

**ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΣΕΡΡΩΝ**

**ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ**

**ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΩΝ**

**ΜΕΛΕΤΗ ΤΗΣ ΣΥΜΦΟΡΗΣΗΣ ΤΟΥ TCP ΠΡΩΤΟΚΟΛΛΟΥ ΣΤΑ  
AD-HOC ΑΣΥΡΜΑΤΑ ΔΙΚΤΥΑ ΜΕ ΤΗΝ ΧΡΗΣΗ ΤΟΥ NETWORK  
SIMULATOR 2**

**Πτυχιακή Εργασία των**

ΠΕΓΓΙΟΥ ΙΩΑΝΝΑ (353)

ΧΑΤΖΗΔΗΜΗΤΡΙΟΥ ΕΙΡΗΝΗ-ΜΑΡΙΑ (467)

Επιβλέπων: Α.Χ. ΠΟΛΙΤΗΣ

## Περιεχόμενα

Πίνακας Περιεχομένων.....	1
<b>1 ΤΟΠΙΚΑ ΑΣΥΡΜΑΤΑ ΔΙΚΤΥΑ.....</b>	<b>4</b>
1.1 Εισαγωγή .....	4
1.2 Σύγκριση WLAN – LAN.....	4
1.3 Εφαρμογές ασύρματων δικτύων.....	7
1.4 Πρωτόκολλα δρομολόγησης στα ad hoc δίκτυα.....	8
<b>2 ΠΡΟΤΥΠΟ 802.11.....</b>	<b>11</b>
2.1 Εισαγωγή.....	11
2.2 Συμπληρωματικά Πρότυπα .....	11
2.3 Διαστρωμάτωση.....	12
2.4 Φυσικό στρώμα.....	13
2.4.1 MAC.....	14
<b>3 TCP/IP.....</b>	<b>16</b>
3.1 Εισαγωγή.....	16
3.2 ΔΙΑΣΤΡΩΜΑΤΩΣΗ.....	17
3.3 Υπηρεσίες TCP.....	19
3.3.1 Η υπηρεσία του TCP στις εφαρμογές.....	20
3.3.2 Υπηρεσία με βάση την σύνδεση.....	21
3.3.3 Υπηρεσία Επικοινωνία σημείο προς σημείο (Point-to -Point)....	22
3.3.4 Υπηρεσία Πλήρης Αξιοπιστία.....	22
3.3.4.1 Λήξη χρονομετρητή και Επαναμετάδοση - Retransmission TimeOut (RTO).....	22
3.3.4.2 Επιβεβαίωσης Λήψης – Duplicate Acknowledgements .....	24
3.4 Έλεγχος Συμφόρησης.....	25
3.4.1 Αργή εκκίνηση - Slow Start .....	26
3.4.2 Γρήγορη Επαναμετάδοση και Γρήγορη Ανάκτηση – Fast Retransmit and Fast Recovery.....	27
3.4.3 Αλγόριθμος Αποφυγής Συμφόρησης - Congestion Avoidance Algorithm.....	28
3.5 Εκδόσεις του TCP.....	29
3.6 Μορφή του τμήματος TCP.....	31

<b>4 NETWORK SIMULATOR ( NS -2 )</b> .....	<b>34</b>
4.1 Εισαγωγή.....	34
4.2 Λεπτομερή περιγραφή.....	34
4.3 Τι είναι η TCL.....	36
4.4 Η TCL σε ενσύρματα δίκτυα.....	37
4.5 Η TCL σε ασύρματα δίκτυα.....	40
4.6 Υποστήριξη ιχνηλάτισης (Trace support).....	43
4.6.1 Μορφή Trace File.....	43
4.7 NETWORK ANIMATOR (NAM).....	47
4.8 Οδηγίες για την εγκατάσταση και εκτέλεση του Προσομοιωτή Network Simulator 2 σε περιβάλλον windows. ....	50
4.8.1 Για την εγκατάσταση των εκτελέσιμων αρχείων.....	50
4.8.2 Εκτέλεση των προγραμμάτων προσομοίωσης .....	51
<b>5 ΠΕΙΡΑΜΑΤΙΚΟ ΜΕΡΟΣ</b> .....	<b>54</b>
5.1 Εισαγωγή.....	54
5.2 Πρώτο Σενάριο.....	54
5.2.1 Γενική Περιγραφή.....	54
5.2.2 Αρχείο εξόδου.....	55
5.2.3 Επεξεργασία των αποτελεσμάτων.....	56
5.2.3.1 Επεξεργασία του cwnd.....	56
5.2.3.2 Επεξεργασία του throughput.....	58
5.2.3.3 Μετρήσεις.....	59
5.3 Δεύτερο Πείραμα.....	59
5.3.1 Γενική Περιγραφή.....	59
5.3.2 Αρχείο εξόδου.....	61
5.3.3 Επεξεργασία των αποτελεσμάτων.....	62
5.3.3.1 Επεξεργασία του cwnd.....	62
5.3.3.2 Επεξεργασία του throughput.....	64
5.3.3.3 Μετρήσεις.....	64
5.4 Τρίτο σενάριο.....	65
5.4.1 Γενική Περιγραφή.....	65
5.4.2 Αρχείο εξόδου.....	67
5.4.3 Επεξεργασία των αποτελεσμάτων.....	68

5.4.3.1 Επεξεργασία του cwnd.....	68
5.4.3.2 Επεξεργασία του throughput.....	71
5.4.3.3 Μετρήσεις.....	71
5.5 Τέταρτο Σενάριο.....	72
5.5.1 Γενική Περιγραφή.....	72
5.5.2 Αρχείο εξόδου.....	73
5.5.3 Επεξεργασία των αποτελεσμάτων.....	74
5.5.3.1 Επεξεργασία του cwnd.....	74
5.5.3.2 Επεξεργασία του throughput.....	76
5.5.3.3 Μετρήσεις.....	76
5.6 Παρατηρήσεις.....	77
<b>6 ΠΑΡΑΡΤΗΜΑ Α.....</b>	<b>80</b>
6.1 Κώδικας φίλτρου AWK.....	80
6.2 Κώδικας Πρώτου Σεναρίου.....	81
6.3 Κώδικας Δεύτερου Σεναρίου.....	84
6.4 Κώδικας Τρίτου Σεναρίου.....	87
6.5 Κώδικας τέταρτου σεναρίου.....	91
<b>Βιβλιογραφία.....</b>	<b>94</b>

# ΚΕΦΑΛΑΙΟ 1

## ΤΟΠΙΚΑ ΑΣΥΡΜΑΤΑ ΔΙΚΤΥΑ

### 1.1 Εισαγωγή

Τα ασύρματα τοπικά δίκτυα (Wireless Local Networks – WLAN) είναι μία σχετικά νέα μορφή τοπικών δικτύων που λειτουργεί συμπληρωματικά ή εναλλακτικά του κοινού ενσύρματου δικτύου (Ethernet). Δεν αποσκοπεί στην αντικατάσταση του. Επιτρέπει την επέκταση της γεωγραφικής κάλυψης του προϋπάρχοντος δικτύου χωρίς να απαιτούνται καλώδια. Τα WLAN είναι κατάλληλα για τη σύνδεση χρηστών σε οποιοδήποτε περιοχή υπάρχει η κατάλληλη υποδομή και τεχνολογία. Η δυνατότητα πρόσβασης κατά την κίνηση, που προσφέρεται στους χρήστες, είναι το κύριο χαρακτηριστικό που διαφοροποιεί τα ασύρματα από τα ενσύρματα δίκτυα. Υπάρχουν πολλά πρότυπα ασύρματης δικτύωσης αλλά το πλέον διαδεδομένο και εμπορικό θεωρείται το 802.11, το οποίο ανέπτυξε το *Institute of Electrical and Electronics Engineers*. [1]

### 1.2 Σύγκριση WLAN – LAN

Σε σχέση με τα ενσύρματα τοπικά δίκτυα (Local Area Networks – LANs) τα WLAN παρουσιάζουν πλεονεκτήματα αλλά και αρκετούς περιορισμούς, που προσδιορίζουν σε μεγάλο βαθμό τον τρόπο χρήσης τους. Στη συνέχεια αναφέρονται τα βασικότερα πλεονεκτήματά τους: [2]

- *Η δυνατότητα πρόσβασης κατά την κίνηση (Mobility) χρηστών:*  
Οι χρήστες έχουν δυνατότητα πρόσβασης, σε πραγματικό χρόνο, ακόμη και αν εκείνοι βρίσκονται εν κινήσει. Για να το

εκμεταλλευτούν πρέπει φυσικά να διαθέτουν το αντίστοιχο κινητό τερματικό (για παράδειγμα laptop με κάρτα ασύρματης δικτύωσης).

- *Ευκολία, ταχύτητα εγκατάστασης και επεκτασιμότητα:* Σε αντίθεση με τα ενσύρματα δίκτυα δεν λαμβάνεται υπ' όψιν η κτιριακή δομή και δεν απαιτούνται μεγάλες παρεμβάσεις στην περιοχή λειτουργίας, όπως είναι η εγκατάσταση καλωδίωσης. Αυτό έχει ως αποτέλεσμα την εύκολη επέκτασή τους, εφόσον το μέσο μετάδοσης που χρησιμοποιούν είναι παντού διαθέσιμο.
- *Κόστος:* Μπορεί το αρχικό κόστος για τον εξοπλισμό ενός ασύρματου τοπικού δικτύου να είναι συγκριτικά ακριβότερο από αυτό ενός ενσύρματου, ωστόσο τα οφέλη είναι μακροπρόθεσμα. Αυτό συμβαίνει κυρίως σε περιπτώσεις δυναμικών χώρων εργασίας που απαιτούν συχνές αλλαγές, καθώς το κόστος επαναδιαμόρφωσης του προϋπάρχοντος ασύρματου δικτύου θα είναι αμελητέο.

Οι βασικότεροι περιορισμοί στην εγκατάσταση και λειτουργία των ασυρμάτων δικτύων είναι οι εξής:

- *Κατανάλωση ισχύος:* Για να εκμεταλλευτούν οι χρήστες την κινητικότητα που τους προσφέρει το ασύρματο δίκτυο πρέπει να χρησιμοποιούν κινητούς σταθμούς (mobile stations). Αυτοί λειτουργούν με μπαταρίες και ο σχεδιασμός του δικτύου πρέπει να τους επιτρέπει όσο το δυνατόν μεγαλύτερη αυτονομία.
- *Ρυθμοροή (throughout):* Ιδανικά η διέλευση των ασύρματων δικτύων θα έπρεπε να είναι περίπου ίση με τη διέλευση των ενσύρματων. Αυτό δε συμβαίνει στην πράξη, λόγω περιορισμών που επιβάλλει η ασύρματη μετάδοση. Αν και έχει

παρατηρηθεί αρκετά μεγάλη αύξηση των ρυθμών μετάδοσης, η διαφορά είναι ακόμα μεγάλη. Το πρωτόκολλο πρόσβασης στο μέσο του ασυρμάτου δικτύου πρέπει να φροντίζει για την επίτευξη μέγιστης διέλευσης στο δίκτυο.

- *Παρεμβολές και αξιοπιστία:* Παρεμβολές μπορεί να προέρχονται από τους ίδιους τους σταθμούς του δικτύου στην προσπάθειά τους να μεταδώσουν ταυτόχρονα. Επίσης μπορεί να προέρχονται από άλλες συσκευές που χρησιμοποιούν το ίδιο φασματικό εύρος. Τέλος, πηγή παρεμβολών μπορεί να είναι το φαινόμενο της ανάκλασης και της διάθλασης του σήματος. Τα παραπάνω πρέπει να αντιμετωπιστούν με χρήση κατάλληλων τεχνικών διαμόρφωσης, κωδικοποίησης και διόρθωσης λαθών.
- *Ασφάλεια επικοινωνιών:* τα ασύρματα δίκτυα υστερούν στον τομέα παρεχόμενης ασφάλειας, καθώς υπάρχουν πολλοί τρόποι επίθεσης από επίδοξους εισβολείς. Δεδομένα που κυκλοφορούν σε ένα ασύρματο δίκτυο είναι εύκολο να υποκλαπούν από οποιονδήποτε, αρκεί να διαθέτει τον κατάλληλο δέκτη και πρόσβαση στην περιοχή κάλυψης του δικτύου. Γι' αυτόν το λόγο πρέπει να χρησιμοποιείται κάποια μέθοδος κρυπτογράφησης των εκπεμπόμενων δεδομένων.
- *Υποστήριξη φορητότητας:* Το ασύρματο δίκτυο πρέπει να υποστηρίζει τη δρομολόγηση της κίνησης σε κινούμενους χρήστες. Αυτό προσθέτει πολυπλοκότητα στη σχεδίασή του.
- *Κατανομή συχνοτήτων:* Πρέπει να βρεθούν οι φασματικές περιοχές στις οποίες θα λειτουργούν τα διάφορα ασύρματα δίκτυα. Αυτό μπορεί να είναι αρκετά δύσκολο, ιδίως όταν στη διαδικασία εμπλέκονται ρυθμιστικές αρχές διαφόρων χωρών.

