλεσε αυτή την επιβεβαίωση. Το `dpck_id` είναι το `dpck_id` αυτού του πακέτου δεδομένων.

## 4.5 Υλοποίηση του πρωτοκόλλου

Η υλοποίηση του DSR πρωτοκόλλου στηρίζεται σε μία αφηρημένη τάξη (abstract class), η οποία δηλώνει ένα γενικό πρωτόκολλο δρομολόγησης για ασύρματα ad-hoc δίκτυα. Η τάξη αυτή έχει την εξής δομή:



```

class adhoc_routing_protocol {
    protected:
        virtual void init() = 0;
    public:
        adhoc_routing_protocol();
        virtual void run(simulator &) = 0;
        virtual ~adhoc_routing_protocol();
};

```

Η τάξη που στην πράξη υλοποιεί το DSR πρωτόκολλο κληρονομεί από αυτή την αφηρημένη τάξη και ορίζει όλες τις γνήσιες εικονικές συναρτήσεις αυτής. Σε κάθε κόμβο του δικτύου αντιστοιχεί ένα αντικείμενο αυτής της τάξης. Η τάξη αυτή έχει την εξής δομή:

```

class dsr_protocol : public adhoc_routing_protocol {
    private:
        send_buffer          my_send_buffer;
        after_send_buffer    my_after_send_buffer;
        route_cache          my_route_cache;
        rreq_cache           my_rreq_cache;

        int data_pck_id;
        int route_req_id;

        void route_request_handle(simulator &, char *);
        void data_packet_handle(simulator &, char *);
        void route_reply_handle(simulator &, char *);
        void route_error_handle(simulator &, char *);
        void ack_data_packet_handle(simulator &, char *);
        void send_buffer_handle(simulator &);
        void after_send_buffer_handle(simulator &);
        void enrich_route_cache(list<int>, int);

    protected:
        void init();

    public:
        dsr_protocol();
        void run(simulator &);
};

```

Στην τάξη `dsr_protocol` ορίζονται ένα σύνολο από ιδιωτικές μεταβλητές και συναρτήσεις. Οι μεν μεταβλητές αντιστοιχούν κυρίως στις δομές που θα πρέπει να υπάρχουν σε κάθε κόμβο του δικτύου για την λειτουργία του πρωτοκόλλου. Αυτές οι δομές είναι ο `send_buffer`, ο `after_send_buffer`, η `route_cache` και η `req_cache`. Οι δε συναρτήσεις αναλαμβάνουν την διαχείριση των δομών αυτών αλλά και την διαχείριση των διαφόρων πακέτων, του πρωτοκόλλου, που φτάνουν στον κόμβο. Συγκεκριμένα οι συναρτήσεις `route_request_handle`, `data_packet_handle`, `route_reply_handle`, `route_error_handle`, `ack_data_packet_handle` χειρίζονται τα αντίστοιχα πακέτα και αποστέλλουν στους γειτονικούς κόμβους όποια πακέτα χρειάζεται. Ακόμα οι συναρτήσεις `send_buffer_handle` και `after_send_buffer_handle` εξετάζουν τις αντίστοιχες δομές και στέλνουν τις ΑΙΔ, τα πακέτα δεδομένων και τα πακέτα ΑΔ που πρέπει, με βάση τους μηχανισμούς του `exponential backoff` που αυτές οι δομές διαθέτουν. Οι λειτουργία αυτών των δύο συναρτήσεων δεν επηρεάζεται, άμεσα τουλάχιστον, από την λήψη ή μη ενός νέου πακέτου.

Η κυριότερη συνάρτηση της τάξης αυτής είναι η `run`, η οποία είναι δημόσια. Αυτή υλοποιεί έναν γύρο του πρωτοκόλλου, ο οποίος εκτελείται σε έναν κόμβο του δικτύου. Η συνάρτηση αυτή έχει την εξής δομή:

```
void run(simulator &sim){

    char msg[Msg.Size];
    int packet_type;
    int tmp_id;

    if(sim.receive(tmp_id, msg, false)){

        istrstream is(msg);
        is >> packet_type;
        switch (packet_type) {

            case Data_Packet:
                data_packet_handle(sim, msg);
                break;
            case Route_Request:
                route_request_handle(sim, msg);
                break;
            case Route_Reply:
                route_reply_handle(sim, msg);
                break;
            case Route_Error:
                route_error_handle(sim, msg);
                break;
            case Ack_Data_Packet:
                ack_data_packet_handle(sim, msg);
                break;
            default:
                break;

        }

        send_buffer_handle(sim);
        after_send_buffer_handle(sim);

    }

}
```

Ο κάθε κόμβος αρχικά διαβάζει το επόμενο μήνυμα το οποίο απευθύνεται σε αυτόν και το χειρίζεται ανάλογα με το είδος του πακέτου στο οποίο το μήνυμα αυτό αντιστοιχεί. Στη συνέχεια γίνεται ο χειρισμός των δομών `send_buffer` και `after_send_buffer` του κόμβου, σύμφωνα πάντα με το πρωτόκολλο. Στην πραγματική εκτέλεση του πρωτοκόλλου σε κάθε κόμβο η παραπάνω διαδικασία επαναλαμβάνεται συνεχώς.

## 4.6 Εκτέλεση του πρωτοκόλλου

Για την εκτέλεση του πρωτοκόλλου απαιτείται ένα μικρό πρόγραμμα το οποίο θα τρέχει σε κάθε κόμβο του δικτύου, όπως το παρακάτω:

```
int main(int argc , char **argv){  
  
    simulator sim;  
    dsr_protocol dsr;  
  
    while(1){  
  
        dsr.run(sim);  
  
    }  
  
    return 0;  
  
}
```

Το πρόγραμμα αυτό αρχικά ορίζει ένα αντικείμενο της τάξης του προσομοιωτή και ένα αντικείμενο της τάξης που υλοποιεί το DSR πρωτόκολλο. Στη συνέχεια καλείται μέσα σε έναν άπειρο βρόχο η συνάρτηση `run`, η οποία υλοποιεί έναν γύρο του πρωτοκόλλου.

Η δημιουργία των πακετών δεδομένων, τα οποία ουσιαστικά εκκινούν τις διαδικασίες του πρωτοκόλλου, μπορεί να γίνεται στο παραπάνω πρόγραμμα. Δηλαδή σε τυχαίες χρονικές στιγμές θα δημιουργούνται νέα πακέτα δεδομένων με τυχαίο κόμβο - προορισμού. Τα πακέτα αυτά θα πρέπει να έχουν σαν `src_id` την τιμή -1, έτσι ώστε να διακρίνονται από τα πακέτα δεδομένων για τα οποία ήδη έχει βρεθεί μία ΠΔ, με την βοήθεια του πρωτοκόλλου και τα οποία προωθούνται μέχρι τον κόμβο - προορισμού τους.

## 4.7 Ειδικές περιπτώσεις και παραδοχές της υλοποίησης

Παραθέτουμε διάφορες ειδικές περιπτώσεις και παραδοχές της υλοποίησης του DSR πρωτοκόλλου:

- Θεωρούμε ότι σε κάθε περίπτωση είναι δυνατή η αμφίδρομη επικοινωνία μεταξύ γειτονικών κόμβων. Αυτό έχει σαν αποτέλεσμα ο υπολογισμός της ΠΔ επιστροφής μίας ΑΠΔ, ενός ΛΔ ή ακόμα και μίας επιβεβαίωσης να γίνεται αντιστρέφοντας ολόκληρη ή κάποιο μέρος της ΠΔ που περιέχεται στην ΑΙΔ ή στο πακέτο δεδομένων αντίστοιχα.

- Οι κόμβοι ακούνε και αξιοποιούν όσο είναι δυνατόν κάθε πακέτο που λαμβάνουν ακόμα και αν αυτό δεν απευθύνεται σε αυτούς (promiscuous mode).
- Κάθε κόμβος που λαμβάνει ένα πακέτο (ΑΙΔ, ΑΠΔ, ΛΔ, πακέτο δεδομένων) χρησιμοποιεί την πληροφορία ΠΔ που περιέχεται μέσα σε αυτό. Από αυτή την ΠΔ μπορεί να προκύψουν πολλές ΠΔ από τον κόμβο αυτό προς άλλους κόμβους του δικτύου. Σε κάθε περίπτωση οι ΠΔ που προκύπτουν αποθηκεύονται στην ΜΠΑΔ του κόμβου αυτού. Για την εύρεση και την αποθήκευση αυτών των ΠΔ χρησιμοποιείται η συνάρτηση `enrich_route_cache` της τάξης που υλοποιεί το πρωτόκολλο.
- Κάθε κόμβος που λαμβάνει ένα ΛΔ, ανεξάρτητα από το αν αυτό απευθύνεται σε αυτόν, αφαιρεί από την ΜΠΑΔ του τις ΠΔ που περιέχουν την χαλασμένη ακμή.
- Κάθε κόμβος που προωθεί ένα πακέτο δεδομένων ζητεί μία άμεση επιβεβαίωση της λήψης από τον επόμενο κόμβο. Παρόλα αυτά οι κόμβοι μπορούν να λάβουν και να χρησιμοποιήσουν και την αντίστοιχη έμμεση επιβεβαίωση. Αυτή η ένεση επιβεβαίωση μπορεί να προέλθει όχι μόνο από τον επόμενο κόμβο αλλά και από έναν άλλο κόμβο που έπετα στην ΠΔ του πακέτου δεδομένων.
- Αν ένας κόμβος λάβει ένα πακέτο δεδομένων που ήδη υπάρχει στον `after_send_buffer` του τότε δεν το προωθεί, αφού αυτό γίνεται χρησιμοποιώντας τους μηχανισμούς του `after_send_buffer` (exponential back-off). Παρόλα αυτά αν σε αυτό το πακέτο δεδομένων έχει τεθεί το `ack_bit` τότε θα επιστραφεί κανονικά μία επιβεβαίωση λήψης.
- Αν ένας κόμβος λάβει μία ΑΙΔ για την οποία έχει στην ΜΠΑΔ του μία ΠΔ που την ικανοποιεί, τότε επιστρέφει μία ΑΠΔ μην προωθώντας αυτή την ΑΙΔ. Πριν την επιστροφή της ΑΠΔ γίνεται έλεγχος αν η νέα ΠΔ που προκύπτει περιέχει κάποιον κύκλο.
- Δεν υλοποιείται ο μηχανισμός καθυστέρησης της αποστολής ΑΠΔ από τους κόμβους, για την αποφυγή μεγάλου αριθμού από ΑΠΔ.
- Κάθε ΑΙΔ περιέχει το πεδίο `hop_limit`, το οποίο δηλώνει τον μέγιστο αριθμό ενδιάμεσων κόμβων από τους οποίους η ΑΙΔ μπορεί να διαδωθεί. Αν το πεδίο αυτό αρχικοποιηθεί με την τιμή -1 τότε η αντίστοιχη ΑΙΔ δεν έχει κανένα περιορισμό στην διάδοση της.
- Δεν υλοποιείται ο μηχανισμός διάσωσης ενός πακέτου δεδομένων όταν αυτό δεν μπορεί να προωθηθεί λόγω κάποιας χαλασμένης ακμής.

- Έχει υλοποιηθεί ο μηχανισμός που επιτρέπει σε έναν ενδιάμεσο κόμβο της ΠΔ ενός πακέτου δεδομένων να στείλει μία ΑΠΔ στον αρχικό κόμβο αν διαπιστώσει πως αυτή η ΠΔ μπορεί να μικρύνει.
- Δεν υλοποιείται ο μηχανισμός διασποράς ενός ΛΔ από μία ΑΙΔ.
- Στην ΜΠΑΔ αποθηκεύονται μόνο ΠΔ και όχι συγκεκριμένες πληροφορίες για την λειτουργία ή μη ακμών του δικτύου.

## Κεφάλαιο 5

# Υλοποίηση του Zone Routing Protocol (ZRP)

### 5.1 Γενικά

Το ZRP πρωτόκολλο υλοποιήθηκε με την γλώσσα προγραμματισμού C++ και με την χρήση της βιβλιοθήκης LEDA. Κατά την υλοποίηση έγινε η υπόθεση, η οποία ήταν βέβαια πραγματική, της ύπαρξης ενός προσομοιωτή.

### 5.2 Γενικό μοντέλο προσομοιωτή

Ο προσομοιωτής αναλαμβάνει όλες τις λειτουργίες που βρίσκονται κάτω από το επίπεδο του πρωτοκόλλου ZRP αλλά και άλλες. Είναι αυτός που γνωρίζει την τοπολογία του δικτύου, αλλά και αυτός που φροντίζει για την τυχαία κίνηση των κόμβων του δικτύου. Έτσι ο προσομοιωτής γνωρίζει κάθε στιγμή και τους γείτονες κάθε κόμβου, υλοποιώντας έμμεσα το NDP. Ο προσομοιωτής είναι ακόμα εκείνος που δίνει σε κάθε κόμβο του δικτύου ένα μοναδικό προσδιοριστικό. Παράλληλα χειρίζεται την ανταλλαγή των μηνυμάτων, του πρωτοκόλλου, μεταξύ των γειτονικών κόμβων του δικτύου. Συγκεκριμένα για κάθε κόμβο διατηρείται μία δομή όπου και αποθηκεύονται τα μηνύματα τα οποία στέλνονται σε αυτόν από τους γείτονες του. Ο προσομοιωτής αφαιρεί και δίνει σε έναν κόμβο, όταν αυτός το ζητήσει, το επόμενο μήνυμα που υπάρχει στην δομή αυτή. Τα μηνύματα αυτά είναι τα πακέτα του πρωτοκόλλου και τα οποία ανταλλάσσονται μεταξύ των κόμβων με την μορφή αλφαριθμητικών. Αυτά βέβαια τα αλφαριθμητικά έχουν μία συγκεκριμένη δομή ανάλογα με το είδος του πακέτου που αναπαριστούν. Τέλος ο προσομοιωτής διατηρεί ένα ρολόι, το οποίο ξεκινά όταν αρχίσει η προσομοίωση και η τιμή του οποίου επιστρέφεται σε έναν κόμβο όταν αυτός την ζητήσει.

Το γενικό μοντέλο του προσομοιωτή:

```
class simulator {  
  
    private:  
        int time;  
  
    public:  
        simulator();  
        bool receive(int &, char *, bool);  
        int present_time();  
        int getMyID();  
        bool send(int, char *);  
        list<int> get_neighbours();  
  
};
```

### 5.3 Δομές δεδομένων

Κάθε κόμβος του δικτύου διατηρεί ένα σύνολο από δομές δεδομένων οι οποίες χρησιμοποιούνται κατά την εκτέλεση του πρωτοκόλλου. Οι κυριότερες από τις δομές αυτές είναι ο `send_buffer`, η λίστα των γειτόνων του κόμβου, ο πίνακας δρομολόγησης του IERP, ο πίνακας δρομολόγησης του IARP, ο Πίνακας Κατάστασης Ακμών (ΠΙΚΑ) και ο Πίνακας Κάλυψης Αιτήσεων (ΠΚΑ).

Ο `send_buffer` είναι η Μνήμη Προσωρινής Αποθήκευσης Πακέτων (ΜΠΑΠ). Η μνήμη αυτή υλοποιείται ως εξής:



```
class send_buffer {  
  
    private:  
        list<data_pck> s_buffer;  
        dictionary<int, int> dest_timestamp;  
        dictionary<int, int> rreq_tries;  
  
        int exp_backoff_sb(int, int);  
        int search_dest_id(int);  
  
    public:  
        void add_dpck(data_pck, int);  
        list<data_pck> remove_dpck(int);  
        list<int> update_timestamps(int);  
  
};
```

Η μνήμη αυτή χρησιμοποιείται από το επίπεδο του δικτύου για την προσωρινή αποθήκευση πακέτων δεδομένων για τα οποία δεν έχει βρεθεί ακόμα διαδρομή προς τον τελικό προορισμό τους. Η λειτουργία της μνήμης αυτής είναι ακριβώς ίδια με την λειτουργία της αντίστοιχης μνήμης στην περίπτωση του DSR πρωτοκόλλου.

Η λίστα των γειτόνων του κόμβου αρχικοποιείται με την βοήθεια του προσομοιωτή και χρησιμοποιείται καθόλη την διάρκεια εκτέλεσης του πρωτοκόλλου για τον εντοπισμό του γεγονότος της αλλαγής των γειτόνων του κόμβου. Στην περίπτωση αυτή η λίστα ξανααρχικοποιείται με τους νέους γείτονες του κόμβου.

Ο πίνακας δρομολόγησης του IERP υλοποιείται ως εξής:

```

class ierp_routing_table {
    private:
        list < list <int> > ierp_rt;

        int duplicate_nodes_exist(list <int>);

    public:
        void add_route(list <int>, int);
        void remove_routes(int, int);
        int search_route(int);
        list <int> return_shortest_route_to_dest(int);
        list < list <int> > return_all_routes_to_dest(int);
        list < list <int> > contents();
        void update(dictionary <int, list <int> >);
};

```

Ο πίνακας δρομολόγησης υλοποιείται σαν μία λίστα από διαδρομές, οι οποίες βέβαια μπορεί να έχουν και τον ίδιο προορισμό. Οι διάφορες συναρτήσεις του πίνακα φροντίζουν για την προσθήκη νέων διαδρομών (`add_route`), την αφαίρεση διαδρομών που περιέχουν μία συγκεκριμένη ακμή (`remove_routes`), για την αναζήτηση και επιστροφή διαδρομών του πίνακα (`search_route`, `return_shortest_route_to_dest`, `return_all_routes_to_dest`, `contents`) και τέλος φροντίζουν για την συντήρηση των διαδρομών του πίνακα χρησιμοποιώντας τις πληροφορίες του πίνακα δρομολόγησης του IARP (`update`).

Ο πίνακας δρομολόγησης του IARP υλοποιείται ως εξής:

```

class iarp_routing_table {
    private:
        dictionary <int, list <int> > iarp_rt;

        node rz_add_node(GRAPH <int, int> &, int);
        void compute_routing_zone(GRAPH <int, int> &,
                                  link_state_table);

    public:
        void recompute(link_state_table, int);
        dictionary <int, list <int> > contents();
};

```

Ο πίνακας δρομολόγησης υλοποιείται σαν μία λίστα από συντομότερες διαδρομές, μία για κάθε κόμβο της ΖΔ του αντίστοιχου κόμβου. Οι διάφορες συναρτήσεις του πίνακα φροντίζουν για τον επαναυπολογισμό του πίνακα χρησιμοποιώντας τον ΠΙΚΑ (recompute) και για την επιστροφή των περιεχομένων του πίνακα (contents). Για τον υπολογισμό των διαδρομών του πίνακα αρχικά χρησιμοποιείται ο ΠΙΚΑ για την κατασκευή του γράφου που αναπαριστά την ΖΔ του αντίστοιχου κόμβου και στην συνέχεια με την βοήθεια του αλγορίθμου Διχαστρα υπολογίζονται οι συντομότερες διαδρομές προς τους διάφορους κόμβους.

Ο Πίνακας Κατάστασης Ακμών (ΠΙΚΑ) υλοποιείται ως εξής:

```
class link_state_table {  
  
    private:  
        dictionary<int , int> ls_timestamps ;  
        dictionary<int , link_state_pck > link_states ;  
  
    public:  
        void add_link_state(link_state_pck , int) ;  
        int remove_old_link_states(int) ;  
        dictionary<int , list<int> >  
            link_state_table_contents () ;  
  
};
```

Ο ΠΙΚΑ περιέχει τα Πακέτα Κατάστασης Ακμών (ΠΑΚΑ) που έχουν ληφθεί από τους κόμβους της ΖΔ του αντίστοιχου κόμβου. Βέβαια κάθε φορά που από έναν κόμβο καταφτάνει ένα νέο ΠΑΚΑ αυτό αντικαθιστά, αν υπάρχει, το παλιό ΠΑΚΑ που είχε προέλθει πάλι από τον ίδιο κόμβο. Ακόμα σε κάθε νέο πακέτο ανατίθεται και μία χρονοσφραγίδα. Οι διάφορες συναρτήσεις του πίνακα φροντίζουν για την προσθήκη των νέων ΠΑΚΑ (add\_link\_state), για την αφαίρεση των ΠΑΚΑ που βρίσκονται στον πίνακα για ένα χρονικό διάστημα μεγαλύτερο από το καθορισμένο (remove\_old\_link\_states) και για την επιστροφή των περιεχομένων του πίνακα (link\_state\_table\_contents).

Ο Πίνακας Κάλυψης Αιτήσεων (ΠΚΑ) υλοποιείται ως εξής:

```

class query_coverage_table {
    private:
        list <query_coverage_entry > queries_coverage;

    public:
        query_coverage_entry add_query(route_request_pck ,
                                       GRAPH<int , int > , int);
        int search_query(int , int);
        int search_query(int);
        query_coverage_entry return_query(int , int);
        query_coverage_entry return_query(int);
        void update_coverage(int , int , list <int >);
};

```

Ο ΠΚΑ υλοποιείται σαν μία λίστα από αιτήσεις ανακάλυψης διαδρομής που έχουν ληφθεί. Κάθε αίτηση ανακάλυψης διαδρομής που έχει ληφθεί αποθηκεύεται στον πίνακα ως εξής:

```

class query_coverage_entry {
    public:
        int query_source;
        int query_id;
        int brp_id;
        GRAPH<int , int > rz_topology;
        list <int > coverage;

        query_coverage_entry ();
        query_coverage_entry (int , int , int ,
                              GRAPH<int , int > , list <int >);
};

```

Για κάθε νέα αίτηση καταγράφονται, πέρα από τα προσδιοριστικά της, η τοπολογία της ΖΔ του κόμβου την στιγμή που η αίτηση αυτή λήφθηκε για πρώτη φορά από τον κόμβο αλλά και οι κόμβοι οι οποίοι θεωρούνται καλυμμένοι για την συγκεκριμένη αίτηση.

Οι διάφορες συναρτήσεις του ΠΚΑ φροντίζουν για την προσθήκη στον πίνακα νέων αιτήσεων (add\_query), για την αναζήτηση αιτήσεων και την επιστροφή πληροφοριών σχετικά με αυτές (search\_query, return\_query) και για την προσθήκη και άλλων κόμβων στους καλυμμένους κόμβους μίας αίτησης (update\_coverage).

## 5.4 Τα πακέτα του πρωτοκόλλου

Για την λειτουργία του πρωτοκόλλου ZRP ανταλλάσσονται μεταξύ των κόμβων διαφόρων ειδών πακέτα. Αυτά τα πακέτα είναι οι αιτήσεις ανακάλυψης διαδρομής (Πακέτο Αίτησης Ανακάλυψης Διαδρομής - ΠΑΑΔ), οι απαντήσεις ανακάλυψης διαδρομής, τα Πακέτα Κατάστασης Ακμών (ΠΑΚΑ) και τα πακέτα δεδομένων.

Ένα ΠΑΑΔ υλοποιείται ως εξής:

```
class route_request_pck {  
  
    public:  
        int query_source;  
        int query_dest;  
        int query_id;  
        int prev_bordercast;  
        list<int> prev_bordercast_neighbours;  
        list<int> query_source_route;  
  
        route_request_pck();  
        route_request_pck(int, int, int, int, list<int>, list<int>);  
        route_request_pck(char *);  
        void rreq_pck_convert_to_text(char *);  
  
};
```

Στο ΠΑΑΔ αυτό δεν υλοποιείται ξεχωριστά το Ενσωματωμένο Πακέτο αλλά τα πεδία του ενσωματώνονται στο ΠΑΑΔ. Έτσι τα πρωτόκολλα ΒΡΠ και ΙΕΡ-Π διαβάζουν από ένα ΠΑΑΔ τα πεδία που τους ενδιαφέρουν. Το query\_source είναι το προσδιοριστικό του κόμβου που δημιουργεί και στέλνει την αίτηση. Το query\_dest είναι το προσδιοριστικό του κόμβου για τον οποίο αυτή αίτηση αναζητεί μία διαδρομή. Το query\_id είναι ένας αριθμός ο οποίος δίνεται από τον query\_source και ο οποίος είναι μοναδικός για όλες τις αιτήσεις από τον κόμβο αυτό. Το prev\_bordercast είναι το προσδιοριστικό του κόμβου ο οποίος τελευταίος προώθησε αυτή την αίτηση. Οι prev\_bordercast\_neighbours είναι οι γείτονες του prev\_bordercast στο multicast-bordercast δέντρο του, οι οποίοι θα πρέπει να λάβουν την αίτηση αυτή. Με την χρήση του πεδίου αυτού ο κόμβος που λαμβάνει μία αίτηση δεν χρειάζεται να υπολογίσει τους γείτονες του prev\_bordercast αλλά απλώς να ελέγξει αυτό το πεδίο. Τέλος η query\_source\_route είναι η διαδρομή που έχει ανακαλυφθεί μέχρι στιγμής από αυτή την αίτηση.

Η υλοποίηση του ΠΑΑΔ πέρα από την κανονική συνάρτηση εγκατάστασης (constructor) έχει και μία δεύτερη συνάρτηση εγκατάστασης. Αυτή παίρνει σαν είσοδο ένα μήνυμα που ο κόμβος έλαβε από έναν γείτονα του και το οποίο αντιστοιχεί σε μία αίτηση. Η συνάρτηση αναλύει, αποκωδικοποιεί κατά κάποιον τρόπο, το μήνυμα αυτό και αρχικοποιεί τις αντίστοιχες μεταβλητές του ΠΑΑΔ. Τέλος υλοποιείται η συνάρτηση `rreq_pck_convert_to_text` η οποία μετατρέπει, κωδικοποιεί, την πληροφορία ενός ΠΑΑΔ σε ένα μήνυμα. Αυτό είναι το μήνυμα που αποστέλεται από έναν κόμβο σε κάποιους από τους γειτονικούς του, μέσω πάντα του προσομοιωτή. Ανάλογες συναρτήσεις υλοποιούνται και για τα υπόλοιπα πακέτα.

Μία απάντηση ανακάλυψης διαδρομής υλοποιείται ως εξής:

```
class route_reply_pck {
    public:
        int query_source;
        int query_dest;
        int query_id;
        int next_node_pos;
        list<int> reply_source_route;

        route_reply_pck();
        route_reply_pck(int, int, int, int, list<int>);
        route_reply_pck(char *);
        void rreply_pck_convert_to_text(char *);
};
```

Το `query_source` είναι το προσδιοριστικό του κόμβου που δημιούργησε και έστειλε την αντίστοιχη αίτηση και ο οποίος είναι ο τελικός προορισμός αυτής της απάντησης. Το `query_dest` είναι το προσδιοριστικό του κόμβου για τον οποίον αυτή η απάντηση περιέχει μία διαδρομή. Το `query_id` είναι ένας αριθμός ο οποίος δίνεται από τον `query_source` και ο οποίος είναι μοναδικός για όλες τις αιτήσεις από τον κόμβο αυτό. Η `reply_source_route` είναι η διαδρομή η οποία βρέθηκε με την διαδικασία της ανακάλυψης διαδρομής. Τέλος το `next_node_pos` χρησιμοποιείται σε σχέση με την `reply_source_route` για να δηλώσει τον κόμβο που θα πρέπει στην συνέχεια να λάβει αυτή την απάντηση.

Ένα Πακέτο Κατάστασης Ακμών (ΠΑΚΑ) υλοποιείται ως εξής:

```
class link_state_pck {  
  
    public:  
        int link_source;  
        int link_state_id;  
        int time_to_live;  
        list<int> source_neighbours;  
  
        link_state_pck();  
        link_state_pck(int, int, int, list<int>);  
        link_state_pck(char *);  
        void ls_pck_convert_to_text(char *);  
  
};
```

Το `link_source` είναι το προσδιοριστικό του κόμβου που δημιούργησε και έστειλε το συγκεκριμένο ΠΑΚΑ. Το `link_state_id` σε συνδυασμό με το `link_source` προσδιορίζουν μοναδικά το συγκεκριμένο ΠΑΚΑ. Το `time_to_live` δηλώνει τον αριθμό των ενδιάμεσων κόμβων που μπορούν να προωθήσουν το πακέτο αυτό μέχρι να απορριφθεί. Τέλος οι `source_neighbours` είναι οι γειτονικοί κόμβοι του `link_source`.

Ένα πακέτο δεδομένων υλοποιείται ως εξής:

```
class data_pck {  
  
    public:  
        int start_node;  
        int end_node;  
        int dpck_id;  
        int next_node_pos;  
        list<int> data_source_route;  
  
        data_pck();  
        data_pck(int, int, int, list<int>);  
        data_pck(char *);  
        void dpck_convert_to_text(char *);  
  
};
```

Το `start_node` είναι το προσδιοριστικό του κόμβου που δημιούργησε και έστειλε το πακέτο δεδομένων. Το `end_node` είναι το προσδιοριστικό του κόμβου στον οποίο πρέπει τελικά να φτάσει αυτό το πακέτο δεδομένων. Το

`prck_id` είναι ένας αριθμός ο οποίος δίνεται από τον `start_node` και ο οποίος είναι μοναδικός για όλα τα πακέτα δεδομένων που ξεκινούν από τον κόμβο αυτό. Η `data_source_route` είναι η διαδρομή που χρησιμοποιείται από αυτό το πακέτο δεδομένων για να φτάσει στον `end_node`. Το `next_node_pos` χρησιμοποιείται σε σχέση με την `data_source_route` για να δηλώσει τον κόμβο που θα πρέπει στην συνέχεια να λάβει αυτό το πακέτο δεδομένων.

## 5.5 Υλοποίηση του πρωτοκόλλου

Τα τρία πρωτόκολλα του ZRP έχουν υλοποιηθεί σαν ξεχωριστές τάξεις. Συγκεκριμένα η τάξη που υλοποιεί το IARP έχει την εξής δομή:

```
class iarp {
    private:
        int ls_id;
        iarp_routing_table iarp_rt;
        link_state_table ls_table;

    public:
        friend class brp;
        friend class ierp;
        friend class zrp_protocol;

        iarp();
        void link_state_pck_handle(char *, ierp &,
                                   simulator &);
        void neighbours_change_handle(ierp &,
                                       simulator &);
        void link_state_table_update(ierp &,
                                      simulator &);
};
```

Μέσα στην τάξη αυτή ορίζονται ο πίνακας δρομολόγησης του IARP και ο ΠΙΚΑ. Ακόμα ορίζονται οι συναρτήσεις οι οποίες είναι απαραίτητες για την λειτουργία του πρωτοκόλλου και οι οποίες χειρίζονται τα ΠΑΚΑ, εκτελούν τις απαραίτητες ενέργειες όταν οι γείτονες του κόμβου αλλάξουν και αφαιρούν από τον ΠΙΚΑ τις παλιές εγγραφές. Η όποια επικοινωνία με τα άλλα πρωτόκολλα γίνεται μέσω αυτών των συναρτήσεων. Τέλος ορίζονται και ορισμένες φίλες τάξεις, έτσι ώστε αυτές να μπορούν άμεσα να διαχειρίζονται τον πίνακα δρομολόγησης του IARP και τον ΠΙΚΑ.



Η τάξη που υλοποιεί το BRP έχει την εξής δομή:

```
class brp {  
  
    private:  
        int brp_id;  
        query_coverage_table query_cov_table;  
  
        list<int> routing_zone_nodes(GRAPH<int, int> &  
                                     int);  
        void compute_routing_zone(GRAPH<int, int> &  
                                   link_state_table);  
        list<int> compute_neighbours(query_coverage_entry  
                                     , int);  
        node rz_add_node(GRAPH<int, int> &, int);  
  
    public:  
        brp();  
        void bordercast_route_request(route_request_pck,  
                                       int, iarp &, simulator &);  
        void deliver_route_request(char *, ierp &, brp &  
                                    iarp &, simulator &);  
  
};
```

Μέσα στην τάξη αυτή ορίζεται ο ΠΚΑ. Ακόμα ορίζονται οι συναρτήσεις οι οποίες είναι απαραίτητες για την λειτουργία του πρωτοκόλλου και οι οποίες χειρίζονται τα ΠΑΑΔ που ο κόμβος λαμβάνει από το δίκτυο αλλά και τα ΕΠ που λαμβάνονται από το IERP. Η όποια επικοινωνία με τα άλλα πρωτόκολλα γίνεται μέσω αυτών των συναρτήσεων.

Η τάξη που υλοποιεί το IERP έχει την εξής δομή:

```

class ierp{
    private:
        int query_id;
        ierp_routing_table ierp_rt;

        list<int> shortest_route(list<list<int>>);
        int duplicate_nodes_exist(list<int> route);

    public:
        friend class iarp;
        friend class brp;
        friend class zrp_protocol;

        ierp();
        void routing_zone_changed(dictionary<int,
                                   list<int>>);
        void start_route_discovery(int, brp &, iarp &,
                                   simulator &);
        void ierp_route_request_handle(route_request_pck,
                                       int, brp &, iarp &, simulator &);
        int route_reply_handle(char *, simulator &);
};

```

Μέσα στην τάξη αυτή ορίζεται ο πίνακας δρομολόγησης του IERP. Ακόμα ορίζονται οι συναρτήσεις οι οποίες είναι απαραίτητες για την λειτουργία του πρωτοκόλλου και οι οποίες χειρίζονται τα ΕΠ που λαμβάνονται από το BRP, χειρίζονται τις απαντήσεις ανακάλυψης διαδρομής, εκκινούν μία νέα διαδικασία ανακάλυψης διαδρομής όταν χρειάζεται και συντηρούν τον πίνακα δρομολόγησης όταν η τοπολογία της ΖΔ του κόμβου αλλάζει. Η όποια επικοινωνία με τα άλλα πρωτόκολλα γίνεται μέσω αυτών των συναρτήσεων. Τέλος ορίζονται και ορισμένες φίλες τάξεις, έτσι ώστε αυτές να μπορούν άμεσα να διαχειρίζονται τον πίνακα δρομολόγησης του IERP.

Η τάξη που συνδυάζει όλα τα παραπάνω και υλοποιεί το ZRP πρωτόκολλο είναι η `zrp_protocol`. Σε κάθε κόμβο του δικτύου αντιστοιχεί ένα αντικείμενο αυτής της τάξης. Η `zrp_protocol` κληρονομεί από την αφηρημένη τάξη `ad-hoc_routing_protocol` η οποία δηλώνει ένα γενικό πρωτόκολλο δρομολόγησης για ασύρματα ad-hoc δίκτυα. Η τάξη `ad-hoc_routing_protocol` έχει την εξής δομή:

```

class adhoc_routing_protocol {
    protected:
        virtual void init() = 0;
    public:
        adhoc_routing_protocol();
        virtual void run(simulator &) = 0;
        virtual ~adhoc_routing_protocol();
};

```

Η τάξη zrp\_protocol έχει την εξής δομή:

```

class zrp_protocol : public adhoc_routing_protocol {
    private:
        int data_pck_id;
        send_buffer my_send_buffer;
        list<int> neighbours;
        ierp my_ierp;
        brp my_brp;
        iarp my_iarp;

        void link_state_pck_handle(simulator &, char *);
        void route_request_handle(simulator &, char *);
        void route_reply_handle(simulator &, char *);
        void data_packet_handle(simulator &, char *);
        void send_buffer_handle(simulator &);
        bool neighbours_changed(list<int>);
        list<int> update(list<int>, int);
        int duplicate_nodes_exist(list<int>);

    protected:
        void init();

    public:
        zrp_protocol();
        void run(simulator &);
};

```

Μέσα στην τάξη zrp\_protocol ορίζονται τα αντικείμενα των τριών πρωτοκόλλων του ZRP, η Μνήμη Προσωρινής Αποθήκευσης Πακέτων (ΜΠΑΠ) και η λίστα των γειτόνων του κόμβου. Ακόμα ορίζονται οι συναρτήσεις οι

οποίες αναλαμβάνουν την διαχείριση των διαφόρων πακέτων του πρωτοκόλλου που φτάνουν στον κόμβο. Εκτός από την περίπτωση των πακέτων δεδομένων οι συναρτήσεις αυτές με την σειρά τους καλούν τις συναρτήσεις των πρωτοκόλλων του ZRP που χειρίζονται τα αντίστοιχα πακέτα. Τα πακέτα δεδομένων τα χειρίζεται αποκλειστικά η τάξη `zrp_protocol`, ελέγχοντας την διαδρομή που αυτά περιέχουν και ανακατευθύνοντάς τα όταν χρειάζεται. Ακόμα ορίζεται η συνάρτηση `send_buffer_handle` η οποία διαχειρίζεται την αντίστοιχη δομή και η συνάρτηση `neighbours_changed` η οποία ελέγχει για τυχόν αλλαγές στους γείτονες του κόμβου.

Η κυριότερη συνάρτηση της τάξης `zrp_protocol` είναι η `run`, η οποία είναι δημόσια. Αυτή υλοποιεί έναν γύρο του πρωτοκόλλου, ο οποίος εκτελείται σε έναν κόμβο του δικτύου. Η συνάρτηση αυτή έχει την εξής δομή:

```
void run(simulator & sim){

    char msg[Msg_Size];
    int packet_type;
    int tmp_id;

    if(sim.receive(tmp_id, msg, false)){

        istrstream is(msg);
        is >> packet_type;

        switch (packet_type) {
            case Link_State_Packet:
                link_state_pck_handle(sim, msg);
                break;
            case Route_Request_Packet:
                route_request_handle(sim, msg);
                break;
            case Route_Reply_Packet:
                route_reply_handle(sim, msg);
                break;
            case Data_Packet:
                data_packet_handle(sim, msg);
                break;
            default:
                break;
        }

    }

    send_buffer_handle(sim);
    my_iarp.link_state_table_update(my_ierp, sim);
    if(neighbours_changed(sim.get_neighbours()))
        my_iarp.neighbours_change_handle(my_ierp, sim);
}
```

Ο κάθε κόμβος αρχικά διαβάζει το επόμενο μήνυμα το οποίο απευθύνεται σε αυτόν και το χειρίζεται ανάλογα με το είδος του πακέτου στο οποίο το μήνυμα αυτό αντιστοιχεί. Στη συνέχεια γίνεται ο χειρισμός του `send_buffer` του κόμβου, διαγράφονται από τον ΠΙΚΑ πιθανές παλιές εγγραφές και σε περίπτωση αλλαγής των γειτόνων του κόμβου ειδοποιείται το IARP πρωτόκολλο να προβεί στις κατάλληλες ενέργειες. Στην πραγματική εκτέλεση του πρωτοκόλλου σε κάθε κόμβο η παραπάνω διαδικασία επαναλαμβάνεται συνεχώς.

## 5.6 Εκτέλεση του πρωτοκόλλου

Για την εκτέλεση του πρωτοκόλλου απαιτείται ένα μικρό πρόγραμμα το οποίο θα τρέχει σε κάθε κόμβο του δικτύου, όπως το παρακάτω:

```
int main(int argc, char **argv){  
  
    simulator sim;  
    zrp_protocol zrp;  
  
    while(1){  
  
        zrp.run(sim);  
    }  
  
    return 0;  
  
}
```

Το πρόγραμμα αυτό αρχικά ορίζει ένα αντικείμενο της τάξης του προσομοιωτή και ένα αντικείμενο της τάξης που υλοποιεί το ZRP πρωτόκολλο. Στη συνέχεια καλείται μέσα σε έναν άπειρο βρόχο η συνάρτηση `run`, η οποία υλοποιεί έναν γύρο του πρωτοκόλλου.

Η δημιουργία των πακετών δεδομένων, τα οποία ουσιαστικά εκκινούν τις διαδικασίες του πρωτοκόλλου, μπορεί να γίνεται στο παραπάνω πρόγραμμα. Δηλαδή σε τυχαίες χρονικές στιγμές θα δημιουργούνται νέα πακέτα δεδομένων με τυχαίο κόμβο - προορισμού. Τα πακέτα αυτά θα πρέπει να έχουν σαν `drck_id` την τιμή `-1`, έτσι ώστε να διακρίνονται από τα πακέτα δεδομένων για τα οποία ήδη έχει βρεθεί μία διαδρομή, με την βοήθεια του πρωτοκόλλου και τα οποία προωθούνται μέχρι τον κόμβο - προορισμού τους.

## 5.7 Ειδικές περιπτώσεις και παραδοχές της υλοποίησης

Παραθέτουμε διάφορες ειδικές περιπτώσεις και παραδοχές της υλοποίησης του ZRP πρωτοκόλλου:

- Θεωρούμε ότι σε κάθε περίπτωση είναι δυνατή η αμφίδρομη επικοινωνία μεταξύ γειτονικών κόμβων. Αυτό έχει σαν αποτέλεσμα ο υπολογισμός της διαδρομής επιστροφής μίας απάντησης αίτησης διαδρομής να γίνεται αντιστρέφοντας τη διαδρομή που περιέχεται στο ΠΑΑΔ.

- Η υλοποίηση χρησιμοποιεί source routing.
- Για την αποφυγή δημιουργίας βρόχων κατά την διαδικασία ανακάλυψης διαδρομής ένας κόμβος που λαμβάνει μία αίτηση ελέγχει τον ΠΚΑ για να διαπιστώσει αν έχει λάβει ξανά την ίδια αίτηση. Στην περίπτωση αυτή η αίτηση απορρίπτεται.
- Πριν σταλλεί μία απάντηση αίτησης διαδρομής ελέγχεται αν η διαδρομή που αυτό περιέχει, έχει κάποιον βρόχο.
- Στην υλοποίηση προσθέσαμε λειτουργίες του επιπέδου δικτύου, οι οποίες βέβαια δεν είναι μέρος του ZRP πρωτοκόλλου. Τέτοιες λειτουργίες είναι η δρομολόγηση των πακέτων δεδομένων αλλά και η αποθήκευση τους στη ΜΠΑΠ ενός κόμβου, στην περίπτωση που δεν είναι γνωστή μία διαδρομή προς τον προορισμό τους.
- Κατά την δρομολόγηση ενός πακέτου δεδομένων οι ενδιάμεσοι κόμβοι δεν επιβεβαιώνουν τη λήψη του.
- Κάθε φορά πριν προωθηθεί ένα πακέτο δεδομένων η διαδρομή που αυτό περιέχει συντηρείται χρησιμοποιώντας την γνώση της τοπολογίας της ΖΔ του κόμβου. Σε περίπτωση που η διαδρομή που περιέχεται στο πακέτο δεδομένων δεν μπορεί να αντικατασταθεί με μία έγκυρη διαδρομή τότε το πακέτο δεδομένων απορρίπτεται. Σε καμία περίπτωση ωστόσο δεν ενημερώνεται ο κόμβος πηγή.





# Κεφάλαιο 6

## Πειραματική αξιολόγηση

### 6.1 Γενικά

Πραγματοποιήσαμε έναν μεγάλο αριθμό πειραμάτων, στόχος αυτών ήταν η επιβεβαίωση της ορθότητας των υλοποιήσεων των πρωτοκόλλων DSR και ZRP και η αξιολόγηση της απόδοσης και της αποτελεσματικότητας τους. Θέλοντας όμως ακόμα να συγκρίνουμε τα πρωτόκολλα αυτά, που στηρίζονται στην εύρεση διαδρομών από σταθμούς (κόμβους), με τα πρωτόκολλα που στηρίζονται στην κίνηση των σταθμών (κόμβων) πραγματοποιήσαμε πειράματα και με το πρωτόκολλο SRP, χρησιμοποιώντας σαν πρωτόκολλο διαχείρισης της υποστήριξης τους δρομείς.

### 6.2 Περιβάλλον προσομοίωσης

Τα πειράματα πραγματοποιήθηκαν με την βοήθεια ενός προσομοιωτή [14]. Ο προσομοιωτής αυτός έχει υλοποιηθεί με την γλώσσα προγραμματισμού C++ και με την χρήση της βιβλιοθήκης LEDA. Τα πρωτόκολλα που μπορούν να προσομοιωθούν σε αυτόν πρέπει να έχουν γραφτεί με την γλώσσα προγραμματισμού C++, όπως δηλαδή συμβαίνει και στην περίπτωση μας.

Ο προσομοιωτής αυτός ειδικεύεται στα ad-hoc δίκτυα, αποφεύγοντας να ενσωματώσει πολλούς μηχανισμούς για την απεικόνιση των λειτουργιών των χαμηλών επιπέδων των δικτύων. Με τον τρόπο αυτό ο προσομοιωτής είναι μικρός, αποδοτικός ως προς την μνήμη και την χρήση του επεξεργαστή και δίνει την δυνατότητα για γρήγορη προσομοίωση μεγάλων δικτύων. Ακόμα είναι εύκολος στην διαχείριση, παρέχοντας τις απαραίτητες λειτουργίες στο επίπεδο διεπαφής και δίνει την δυνατότητα για την αυτοματοποίηση της εκτέλεσης επαναλήψεων των ίδιων προσομοιώσεων. Παράλληλα κατά την διάρκεια αλλά και στο τέλος μίας προσομοίωσης, ο προσομοιωτής, επιστρέφει στατιστικά

και συγκεντρωτικά αποτελέσματα.

Ο προσομοιωτής χωρίζει την λειτουργία του σε γύρους προσομοίωσης. Οι γύροι αυτοί παίζουν και τον ρόλο του χρόνου της προσομοίωσης. Σε κάθε γύρο εξετάζονται με τυχαία σειρά όλοι οι κινητοί σταθμοί και εκτελούνται για κάθε έναν μία ή περισσότερες από τις ακόλουθες διαδικασίες:

- Μετακίνηση σε μία γειτονική κορυφή του γραφήματος κίνησης
- Δημιουργία νέου πακέτου δεδομένων
- Επεξεργασία των πακέτων που λήφθηκαν κατά τον προηγούμενο γύρο
- Μετάδοση ενός ή περισσότερων πακέτων
- Εκτέλεση κάποιου υπολογισμού

Τα πακέτα που ένας σταθμός στέλνει σε έναν γύρο παραδίδονται στους γειτονικούς σταθμούς στον επόμενο γύρο. Ακόμα οι διαδικασίες που εκτελεί σε κάθε γύρο ένας κινητός σταθμός θα πρέπει να μην περιέχουν κάποιο μηχανισμό εμπλοκής, γιατί σε μία τέτοια περίπτωση η εκτέλεση όλης της προσομοίωσης θα σταματήσει.

Με την χρήση του αντικειμενοστρεφούς προγραμματισμού ο προσομοιωτής υλοποιήθηκε σαν ένα σύνολο από κλάσεις με καθορισμένες μεθόδους. Ορισμένες από αυτές είναι υπεύθυνες για την καθαυτή λειτουργία του προσομοιωτή, δηλαδή για την μοντελοποίηση της περιοχής που καλύπτει το δίκτυο (γράφημα κίνησης), για τον έλεγχο της κίνησης, της θέσης και της επικοινωνίας των κινητών σταθμών, για την εκτέλεση των γύρων προσομοίωσης αλλά και για την διατήρηση των στατιστικών και των συγκεντρωτικών αποτελεσμάτων της προσομοίωσης. Άλλες κλάσεις απεικονίζουν οντότητες του δικτύου, τους κινητούς σταθμούς και τα μηνύματα που αυτοί ανταλλάσσουν. Αυτές οι κλάσεις είναι σχεδιασμένες έτσι ώστε να μπορούν να επεκταθούν εύκολα (με κληρονομικότητα), για να υλοποιηθεί ένα συγκεκριμένο πρωτόκολλο. Η κλάση που απεικονίζει ένα κινητό σταθμό, η `mh`, έχει όλες τις απαραίτητες συναρτήσεις για την εκτέλεση των βασικών λειτουργιών ασύρματης επικοινωνίας (μετάδοση και λήψη πακέτων) αλλά και άλλες για την ανίχνευση των γειτονικών σταθμών, την επιστροφή του παρόντος χρόνου - γύρου προσομοίωσης κ.α. Ακόμα η κλάση αυτή έχει τις απαραίτητες συναρτήσεις για την κίνηση του αντίστοιχου σταθμού. Τέλος προσφέρει μία εικονική συνάρτηση, την `executeProtocol`, η οποία αντιστοιχεί στην εκτέλεση του πρωτοκόλλου στον αντίστοιχο κινητό σταθμό σε έναν γύρο. Αυτή τη συνάρτηση καλεί ο προσομοιωτής για κάθε κινητό σταθμό σε ένα γύρο.

Ο προσομοιωτής καταρχήν δέχεται σαν παράμετρο το είδος του γραφήματος κίνησης που θα χρησιμοποιηθεί καθώς και το πλήθος των κορυφών αυτού. Υπάρχει η δυνατότητα για την δημιουργία διαφόρων γραφημάτων κίνησης, τα οποία αναπαριστούν την περιοχή που καλύπτει ένα δίκτυο και στην οποία κινούνται οι κινητοί σταθμοί του. Ακόμα ο προσομοιωτής δέχεται σαν παράμετρο τον αριθμό των κινητών σταθμών του αντίστοιχου δικτύου. Κάθε κινητός σταθμός θεωρείται ότι βρίσκεται σε μία κορυφή του γραφήματος κίνησης και κινείται πάνω στις ακμές αυτού από κορυφή σε κορυφή. Ο ρυθμός αυτής της κίνησης δίνεται πάλι σαν παράμετρος στον προσομοιωτή (μία τιμή μεταξύ 0 και 1). Με βάση αυτή την παράμετρο, το αν ένας σταθμός θα κινηθεί σε κάποιον γύρο υπολογίζεται ως εξής:

```
if (rand_int(0,1000000) < 1000000*_motion_rate)
    MOVERANDOMLY();
```

Ακόμα κάθε κινητός σταθμός ανταλλάσσει πακέτα (ελέγχου και δεδομένων) με κινητούς σταθμούς που βρίσκονται σε γειτονικές κορυφές. Ο αριθμός των πακέτων δεδομένων που ένας κινητός σταθμός θα δημιουργήσει για να τα μεταδώσει σε άλλους τυχαίους κινητούς σταθμούς καθώς και ο ρυθμός δημιουργίας αυτών (μία τιμή μεταξύ 0 και 1) δίνονται σαν παράμετροι στον προσομοιωτή. Με βάση αυτή την παράμετρο, το αν ένας σταθμός θα δημιουργήσει ένα νέο πακέτο δεδομένων σε κάποιον γύρο υπολογίζεται ως εξής:

```
if (rand_int(0,1000000) <= 1000000*_messages_generate_rate)
    genNewMessages();
```

### 6.3 Διασύνδεση των υλοποιήσεων των πρωτοκόλλων με το περιβάλλον προσομοίωσης

Οι υλοποιήσεις των πρωτοκόλλων DSR και ZRP δεν βασίζονται σε έναν συγκεκριμένο προσομοιωτή. Αυτή η ευελιξία επιτυγχάνεται μέσω του γενικού μοντέλου προσομοιωτή που χρησιμοποιούν και τα δύο πρωτόκολλα στις υλοποιήσεις τους, simulator και simul αντίστοιχα. Μέσω αυτού και οι δύο υλοποιήσεις καλούν τις συναρτήσεις που χρειάζονται για την λειτουργία τους (π.χ. για την λήψη, την αποστολή πακέτων κ.α.) χωρίς να ενδιαφέρονται πως αυτές υλοποιούνται από το γενικό μοντέλο. Οι υλοποιήσεις αυτές μπορούν με λίγες

προσθήκες και χωρίς να αλλαχτεί ο κώδικας τους, να χρησιμοποιήσουν τον προσομοιωτή που περιγράφηκε παραπάνω.

Για τον σκοπό αυτό υλοποιήσαμε τις συναρτήσεις του γενικού μοντέλου προσομοιωτή, των υλοποιήσεων, χρησιμοποιώντας τις αντίστοιχες συναρτήσεις του προσομοιωτή που περιγράφηκε. Επιπλέον ορίσαμε για κάθε μία από τις υλοποιήσεις μία καινούρια κλάση, την `dSrAgent` και την `zrpAgent`. Αυτές οι δύο κλάσεις απεικονίζουν έναν κινητό σταθμό που εκτελεί είτε το DSR είτε το ZRP πρωτόκολλο αντίστοιχα. Ο προσομοιωτής κατά την διάρκεια αρχικοποίησης της προσομοίωσης, ενός από τα πρωτόκολλα, δημιουργεί τόσα αντικείμενα της αντίστοιχης κλάσης όσοι και οι κινητοί σταθμοί του δικτύου. Αυτά τα αντικείμενα τα χρησιμοποιεί κατά την διάρκεια των γύρων προσομοίωσης. Οι κλάσεις `dSrAgent` και `zrpAgent` κληρονομούν τη `mh` και υλοποιούν τη συνάρτηση `executeProtocol` αυτής. Στην υλοποίηση της `executeProtocol` καλείται η συνάρτηση `run`, της `dSr_protocol` ή της `zrp_protocol` αντίστοιχα. Μέσω της `run` ο αντίστοιχος σταθμός εκτελεί διάφορες ενέργειες με βάση το πρωτόκολλο για ένα γύρο της προσομοίωσης. Ακόμα στην `executeProtocol` καλούνται και οι συναρτήσεις της `mh` για τη δημιουργία ενός νέου πακέτου δεδομένων και τη κίνηση του κινητού σταθμού, με βάση πάντα τις παραμέτρους που έχουν δωθεί στον προσομοιωτή.

## 6.4 Παράμετροι και μετρούμενα μεγέθη πειραμάτων

Στα πειράματα που πραγματοποιήσαμε χρησιμοποιήσαμε δύο ειδών γραφημάτων κίνησης, τα γραφήματα δισδιάστατου πλέγματος και τα γραφήματα τρισδιάστατου πλέγματος. Στο πρώτο είδος γραφημάτων κίνησης οι σταθμοί κινούνται σε μία επίπεδη επιφάνεια, ενώ στο δεύτερο κινούνται στον τρισδιάστατο χώρο. Για τα γραφήματα δισδιάστατου και τρισδιάστατου πλέγματος πειραματιστήκαμε με τρία πλήθη κορυφών: 121, 529, 1024 και 125, 512, 1000 αντίστοιχα. Ακόμα σε κάθε εκτέλεση ενός πειράματος το συνολικό πλήθος των πακέτων δεδομένων που οι κινητοί σταθμοί δημιουργούσαν ήταν περίπου 5000. Σε κάθε πείραμα που εκτελούσαμε αλλάζαμε μία από τις εξής τρεις παραμέτρους: το πλήθος των κινητών σταθμών (προκαθορισμένος αριθμός 50, μέγιστος 100), τον ρυθμό δημιουργίας νέων πακέτων δεδομένων προς αποστολή (προκαθορισμένος αριθμός 0.02, μέγιστος 1 - δημιουργία πακέτου δεδομένων από κάθε σταθμό σε κάθε γύρο) και τον ρυθμό κίνησης κάθε σταθμού (προκαθορισμένος αριθμός 0.01, μέγιστος 1 - κίνηση κάθε σταθμού σε κάθε γύρο).

Βασικό μέτρο αξιολόγησης των πρωτοκόλλων και των υλοποιήσεων τους, για κάθε είδος πειράματος, ήταν το ποσοστό των πακέτων δεδομένων που

παρελήφθησαν από τους αντίστοιχους προορισμούς σε σχέση με τα πακέτα δεδομένων που δημιουργήθηκαν συνολικά στο δίκτυο. Άλλα μέτρα αξιολόγησης ήταν η μέση καθυστέρηση λήψης (σε γύρους) ενός πακέτου δεδομένων από τον προορισμό του από τη στιγμή που το πακέτο αυτό δημιουργούνταν, καθώς και ο μέσος αριθμός βημάτων που ένα πακέτο δεδομένων διάνυε για να φτάσει στον προορισμό του. Τέλος η χρήση της μνήμης και ο αριθμός των πακέτων ελέγχου που στάλθηκαν ήταν κάποια άλλα μέτρα αξιολόγησης των πρωτοκόλλων και των υλοποιήσεων τους. Η χρήση της μνήμης για κάθε πείραμα ήταν το άθροισμα των μέγιστων αριθμών εγγραφών των διαφόρων δομών, που κάθε σταθμός χρησιμοποίησε σε κάποιο γύρο.

## 6.5 Dynamic Source Routing (DSR)

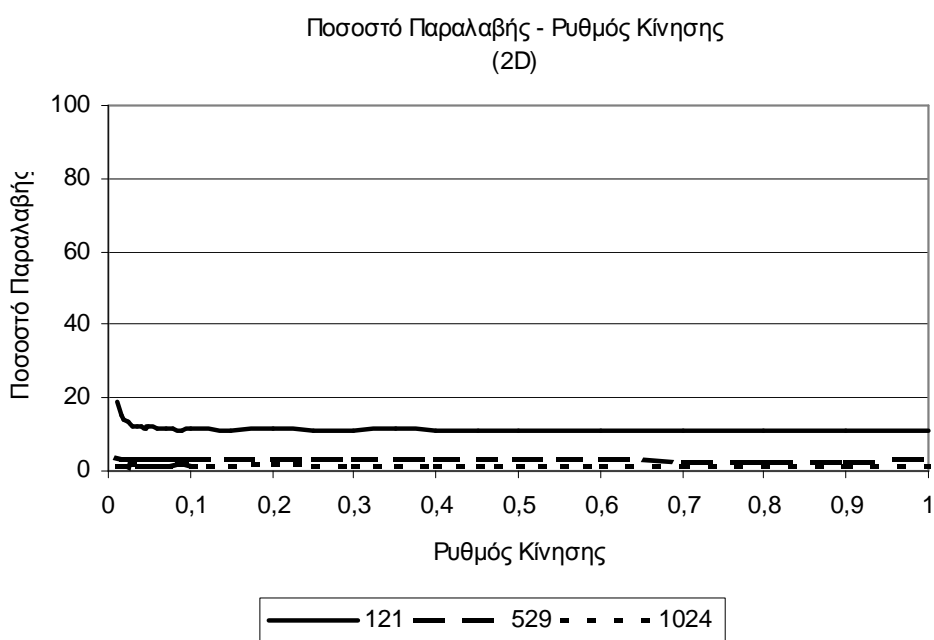
### 6.5.1 Γενικά

Για το DSR ο μέγιστος αριθμός των επαναλήψεων της αποστολής μία ΑΙΔ για έναν προορισμό ήταν δύο. Ακόμα κάθε είδος πειράματος, δηλαδή με συγκεκριμένες παραμέτρους, επαναληφθεί τόσες φορές ώστε ο συνολικός αριθμός πακέτων δεδομένων που δημιουργήθηκαν, για το συγκεκριμένο είδος πειράματος, να είναι περίπου 100000.

Γενικά από τα πειράματα που διεξάγαμε παρατηρήσαμε ότι η συμπεριφορά του πρωτοκόλλου (ποσοστό παράδοσης, μέσος αριθμός βημάτων, μέση καθυστέρηση λήψης), για τους διάφορους ρυθμούς κίνησης και δημιουργίας πακέτων δεδομένων αλλά και για τα διάφορα πλήθη κινητών σταθμών, είναι άμεσα εξαρτώμενη από την πυκνότητα των αντίστοιχων δικτύων. Έτσι για τα γραφήματα 121 (ή 125) κορυφών η συμπεριφορά του πρωτοκόλλου είναι γενικά καλύτερη σε σχέση με τα άλλα μεγέθη γραφημάτων. Ακόμα τα αποτελέσματα των πειραμάτων μας για τα δισδιάστατα και τα τρισδιάστατα γραφήματα είναι παρόμοια, οι όποιες διαφορές οφείλονται στην διαφορετική πυκνότητα των αντίστοιχων δικτύων. Όσον αφορά τα διάφορα μεταβαλλόμενα μεγέθη, καταρχήν παρατηρήσαμε ότι η αύξηση της κινητικότητας των σταθμών έχει σαν αποτέλεσμα την δημιουργία διαδρομών μικρότερου μήκους και την ανταλλαγή πακέτων δεδομένων μόνο μεταξύ γειτονικών σταθμών. Αυτό βέβαια οδηγεί και στην μείωση του ποσοστού των πακέτων δεδομένων που παρελήφθησαν. Ακόμα πρέπει να παρατηρήσουμε ότι από ένα σημείο και μετά η περαιτέρω αύξηση της κινητικότητας των σταθμών δεν επηρεάζει ιδιαίτερα την συμπεριφορά του πρωτοκόλλου. Η αύξηση του πλήθους των κινητών σταθμών αυξάνει την πυκνότητα των δικτύων, όπως παράλληλα και το πλήθος των δυνατών προορισμών των πακέτων δεδομένων που δημιουργούνται, με τις ανάλογες βέβαια συνέπειες. Η αύξηση του ρυθμού δημιουργίας νέων πακέτων δεδομένων ευνοεί ιδιαίτερα το

πρωτόκολλο. Έτσι για γραφήματα 121 (ή 125) κορυφών το ποσοστό παράδοσης και ο μέσος αριθμός βημάτων αυξάνονται, φτάνοντας στις μεγαλύτερες τιμές τους σε σχέση με τα άλλα πειράματα. Τέλος από τα πειράματα που διεξάγαμε επαληθεύσαμε την ορθότητα της υλοποίησης του πρωτοκόλλου. Η υλοποίηση επηρεάζει ως ένα βαθμό την μέση καθυστέρηση λήψης (λόγω ΜΠΑΠ), την χρήση της μνήμης και τον αριθμό των πακέτων ελέγχου που στάλθηκαν στο δίκτυο.

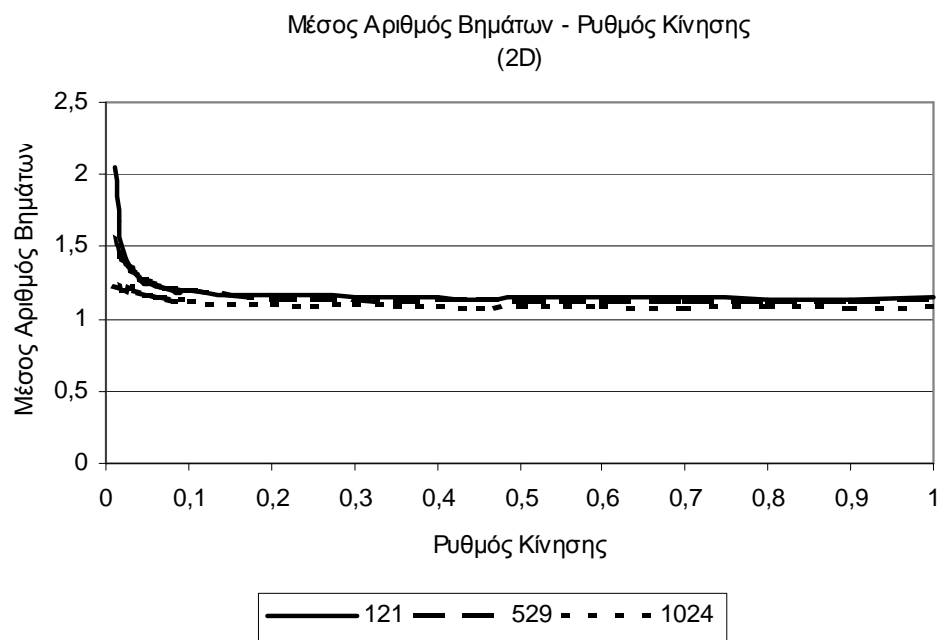
### 6.5.2 Μεταβολή ρυθμού κίνησης



Σχήμα 6.1: Ποσοστό Παραλαβής - Ρυθμός Κίνησης, για δισδιάστατα γραφήματα (DSR)

Στο σχήμα 6.1 παρατηρούμε ότι το ποσοστό των πακέτων δεδομένων που παρελήφθησαν για γράφημα 121 κορυφών μειώνεται απότομα στην αρχή και στην συνέχεια σταθεροποιείται. Η μείωση του ποσοστού οφείλεται στο ότι η αυξανόμενη κίνηση των σταθμών οδηγεί στην γρήγορη διάσπαση των διαδρομών που ανακαλύπτονται, με αποτέλεσμα να μην μπορούν να μεταδωθούν πακέτα δεδομένων μέσω αυτών. Ακόμα πολλές φορές μία ΑΠΔ δεν μπορεί να επιστρέψει στο σταθμό που δημιούργησε την αντίστοιχη ΑΙΔ μιας και η αντεστραμμένη διαδρομή επιστροφής δεν ισχύει πια. Η σταθεροποίηση του ποσοστού οφείλεται στο ότι από ένα σημείο και μετά τα πακέτα δεδομένων που παραδίδονται είναι κυρίως αυτά μεταξύ γειτονικών σταθμών, κάτι που φαίνεται και από

το σχήμα 6.2. Ανεξάρτητα της αύξησης της κίνησης πάντα κάποιοι σταθμοί είναι γειτονικοί με κάποιους άλλους, παρόλο που τα ζευγάρια αυτά μπορεί να αλλάζουν με αυξανόμενο ρυθμό. Ακόμα το ποσοστό των πακέτων δεδομένων που παρελήφθησαν για γράφημα 121 κορυφών είναι υψηλότερο από τα ποσοστά των άλλων γραφημάτων, για κάθε ρυθμό κίνησης. Αυτό οφείλεται στο ότι τα δίκτυα που αντιστοιχούν στο πρώτο είδος γραφήματος είναι σαφώς πιο πυκνά (50 σταθμοί - 121 κορυφές), με αποτέλεσμα να υπάρχει μεγαλύτερη πιθανότητα κάποιοι προορισμοί να είναι προσβάσιμοι ή να γίνουν προσβάσιμοι σύντομα λόγω της κίνησης των σταθμών. Το γεγονός αυτό εξηγεί και το γιατί το ποσοστό των πακέτων δεδομένων που παρελήφθησαν για τα δίκτυα που αντιστοιχούν σε γραφήματα 529 και 1024 κορυφών, παραμένει σχεδόν σταθερό για όλους τους ρυθμούς κίνησης. Τα δίκτυα αυτά είναι αραιά (50 σταθμοί - 529 και 1024 κορυφές) με αποτέλεσμα τα πακέτα που παραδίδονται να είναι κυρίως αυτά μεταξύ γειτονικών ή πολύ κοντινών σταθμών ανεξάρτητα του ρυθμού κίνησης, κάτι που φαίνεται και από το σχήμα 6.2.

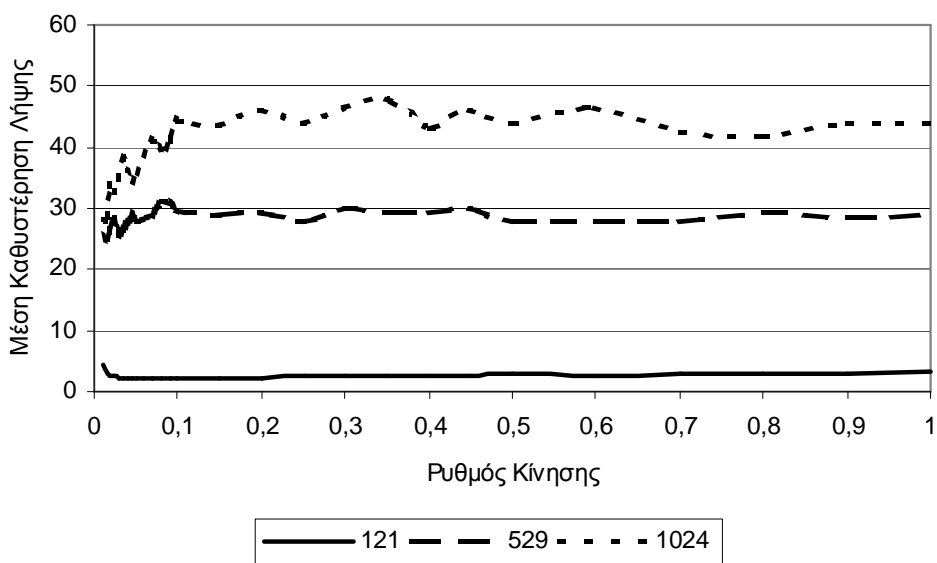


Σχήμα 6.2: Μέσος Αριθμός Βημάτων - Ρυθμός Κίνησης, για διδιάστατα γραφήματα (DSR)

Στο σχήμα 6.2 παρατηρούμε ότι και για τα τρία μεγέθη γραφημάτων ο μέσος αριθμός βημάτων μειώνεται απότομα στην αρχή και στη συνέχεια σταθεροποιείται κοντά στο 1, στην μικρότερη δηλαδή δυνατή τιμή. Αυτό οφείλεται στο ότι καθώς ο ρυθμός της κίνησης μεγαλώνει τα μόνο πακέτα που παραδίδονται είναι

αυτά μεταξύ γειτονικών σταθμών. Στην περίπτωση βέβαια των δικτύων που αντιστοιχούν στα γραφήματα 529 και 1024 κορυφών, όπως αναφέραμε, αυτά είναι πολύ αραιά με αποτέλεσμα πάλι πακέτα να παραδίδονται κυρίως μεταξύ γειτονικών ή πολύ κοντινών σταθμών, ανεξάρτητα του ρυθμού κίνησης. Για αυτό άλλωστε στην αρχή ο μέσος αριθμός βημάτων για τα δίκτυα αυτά είναι μικρότερος από τον μέσο αριθμό βημάτων για τα δίκτυα που αντιστοιχούν στα γραφήματα 121 κορυφών. Πρέπει να παρατηρήσουμε ωστόσο ότι ο μέσος αριθμός βημάτων στην περίπτωση των δικτύων που αντιστοιχούν στα γραφήματα 121 κορυφών είναι πολύ μικρός ακόμα και για σχετικά μικρούς ρυθμούς κίνησης, παρόλο που τα δίκτυα αυτά είναι πιο πυκνά. Αυτό πιθανός να οφείλεται στο ότι στην πλειοψηφία των περιπτώσεων πακέτα μεταδίδονται επιτυχώς μεταξύ γειτονικών ή πολύ κοντινών σταθμών. Φανερώνεται έτσι η γενικότερη αδυναμία των πρωτοκόλλων που στηρίζονται στην δημιουργία διαδρομών για την δημιουργία διαδρομών μεγάλου μήκους.

Μέση Καθυστέρηση Λήψης - Ρυθμός Κίνησης  
(2D)



Σχήμα 6.3: Μέση Καθυστέρηση Λήψης - Ρυθμός Κίνησης, για δισδιάστατα γραφήματα (DSR)

Η μέση καθυστέρηση λήψης δεν εξαρτάται μόνο από τον μέσο αριθμό βημάτων που το πακέτο διανύει, αλλά και από το χρόνο που αυτό παραμένει στην ΜΠΑΠ μέχρι να βρεθεί μία διαδρομή προς τον προορισμό. Παρατηρώντας τα σχήματα 6.2 και 6.3 συμπεραίνουμε ότι ο χρόνος παραμονής στην ΜΠΑΠ επηρεάζει περισσότερο την μέση καθυστέρηση λήψης, μιας και ο μέσος αριθμός βημάτων είναι μικρότερος από τον μέσο αριθμό βημάτων για τα δίκτυα που αντιστοιχούν στα γραφήματα 121 κορυφών.



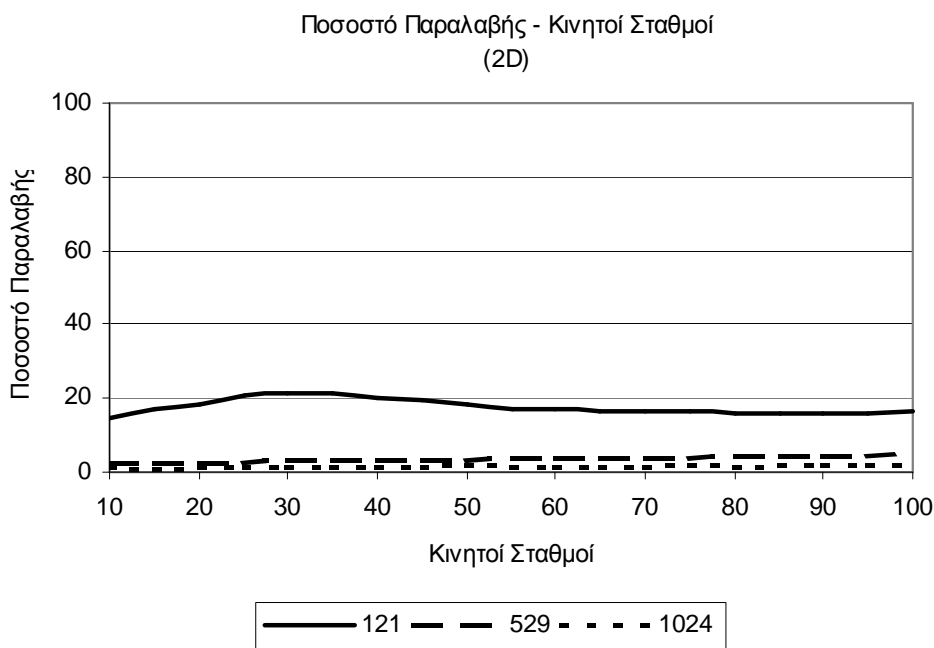
θμός βημάτων είναι πολύ μικρός για όλα τα γραφήματα. Για το γράφημα 121 κορυφών η μέση καθυστέρηση λήψης είναι πολύ μικρή (μικρότερη των 5 γύρων), μικρότερη σε σχέση με τα άλλα γραφήματα και σχεδόν σταθερή για τους διάφορους ρυθμούς κίνησης. Λαμβάνοντας υπόψη το σχήμα 6.2, όπου ο μέσος αριθμός βημάτων είναι μικρότερος του 2, οδηγούμαστε στο συμπέρασμα ότι τα περισσότερα πακέτα που εντέλει παραδίδονται στον προορισμό τους περιμένουν για λίγο χρόνο στην ΜΠΑΠ του σταθμού που τα δημιουργεί. Αυτό οφείλεται στο ότι τα αντίστοιχα δίκτυα είναι σχετικά πυκνά με αποτέλεσμα να υπάρχει μεγάλη πιθανότητα κάποιος προορισμός να είναι άμεσα προσβάσιμος ή να γίνουν προσβάσιμοι σύντομα λόγω της κίνησης των σταθμών. Η πιθανότητα αυτή φαίνεται να είναι σχεδόν σταθερή ανεξάρτητα της περαιτέρω αύξησης του ρυθμού κίνησης, με αποτέλεσμα ο χρόνος παραμονής στην ΜΠΑΠ να παραμένει περίπου ο ίδιος όπως και η μέση καθυστέρηση λήψης. Για τα γραφήματα 529 και 1024 κορυφών η μέση καθυστέρηση λήψης για ρυθμούς κίνησης μικρότερους από 0.1 αυξάνεται, ενώ για μεγαλύτερους ρυθμούς κίνησης είναι σχεδόν σταθερή. Για τα γραφήματα αυτά ο χρόνος παραμονής στην ΜΠΑΠ είναι προφανώς μεγαλύτερος. Αυτό οφείλεται στο ότι τα αντίστοιχα δίκτυα είναι αραιά με αποτέλεσμα ακόμα και όταν οι σταθμοί κινούνται να υπάρχει μικρή πιθανότητα γρήγορα κάποιος πιθανός προορισμός να γίνει προσβάσιμος ή να είναι ήδη προσβάσιμος. Όπως και προηγουμένως η πιθανότητα αυτή φαίνεται να είναι σχεδόν σταθερή, με εξαίρεση την αρχή, ανεξάρτητα της περαιτέρω αύξησης του ρυθμού κίνησης, με αποτέλεσμα ο χρόνος παραμονής στην ΜΠΑΠ να παραμένει περίπου ο ίδιος. Η αρχική αύξηση της μέσης καθυστέρησης λήψης οφείλεται στο ότι η αυξανόμενη κίνηση των σταθμών προκαλεί μεγαλύτερους χρόνους παραμονής στην ΜΠΑΠ.

Παράλληλα από τα πειράματα που εκτελέσαμε παρατηρήσαμε ότι το πλήθος των πακέτων ελέγχου που στάλθηκαν στο δίκτυο μειώνεται καθώς αυξάνεται ο ρυθμός κίνησης (βλ. παράρτημα). Η αυξανόμενη κίνηση των σταθμών έχει σαν αποτέλεσμα τα πακέτα ΑΙΔ να διασπείρονται πιο γρήγορα στο δίκτυο και με λιγότερες αποστολές πακέτων. Ακόμα με την αύξηση του ρυθμού κίνησης όλο και λιγότερα πακέτα ΑΠΔ επιστρέφουν στον σταθμό πηγή, μιας και η αντεστραμμένη διαδρομή επιστροφής δεν ισχύει πια. Επιπλέον επειδή όλο και λιγότερες διαδρομές δημιουργούνται και λιγότερα πακέτα δεδομένων αποστέλλονται αυτό έχει σαν αποτέλεσμα το πλήθος των επιβεβαιώσεων να είναι μικρότερο.

Τα πειράματα που εκτελέσαμε για τρισδιάστατα γραφήματα μεταβάλλοντας τον ρυθμό κίνησης έδωσαν παρόμοια αποτελέσματα με τα αντίστοιχα πειράματα για δισδιάστατα γραφήματα (βλ. παράρτημα). Ωστόσο το γεγονός ότι τα δίκτυα που αντιστοιχούν στα τρισδιάστατα γραφήματα είναι σχετικά πιο πυκνά είχε σαν αποτέλεσμα τα ποσοστά των πακέτων δεδομένων που παρελήφθησαν να είναι μεγαλύτερα και οι μέσες καθυστερήσεις λήψης να είναι μικρότερες και για τα

τρία μεγέθη γραφημάτων (125, 512, 1000).

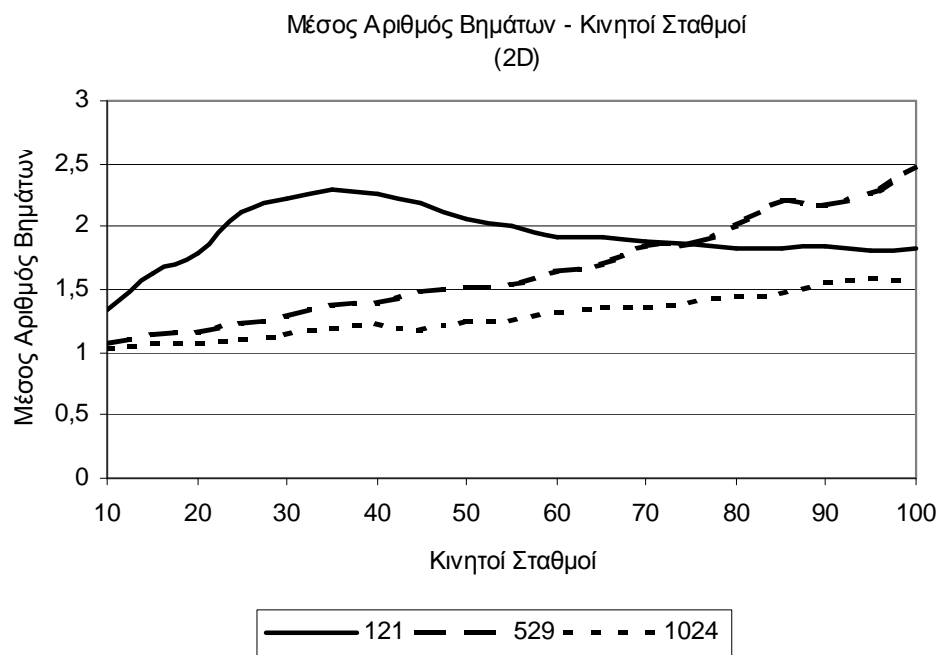
### 6.5.3 Μεταβολή πλήθους κινητών σταθμών



Σχήμα 6.4: Ποσοστό Παραλαβής - Κινητοί Σταθμοί, για δισδιάστατα γραφήματα (DSR)

Στο σχήμα 6.4 παρατηρούμε ότι το ποσοστό των πακέτων δεδομένων που παρελήφθησαν για γράφημα 121 κορυφών αυξάνεται στην αρχή, στη συνέχεια μειώνεται και σταθεροποιείται σε μία τιμή. Η αρχική αύξηση που παρουσιάζει το ποσοστό αυτό οφείλεται στην αύξηση της πυκνότητας του αντίστοιχου δικτύου, που προκαλείται από την ύπαρξη περισσότερων κινητών σταθμών. Παρόλα αυτά η μείωση του ποσοστού που ακολουθεί οφείλεται στο γεγονός ότι η αύξηση των κινητών σταθμών οδηγεί και στην αύξηση των πιθανών προορισμών των πακέτων δεδομένων που δημιουργούνται. Στην αρχή δηλαδή το δίκτυο μπορεί να είναι πιο αραιό από ότι στη συνέχεια αλλά οι πιθανοί προορισμοί των πακέτων δεδομένων που δημιουργούνται είναι πολλοί λίγοι, αυξάνοντας έτσι την πιθανότητα παράδοσης ενός πακέτου δεδομένων. Ακόμα το γεγονός ότι το ποσοστό των πακέτων δεδομένων που παρελήφθησαν για γράφημα 121 κορυφών είναι σε κάθε περίπτωση μεγαλύτερο από το αντίστοιχο ποσοστό για τα άλλα γραφήματα οφείλεται πάλι στην διαφορά πυκνότητας των αντίστοιχων

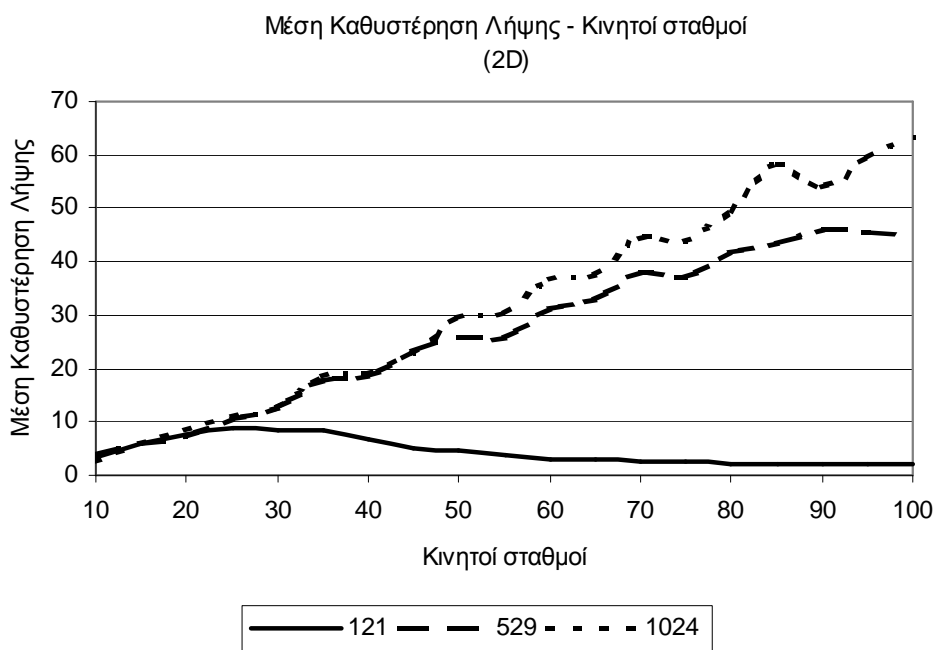
δικτύων. Άλλωστε το γεγονός ότι τα δίκτυα που αντιστοιχούν στα γραφήματα 529 και 1024 κορυφών είναι πολύ αραιά σε κάθε περίπτωση, έχει σαν αποτέλεσμα η αλλαγή του πλήθους των δυνατών προορισμών των πακέτων δεδομένων να μην επηρεάζει τα ποσοστά τους. Η μικρή αύξηση των ποσοστών αυτών των γραφημάτων οφείλεται στην μικρή αύξηση της πυκνότητας τους.



Σχήμα 6.5: Μέσος Αριθμός Βημάτων - Κινητοί Σταθμοί, για διδιάστατα γραφήματα (DSR)

Στο σχήμα 6.5 παρατηρούμε ότι ο μέσος αριθμός των βημάτων για γράφημα 121 κορυφών αυξάνεται στην αρχή, στη συνέχεια μειώνεται και σταθεροποιείται σε μία τιμή. Η αρχική αύξηση οφείλεται στο ότι καθώς αυξάνεται ο αριθμός των κινητών σταθμών το αντίστοιχο δίκτυο γίνεται πιο πυκνό με αποτέλεσμα να δημιουργούνται μεγαλύτερες διαδρομές. Στην συνέχεια ωστόσο το δίκτυο γίνεται τόσο πυκνό ώστε πολλοί σταθμοί να ανήκουν στην ίδια κορυφή του γραφήματος κίνησης, με αποτέλεσμα ο μέσος αριθμός βημάτων να μειώνεται. Τα άλλα γραφήματα έχουν μεγάλο αριθμό κορυφών με αποτέλεσμα να μην εμφανίζονται τέτοια φαινόμενα. Στα γραφήματα αυτά η αύξηση του αριθμού των κινητών σταθμών οδηγεί στην αύξηση της πυκνότητας των αντίστοιχων δικτύων και στη δημιουργία διαδρομών μεγαλύτερου μήκους.

Στο σχήμα 6.6 παρατηρούμε ότι η μέση καθυστέρηση λήψης για γράφημα 121 κορυφών αυξάνεται στην αρχή, στη συνέχεια μειώνεται και σταθεροποιείται



Σχήμα 6.6: Μέση Καθυστέρηση Λήψης - Κινητοί Σταθμοί, για δισδιάστατα γραφήματα (DSR)

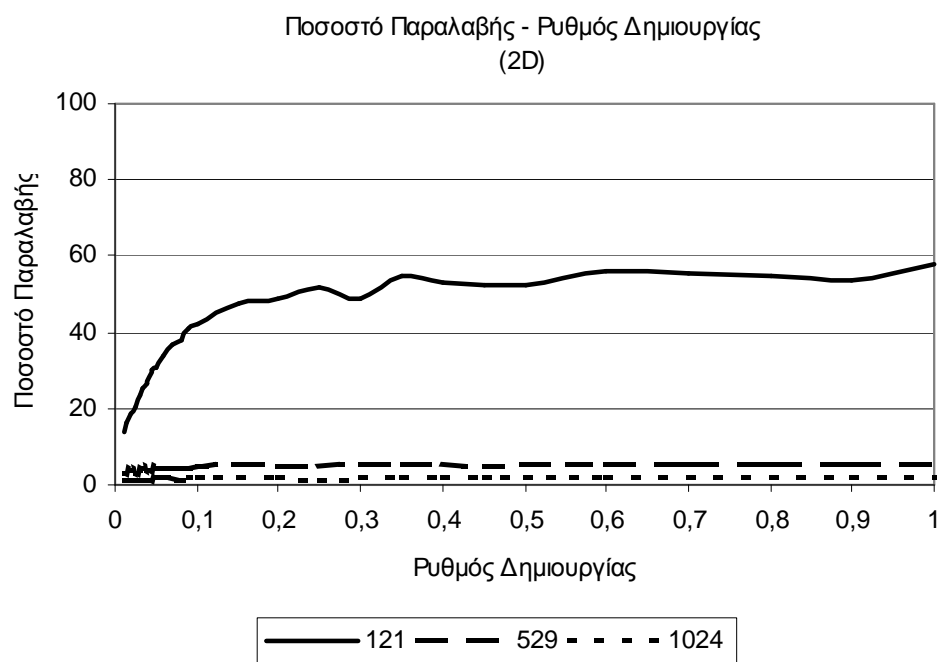
σε μία τιμή. Η μέση καθυστέρηση λήψης φαίνεται να συμπεριφέρεται ανάλογα με τον μέσο αριθμό βημάτων. Επιρεάζεται βέβαια και από την καθυστέρηση των πακέτων δεδομένων από την παραμονή τους στην ΜΠΑΠ, η οποία όμως ομοίως συμπεριφέρεται ανάλογα μιας και το αντίστοιχο δίκτυο γίνεται πιο πυκνό. Τα άλλα γραφήματα επιρεάζονται περισσότερο από την καθυστέρηση των πακέτων δεδομένων από την παραμονή τους στην ΜΠΑΠ. Η καθυστέρηση αυτή αυξάνεται παρόλο που τα αντίστοιχα δίκτυα γίνονται πιο πυκνά. Αυτό οφείλεται στο ότι ακόμα και για 100 κινητούς σταθμούς ο αριθμός κορυφών των γραφημάτων είναι αρκετά μεγαλύτερος. Ενώ παράλληλα το πλήθος των δυνατών προορισμών των πακέτων δεδομένων γίνεται μεγαλύτερο καθώς αυξάνεται ο αριθμός των κινητών σταθμών, μειώνοντας έτσι την πιθανότητα κάποιος προορισμός να είναι προσβάσιμος ή να γίνει προσβάσιμος σύντομα.

Παράλληλα από τα πειράματα που εκτελέσαμε παρατηρήσαμε ότι το πλήθος των πακέτων ελέγχου που στάλθηκαν στο δίκτυο αλλά και η συνολική χρήση της μνήμης αυξάνεται καθώς αυξάνεται αριθμός των κινητών σταθμών, κάτι που είναι λογικό (βλ. παράρτημα).

Τα πειράματα που εκτελέσαμε για τρισδιάστατα γραφήματα αλλάζοντας τον αριθμό των κινητών σταθμών έδωσαν παρόμοια αποτελέσματα με τα αντίστοιχα πειράματα για δισδιάστατα γραφήματα (βλ. παράρτημα). Ωστόσο το γεγονός

ότι τα δίκτυα που αντιστοιχούν στα τρισδιάστατα γραφήματα είναι σχετικά πιο πυκνά είχε σαν αποτέλεσμα η καμπύλη (η αύξηση και στη συνέχεια η μείωση) που παρατηρείται στα γραφήματα 125 κορυφών, στα διάφορα μετρούμενα μεγέθη, να εμφανίζεται σε μικρότερους αριθμούς κινητών σταθμών. Ακόμα εξαιτίας της αύξησης της πυκνότητας των αντίστοιχων δικτύων και στα γραφήματα 512 κορυφών εμφανίζεται μία παρόμοια καμπύλη στα διάφορα μετρούμενα μεγέθη. Αυτή η καμπύλη εμφανίζεται σε έναν μεγαλύτερο αριθμό σταθμών. Παράλληλα για όλα τα μεγέθη τρισδιάστατων γραφημάτων τα ποσοστά των πακέτων δεδομένων που παρελήφθησαν είναι μεγαλύτερα σε σχέση με εκείνα των δισδιάστατων γραφημάτων, γεγονός που πάλι οφείλεται στην αύξηση της πυκνότητας των αντίστοιχων δικτύων.

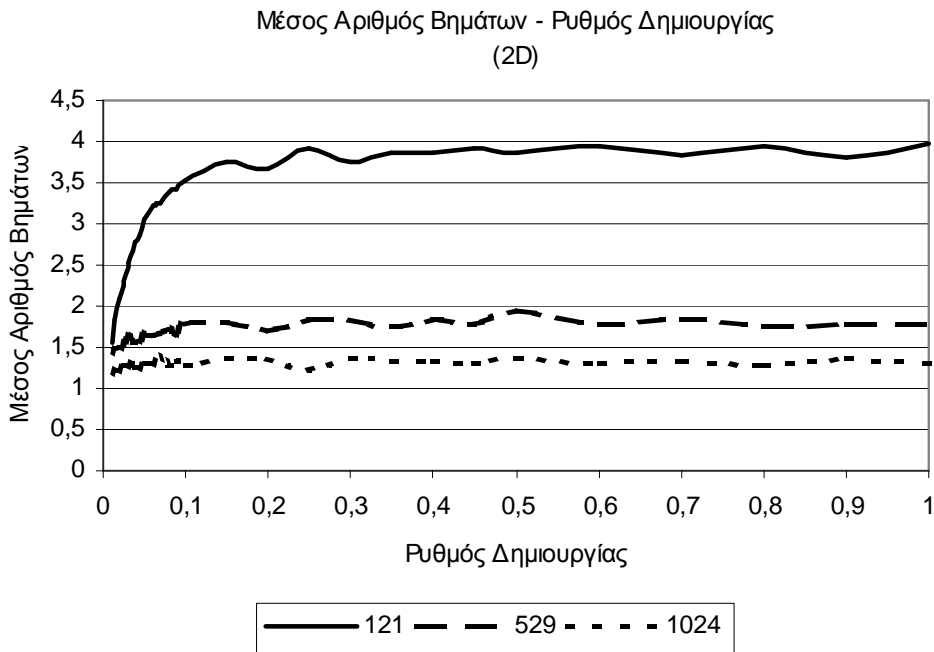
#### 6.5.4 Μεταβολή ρυθμού δημιουργίας πακέτων δεδομένων



Σχήμα 6.7: Ποσοστό Παραλαβής - Ρυθμός Δημιουργίας Πακέτων Δεδομένων, για δισδιάστατα γραφήματα (DSR)

Στο σχήμα 6.7 παρατηρούμε ότι το ποσοστό των πακέτων δεδομένων που παρελήφθησαν για γράφημα 121 κορυφών αυξάνεται. Αυτό οφείλεται στο ότι καθώς ο ρυθμός δημιουργίας πακέτων δεδομένων αυξάνεται τόσο περισσότερα

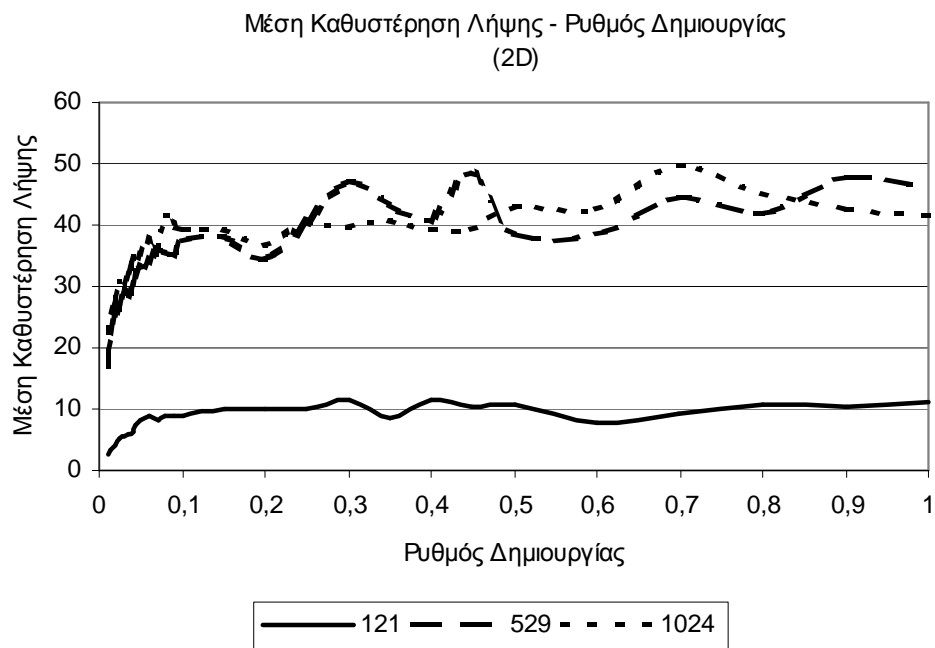
πακέτα διατηρεί ένας κινητός σταθμός για κάποιον προορισμό. Όταν ο προορισμός αυτός γίνει προσβάσιμος (π.χ. γειτονικός) αποστέλλονται σε αυτόν περισσότερα πακέτα δεδομένων από ότι κανονικά, πριν χαθεί πάλι η επαφή με αυτόν. Για τα άλλα μεγέθη γραφημάτων το ποσοστό πακέτων δεδομένων που παρελήφθησαν είναι πιο μικρό και σταθερό, αυτό οφείλεται στο γεγονός ότι τα αντίστοιχα δίκτυα είναι πολύ αραιά. Αυτό έχει σαν αποτέλεσμα πολύ λιγότεροι προορισμοί να γίνονται προσβάσιμοι. Ακόμα συγκρίνοντας τα σχήματα 6.1 και 6.4 με το 6.7 διαπιστώνουμε μία αύξηση στο ποσοστό των πακέτων δεδομένων που παρελήφθησαν, για όλα τα γραφήματα. Αυτό οφείλεται, όπως αναφέραμε, στο ότι όταν ένας προορισμός γίνει προσβάσιμος (π.χ. γειτονικός) αποστέλλονται σε αυτόν περισσότερα πακέτα δεδομένων από ότι κανονικά, πριν χαθεί πάλι η επαφή με αυτόν.



Σχήμα 6.8: Μέσος Αριθμός Βημάτων - Ρυθμός Δημιουργίας Πακέτων Δεδομένων, για δισδιάστατα γραφήματα (DSR)

Στο σχήμα 6.8 παρατηρούμε ότι ο μέσος αριθμός βημάτων για γράφημα 121 κορυφών αυξάνεται απότομα στην αρχή και στην συνέχεια σταθεροποιείται. Τα δίκτυα που αντιστοιχούν στα γραφήματα 121 κορυφών είναι σχετικά πυκνά με αποτέλεσμα να δημιουργούνται διαδρομές μεσαίου μήκους. Όταν ωστόσο ο ρυθμός δημιουργίας πακέτων δεδομένων είναι μικρός και ένας προορισμός γίνει προσβάσιμος μέσω μίας τέτοιας διαδρομής, τότε λίγα πακέτα δεδομένων

ων στέλνονται μέσω αυτής πριν χαθεί πάλι η επαφή. Στην περίπτωση αυτή τα περισσότερα πακέτα δεδομένων μεταφέρονται μεταξύ γειτονικών ή πολύ κοντινών κόμβων. Όταν ωστόσο ο ρυθμός δημιουργίας πακέτων δεδομένων αυξηθεί τότε τα πακέτα δεδομένων που στέλνονται μέσω τέτοιων διαδρομών είναι περισσότερα με αποτέλεσμα να αυξάνεται και ο μέσος αριθμός βημάτων. Το γεγονός ότι ο μέσος αριθμός βημάτων σταθεροποιείται πιθανώς οφείλεται στο ότι δεν μπορούν να δημιουργηθούν διαδρομές μεγαλύτερου μήκους στα αντίστοιχα δίκτυα. Τα δίκτυα που αντιστοιχούν στα άλλα γραφήματα είναι αραιά με αποτέλεσμα να μην δημιουργούνται διαδρομές μεγάλου μήκους και η αύξηση του ρυθμού δημιουργίας πακέτων δεδομένων να έχει πολύ μικρή επίδραση. Ο μέσος αριθμός βημάτων στα γραφήματα του σχήματος 6.8 είναι μεγαλύτερος από ότι στα γραφήματα των σχημάτων 6.2 και 6.5, για τον ίδιο με προηγουμένως λόγο.



Σχήμα 6.9: Μέση Καθυστέρηση Λήψης - Ρυθμός Δημιουργίας Πακέτων Δεδομένων, για δισδιάστατα γραφήματα (DSR)

Στο σχήμα 6.9 η μέση καθυστέρηση λήψης για γράφημα 121 κορυφών φαίνεται να συμπεριφέρεται ανάλογα με τον μέσο αριθμό βημάτων. Η καθυστέρηση των πακέτων δεδομένων από την παραμονή τους στην ΜΠΑΠ δεν επηρεάζει ιδιαίτερα. Ωστόσο η μέση καθυστέρηση λήψης για τα άλλα γραφήματα δεν συμπεριφέρεται ανάλογα με τον αντίστοιχο μέσο αριθμό βημάτων και επηρεάζε-

ται από την καθυστέρηση των πακέτων δεδομένων από την παραμονή τους στην ΜΠΑΠ. Αυτή είναι μεγάλη και μη σταθερή εξαιτίας του γεγονός ότι τα αντίστοιχα δίκτυα είναι αραιά.

Παράλληλα από τα πειράματα που εκτελέσαμε παρατηρήσαμε ότι το πλήθος των πακέτων ελέγχου που στάλθηκαν στο δίκτυο μειώνεται καθώς αυξάνεται ο ρυθμός δημιουργίας πακέτων δεδομένων (βλ. παράρτημα). Αυτό οφείλεται ως ένα βαθμό στην χρήση της ΜΠΑΠ και στην μη συνεχή αποστολή ΑΙΔ για τον ίδιο προορισμό. Ακόμα η χρήση της μνήμης αυξάνεται καθώς αυξάνεται ο ρυθμός δημιουργίας πακέτων δεδομένων (βλ. παράρτημα).

Τα πειράματα που εκτελέσαμε για τρισδιάστατα γραφήματα αλλάζοντας τον ρυθμό δημιουργίας πακέτων δεδομένων έδωσαν παρόμοια αποτελέσματα με τα αντίστοιχα πειράματα για δισδιάστατα γραφήματα (βλ. παράρτημα). Οι όποιες διαφορές οφείλονται κυρίως στο ότι τα αντίστοιχα δίκτυα είναι πιο πυκνά. Σε αυτό άλλωστε οφείλεται η αύξηση των ποσοστών των πακέτων δεδομένων που παρελήφθησαν για τα γραφήματα όλων των μεγεθών, καθώς και η αύξηση του μέσου αριθμού βημάτων για γράφημα 532 κορυφών.



## 6.6 Zone Routing Protocol (ZRP)

### 6.6.1 Γενικά

Για το ZRP προσπαθήσαμε αρχικά να πραγματοποιήσουμε ακριβώς τα ίδια πειράματα με αυτά για το DSR, γρήγορα ωστόσο διαπιστώσαμε ότι αυτό δεν ήταν δυνατό. Αιτία ήταν ο μεγάλος χρόνος ολοκλήρωσης κάθε είδους πειράματος, δηλ. με συγκεκριμένες παραμέτρους. Ο μεγάλος χρόνος ολοκλήρωσης κάθε είδους πειράματος οφείλεται τόσο στο τρόπο λειτουργίας του ZRP όσο και στην υλοποίηση του. Στο ZRP όταν ένας σταθμός λάβει ένα νέο ΠΑΚΑ ή υπάρξει μία αλλαγή στους γείτονες του, τότε πρέπει να υπολογίσει ξανά τη τοπολογία της ΖΔ του. Ο υπολογισμός αυτός υλοποιείται με την χρήση του αλγορίθμου Dijkstra. Το γεγονός ότι, όπως παρατηρήσαμε, σε κάθε γύρο ο υπολογισμός αυτός πραγματοποιείται από πολλούς από τους σταθμούς του δικτύου είναι η βασικότερη αιτία του μεγάλου χρόνου ολοκλήρωσης ενός πειράματος. Ακόμα το πρωτόκολλο ZRP, όπως έχουμε αναφέρει, αποτελείται από τρία επιμέρους πρωτόκολλα, κάθε ένα από τα οποία εκτελεί ορισμένους σχετικά πολύπλοκους και χρονοβόρους υπολογισμούς. Χαρακτηριστικό παράδειγμα ο υπολογισμός, που πραγματοποιείται στο πρωτόκολλο BRP, του multicast - bordercast δέντρου πάνω στο οποίο θα κινηθεί μία αίτηση ανακάλυψης διαδρομής. Αυτός υλοποιείται με την χρήση του αλγορίθμου Dijkstra. Ακόμα τα επιμέρους πρωτόκολλα αλληλεπιδρούν μεταξύ τους είτε για την ανταλλαγή πληροφοριών είτε για την ειδοποίηση για ένα γεγονός, αυξάνοντας έτσι το χρόνο ολοκλήρωσης ενός πειράματος. Χαρακτηριστικό παράδειγμα αποτελεί η λήψη μίας αίτηση ανακάλυψης διαδρομής από το πρωτόκολλο BRP, το οποίο τη στέλνει στο IERP. Το IERP με τη σειρά του αν δεν μπορεί να ικανοποιήσει την αίτηση αυτή τη στέλνει πίσω στο BRP για να την προωθήσει παραπέρα.

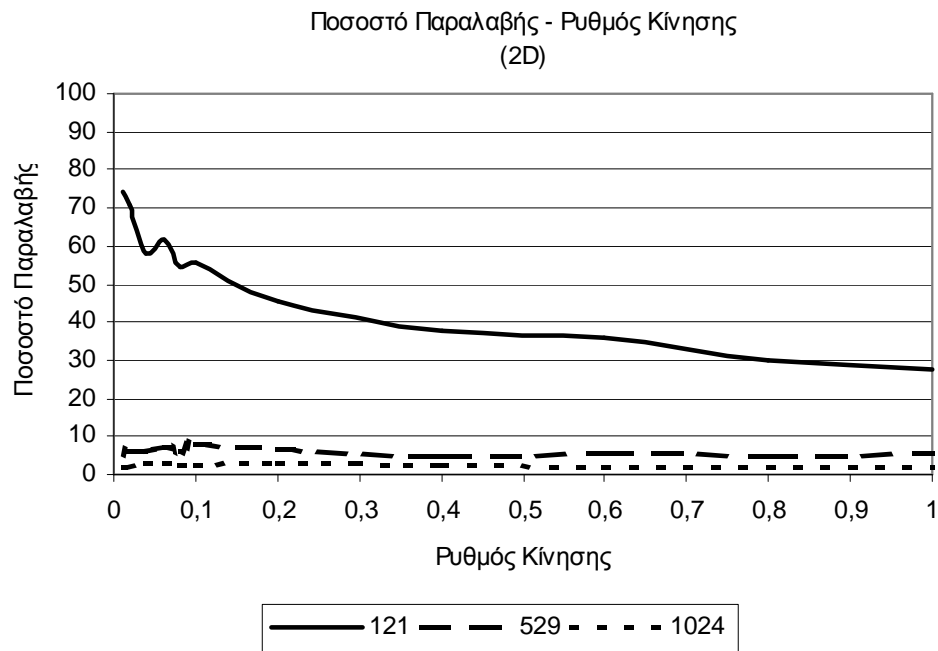
Ο μεγάλος χρόνος ολοκλήρωσης κάθε είδους πειράματος, για τους λόγους που εξηγήσαμε, μας οδήγησε στο να πραγματοποιήσουμε κάποιες αλλαγές στα πειράματα που εκτελέσαμε, με σκοπό την μείωση του χρόνου αυτού διατηρώντας ωστόσο την ικανότητα μας να βγάζουμε αξιόπιστα και συγκρίσιμα αποτελέσματα. Έτσι καταρχάς μειώσαμε το πλήθος των πακέτων δεδομένων που συνολικά οι σταθμοί δημιουργούσαν, σε κάθε πείραμα, σε 1000. Ακόμα μεγαλώσαμε το βήμα αύξησης του ρυθμού κίνησης και του ρυθμού δημιουργίας νέων πακέτων δεδομένων. Επιπλέον δεν πραγματοποιήσαμε πειράματα για τα τρισδιάστατα γραφήματα, παρατηρώντας άλλωστε από άλλα πειράματα μας πως σε κάθε περίπτωση τα αποτελέσματα ήταν παρόμοια. Ακόμα επιλέξαμε η ΖΔ κάθε σταθμού να έχει ακτίνα 3, βασιζόμενοι στην [12]. Όσον αφορά το μέγιστο αριθμό των επαναλήψεων της αποστολής μίας αίτησης ανακάλυψης διαδρομής, αυτός διατηρήθηκε στο δύο. Τέλος κάθε είδος πειράματος επαναληφθεί τόσες φορές ώστε ο συνολικός αριθμός πακέτων δεδομένων που δημιουργήθηκαν, για

το συγκεκριμένο είδος πειράματος, να είναι περίπου 10000.

Γενικά από τα πειράματα που διεξάγαμε παρατηρήσαμε ότι η συμπεριφορά του πρωτοκόλλου (ποσοστό παράδοσης, μέσος αριθμός βημάτων, μέση καθυστέρηση λήψης), για τους διάφορους ρυθμούς κίνησης και δημιουργίας πακέτων δεδομένων αλλά και για τα διάφορα πλήθη κινητών σταθμών, είναι άμεσα εξαρτώμενη από την πυκνότητα των αντίστοιχων δικτύων. Έτσι για τα γραφήματα 121 κορυφών η συμπεριφορά του πρωτοκόλλου είναι γενικά καλύτερη σε σχέση με τα άλλα μεγέθη γραφημάτων. Όσον αφορά τα διάφορα μεταβαλλόμενα μεγέθη, καταρχήν παρατηρήσαμε ότι η αύξηση της κινητικότητας των σταθμών έχει σαν αποτέλεσμα την δημιουργία διαδρομών μικρότερου μήκους και την ανταλλαγή πακέτων δεδομένων μόνο μεταξύ γειτονικών σταθμών. Αυτό βέβαια οδηγεί και στην μείωση του ποσοστού των πακέτων δεδομένων που παρελήφθησαν. Η αύξηση του πλήθους των κινητών σταθμών αυξάνει την πυκνότητα των δικτύων, όπως παράλληλα και το πλήθος των δυνατών προορισμών των πακέτων δεδομένων που δημιουργούνται, με τις ανάλογες βέβαια συνέπειες. Τέλος από τα πειράματα που διεξάγαμε επαληθεύσαμε την ορθότητα της υλοποίησης του πρωτοκόλλου. Η υλοποίηση επηρεάζει ως ένα βαθμό την μέση καθυστέρηση λήψης (λόγω ΜΠΑΠ), την χρήση της μνήμης και τον αριθμό των πακέτων ελέγχου που στάλθηκαν στο δίκτυο.