- *Ασφάλεια χρηστών:* Η ασφάλεια των χρηστών κατά τη χρήση κάθε είδους ασύρματων συσκευών είναι ένα θέμα που μελετάται διαρκώς. Στα ασύρματα δίκτυα ένας από τους λόγους περιορισμού της εκπεμπόμενης ισχύος είναι η προστασία των χρηστών.

### 1.3 Εφαρμογές ασύρματων δικτύων

Τα ασύρματα δίκτυα έχουν σήμερα τέσσερις βασικές εφαρμογές. **[5]**

- *Επέκταση των ενσύρματων LAN:* Τα ασύρματα δίκτυα μπορούν να χρησιμοποιηθούν για τη διασύνδεση των χρηστών με το βασικό κορμό (backbone) του ενσύρματου δικτύου, όταν η ύπαρξη καλωδίωσης είναι δύσκολο και οικονομικά ασύμφορο να εγκατασταθεί μέχρι τον τελικό χρήστη.
- *Διασύνδεση μεταξύ κτιρίων:* Εναλλακτικά της χρήσης ενσύρματης δικτύωσης για την διασύνδεση μεταξύ κτιρίων μπορεί να χρησιμοποιηθεί η τεχνολογία ασύρματων ζεύξεων. Για την οποία οι συσκευές που συνδέονται στα δύο άκρα της ζεύξης είναι συνήθως δρομολογητές (routers) ή γέφυρες (bridges) ή ζεύξη οπτικής επαφής όπου χρησιμοποιούνται laser και φωτοανιχνευτές.
- *Σποραδική πρόσβαση στο δίκτυο:* Ασύρματα δίκτυα μπορούν να εγκατασταθούν σε χώρους όπου κινούνται χρήστες για να προσφέρουν πρόσβαση στο **υπάρχων ενσύρματο δίκτυο**. Σημαντικό θέμα σε αυτήν την περίπτωση είναι φυσικά η ασφάλεια των δεδομένων.



- *Δημιουργία δικτύων άμεσης σύνδεσης (Ad-hoc):* Τα Ad-hoc δίκτυα συνήθως δημιουργούνται για να ικανοποιήσουν άμεσα μία συγκεκριμένη ανάγκη. Τέτοια δίκτυα μπορούν να χρησιμοποιηθούν για παράδειγμα σε συνεδριακούς χώρους ή σε αίθουσες διδασκαλίας, οπότε οι συμμετέχοντες μπορούν να ανταλλάσσουν δεδομένα μέσω του **προσωρινού ασυρμάτου δικτύου**, χωρίς να απαιτείται οποιαδήποτε εκ των προτέρων διαμόρφωση του χώρου .

#### **1.4 Πρωτόκολλα δρομολόγησης στα ad hoc δίκτυα**

Στα ad hoc δίκτυα οι κόμβοι είναι κινητοί και μπορούν να συνδεθούν δυναμικά. Όλοι οι κόμβοι αυτών των δικτύων συμπεριφέρονται ως δρομολογητές και συμμετέχουν στην ανακάλυψη και τη συντήρηση των διαδρομών σε άλλους κόμβους στο δίκτυο.

Τα ad hoc πρωτόκολλα δρομολόγησης μπορούν να διαιρεθούν σε δύο κατηγορίες:

1. «Οδηγούμενα από Πίνακα», “Table-driven” πρωτόκολλα δρομολόγησης.

Τα οδηγούμενα από πίνακα πρωτόκολλα δρομολόγησης, διατηρούν σε κάθε κόμβο συνεπείς και ενημερωμένες πληροφορίες δρομολόγησης προς όλους τους κόμβους.

2. «Δυναμικά» “On demand” πρωτόκολλα δρομολόγησης.

Στα δυναμικά πρωτόκολλα δρομολόγησης, οι διαδρομές δημιουργούνται όταν και όπως ζητηθεί. Όταν μια πηγή θέλει να

στείλει σε έναν προορισμό, επικαλείται τους μηχανισμούς ανακαλύψεων διαδρομών για να βρει την πορεία στον προορισμό.

Τα τελευταία χρόνια έχουν αναπτυχθεί ποικίλα νέα πρωτόκολλα δρομολόγησης. Υπάρχουν τέσσερα ασύρματα ειδικά πρωτόκολλα δρομολόγησης δικτύων πολλαπλών-αναπηδήσεων (multi-hop) που καλύπτουν μια σειρά των επιλογών σχεδίου:

1. Destination-Sequenced Distance-Vector (DSDV)
2. Temporally Ordered Routing Algorithm (TORA)
3. Dynamic Source Routing (DSR)
4. Ad Hoc On-Demand Distance Vector Routing (AODV).

Το DSDV είναι ένα «Οδηγούμενο από Πίνακα» πρωτόκολλο δρομολόγησης, ενώ τα TORA, DSR και AODV, βρίσκονται στην κατηγορία των «Κατόπιν διαταγής» πρωτοκόλλων δρομολόγησης.

### **DSDV**

Σε αυτό το πρωτόκολλο δρομολόγησης, τα μηνύματα δρομολόγησης ανταλλάσσονται μεταξύ των γειτονικών κόμβων (γειτονικοί κόμβοι θεωρούνται οι κόμβοι που είναι ο ένας στην εμβέλεια του άλλου). Ανανεώσεις των πινάκων δρομολόγησης γίνονται σε περίπτωση που αλλάξει κάτι στον πίνακα δρομολόγησης ενός γειτονικού κόμβου. Τα πακέτα, τα οποία έχουν άγνωστο προορισμό, φυλάσσονται μέχρι ο κόμβος να λάβει σχετικές πληροφορίες. Υπάρχει φυσικά ένα μέγιστο μέγεθος για τον buffer, που φυλάσσει τα πακέτα αυτά, πέρα από το οποίο τα πακέτα αρχίζουν και απορρίπτονται.

## **DSR (Dynamic Source Routing)**

Πρόκειται για ένα δυναμικό πρωτόκολλο δρομολόγησης. Ελέγχει για κάθε εισερχόμενο πακέτο τί πληροφορίες δρομολόγησης έχει και το προωθεί ανάλογα. Σε περίπτωση που δεν υπάρχουν πληροφορίες δρομολόγησης για κάποιο πακέτο, το βάζει σε αναμονή και ρωτάει τους υπόλοιπους κόμβους.

## **TORA**

Είναι ένα πρωτόκολλο δρομολόγησης βασισμένο σε έναν αλγόριθμο " link reversal " ο οποίος παρέχει αξιόπιστη παράδοση των μηνυμάτων δρομολόγησης (route-messaging) και πληροφορεί τους κόμβους για οποιαδήποτε αλλαγή συμβεί στους γειτονικούς τους κόμβους. Σχεδιάστηκε για να ανακαλύπτει τις διαδρομές δυναμικά, να παρέχει πολλαπλούς δρομολογητές προς έναν προορισμό, να καθιερώνει τις διαδρομές γρήγορα, και να ελαχιστοποιεί το κόστος της επικοινωνίας.

## **AODV**

Αποτελεί έναν συνδυασμό των DSR και DSDV πρωτοκόλλων. Έχει τους βασικούς μηχανισμούς εύρεσης των κατάλληλων μονοπατιών. Υιοθετεί έναν πολύ διαφορετικό μηχανισμό για να διατηρήσει τις πληροφορίες δρομολόγησης. Χρησιμοποιεί τους παραδοσιακούς πίνακες δρομολόγησης, μια καταχώρηση ανά προορισμό.

## ΚΕΦΑΛΑΙΟ 2

### ΠΡΟΤΥΠΟ 802.11

#### 2.1 Εισαγωγή

Σε ένα τοπικό δίκτυο είναι δυνατόν το φυσικό μέσο επικοινωνίας να διατίθεται σε έναν μόνο κόμβο (σύνδεση σημείο προς σημείο) ή σε όλους τους κόμβους του δικτύου (σύνδεση εκπομπής). Οι συνδέσεις καθορίζονται από κοινά αποδεκτούς κανόνες επικοινωνίας, οι οποίοι αναφέρονται είτε στις διάφορες τερματικές διατάξεις ενός τοπικού δικτύου, είτε στα ίδια τα τοπικά δίκτυα όταν αυτά διασυνδέονται με άλλα δίκτυα των ίδιων ή διαφορετικών προδιαγραφών. Κάθε πρωτόκολλο επικοινωνίας (communication protocol) είναι μία συμφωνία που καθορίζει την μορφή και την σημασία των μηνυμάτων που ανταλλάσσουν οι υπολογιστές. Οι εφαρμογές που χρησιμοποιούν ένα δίκτυο δεν αλληλεπιδρούν απευθείας με το υλικό του δικτύου. Μια εφαρμογή αλληλεπιδρά με το λογισμικό πρωτοκόλλων, το οποίο ακολουθεί τους κανόνες ενός δεδομένου πρωτοκόλλου όταν πραγματοποιείται επικοινωνία.

#### 2.2 Συμπληρωματικά Πρότυπα

Για την επικοινωνία στα ασύρματα τοπικά δίκτυα χρησιμοποιείται το πρότυπο 802.11 που ανακοινώθηκε από την IEEE επίσημα το 1997. Στη συνέχεια ανακοινώθηκαν τα εξής συμπληρωματικά πρότυπα:

**802.11a:** παρέχει μετάδοση μέχρι 54 Mbps στη ζώνη 5GHz. Λιγότερο δυναμικό για παρεμβολή σε ραδιοσυχνότητα από το 802.11b και το 802.11g. Σχετικά μικρότερη εμβέλεια (περίπου 60 μέτρα) από το 802.11b. Δεν είναι συμβατό με το 802.11b.

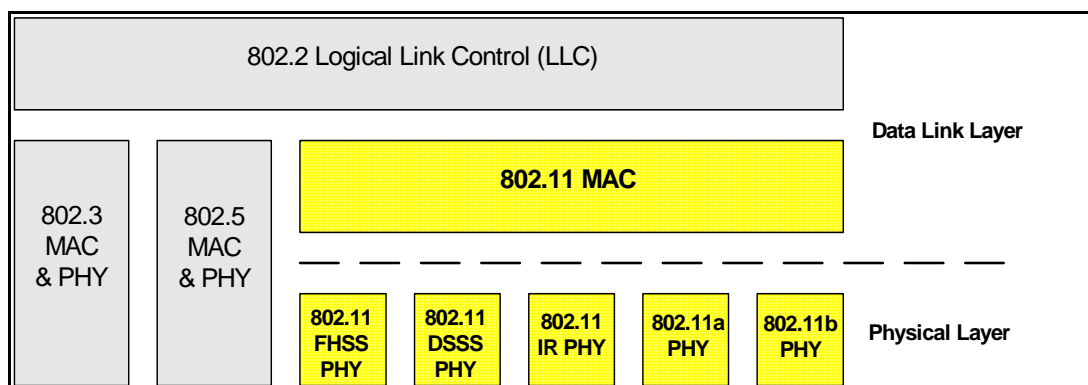
**802.11b:** παρέχει μετάδοση 11 Mbps στη ζώνη 2.4 GHz. Δεν είναι συμβατό με το 802.11a. Προσφέρει πρόσβαση σε δεδομένα σε απόσταση μέχρι 100 μέτρα από το σταθμό βάσης.

**802.11g:** παρέχει μετάδοση μέχρι 54 Mbps (τυπικά 22 Mbps) στη ζώνη 2.4 GHz. Θεωρείται ότι είναι ο διάδοχος και συμβατός με το πρότυπο 802.11b. Προσφέρει πρόσβαση υψηλής ταχύτητας σε δεδομένα σε απόσταση μέχρι 100 μέτρα από το σταθμό βάσης.