### 6.6.2 Μεταβολή ρυθμού κίνησης

Στο σχήμα 6.10 παρατηρούμε ότι το ποσοστό των πακέτων δεδομένων που παρελήφθησαν για γράφημα 121 κορυφών μειώνεται καθώς αυξάνεται ο ρυθμός κίνησης. Η μείωση του ποσοστού οφείλεται στο ότι η αυξανόμενη κίνηση των σταθμών οδηγεί στην γρήγορη διάσπαση των διαδρομών που ανακαλύπτονται, με αποτέλεσμα να μην μπορούν να μεταδωθούν πακέτα δεδομένων μέσω αυτών. Ακόμα πολλές φορές μία απάντηση αίτησης διαδρομής δεν μπορεί να επιστρέψει στο σταθμό που δημιούργησε την αντίστοιχη αίτηση ανακάλυψης διαδρομής μιας και η αντεστραμμένη διαδρομή επιστροφής δεν ισχύει πια. Ακόμα το ποσοστό των πακέτων δεδομένων που παρελήφθησαν για γράφημα 121 κορυφών είναι υψηλότερο από τα ποσοστά των άλλων γραφημάτων, για κάθε ρυθμό κίνησης. Αυτό οφείλεται στο ότι τα δίκτυα που αντιστοιχούν στο πρώτο είδος γραφήματος είναι σαφώς πιο πυκνά (50 σταθμοί - 121 κορυφές), με αποτέλεσμα να υπάρχει μεγαλύτερη πιθανότητα κάποιοι προορισμοί να είναι προσβάσιμοι ή να γίνουν προσβάσιμοι σύντομα λόγω της κίνησης των σταθμών. Το γεγονός αυτό εξηγεί και το γιατί τα ποσοστά των πακέτων δεδομένων που παρελήφθησαν για τα δίκτυα που αντιστοιχούν σε γραφήματα 529 και 1024 κορυφών, παραμένει σχεδόν σταθερό για όλους τους ρυθμούς κίνησης. Τα δίκτυα αυτά είναι αραιά (50 σταθμοί - 529 και 1024 κορυφές) με αποτέλεσμα τα πακέτα που παραδίδονται να είναι κυρίως αυτά μεταξύ γειτονικών ή πολύ

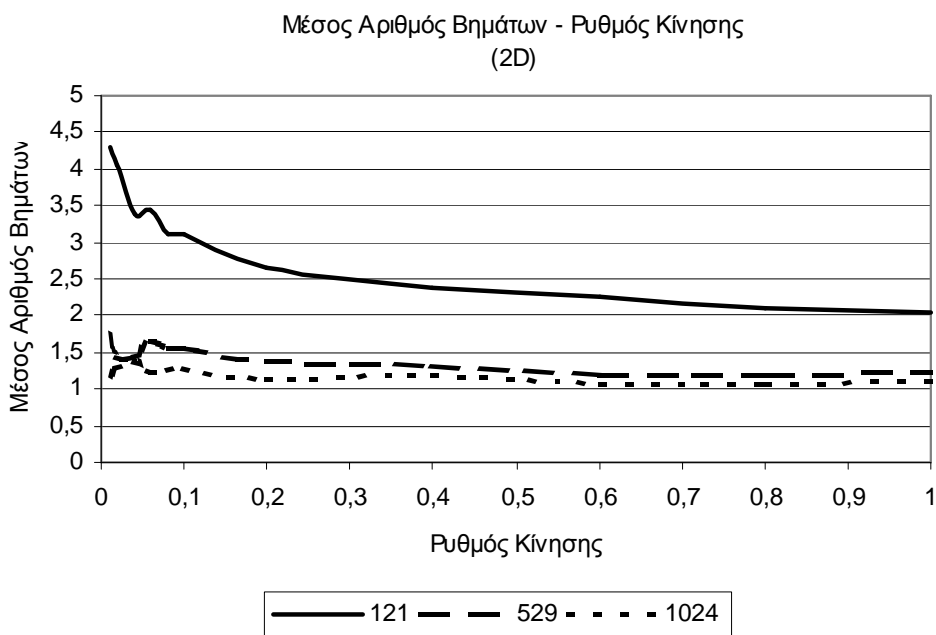


Σχήμα 6.10: Ποσοστό Παραλαβής - Ρυθμός Κίνησης, για δισδιάστατα γραφήματα (ZRP)

κοντινών σταθμών ανεξάρτητα του ρυθμού κίνησης, κάτι που φαίνεται και από το σχήμα 6.11.

Στο σχήμα 6.11 παρατηρούμε ότι και για τα τρία μεγέθη γραφημάτων ο μέσος αριθμός βημάτων μειώνεται καθώς αυξάνεται ο ρυθμός κίνησης. Αυτό οφείλεται στο γεγονός ότι η αύξηση της κίνησης των σταθμών έχει σαν αποτέλεσμα πακέτα δεδομένων να παραδίδονται κυρίως μεταξύ γειτονικών ή πολύ κοντινών σταθμών. Ακόμα σε κάθε περίπτωση ο μέσος αριθμός βημάτων για τα γραφήματα 121 κορυφών είναι μεγαλύτερος από αυτόν για τα γραφήματα 529 και 1024 κορυφών, για τα οποία ο μέσος αριθμός βημάτων είναι πολύ μικρός, κοντά στο 1, και από ένα σημείο και μετά σταθερός. Αυτό οφείλεται στη, όπως αναφέραμε, διαφορά πυκνότητας των αντίστοιχων δικτύων. Έτσι στα γραφήματα 529 και 1024 κορυφών πακέτα παραδίδονται κυρίως μεταξύ γειτονικών ή πολύ κοντινών σταθμών ανεξάρτητα του ρυθμού κίνησης.

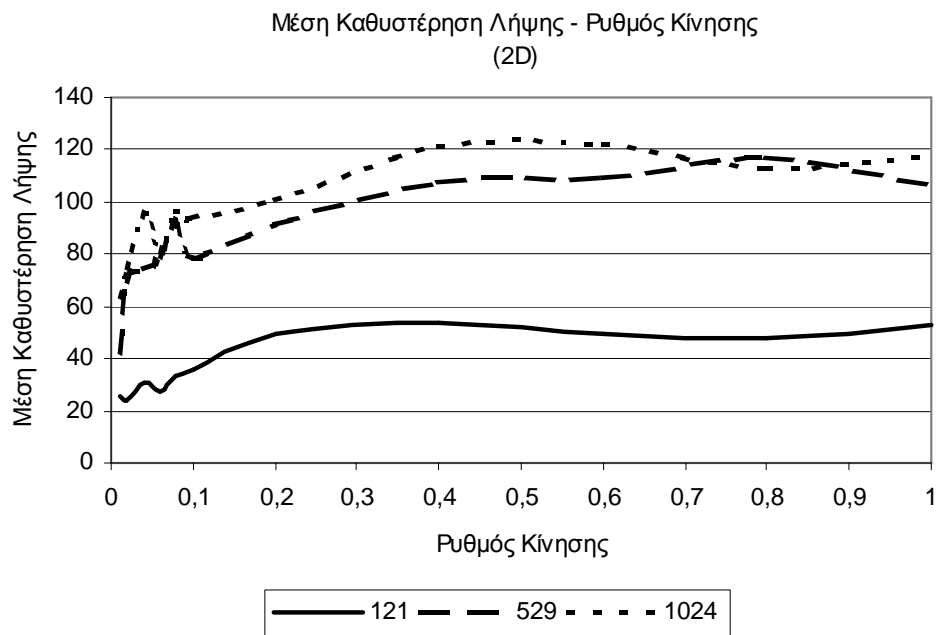
Η μέση καθυστέρηση λήψης δεν εξαρτάται μόνο από το μέσο αριθμό βημάτων που το πακέτο διανύει, αλλά και από το χρόνο που αυτό παραμένει στην ΜΠΑΠ μέχρι να βρεθεί μία διαδρομή προς τον προορισμό. Παρατηρώντας τα σχήματα 6.11 και 6.12 συμπεραίνουμε ότι ο χρόνος παραμονής στην ΜΠΑΠ επηρεάζει περισσότερο την μέση καθυστέρηση λήψης, μιας και ο μέσος αριθ-



Σχήμα 6.11: Μέσος Αριθμός Βημάτων - Ρυθμός Κίνησης, για δισδιάστατα γραφήματα (ZRP)

θμός βημάτων είναι πολύ μικρός για όλα τα γραφήματα. Ακόμα για όλα τα γραφήματα η μέση καθυστέρηση λήψης αυξάνεται αρχικά και στην συνέχεια σταθεροποιείται. Αυτό οφείλεται στο γεγονός ότι η αυξανόμενη κίνηση των σταθμών προκαλεί μεγαλύτερους χρόνους παραμονής στην ΜΠΑΠ, μέχρι να βρεθεί μία διαδρομή προς έναν προορισμό. Η σταθεροποίηση της μέσης καθυστέρησης λήψης δείχνει ότι από ένα σημείο και μετά ο χρόνος παραμονής στην ΜΠΑΠ παραμένει σταθερός. Ακόμα τα γραφήματα 121 κορυφών έχουν μικρότερη μέση καθυστέρηση λήψης από τα γραφήματα 529 και 1024 κορυφών, εξαιτίας το γεγονόςτος ότι τα αντίστοιχα δίκτυα είναι πιο πυκνά με αποτέλεσμα να υπάρχει μεγάλη πιθανότητα κάποιοι προορισμοί να είναι άμεσα προσβάσιμοι ή να γίνουν προσβάσιμοι σύντομα λόγω της κίνησης των σταθμών.

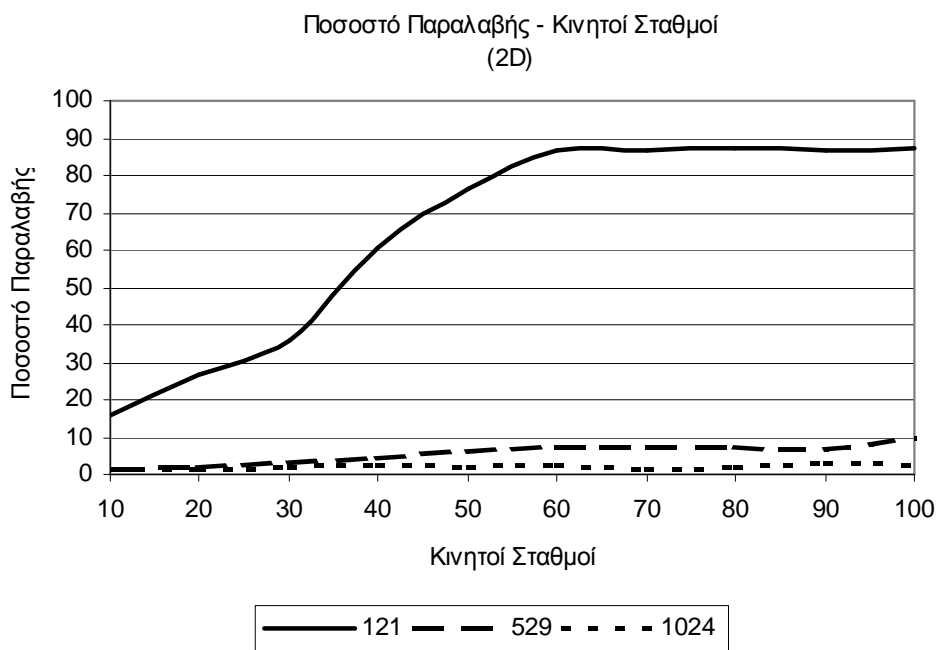
Παράλληλα από τα πειράματα που εκτελέσαμε παρατηρήσαμε ότι το πλήθος των πακέτων ελέγχου που στάλθηκαν στο δίκτυο αυξάνεται για όλα τα γραφήματα (βλ. παράρτημα). Αυτό οφείλεται στο γεγονός ότι η αύξηση του ρυθμού κίνησης έχει σαν αποτέλεσμα να μεταδίδονται περισσότερα ΠΑΚΑ. Τα γραφήματα βέβαια 121 κορυφών μεταδίδουν περισσότερα ΠΑΚΑ μιας και είναι πιο πυκνά.



Σχήμα 6.12: Μέση Καθυστέρηση Λήψης - Ρυθμός Κίνησης, για διαστάσιμα γραφήματα (ZRP)

### 6.6.3 Μεταβολή πλήθους κινητών σταθμών

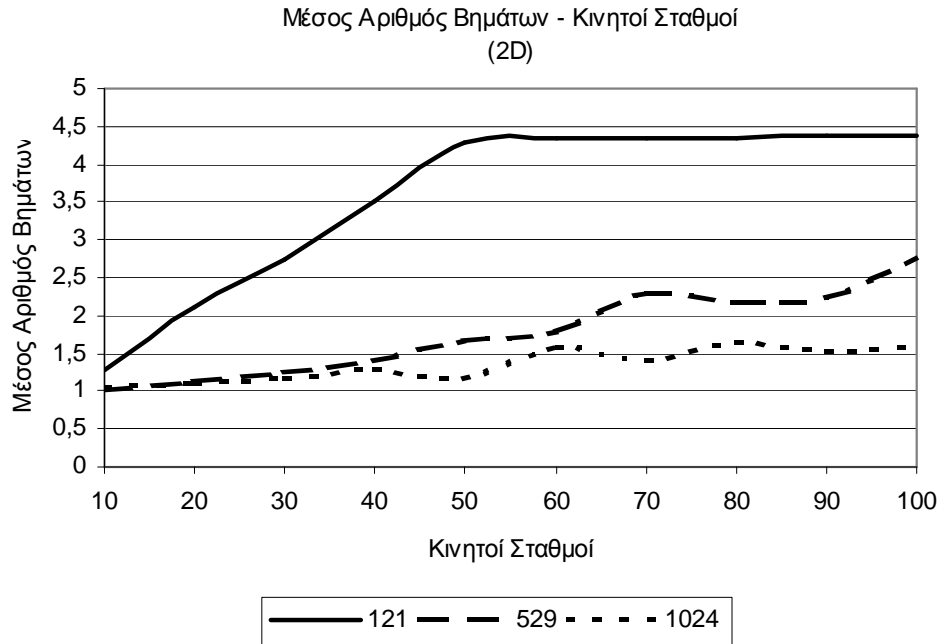
Στο σχήμα 6.13 παρατηρούμε ότι το ποσοστό των πακέτων δεδομένων που παρελήφθησαν για γράφημα 121 κορυφών αυξάνεται στην αρχή και στη συνέχεια σταθεροποιείται σε μία τιμή. Η αρχική αύξηση που παρουσιάζει το ποσοστό αυτό οφείλεται στην αύξηση της πυκνότητας του αντίστοιχου δικτύου, που προκαλείται από την ύπαρξη περισσότερων κινητών σταθμών. Η σταθεροποίηση του ποσοστού που ακολουθεί οφείλεται στο γεγονός ότι η αύξηση των κινητών σταθμών οδηγεί και στην αύξηση των πιθανών προορισμών των πακέτων δεδομένων που δημιουργούνται. Στην αρχή δηλαδή το δίκτυο μπορεί να είναι πιο αραιό από ότι στη συνέχεια αλλά οι πιθανοί προορισμοί των πακέτων δεδομένων που δημιουργούνται είναι πολλοί λίγοι, αυξάνοντας έτσι την πιθανότητα παράδοσης ενός πακέτου δεδομένων. Ακόμα το γεγονός ότι το ποσοστό των πακέτων δεδομένων που παρελήφθησαν για γράφημα 121 κορυφών είναι σε κάθε περίπτωση μεγαλύτερο από το αντίστοιχο ποσοστό για τα άλλα γραφήματα οφείλεται πάλι στη διαφορά πυκνότητας των αντίστοιχων δικτύων. Άλλωστε το γεγονός ότι τα δίκτυα που αντιστοιχούν στα γραφήματα 529 και 1024 κορυφών είναι πολύ αραιά σε κάθε περίπτωση, έχει σαν αποτέλεσμα η αύξηση του πλήθους των σταθμών να μην αυξάνει ιδιαίτερα τα ποσοστά τους.



Σχήμα 6.13: Ποσοστό Παραλαβής - Κινητοί Σταθμοί, για δισδιάστατα γραφήματα (ZRP)

Στο σχήμα 6.14 παρατηρούμε ότι ο μέσος αριθμός των βημάτων για γράφημα 121 κορυφών αυξάνεται στην αρχή και στη συνέχεια σταθεροποιείται σε μία τιμή. Η αρχική αύξηση οφείλεται στο ότι καθώς αυξάνεται ο αριθμός των κινητών σταθμών το αντίστοιχο δίκτυο γίνεται πιο πυκνό με αποτέλεσμα να δημιουργούνται μεγαλύτερες διαδρομές. Στην συνέχεια ωστόσο το δίκτυο γίνεται τόσο πυκνό ώστε πολλοί σταθμοί να ανήκουν στην ίδια κορυφή του γραφήματος κίνησης, με αποτέλεσμα ο μέσος αριθμός βημάτων να μην αυξάνεται άλλο παρόλο που το δίκτυο γίνεται πιο πυκνό. Τα άλλα γραφήματα έχουν μεγάλο αριθμό κορυφών με αποτέλεσμα να μην εμφανίζονται τέτοια φαινόμενα. Στα γραφήματα αυτά η αύξηση του αριθμού των κινητών σταθμών οδηγεί στην αύξηση της πυκνότητας των αντίστοιχων δικτύων και στη δημιουργία διαδρομών μεγαλύτερου μήκους.

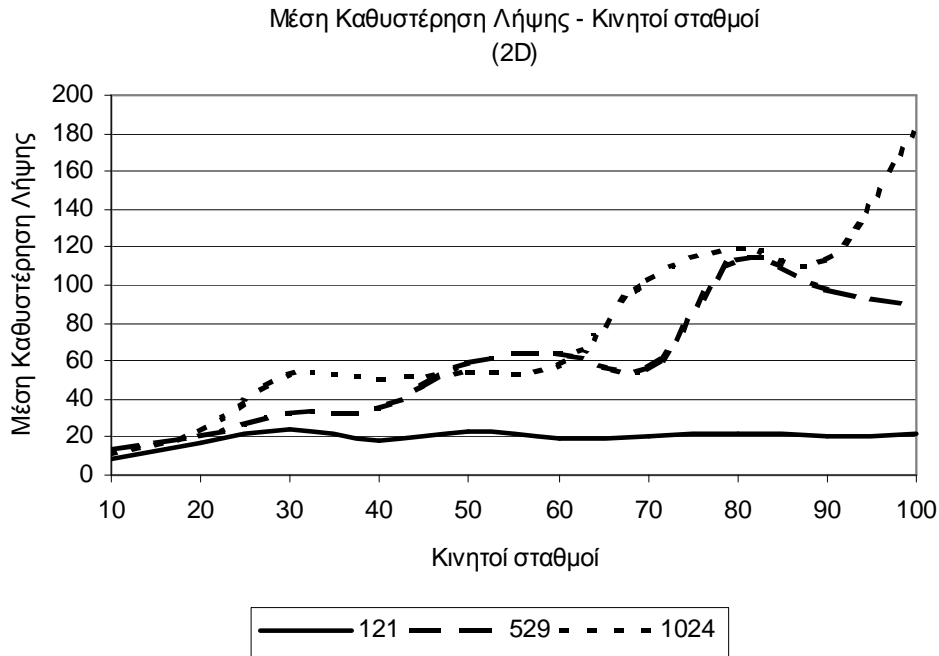
Στο σχήμα 6.15 παρατηρούμε ότι η μέση καθυστέρηση λήψης για γράφημα 121 κορυφών αυξάνεται στην αρχή και στη συνέχεια σταθεροποιείται, ενώ στα άλλα γραφήματα μόνο αυξάνεται. Θα περίμενε κανείς το γεγονός ότι τα αντίστοιχα δίκτυα γίνονται πιο πυκνά να είχε ως αποτέλεσμα την μείωση της μέσης καθυστέρησης λήψης, μιας και έτσι ανακαλύπτονται με μεγαλύτερη πιθανότητα διαδρομές προς τους διάφορους προορισμούς. Ωστόσο η αύξηση του αριθμού των κινητών σταθμών έχει σαν αποτέλεσμα το πλήθος των δυνατών



Σχήμα 6.14: Μέσος Αριθμός Βημάτων - Κινητοί Σταθμοί, για δισδιάστατα γραφήματα (ZRP)

προορισμών των πακέτων δεδομένων να γίνεται μεγαλύτερο, μειώνοντας έτσι την πιθανότητα κάποιος προορισμός να είναι προσβάσιμος ή να γίνει προσβάσιμος σύντομα. Έτσι τα πακέτα δεδομένων παραμένουν για περισσότερο χρόνο στην ΜΠΑΠ. Ακόμα η μέση καθυστέρηση λήψης είναι μεγαλύτερη στα γραφήματα 529 και 1024 κορυφών από ότι στο γράφημα 121 κορυφών. Αυτό οφείλεται στο ότι ακόμα και για 100 κινητούς σταθμούς ο αριθμός κορυφών των γραφημάτων 529 και 1024 κορυφών είναι αρκετά μεγαλύτερος και έτσι αυτά παραμένουν σχετικά αραιά. Επομένως ο χρόνος που χρειάζεται για να γίνει κάποιος προορισμός προσβάσιμος είναι υψηλός.

Παράλληλα από τα πειράματα που εκτελέσαμε παρατηρήσαμε ότι το πλήθος των πακέτων ελέγχου που στάλθηκαν στο δίκτυο αλλά και η συνολική χρήση της μνήμης αυξάνεται καθώς αυξάνεται αριθμός των κινητών σταθμών, κάτι που είναι λογικό (βλ. παράρτημα).



Σχήμα 6.15: Μέση Καθυστέρηση Λήψης - Κινητοί Σταθμοί, για διαστάτα γραφήματα (ZRP)

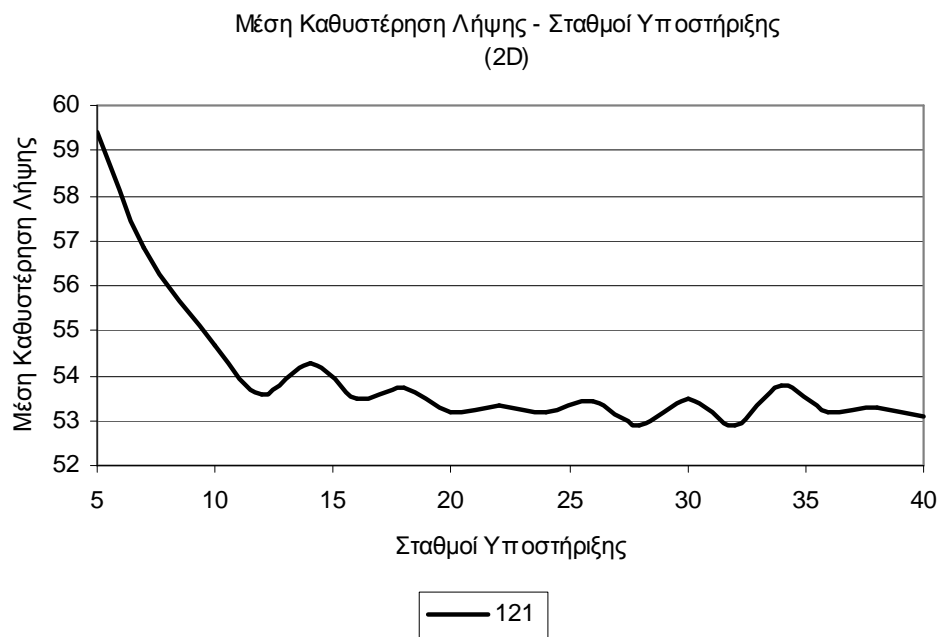
#### 6.6.4 Μεταβολή ρυθμού δημιουργίας πακέτων δεδομένων

Από τα πειράματα που πραγματοποιήσαμε αλλάζοντας τον ρυθμό δημιουργίας νέων πακέτων δεδομένων δεν μπορέσαμε να βγάλουμε συμπεράσματα μιας και τα αποτελέσματα δεν παρουσίαζαν κάποια συγκεκριμένη συμπεριφορά, πέρα από το γεγονός ότι τα γραφήματα 121 κορυφών υπερτερούσαν σε κάθε περίπτωση από τα γραφήματα των άλλων μεγεθών λόγω της μεγαλύτερης πυκνότητας τους. Βασικός λόγος για την συμπεριφορά αυτή ήταν ο μικρός αριθμός πακέτων δεδομένων που κάθε σταθμός δημιουργούσε. Παρόλα αυτά μπορούμε να υποθέσουμε ότι η αύξηση του ρυθμού δημιουργίας νέων πακέτων δεδομένων θα έχει ως αποτέλεσμα περισσότερα πακέτα δεδομένων να αποστέλλονται μέσω μίας διαδρομής που ανακαλύφθηκε, αυξάνοντας έτσι το ποσοστό παράδοσης. Ο ίδιος λόγος θα οδηγεί και στην αύξηση του μέσου αριθμού βημάτων, μιας και διαδρομές μεγαλύτερου από κανονικά μήκους θα χρησιμοποιούνται περισσότερο πριν η κίνηση των σταθμών προκαλέσει την διάσπασή τους.



## 6.7 Support Routing Protocol (SRP)

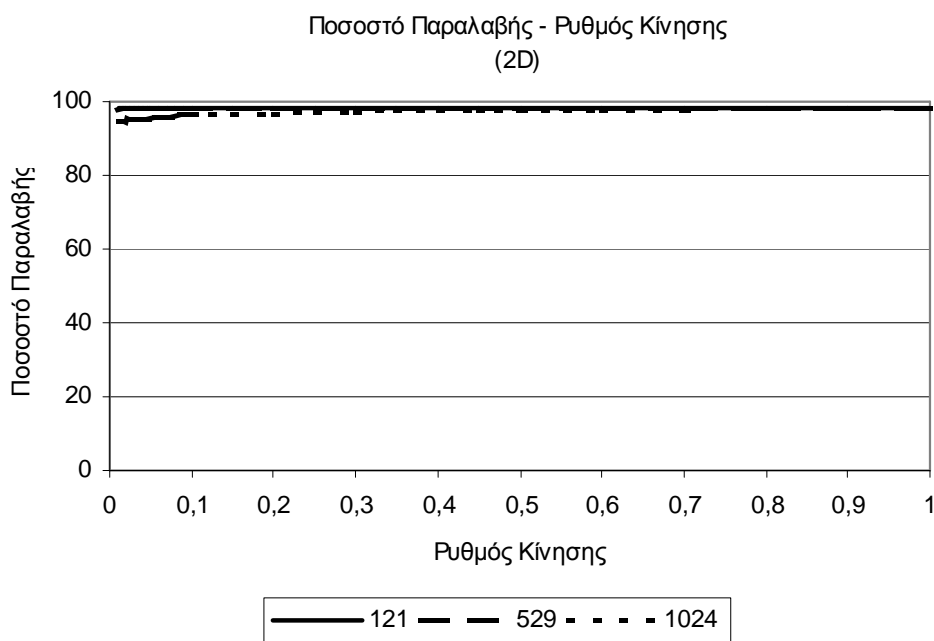
Θέλοντας να συγκρίνουμε τα πρωτόκολλα DSR και ZRP, που στηρίζονται στην ανακάλυψη διαδρομών από σταθμούς, με πρωτόκολλα που στηρίζονται στην κίνηση των σταθμών, πραγματοποιήσαμε πειράματα και με το πρωτόκολλο SRP, χρησιμοποιώντας σαν πρωτόκολλο διαχείρισης της υποστήριξης τους δρομείς [14]. Στα πειράματα μας χρησιμοποιήσαμε την υλοποίηση του πρωτοκόλλου SRP της [14]. Τα πειράματα που εκτελέσαμε ήταν παρόμοια με αυτά για τον DSR και τον ZRP, με την διαφορά ότι τα μετρούμενα μεγέθη ήταν το ποσοστό των πακέτων δεδομένων που παρελήφθησαν και η μέση καθυστέρηση λήψης. Άλλωστε ο υπολογισμός του μέσου αριθμού βημάτων δεν έχει νόημα σε ένα πρωτόκολλο σαν το SRP, μιας και δεν στηρίζεται στην ανακάλυψη διαδρομών. Κάθε είδος πειράματος, δηλαδή με συγκεκριμένες παραμέτρους, επαναληφθεί τόσες φορές ώστε ο συνολικός αριθμός πακέτων δεδομένων που δημιουργήθηκαν, για το συγκεκριμένο είδος πειράματος, να είναι περίπου 200000.



Σχήμα 6.16: Μέση Καθυστέρηση Λήψης - Σταθμοί Υποστήριξης, για δισδιάστατα γραφήματα (121) (SRP)

Πριν βέβαια ξεκινήσουμε την εκτέλεση των πειραμάτων έπρεπε να αποφασίσουμε ποιο θα ήταν το κατάλληλο πλήθος των σταθμών της υποστήριξης για κάθε μέγεθος γραφήματος. Για το σκοπό αυτό εκτελέσαμε έναν αριθμό

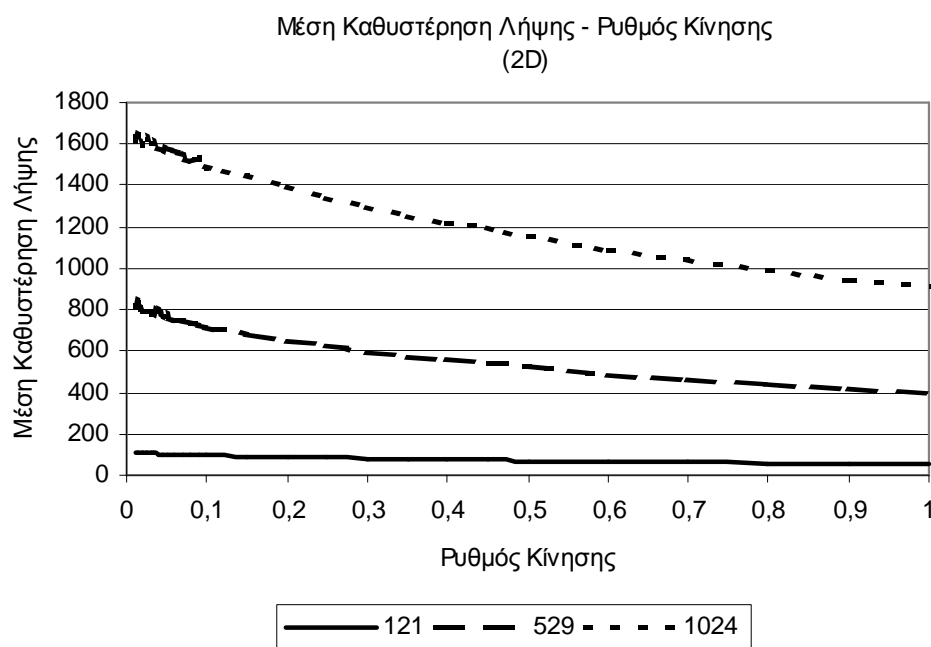
πειραμάτων όπου μετρούσαμε τη μέση καθυστέρηση λήψης μεταβάλλοντας το πλήθος των σταθμών της υποστήριξης (π.χ. σχήμα 6.16). Στα πειράματα αυτά οι υπόλοιπες παράμετροι ήταν σταθερές στις προκαθορισμένες τιμές τους, εκτός του ρυθμού κίνησης των σταθμών που δεν ανήκουν στην υποστήριξη, που ήταν 1 (μέγιστος - κίνηση κάθε σταθμού σε κάθε γύρο). Οι σταθμοί της υποστήριξης πάντοτε κινούνται με το μέγιστο δυνατό ρυθμό. Από τα πειράματα αυτά αλλά και από τα αποτελέσματα της [14] καταλήξαμε στα πλήθη των σταθμών της υποστήριξης. Συγκεκριμένα για γραφήματα 121 (ή 125) κορυφών επιλέξαμε το μέγεθος της υποστήριξης να είναι 10 σταθμοί, για γραφήματα 529 (ή 512) κορυφών να είναι 15 σταθμοί και για γραφήματα 1024 (ή 1000) κορυφών να είναι 20 σταθμοί. (Όλα τα υπόλοιπα πειράματα περιέχονται στο παράρτημα.)



Σχήμα 6.17: Ποσοστό Παραλαβής - Ρυθμός Κίνησης, για δισδιάστατα γραφήματα (SRP)

Από τα υπόλοιπα πειράματα που εκτελέσαμε, χρησιμοποιώντας τα παραπάνω πλήθη σταθμών υποστήριξης, διαπιστώσαμε ότι το ποσοστό των πακέτων δεδομένων που παρελήφθησαν πλησιάζει σε κάθε περίπτωση το 100% (π.χ. σχήμα 6.17). Ακόμα η αυξημένη κινητικότητα των σταθμών που δεν ανήκουν στην υποστήριξη καθώς και η αυξημένη πυκνότητα του δικτύου γενικά ευνοούν το ποσοστό παράδοσης μιας και αυξάνουν την πιθανότητα συνάντησης των σταθμών με την υποστήριξη. Οι ίδιοι παράγοντες μειώνουν και την μέση

καθυστέρηση λήψης, που γενικώς είναι ιδιαίτερα υψηλή (π.χ. σχήμα 6.18). Η μείωση της μέσης καθυστέρησης λήψης με την αύξηση της πυκνότητας των δικτύων γίνεται εύκολα εμφανής παρατηρώντας τα αντίστοιχα πειράματα για δισδιάστατα και τρισδιάστατα γραφήματα, τα οποία κατά τα άλλα παρουσιάζουν παρόμοια συμπεριφορά. Αυτή η μείωση γίνεται ακόμα εμφανής παρατηρώντας και γραφήματα διαφορετικών μεγεθών (121, 529, 1024 ή 121, 512, 1000). Τέλος οι μεγαλύτεροι ρυθμοί δημιουργίας πακέτων δεδομένων δεν επηρεάζουν ιδιαίτερα το ποσοστό παράδοσης ούτε και τη μέση καθυστέρηση λήψης, τα οποία διατηρούνται σχεδόν σταθερά στις μέγιστες τιμές τους. (Όλα τα υπόλοιπα πειράματα περιέχονται στο παράρτημα.)

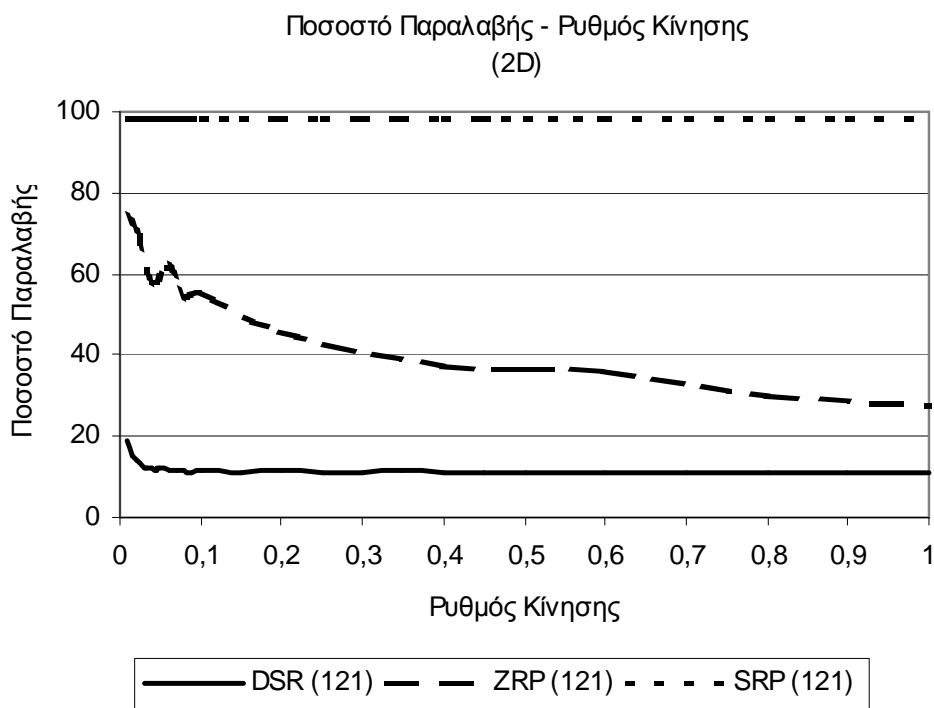


Σχήμα 6.18: Μέση Καθυστέρηση Λήψης - Ρυθμός Κίνησης, για δισδιάστατα γραφήματα (SRP)

## 6.8 Σύγκριση DSR, ZRP και SRP

Συγκρίνοντας τα πρωτόκολλα DSR και ZRP η πρώτη παρατήρηση που μπορούμε να κάνουμε είναι ότι τα διάφορα χαρακτηριστικά τους (ποσοστό παράδοσης, μέσος αριθμός βημάτων, μέση καθυστέρηση λήψης) επηρεάζονται με τον ίδιο τρόπο από τα διάφορα μεταβαλλόμενα μεγέθη (ρυθμός κίνησης και

δημιουργίας πακέτων δεδομένων, πλήθος κινητών σταθμών). Ακόμα τα διάφορα χαρακτηριστικά τους επηρεάζονται με τον ίδιο τρόπο και από την πυκνότητα των δικτύων στα οποία εφαρμόζονται. Η αιτία για αυτή την ομοιότητα είναι βέβαια το γεγονός ότι και τα δύο πρωτόκολλα στηρίζονται στην ανακάλυψη διαδρομών από σταθμούς. Παρόλα αυτά διαφέρουν ως προς τα μεγέθη αυτών των χαρακτηριστικών τους. Επιπλέον πρέπει να αναφέρουμε ότι το ZRP είναι πιο πολύπλοκο στη λειτουργία και στην υλοποίησή του, αλλά και πιο αργό στην εκτέλεση του από το DSR.



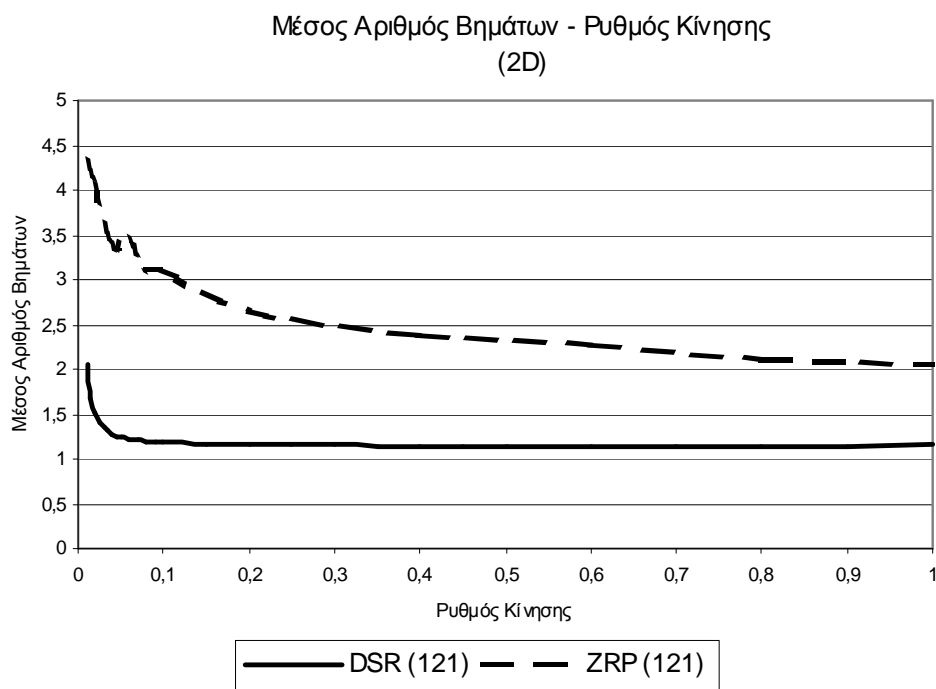
Σχήμα 6.19: Ποσοστό Παραλαβής - Ρυθμός Κίνησης, για δισδιάστατα γραφήματα (DSR, ZRP, SRP)

Το ποσοστό παράδοσης στο πρωτόκολλο ZRP είναι σε κάθε περίπτωση υψηλότερο από αυτό του DSR (π.χ. σχήμα 6.19). Αυτό οφείλεται στο ότι στο ZRP κάθε σταθμός διατηρεί γνώση της τοπολογίας της ΖΔ του, φροντίζοντας να την ενημερώνει ύστερα από τις όποιες αλλαγές. Με τον τρόπο αυτό όταν ο προορισμός ενός πακέτου δεδομένων βρίσκεται μέσα στην ΖΔ του σταθμού πηγής, η πιθανότητα παραλαβής του πακέτου δεδομένων από τον προορισμό είναι μεγάλη. Ακόμα και σε περίπτωση που χρειαστεί να εφαρμοστεί η διαδικασία ανακάλυψης διαδρομής, η πιθανότητα ολοκλήρωσης αυτής με την επιστροφή της αντίστοιχης απάντησης αίτησης διαδρομής αλλά και η πιθανότητα η δι-

αδρομή που ανακαλύφθηκε να συνεχίζει να είναι έγκυρη είναι αυξημένη. Αυτό οφείλεται στο γεγονός ότι στο ZRP επιταγχύνεται η διαδικασία ανακάλυψης διαδρομής, αφού μία αίτηση ανακάλυψης διαδρομής δεν χρειάζεται σχεδόν ποτέ να φτάσει στον προορισμό αλλά σε ένα σταθμό μέσα στη ΖΔ του οποίου υπάρχει ο προορισμός. Έτσι πολλές φορές ούτε η αίτηση ανακάλυψης διαδρομής αλλά ούτε και η αντίστοιχη απάντηση διανύουν μεγάλες διαδρομές, μειώνοντας την πιθανότητα διάσπασης της διαδρομής που ανακαλύφθηκε λόγω της κίνησης των σταθμών. Αυτό βρίσκει ιδιαίτερα εφαρμογή για προορισμούς που βρίσκονται λίγο πιο έξω από τα όρια της ΖΔ του σταθμού πηγής. Ακόμα κάθε ενδιαμέσος σταθμός ελέγχει αν είναι έγκυρη η πληροφορία δρομολόγησης που μεταφέρει ένα πακέτο δεδομένων και σε περίπτωση που δεν είναι έγκυρη προσπαθεί να την διορθώσει. Αυτό έχει σαν αποτέλεσμα την αύξηση του ποσοστού παράδοσης. Επιπλέον στο ZRP σε περίπτωση αλλαγής της τοπολογίας της ΖΔ ανανεώνεται κατάλληλα και ο πίνακας δρομολόγησης του IERP, μειώνοντας την πιθανότητα μία διαδρομή που περιέχεται μέσα στον πίνακα να μην είναι έγκυρη. Το DSR από την άλλη μεριά στηρίζεται σχεδόν αποκλειστικά στη διαδικασία ανακάλυψης διαδρομής. Η πιθανότητα ολοκλήρωσης αυτής με την επιστροφή της αντίστοιχης ΑΠΔ αλλά και η πιθανότητα η διαδρομή που ανακαλύφθηκε να συνεχίζει να είναι έγκυρη είναι μικρότερη. Αυτό οφείλεται στο ότι η αντίστοιχη ΑΙΔ στις περισσότερες περιπτώσεις θα πρέπει να φτάσει μέχρι τον προορισμό. Ωστόσο η κίνηση των σταθμών μπορεί να έχει ως αποτέλεσμα η διαδρομή που ανακαλύφθηκε να μην είναι πλέον έγκυρη. Ακόμα στο DSR δεν υπάρχει καμμία δυνατότητα διόρθωσης από ενδιαμέσους σταθμούς της πληροφορίας δρομολόγησης που μεταφέρει ένα πακέτο δεδομένων, μειώνοντας έτσι την πιθανότητα παράδοσης του. Τέλος οι πληροφορίες που κάθε σταθμός διατηρεί στην ΜΠΑΔ του δεν ανανεώνονται τόσο ώστε να αποτραπεί η χρήση διαδρομών οι οποίες δεν είναι πλέον έγκυρες. (Όλα τα υπόλοιπα πειράματα περιέχονται στο παράρτημα.)

Σε όλα όσα αναφέρθηκαν παραπάνω οφείλεται και το γεγονός ότι ο μέσος αριθμός βημάτων στο πρωτόκολλο ZRP είναι σε κάθε περίπτωση μεγαλύτερος από αυτόν του DSR (π.χ. σχήμα 6.20). Στο ZRP ανακαλύπτονται και χρησιμοποιούνται με μεγαλύτερη πιθανότητα διαδρομές μεγάλου σχετικά μήκους. Άλλωστε η χρήση των ΖΔ επιτρέπει την άμεση εκμετάλλευση γνωστών και συνήθως έγκυρων διαδρομών ίσου ή και μεγαλύτερου μήκους από αυτές που χρησιμοποιούνται στην πλειοψηφία των περιπτώσεων στο DSR. Στο DSR πολλές διαδρομές μεγάλου σχετικά μήκους που ανακαλύπτονται εντέλει δεν χρησιμοποιούνται εξαιτίας της διάσπασης τους. (Όλα τα υπόλοιπα πειράματα περιέχονται στο παράρτημα.)

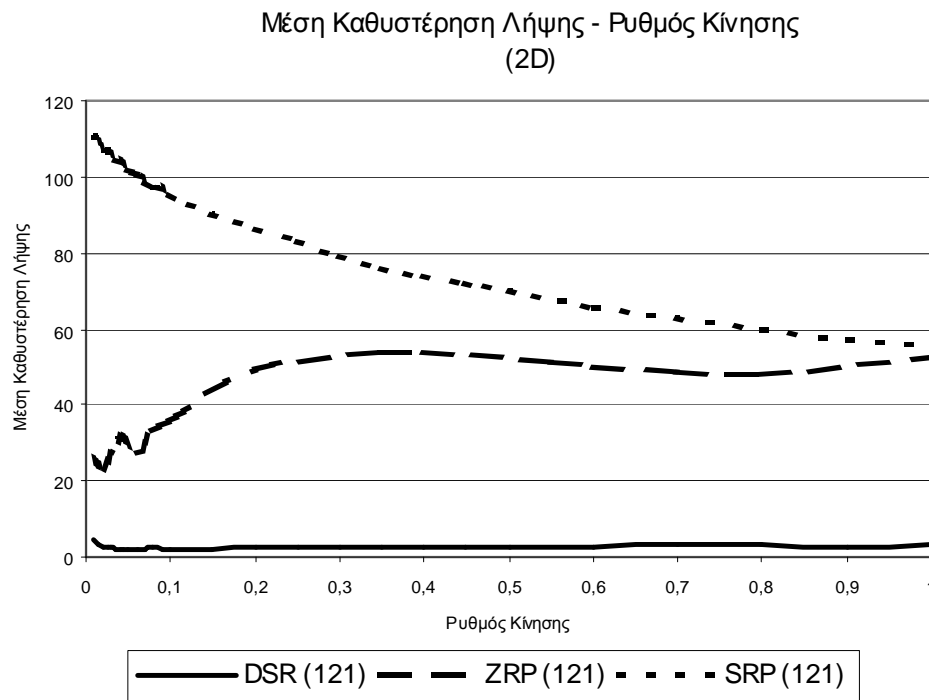
Όσον αφορά τη μέση καθυστέρηση λήψης θα περίμενε κανείς, από όσα έχουν αναφερθεί, να είναι μικρότερη στο πρωτόκολλο ZRP από ότι στο DSR. Σε αυτό συνηγορούν η άμεση χρήση των διαδρομών των ΖΔ και η επιτάχυνση της



Σχήμα 6.20: Μέσος Αριθμός Βημάτων - Ρυθμός Κίνησης, για διδιάστατα γραφήματα (DSR, ZRP)

διαδικασίας ανακάλυψης διαδρομής. Ωστόσο στην πράξη αυτό δεν συμβαίνει, δηλαδή η μέση καθυστέρηση λήψης στο πρωτόκολλο ZRP είναι μεγαλύτερη από ότι στο πρωτόκολλο DSR (σχήμα 6.21). Αυτό οφείλεται κυρίως στη χρήση της ΜΠΑΠ. Στο DSR τα πακέτα δεδομένων που παραμένουν στην ΜΠΑΠ και μεταδίδονται μετά από ένα σχετικά αυξημένο χρονικό διάστημα είναι λίγα. Αντίθετα στο ZRP αυτά τα πακέτα δεδομένων είναι περισσότερα, εξαιτίας των αυξημένων δυνατοτήτων του πρωτοκόλλου για παράδοση πακέτων δεδομένων στον προορισμό τους. Θα πρέπει ωστόσο εδώ να αναφέρουμε ότι η εικόνα αυτή ίσως αδικεί το πρωτόκολλο ZRP. Δηλαδή η χρήση του ΜΠΑΠ στο ZRP δεν βρίσκεται μέσα στις προδιαγραφές του πρωτοκόλλου. Ωστόσο επειδή σε αυτές απουσιάζει και οποιαδήποτε ουσιαστική αναφορά σε μηχανισμούς μεταφοράς πακέτων δεδομένων αποφασίσαμε να εισάγουμε τον μηχανισμό της ΜΠΑΠ στην υλοποίηση του ZRP. (Όλα τα υπόλοιπα πειράματα περιέχονται στο παράρτημα.)

Όσον αφορά τη χρήση της μνήμης και το πλήθος των πακέτων ελέγχου που στέλνονται στο δίκτυο από καθένα από τα δύο πρωτόκολλα, παρόλο που δεν μπορούμε να συγκρίνουμε τα αποτελέσματα των πειραμάτων μας λόγω των δι-



Σχήμα 6.21: Μέση Καθυστέρηση Λήψης - Ρυθμός Κίνησης, για δισδιάστατα γραφήματα (DSR, ZRP, SRP)

αφορετικών παραμέτρων τους, ωστόσο μπορούμε να κάνουμε κάποιες διαπιστώσεις βασιζόμενοι στο τρόπο λειτουργίας τους. Έτσι και τα δύο πρωτόκολλα διατηρούν αρκετές δομές αλλά το ZRP διατηρεί περισσότερες, οι οποίες και χρησιμοποιούνται από τα επιμέρους πρωτόκολλα του. Τέλος η χρήση από το πρωτόκολλο ZRP των ΠΑΚΑ έχει ως αποτέλεσμα την αποστολή περισσότερων πακέτων ελέγχου σε σχέση με το DSR.

Συγκρίνοντας το πρωτόκολλο SRP με τα πρωτόκολλα DSR και ZRP παρατηρούμε ότι το ποσοστό των πακέτων δεδομένων που παρελήφθησαν στο SRP, σε κάθε είδος πειράματος, είναι υψηλότερο (π.χ. σχήμα 6.19). Αυτό οφείλεται στο ότι στο SRP οι σταθμοί της υποστήριξης κινούνται τυχαία, καλύπτοντας γρήγορα ολόκληρη την περιοχή του δικτύου και παραδίδοντας έτσι τα περισσότερα πακέτα δεδομένων στους προορισμούς τους όταν τους συναντήσουν. Η μη παράδοση ορισμένων, πολύ λίγων, πακέτων δεδομένων στους προορισμούς τους μπορεί να οφείλεται στους χαμηλούς ρυθμούς κίνησης των σταθμών που δεν ανήκουν στην υποστήριξη αλλά και στην περιοχή που καλύπτει το δίκτυο, η οποία μπορεί να είναι μεγάλη ενώ το πλήθος των σταθμών μικρό (αραιό δίκτυο). Στα πρωτόκολλα DSR και ZRP τα ποσοστά παράδοσης είναι πιο μικρά. Αυτό οφείλεται στο

ότι πολλές φορές δεν είναι δυνατή η επικοινωνία μεταξύ απομακρυσμένων σταθμών μέσω διαδρομών από κινητούς σταθμούς, ιδιαίτερα όταν τα δίκτυα είναι αραιά. Αλλά ακόμα και όταν ανακαλύπτονται τέτοιες διαδρομές η κίνηση των κόμβων οδηγεί στην γρήγορη διάσπασή τους. Στα πρωτόκολλα αυτά πακέτα δεδομένων ανταλλάσσονται συνήθως μεταξύ γειτονικών ή πολύ κοντινών σταθμών, αυτό άλλωστε υποδηλώνει και ο μικρός μέσος αριθμός βημάτων των πειραμάτων του DSR και του ZRP που παρουσιάσαμε παραπάνω. Έτσι πολύ λιγότερα πακέτα δεδομένων παραδίδονται στους προορισμούς τους. (Όλα τα υπόλοιπα πειράματα περιέχονται στο παράρτημα.)

Ακόμα παρατηρούμε ότι στο πρωτόκολλο SRP η μέση καθυστέρηση λήψης, σε κάθε είδος πειράματος, είναι πολύ πιο μεγάλη σε σχέση με αυτήν των πρωτοκόλλων DSR και ZRP (π.χ. σχήμα 6.21). Η μεγάλη καθυστέρηση λήψης, στο SRP, οφείλεται στις επιμέρους καθυστερήσεις που επισέρχονται πρώτον στο σταθμό πηγή μέχρι να συναντήσει ένα μέλος της υποστήριξης και κατά δεύτερον στους σταθμούς της υποστήριξης μέχρι κάποιος από αυτούς να συναντήσει τον προορισμό. Ακόμα ένας σταθμός της υποστήριξης δεν γνωρίζει που βρίσκεται ένας προορισμός και δεν κατευθύνεται προς αυτόν, απλώς κινείται τυχαία στην περιοχή του δικτύου γνωρίζοντας ότι κάποια στιγμή θα τον συναντήσει. Αντίθετα στα άλλα πρωτόκολλα οι καθυστερήσεις είναι σαφώς μικρότερες, αυτό οφείλεται κυρίως στο γεγονός ότι τα περισσότερα πακέτα δεδομένων μεταφέρονται μεταξύ γειτονικών ή πολύ κοντινών σταθμών με την χρήση σύντομων διαδρομών. Οι σταθμοί που μεταδίδουν ένα πακέτο δεδομένων γνωρίζουν πως αυτό μπορεί, ακολουθώντας μια διαδρομή, να φτάσει στον προορισμό του. Με το τρόπο αυτό ένα πακέτο δεδομένων δεν κινείται άσκοπα στο δίκτυο από σταθμό σε σταθμό, αλλά κατευθύνεται προς τον προορισμό του. (Όλα τα υπόλοιπα πειράματα περιέχονται στο παράρτημα.)

Όσον αφορά τη χρήση της μνήμης και το πλήθος των πακέτων ελέγχου που στάλθηκαν στο δίκτυο παρόλο που δεν πραγματοποιήσαμε ανάλογες μετρήσεις για το πρωτόκολλο SRP σε αντίθεση με τα άλλα πρωτόκολλα, ωστόσο μπορούμε να κάνουμε κάποιες παρατηρήσεις βασιζόμενοι στον τρόπο λειτουργίας τους. Στο πρωτόκολλο SRP οι σταθμοί (της υποστήριξης και μη) χρησιμοποιούν λιγότερη μνήμη και μεταδίδουν λιγότερα πακέτα ελέγχου σε σχέση με τους σταθμούς των πρωτοκόλλων DSR και ZRP. Στο πρωτόκολλο SRP οι σταθμοί που δεν ανήκουν στην υποστήριξη διατηρούν στην μνήμη τους μόνο πακέτα δεδομένων, μέχρι να συναντήσουν ένα μέλος της υποστήριξης. Οι σταθμοί της υποστήριξης ωστόσο πρέπει να έχουν μεγαλύτερες αποθηκευτικές δυνατότητες, τόσο για την διατήρηση των πακέτων δεδομένων που λαμβάνουν από τους άλλους σταθμούς (της υποστήριξης και μη) όσο και για τις επιβεβαιώσεις λήψης πακέτων δεδομένων από τους προορισμούς. Ακόμα τα πακέτα ελέγχου που μεταδίδονται είναι μόνο αυτά για τον συγχρονισμό των σταθμών της υποστήριξης και για την ανίχνευση των γειτονικών σταθμών. Τα πακέτα αυτά



μεταδίδονται μόνο μεταξύ γειτονικών σταθμών και δεν εξαπλώνονται στο δίκτυο. Από την άλλη μεριά οι σταθμοί που μετέχουν στα πρωτόκολλα DSR και ZRP χρησιμοποιούν περισσότερη μνήμη διατηρώντας διάφορες δομές π.χ. για την αποθήκευση διαδρομών, πακέτων δεδομένων, αιτήσεων ανακάλυψης διαδρομής κ.α. και μεταδίδουν περισσότερα πακέτα ελέγχου πολλά από τα οποία εξαπλώνονται στο δίκτυο π.χ. αιτήσεις ανακάλυψης διαδρομής.



## Κεφάλαιο 7

# Συνδυασμοί DSR και ZRP με το SRP

### 7.1 Συνδυασμοί

Στα πειράματα που διεξάγαμε με τα πρωτόκολλα DSR, ZRP και SRP παρατηρήσαμε ότι το πρωτόκολλο SRP έχει μεγαλύτερο ποσοστό παράδοσης και μεγαλύτερη μέση καθυστέρηση λήψης σε σχέση με τα άλλα δύο πρωτόκολλα. Ακόμα από τον τρόπο λειτουργίας των πρωτοκόλλων γίνεται εμφανές ότι το πρωτόκολλο SRP κάνει μικρότερη χρήση της μνήμης και στέλνει στο δίκτυο λιγότερα πακέτα ελέγχου. Έτσι σκεφτήκαμε να συνδυάσουμε τα πρωτόκολλα αυτά προσπαθώντας να μειώσουμε την μέση καθυστέρηση λήψης που παρουσιάζεται στο SRP, διατηρώντας ωστόσο το ποσοστό παράδοσης σε υψηλά επίπεδα και την χρήση της μνήμης και το πλήθος των πακέτων ελέγχου σε χαμηλά. Η μεγάλη καθυστέρηση λήψης στο SRP οφείλεται στις επιμέρους καθυστερήσεις που επισέρχονται πρώτον στο σταθμό πηγή μέχρι να συναντήσει ένα μέλος της υποστήριξης και κατά δεύτερον στους σταθμούς της υποστήριξης μέχρι κάποιος από αυτούς να συναντήσει τον προορισμό. Ακόμα ένας σταθμός της υποστήριξης δεν γνωρίζει που βρίσκεται ένας προορισμός και δεν κατευθύνεται προς αυτόν, απλώς κινείται τυχαία στην περιοχή του δικτύου γνωρίζοντας ότι κάποια στιγμή θα τον συναντήσει. Αυτό έχει σαν αποτέλεσμα ότι ακόμα και αν ένας προορισμός ενός πακέτου δεδομένων είναι προσβάσιμος και γειτονικός ή πολύ κοντινός η καθυστέρηση λήψης του πακέτου από τον προορισμό θα είναι δυσανάλογη της απόστασης (σε βήματα) της πηγής από αυτόν. Υλοποιήσαμε δύο συνδυασμούς των πρωτοκόλλων, το συνδυασμό του DSR με το SRP και το συνδυασμό του ZRP με το SRP. Στις υλοποιήσεις των συνδυασμών αυτών χρησιμοποιήσαμε και αλλάξαμε όπου χρειαζόνταν τις υλοποιήσεις των DSR, ZRP και SRP που περιγράψαμε παραπάνω. Η λογική και στους δύο αυτούς

συνδυασμούς είναι η ίδια. Δηλαδή ένα πακέτο δεδομένων μεταφέρεται στον προορισμό του με την χρήση του DSR ή του ZRP για κοντινούς και προσβάσιμους προορισμούς και σε κάθε άλλη περίπτωση χρησιμοποιείται το SRP.

Στον πρώτο συνδυασμό, του DSR με το SRP, για ένα πακέτο δεδομένων που δημιουργείται σε ένα σταθμό αρχικά χρησιμοποιείται το DSR πρωτόκολλο. Δηλαδή αν δεν υπάρχει μία διαδρομή προς τον προορισμό του πακέτου στην ΜΠΑΔ, τότε εκκινείται μία διαδικασία ΑΔ. Η αντίστοιχη ΑΙΔ έχει περιορισμένο εύρος μετάδοσης, χρησιμοποιώντας το πεδίο Μέγιστος Αριθμός Ενδιάμεσων Κόμβων. Το πεδίο αυτό χρησιμοποιείται για να περιοριστεί ο αριθμός των ενδιάμεσων κόμβων που επιτρέπεται να προωθήσουν μία συγκεκριμένη ΑΙΔ. Κάθε κόμβος που λαμβάνει ένα μήνυμα ΑΙΔ, με αυτό το πεδίο, μειώνει κατά ένα τον αριθμό στο πεδίο πριν προωθήσει το μήνυμα. Αν ο αριθμός στο πεδίο αυτό γίνει 0 πριν το μήνυμα φτάσει στον προορισμό ή σε κάποιον ενδιάμεσο σταθμό που ξέρει μια ΠΔ προς τον προορισμό τότε η αντίστοιχη ΑΙΔ απορρίπτεται. Ο σταθμός πηγής περιμένει για την αντίστοιχη ΑΠΔ για ένα προκαθορισμένο χρόνο διάστημα. Αν αυτό λήξει χωρίς να επιστρέψει η αντίστοιχη ΑΠΔ τότε το πακέτο δεδομένων αποθηκεύεται σε μία λίστα μέχρι αυτό να δοθεί σε ένα μέλος της υποστήριξης. Από εκεί και πέρα το πρωτόκολλο SRP αναλαμβάνει την παράδοση του πακέτου στον προορισμό του. Στην λίστα αυτή προστίθεται ένα πακέτο δεδομένων και σε μία ακόμα περίπτωση. Στην περίπτωση που βρεθεί μία διαδρομή προς έναν προορισμό και σταλλεί το αντίστοιχο πακέτο δεδομένων, αν δεν επιστρέψει μία επιβεβαίωση λήψης από τον επόμενο σταθμό της διαδρομής μέσα στο προκαθορισμένο χρονικό διάστημα, τότε πάλι το πακέτο αποθηκεύεται στην λίστα. Στην υλοποίηση αυτού του συνδυασμού αφαιρέσαμε τα κομμάτια κώδικα της υλοποίησης του πρωτοκόλλου DSR που δεν χρειαζόνταν.

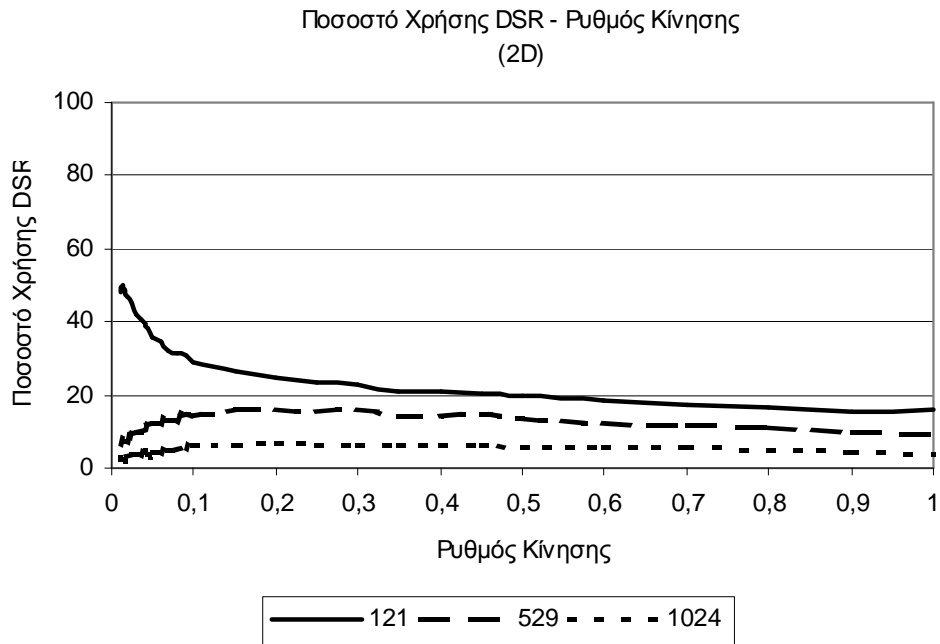
Στο δεύτερο συνδυασμό, του ZRP με το SRP, γίνεται χρήση της γνώσης της τοπολογίας της ΖΔ του κάθε σταθμού. Επομένως όταν δημιουργηθεί ένα νέο πακέτο δεδομένων ελέγχεται αν ο προορισμός του βρίσκεται μέσα στην ΖΔ του σταθμού πηγής. Στην περίπτωση που βρίσκεται, τότε το πακέτο δεδομένων αποστέλεται στον προορισμό του. Σε διαφορετική περίπτωση προστίθεται σε μία λίστα και δεν εκκινείται μία διαδικασία ΑΔ, όπως γίνεται στο ZRP. Από εκεί και πέρα το πρωτόκολλο SRP αναλαμβάνει την παράδοση του πακέτου στον προορισμό του. Στην υλοποίηση αυτού του συνδυασμού αφαιρέσαμε τα κομμάτια κώδικα της υλοποίησης του πρωτοκόλλου ZRP που δεν χρειαζόνταν, π.χ. την υλοποίηση του IERP και του BRP επιπέδου.

## 7.2 Παράμετροι και μετρούμενα μεγέθη πειραμάτων

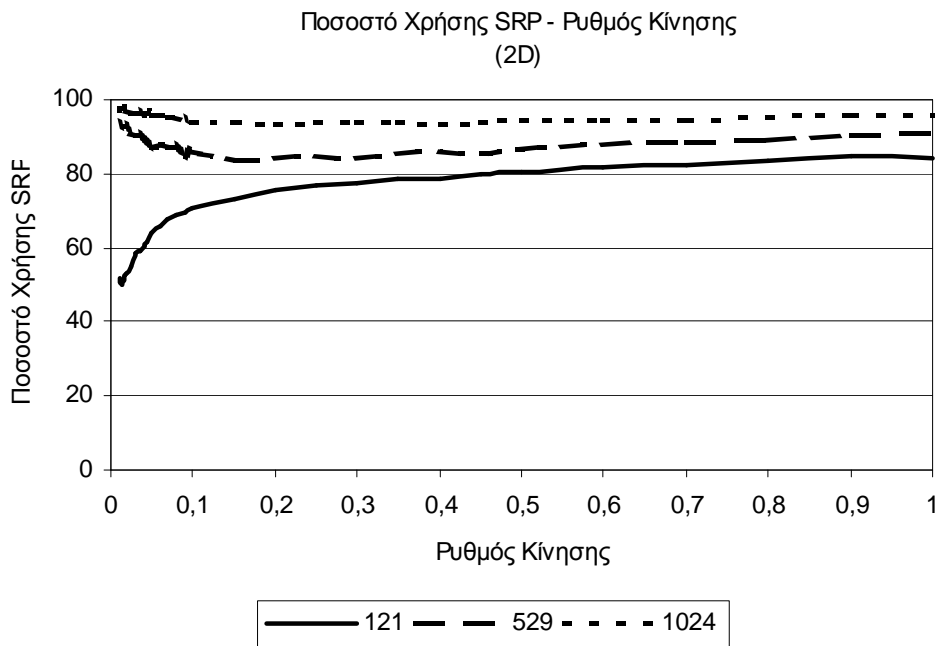
Τα πειράματα που εκτελέσαμε για τους συνδυασμούς των πρωτοκόλλων ήταν παρόμοια με αυτά που εκτελέσαμε για το DSR και το ZRP. Τα μετρούμενα μεγέθη ήταν το ποσοστό των πακέτων δεδομένων που παρελήφθησαν, η μέση καθυστέρηση λήψης και ορισμένα ακόμα ειδικά για τους συνδυασμούς. Συγκεκριμένα για κάθε συνδυασμό μετρήσαμε τα ποσοστά χρήσης των δύο πρωτοκόλλων, δηλαδή τα ποσοστά των πακέτων δεδομένων που στάλθηκαν με την χρήση του ενός (DSR, ZRP) και του άλλου πρωτοκόλλου (SRP), σε σχέση με το σύνολο των πακέτων που δημιουργούνται σε κάθε πείραμα. Ακόμα μετρήσαμε τα ποσοστά επιτυχούς χρήσης των δύο πρωτοκόλλων, δηλαδή τα ποσοστά των πακέτων δεδομένων που παρελήφθησαν με την χρήση του ενός (DSR, ZRP) και του άλλου πρωτοκόλλου (SRP), σε σχέση με αυτά που το κάθε πρωτόκολλο έστειλε σε κάθε πείραμα. Ο υπολογισμός του μέσου αριθμού βημάτων δεν έχει νόημα σε κανέναν από τους συνδυασμούς αφού χρησιμοποιείται το πρωτόκολλο SRP, το οποίο δεν στηρίζεται στην ανακάλυψη διαδρομών.

## 7.3 Πειράματα συνδυασμού DSR με το SRP

Στα πειράματα του συνδυασμού DSR με το SRP το πεδίο Μέγιστος Αριθμός Ενδιάμεσων Κόμβων κάθε ΑΙΔ είχε την τιμή 2. Επιλέξαμε αυτή την τιμή παρατηρώντας τα αποτελέσματα των πειραμάτων του DSR, όπου ο μέσος αριθμός βημάτων είναι μεταξύ των τιμών 1 και 4. Παράλληλα το προκαθορισμένο χρονικό διάστημα που ένας σταθμός περιμένει για μία ΑΠΔ τέθηκε ίσο με δύο φορές τον αριθμό του πεδίου Μέγιστος Αριθμός Ενδιάμεσων Κόμβων. Η αποστολή μίας ΑΙΔ και η λήψη της αντίστοιχης ΑΠΔ στην χειρότερη περίπτωση θα χρειαστούν τόσους γύρους για να ολοκληρωθούν, αφού σε κάθε γύρο ένα πακέτο (δεδομένων ή μη) κινείται κατά ένα βήμα. Με την ίδια λογική επιλέξαμε να είναι 2 το προκαθορισμένο χρονικό διάστημα που ο σταθμός πηγή περιμένει για μία άμεση επιβεβαίωση λήψης ενός πακέτου δεδομένων, από τον επόμενο σταθμό της διαδρομής. Τέλος ο αριθμός επαναλήψεων αποστολής μίας ΑΙΔ ή ενός πακέτου δεδομένων, μετά την λήξη του αντίστοιχου προκαθορισμένου χρονικού διαστήματος, τέθηκε ίσος με 1. Όσον αφορά το πρωτόκολλο SRP, του συνδυασμού, τα πλήθη των σταθμών υποστήριξης για κάθε πείραμα τέθηκαν στις ίδιες τιμές με αυτές των πειραμάτων για το γνήσιο SRP. Τέλος κάθε είδος πειράματος, δηλαδή με συγκεκριμένες παραμέτρους, επαναληφθεί τόσες φορές ώστε ο συνολικός αριθμός πακέτων δεδομένων που δημιουργήθηκαν, για το συγκεκριμένο είδος πειράματος, να είναι περίπου 50000.

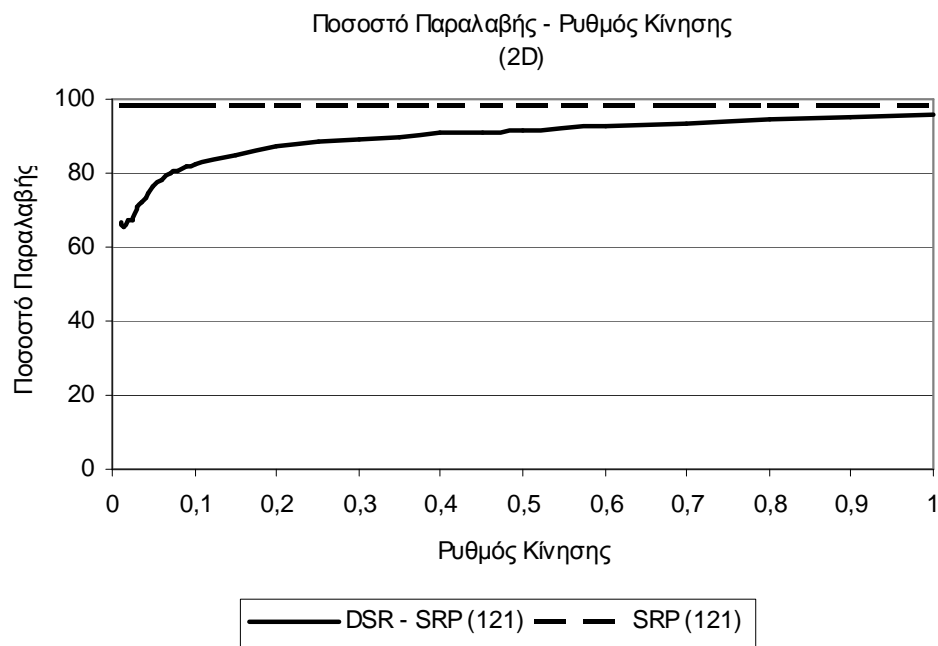


Σχήμα 7.1: Ποσοστό Χρήσης DSR - Ρυθμός Κίνησης, για δισδιάστατα γραφήματα (συνδυασμός DSR - SRP)

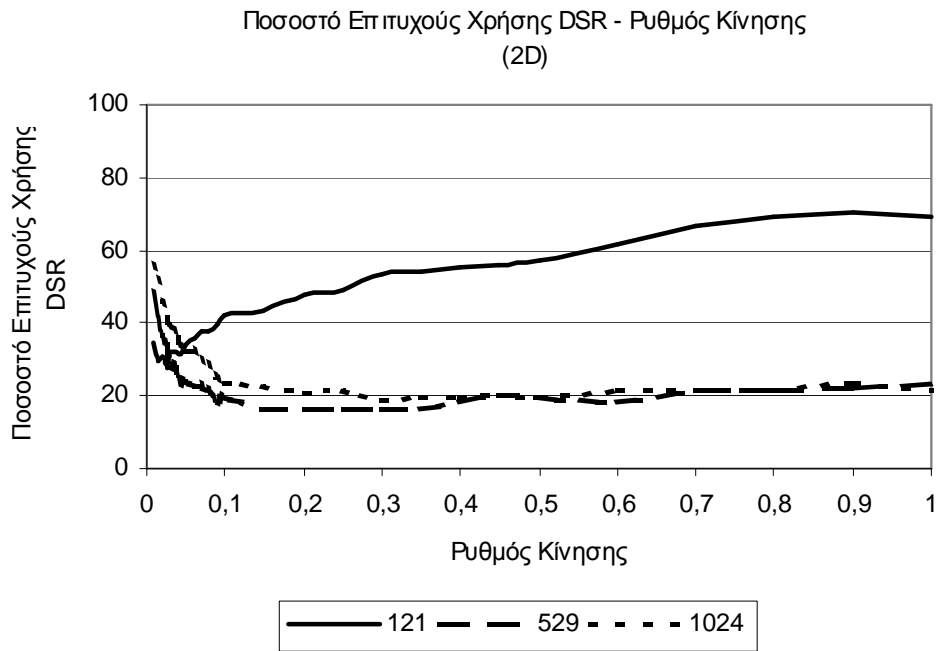


Σχήμα 7.2: Ποσοστό Χρήσης SRP - Ρυθμός Κίνησης, για δισδιάστατα γραφήματα (συνδυασμός DSR - SRP)

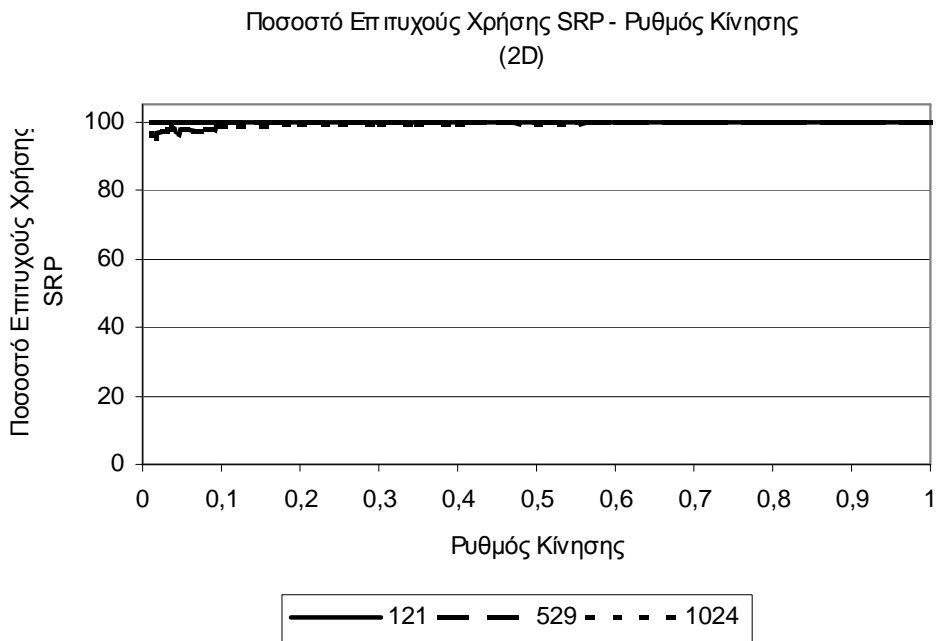
Το ποσοστό χρήσης του SRP εξαρτάται από το ποσοστό χρήσης του DSR, δηλαδή όσο περισσότερο χρησιμοποιείται το DSR για τη μεταφορά πακέτων δεδομένων τόσο λιγότερη είναι και η χρήση του SRP. Από τα πειράματα που διεξάγαμε με το συνδυασμό του DSR με το SRP παρατηρήσαμε ότι η χρήση του DSR αυξάνεται σε περιπτώσεις όπου διαδρομές προς τους διάφορους προορισμούς ανακαλύπτονται και χρησιμοποιούνται με μεγαλύτερη πιθανότητα (π.χ. σχήματα 7.1 και 7.2). Στα πειράματα του γνήσιου DSR παρατηρήσαμε και εξηγήσαμε πότε και γιατί αυτό συμβαίνει. Έτσι από τα πειράματα που διεξάγαμε με τον συνδυασμό παρατηρήσαμε ότι η χρήση του DSR είναι μεγαλύτερη στα γραφήματα 121 (125) κορυφών από ότι στα γραφήματα 529 και 1024 (512 και 1000) κορυφών. Ομοίως στα δισδιάστατα γραφήματα η χρήση του DSR είναι μεγαλύτερη από ότι στα τρισδιάστατα. Τέλος η χρήση του DSR αυξάνεται με την αύξηση του αριθμού των σταθμών, με την αύξηση του ρυθμού δημιουργίας πακέτων δεδομένων και με την μείωση του ρυθμού κίνησης. (Όλα τα υπόλοιπα πειράματα περιέχονται στο παράρτημα.)



Σχήμα 7.3: Ποσοστό Παραλαβής - Ρυθμός Κίνησης, για δισδιάστατα γραφήματα (συνδυασμός DSR - SRP, SRP)



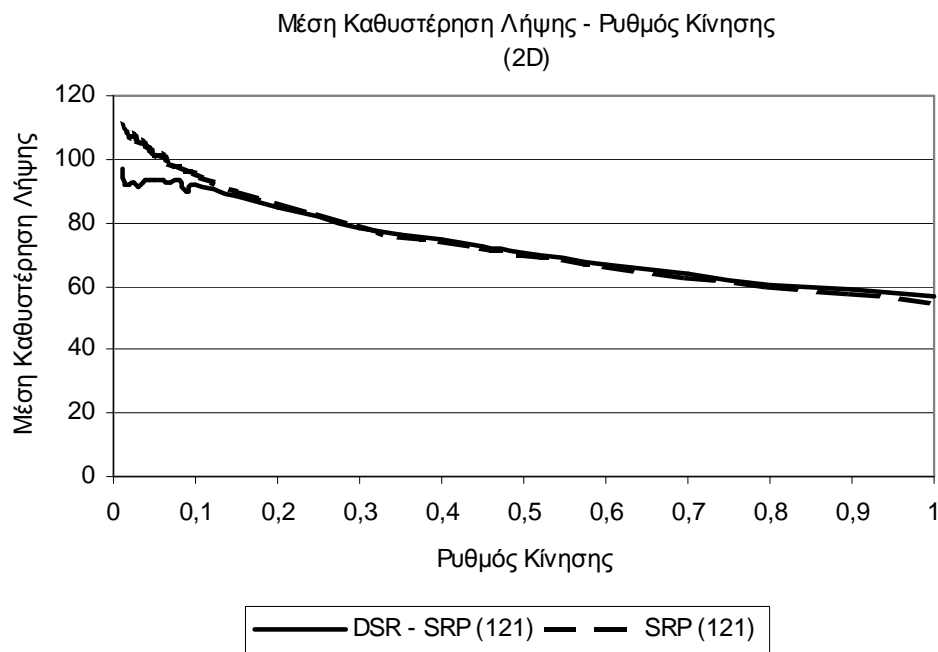
Σχήμα 7.4: Ποσοστό Επιτυχούς Χρήσης DSR - Ρυθμός Κίνησης, για δισδιάστατα γραφήματα (συνδυασμός DSR - SRP)



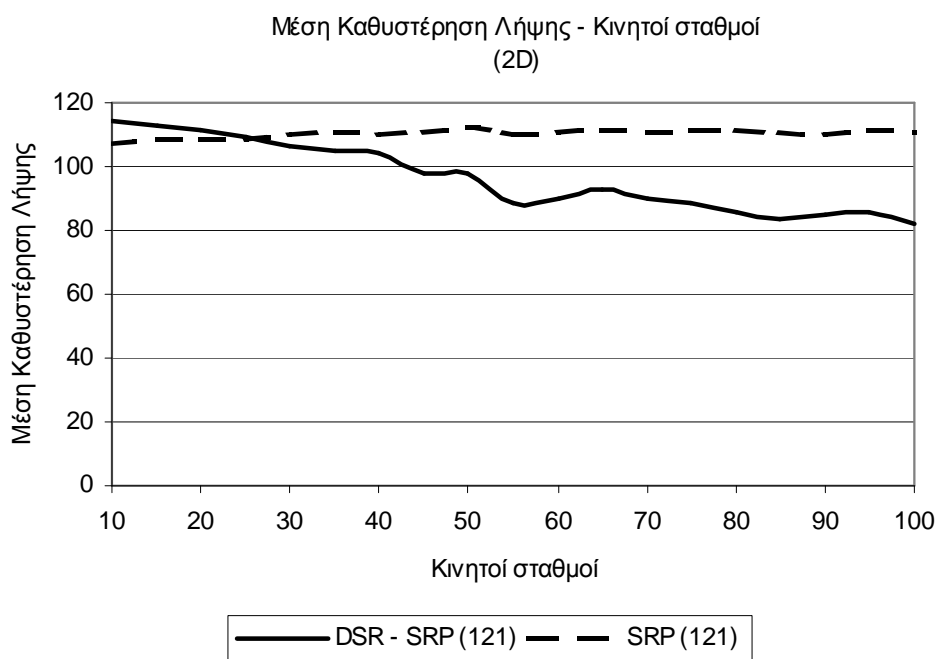
Σχήμα 7.5: Ποσοστό Επιτυχούς Χρήσης SRP - Ρυθμός Κίνησης, για δισδιάστατα γραφήματα (συνδυασμός DSR - SRP)



Το ποσοστό παράδοσης των πακέτων δεδομένων, του συνδυασμού, επηρεάζεται από τα ποσοστά χρήσης του DSR και του SRP (π.χ. σχήμα 7.3). Αυτό οφείλεται στο ότι τα δύο αυτά πρωτόκολλα έχουν διαφορετικά ποσοστά επιτυχούς χρήσης, άλλωστε και από τα πειράματα που διεξάγαμε ξεχωριστά για τα πρωτόκολλα αυτά παρατηρήσαμε σημαντικές διαφορές στα ποσοστά παράδοσης τους. Το ποσοστό επιτυχούς χρήσης του DSR είναι σχεδόν σε κάθε περίπτωση μικρότερο από αυτό του SRP (π.χ. σχήματα 7.4 και 7.5). Αυτό οφείλεται κυρίως στο ότι ο DSR ακόμα και αν ανακαλύψει μία διαδρομή προς έναν προορισμό αυτό δεν εξασφαλίζει την επιτυχή παράδοση ενός πακέτου δεδομένων, λόγω της πιθανής διάσπασης της διαδρομής, αντίθετα το SRP δεν στηρίζεται στην ανακάλυψη διαδρομών. Έτσι όσο αυξάνεται η χρήση του DSR τόσο μειώνεται το ποσοστό παράδοσης του συνδυασμού, μάλιστα σε όλα τα πειράματα το ποσοστό παράδοσης είναι μικρότερο από αυτό του γνήσιου SRP (π.χ. σχήμα 7.3). Από την άλλη μεριά όσο αυξάνεται η χρήση του SRP τόσο μεγαλώνει το ποσοστό παράδοσης του συνδυασμού, φτάνοντας μάλιστα αυτό του γνήσιου SRP (π.χ. σχήμα 7.3). Ακόμα σε κάθε περίπτωση το ποσοστό παράδοσης του συνδυασμού είναι μεγαλύτερο από αυτό του γνήσιου DSR. (Όλα τα υπόλοιπα πειράματα περιέχονται στο παράρτημα.)



Σχήμα 7.6: Μέση Καθυστέρηση Λήψης - Ρυθμός Κίνησης, για δισδιάστατα γραφήματα (συνδυασμός DSR - SRP, SRP)



Σχήμα 7.7: Μέση Καθυστέρηση Λήψης - Κινητοί Σταθμοί, για δισδιάστατα γραφήματα (συνδυασμός DSR - SRP, SRP)

Η μέση καθυστέρηση λήψης, του συνδυασμού, επηρεάζεται και αυτή από τα ποσοστά χρήσης του DSR και του SRP. Όταν το ποσοστό χρήσης του DSR είναι μεγάλο τότε η μέση καθυστέρηση λήψης είναι μικρότερη σε σχέση με αυτή του γνήσιου SRP (π.χ. σχήματα 7.6 και 7.7). Αυτό οφείλεται στο ότι με την χρήση του DSR πολλά πακέτα δεδομένων μεταφέρονται στον προορισμό τους, ο οποίος είναι γειτονικός ή πολύ κοντινός, πολύ γρηγορότερα σε σχέση με τον γνήσιο SRP. Σε άλλες περιπτώσεις η μείωση της μέσης καθυστέρησης λήψης είναι μικρή και ορισμένες φορές η μέση καθυστέρηση είναι μεγαλύτερη ή ίση με αυτή του γνήσιου SRP (π.χ. σχήματα 7.6 και 7.7). Αυτό μπορεί να οφείλεται στη μειωμένη χρήση του DSR ή στο χαμηλό ποσοστό επιτυχούς χρήσης του, όταν π.χ. υπάρχει αυξημένη κινητικότητα των σταθμών. Αυτό έχει σαν αποτέλεσμα λίγα πακέτα δεδομένων να φτάνουν στον προορισμό τους, των οποίων και η μικρή καθυστέρηση προσμετράται στον υπολογισμό της μέσης καθυστέρησης λήψης. Ακόμα η μέση καθυστέρηση λήψης για τα πακέτα που παραδίδονται μέσω του SRP, του συνδυασμού, είναι μεγαλύτερη σε σχέση με αυτή στο γνήσιο SRP. Αυτό οφείλεται στο ότι ένα πακέτο δεδομένων πριν δωθεί στο SRP καθυστερεί εκτελώντας τις διαδικασίες του DSR, όπου και αποτυγχάνει. Έτσι όταν η χρήση του SRP είναι μεγάλη τότε η μέση καθυστέρηση

λήψης του συνδυασμού επηρεάζεται αναλόγως (π.χ. σχήματα 7.6 και 7.7). Σε αυτό βέβαια πρέπει να λάβουμε υπόψη μας ότι η μέση καθυστέρηση λήψης στο γνήσιο SRP είναι αρκετά μεγαλύτερη σε σχέση με την καθυστέρηση που εισάγει αρχικά ο DSR στο SRP του συνδυασμού. Γενικά η μέση καθυστέρηση λήψης του συνδυασμού είναι μικρότερη ή ίση από τη μέση καθυστέρηση του γνήσιου SRP. (Όλα τα υπόλοιπα πειράματα περιέχονται στο παράρτημα.)

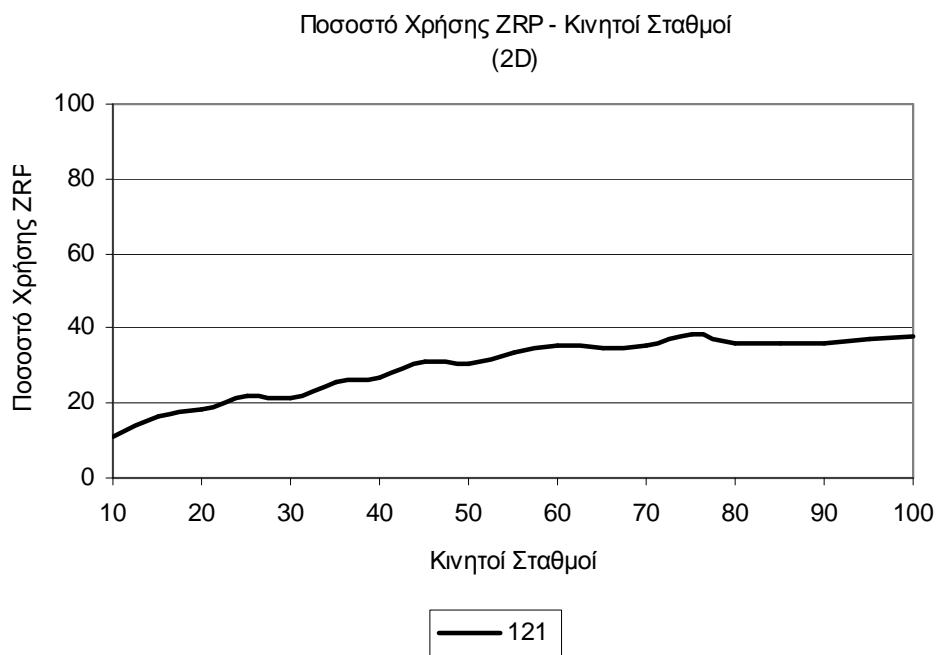
Όσον αφορά τη χρήση της μνήμης και το πλήθος των πακέτων ελέγχου που στέλνονται στο δίκτυο παρόλο που δεν πραγματοποιήσαμε ανάλογες μετρήσεις για το συνδυασμό του DSR με το SRP, ωστόσο μπορούμε να κάνουμε κάποιες παρατηρήσεις βασιζόμενοι στον τρόπο λειτουργίας του. Έτσι ο συνδυασμός κάνει μεγαλύτερη χρήση της μνήμης και στέλνει περισσότερα πακέτα ελέγχου στο δίκτυο σε σχέση με το γνήσιο SRP, εξαιτίας της χρήσης του DSR.

Καταλήγοντας για το συνδυασμό του DSR με το SRP μπορούμε να πούμε ότι δεν υπερτερεί έναντι του γνήσιου SRP. Ο συνδυασμός έχει μικρότερο ποσοστό παράδοσης, κάνει μεγαλύτερη χρήση της μνήμης και στέλνει περισσότερα πακέτα ελέγχου στο δίκτυο. Τέλος η μέση καθυστέρηση λήψης είναι μικρότερη ή ίση από την μέση καθυστέρηση του γνήσιου SRP.

## 7.4 Πειράματα συνδυασμού ZRP με το SRP

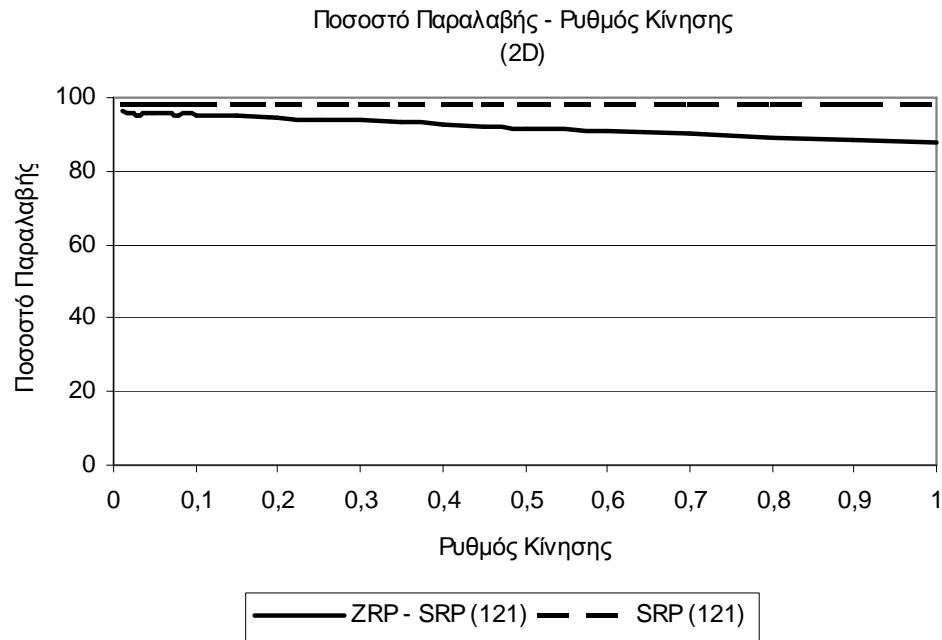
Για το συνδυασμό του ZRP με το SRP πραγματοποιήσαμε πειράματα μόνο με δισδιάστατα γραφήματα 121 κορυφών, ενώ κάθε σταθμός σε κάθε πείραμα δημιουργούσε τουλάχιστον 2500 πακέτα δεδομένων. Επιλέξαμε να πραγματοποιήσουμε πειράματα μόνο με γραφήματα 121 κορυφών επειδή σε αυτά τα διάφορα πρωτόκολλα, όπως έγινε φανερό από προηγούμενα πειράματα, παρουσιάζουν καλύτερη συμπεριφορά σε σχέση με τα άλλα γραφήματα (529 και 1024 κορυφών). Αυτό οφείλεται στο ότι τα δίκτυα που αντιστοιχούν στα γραφήματα 121 κορυφών είναι πιο πυκνά. Στα πειράματα του συνδυασμού ZRP με το SRP επιλέξαμε, για το ZRP, η ΖΔ κάθε σταθμού να έχει ακτίνα 3, βασιζόμενοι στην [12]. Όσον αφορά το πρωτόκολλο SRP, του συνδυασμού, τα πλήθη των σταθμών υποστήριξης για κάθε πείραμα τέθηκαν στις ίδιες τιμές με αυτές των πειραμάτων για το γνήσιο SRP. Τέλος κάθε είδος πειράματος, δηλαδή με συγκεκριμένες παραμέτρους, επαναληφθεί τόσες φορές ώστε ο συνολικός αριθμός πακέτων δεδομένων που δημιουργήθηκαν, για το συγκεκριμένο είδος πειράματος, να είναι τουλάχιστον 25000.

Το ποσοστό χρήσης του SRP εξαρτάται από το ποσοστό χρήσης του ZRP, δηλαδή όσο περισσότερο χρησιμοποιείται το ZRP για τη μεταφορά πακέτων δεδομένων τόσο λιγότερη είναι και η χρήση του SRP. Το ποσοστό χρήσης του ZRP σε κάθε περίπτωση είναι μικρότερο από το 40 τις εκατό των πακέτων δεδομένων που δημιουργήθηκαν. Θα πρέπει γενικώς να λάβουμε υπόψη μας

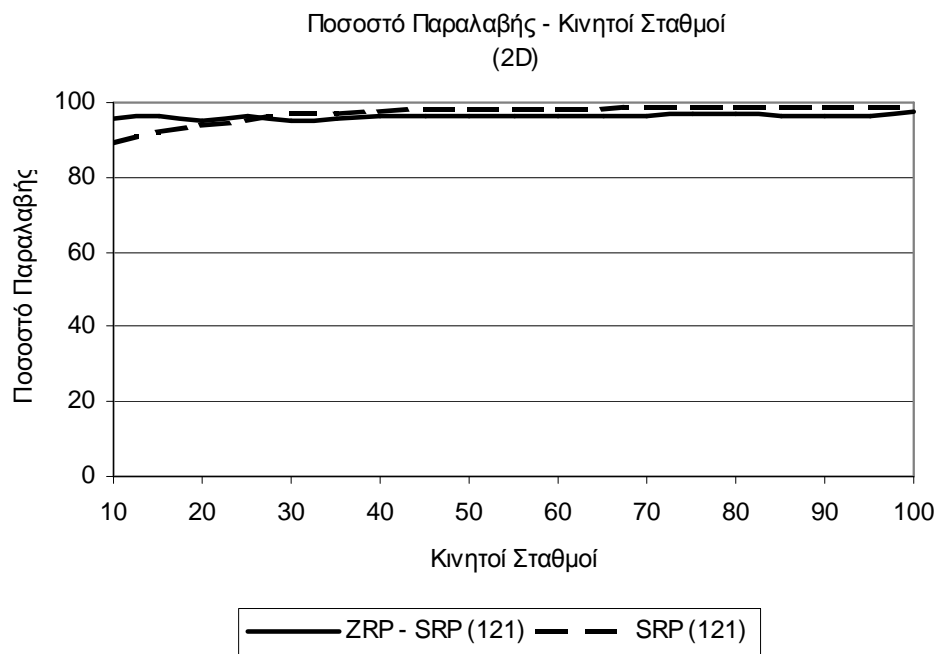


Σχήμα 7.8: Ποσοστό Χρήσης ZRP - Κινητοί Σταθμοί, για δισδιάστατα γραφήματα (συνδυασμός ZRP - SRP)

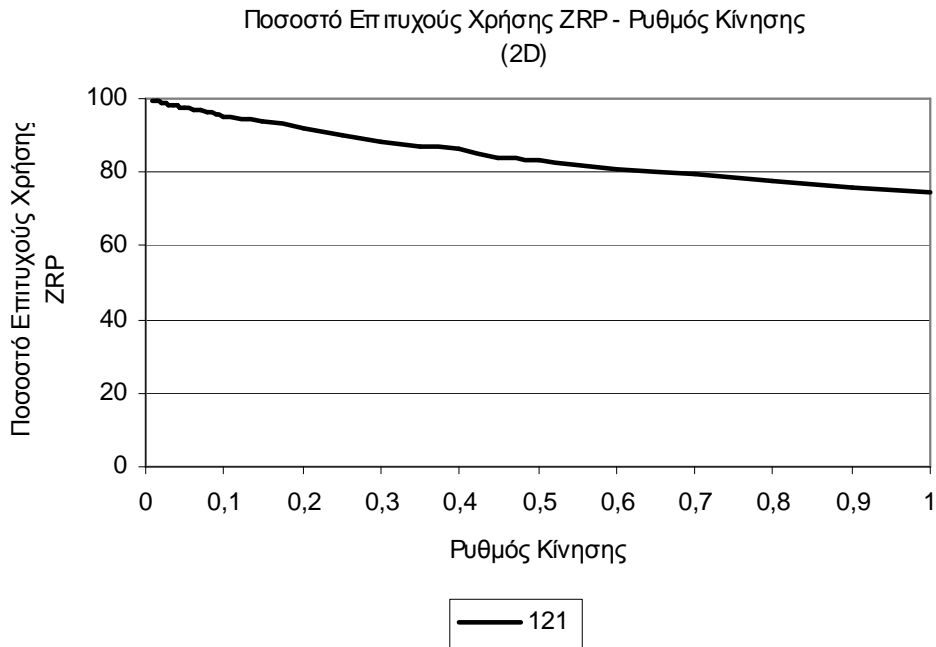
ότι το ZRP του συνδυασμού διατηρεί σε σχέση με το γνήσιο ZRP μόνο τη διαδικασία της συντήρησης της γνώσης της τοπολογίας της ΖΔ κάθε σταθμού και όχι τη διαδικασία ανακάλυψης διαδρομής. Η χρήση του ZRP επηρεάζεται κυρίως από την αύξηση του αριθμού των κινητών σταθμών, που δεν ανήκουν στην υποστήριξη (σχήμα 7.8). Με το τρόπο αυτό το δίκτυο γίνεται πιο πυκνό, με αποτέλεσμα περισσότεροι πιθανοί προορισμοί να ανήκουν στη ΖΔ ενός σταθμού. Ακόμα η χρήση του ZRP αυξάνεται ελαφρώς με την αύξηση του ρυθμού κίνησης, μιας και ένας σταθμός λαμβάνει ΠΑΚΑ από περισσότερους σταθμούς, ανακαλύπτοντας έτσι περισσότερους πιθανούς προορισμούς. Τέλος η αύξηση του ρυθμού δημιουργίας νέων πακέτων δεδομένων δεν επηρεάζει το ποσοστό χρήσης του ZRP, γεγονός που οφείλεται κυρίως στο τρόπο λειτουργίας του ZRP του συνδυασμού αλλά και στο χαμηλό (προκαθορισμένο) ρυθμό κίνησης που δεν δημιουργεί συχνές αλλαγές στη τοπολογία της ΖΔ ενός σταθμού. (Όλα τα υπόλοιπα πειράματα περιέχονται στο παράρτημα.)



Σχήμα 7.9: Ποσοστό Παραλαβής - Ρυθμός Κίνησης, για διδιάστατα γραφήματα (συνδυασμός ZRP - SRP, SRP)

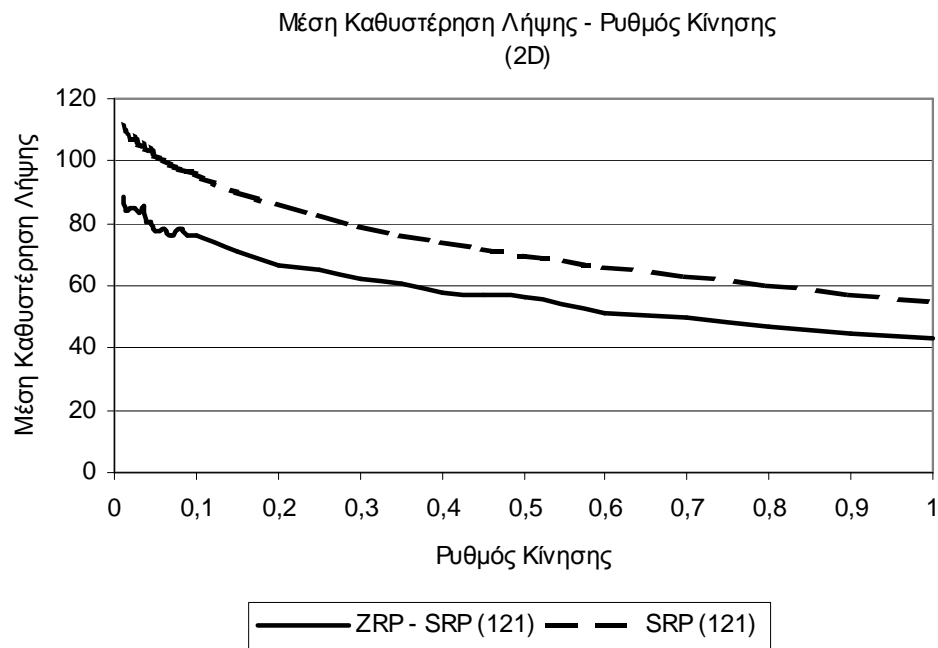


Σχήμα 7.10: Ποσοστό Παραλαβής - Κινητοί Σταθμοί, για διδιάστατα γραφήματα (συνδυασμός ZRP - SRP, SRP)



Σχήμα 7.11: Ποσοστό Επιτυχούς Χρήσης ZRP - Ρυθμός Κίνησης, για διαστάσιμα γραφήματα (συνδυασμός ZRP - SRP)

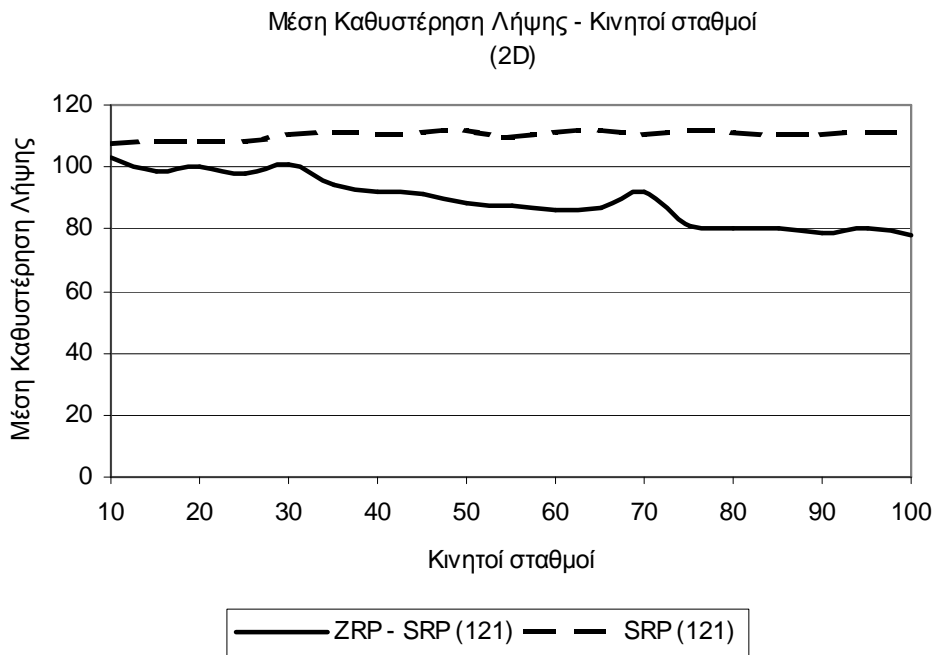
Το ποσοστό παράδοσης των πακέτων δεδομένων, του συνδυασμού του ZRP με το SRP, είναι υψηλότερο από αυτό του προηγούμενου συνδυασμού και μάλιστα παραπλήσιο, σε κάθε περίπτωση, με αυτό του γνήσιου SRP (π.χ. σχήματα 7.9 και 7.10). Αυτό οφείλεται στο ότι το ZRP του συνδυασμού έχει μεγάλα ποσοστά επιτυχούς χρήσης, μιας και χρησιμοποιείται μόνο η γνώση της τοπολογίας της ΖΔ ενός σταθμού και δεν εκκινείται ποτέ η διαδικασία ανακάλυψης διαδρομής. Η γνώση της τοπολογίας της ΖΔ ενός σταθμού ανανεώνεται σε κάθε αλλαγή, με αποτέλεσμα αυτή να είναι συνήθως έγκυρη όταν πρόκειται να χρησιμοποιηθεί για τη μεταφορά ενός πακέτου δεδομένων. Επιπλέον το γεγονός ότι κάθε ενδιαμέσος σταθμός ελέγχει και μπορεί να διορθώσει την πληροφορία δρομολόγησης που μεταφέρει ένα πακέτο δεδομένων, αυξάνει την πιθανότητα παράδοσης του στον προορισμό του. Το ποσοστό επιτυχούς χρήσης του ZRP επηρεάζεται μόνο από την αύξηση του ρυθμού κίνησης, μιας και η κίνηση των σταθμών μπορεί να προκαλέσει διασπάσεις των διαφόρων διαδρομών (σχήμα 7.11), παρόλα αυτά παραμένει υψηλό. Σε όλες τις άλλες περιπτώσεις το ποσοστό αυτό διατηρείται σταθερό και πολύ υψηλό. Τέλος το ποσοστό επιτυχούς χρήσης του SRP είναι σε όλες τις περιπτώσεις σταθερό και πολύ υψηλό. (Όλα τα υπόλοιπα πειράματα περιέχονται στο παράρτημα.)



Σχήμα 7.12: Μέση Καθυστέρηση Λήψης - Ρυθμός Κίνησης, για δισδιάστατα γραφήματα (συνδυασμός ZRP - SRP, SRP)

Η μέση καθυστέρηση λήψης, είναι μικρότερη σε σχέση με αυτή του γνήσιου SRP (π.χ. σχήματα 7.12 και 7.13). Αυτό οφείλεται καταρχάς στη χρήση του ZRP για τη μεταφορά πακέτων δεδομένων σε προορισμούς που βρίσκονται μέσα στη ΖΔ του σταθμού πηγή. Με το τρόπο αυτό ένα πακέτο δεδομένων παραδίδεται κατευθείαν στον προορισμό του και δεν χρειάζεται να ταξιδεύει στο δίκτυο μέχρι ένα μέλος της υποστήριξης να συναντήσει τον προορισμό. Επιπλέον η χρήση του ZRP, αρχικά, δεν επιφέρει καμμία καθυστέρηση στα πακέτα δεδομένων που παραδίδονται εντέλει μέσω του SRP. Όταν δημιουργηθεί ένα νέο πακέτο δεδομένων αρχικά ελέγχεται αν ο προορισμός του βρίσκεται μέσα στην ΖΔ του σταθμού πηγή, στην περίπτωση που δεν βρίσκεται τότε για το πακέτο αυτό, άμεσα και χωρίς καμμία καθυστέρηση, ακολουθείται η διαδικασία του SRP. Τέλος η μέση καθυστέρηση λήψης του συνδυασμού σε κάθε περίπτωση είναι μεγαλύτερη από τις μέσες καθυστερήσεις λήψης στα γνήσια πρωτόκολλα DSR και ZRP. (Όλα τα υπόλοιπα πειράματα περιέχονται στο παράρτημα.)