### 2.3 Διαστρωμάτωση

Το πρότυπο 802.11 καλύπτει τις προδιαγραφές για το υπό-επίπεδο MAC (Medium Access Control), του επιπέδου σύνδεσης δεδομένων (Data Link Layer), και το φυσικό επίπεδο (Physical), του μοντέλου διαστρωμάτωσης (OSI). Ουσιαστικά, ένα μοντέλο διαστρωμάτωσης περιγράφει έναν τρόπο με τον οποίο το πρόβλημα της επικοινωνίας μπορεί να υποδιαιρεθεί σε μέρη, τα οποία λέγονται επίπεδα (layers).

Το άλλο υπόστρωμα του επιπέδου σύνδεσης δεδομένων, δηλαδή το υπόστρωμα ελέγχου λογικής ζεύξης (Logical Link Control – LLC), είναι αυτό που έχει προτυποποιηθεί ως IEEE 802.2 και χρησιμοποιείται σε συνδυασμό με όλα τα διαφορετικά MAC της σειράς IEEE 802, όπως φαίνεται στο Σχήμα 2.1.



**Σχήμα 2.1** Διαστρωμάτωση του προτύπου 802.11

## 2.4 Φυσικό στρώμα

Αποτελεί το πρώτο επίπεδο του μοντέλου αναφοράς OSI. Είναι υπεύθυνο για την μετατροπή των δυαδικών ψηφίων που περιλαμβάνει από τον δέκτη σε σήμα κατάλληλο για μετάδοση από το μέσο επικοινωνίας, την μετάδοση τους και την επαναφορά τους σε δυαδική μορφή. Τα χαρακτηριστικά του δικτύου που ορίζει το φυσικό επίπεδο αφορούν κυρίως το χρησιμοποιούμενο κανάλι επικοινωνίας. Οι υπηρεσίες που προσφέρει είναι η ενεργοποίηση και απενεργοποίηση της φυσικής σύνδεσης όπως επίσης η μεταφορά των δεδομένων σε μορφή δυαδικού ψηφίου και επισημαίνει τα σφάλματα στην μετάδοση.

Όπως αναφέρθηκε και παραπάνω, τρία φυσικά στρώματα είχαν οριστεί αρχικά για το πρότυπο 802.11. Αυτά είναι τα εξής:

- Direct Sequence Spread Spectrum (Απλωμένο Φάσμα Ευθείας Ακολουθίας) στην ISM (Industrial, Scientific, Medical) μπάντα των 2,4 GHz με ρυθμούς μετάδοσης 1 και 2 Mbps. Η τεχνική Direct Sequence είναι η πιο επιτυχημένη τεχνική που έχει χρησιμοποιηθεί σε συνδυασμό με τα ασύρματα δίκτυα. Το μεγάλο

πλεονέκτημά της είναι ότι μπορεί εύκολα να αναβαθμιστεί για την επίτευξη υψηλότερων ρυθμών μετάδοσης.

➤ Frequency Hopping Spread Spectrum (Απλωμένο Φάσμα και Αναπήδηση Συχνότητας) στην ISM (Industrial, Scientific, Medical) μπάντα των 2,4 GHz με ρυθμούς μετάδοσης 1 και 2 Mbps. Το φυσικό στρώμα αυτό ήταν το πρώτο που χρησιμοποιήθηκε ευρέως σε εμπορικά προϊόντα. Πλεονεκτήματά του είναι τα απλούστερα και φθηνότερα ηλεκτρονικά για την υλοποίηση των ανάλογων συσκευών, η χαμηλότερη κατανάλωση ενέργειας και η δυνατότητα συνύπαρξης πολλών τέτοιων δικτύων στην ίδια περιοχή χωρίς να επηρεάζεται η συνολική διέλευση.

➤ Infrared (Υπέρυθρες Ακτίνες) σε μήκη κύματος μεταξύ 850 και 950 nm με ρυθμούς μετάδοσης 1 και 2 Mbps. Το φυσικό στρώμα υπέρυθρων ακτινών (Infrared – IR) δεν χρησιμοποιείται ιδιαίτερα. Η λειτουργία του βασίζεται στην εκπομπή παλμών διάρκειας 250 nsec, που παράγονται από τα LEDs (Light Emitting Diode) του πομπού. Η ακτίνα λειτουργίας του μπορεί να φτάσει περίπου τα 20 μέτρα, σε ελεύθερο φυσικά οπτικό πεδίο. Άλλη περίπτωση είναι η ανάκλαση των υπέρυθρων ακτινών από κατάλληλη επιφάνεια.

### 2.4.1 MAC

Το υπόστρωμα MAC του 802.11 είναι ίσως το πιο σημαντικό κομμάτι της προτυποποίησης. Αναλαμβάνει την επικοινωνία με το φυσικό επίπεδο. Σκοπός του είναι να ελέγχει την αξιοπιστία της ροής της πληροφορίας από και προς τον κόμβο στον οποίο βρίσκεται. Οι όποιες διαφοροποιήσεις του από το αντίστοιχο MAC ενσύρματων δικτύων οφείλονται στις ιδιαιτερότητες του ασύρματου μέσου μετάδοσης που χρησιμοποιείται στο φυσικό επίπεδο.

Σαν μηχανισμός πρόσβασης στο μέσο έχει επιλεχτεί η Πολλαπλή Πρόσβαση μέσω Ανίχνευσης Φέροντος Σήματος με αποφυγή Συγκρούσεων (CSMA /CA, Carrier Sense Multiple Access with Collision Avoidance). Αντί να βασίζεται στην λήψη όλων των μεταδόσεων από όλους τους άλλου υπολογιστές, ο μηχανισμός CSMA/CA που χρησιμοποιείται στα ασύρματα LAN προκαλεί μια σύντομη μετάδοση από τον αποστολέα πριν μεταδοθεί ένα πακέτο.

Η αξιόπιστη μεταφορά δεδομένων μεταξύ των διαφόρων κόμβων δυσχεραίνεται ακόμα περισσότερο εξαιτίας του ασύρματου φυσικού μέσου. Το MAC του 802.11 προσφέρει τους κατάλληλους μηχανισμούς, όπως η θετική επιβεβαίωση (positive acknowledgment) κάθε πλαισίου και την ανταλλαγή πλαισίων RTS (Ready To Send) και CTS (Clear To Send), πριν την μετάδοση κάποιου πλαισίου. Τα πλαίσια RTS (Ready To Send) και CTS (Clear To Send) βελτιώνουν την απόδοση της χρήσης του ασύρματου δικτύου σε περιπτώσεις μεγάλου φόρτου εξαιτίας της ύπαρξης πολλών τερματικών. Αν όμως χρησιμοποιείται χωρίς λόγο, έχει το ακριβώς αντίθετο αποτέλεσμα, αφού προσθέτει επιπλέον φόρτο στο ασύρματο δίκτυο.



## ΚΕΦΑΛΑΙΟ 3

### TCP/IP

#### 3.1 Εισαγωγή

Η οικογένεια του πρωτοκόλλου επικοινωνίας TCP/IP επιτρέπει σε ηλεκτρονικούς υπολογιστές, διαφορετικών μεγεθών και κατασκευαστών, οι οποίοι διαθέτουν διαφορετικά λειτουργικά συστήματα, να επικοινωνούν μεταξύ τους. Είναι εντυπωσιακό το πώς η χρησιμότητα του ξεπέρασε τους αρχικούς στόχους. **[4]**

Στα τέλη της δεκαετίας του '60 ξεκίνησε ως κυβερνητική – οικονομική –ερευνητική μελέτη σε δίκτυα μεταγωγής πακέτων. Στα τέλη του 1990 έχει εξελιχθεί στη πιο διαδεδομένη οικογένεια πρωτοκόλλων και το χρησιμοποιεί η πλειοψηφία των υπολογιστικών συστημάτων.

Αποτελεί την βάση για ότι αποκαλούμε παγκόσμιο ιστό του διαδικτύου, ή Internet, ένα δίκτυο ευρείας περιοχής (WAN) αποτελούμενο από ογδόντα δυο εκατομμύρια υπολογιστές που στην κυριολεξία διασυνδέουν την υφήλιο. **[4]**

Η παγκόσμια ζήτηση για προϊόντα διαδικτύωσης έχει επηρεάσει τις περισσότερες εταιρείες που διαθέτουν στην αγορά τεχνολογίες δικτύωσης. Ο ανταγωνισμός έχει αυξηθεί επειδή έχουν δημιουργηθεί νέες εταιρείες που παρέχουν υλικό και λογισμικό που χρειάζονται για τη διαδικτύωση. Ακόμα, πολλές εταιρείες έχουν τροποποιήσει τη σχεδίαση των πρωτοκόλλων τους ώστε να ανταποκρίνονται στη διαδικτύωση. Συγκεκριμένα, τα περισσότερα πρωτόκολλα δικτύων αρχικά είχαν σχεδιαστεί για να δουλεύουν με μια μόνο τεχνολογία δικτύου και με ένα φυσικό δίκτυο κάθε φορά. Για να παρέχουν

δυνατότητες διαδικτύωσης, οι εταιρείες έχουν επεκτείνει αυτές τις σχεδιάσεις ως εξής: Τα πρωτόκολλα έχουν προσαρμοστεί ώστε να δουλεύουν με τις τεχνολογίες δικτύων, και έχουν προστεθεί νέες δυνατότητες που επιτρέπουν στα πρωτόκολλα να μεταφέρουν δεδομένα μέσω δια-δικτύων. [5]

### 3.2 ΔΙΑΣΤΡΩΜΑΤΩΣΗ

Τα πρωτόκολλα διαδικτύωσης είναι συνήθως δομημένα σε επίπεδα, layers, όπου κάθε επίπεδο είναι υπεύθυνο για διαφορετική πλευρά της επικοινωνίας. Η οικογένεια πρωτοκόλλων, όπως το TCP/IP, είναι ο συνδυασμός διαφορετικών πρωτοκόλλων σε ποικίλα επίπεδα. Το μοντέλο διαστρωμάτωσης TCP/IP (TCP/IP Layering Model), το οποίο λέγεται και μοντέλο διαστρωμάτωσης του Internet (Internet Layering Model) ή μοντέλο αναφοράς του Internet (Internet Reference Model) αποτελείται από 4 επίπεδα, όπως φαίνεται παρακάτω στο σχήμα 3.1.

Application	telnet, FTP, e-mail κλπ.
Transport	TCP, UDP
Network	IP, ICMP, IGMP
Link	Device driver και interface card

Σχήμα 3.1 Τα 4 επίπεδα του πρωτοκόλλου TCP/IP

Κάθε επίπεδο έχει διαφορετικές ευθύνες:

1. Το φυσικό επίπεδο (Link Layer), ή όπως μερικές φορές το αποκαλούν Data-link layer ή network interface layer, συνήθως περιλαμβάνει τους οδηγούς συσκευών για το λειτουργικό

σύστημα και την κάρτα δικτύου του υπολογιστή. Μαζί χειρίζονται όλες τις λεπτομέρειες του υλικού για την φυσική ζεύξη.

2. Το επίπεδο δικτύου ή διαδικτύου (network or internet layer) χειρίζεται την κίνηση των πακέτων μέσα στο δίκτυο. Η δρομολόγηση των αυτοδύναμων πακέτων (IP), για παράδειγμα, είναι μία από τις υπηρεσίες του επιπέδου αυτού.
3. Το επίπεδο μεταφοράς (Transport layer) παρέχει, στο ανώτερο επίπεδο εφαρμογών, την ροή δεδομένων μεταξύ δύο κόμβων. Στην οικογένεια πρωτοκόλλων TCP/IP δύο είναι τα μεγαλύτερα διαφορετικά πρωτόκολλα μεταφοράς: το πρωτόκολλο ελέγχου μετάδοσης (TCP, Transmission Control Protocol) και το UDP (User Data-gram Protocol).
  - Το TCP παρέχει αξιόπιστη μεταφορά δεδομένων μεταξύ δυο υπολογιστών μέσω ενός μηχανισμού θετικών επιβεβαιώσεων (Acknowledgments). Ασχολείται με την υποδιαίρεση των δεδομένων, που παραλαμβάνει από το ανώτερο επίπεδο εφαρμογών, σε κατάλληλο μέγεθος πακέτα για το κατώτερο επίπεδο διαδικτύου, λαμβάνει τα πακέτα επιβεβαίωσης (acknowledge packet) και θέτει χρονικούς περιορισμούς στα πακέτα επιβεβαίωσης που λαμβάνονται από το άλλο άκρο.
  - Το UDP από την άλλη, παρέχει μια πολύ πιο απλή υπηρεσία στο επίπεδο εφαρμογών. Απλά στέλνει τα πακέτα δεδομένων, τα λεγόμενα data-grams, από τον έναν υπολογιστή στον άλλον, αλλά δεν υπάρχει καμία εγγύηση ότι τα data-grams θα φτάσουν στον προορισμό

τους. Κάθε επιθυμητή αξιοπιστία πρέπει να προστεθεί από το επίπεδο εφαρμογών.