Όσον αφορά τη χρήση της μνήμης και το πλήθος των πακέτων ελέγχου που στέλνονται στο δίκτυο παρόλο που δεν πραγματοποιήσαμε ανάλογες μετρήσεις για το συνδυασμό του ZRP με το SRP, ωστόσο μπορούμε να κάνουμε κάποιες παρατηρήσεις βασιζόμενοι στο τρόπο λειτουργίας του. Έτσι ο συνδυασμός



Σχήμα 7.13: Μέση Καθυστέρηση Λήψης - Κινητοί Σταθμοί, για δισδιάστατα γραφήματα (συνδυασμός ZRP - SRP, SRP)

κάνει μεγαλύτερη χρήση της μνήμης και στέλνει περισσότερα πακέτα ελέγχου στο δίκτυο σε σχέση με το γνήσιο SRP, εξαιτίας της χρήσης του ZRP (π.χ. ΠΙΚΑ, ΠΑΚΑ).

Καταλήγοντας για το συνδυασμό του ZRP με το SRP μπορούμε να πούμε ότι αποτελεί μία πολύ καλή πρόταση. Ο συνδυασμός έχει στις περισσότερες περιπτώσεις παραπλήσιο ποσοστό παράδοσης με αυτό του γνήσιου SRP, ενώ η μέση καθυστέρηση λήψης είναι πάντα μικρότερη. Παρόλα αυτά ο συνδυασμός κάνει μεγαλύτερη χρήση της μνήμης και στέλνει περισσότερα πακέτα ελέγχου στο δίκτυο.



# Κεφάλαιο 8

## Συμπεράσματα

Στη διπλωματική αρχικά κάναμε μία γενική παρουσίαση των πρωτοκόλλων δρομολόγησης για ad-hoc δίκτυα. Αναλυτικά εξετάσαμε δύο πρωτόκολλα δρομολόγησης, το DSR και το ZRP, τα οποία και υλοποιήσαμε. Με τα πρωτόκολλα αυτά καθώς και με το SRP πραγματοποιήσαμε πειράματα. Μέσω αυτών αποκτήσαμε γνώση τόσο για τα πρωτόκολλα που στηρίζονται στην εύρεση διαδρομών από σταθμούς όσο και για τα πρωτόκολλα που στηρίζονται στη κίνηση των σταθμών. Συγκρίνοντας μάλιστα τα αποτελέσματα των πειραμάτων μας διαπιστώσαμε τα πλεονεκτήματα και τα μειονεκτήματα των παραπάνω πρωτοκόλλων αλλά και των αντίστοιχων κατηγοριών τους. Τέλος υλοποιώντας και εκτελώντας πειράματα με συνδυασμούς των τριών αυτών πρωτοκόλλων εξαγάγαμε ορισμένα ιδιαίτερα ενδιαφέροντα συμπεράσματα.

Τα πρωτόκολλα DSR και ZRP στηρίζονται στην εύρεση διαδρομών από σταθμούς. Τα χαρακτηριστικά των πρωτοκόλλων αυτών (ποσοστό παράδοσης, μέσος αριθμός βημάτων, μέση καθυστέρηση λήψης) επιρεάζονται με τον ίδιο τρόπο από τα διάφορα μεταβαλλόμενα μεγέθη (ρυθμός κίνησης και δημιουργίας πακέτων δεδομένων, πλήθος κινητών σταθμών), ωστόσο οι τιμές αυτών των χαρακτηριστικών τους διαφέρουν. Το ποσοστό παράδοσης, ο μέσος αριθμός βημάτων και η μέση καθυστέρηση λήψης στο πρωτόκολλο ZRP είναι σε κάθε περίπτωση μεγαλύτερα από αυτά στο DSR. Επιπλέον το πρωτόκολλο ZRP κάνει μεγαλύτερη χρήση της μνήμης και στέλνει περισσότερα πακέτα ελέγχου στο δίκτυο. Τέλος πρέπει να αναφέρουμε ότι το ZRP είναι πιο πολύπλοκο στη λειτουργία και στην υλοποίηση του, αλλά και πιο αργό στην εκτέλεση του από το DSR.

Το πρωτόκολλο SRP στηρίζεται στη κίνηση των σταθμών. Συγκρίνοντας το με τα πρωτόκολλα DSR και ZRP διαπιστώσαμε ότι το ποσοστό παράδοσης του είναι πολύ μεγαλύτερο σε σχέση με το ποσοστό αυτών των δύο πρωτοκόλλων. Μάλιστα το ποσοστό παράδοσης του SRP πλησιάζει, σε πολλές περιπτώσεις, το 100%. Παρόλα αυτά στο πρωτόκολλο SRP η μέση καθυστέρηση λήψης-

ς, σε κάθε είδος πειράματος, είναι πολύ πιο μεγάλη σε σχέση με αυτήν των πρωτοκόλλων DSR και ZRP. Τέλος η χρήση της μνήμης και το πλήθος των πακέτων ελέγχου που στάλθηκαν στο δίκτυο είναι μικρότερα στο SRP.

Θέλοντας να μειώσουμε τη μέση καθυστέρηση λήψης που παρουσιάζεται στο SRP διατηρώντας ωστόσο το ποσοστό παράδοσης σε υψηλά επίπεδα και τη χρήση της μνήμης και το πλήθος των πακέτων ελέγχου σε χαμηλά, σκεφτήκαμε να συνδυάσουμε το SRP με το DSR και το ZRP. Έτσι υλοποιήσαμε δύο συνδυασμούς, το συνδυασμό του DSR με το SRP και το συνδυασμό του ZRP με το SRP. Η λογική και στους δύο αυτούς συνδυασμούς ήταν η ίδια. Δηλαδή ένα πακέτο δεδομένων μεταφέρεται στον προορισμό του με την χρήση του DSR ή του ZRP για κοντινούς και προσβάσιμους προορισμούς και σε κάθε άλλη περίπτωση χρησιμοποιείται το SRP. Από τα πειράματα με το συνδυασμό του DSR με το SRP διαπιστώσαμε ότι αυτός δεν υπερτερεί έναντι του γνήσιου SRP. Ο συνδυασμός έχει μικρότερο ποσοστό παράδοσης, κάνει μεγαλύτερη χρήση της μνήμης και στέλνει περισσότερα πακέτα ελέγχου στο δίκτυο. Ακόμα η μέση καθυστέρηση λήψης είναι μικρότερη ή ίση από τη μέση καθυστέρηση του γνήσιου SRP. Από τα πειράματα με το συνδυασμό του ZRP με το SRP διαπιστώσαμε ότι αυτός αποτελεί μία πολύ καλή πρόταση. Ο συνδυασμός αυτός έχει στις περισσότερες περιπτώσεις παραπλήσιο ποσοστό παράδοσης με αυτό του γνήσιου SRP, ενώ η μέση καθυστέρηση λήψης είναι πάντα μικρότερη. Παρόλα αυτά ο συνδυασμός κάνει μεγαλύτερη χρήση της μνήμης και στέλνει περισσότερα πακέτα ελέγχου στο δίκτυο.

## Κεφάλαιο 9

### Μελλοντική εργασία

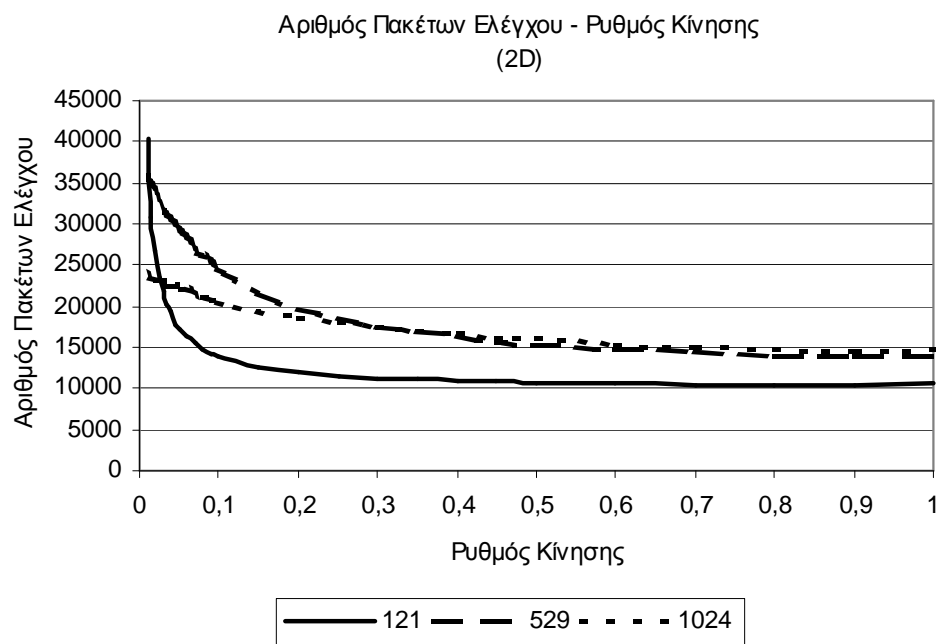
Τα ιδιαίτερα ενδιαφέροντα συμπεράσματα στα οποία καταλήξαμε στη διπλωματική και ειδικά αυτά που αφορούν τους συνδυασμούς του SRP με τα πρωτόκολλα DSR και ZRP αποτελούν μία πολύ καλή αφορμή για τη μελέτη παρόμοιων συνδυασμών. Τέλος ο συνδυασμός του SRP με το ZRP ίσως να αξίζει να μελετηθεί λίγο περισσότερο, κάνοντας και άλλα πειράματα ή ακόμα και ορισμένες αλλαγές. Μία τέτοια πιθανή αλλαγή θα ήταν ο κάθε σταθμός υποστήριξης να μετέδιδε σε ένα σταθμό προορισμού όχι μόνο τα πακέτα δεδομένων που τον αφορούν αλλά και τα πακέτα δεδομένων που αφορούν σταθμούς που βρίσκονται μέσα στη ΖΔ του.



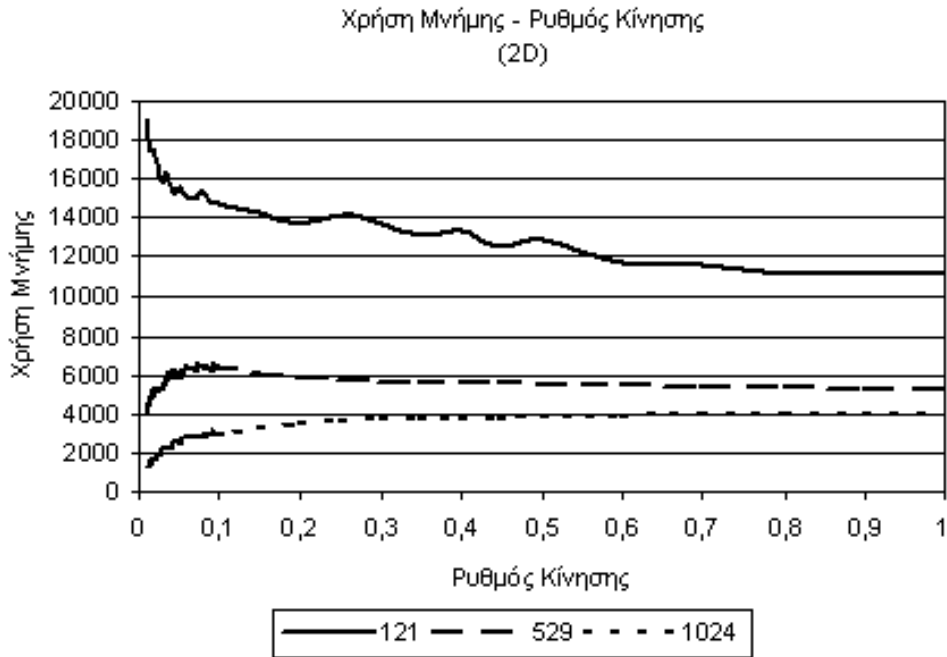
# Παράρτημα Α΄

## Γραφικές παραστάσεις για το DSR

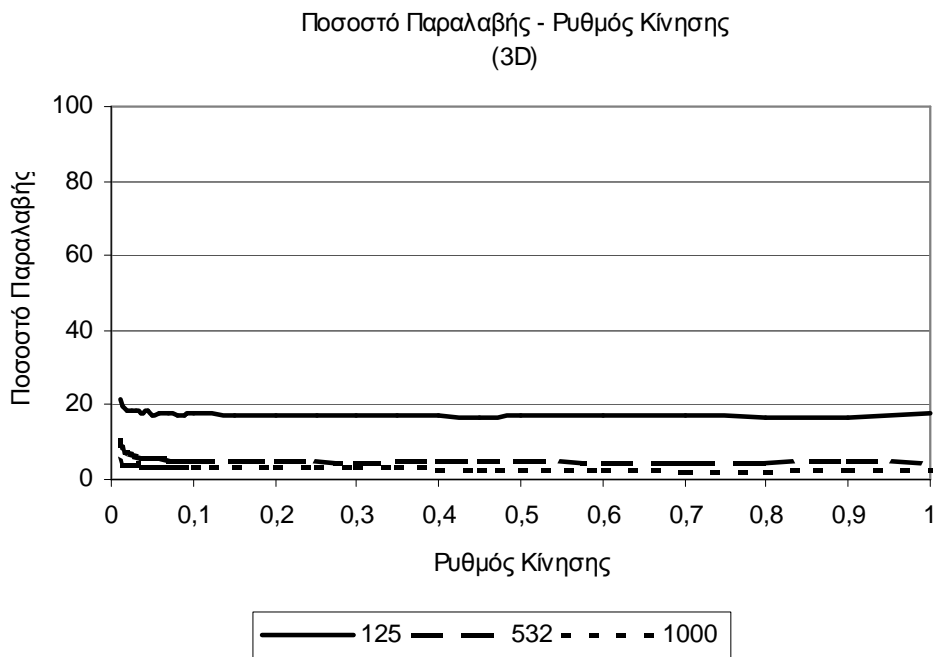
### Α΄.1 Ρυθμός κίνησης



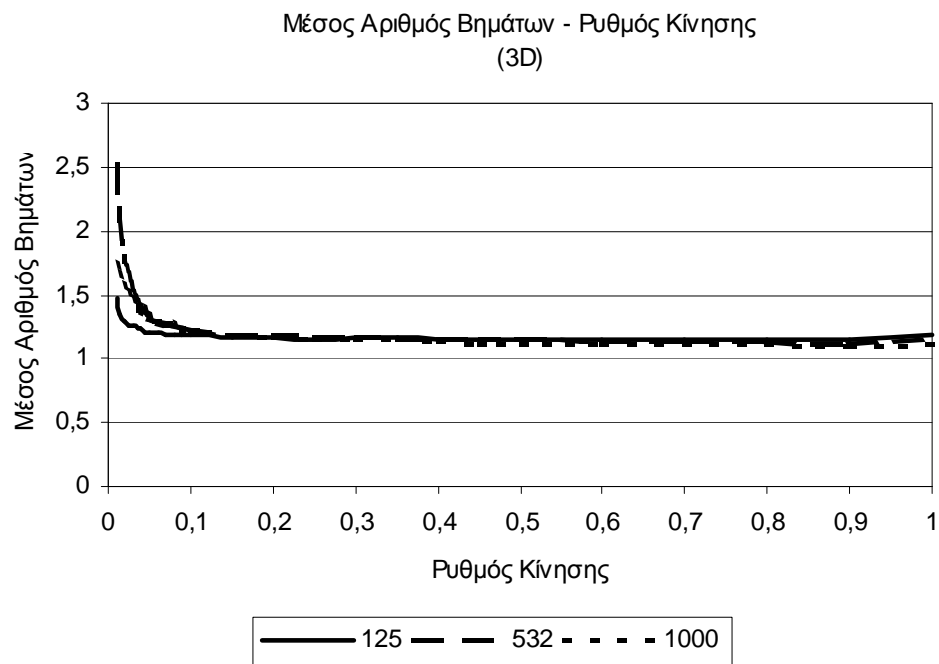
Σχήμα Α΄.1: Αριθμός Πακέτων Ελέγχου - Ρυθμός Κίνησης, για διαστάσιμα γραφήματα (DSR)



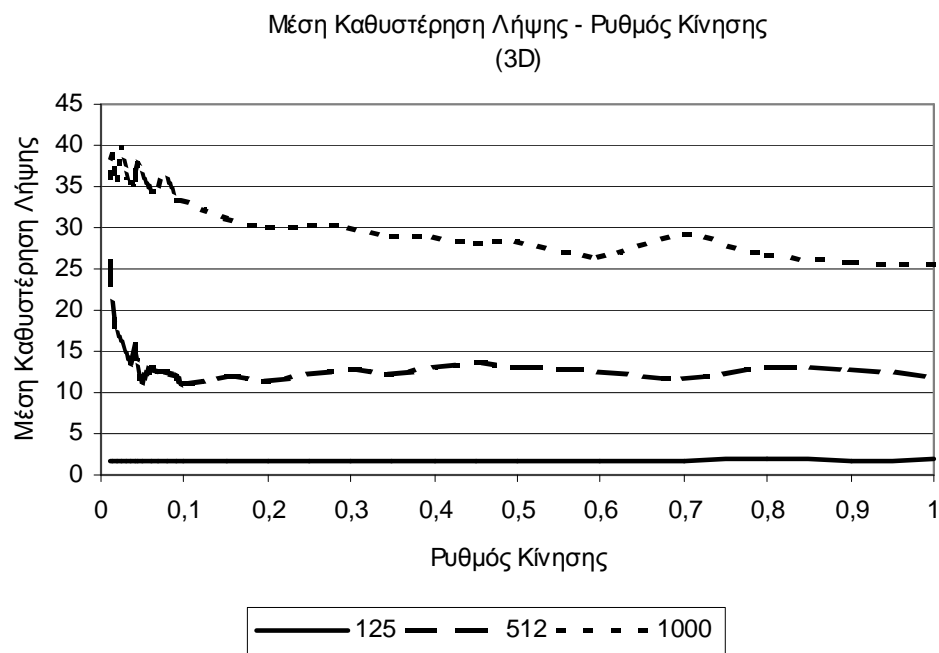
Σχήμα Α'.2: Χρήση Μνήμης - Ρυθμός Κίνησης, για δισδιάστατα γραφήματα (DSR)



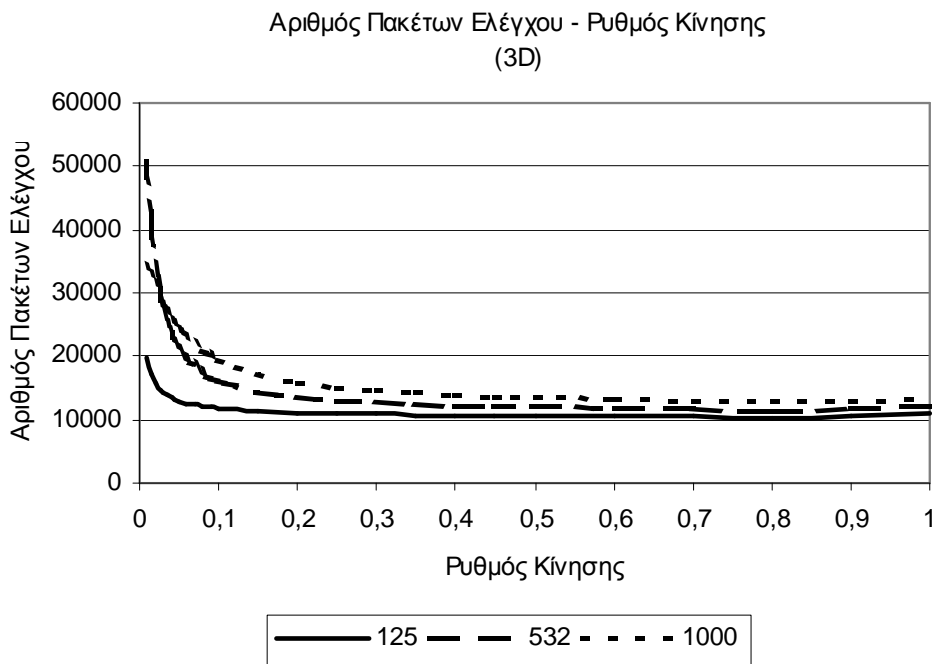
Σχήμα Α'.3: Ποσοστό Παραλαβής - Ρυθμός Κίνησης, για τρισδιάστατα γραφήματα (DSR)



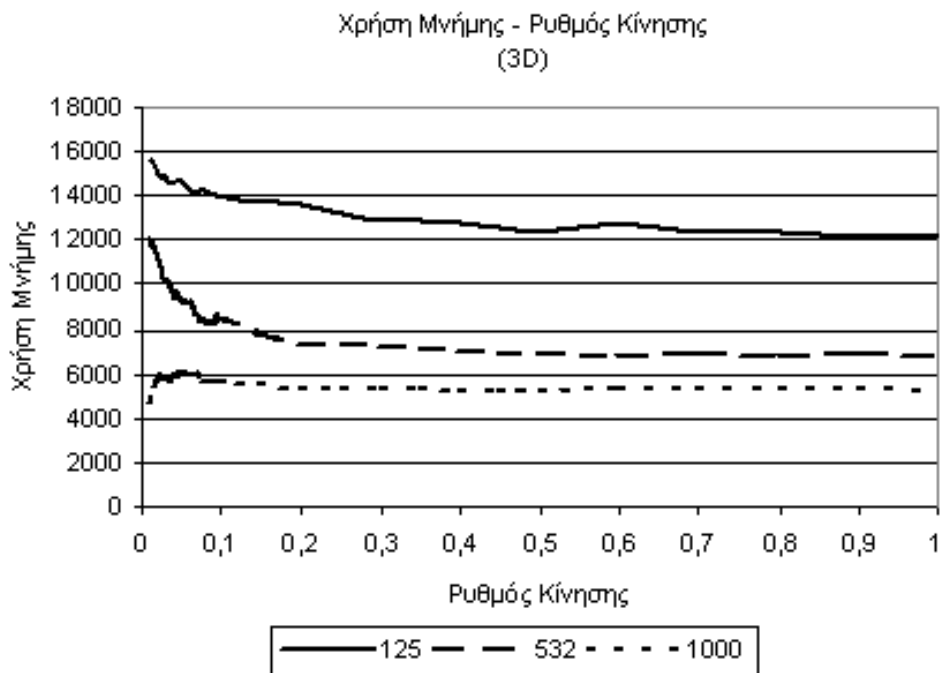
Σχήμα Α'.4: Μέσος Αριθμός Βημάτων - Ρυθμός Κίνησης, για τρισδιάστατα γραφήματα (DSR)



Σχήμα Α'.5: Μέση Καθυστερήση Λήψης - Ρυθμός Κίνησης, για τρισδιάστατα γραφήματα (DSR)



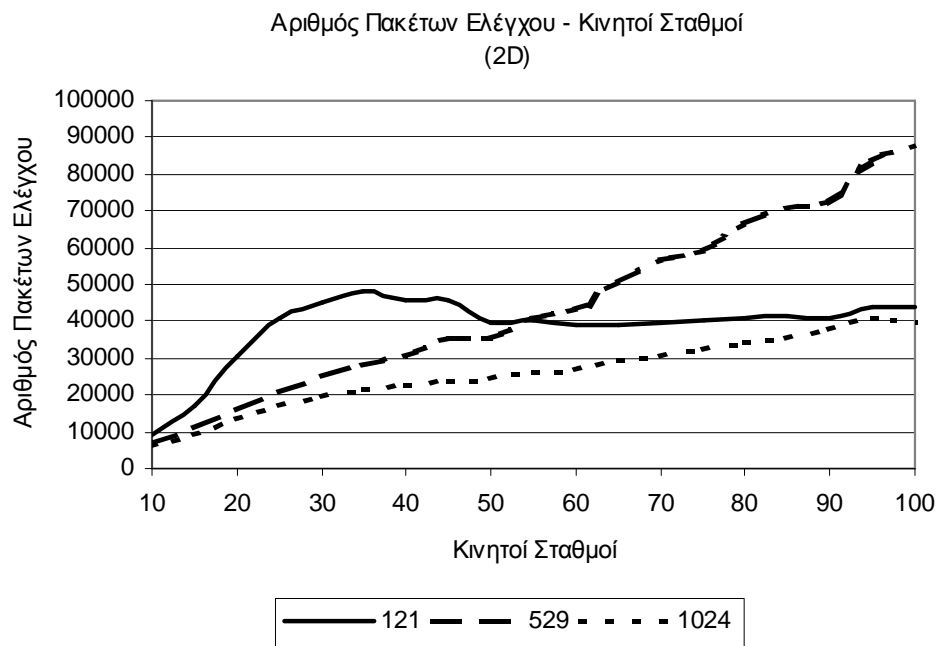
Σχήμα Α'6: Αριθμός Πακέτων Ελέγχου - Ρυθμός Κίνησης, για τρισδιάστατα γραφήματα (DSR)



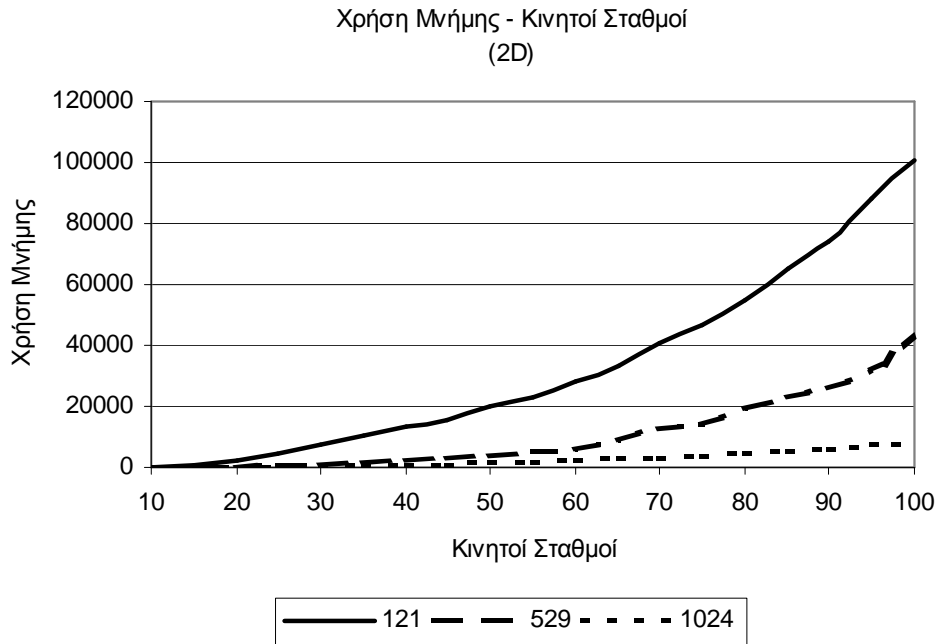
Σχήμα Α'7: Χρήση Μνήμης - Ρυθμός Κίνησης, για τρισδιάστατα γραφήματα (DSR)



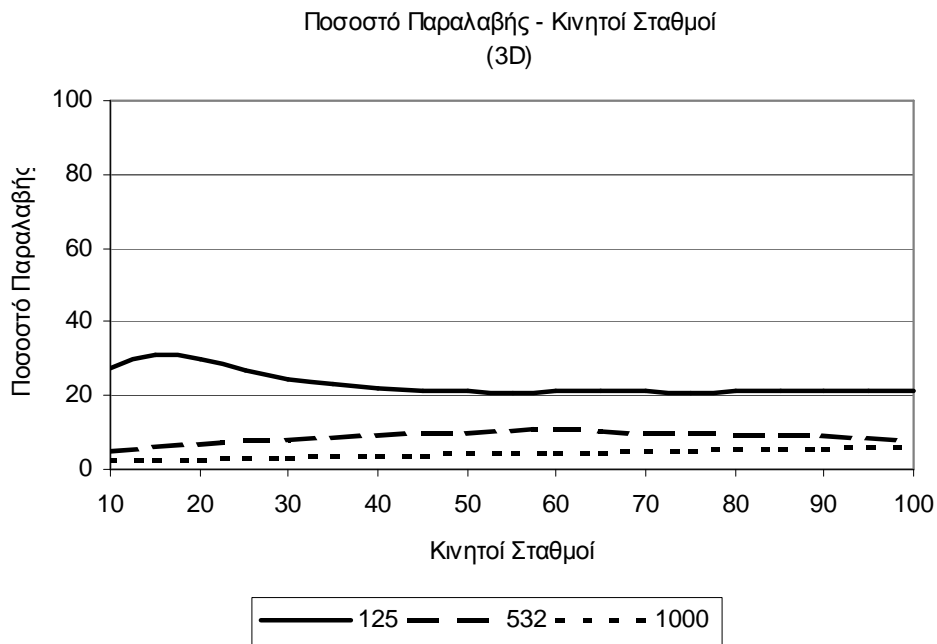
## Α'.2 Κινητοί σταθμοί



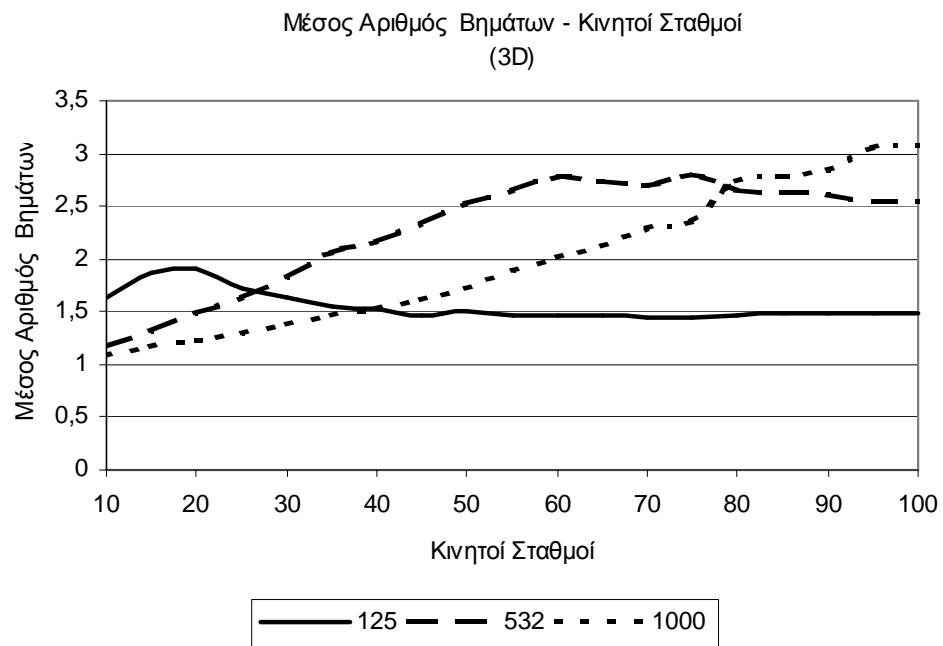
Σχήμα Α'.8: Αριθμός Πακέτων Ελέγχου - Κινητοί Σταθμοί, για διαστάτα γραφήματα (DSR)



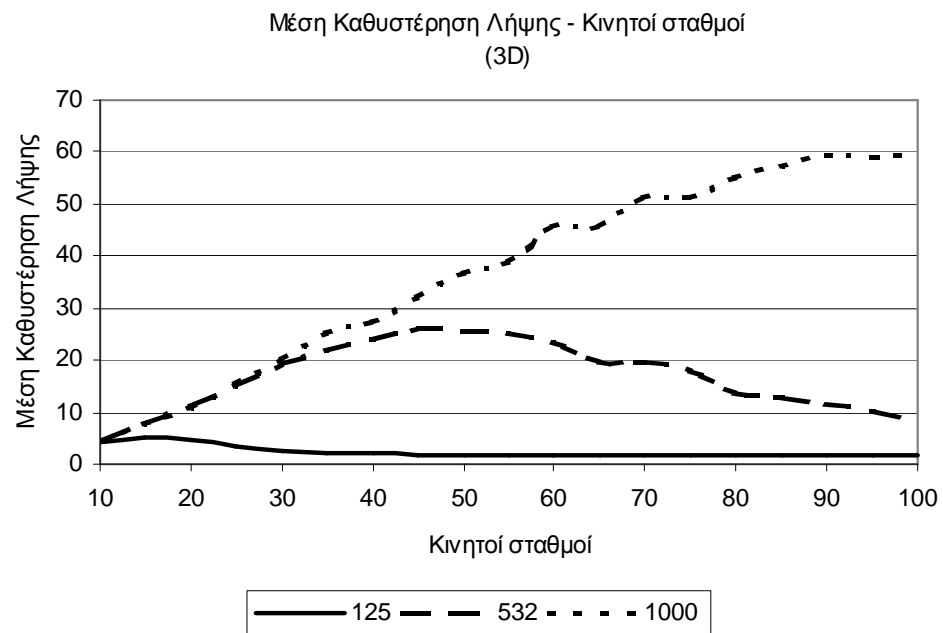
Σχήμα Α'.9: Χρήση Μνήμης - Κινητοί Σταθμοί, για δισδιάστατα γραφήματα (DSR)



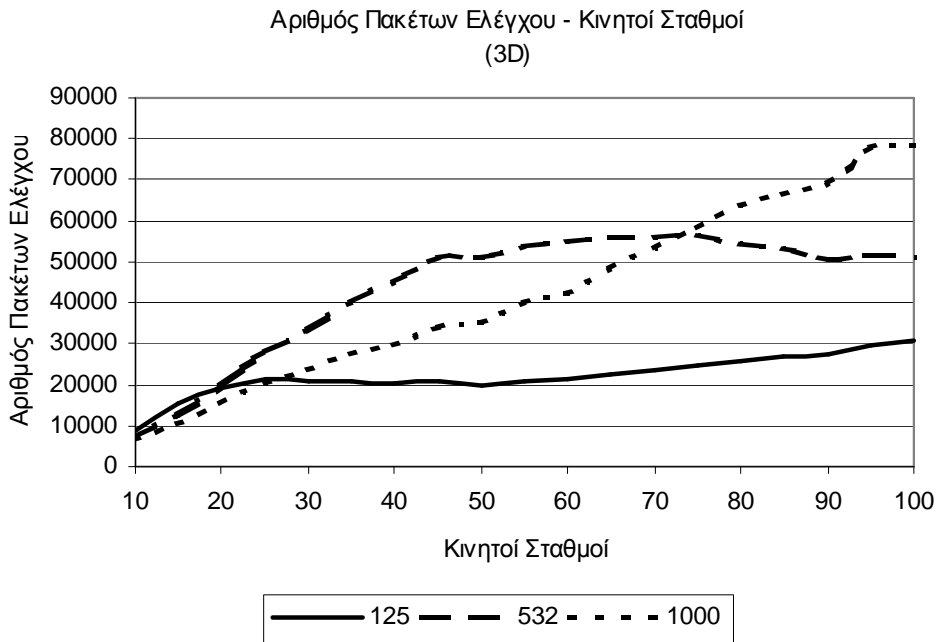
Σχήμα Α'.10: Ποσοστό Παραλαβής - Κινητοί Σταθμοί, για τρισδιάστατα γραφήματα (DSR)



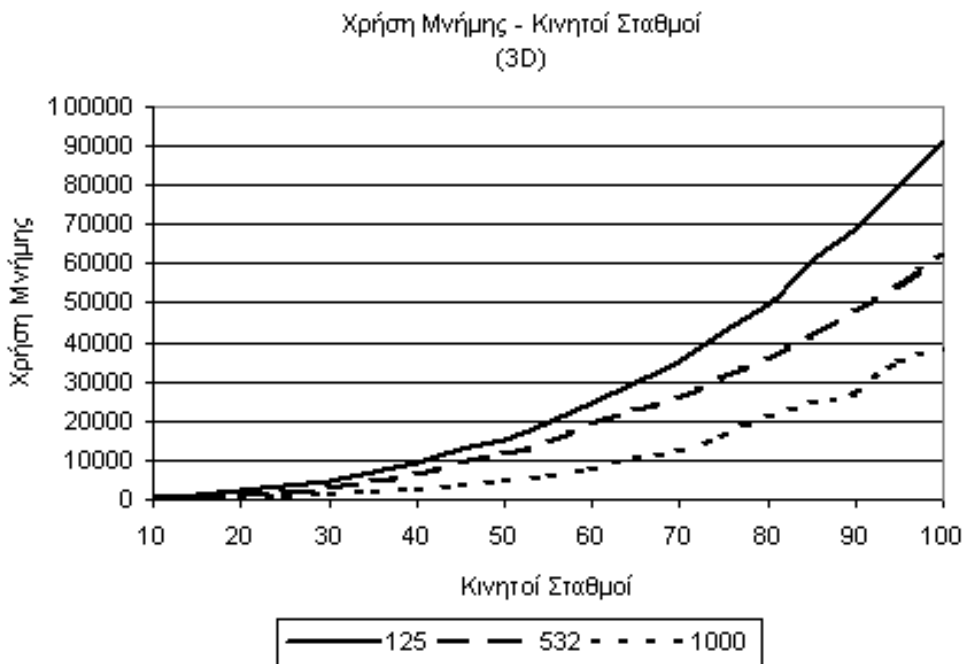
Σχήμα Α'.11: Μέσος Αριθμός Βημάτων - Κινητοί Σταθμοί, για τρισδιάστατα γραφήματα (DSR)



Σχήμα Α'.12: Μέση Καθυστέρηση Λήψης - Κινητοί Σταθμοί, για τρισδιάστατα γραφήματα (DSR)

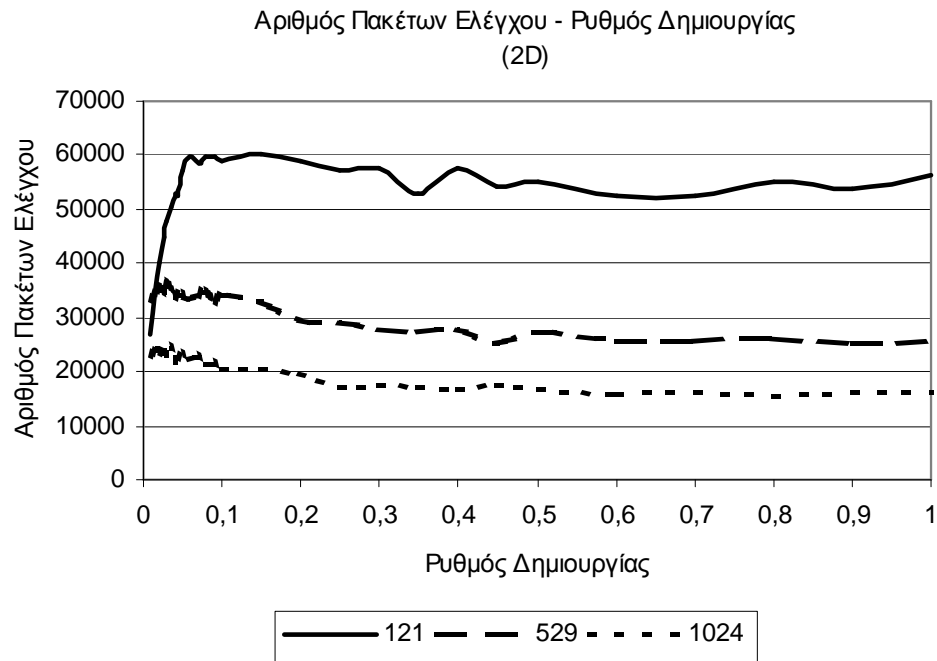


Σχήμα Α'.13: Αριθμός Πακέτων Ελέγχου - Κινητοί Σταθμοί, για τρισδιάστατα γραφήματα (DSR)

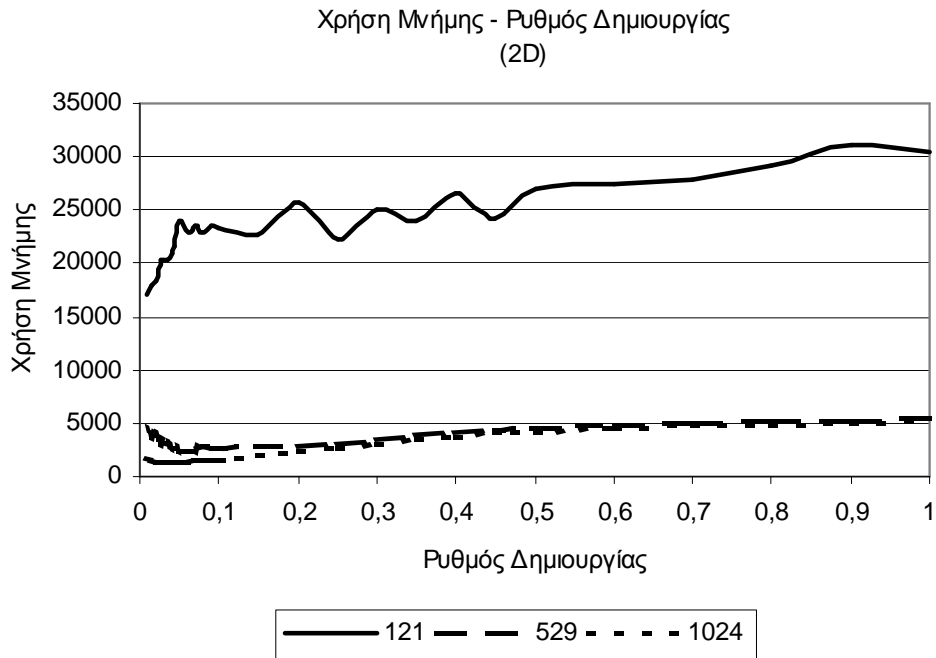


Σχήμα Α'.14: Χρήση Μνήμης - Κινητοί Σταθμοί, για τρισδιάστατα γραφήματα (DSR)

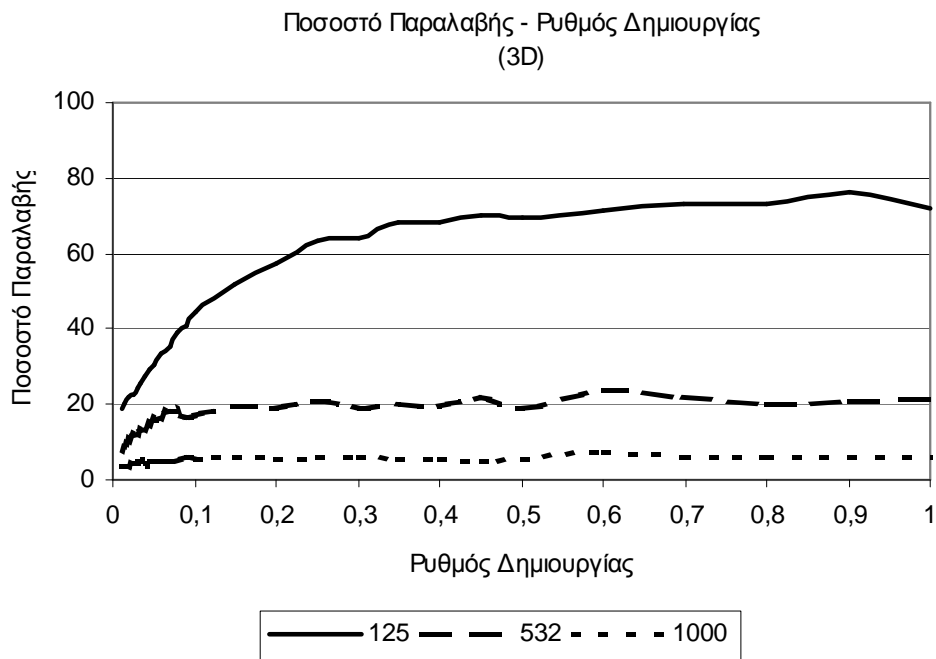
## Α'.3 Ρυθμός δημιουργίας πακέτων δεδομένων



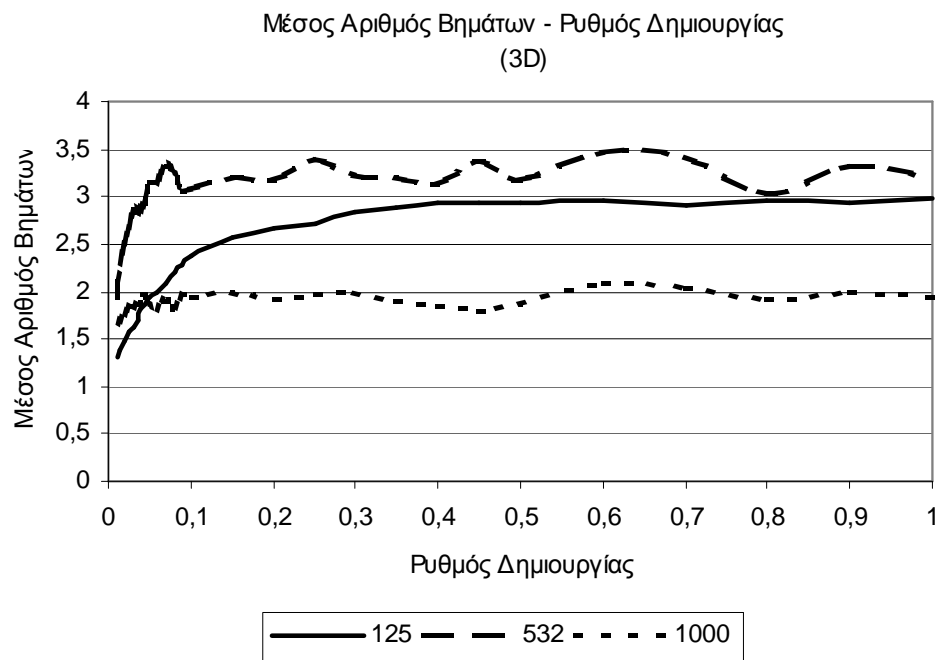
Σχήμα Α'.15: Αριθμός Πακέτων Ελέγχου - Ρυθμός Δημιουργίας Πακέτων Δεδομένων, για δισδιάστατα γραφήματα (DSR)



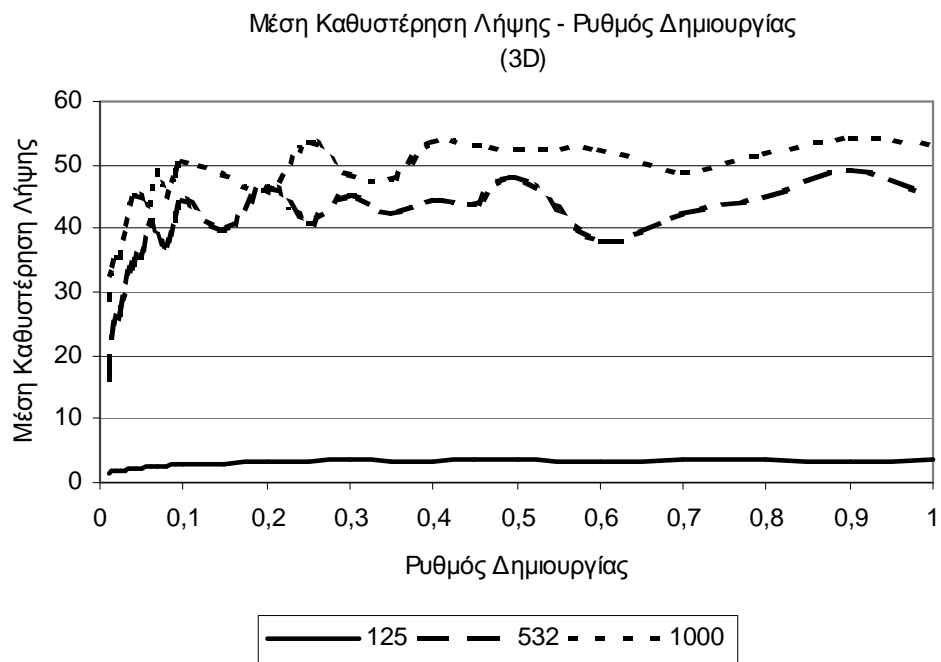
Σχήμα Α'.16: Χρήση Μνήμης - Ρυθμός Δημιουργίας Πακέτων Δεδομένων, για δισδιάστατα γραφήματα (DSR)



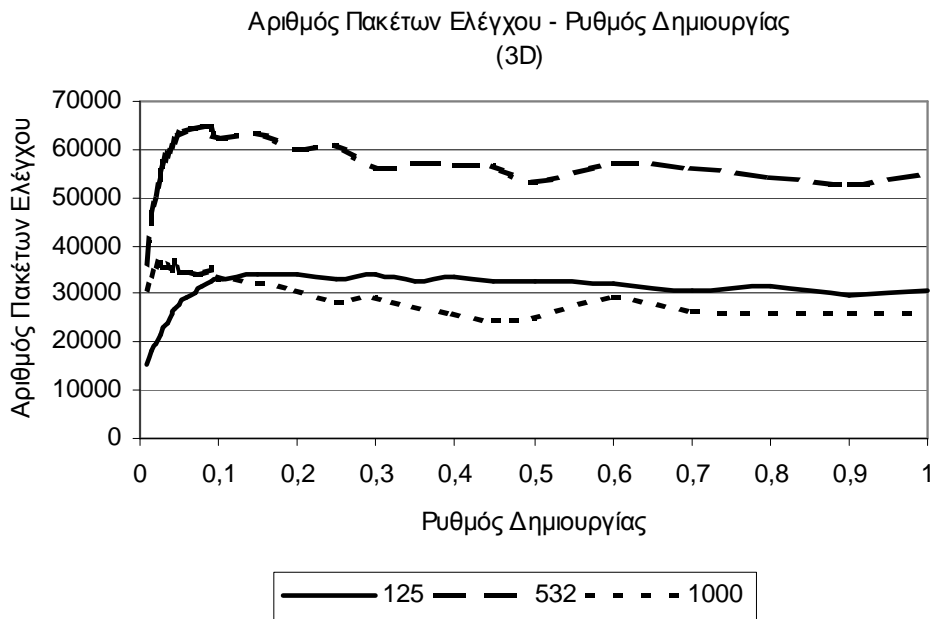
Σχήμα Α'.17: Ποσοστό Παραλαβής - Ρυθμός Δημιουργίας Πακέτων Δεδομένων, για τρισδιάστατα γραφήματα (DSR)



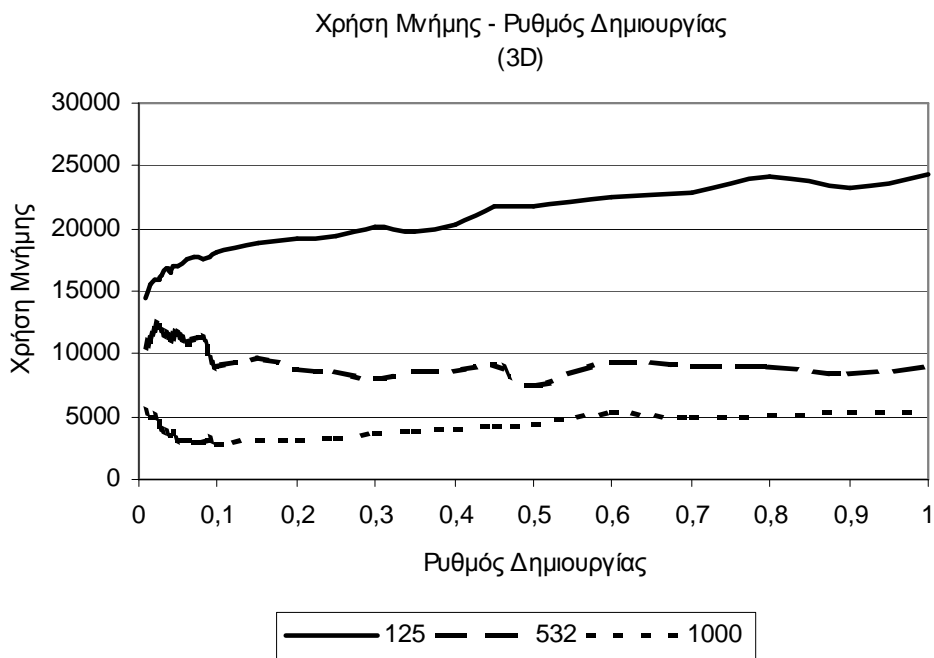
Σχήμα Α'.18: Μέσος Αριθμός Βημάτων - Ρυθμός Δημιουργίας Πακέτων Δεδομένων, για τρισδιάστατα γραφήματα (DSR)



Σχήμα Α'.19: Μέση Καθυστέρηση Λήψης - Ρυθμός Δημιουργίας Πακέτων Δεδομένων, για τρισδιάστατα γραφήματα (DSR)



Σχήμα Α'.20: Αριθμός Πακέτων Ελέγχου - Ρυθμός Δημιουργίας Πακέτων Δεδομένων, για τρισδιάστατα γραφήματα (DSR)



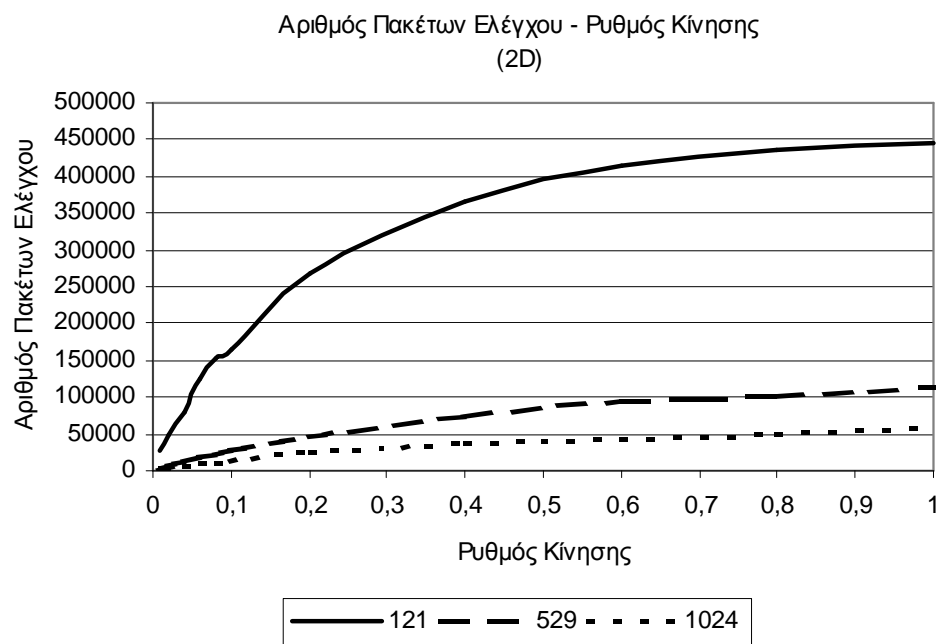
Σχήμα Α'.21: Χρήση Μνήμης - Ρυθμός Δημιουργίας Πακέτων Δεδομένων, για τρισδιάστατα γραφήματα (DSR)