4. Το επίπεδο εφαρμογών (Application Layer) χειρίζεται τις λεπτομέρειες από συγκεκριμένη εφαρμογή. Υπάρχουν πολλές κοινές εφαρμογές του TCP/IP αλλά σχεδόν όλες οι παραλλαγές παρέχουν:

- Telnet για απομακρυσμένες συνδέσεις
- FTP (File Transfer Protocol), το πρωτόκολλο μεταφοράς αρχείων
- SMTP (Simple Mail Transfer Protocol), το πρωτόκολλο μεταφοράς ηλεκτρονικής αλληλογραφίας
- SNMP (Simple Network Management Protocol), το πρωτόκολλο διαχείρισης δικτύου.

### **3.3 Υπηρεσίες TCP**

Το πρωτόκολλο TCP παρέχει μια εντελώς αξιόπιστη, χωρίς απώλειες ή επαναλήψεις δεδομένων, με βάση την σύνδεση, πλήρως αμφίδρομη υπηρεσία μεταφοράς ροής, η οποία επιτρέπει σε δύο εφαρμογές να δημιουργούν μια σύνδεση, να στέλνουν δεδομένα και προς τις δύο κατευθύνσεις και μετά να τερματίζουν την σύνδεση. Κάθε σύνδεση TCP ξεκινά με αξιοπιστία και τερματίζει ομαλά, και όλα τα δεδομένα παραδίδονται πριν από τον τερματισμό της σύνδεσης.

### 3.3.1 Η υπηρεσία του TCP στις εφαρμογές

Η υπηρεσία που παρέχεται από το πρωτόκολλο TCP στις εφαρμογές έχει τα εξής χαρακτηριστικά:

- **Βασισμένη στην σύνδεση** (Connection Orientation): Το TCP παρέχει υπηρεσία με βάση την σύνδεση, όπου μία εφαρμογή πρέπει πρώτα να αιτηθεί την νοητή αυτή σύνδεση με έναν προορισμό, και έπειτα χρησιμοποιεί την σύνδεση για να μεταφέρει δεδομένα. Η ροή των δεδομένων από την στιγμή της σύνδεσης είναι αμφίδρομη. Επιπλέον έχει την δυνατότητα να αποθηκεύει προσωρινά τα εισερχόμενα και τα εξερχόμενα δεδομένα, επιτρέποντας σε μία εφαρμογή να στέλνει τα δεδομένα και κατά την διάρκεια της μεταφοράς τους να απασχολείται με άλλες διεργασίες.
- **Επικοινωνία σημείο προς σημείο** (Point – to -Point) : παρέχει απευθείας σύνδεση από μία εφαρμογή σ' έναν υπολογιστή προς μία εφαρμογή σε έναν απομακρυσμένο υπολογιστή ανταλλάσσοντας μηνύματα.
- **Πλήρης αξιοπιστία** : Πρόκειται για το βασικό και χαρακτηριστικό γνώρισμα του TCP. Το πρωτόκολλο φροντίζει να εξασφαλιστεί όχι μόνο η άφιξη των μηνυμάτων, που στέλνονται, στον προορισμό τους, αλλά και ότι αυτά φτάνουν εκεί με την σειρά με την οποία στάλθηκαν. Αν κάποιο μήνυμα λείπει το πρωτόκολλο θα ειδοποιήσει τον υπολογιστή-αποστολέα προκειμένου αυτό να σταλεί ξανά.

### 3.3.2 Υπηρεσία με βάση την σύνδεση

Το TCP είναι υπεύθυνο για τη σύνδεση δυο σημείων, καθώς και για τη μεταξύ τους επικοινωνία, εξασφαλίζοντας την αποστολή και τη λήψη μηνυμάτων. Οι συνδέσεις που παρέχονται λέγονται εικονικές συνδέσεις (virtual connections), επειδή επιτυγχάνονται με λογισμικό.

Χρησιμοποιείται ένας μηχανισμός *παραθύρου (window)* για να ελέγχει τη ροή των δεδομένων. Όταν επιτυγχάνεται μια σύνδεση, το κάθε άκρο της κατανέμει ένα χώρο προσωρινής αποθήκευσης (buffer) για να αποθηκεύονται τα εισερχόμενα δεδομένα, και ενημερώνει το άλλο άκρο για το μέγεθος του χώρου προσωρινής αποθήκευσης. Ο διαθέσιμος χώρος προσωρινής αποθήκευσης ονομάζεται *διαφημιζόμενο παράθυρο (advertised window)*.

Αν η εφαρμογή-παραλήπτης μπορεί να διαβάσει δεδομένα με την ταχύτητα που έρχονται, ο παραλήπτης θα στέλνει μια αναγγελία θετικού παραθύρου μαζί με μια επιβεβαίωση. Αν όμως η πλευρά του αποστολέα λειτουργεί γρηγορότερα από του παραλήπτη, τα εισερχόμενα δεδομένα θα γεμίσουν τελικά το χώρο προσωρινής αποθήκευσης του παραλήπτη, με αποτέλεσμα αυτός να αναγγείλει ένα *μηδενικό παράθυρο (zero window)*. Ένας αποστολέας που λαμβάνει μια αναγγελία μηδενικού παραθύρου πρέπει να σταματήσει την μετάδοση, μέχρι να αναγγείλει ξανά ο παραλήπτης ένα θετικό παράθυρο.

### **3.3.3 Υπηρεσία Επικοινωνία σημείο προς σημείο (Point – to –Point)**

Κάθε μήνυμα του TCP έχει ένα συγκεκριμένο τελικό άκρο (σημείο προορισμού), το οποίο προσδιορίζεται μέσα από το μήνυμα. Το λογισμικό του TCP χρειάζεται σε κάθε άκρο μιας εικονικής σύνδεσης, αλλά όχι στους ενδιάμεσους δρομολογητές. Από την άποψη του TCP, ολόκληρο το διαδίκτυο είναι ένα σύστημα επικοινωνίας που μπορεί να δέχεται και να παραδίδει μηνύματα χωρίς να αλλάζει ή να ερμηνεύει το περιεχόμενό τους.

### **3.3.4 Υπηρεσία Πλήρης Αξιοπιστία**

Μια από τις πιο σημαντικές τεχνικές με την οποία επιτυγχάνεται η αξιοπιστία στο TCP είναι η *επαναμετάδοση* (retransmission). Όταν το TCP στέλνει δεδομένα, ο αποστολέας αντισταθμίζει την απώλεια πακέτων χρησιμοποιώντας έναν μηχανισμό επαναμετάδοσης. Υπάρχουν δυο τρόποι επαναμετάδοσης πακέτων: Retransmission Timeout (RTO) Λήξη χρονομετρητή και Επαναμετάδοση και Duplicate Acknowledgements (DupAcks) Διπλό Αντίγραφο Επιβεβαίωσης Λήψης.

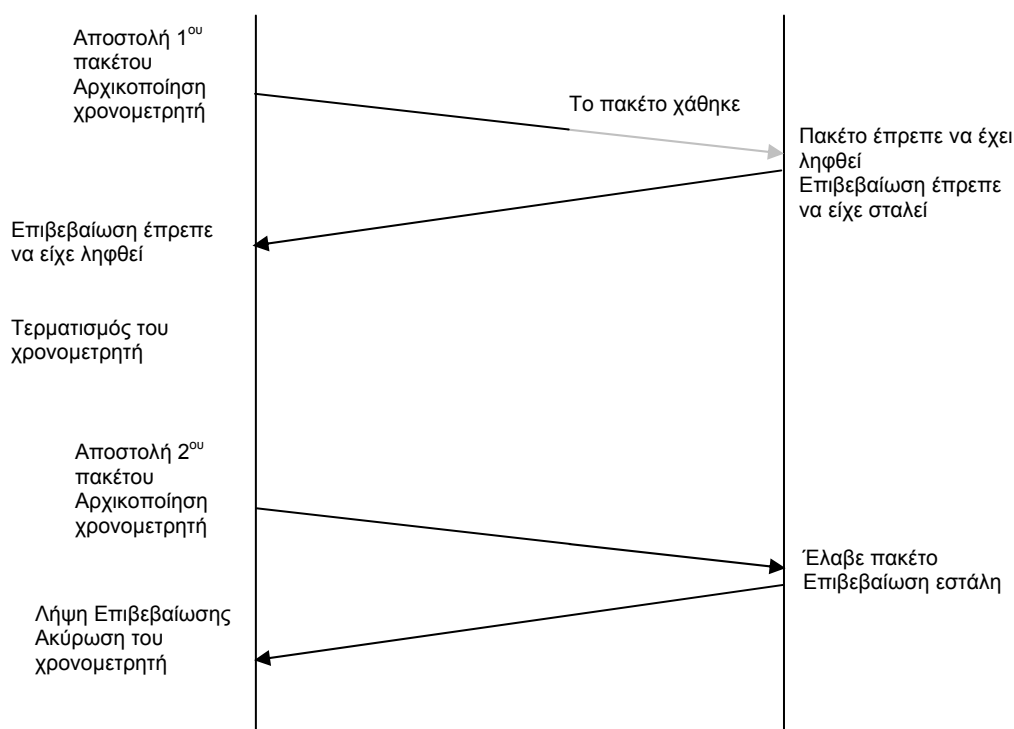
#### **3.3.4.1 Λήξη χρονομετρητή και Επαναμετάδοση - Retransmission TimeOut (RTO)**

Το TCP για κάθε πακέτο που στέλνει, ξεκινά ένα χρονόμετρο : Αν ο χρόνος λήξει πριν έρθει επιβεβαίωση του πακέτου, ο αποστολέας επαναμεταδίδει το πακέτο, όπως βλέπουμε στο σχήμα 3.2. Η καθυστέρηση, για να φτάσουν τα δεδομένα σε έναν προορισμό και

να επιστραφεί μία επιβεβαίωση, εξαρτάται από την κυκλοφορία στο διαδίκτυο καθώς και από την απόσταση μέχρι τον προορισμό. Επειδή το TCP επιτρέπει να επικοινωνούν ταυτόχρονα πολλές εφαρμογές με πολλούς προορισμούς, και επειδή οι συνθήκες της κυκλοφορίας επηρεάζουν την καθυστέρηση, το TCP πρέπει να χειρίζεται μία ποικιλία καθυστερήσεων, οι οποίες μπορεί να μεταβάλλονται γρήγορα.

Το TCP δεν μπορεί να γνωρίζει τις ακριβείς καθυστερήσεις σε όλα τα μέρη ενός διαδικτύου κάθε στιγμή. Εκτιμά την *καθυστερήση μετάβασης και επιστροφής (round – trip delay)* για την κάθε ενεργή σύνδεση, μετρώντας το χρόνο που απαιτείται για να λάβει μια απόκριση. Κάθε φορά που στέλνει ένα μήνυμα για το οποίο περιμένει απόκριση, το TCP καταγράφει την ώρα αποστολής του μηνύματος. Όταν έρθει η απόκριση, το TCP αφαιρεί την ώρα αποστολής του μηνύματος από την τρέχουσα ώρα για να αποκτήσει μια νέα εκτίμηση της καθυστέρησης μετάβασης και επιστροφής για αυτήν την σύνδεση. Καθώς στέλνει πακέτα δεδομένων και λαμβάνει επιβεβαιώσεις, το TCP δημιουργεί μια ακολουθία εκτιμήσεων χρόνου μετάβασης και επιστροφής, και χρησιμοποιεί μια στατιστική συνάρτηση για να υπολογίσει ένα σταθμισμένο μέσο όρο. Η χρήση του σταθμισμένου μέσου όρου βοηθά το TCP να επαναφέρει το χρονόμετρο επαναμετάδοσης αν η καθυστέρηση επανέλθει σε μια χαμηλότερη τιμή. Όταν η καθυστέρηση παραμένει σταθερή, το TCP προσαρμόζει το χρόνο λήξης για την επαναμετάδοση σε μια τιμή ελάχιστα μεγαλύτερη από την μέση καθυστέρηση μετάβασης και επιστροφής.



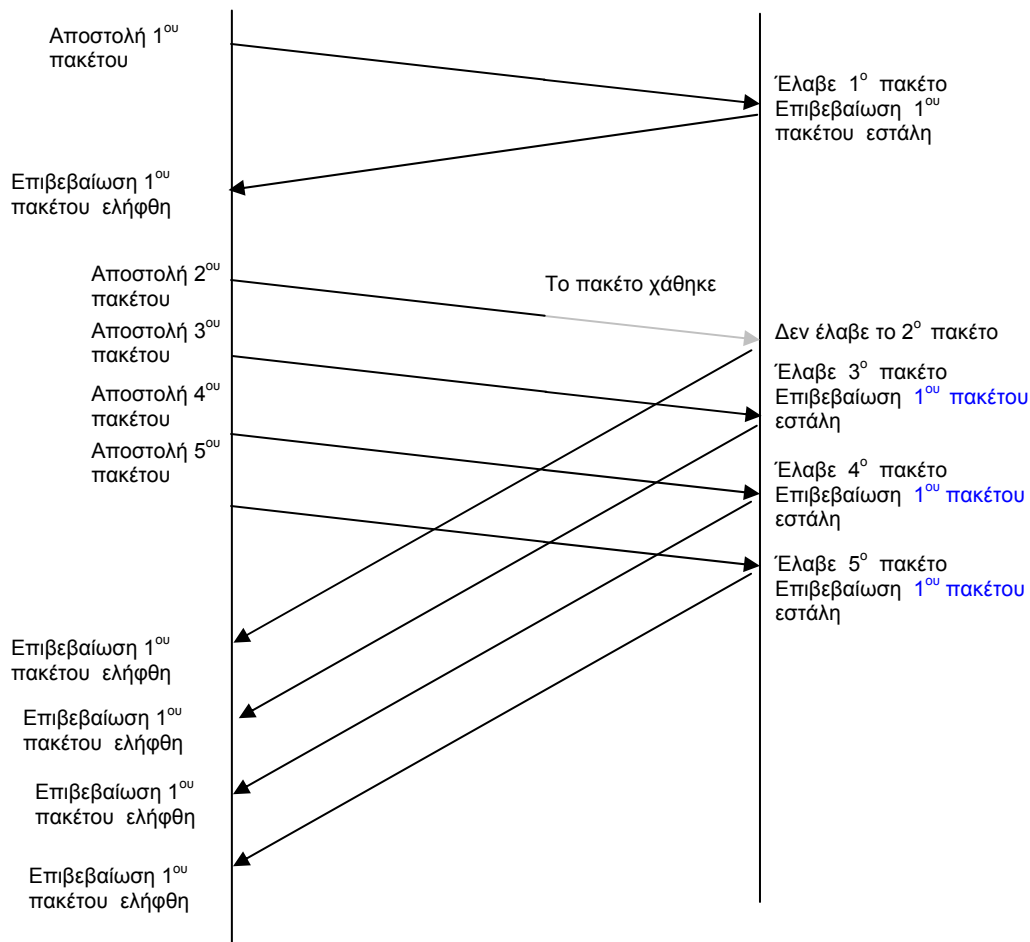


Σχήμα 3.2 Παράδειγμα χρήσης της μεθόδου της «Λήξης Χρονομετρητή και Επαναμετάδοσης

### 3.3.4.2 Διπλότυπη Επιβεβαίωσης Λήψης – Duplicate Acknowledgements

Πολλά πρωτόκολλα, όπως το TCP, διαθέτουν μηχανισμό θετικών επιβεβαιώσεων (acknowledgement) που παράγονται και στέλνονται για κάθε σωστά λαμβανόμενο πακέτο της ακολουθίας, που προσδιορίζεται από έναν σειριακό αριθμό (sequence number). Οι Διπλότυπες Επιβεβαιώσεις, δηλαδή τα **DupAcks**, παράγονται λόγω του ότι χάθηκε κάποιο πακέτο ή κάποιο πακέτο έχει ληφθεί σε λάθος σειρά. Το TCP αντιλαμβάνεται ότι κάποιο πακέτο έχει χαθεί όταν λάβει τρία συνεχόμενα DupAcks με το ίδιο αριθμό επιβεβαίωσης. Για παράδειγμα στο σχήμα 3.3 φαίνεται ότι στέλνεται το πρώτο πακέτο και ο παραλήπτης, όταν το λάβει, στέλνει μια επιβεβαίωση. Έπειτα

στέλνονται με τη σειρά από το δεύτερο πακέτο έως το πέμπτο πακέτο. Το δεύτερο όμως πακέτο χάνεται στη διαδρομή. Όταν θα ληφθούν τα πακέτα 3, 4, και 5 τότε ο παραλήπτης θα στείλει τρεις συνεχόμενες επιβεβαιώσεις, που είναι τα λεγόμενα duracks, με τον αριθμό επιβεβαίωσης όμως του πρώτου πακέτου. Έτσι δηλώνει ότι περιμένει να λάβει το δεύτερο πακέτο, το οποίο δεν έλαβε.



Σχήμα 3.3 Παράδειγμα χρήσης της μεθόδου Επιβεβαίωση Λήψης

### 3.4 Έλεγχος Συμφόρησης

Στα περισσότερα διαδίκτυα, η απώλεια πακέτων είναι πιο πιθανό να προκληθεί από συμφόρηση παρά από μια βλάβη υλικού. Τα

πρωτόκολλα μεταφοράς που χρησιμοποιούν επαναμετάδοση μπορεί να επιτείνουν το πρόβλημα της συμφόρησης, εισάγοντας πρόσθετα αντίγραφα ενός μηνύματος. Αν η συμφόρηση προκαλέσει υπερβολικές επαναμεταδόσεις, ολόκληρο το σύστημα μπορεί να φτάσει σε μια κατάσταση κατάρρευσης λόγω συμφόρησης. Για να αποφευχθεί αυτό το πρόβλημα, το TCP χρησιμοποιεί πάντα την απώλεια πακέτων ως μέτρο της συμφόρησης, και αποκρίνεται μειώνοντας το ρυθμό με το οποίο επαναμεταδίδει δεδομένα.

### **3.4.1 Αργή εκκίνηση - Slow Start**

Όποτε χάνεται ένα μήνυμα, το TCP αρχίζει τον έλεγχο της συμφόρησης. Μια μέθοδος για τον έλεγχο της συμφόρησης που χρησιμοποιεί το TCP είναι η Αργή εκκίνηση. Αυτήν η μέθοδος προσθέτει ένα επιπλέον *παράθυρο συμφόρησης* (congestion window, cwnd), στο πρώτο πακέτο που στέλνεται στον παραλήπτη. Όταν επιτευχθεί μια καινούργια σύνδεση, αρχικοποιείται το cwnd σε ένα πακέτο. Το μέγεθος του πακέτου το καθορίζει ο παραλήπτης. Ο αποστολέας στέλνει το πρώτο πακέτο και περιμένει την επιβεβαίωση. Όταν λάβει την επιβεβαίωση τότε αυξάνει κατά ένα την τιμή του cwnd οπότε θα στείλει δύο πακέτα. Όταν λάβει την επιβεβαίωση και για τα δύο πακέτα τότε θα διπλασιάσει την τιμή του cwnd και θα στείλει 4 τμήματα. Η εκθετική αύξηση συνεχίζεται μέχρι το TCP να στείλει τα μισά από όσα χωρούν στο αναγγελλλόμενο *παράθυρο* του παραλήπτη, οπότε επιβραδύνει το ρυθμό της αύξησης.

### 3.4.2 Γρήγορη Επαναμετάδοση και Γρήγορη Ανάκτηση – Fast Retransmit and Fast Recovery

Από το TCP απαιτείται να δημιουργήσει μια διπλή επιβεβαίωση (dupack) όταν ληφθεί ένα πακέτο εκτός σειράς. Σκοπός του dupack είναι να ενημερώσει τον αποστολέα ότι κάποιο πακέτο έχει ληφθεί σε λάθος σειρά και να του υποδείξει τον σειριακό αριθμό του πακέτου που περιμένει. Μετά από τη λήψη τριών συνεχόμενων dupacks θεωρούμε πιο πιθανό το πακέτο να έχει χαθεί. Τότε στέλνουμε μόνο το πακέτο, που πιθανολογείται να έχει χαθεί, πριν την λήξη του χρονομετρητή. Αυτός ο αλγόριθμος είναι η Γρήγορη Επαναμετάδοση και χρησιμοποιείται σε συνδυασμό με τον αλγόριθμο της Γρήγορης Ανάκτησης ως εξής:

1. Όταν ληφθεί το τρίτο dupack, θέτουμε το κατώτατο κατώφλι *ssthresh*, στο μισό της ελάχιστης τιμής του τρέχοντος παραθύρου *συμφόρησης (cwnd)* και του διαφημιζόμενου παραθύρου (*advertised window*).  
Επαναμεταδίδει το χαμένο πακέτο.  
Προσθέτει στο *cwnd* το *ssthresh* και τρεις φορές το μέγεθος του πακέτου.
2. Κάθε φορά που λαμβάνει ένα dupack, αυξάνει το *cwnd* με βάση το μέγεθος του πακέτου και μεταδίδει το πακέτο.
3. Όταν παραλάβει την επόμενη επιβεβαίωση λήψης, που επιβεβαιώνει αποστολή νέων δεδομένων, θέτει την τιμή του *ssthresh* ίση με την τιμή του *cwnd*.

### 3.4.3 Αλγόριθμος Αποφυγής Συμφόρησης - Congestion Avoidance Algorithm

Ο αλγόριθμος Congestion avoidance είναι ένας τρόπος να εξεταστούν τα χαμένα πακέτα. Υποθέτει ότι εκείνη η απώλεια πακέτων που προκαλείται από τη βλάβη είναι πολύ μικρή (πολύ λιγότερο από 1 %), επομένως η απώλεια ενός πακέτου γίνεται από συμφόρηση κάπου στο δίκτυο μεταξύ της πηγής και του προορισμού. Υπάρχουν δύο ενδεικτικές απώλειες πακέτων: η λήξη χρόνου και η λήψη των διπλότυπων αναγνωρίσεων, ACKs. Οι Congestion avoidance και slow start είναι ανεξάρτητοι αλγόριθμοι με διαφορετικά αντικείμενα. Αλλά όταν έχουμε συμφόρηση θέλουμε να επιβραδύνουμε το ποσοστό μετάδοσης πακέτων στο δίκτυο, και έπειτα να βάλουμε σε λειτουργία τον slow start για να πάρουν πάλι τα πράγματα την σειρά τους. Στην πράξη εφαρμόζονται από κοινού.

Οι αλγόριθμοι Congestion avoidance και slow start απαιτούν ότι αυτές οι δύο μεταβλητές πρέπει να διατηρούνται για κάθε σύνδεση: ένα παράθυρο συμφόρησης, *cwnd*, και ένα slow start κατώτατο κατώφλι, *ssthresh*. Συνδυασμένοι οι αλγόριθμοι λειτουργούν ως εξής:

1. Αρχικοποίηση τιμών για μία υπάρχουσα σύνδεση, του παραθύρου συμφόρησης, *cwnd*, σε ένα τμήμα και του κατώτατο κατώφλι, *ssthresh* στα 65535 bytes.
2. Η ρουτίνα της εξόδου του TCP ποτέ δεν στέλνει περισσότερα από το ελάχιστο του παραθύρου συμφόρησης, *cwnd*, και του λαμβανόμενου διαφημιζόμενου παραθύρου, receiver's advertised window ( $window = \min \{cwnd, awnd\}$ ). Congestion avoidance είναι έλεγχος ροής που επιβάλλεται από τον αποστολέα, ενώ το διαφημισμένο παράθυρο είναι έλεγχος ροής που επιβάλλεται από

τον παραλήπτη. Το πρώτο βασίζεται στην αξιολόγηση του αποστολέα της αντιλαμβανόμενης συμφόρησης του δικτύου, το τελευταίο συσχετίζεται με το ποσό του ελευθέρου χώρου στην προσωρινή μνήμη του παραλήπτη για αυτήν την σύνδεση.