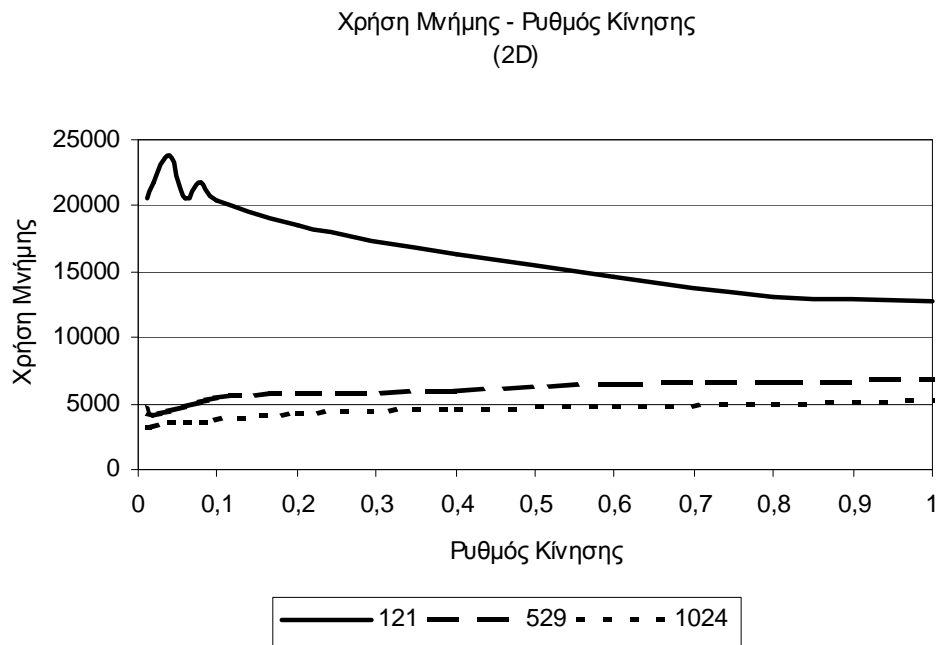
## Παράρτημα Β΄

# Γραφικές παραστάσεις για το ZRP

### Β΄.1 Ρυθμός κίνησης

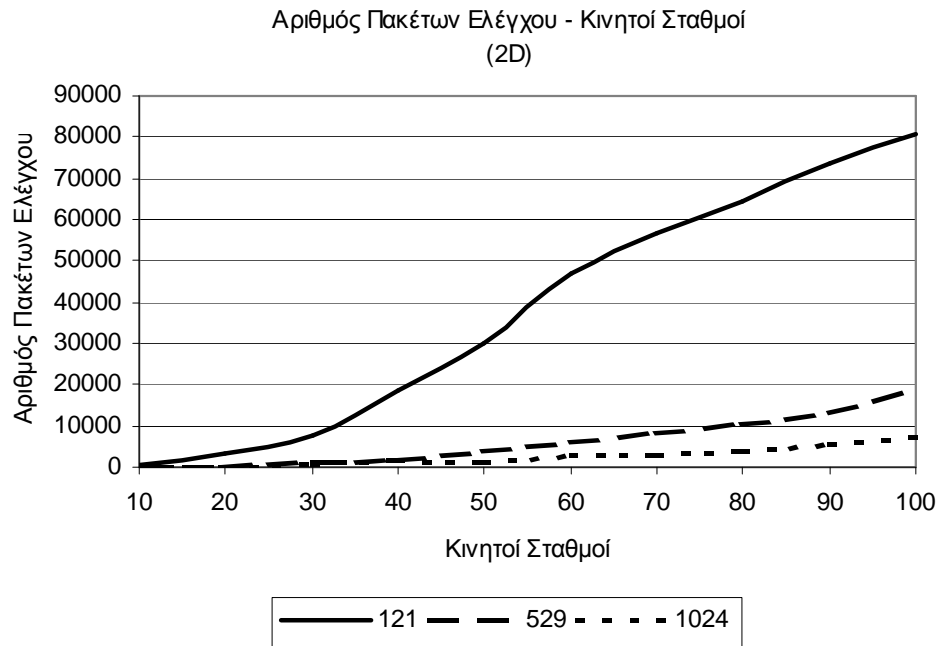


Σχήμα Β΄.1: Αριθμός Πακέτων Ελέγχου - Ρυθμός Κίνησης, για δισδιάστατα γραφήματα (ZRP)

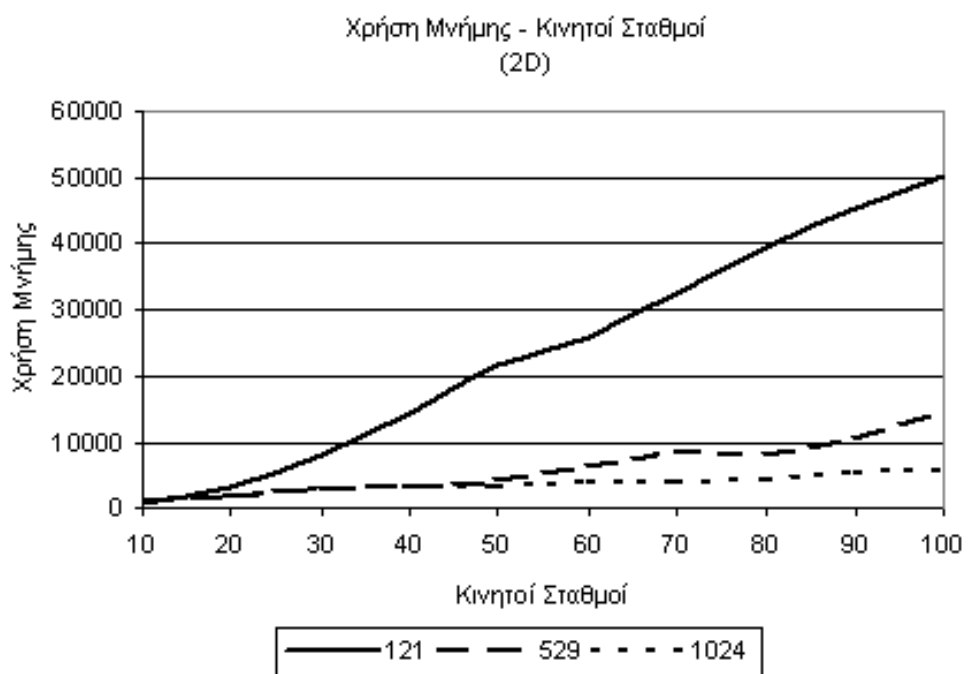


Σχήμα Β'.2: Χρήση Μνήμης - Ρυθμός Κίνησης, για δισδιάστατα γραφήματα (ZRP)

## B'.2 Κινητοί σταθμοί



Σχήμα B'.3: Αριθμός Πακέτων Ελέγχου - Κινητοί Σταθμοί, για δισδιάστατα γραφήματα (ZRP)

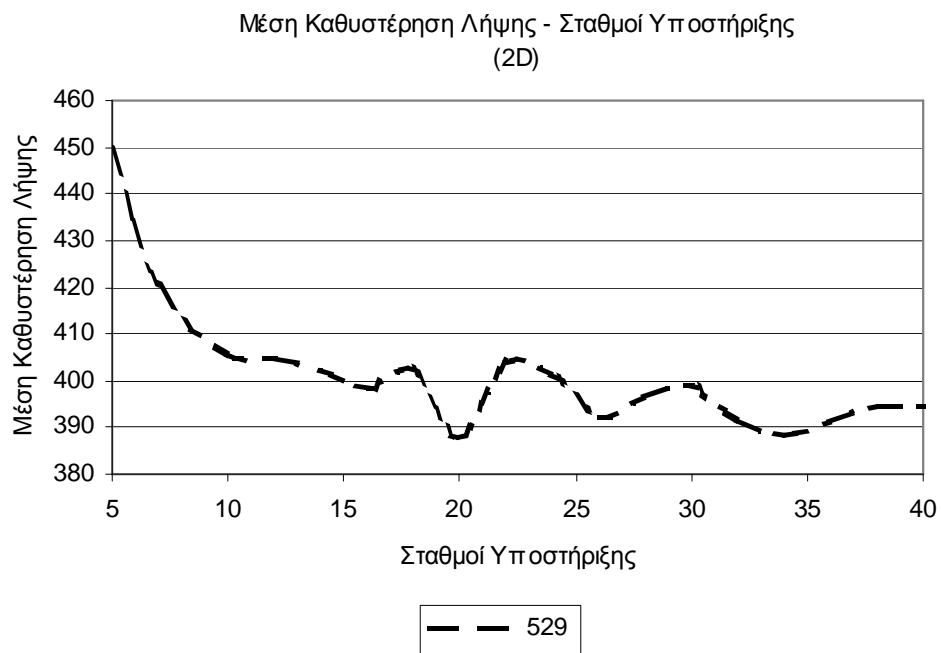


Σχήμα Β'.4: Χρήση Μνήμης - Κινητοί Σταθμοί, για δισδιάστατα γραφήματα (ZRP)

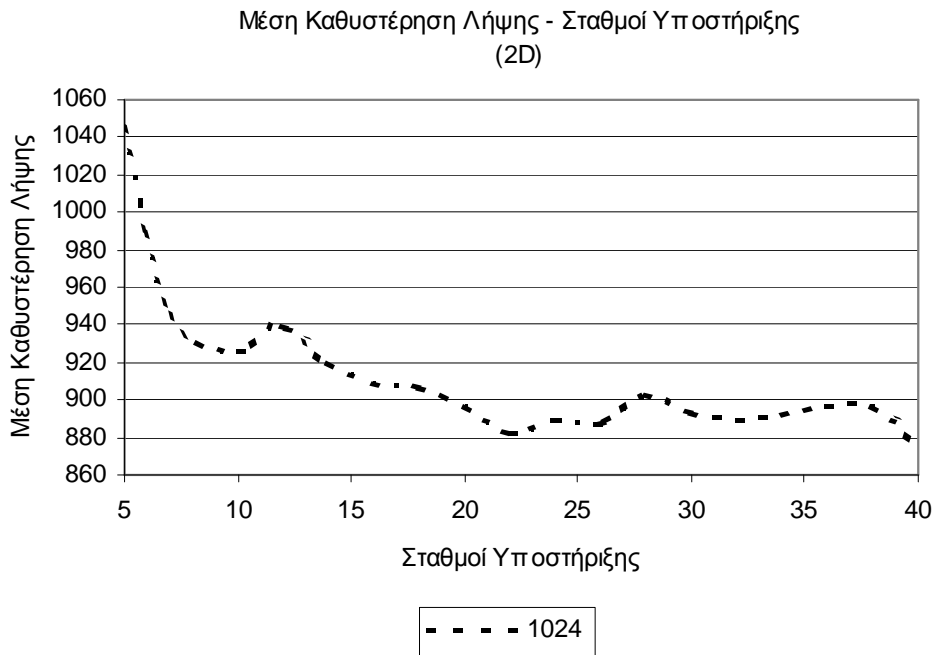
## Παράρτημα Γ'

# Γραφικές παραστάσεις για το SRP

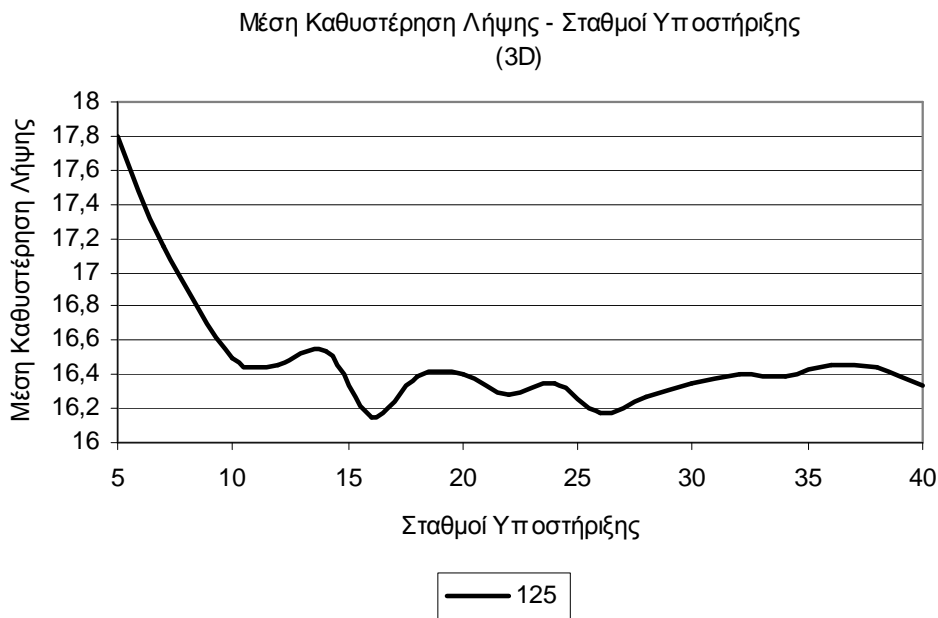
### Γ'.1 Σταθμοί υποστήριξης



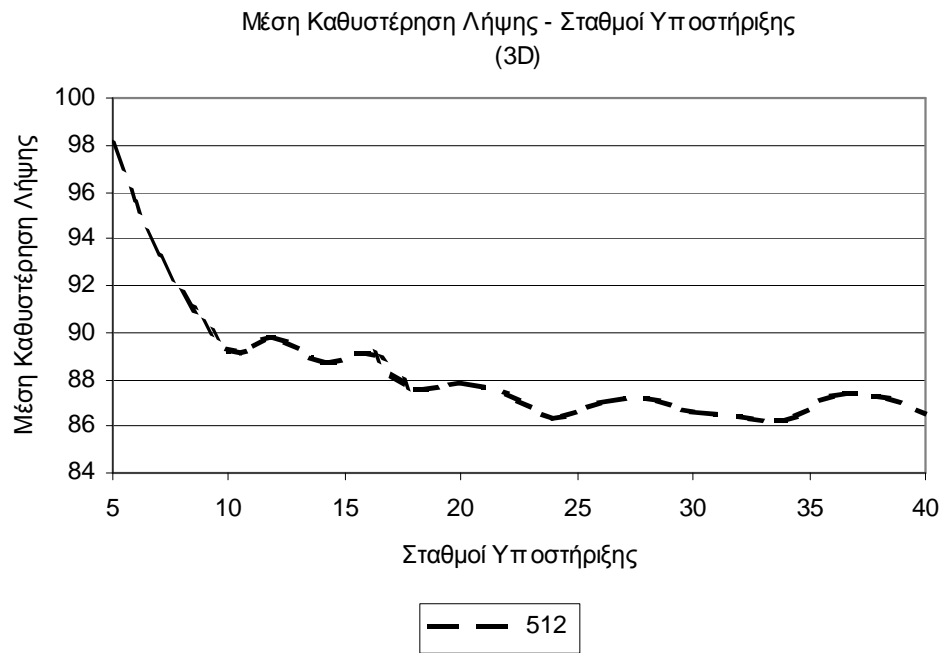
Σχήμα Γ'.1: Μέση Καθυστέρηση Λήψης - Σταθμοί Υποστήριξης, για δισδιάστατα γραφήματα (529) (SRP)



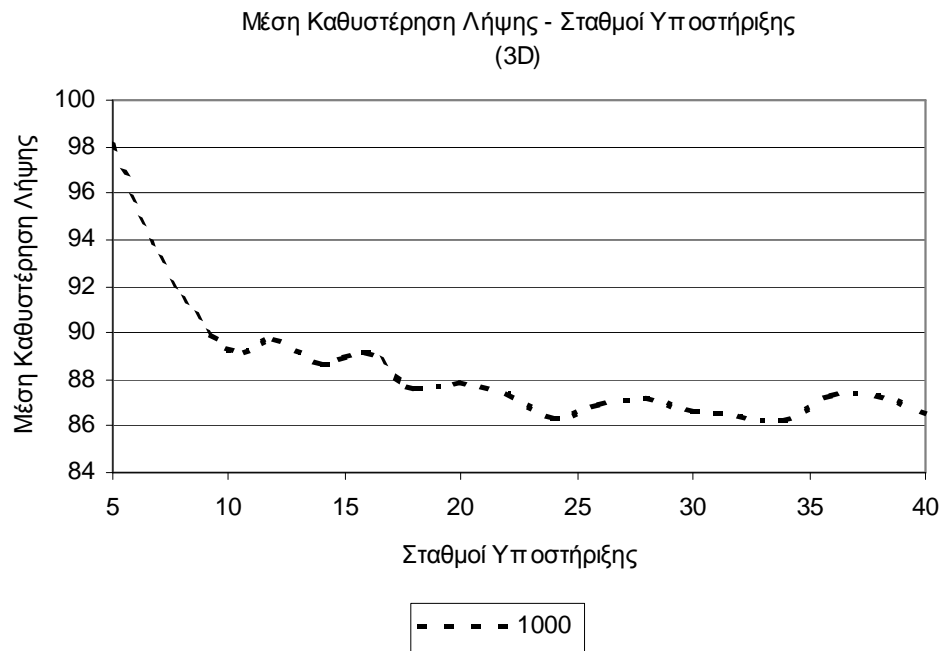
Σχήμα Γ'.2: Μέση Καθυστέρηση Λήψης - Σταθμοί Υποστήριξης, για διδιάστατα γραφήματα (1024) (SRP)



Σχήμα Γ'.3: Μέση Καθυστέρηση Λήψης - Σταθμοί Υποστήριξης, για τριδιάστατα γραφήματα (125) (SRP)

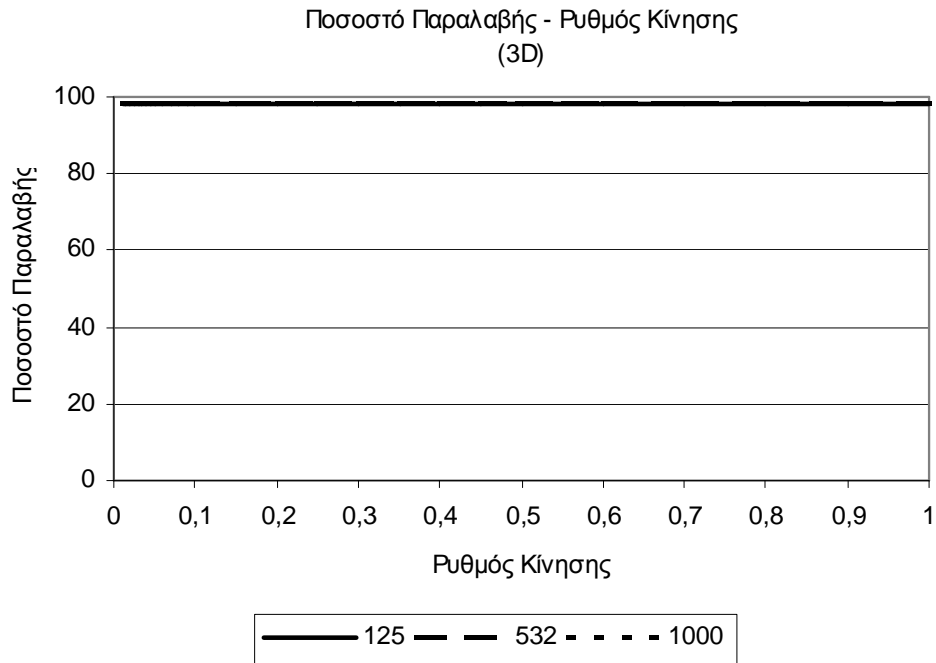


Σχήμα Γ'.4: Μέση Καθυστέρηση Λήψης - Σταθμοί Υποστήριξης, για τρισδιάστατα γραφήματα (512) (SRP)



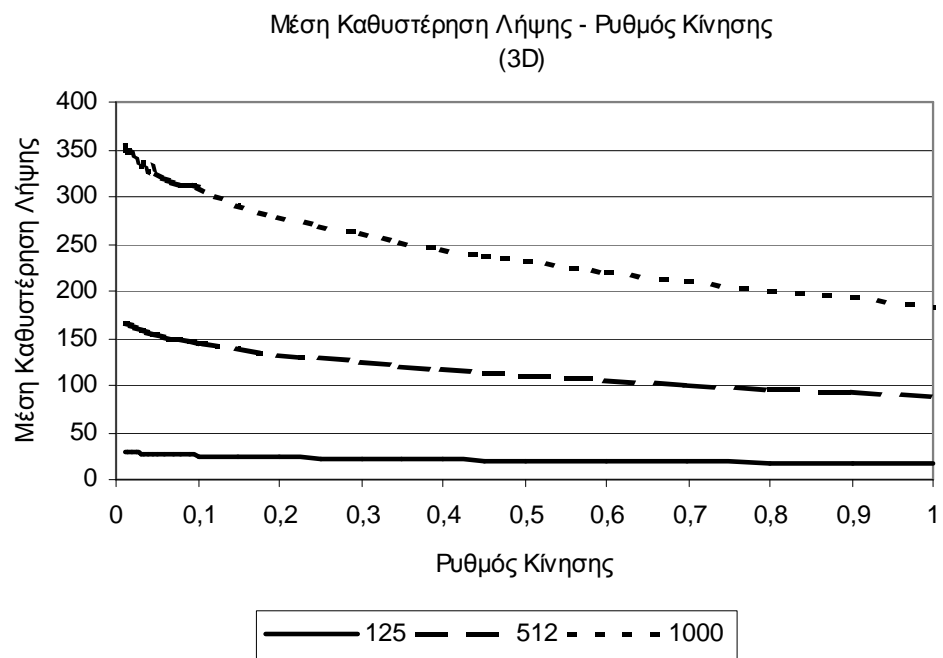
Σχήμα Γ'.5: Μέση Καθυστέρηση Λήψης - Σταθμοί Υποστήριξης, για τρισδιάστατα γραφήματα (1000) (SRP)

## Γ'.2 Ρυθμός κίνησης



Σχήμα Γ'.6: Ποσοστό Παραλαβής - Ρυθμός Κίνησης, για τρισδιάστατα γραφήματα (SRP)

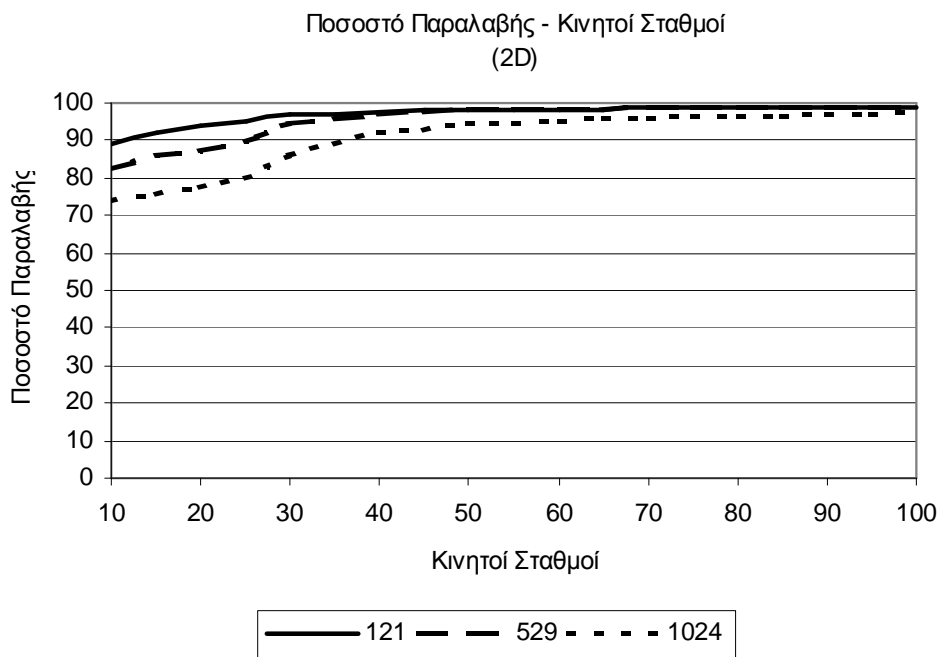




Σχήμα Γ'.7: Μέση Καθυστέρηση Λήψης - Ρυθμός Κίνησης, για τρισδιάστατα γραφήματα (SRP)

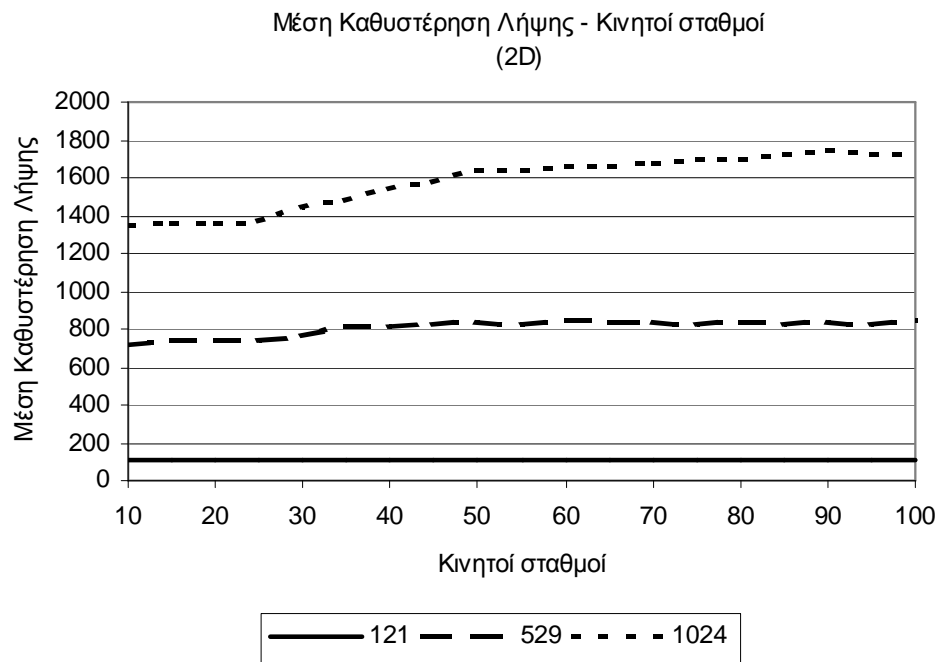
### Γ'.3 Κινητοί σταθμοί

1

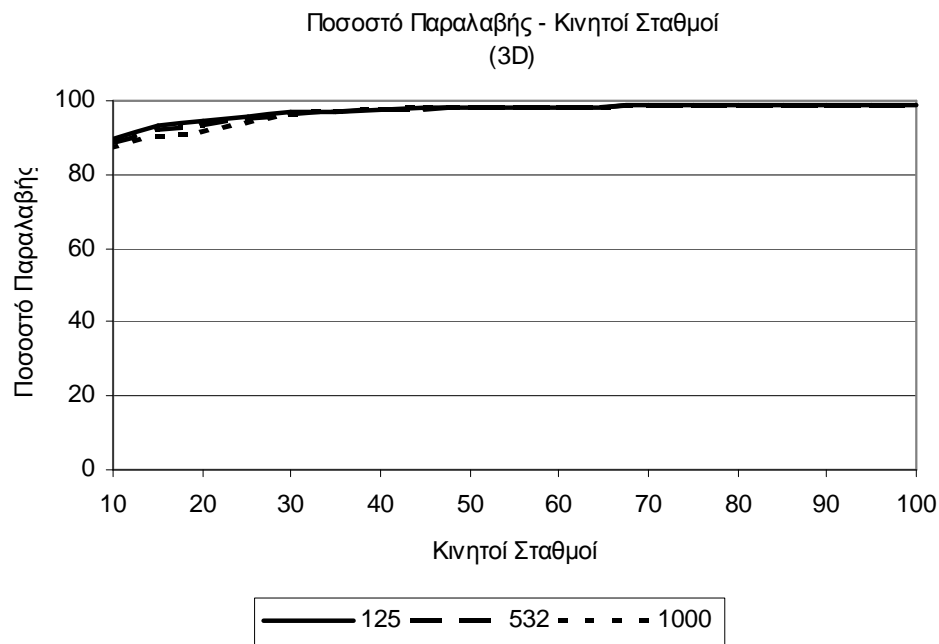


Σχήμα Γ'.8: Ποσοστό Παραλαβής - Κινητοί Σταθμοί, για δισδιάστατα γραφήματα (SRP)

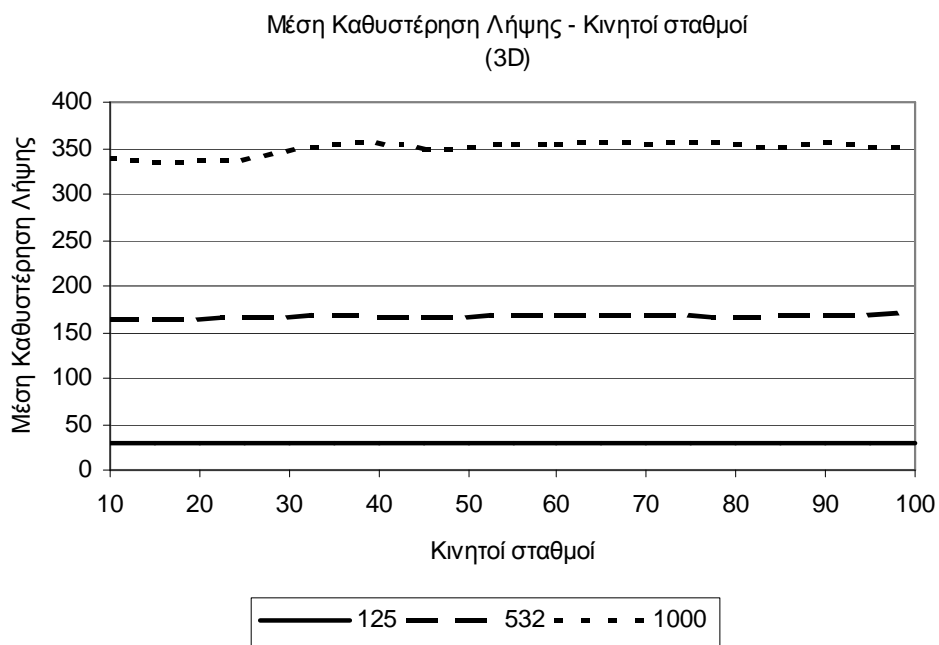
<sup>1</sup>Στους αριθμούς των κινητών σταθμών δεν συμπεριλαμβάνεται ο αριθμός των σταθμών της υποστήριξης.



Σχήμα Γ'.9: Μέση Καθυστέρηση Λήψης - Κινητοί Σταθμοί, για δισδιάστατα γραφήματα (SRP)

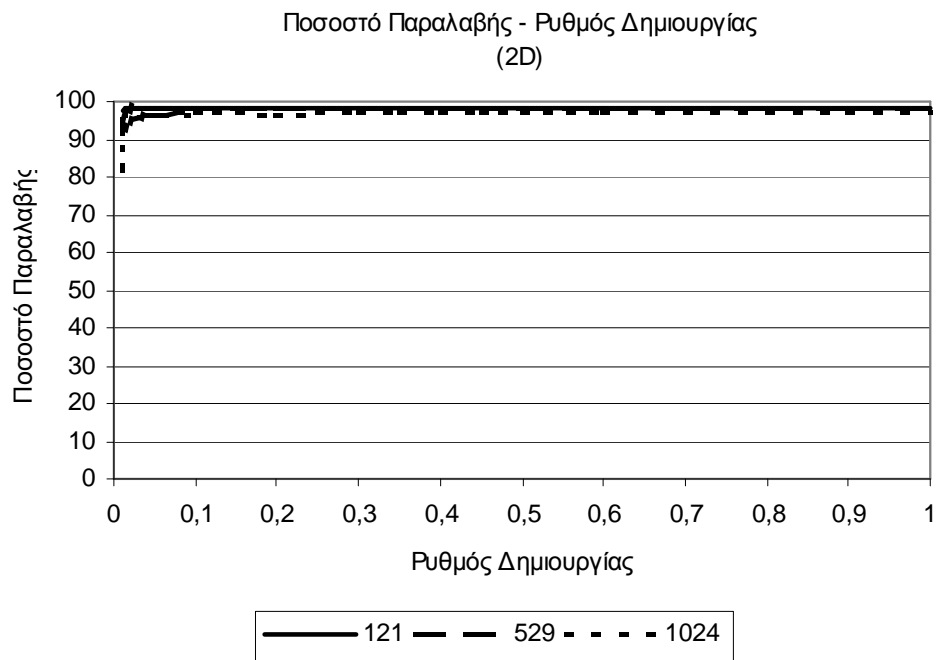


Σχήμα Γ'.10: Ποσοστό Παραλαβής - Κινητοί Σταθμοί, για τρισδιάστατα γραφήματα (SRP)

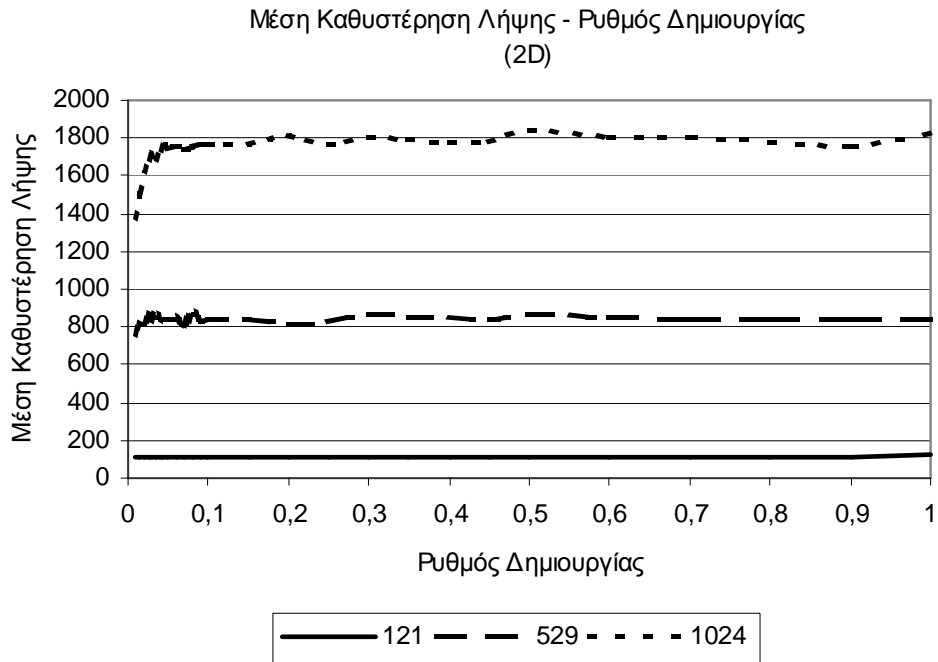


Σχήμα Γ'.11: Μέση Καθυστέρηση Λήψης - Κινητοί Σταθμοί, για τρισδιάστατα γραφήματα (SRP)

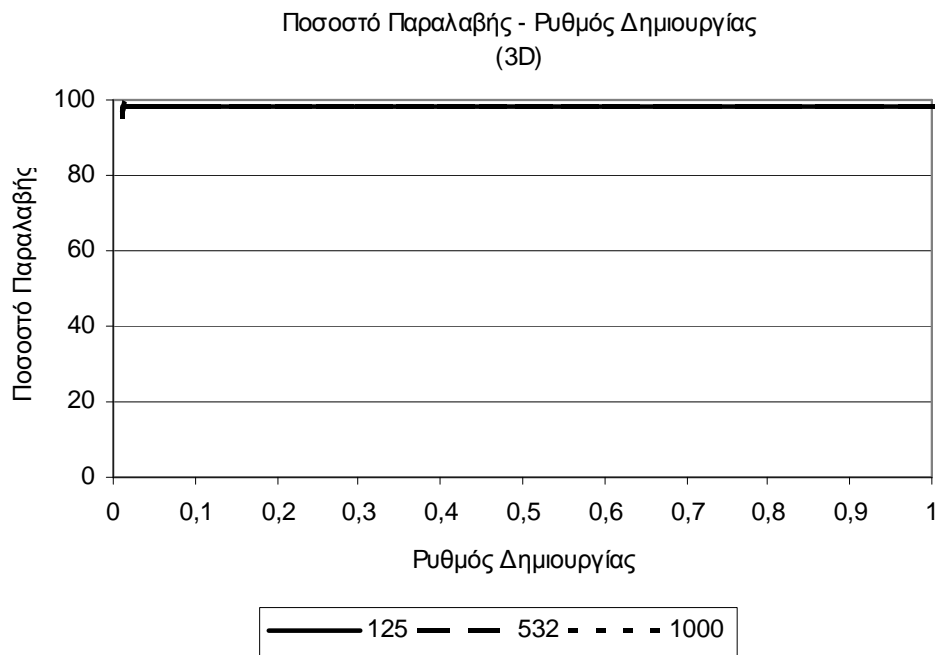
## Γ'.4 Ρυθμός δημιουργίας πακέτων δεδομένων



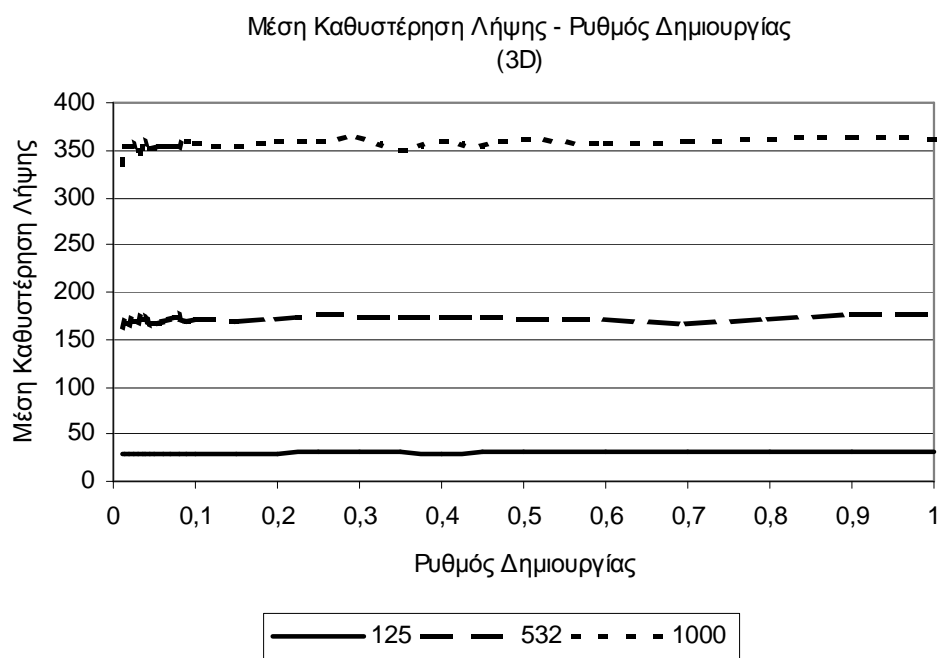
Σχήμα Γ'.12: Ποσοστό Παραλαβής - Ρυθμός Δημιουργίας Πακέτων Δεδομένων, για δισδιάστατα γραφήματα (SRP)



Σχήμα Γ'.13: Μέση Καθυστέρηση Λήψης - Ρυθμός Δημιουργίας Πακέτων Δεδομένων, για δισδιάστατα γραφήματα (SRP)



Σχήμα Γ'.14: Ποσοστό Παραλαβής - Ρυθμός Δημιουργίας Πακέτων Δεδομένων, για τρισδιάστατα γραφήματα (SRP)



Σχήμα Γ'.15: Μέση Καθυστέρηση Λήψης - Ρυθμός Δημιουργίας Πακέτων Δεδομένων, για τρισδιάστατα γραφήματα (SRP)

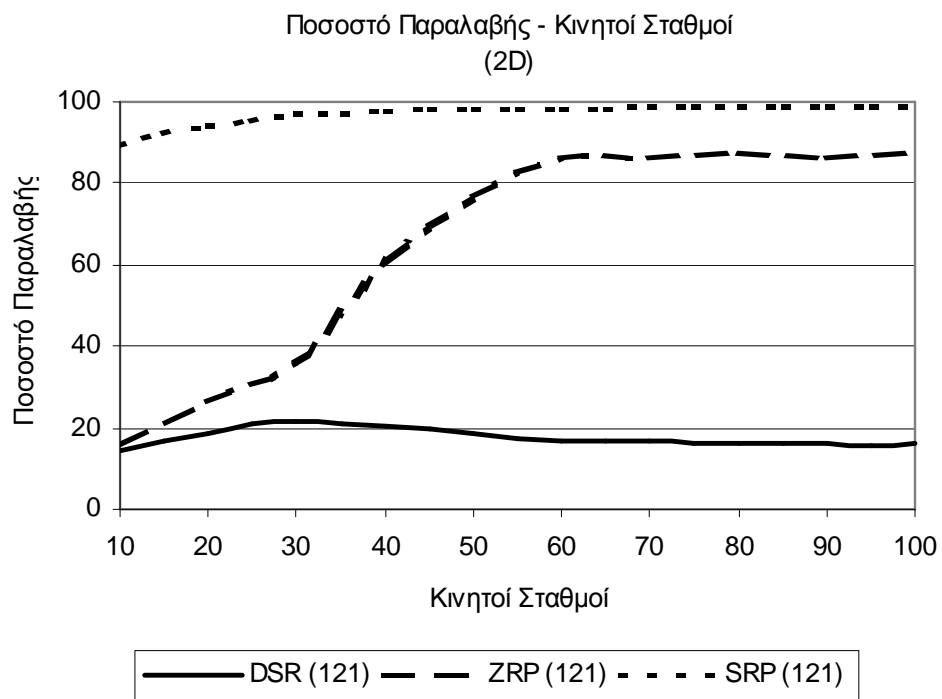




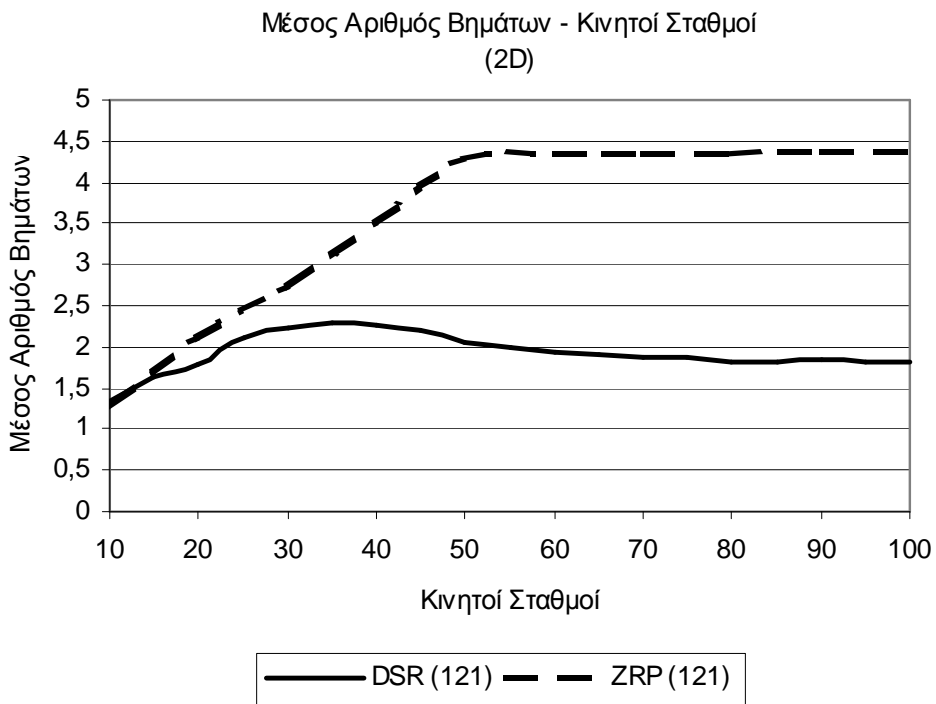
## Παράρτημα Δ'

### Γραφικές παραστάσεις για τη σύγκριση DSR, ZRP και SRP

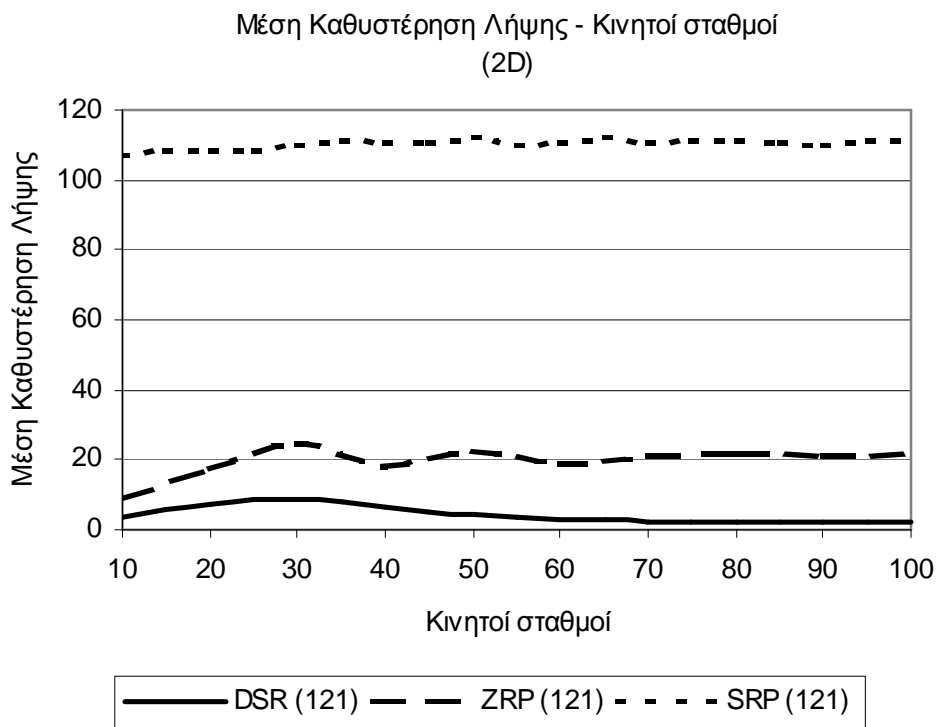
#### Δ'.1 Κινητοί σταθμοί



Σχήμα Δ'.1: Ποσοστό Παραλαβής - Κινητοί Σταθμοί, για δισδιάστατα γραφήματα (DSR, ZRP, SRP)

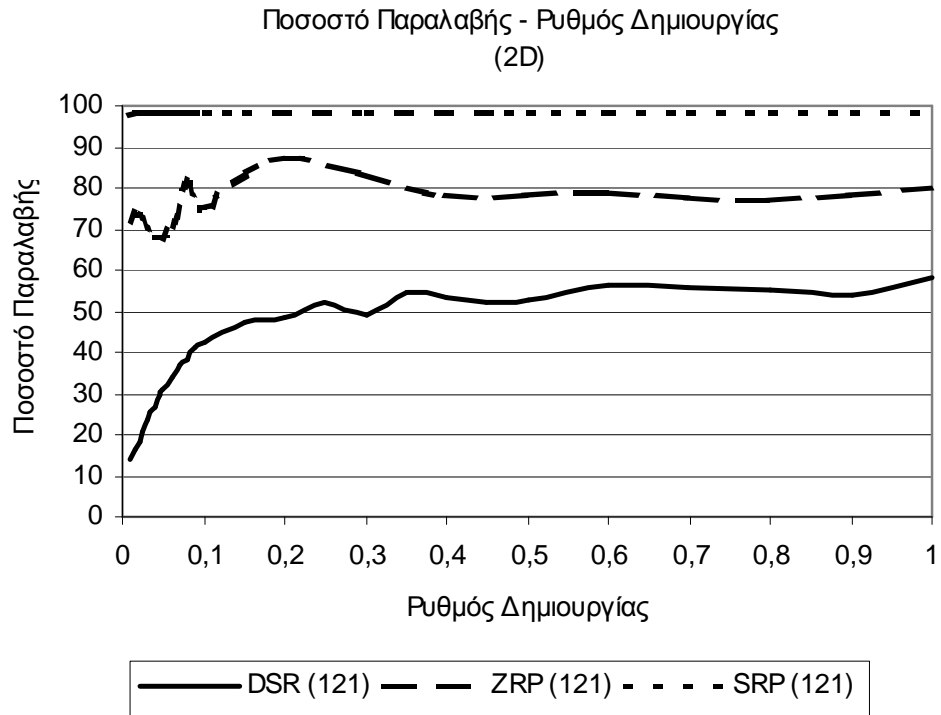


Σχήμα Δ'2: Μέσος Αριθμός Βημάτων - Κινητοί Σταθμοί, για δισδιάστατα γραφήματα (DSR, ZRP)

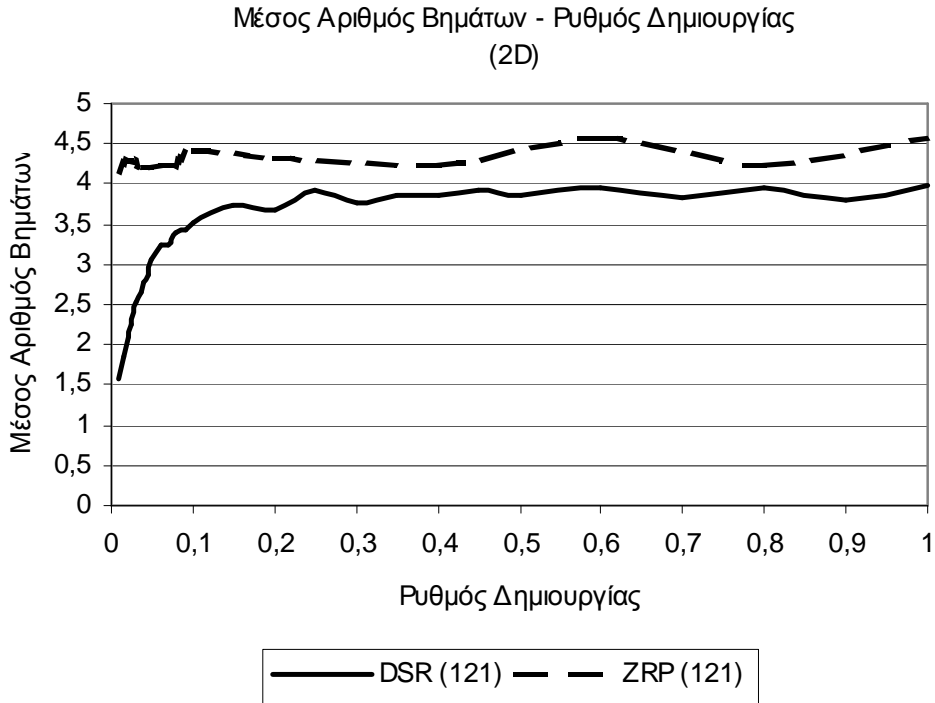


Σχήμα Δ'3: Μέση Καθυστέρηση Λήψης - Κινητοί Σταθμοί, για δισδιάστατα γραφήματα (DSR, ZRP, SRP)

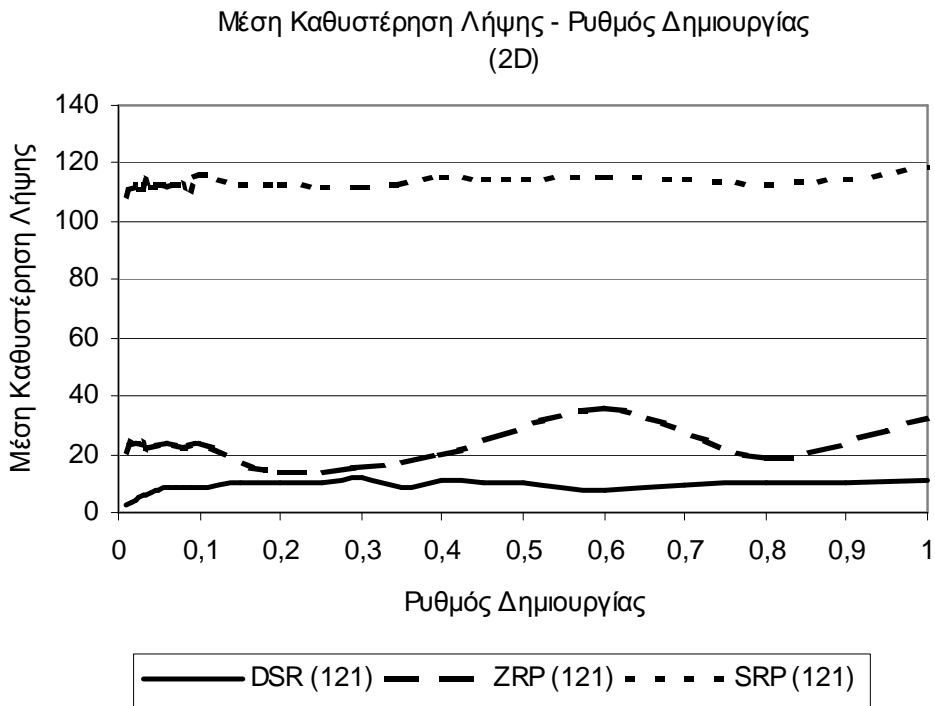
Δ'.2 Ρυθμός δημιουργίας πακέτων δεδομένων



Σχήμα Δ'.4: Ποσοστό Παραλαβής - Ρυθμός Δημιουργίας Πακέτων Δεδομένων, για δισδιάστατα γραφήματα (DSR, ZRP, SRP)



Σχήμα Δ.5: Μέσος Αριθμός Βημάτων - Ρυθμός Δημιουργίας Πακέτων Δεδομένων, για δισδιάστατα γραφήματα (DSR, ZRP)

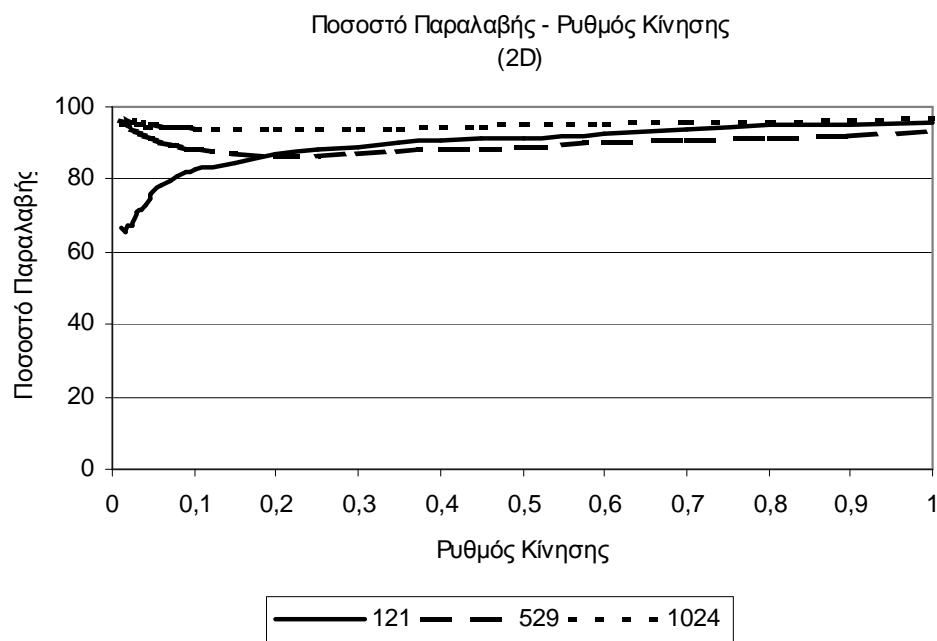


Σχήμα Δ.6: Μέση Καθυστέρηση Λήψης - Ρυθμός Δημιουργίας Πακέτων Δεδομένων, για δισδιάστατα γραφήματα (DSR, ZRP, SRP)