3. Όταν η συμφόρηση εμφανίζεται, το μισό του τρέχοντος μεγέθους του παραθύρου (window) αποθηκεύεται στο κατώτατο κατώφλι, *ssthresh*. Επιπλέον, εάν η συμφόρηση υποδεικνύεται από λήξη χρόνου, το *cwnd* τίθεται σε ένα πακέτο.

### 3.5 Εκδόσεις του TCP

- Η έκδοση **TCP-Tahoe** είναι η πρώτη έκδοση που εισήχθη το 1990 για να παρέχει τους μηχανισμούς, όπως την Αργή Εκκίνηση (Slow Start), Αποφυγή Συμφόρησης (Congestion Avoidance) και τη Γρήγορη Επαναμετάδοση (Fast Retransmit), οι οποίοι βελτιώνουν την απόδοση του TCP και την σταθερότητα του δικτύου. Είναι προορισμένοι στο να εξετάζουν αργά το δίκτυο και να μειώνουν δραστικά τον ρυθμό αποστολής για την αποσυμφόρηση. Ο μηχανισμός της Γρήγορης Επαναμετάδοσης (Fast Retransmit) επιτρέπει την γρήγορη επαναμετάδοση του χαμένου πακέτου αφού ο αποστολέας λάβει τρεις συνεχόμενες επιβεβαιώσεις για το ίδιο TCP πακέτο.
- Η έκδοση **TCP-Reno** περιλαμβάνει τον μηχανισμό της Γρήγορης Ανάκαμψης (Fast Recovery), ο οποίος ενεργοποιείται μετά την Γρήγορη Επαναμετάδοση (Fast Retransmit) και δεν μειώνει την τιμή του παράθυρου συμφόρησης *cwnd* (congestion window) σε ένα, αλλά στο μισό της τρέχουσας τιμής του. Η λήψη τριών διπλών επιβεβαιώσεων (DupAcks)

είναι μια ένδειξη ότι το δίκτυο δεν είναι βαριά κορεσμένο, επομένως η ρυθμοροή του TCP μπορεί να βελτιωθεί χωρίς να μειώσουμε δραστικά το παράθυρο συμφόρησης cwnd..

- Η έκδοση **TCP-New Reno** εισήχθη για να αποτρέψει την Λήξη Χρόνου, στην περίπτωση που έχουν χαθεί πολλαπλά πακέτα του παράθυρου συμφόρησης cwnd, παραμένοντας στον μηχανισμό Γρήγορης Επαναμετάδοσης (Fast Retransmit) / Γρήγορης Ανάκαμψης (Fast Recovery) μέχρι να ληφθούν σωστά όλα τα πακέτα.
- Η έκδοση **TCP-Vegas** χρησιμοποιεί τους μηχανισμούς fast retransmit και fast recovery, ενώ διαφέρει από την έκδοση TCP-Reno στον τρόπο με τον οποίο ανιχνεύει τις απώλειες, το διαθέσιμο εύρος ζώνης και επιπλέον έχει διαφορετική slow start συμπεριφορά. Η TCP-Vegas ελέγχει το μέγεθος του παραθύρου συμφόρησης cwnd, με τη *καθυστέρηση μετάβασης και επιστροφής (round – trip delay)* των πακέτων, που έχουν σταλεί. Εάν η καθυστέρηση μετάβασης και επιστροφής (round – trip delay) γίνει μεγάλη, αντιλαμβάνεται ότι το δίκτυο αρχίζει να υπερφορτώνεται και μειώνει το μέγεθος του παραθύρου συμφόρησης cwnd. Εάν από την άλλη η καθυστέρηση μετάβασης και επιστροφής (round – trip delay) γίνει μικρή, η TCP-Vegas καταλαβαίνει ότι δεν υπάρχει πια συμφόρηση στο δίκτυο και αυξάνει πάλι το μέγεθος του παραθύρου συμφόρησης cwnd. Επομένως, το μέγεθος του παραθύρου συμφόρησης cwnd σε μια ιδανική κατάσταση περιμένει να τεθεί σε μια κατάλληλη τιμή.
- Η έκδοση **TCP-Westwood (TCPW)** βασίζεται στην αποσταθεροποίηση της ροής των ACK, για πληροφόρηση έτσι ώστε να βοηθάει στην καλύτερη ενημέρωση των παραμέτρων

συμφόρησης ελέγχου: Του κατώτατο κατωφλίου του Slow Start (ssthresh) και του Παραθύρου Συμφόρησης (cwnd). Στο TCPW, εκτιμάται ένας «Επιθυμητός Ρυθμός» που χρησιμοποιείται από τον αποστολέα για να ενημερώσει το ssthresh και το cwnd σε ενδεικτική απώλεια ή κατά τη διάρκεια της φάσης «Ευκίνητης εντιμότητας» (Agile Probing), που είναι μια προτεινόμενη τροποποίηση της γνωστής φάσης Slow Start. Σε αντίθεση με την Slow Start, σχεδιάζεται ένα πλάνο ενεργειών που ονομάζεται «Συνεχή Ανίχνευση Μη Συμφόρησης» (Persistent Non Congestion Detection, PNCD) για να ανιχνεύει συνεχή έλλειψη συμφόρησης και να επιφέρει την φάση «Ευκίνητης εντιμότητας» για γρήγορη δυναμική αξιοποίηση μεγάλου εύρους ζώνης. [6]

### 3.6 Μορφή του τμήματος TCP

Το πρωτόκολλο TCP μια μόνο μορφή για όλα τα μηνύματα, συμπεριλαμβανομένου των μηνυμάτων που μεταφέρουν δεδομένα, εκείνων που μεταφέρουν επιβεβαιώσεις, και των μηνυμάτων που χρησιμοποιούνται για να δημιουργηθεί ή να τερματιστεί μια σύνδεση. Το TCP χρησιμοποιεί τον όρο *τμήμα (segment)* όταν αναφέρετε σε ένα μήνυμα. Η μορφή ενός τμήματος TCP παρουσιάζεται στο σχήμα 3.4.

ΘΥΡΑ ΑΦΕΤΗΡΙΑΣ				ΘΥΡΑ ΠΡΟΟΡΙΣΜΟΥ				
ΣΕΙΡΙΑΚΟΣ ΑΡΙΘΜΟΣ								
ΑΡΙΘΜΟΣ ΕΠΙΒΕΒΑΙΩΣΗΣ								
ΜΗΚΟΣ ΕΠΙΚΕΦΑΛΙ- ΔΑΣ	ΔΕΣΜΕΥ- ΜΕΝΟ	U	A	P	R	S	F	ΠΑΡΑΘΥΡΟ
ΑΘΡΟΙΣΜΑ ΕΛΕΓΧΟΥ				ΕΠΕΙΓΟΝ ΔΕΙΚΤΗΣ				



ΕΠΙΛΟΓΕΣ (αν υπάρχουν)
ΑΡΧΗ ΤΩΝ ΔΕΔΟΜΕΝΩΝ

Σχήμα 3.4 Μορφή τμήματος TCP

Τα πεδία που αποτελούν το τμήμα του TCP είναι τα εξής:

- ✓ Τα πεδία **Θύρα Αφιετηρίας (source port)** και **Θύρα Προορισμού (destination port)** αναφέρονται στον αριθμό της θύρας που αντιστοιχεί στην τρέχουσα εφαρμογή.
- ✓ Το πεδίο **Σειριακός Αριθμός (sequence number)** είναι 32bit και προσδιορίζει τον σειριακό αριθμό του πρώτου byte των δεδομένων που μεταδίδονται. Ο παραλήπτης χρησιμοποιεί τον σειριακό αριθμό για να βάλει στην σωστή σειρά τα πακέτα που έρχονται εκτός σειράς και να υπολογίσει τον αριθμό επιβεβαίωσης.
- ✓ Το πεδίο **Αριθμός Επιβεβαίωσης (acknowledgment number)** που είναι 32bit ,επιβεβαιώνει την λήψη των δεδομένων και περιέχει τον σειριακό αριθμό του επόμενου byte που το TCP περιμένει να λάβει.
- ✓ Το πεδίο **Μήκος Επικεφαλίδας (header length)** είναι 4bit και αναφέρει πόσες λέξεις των 32bit βρίσκονται στην επικεφαλίδα TCP.
- ✓ Το πεδίο **Δεσμευμένο** είναι 6bit και πρόκειται να χρησιμοποιηθεί μελλοντικά. Η τιμή του σήμερα είναι μηδέν.
- ✓ Ακολουθούν 6 bit τα οποία είναι:
  - **U (Urgent): Επείγον** το οποίο ενημερώνει τον παραλήπτη για την ύπαρξη επειγόντων δεδομένων ώστε να ενεργήσει κατάλληλα.
  - **A (Acknowledge): Επιβεβαίωση** η οποία χρησιμοποιείται για να δηλώσει ότι η σύνδεση αποκαταστάθηκε.

- **P (Push): Τέλος Μηνύματος** ενεργοποιείται όταν λάβει όλα τα δεδομένα ώστε ο παραλήπτης να τα προωθήσει στην εφαρμογή αμέσως.
- **R (Reset): Επανεκκίνηση** χρησιμοποιείται για να αρχικοποιήσει μια σύνδεση που έχει αποτύχει, να απορρίψει ένα άκυρο πακέτο ή για να αρνηθεί το άνοιγμα μιας σύνδεσης.
- **S (Syn): Συγχρονισμός** όταν εγκαθιδρύσει την σύνδεση επιλέγει έναν μοναδικό σειριακό αριθμό για αυτήν την σύνδεση.
- **F (Fin): Τέλος** το οποίο χρησιμοποιείται για να δηλώσει το τέλος της αποστολής των δεδομένων.

✓ Το πεδίο **Παράθυρο (Window)** είναι 16bit και καθορίζει πόσος ακόμη χώρος προσωρινής αποθήκευσης είναι διαθέσιμος για δεδομένα.

✓ Το πεδίο **Άθροισμα Ελέγχου (Checksum)** είναι 16bit και χρησιμοποιείται για τον έλεγχο της ορθότητας των δεδομένων.

✓ Το πεδίο **Επείγον Δείκτης (Urgent Pointer)** είναι 16bit και χρησιμοποιείται για να εντοπιστεί ο πρώτος χαρακτήρας δεδομένων μετά από αυτά που έχουν χαρακτηριστεί ως επείγοντα.

✓ Το πεδίο **Επιλογές (Options)** είναι μεταβλητού μήκους και χρησιμοποιείται για να προσδιορίσει διάφορες λειτουργίες που απαιτούνται από το TCP.

✓ Το πεδίο **Δεδομένα (Data)** είναι μέρος των πραγματικών δεδομένων που στέλνει η μια εφαρμογή στην άλλη.

## **ΚΕΦΑΛΑΙΟ 4**

### **NETWORK SIMULATOR ( NS -2 )**

#### **4.1 Εισαγωγή**

Ο Network Simulator (NS-2) είναι ένας προσομοιωτής δικτύου οδηγούμενων γεγονότων, event-driven , ο οποίος προσομοιώνει διαφόρων ειδών IP δίκτυα. Το εργαλείο αυτό ξεκίνησε αρχικά σαν παραλλαγή του "REAL network simulator" το 1989 και αναπτύχθηκε σταδιακά τα τελευταία χρόνια. Από το 1995 και μετά, ο NS αποτελεί πλέον μέρος του VINT project που αναπτύσσει εργαλεία για τη μελέτη των δικτύων. Ο κώδικας του αποτελεί συνεισφορά από διαφορετικές ερευνητικές ομάδες. Το πλεονέκτημα του είναι ότι ο κώδικας του περιέχει μεγάλη συλλογή από πακέτα προσομοίωσης όμως πολλά από αυτά δεν είναι αρκετά ξεκάθαρα.

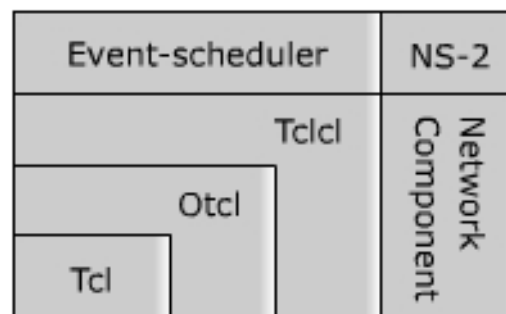
#### **4.2 Λεπτομερή περιγραφή**

Ο NS αρχικά αναπτύχθηκε για να προσομοιώνει τοπικά και ευρείας κάλυψης δίκτυα (LAN/WAN), ενώ σήμερα μπορεί και προσομοιώνει ακόμα ασύρματα και δορυφορικά δίκτυα. Μπορεί να υλοποιήσει διάφορα πρωτοκόλλα δικτύων όπως είναι το TCP και το UDP, διάφορα είδη κίνησης όπως είναι το CBR, FTP, Telnet, Web και VBR, διάφορους μηχανισμούς διαχείρισης της ουράς στους κόμβους, όπως είναι η Drop Tail, η RED και η CBQ, διάφορους αλγόριθμους δρομολόγησης.

Επί του παρόντος,ο NS (έκδοση 2) είναι γραμμένο σε C++ και σε OTcl - πρόκειται για μια Tcl scripting γλώσσα προγραμματισμού με αντικειμενοστραφής προεκτάσεις.

Χρησιμοποιούμε την γλώσσα προγραμματισμού C++ για προσομοίωση πολύπλοκων /σύνθετων πρωτοκόλλων και επίσης για την εισαγωγή, τροποποίηση, δημιουργία νέου μοντέλου και για οτιδήποτε απαιτεί επεξεργασία ανά πακέτο.

Την Tcl την χρησιμοποιούμε για την δημιουργία και αρχικοποίηση του σεναρίου της προσομοίωσης και για να δώσουμε τιμές σε παραμέτρους προσομοίωσης. Ενδείκνυται για να δημιουργήσουμε απλές τοπολογίες. Για συγγραφή του κώδικα σε γλώσσα Tcl χρησιμοποιήστε το notepad ή κάποιον text editor για μεγαλύτερο αριθμό γραμμών. Το σχήμα 4.1 απεικονίζει την γενική αρχιτεκτονική του NS. Σε αυτό το σχήμα, ο απλός χρήστης μπορεί να θεωρηθεί ότι βρίσκεται στην κάτω αριστερή γωνία, ο οποίος σχεδιάζει και τρέχει προσομοιώσεις μέσω των Tcl προγραμμάτων, χρησιμοποιώντας τα αντικείμενα της OTcl.



Σχήμα 4.1 Αρχιτεκτονική NS

Ο NS παράγει ένα ή περισσότερα αρχεία κειμένου, που περιέχουν λεπτομερή δεδομένα σχετικά με την διαδικασία της προσομοίωσης. Τα δεδομένα αυτά μπορούν να χρησιμοποιηθούν είτε για ανάλυση και διεξαγωγή κάποιων συμπερασμάτων στο awk, στο excel ή στο xgraph, είτε σαν είσοδο για το NAM (Network Animator), το οποίο αποτελεί μια γραφική διεπαφή με τον χρήστη και μπορεί να παρουσιάσει εικονικά

την τοπολογία που δημιουργήσαμε καθώς και διάφορες πληροφορίες, όπως τον αριθμό των πακέτων που χάνονται από μια γραμμή ή την μεταφορά των πακέτων από τον ένα κόμβο στον άλλο, με γραφικό τρόπο.

### 4.3 Τι είναι η TCL

Η Tcl ή « tool command language » είναι πολύ απλή γλώσσα προγραμματισμού ανοιχτής πηγής. Η Tcl παρέχει βασικά γλωσσικά χαρακτηριστικά όπως μεταβλητές, διαδικασίες ελέγχου και τρέχει σχεδόν σε όλα τα σύγχρονα λειτουργικά όπως στο Unix, Macintosh και στα Windows 95/98/NT. Αλλά το κλειδί της Tcl είναι η επεκτασιμότητα της.

Η Tcl αρχικά αναπτύχθηκε σαν επαναχρησιμοποιήσιμη γλώσσα διαχείρισης εντολών για πειραματικά βοηθητικά σχεδιαστικά εργαλεία Η/Υ. Ο διερμηνέας ήταν υλοποιήσιμος σαν βιβλιοθήκη της γλώσσας προγραμματισμού C και μπορούσε να συνδεθεί μέσα από κάθε εφαρμογή. Είναι πολύ εύκολο να προσθέσεις νέες λειτουργίες στον Tcl πυρήνα, έτσι ώστε να είναι η ιδανική επαναχρησιμοποιήσιμη γλώσσα που μπορεί να ενοποιηθεί σε οποιοδήποτε εφαρμογή.

Αλλά η Tcl είναι γλώσσα προγραμματισμού από μόνη της ορθή, που μπορείς να την περιγράψεις πρόχειρα σαν διασταύρωση μεταξύ LISP/Scheme (με λιγότερα παρεγχύματα) και κελιά (με πιο ισχυρή δομή)- βλέπε TCL heritage. Μπορείς να γράψεις όσα προγράμματα θέλεις σε Tcl, όπως και σε οποιαδήποτε άλλη γλώσσα.

Τα Tcl προγράμματα συνήθως αποκαλούνται σενάρια, scripts, γιατί το πρόγραμμα δεν χρειάζεται να ολοκληρωθεί σε γλώσσα μηχανής. Η σχετική σελίδα "Beginning Tcl", σε ξεκινά από πρακτικές

επιφάνειες εργασίας χρήστη. Μπορείς πολύ εύκολα να δημιουργήσεις μια επιφάνεια εργασίας για κάθε Tcl εφαρμογή ή ένα ολοκληρωμένο πρόγραμμα. Μπορείς να χρησιμοποιήσεις και άλλες επεκτάσεις όπως 'Expect' για να ελέγχεις τις γραμμές εντολών του προγράμματος για τις επιφάνειες εργασίας των χρηστών.

Η Tcl έχει πληθώρα επεκτάσεων, ώστε να μπορείς να γράψεις οποιοδήποτε είδος προγράμματος σαν Tcl script. Υπάρχουν προεκτάσεις για CGI και HTTP εφαρμογές, για γραφική καταγραφή σχεδίου και πρόσβαση σε βάσεις δεδομένων.

#### 4.4 Η TCL σε ενσύρματα δίκτυα

```
#Δημιουργεί ένα αντικείμενο προσομοίωσης  
set ns [new Simulator]
```

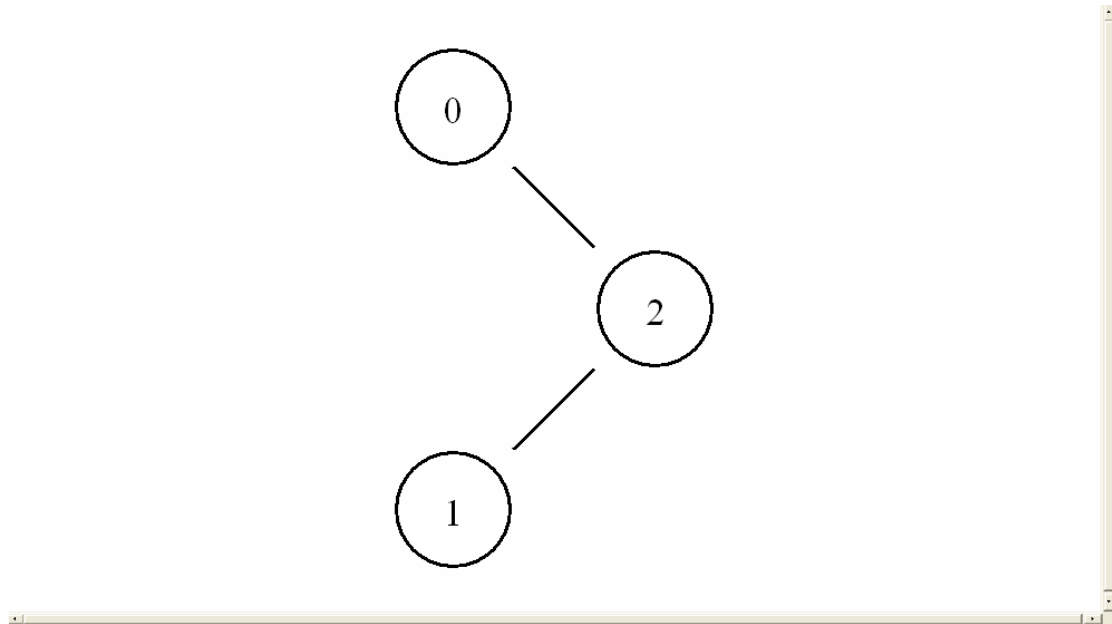
```
#Ανοίγει το αρχείο trace file  
set nf [open out.nam w]  
$ns namtrace-all $nf
```

```
#Καθορίζει την διαδικασία τέλους  
proc finish {} {  
    global ns nf  
    $ns flush-trace  
  
    #Κλείνει το αρχείο trace file  
    close $nf  
    #Εκτελεί το nam του αρχείου trace  
    exec nam out.nam &  
    exit 0  
}
```

```
#Δημιουργεί τρεις κόμβους  
set n0 [$ns node]  
set n1 [$ns node]  
set n2 [$ns node]
```

```
#Δημιουργεί συνδέσεις μεταξύ των κόμβων  
$ns duplex-link $n0 $n2 1Mb 10ms Drop Tail  
$ns duplex-link $n1 $n2 1Mb 10ms Drop Tail  
$ns duplex-link-op $n0 $n2 orient right-down  
$ns duplex-link-op $n1 $n2 orient right-u
```

Το παρακάτω σχήμα (4.2) είναι η γραφική απεικόνιση των συνδεδεμένων κόμβων στο nam



Σχήμα 4.2 Τοπολογία Δικτύου

#Δημιουργεί τον UDP agent και τον συνδέει στον κόμβο n0

```
set udp0 [new Agent/UDP]
$ns attach-agent $n(0) $udp0
```

#Δημιουργεί μια πηγή κίνηση(traffic source) CBR και τη συνδέει στο udp0

```
set cbr0 [new Application/Traffic/CBR]
$cbr0 set packetSize_ 500
$cbr0 set interval_ 0.005
$cbr0 attach-agent $udp0
set udp1 [new Agent/UDP]
$ns attach-agent $n1 $udp1
$udp0 set class_ 1
```

```
set cbr1 [new Application/Traffic/CBR]
```

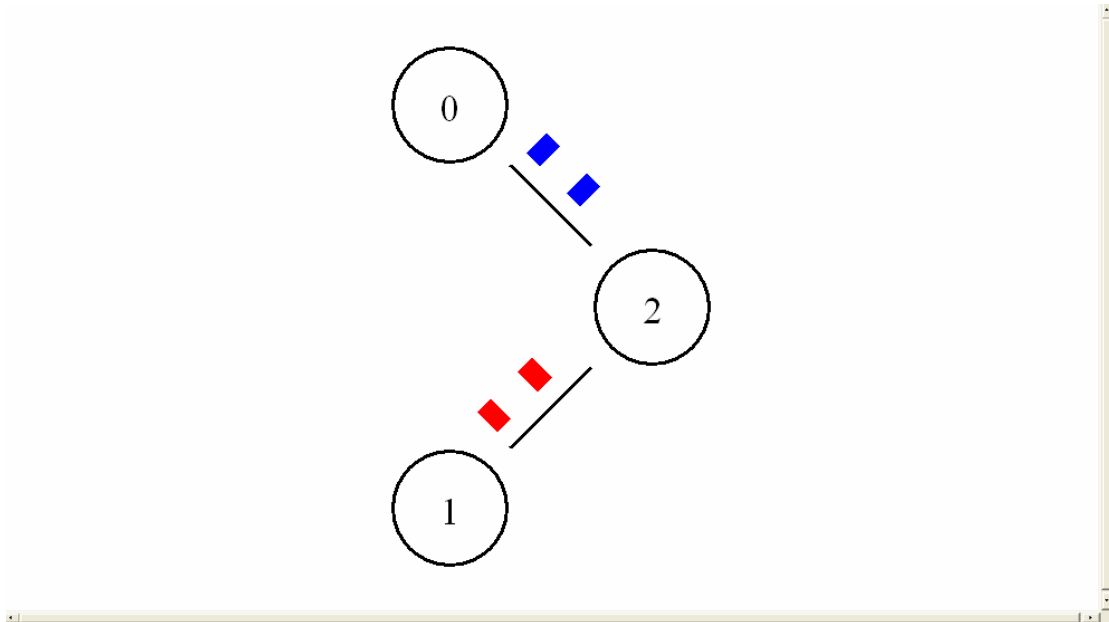
```
$cbr1 set pocketsize_ 500
$cbr1 set interval_ 0.005
$cbr1 attach-agent $udp1
$udp1 set class_ 2
```

#Δημιουργεί τον Null agent (a traffic sink) και τον συνδέει στους κόμβους n(0) και n(1).