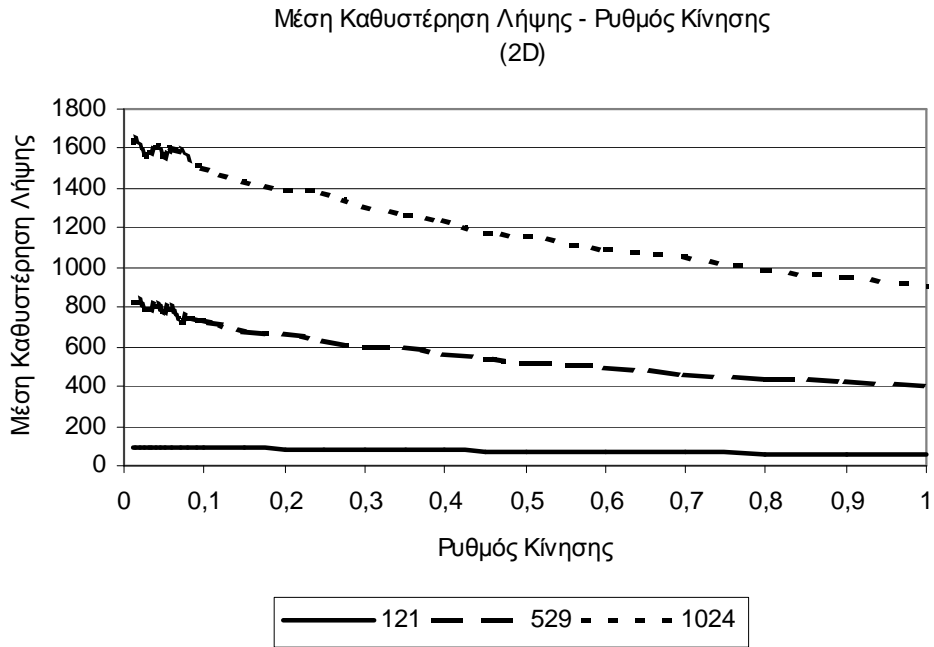
## Παράρτημα Ε΄

### Γραφικές παραστάσεις για το συνδυασμό DSR με το SRP

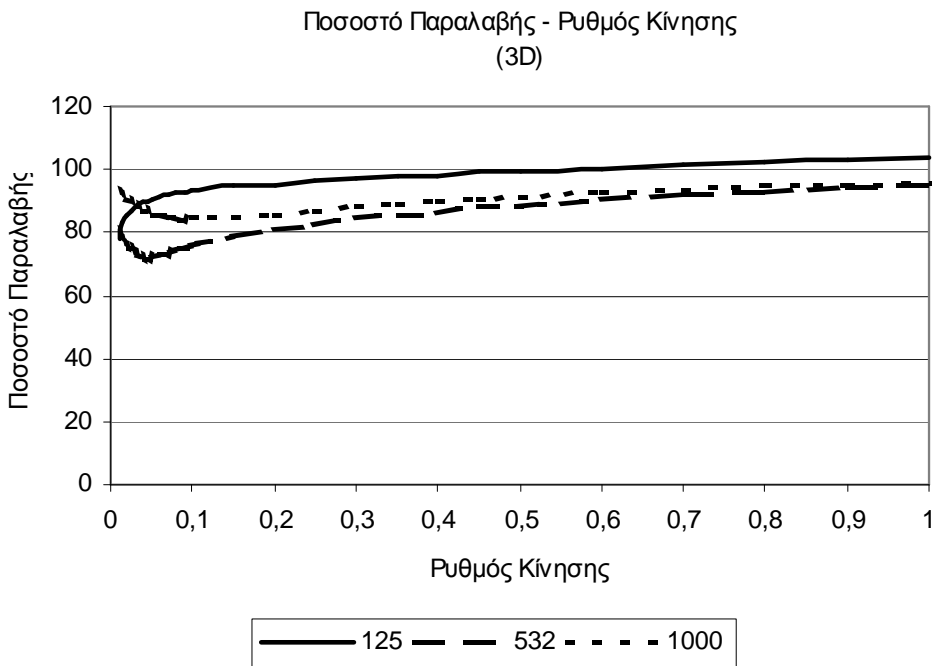
#### Ε΄.1 Ρυθμός κίνησης



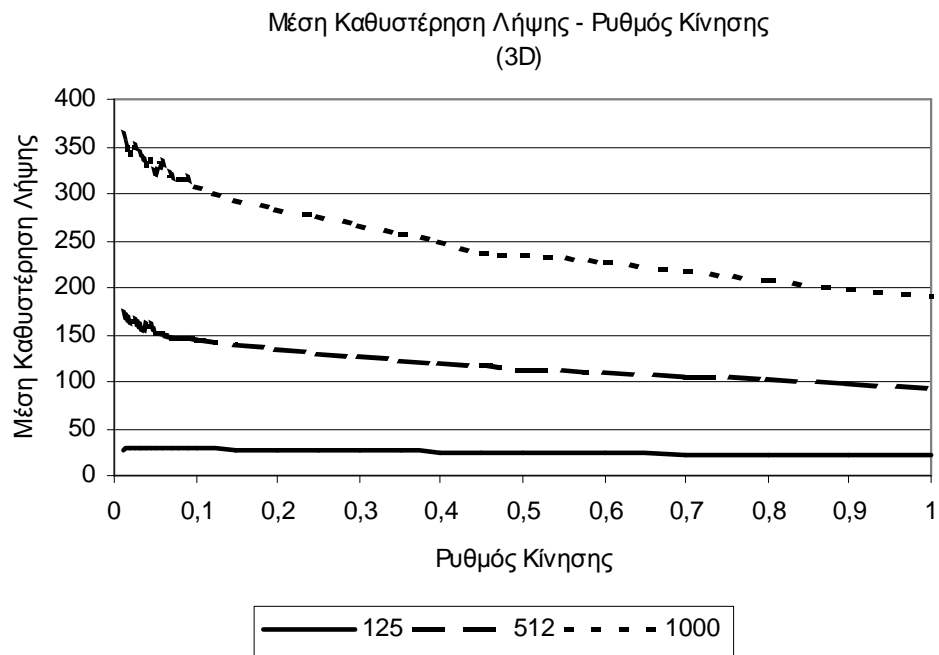
Σχήμα Ε΄.1: Ποσοστό Παραλαβής - Ρυθμός Κίνησης, για δισδιάστατα γραφήματα (συνδυασμός DSR - SRP)



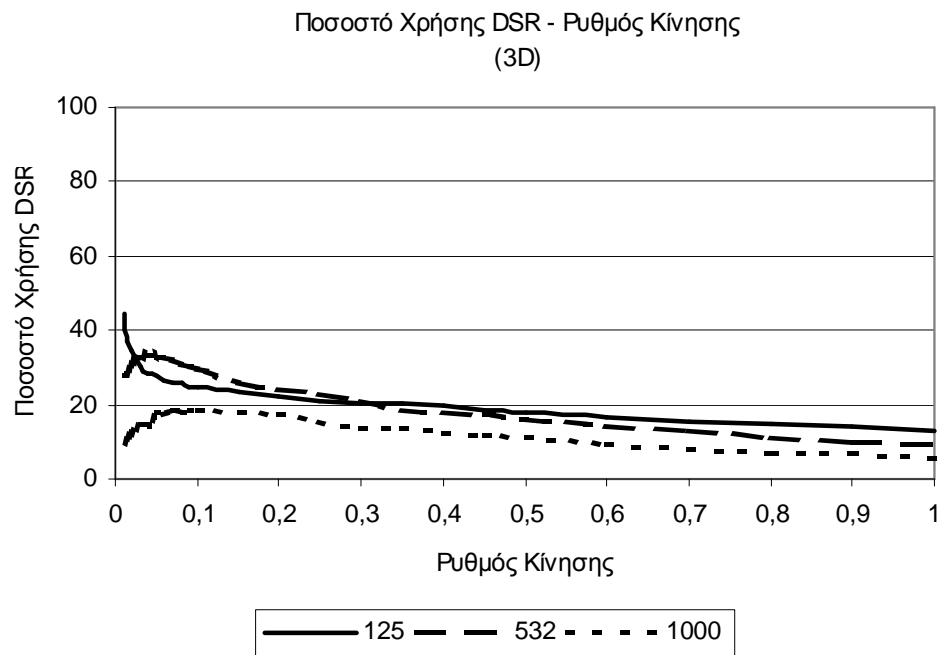
Σχήμα Ε'.2: Μέση Καθυστέρηση Λήψης - Ρυθμός Κίνησης, για δισδιάστατα γραφήματα (συνδυασμός DSR - SRP)



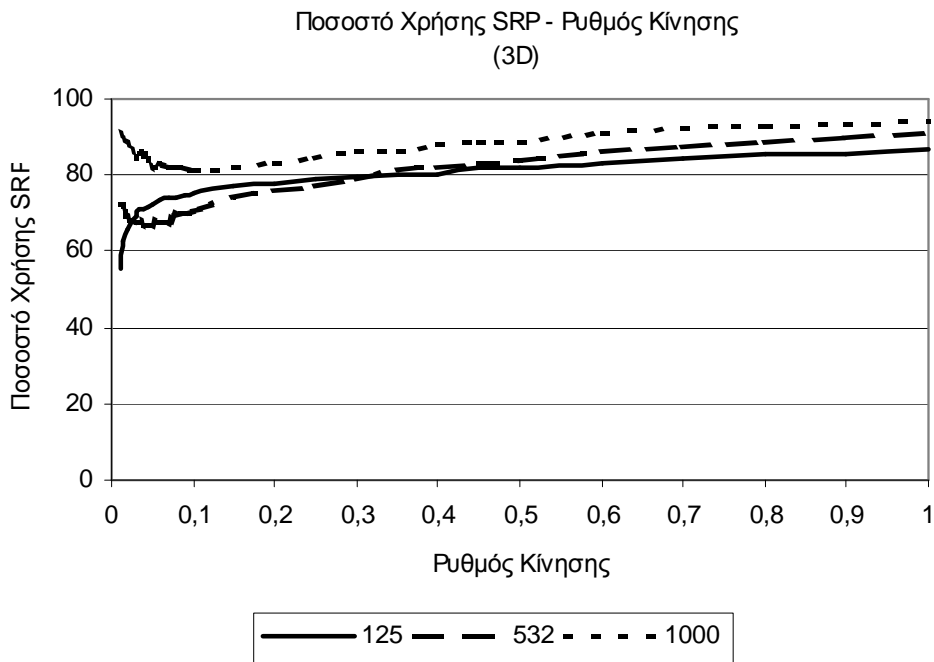
Σχήμα Ε'.3: Ποσοστό Παραλαβής - Ρυθμός Κίνησης, για τρισδιάστατα γραφήματα (συνδυασμός DSR - SRP)



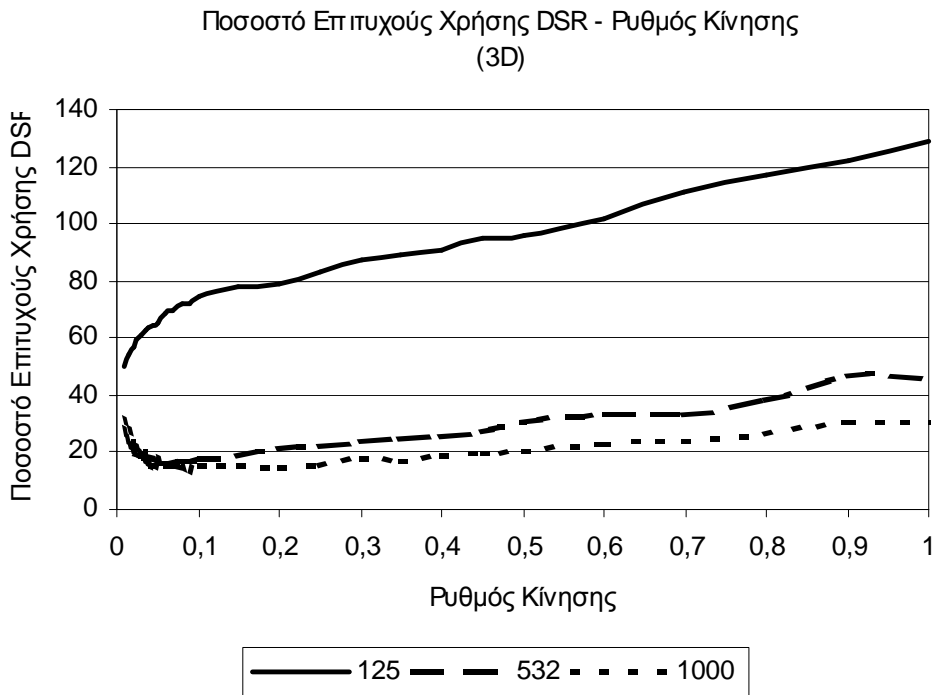
Σχήμα Ε'.4: Μέση Καθυστέρηση Λήψης - Ρυθμός Κίνησης, για τρισδιάστατα γραφήματα (συνδυασμός DSR - SRP)



Σχήμα Ε'.5: Ποσοστό Χρήσης DSR - Ρυθμός Κίνησης, για τρισδιάστατα γραφήματα (συνδυασμός DSR - SRP)

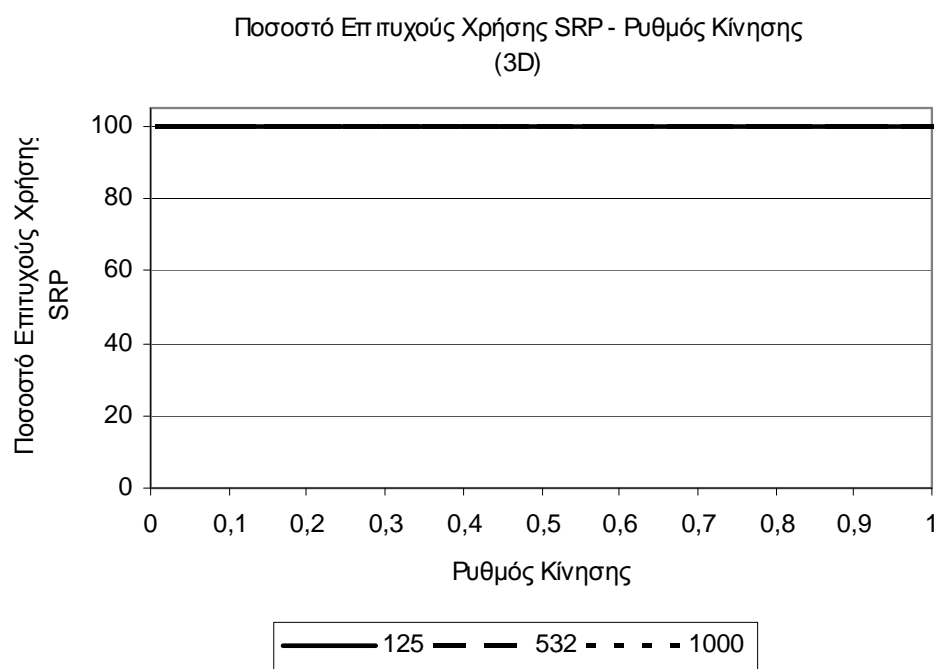


Σχήμα Ε'.6: Ποσοστό Χρήσης SRP - Ρυθμός Κίνησης, για τρισδιάστατα γραφήματα (συνδυασμός DSR - SRP)



Σχήμα Ε'.7: Ποσοστό Επιτυχούς Χρήσης DSR - Ρυθμός Κίνησης, για τρισδιάστατα γραφήματα (συνδυασμός DSR - SRP)

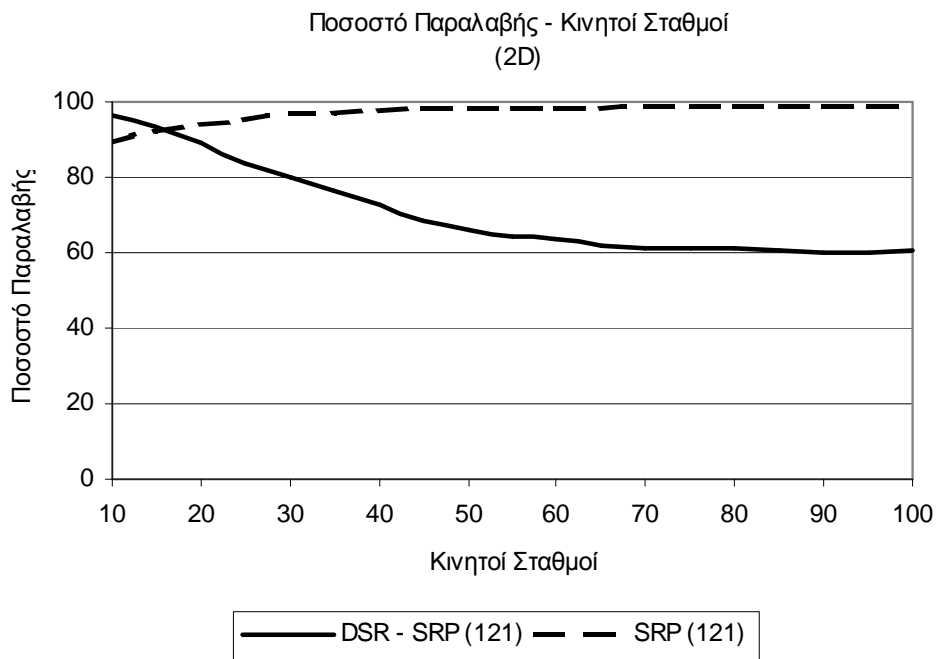




Σχήμα Ε'.8: Ποσοστό Επιτυχούς Χρήσης SRP - Ρυθμός Κίνησης, για τρισδιάστατα γραφήματα (συνδυασμός DSR - SRP)

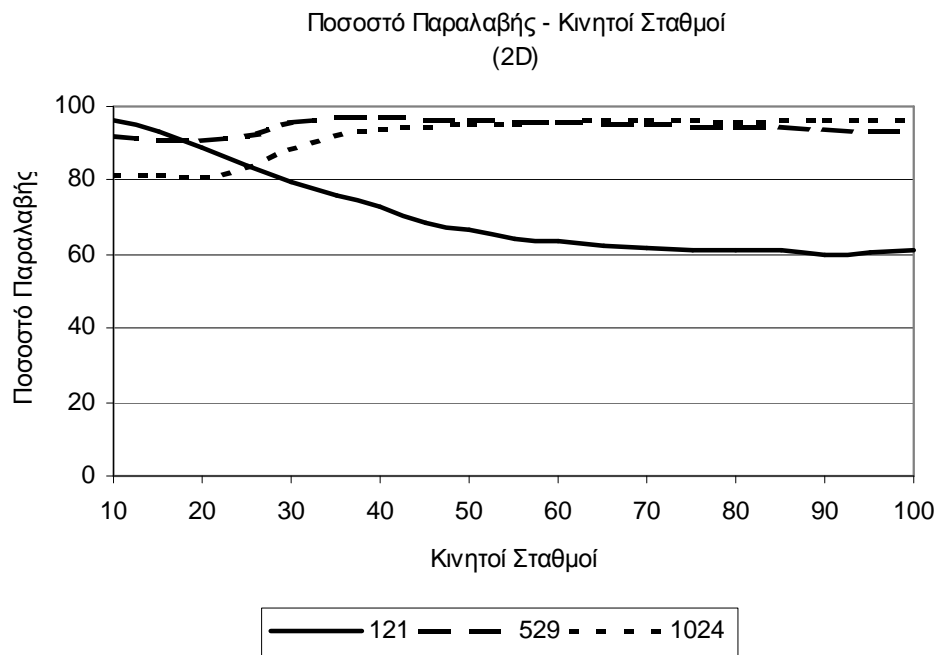
## Ε'.2 Κινητοί σταθμοί

1

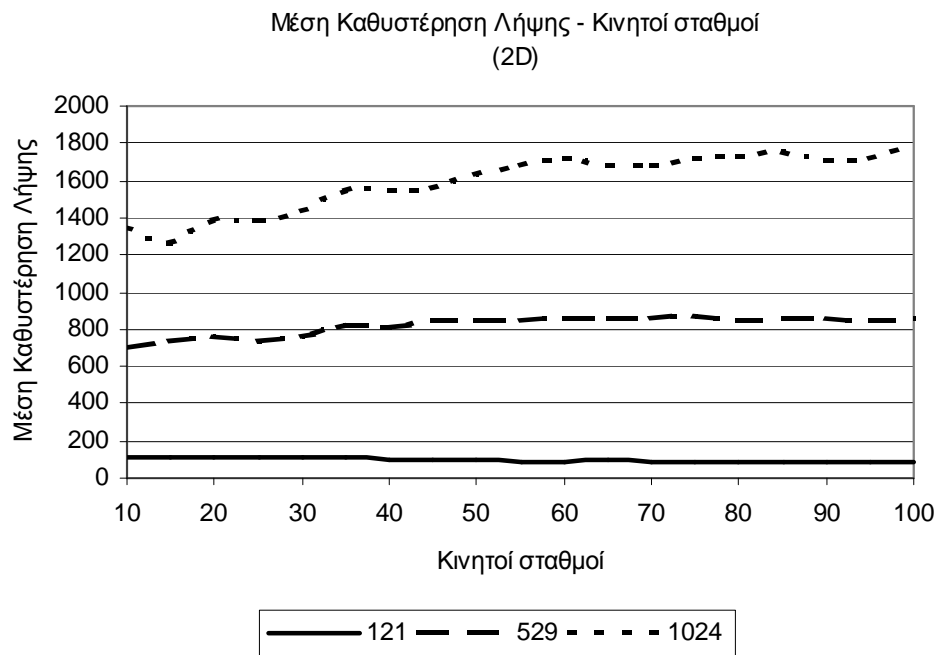


Σχήμα Ε'.9: Ποσοστό Παραλαβής - Κινητοί Σταθμοί, για διδιάστατα γραφήματα (συνδυασμός DSR - SRP, SRP)

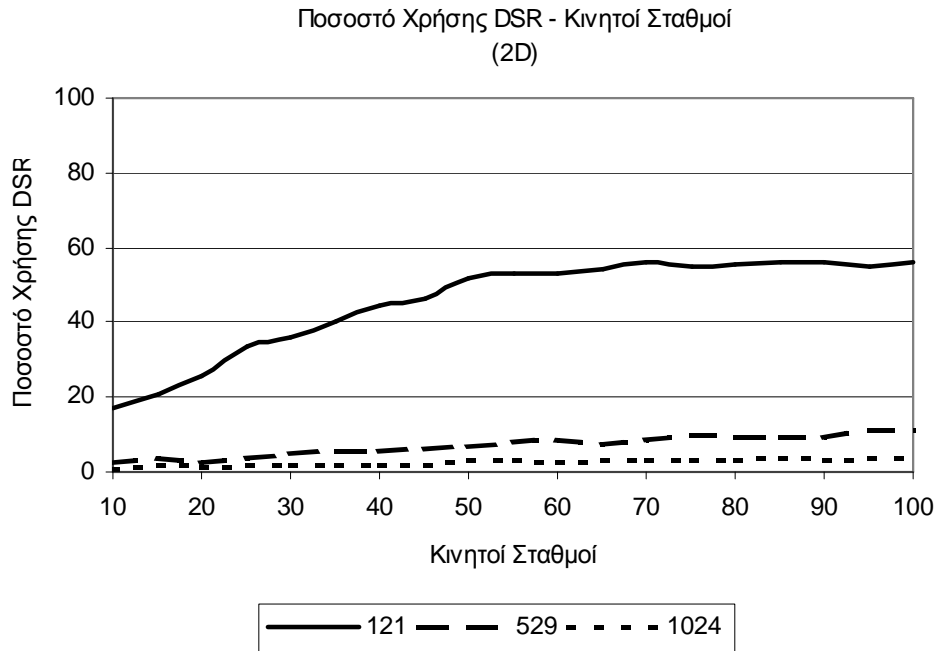
<sup>1</sup>Στους αριθμούς των κινητών σταθμών δεν συμπεριλαμβάνεται ο αριθμός των σταθμών της υποστήριξης.



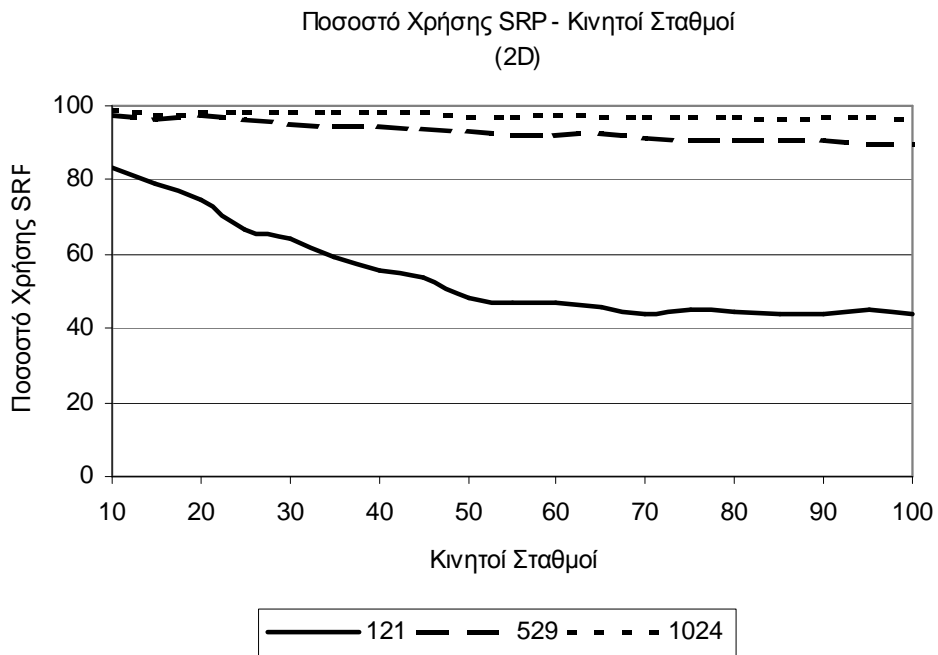
Σχήμα Ε'.10: Ποσοστό Παραλαβής - Κινητοί Σταθμοί, για δισδιάστατα γραφήματα (συνδυασμός DSR - SRP)



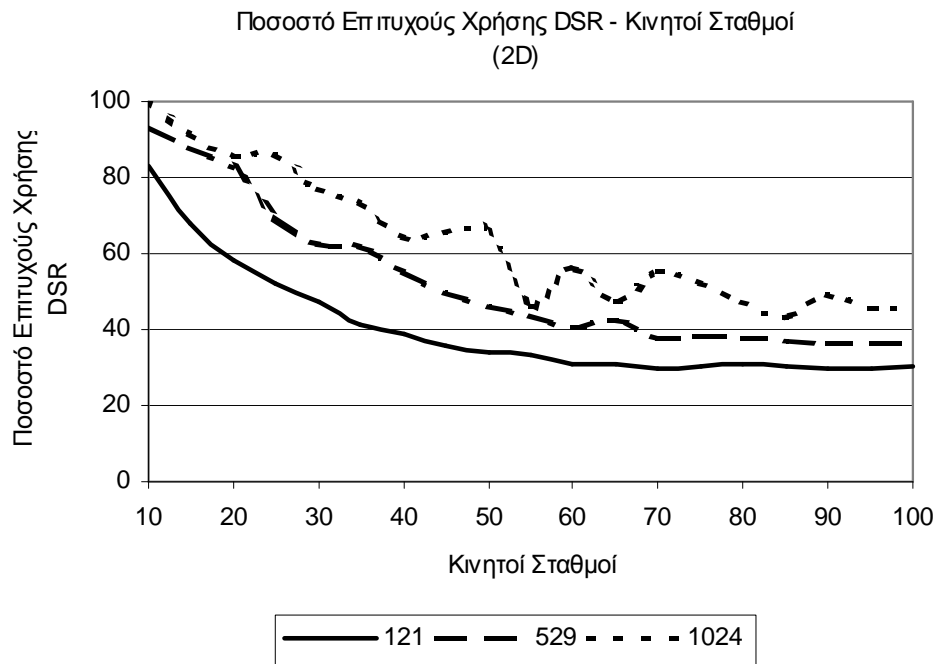
Σχήμα Ε'.11: Μέση Καθυστέρηση Λήψης - Κινητοί Σταθμοί, για δισδιάστατα γραφήματα (συνδυασμός DSR - SRP)



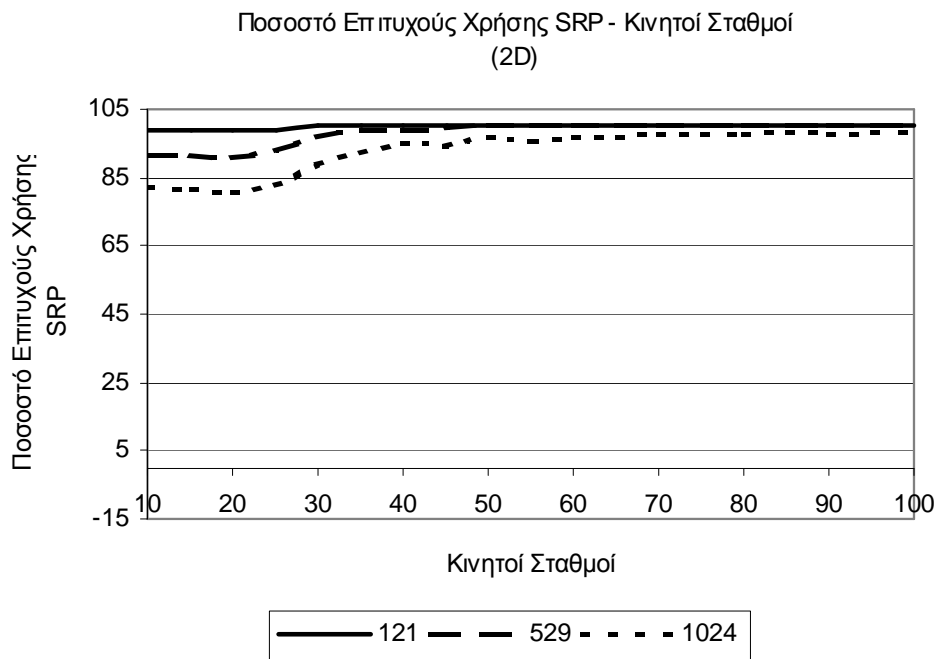
Σχήμα Ε'.12: Ποσοστό Χρήσης DSR - Κινητοί Σταθμοί, για δισδιάστατα γραφήματα (συνδυασμός DSR - SRP)



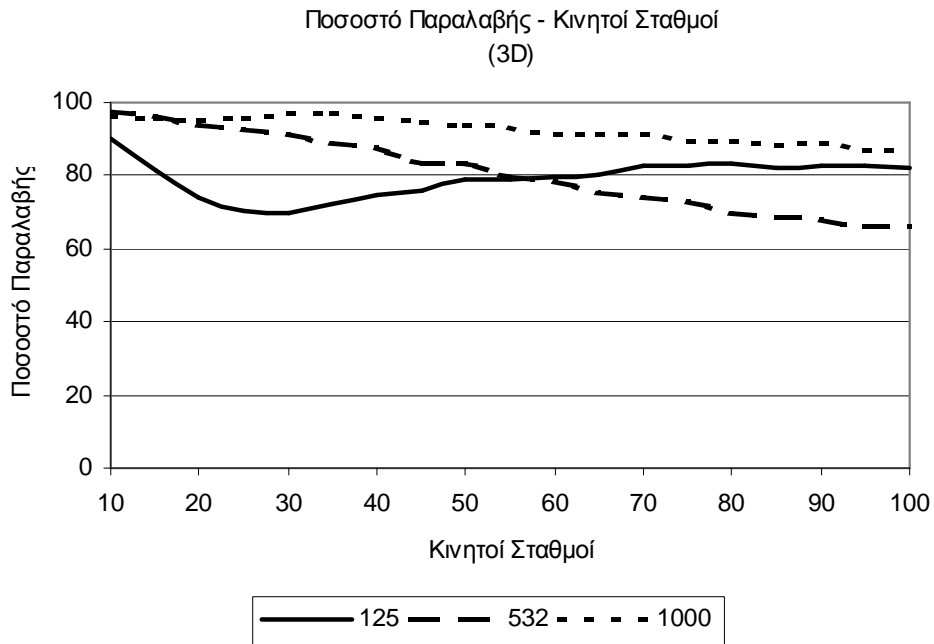
Σχήμα Ε'.13: Ποσοστό Χρήσης SRP - Κινητοί Σταθμοί, για δισδιάστατα γραφήματα (συνδυασμός DSR - SRP)



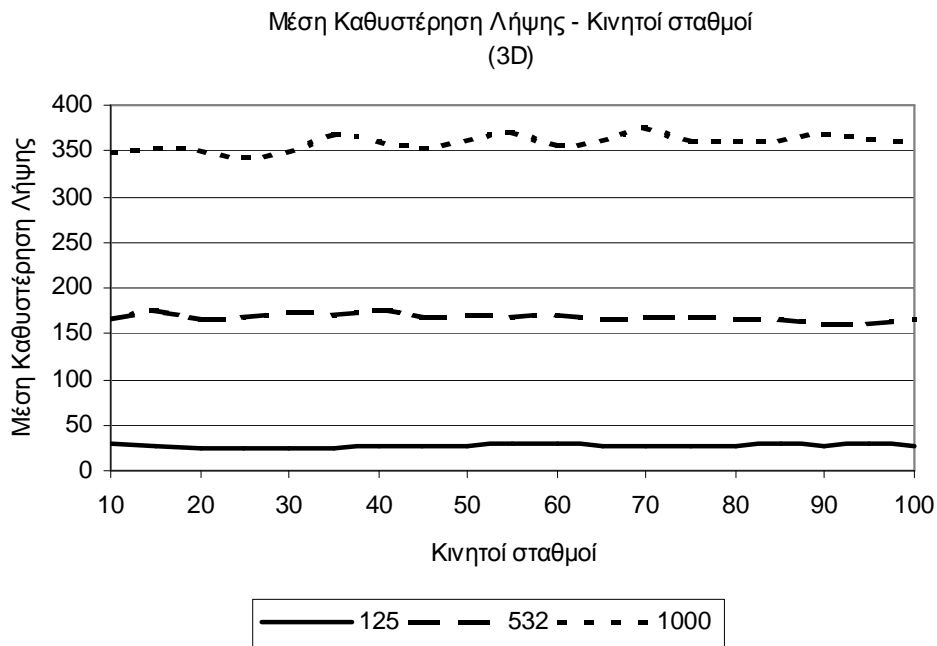
Σχήμα Ε'.14: Ποσοστό Επιτυχούς Χρήσης DSR - Κινητοί Σταθμοί, για διασδιάστατα γραφήματα (συνδυασμός DSR - SRP)



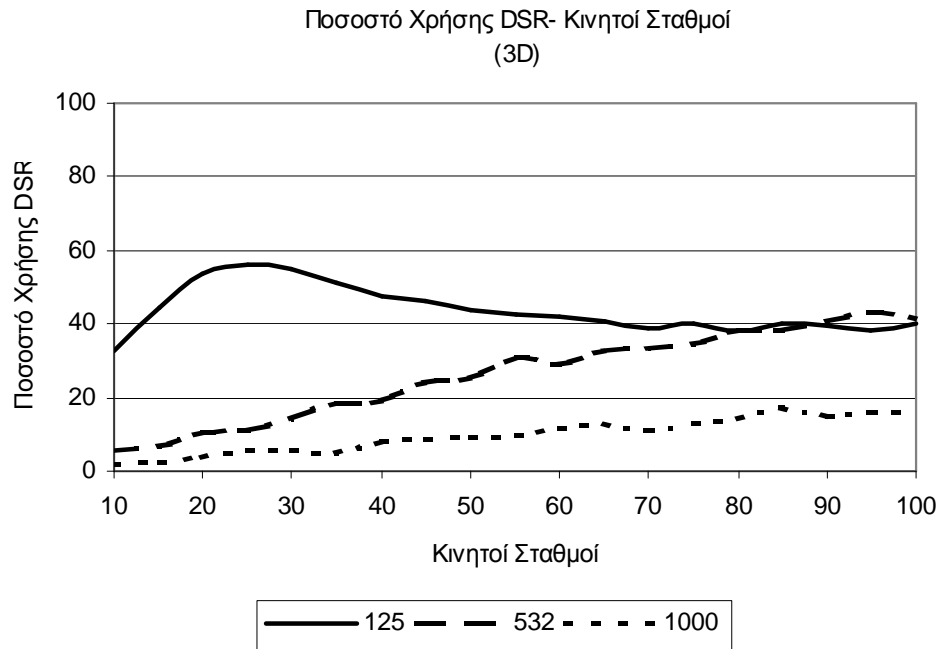
Σχήμα Ε'.15: Ποσοστό Επιτυχούς Χρήσης SRP - Κινητοί Σταθμοί, για διασδιάστατα γραφήματα (συνδυασμός DSR - SRP)



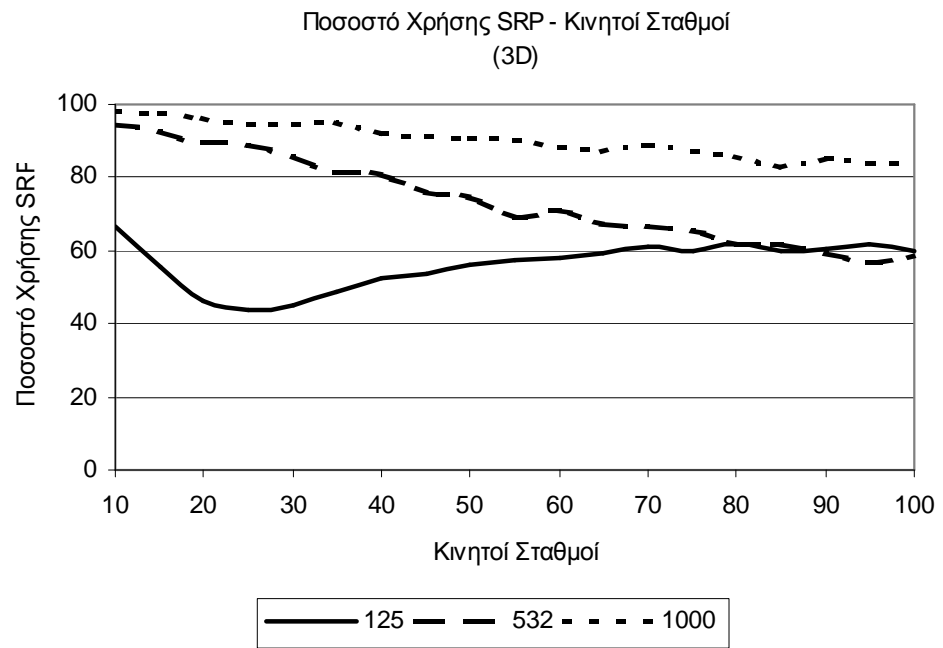
Σχήμα Ε'.16: Ποσοστό Παραλαβής - Κινητοί Σταθμοί, για τρισδιάστατα γραφήματα (συνδυασμός DSR - SRP)



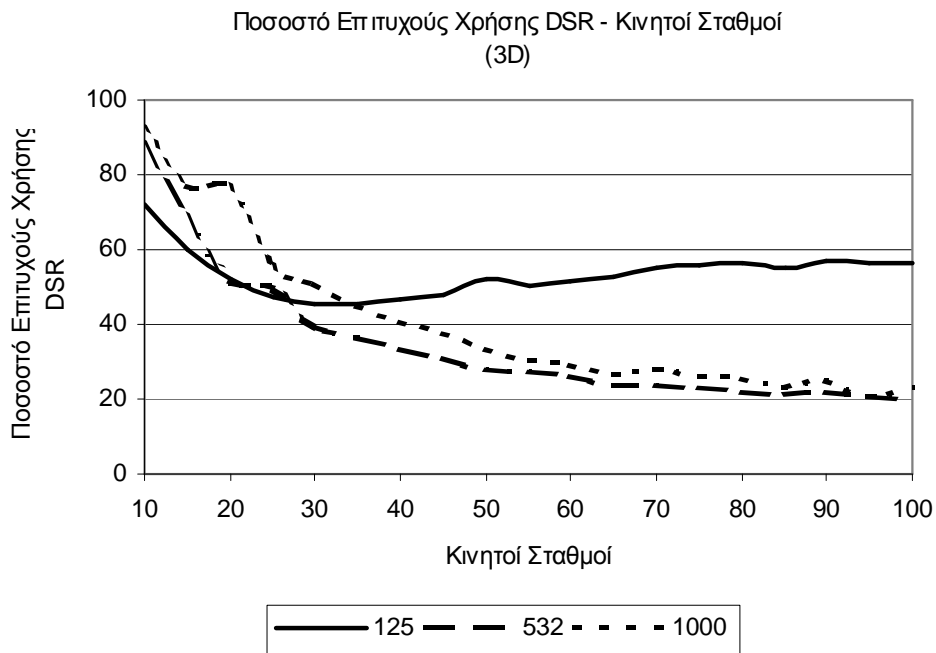
Σχήμα Ε'.17: Μέση Καθυστέρηση Λήψης - Κινητοί Σταθμοί, για τρισδιάστατα γραφήματα (συνδυασμός DSR - SRP)



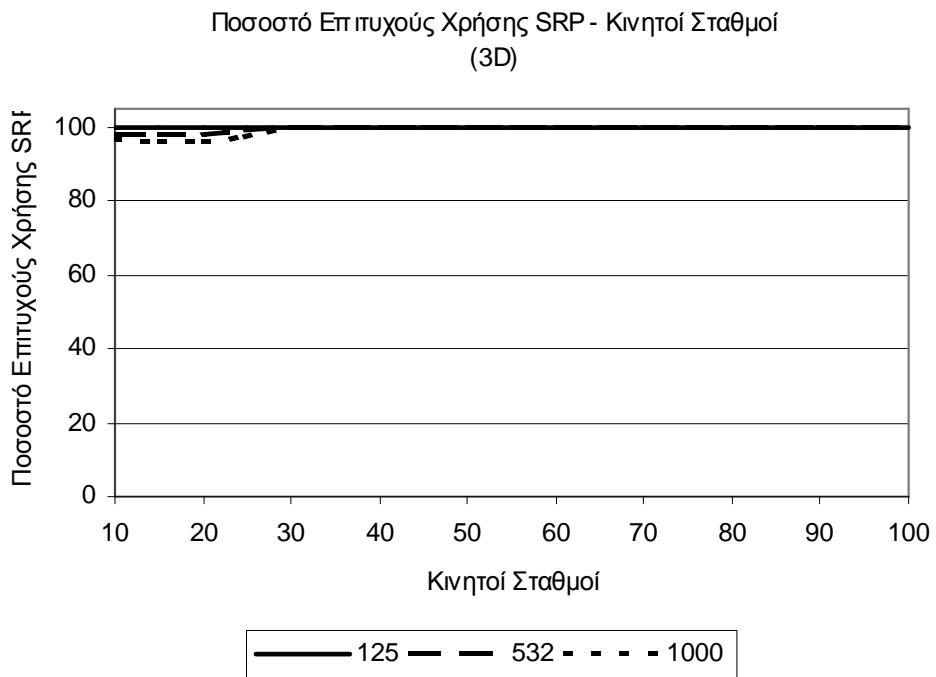
Σχήμα Ε'.18: Ποσοστό Χρήσης DSR - Κινητοί Σταθμοί, για τρισδιάστατα γραφήματα (συνδυασμός DSR - SRP)



Σχήμα Ε'.19: Ποσοστό Χρήσης SRP - Κινητοί Σταθμοί, για τρισδιάστατα γραφήματα (συνδυασμός DSR - SRP)



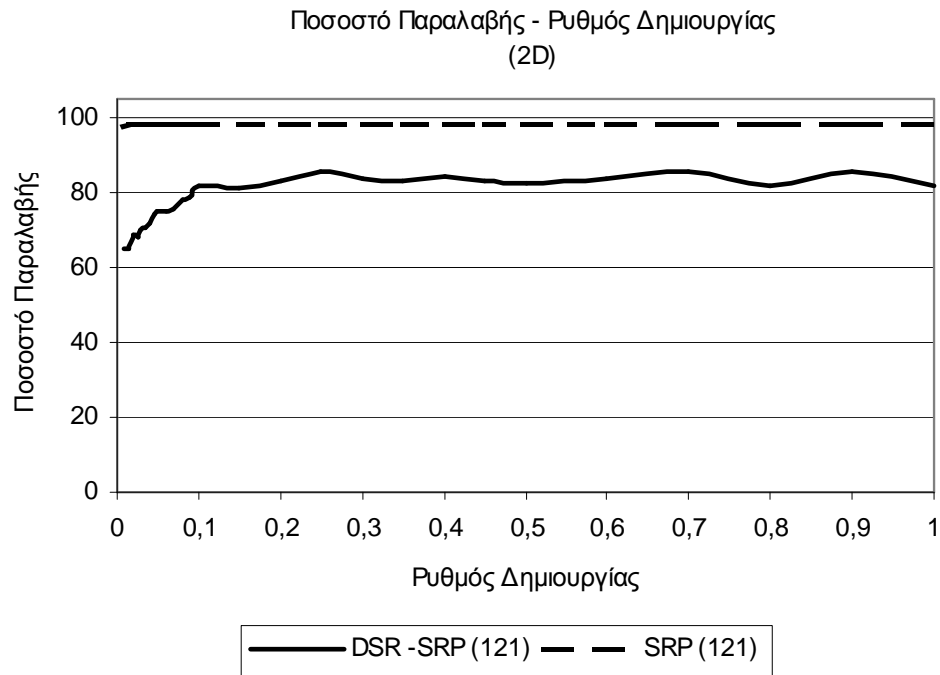
Σχήμα Ε'.20: Ποσοστό Επιτυχούς Χρήσης DSR - Κινητοί Σταθμοί, για τρισδιάστατα γραφήματα (συνδυασμός DSR - SRP)



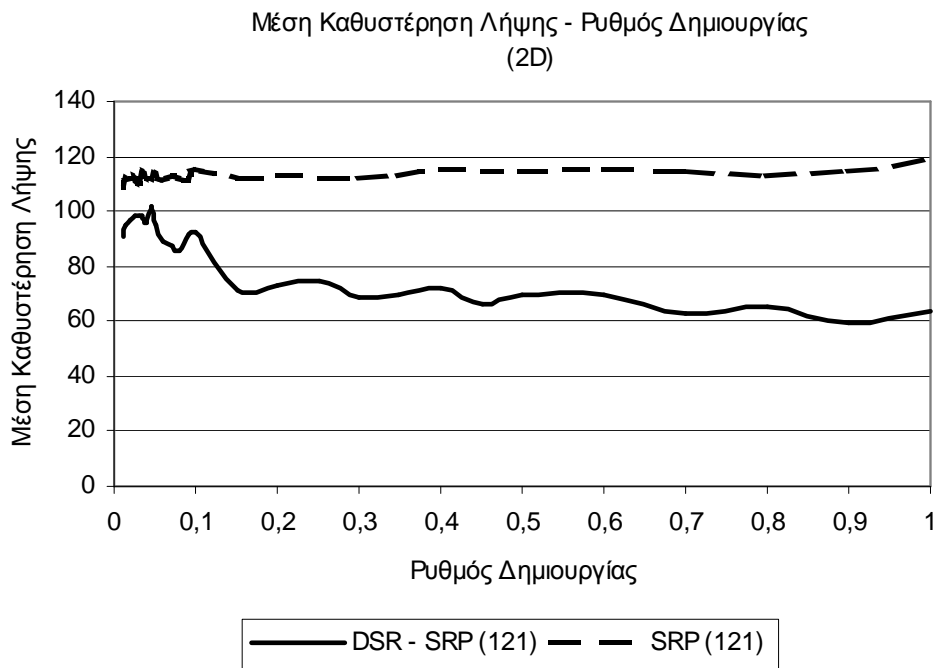
Σχήμα Ε'.21: Ποσοστό Επιτυχούς Χρήσης SRP - Κινητοί Σταθμοί, για τρισδιάστατα γραφήματα (συνδυασμός DSR - SRP)



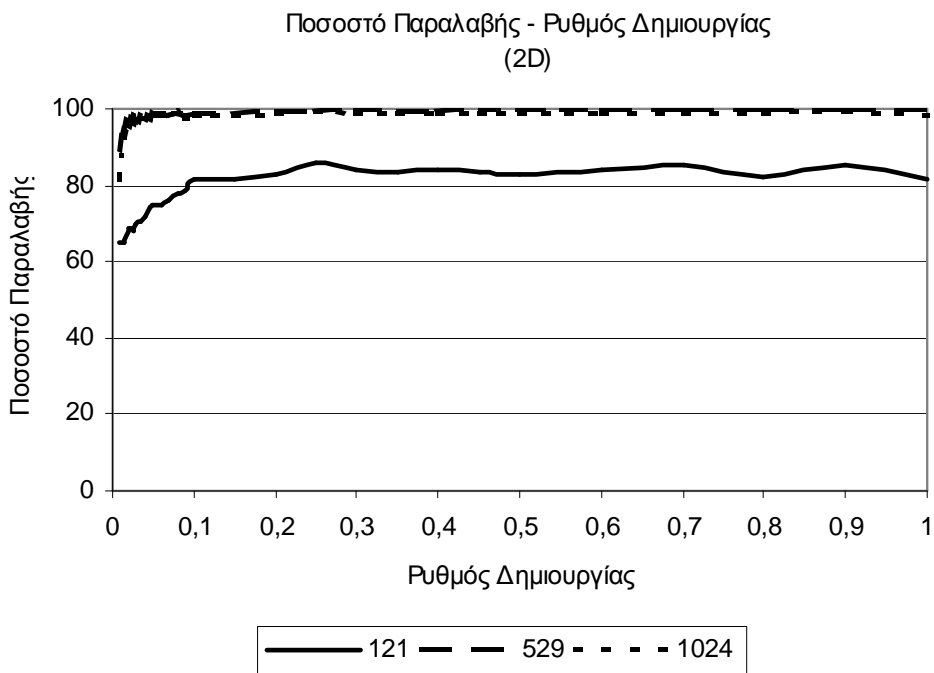
## Ε'.3 Ρυθμός δημιουργίας πακέτων δεδομένων



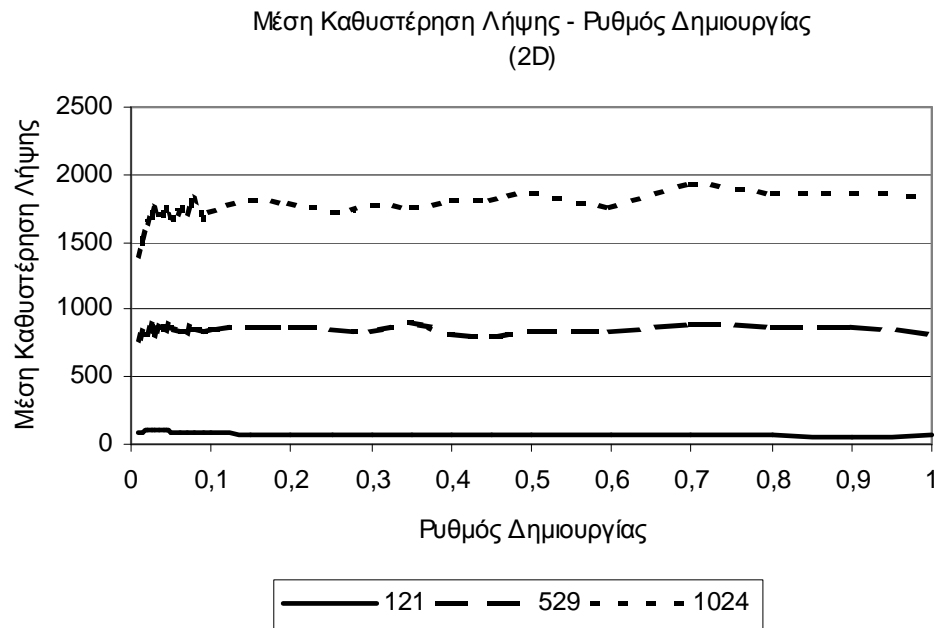
Σχήμα Ε'.22: Ποσοστό Παραλαβής - Ρυθμός Δημιουργίας Πακέτων Δεδομένων, για διδιάστατα γραφήματα (συνδυασμός DSR - SRP, SRP)



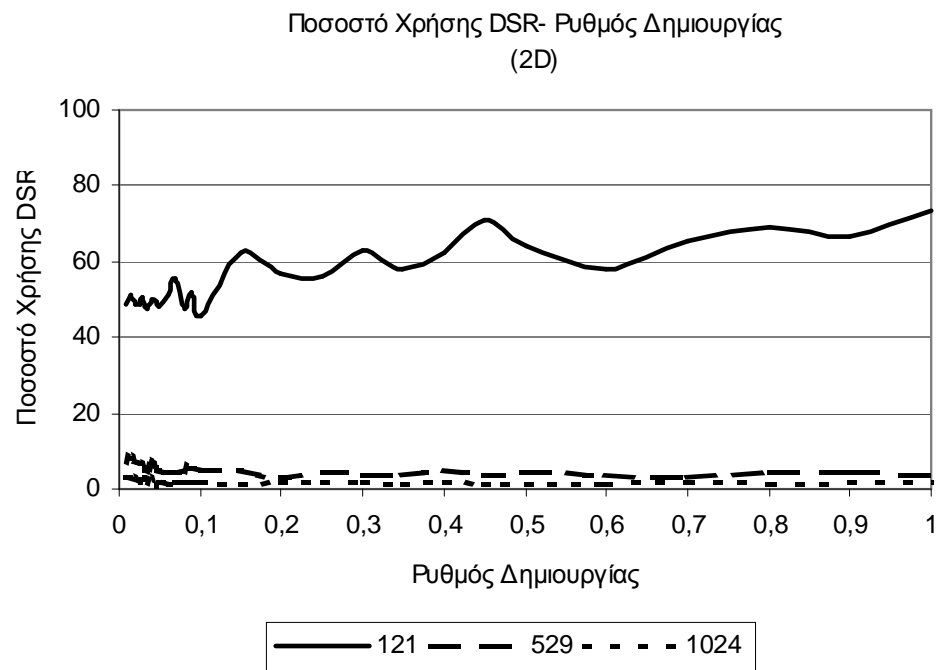
Σχήμα Ε'.23: Μέση Καθυστέρηση Λήψης - Ρυθμός Δημιουργίας Πακέτων Δεδομένων, για δισδιάστατα γραφήματα (συνδυασμός DSR - SRP, SRP)



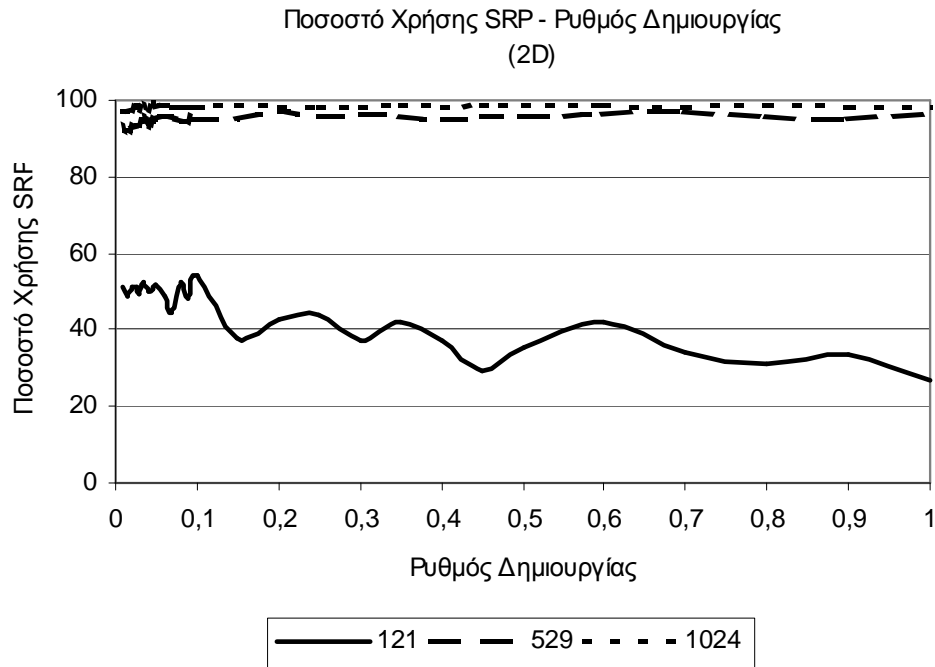
Σχήμα Ε'.24: Ποσοστό Παραλαβής - Ρυθμός Δημιουργίας Πακέτων Δεδομένων, για δισδιάστατα γραφήματα (συνδυασμός DSR - SRP)



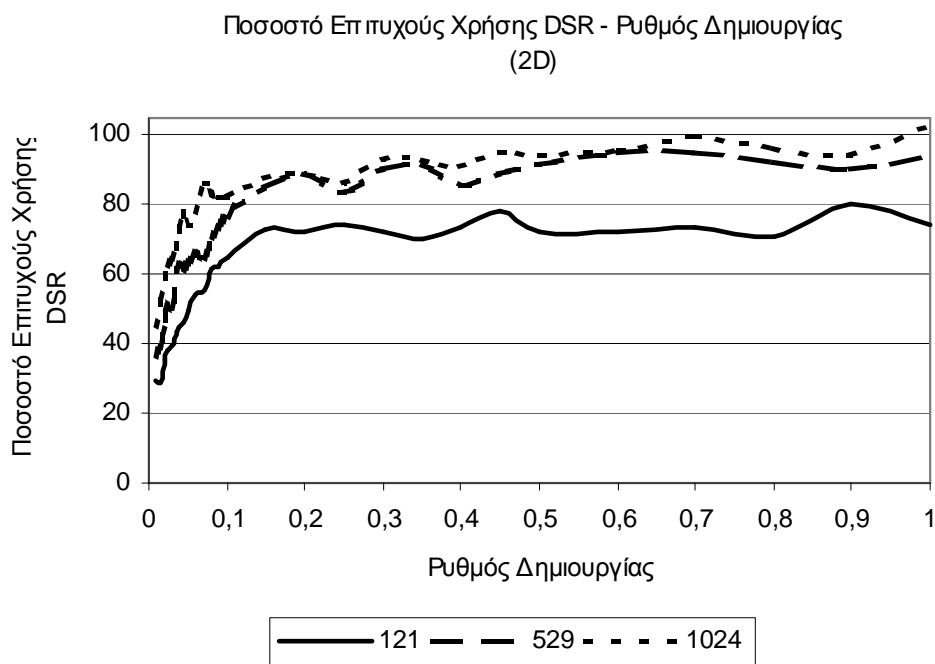
Σχήμα Ε'.25: Μέση Καθυστέρηση Λήψης - Ρυθμός Δημιουργίας Πακέτων Δεδομένων, για δισδιάστατα γραφήματα (συνδυασμός DSR - SRP)



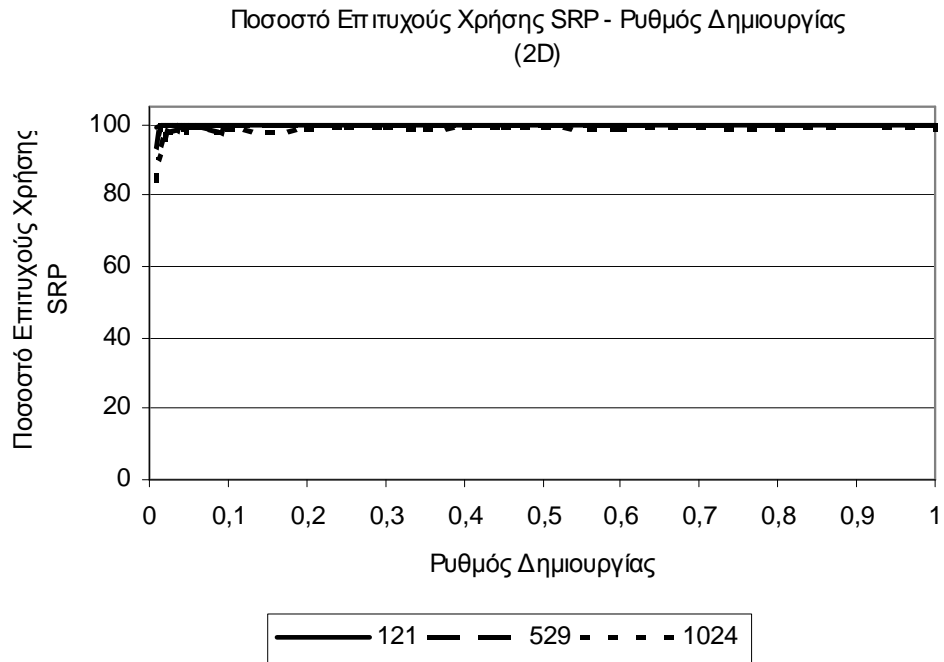
Σχήμα Ε'.26: Ποσοστό Χρήσης DSR - Ρυθμός Δημιουργίας Πακέτων Δεδομένων, για δισδιάστατα γραφήματα (συνδυασμός DSR - SRP)



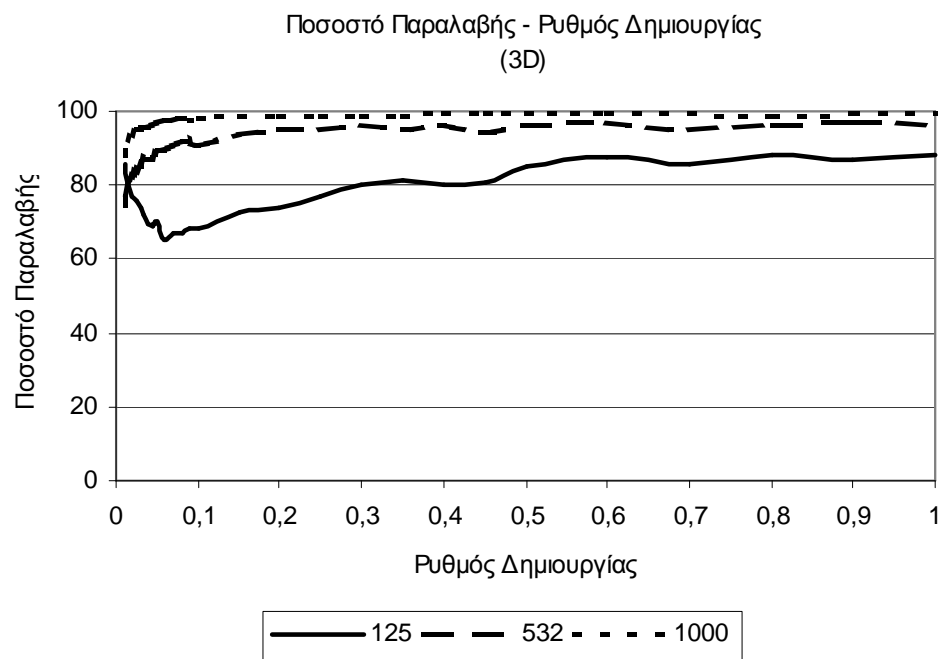
Σχήμα Ε'.27: Ποσοστό Χρήσης SRP - Ρυθμός Δημιουργίας Πακέτων Δεδομένων, για δισδιάστατα γραφήματα (συνδυασμός DSR - SRP)



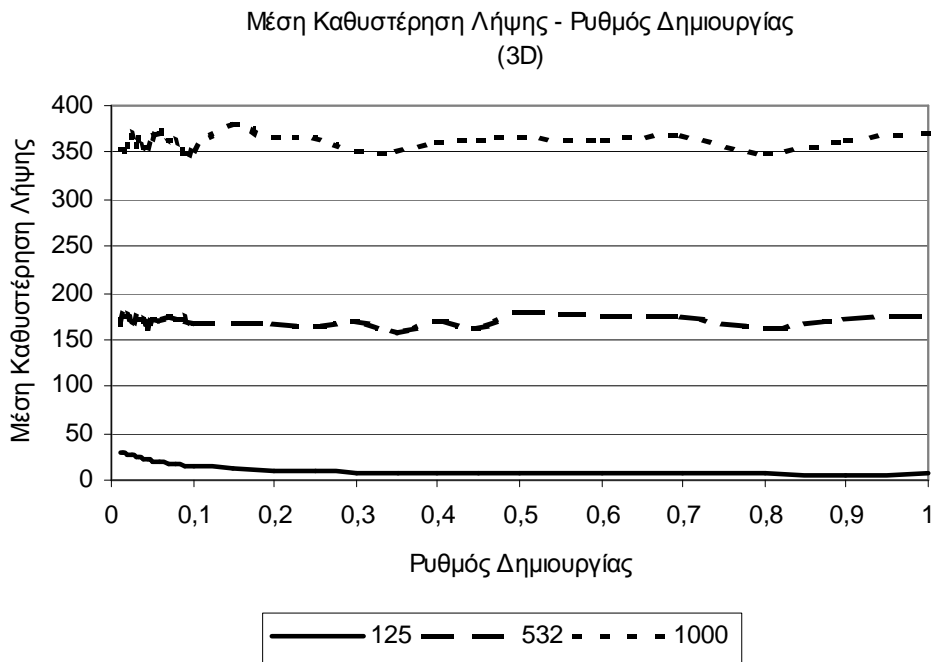
Σχήμα Ε'.28: Ποσοστό Επιτυχούς Χρήσης DSR - Ρυθμός Δημιουργίας Πακέτων Δεδομένων, για δισδιάστατα γραφήματα (συνδυασμός DSR - SRP)



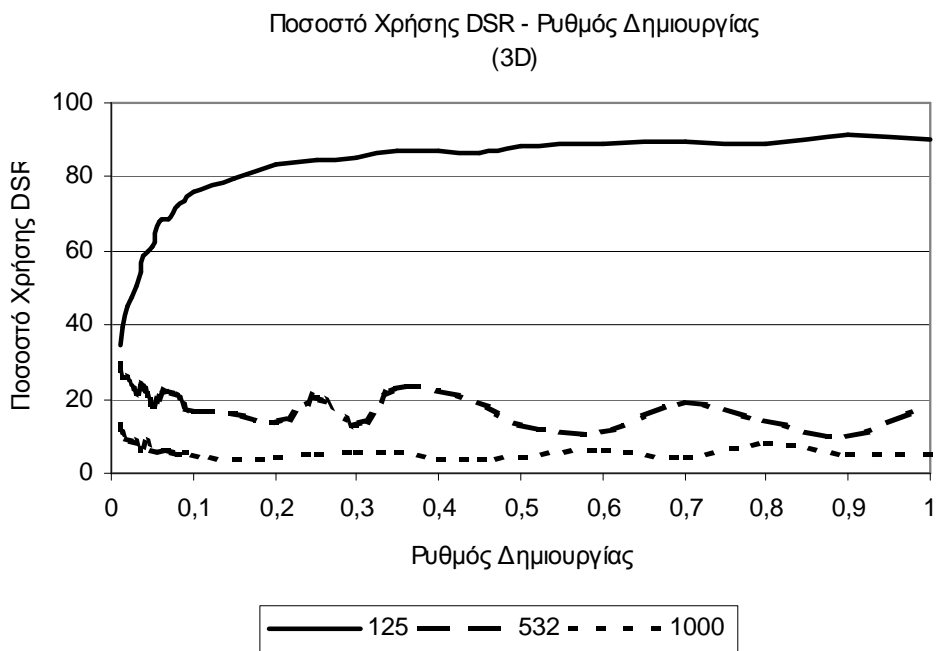
Σχήμα Ε'.29: Ποσοστό Επιτυχούς Χρήσης SRP - Ρυθμός Δημιουργίας Πακέτων Δεδομένων, για δισδιάστατα γραφήματα (συνδυασμός DSR - SRP)



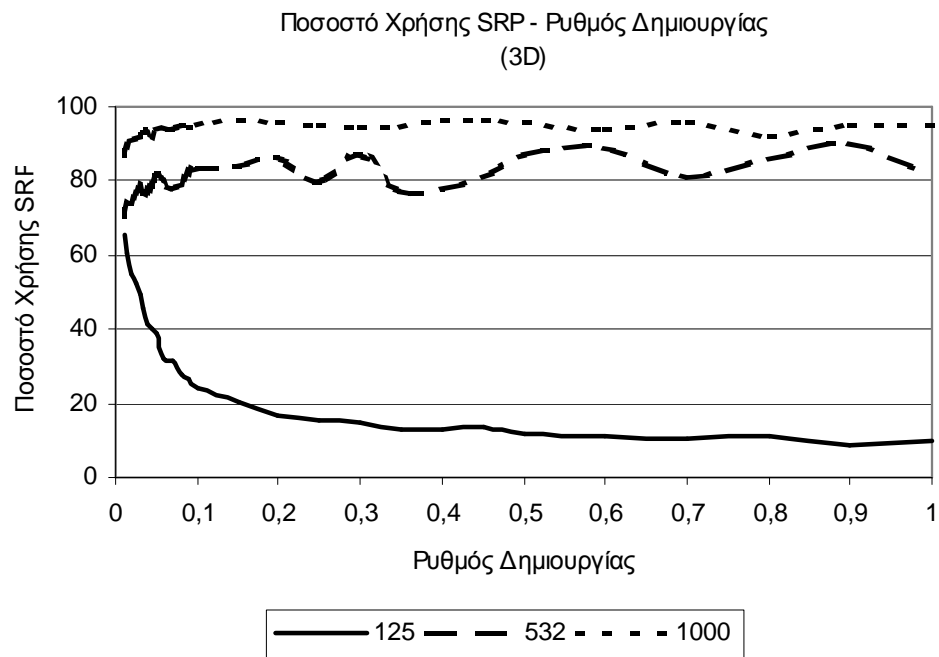
Σχήμα Ε'.30: Ποσοστό Παραλαβής - Ρυθμός Δημιουργίας Πακέτων Δεδομένων, για τρισδιάστατα γραφήματα (συνδυασμός DSR - SRP)



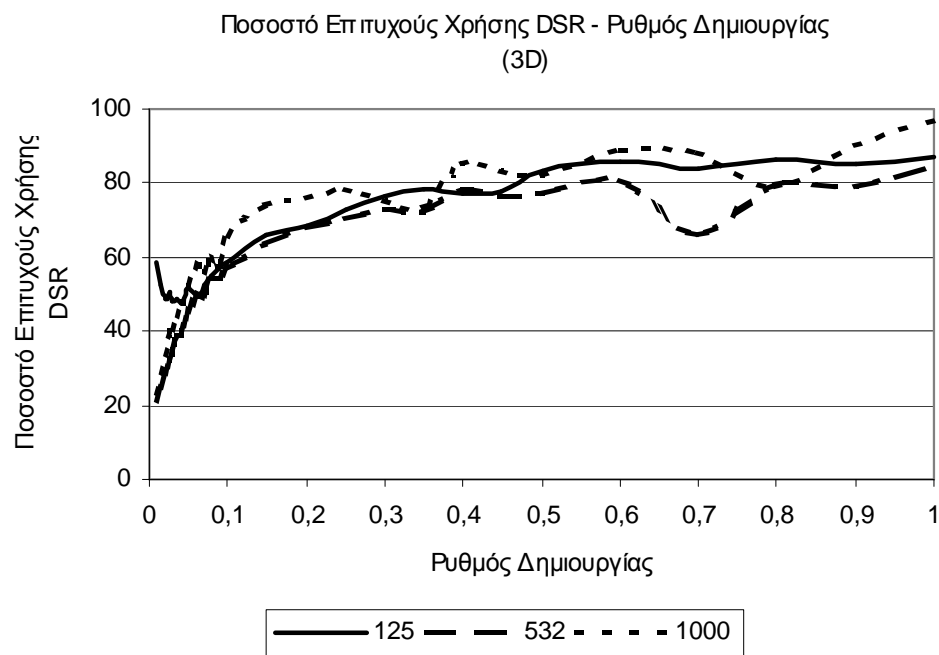
Σχήμα Ε'.31: Μέση Καθυστέρηση Λήψης - Ρυθμός Δημιουργίας Πακέτων Δεδομένων, για τρισδιάστατα γραφήματα (συνδυασμός DSR - SRP)



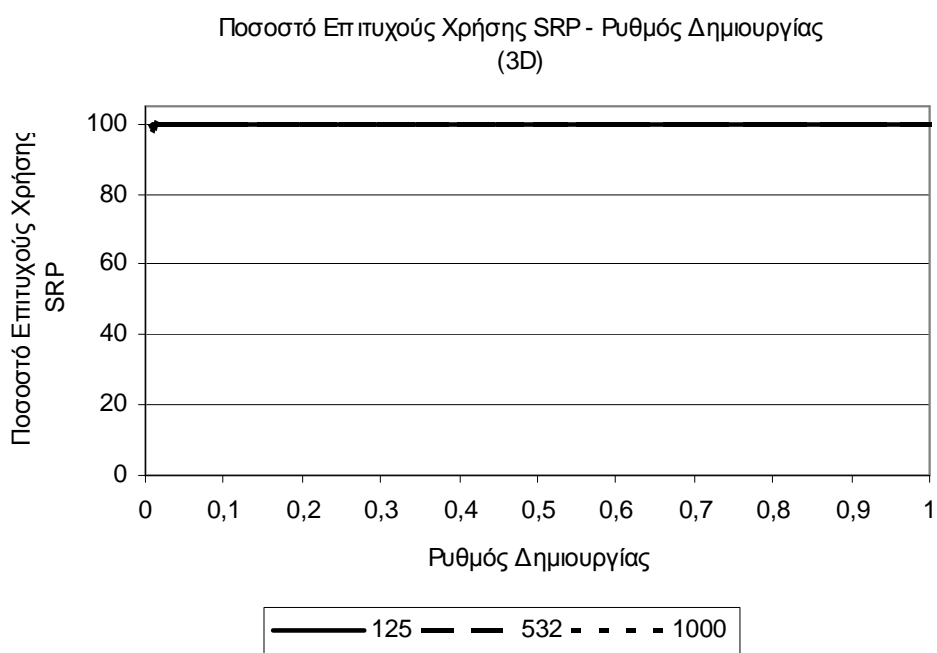
Σχήμα Ε'.32: Ποσοστό Χρήσης DSR - Ρυθμός Δημιουργίας Πακέτων Δεδομένων, για τρισδιάστατα γραφήματα (συνδυασμός DSR - SRP)



Σχήμα Ε'.33: Ποσοστό Χρήσης SRP - Ρυθμός Δημιουργίας Πακέτων Δεδομένων, για τρισδιάστατα γραφήματα (συνδυασμός DSR - SRP)



Σχήμα Ε'.34: Ποσοστό Επιτυχούς Χρήσης DSR - Ρυθμός Δημιουργίας Πακέτων Δεδομένων, για τρισδιάστατα γραφήματα (συνδυασμός DSR - SRP)



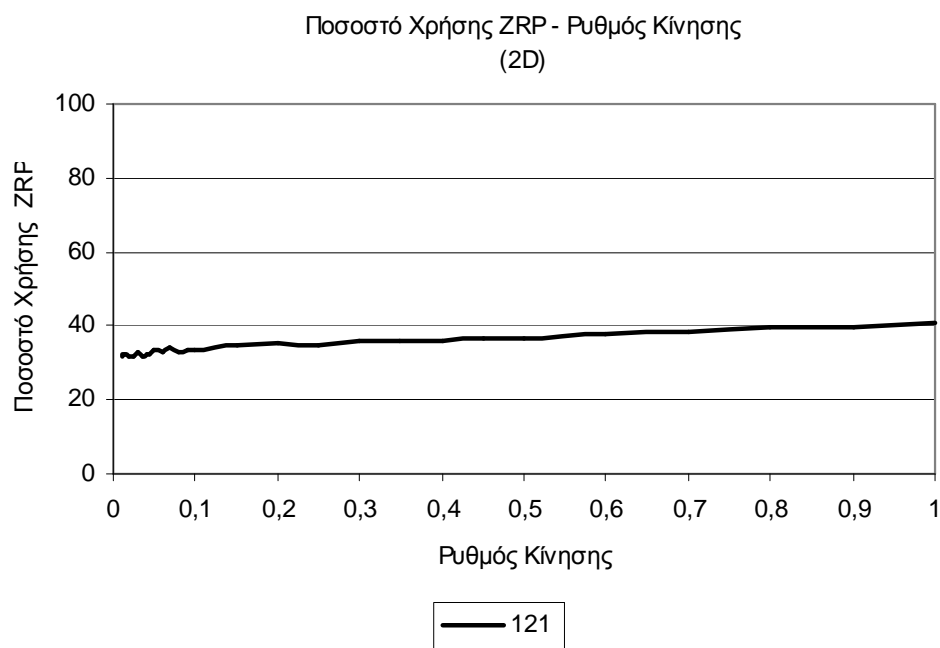
Σχήμα Ε'.35: Ποσοστό Επιτυχούς Χρήσης SRP - Ρυθμός Δημιουργίας Πακέτων Δεδομένων, για τρισδιάστατα γραφήματα (συνδυασμός DSR - SRP)



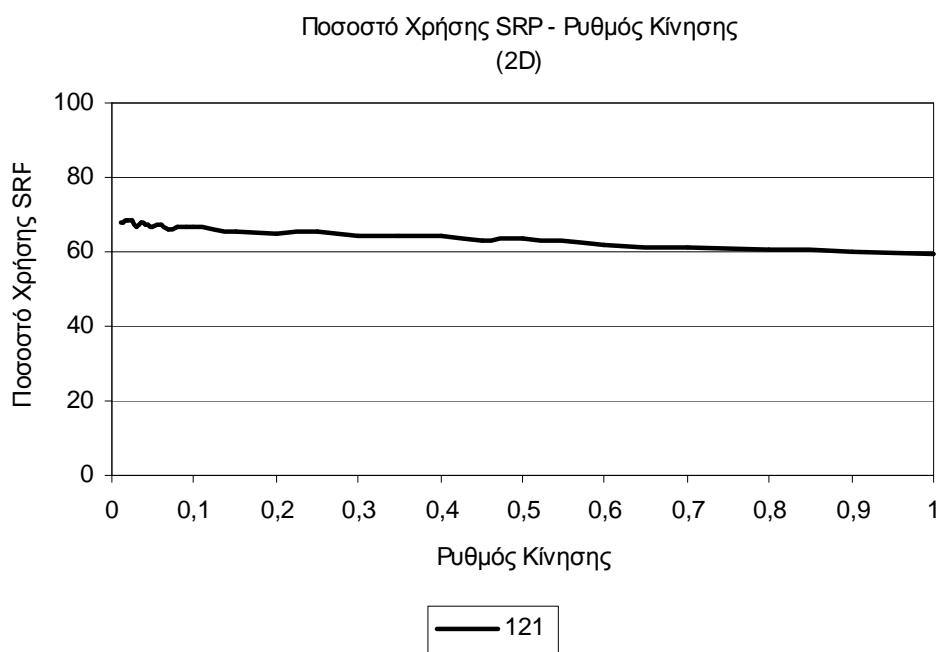
## Παράρτημα ε΄

# Γραφικές παραστάσεις για το συνδυασμό ZRP με το SRP

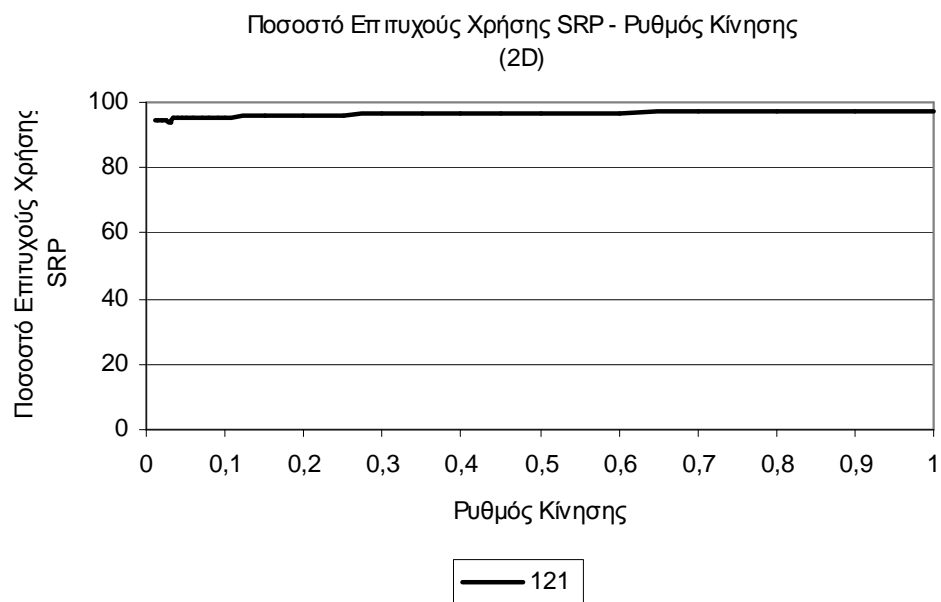
### ε΄.1 Ρυθμός κίνησης



Σχήμα ε΄.1: Ποσοστό Χρήσης ZRP - Ρυθμός Κίνησης, για διδιάστατα γραφήματα (συνδυασμός ZRP - SRP)



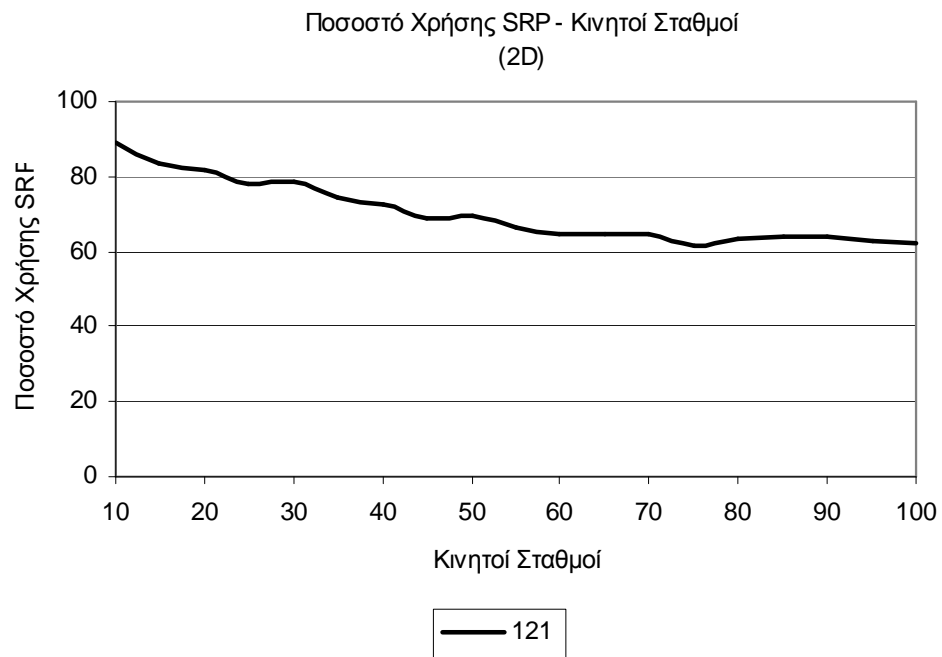
Σχήμα 5'.2: Ποσοστό Χρήσης SRP - Ρυθμός Κίνησης, για διδιάστατα γραφήματα (συνδυασμός ZRP - SRP)



Σχήμα 5'.3: Ποσοστό Επιτυχούς Χρήσης SRP - Ρυθμός Κίνησης, για διδιάστατα γραφήματα (συνδυασμός ZRP - SRP)

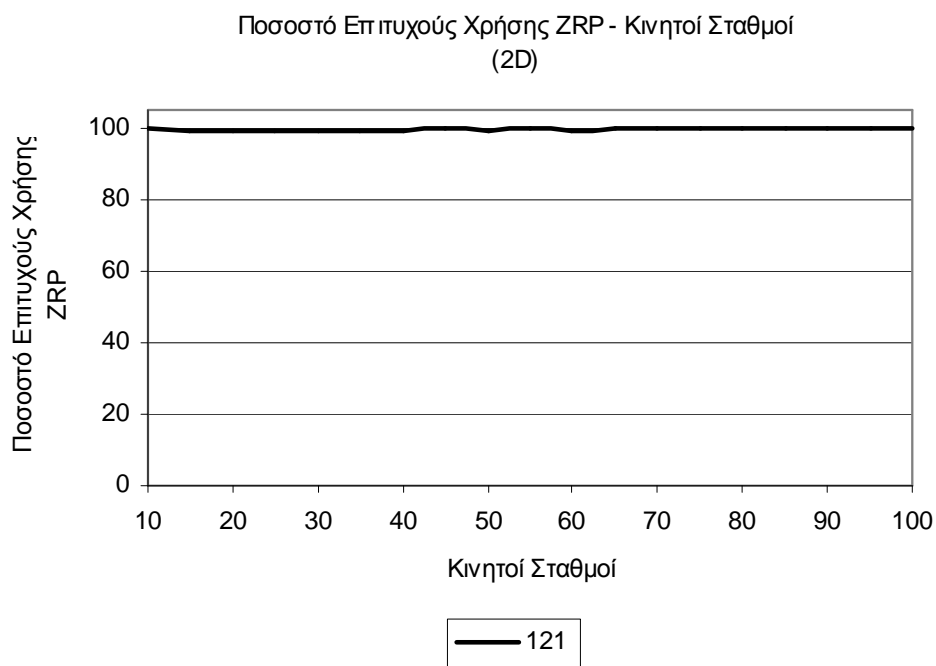
## ε'.2 Κινητοί σταθμοί

1

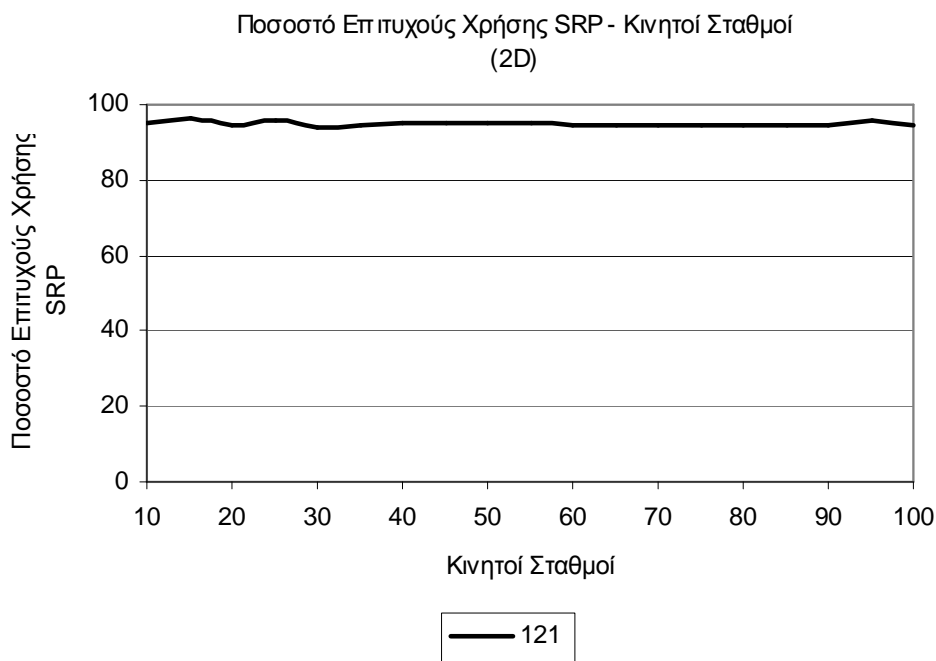


Σχήμα ε'.4: Ποσοστό Χρήσης SRP - Κινητοί Σταθμοί, για δισδιάστατα γραφήματα (συνδυασμός ZRP - SRP)

<sup>1</sup>Στους αριθμούς των κινητών σταθμών δεν συμπεριλαμβάνεται ο αριθμός των σταθμών της υποστήριξης.

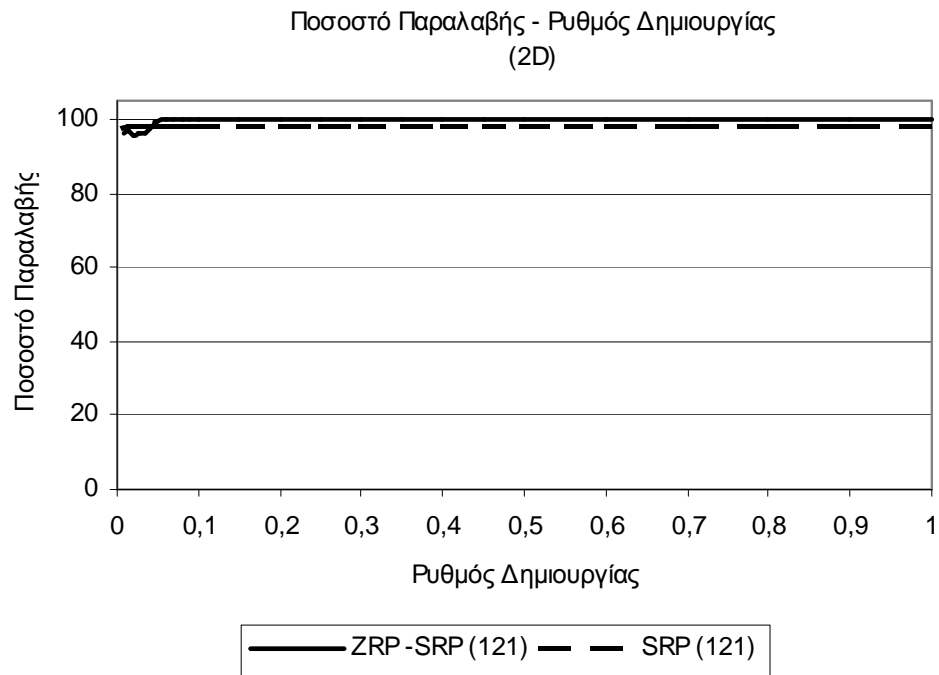


Σχήμα 5: Ποσοστό Επιτυχούς Χρήσης ZRP - Κινητοί Σταθμοί, για δισδιάστατα γραφήματα (συνδυασμός ZRP - SRP)

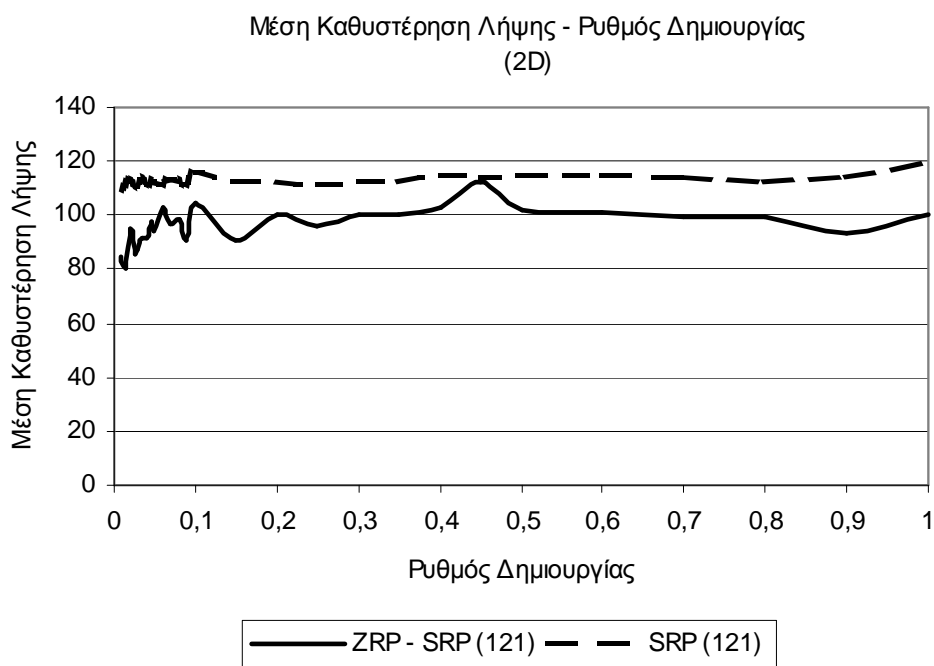


Σχήμα 6: Ποσοστό Επιτυχούς Χρήσης SRP - Κινητοί Σταθμοί, για δισδιάστατα γραφήματα (συνδυασμός ZRP - SRP)

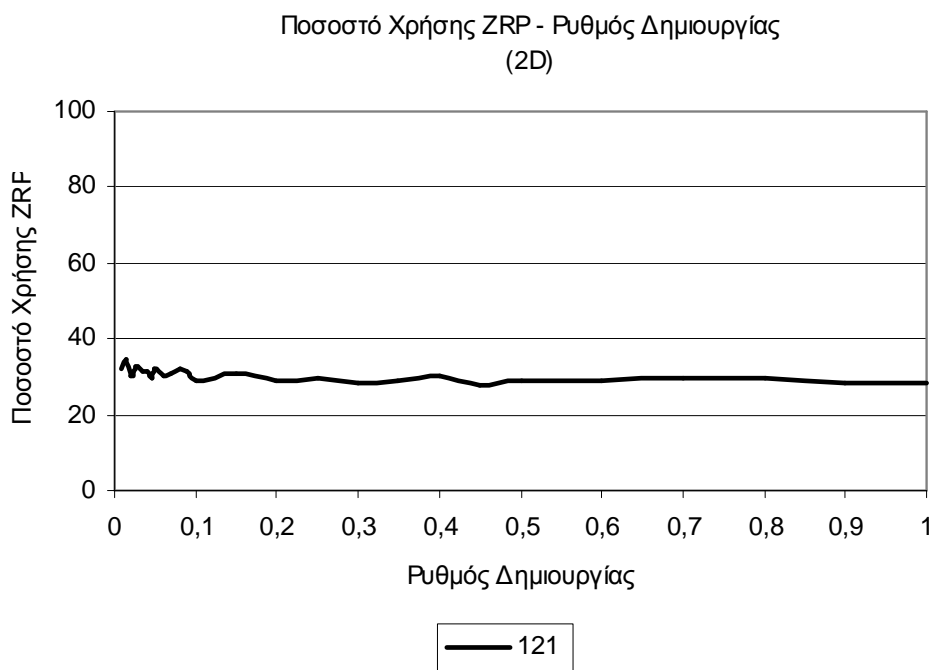
### ε'.3 Ρυθμός δημιουργίας πακέτων δεδομένων



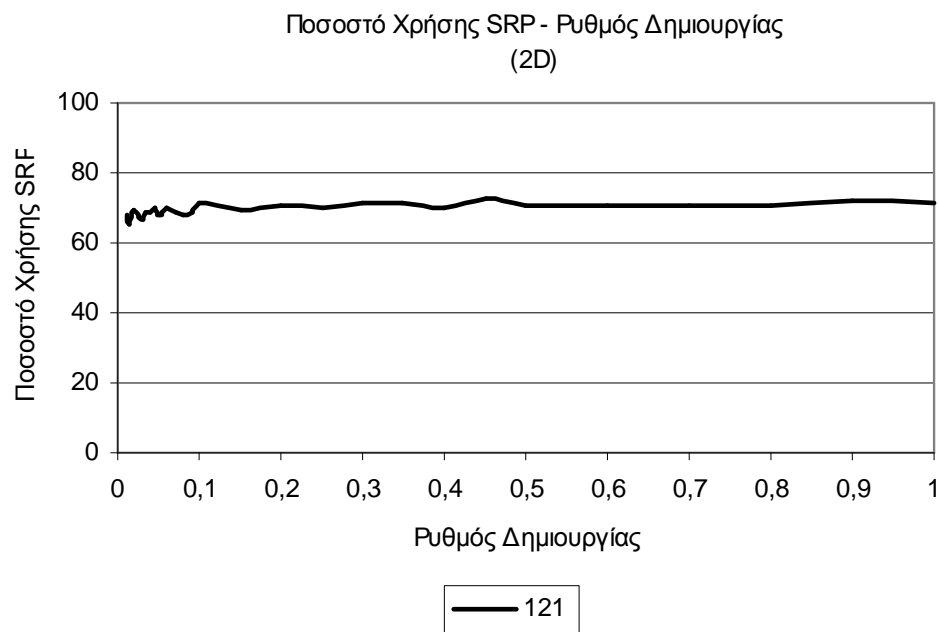
Σχήμα ε'.7: Ποσοστό Παραλαβής - Ρυθμός Δημιουργίας Πακέτων Δεδομένων, για δισδιάστατα γραφήματα (συνδυασμός ZRP - SRP, SRP)



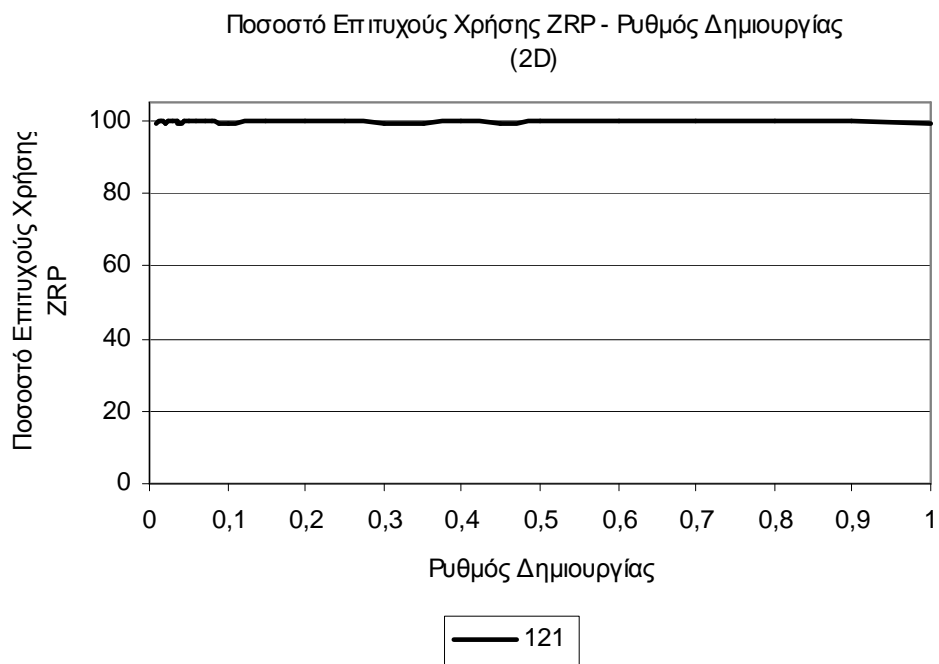
Σχήμα 8: Μέση Καθυστέρηση Λήψης - Ρυθμός Δημιουργίας Πακέτων Δεδομένων, για διδιάστατα γραφήματα (συνδυασμός ZRP - SRP, SRP)



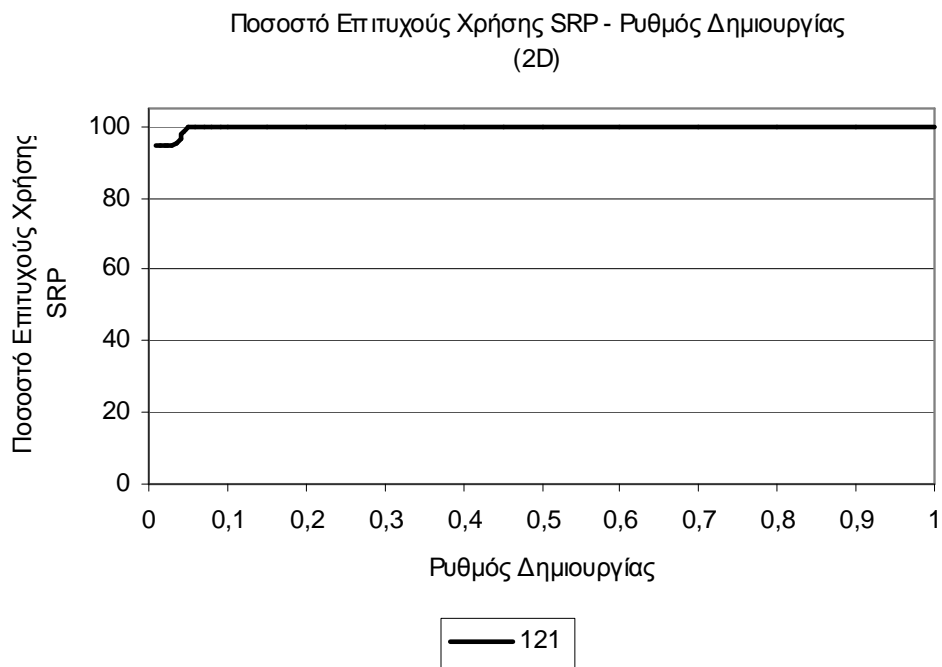
Σχήμα 9: Ποσοστό Χρήσης ZRP - Ρυθμός Δημιουργίας Πακέτων Δεδομένων, για διδιάστατα γραφήματα (συνδυασμός ZRP - SRP)



Σχήμα ε'.10: Ποσοστό Χρήσης SRP - Ρυθμός Δημιουργίας Πακέτων Δεδομένων, για δισδιάστατα γραφήματα (συνδυασμός ZRP - SRP)



Σχήμα ε'.11: Ποσοστό Επιτυχούς Χρήσης ZRP - Ρυθμός Δημιουργίας Πακέτων Δεδομένων, για δισδιάστατα γραφήματα (συνδυασμός ZRP - SRP)



Σχήμα ε'.12: Ποσοστό Επιτυχούς Χρήσης SRP - Ρυθμός Δημιουργίας Πακέτων Δεδομένων, για δισδιάστατα γραφήματα (συνδυασμός ZRP - SRP)



# Παράρτημα Ζ΄

## Συντμήσεις και άλλες παρατηρήσεις

- Οι όροι κινητός σταθμός, σταθμός, κόμβος χρησιμοποιούνται για να περιγράψουν μία συσκευή με δυνατότητες ασύρματης επικοινωνίας, υπολογισμού και κίνησης (ή φορητότητας). Οι όροι αυτοί εμφανίζονται εναλλάξ μέσα στο κείμενο ανάλογα το περιεχόμενο.
- Οι όροι ακμή και βήμα χρησιμοποιούνται για να περιγράψουν την άμεση ασύρματη επικοινωνία δύο σταθμών του δικτύου. Οι όροι αυτοί εμφανίζονται εναλλάξ μέσα στο κείμενο ανάλογα το περιεχόμενο.
- Οι όροι ποσοστό παράδοσης, ποσοστό παραλαβής και ποσοστό πακέτων δεδομένων που παρελήφθησαν χρησιμοποιούνται για να περιγράψουν το ίδιο μετρήσιμο μέγεθος. Οι όροι αυτοί εμφανίζονται εναλλάξ μέσα στο κείμενο ανάλογα το περιεχόμενο.
- ABR: Associativity Based Routing
- AODV: Ad-hoc On Demand Distance Vector
- ASRP: Adaptive Support Routing Protocol
- BRP: Bordercast Resolution Protocol
- CGSR: Clusterhead Gateway Switch Routing
- DSDV: Destination-Sequenced Distance-Vector
- DSR: Dynamic Source Routing
- HASRP: Hierarchical Adaptive Support Routing Protocol

- HSRP: Hierarchical Support Routing Protocol
- IARP: Intrazone Routing Protocol
- IERP: Interzone Routing Protocol
- IETF: Internet Engineering Task Force
- LCC: Least Cluster Change
- MANET: Mobile Ad-hoc Networking
- NDP: Neighbor Discovery Protocol
- SRP: Support Routing Protocol
- TORA: Temporally Ordered Routing Algorithm
- TTL: Time-To-Live
- WEP: Wired Equivalent Privacy
- WLAN: Wireless LAN
- WRP: Wireless Routing Protocol
- ZRP: Zone Routing Protocol
- ΑΔ: Ανακάλυψη Διαδρομής
- ΑΙΔ: Αίτηση Διαδρομής
- ΑΠΔ: Απάντηση Διαδρομής
- ΑΖ: Ακτίνα Ζώνης
- ΕΠ: Ενσωματωμένο Πακέτο
- ΜΠΑΔ: Μνήμη Προσωρινής Αποθήκευσης Διαδρομών
- ΜΠΑΠ: Μνήμη Προσωρινής Αποθήκευσης Πακέτων
- ΠΑΑΔ: Πακέτο Αίτησης Ανακάλυψης Διαδρομής
- ΠΑΚΑ: Πακέτο Κατάστασης Ακμών
- ΠΓ: Πίνακας Γειτόνων
- ΠΔ: Πλήρη Διαδρομή

- ΠΙΚΑ: Πίνακας Κατάστασης Ακμών
- ΠΚ: Προσδιοριστικό Κόμβου
- ΠΚΑ: Πίνακας Κάλυψης Αιτήσεων
- ΣΔ: Συντήρηση Διαδρομής
- ΖΔ: Ζώνη Δρομολόγηση



## Βιβλιογραφία

- [1] I.Chatziannakis and S.Nikolseas, An Adaptive Compulsory Protocol for Basic Communication in Highly Changing Ad-hoc Mobile Networks, In Proc. 2nd IEEE/ACM International Workshop on Parallel and Distributed Computing Issues in Wireless networks and Mobile Computing, IPDPS 2002 Workshops, Fort Lauderdale, Florida, USA, April 15-19, 2002.
- [2] I.Chatziannakis, S.Nikolseas and P.Spirakis, An Efficient Communication Strategy for Ad-hoc Mobile Networks, In Proc. of 15th Symposium on Distributed Computing (DISC'2001), Informatics Department, Faculty of Sciences, University of Lisbon, Portugal, October 2-5, 2001, Lecture Notes in Computer Science, Volume 2180, pp. 285-299 (Springer-Verlag, 2001).
- [3] I.Chatziannakis, S.Nikolseas, N.Paspalis, P.Spirakis and C.Zaroliadis, An Experimental Study of Basic Communication Protocols in Ad-hoc Mobile Networks, In Proc. of 5th Workshop on Algorithmic Engineering (WAE'2001), BRICS, University of Aarhus, Denmark, August 28-30, 2001, Lecture Notes in Computer Science, Volume 2141, pp. 159-171 (Springer-Verlag, 2001).
- [4] I.Chatziannakis, S.Nikolseas and P.Spirakis, Analysis and Experimental Evaluation of an Innovative and Efficient Routing Protocol for Ad-hoc Mobile Networks, In Proc. of 4th Workshop on Algorithmic Engineering (WAE'2000), Max-Planck-Institute for Computer Science, Saarbr'ocken, Germany, September 5-8, 2000. Lecture Notes in Computer Science, Volume 1982, pp. 99-110 (Springer-Verlag, 2000).
- [5] S. Corson, J. Macker, Routing Protocol Performance Issues and Evaluation Considerations, Mobile Ad-hoc Networking (MANET), January 1999.

- 
- [6] Z.J. Haas, M.R. Pearlman and P. Samar, Zone Routing Protocol (ZRP), IETF Internet Draft, July 2002.
  - [7] Z.J. Haas, M.R. Pearlman and P. Samar, Intrazone Routing Protocol (IARP), IETF Internet Draft, July 2002.
  - [8] Z.J. Haas, M.R. Pearlman and P. Samar, Bordercasting Resolution Protocol (BRP), IETF Internet Draft, July 2002.
  - [9] Z.J. Haas, M.R. Pearlman and P. Samar, Interzone Routing Protocol (IERP), IETF Internet Draft, July 2002.
  - [10] D.B. Johnson, D.A. Maltz, Yih-Chun Hu, The Dynamic Source Routing Protocol for Mobile Ad-Hoc Networks (DSR), IETF Internet Draft, April 2003.
  - [11] S. Murthy and J.J. Garcia-Luna-Aceves, A Routing Protocol for Packet Radio Networks, Proc. ACM Mobile Computing and Networking, November 14-15 1995.
  - [12] M.R. Pearlman and Z.J. Haas, Determining the Optimal Configuration for the Zone Routing Protocol, IEEE JSAC, August 1999.
  - [13] C.E. Perkins, Ad-hoc networking, Addison-Wesley, 2001.
  - [14] Ιωάννης Χατζηγιαννάκης, Κατανεμημένοι Αλγόριθμοι για Θεμελιώδη Ζητήματα Επικοινωνίας σε Ad-hoc Κινητά Δίκτυα, Διδακτορική Διατριβή, Τμήμα Μηχανικών Ηλεκτρονικών Υπολογιστών και Πληροφορικής, Πανεπιστήμιο Πατρών, Μάιος 2003.