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

```
$ns attach-agent $n2 $null0  
$ns connect $udp0 $null0  
$ns connect $udp1 $null0
```

Αυτή είναι η γραφική απεικόνιση (Σχήμα 4.3) που θα εμφανιστεί στο nam όταν τρέξουμε το script.



Σχήμα 4.3 Αποστολή δεδομένων στον κόμβο 2

#Εκτελεί την διαδικασία εκκίνησης και τερματισμού της κίνησης CBR

```
$ns at 0.5 "$cbr0 start"  
$ns at 1.0 "$cbr1 start"  
$ns at 4.0 "$cbr1 stop"  
$ns at 4.5 "$cbr0 stop"
```

#Εκτελεί την διαδικασία τερματισμού, 5 δευτερόλεπτα μετά από την έναρξη της προσομοίωσης.

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

#Εκτελεί την προσομοίωση

```
$ns run
```



## 4.5 Η TCL σε ασύρματα δίκτυα

Σε αντίθεση με τα nodes των ενσύρματων δικτύων στα ασύρματα δίκτυα έχουμε τα mobilenodes. Η βασική τους διαφορά είναι ότι τα mobilenodes έχουν πρόσθετες λειτουργίες, δηλαδή είναι ασύρματα και κινητά. Έχουν την δυνατότητα να κινούνται μέσα σ' ένα ορισμένο χώρο, να στέλνουν αλλά και να λαμβάνουν σήματα από ασύρματο κανάλι κτλ.

Ένα MobileNode αποτελείται από διάφορα συστατικά δικτύου (network component), όπως είναι το Link Layer, το Interface Queue και το MAC επίπεδο.

Με τις παρακάτω εντολές καθορίζουμε τις παραμέτρους διαμόρφωσης μιας ασύρματης τοπολογίας, δηλαδή πρέπει να ορίσουμε τον τύπο για κάθε ένα απ' τα στοιχεία του δικτύου. Δίπλα σε κάθε παράμετρο διαμόρφωσης, εμφανίζονται όλες οι πιθανές τιμές που μπορεί να πάρει η κάθε μεταβλητή.

```

set val(chan)      Channel/WirelessChannel  ;# channel type

set val(prop)     Propagation/TwoRayGround  ;# radio-propagation model

set val(netif)    Phy/WirelessPhy         ;# network interface type

set val(mac)      Mac/802_11              ;# MAC type

set val(ifq)      Queue/DropTail/PriQueue  ;# interface queue type

set val(ll)       LL                       ;# link layer type

set val(ant)      Antenna/OmniAntenna     ;# antenna model

set
val(ifqlen)      50                        ;# max packet in ifq

set val(nn)       2                        ;# number of mobilenodes

```

Μετά από τον καθορισμό των παραμέτρων περνάμε στη κύρια διαδικασία να φτιάξουμε την ασύρματη προσομοίωση:

```
# Configure nodes
  $ns_ node-config -adhocRouting $val(rp) \
    -llType $val(ll) \
    -macType $val(mac) \
    -ifqType $val(ifq) \
    -ifqLen $val(ifqlen) \
    -antType $val(ant) \
    -propType $val(prop) \
    -phyType $val(netif) \
    -topoInstance $topo \
    -channelType $val(chan) \
    -agentTrace ON \
    -routerTrace ON \
    -macTrace OFF \
    -movementTrace OFF
```

Ορίζουμε το αντικείμενο προσομοίωσης

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

Μετά δημιουργούμε το αντικείμενο τοπολογίας (topology object) το οποίο παρακολουθεί τις κινήσεις των κόμβων μέσα στα όρια που του έχουμε ορίσει.

```
set topo [new Topography]
```

Στη συνέχεια αρχικοποιείται το πλέγμα (grid) του αντικειμένου τοπολογίας. Δηλαδή ορίζονται οι x και y συντεταγμένες του ορίου. Μπορούμε επίσης να καθορίσουμε την ανάλυση του πλέγματος, μέσω τις προαιρετικής παραμέτρου <res>. Συνήθως όμως χρησιμοποιείται η προκαθορισμένη τιμή 1.

```
$topo load_flatgrid <X> <Y> <optional:res> :
```

Η εντολή αυτή δημιουργεί ένα God αντικείμενο, το οποίο χρησιμοποιείται για την αποθήκευση γενικών πληροφοριών σχετικά με την κατάσταση του ασύρματου περιβάλλοντος. Χρησιμοποιεί ως παράμετρο την μεταβλητή <num\_nodes> - τον αριθμό δηλαδή των ασύρματων κόμβων - για να δημιουργήσει έναν πίνακα, που θα περιέχει τον μικρότερο αριθμό επόμενων βημάτων (next hops) που χρειάζεται ένας κόμβος για να φτάσει κάποιον άλλο.

**create-god <num\_nodes>**

Με τις παρακάτω εντολές καθορίζουμε κάποιες πρόσθετες ρυθμίσεις των ασύρματων κόμβων.

Χρησιμοποιούμε την παρακάτω εντολή όταν θέλουμε να απενεργοποιήσουμε (τιμή 0) ή να ενεργοποιήσουμε (τιμή 1) την «τυχαία κίνηση» των mobilenodes.

**\$mobile\_node random-motion <value>**

Η εντολή αυτή ορίζει την κίνηση του mobilenode. Δηλαδή οι x και y συντεταγμένες δείχνουν την κατεύθυνση που θα αρχίσει η κίνηση και το s την ταχύτητα m/sec.

**\$mobile\_node setdest <X> <Y> <s>**

Η επαναφορά του ασύρματο κόμβου γίνεται με την παρακάτω εντολή:

**\$mobile\_node reset**

## 4.6 Υποστήριξη ιχνηλάτισης (Trace support)

Κατά την προσομοίωση εξάγονται κάποια δεδομένα, που χρησιμεύουν στην διεξαγωγή κάποιων συμπερασμάτων. Τα δεδομένα αυτά ονομάζονται trace data. Συνήθως αποθηκεύονται σε ένα αρχείο, το trace file, για να τα επεξεργαστεί και να τα αναλύσει κανείς, μετά το τέλος της προσομοίωσης. Ο τύπος παρακολούθησης ονομάζεται traces και καταγράφει διάφορα στοιχεία για κάθε πακέτο που έρχεται, φεύγει ή απορρίπτεται σε μια γραμμή ή σε μια ουρά.

Οι κυριότερες εντολές, που σχετίζονται με την δημιουργία trace αρχείων, είναι:

**\$ns trace-all** <trace\_file> : η εντολή αυτή καταγράφει όλες τις πληροφορίες της προσομοίωσης και τις αποθηκεύει στο αρχείο <trace\_file>.

**\$ns namtrace-all** <nam\_trace\_file> : η εντολή αυτή ξεκινάει το nam tracing στο NS. Όλα τα nam traces αποθηκεύονται στο <nam\_trace\_file>.

**\$ns namtrace-all-wireless** <nam\_trace\_file> <X> <Y>: η εντολή αυτή ξεκινάει το wireless nam tracing. Τα <X> και <Y> είναι οι συντεταγμένες της ασύρματης τοπολογίας και όλα τα wireless nam traces αποθηκεύονται στο <nam\_trace\_file>.

**\$ns nam-end-wireless** <stop\_time> : η εντολή αυτή λέει στο nam να σταματήσει την χρονική στιγμή <stop\_time>.

**\$ns flush-trace** : η εντολή αυτή αδειάζει τον trace buffer και καλείται λίγο πριν το τέλος της προσομοίωσης.

### 4.6.1 Μορφή Trace File

Ανάλογα με τον τύπο του δικτύου, ενσύρματο ή ασύρματο, έχουμε και διαφορετική μορφή αρχείου. Παρακάτω παραθέτουμε αναλυτική εξήγηση των μορφών του trace file.

Μορφή trace file σε ενσύρματο δίκτυο :

event	time	From node	To node	Pkt type	Pkt size	flags	fid	Src address	Dst address	Seq number	Pkt id
+	1.84375	0	2	cbr	210	-----	0	0.0	3.1	225	610
-	1.84375	0	2	cbr	210	-----	0	0.0	3.1	225	610
r	1.84471	2	1	cbr	210	-----	1	3.0	1.0	195	600
r	1.84566	2	0	ack	40	-----	2	3.2	0.1	82	602
+	1.84566	0	2	tcp	1000	-----	2	0.1	3.2	102	611
-	1.84566	0	2	tcp	1000	-----	2	0.1	3.2	102	611
r	1.84609	0	2	cbr	210	-----	0	0.0	3.1	225	610
+	1.84609	2	3	cbr	210	-----	0	0.0	3.1	225	610
d	1.84609	2	3	cbr	210	-----	0	0.0	3.1	225	610
-	1.8461	2	3	cbr	210	-----	0	0.0	3.1	192	511
r	1.84612	3	2	cbr	210	-----	1	3.0	1.0	196	603
+	1.84612	2	1	cbr	210	-----	1	3.0	1.0	196	603
-	1.84612	2	1	cbr	210	-----	1	3.0	1.0	196	603
+	1.84625	3	2	cbr	210	-----	1	3.0	1.0	199	612

Στο παράδειγμα μας έχουμε τη μορφή καταγραφής των αποτελεσμάτων μιας προσομοίωσης σε ένα trace file. Η Κάθε εγγραφή αποτελείται από δώδεκα στήλες.

Η πρώτη στήλη, «event», δηλαδή το γεγονός, δείχνει τον τύπο της διεργασίας που περιγράφει η συγκεκριμένη εγγραφή. Η εισαγωγή στην ουρά συμβολίζεται με το «+» ενώ η αποχώρηση από την ουρά με το «-». Το γεγονός παραλαβής του πακέτου συμβολίζεται με το «r» και της απόρριψης με το «d». Στην δεύτερη στήλη εμφανίζεται ο χρόνος προσομοίωσης, σε δευτερόλεπτα, στον οποίο συνέβη κάποιο από τα γεγονότα που αναφέραμε παραπάνω.

Οι επόμενες δύο στήλες, η τρίτη και η τέταρτη, μας δείχνουν από ποιόν κόμβο συμβαίνει το γεγονός και προς ποιόν κόμβο.

Η πέμπτη στήλη συμβολίζει τον τύπο του πακέτου.

Στην συνέχεια έχουμε το μέγεθος του πακέτου όπως αυτό κωδικοποιείται στην IP επικεφαλίδα.

Οι τέσσερις επόμενοι χαρακτήρες αναπαριστούν κάποιες ειδικές σημαίες που μπορεί να είναι ενεργοποιημένες. Προς το παρόν χρησιμοποιείται μόνο μια τέτοια σημαία η «Σαφή Κοινοποίηση

Συμφόρησης» (ECN, explicit congestion notification). Στο παράδειγμα μας δεν χρησιμοποιείται.

Η όγδοη στήλη μας δίνει το IP *flow identifier* όπως αυτό καθορίζεται στο IP έκδοση 6.1.

Οι επόμενες δύο στήλες καθορίζουν τις διευθύνσεις των κόμβων του παραλήπτη και του αποστολέα.

Η προτελευταία στήλη μας δείχνει τον σειριακό αριθμό του πακέτου. Το τελευταίο πεδίο είναι ένας δείκτης μοναδικός για κάθε πακέτο που αποδίδεται σε κάθε πακέτο με τη στιγμή της δημιουργίας του.

Μορφή trace file σε ασύρματο δίκτυο :

Η γενική μορφή αυτών των αρχείων στα ασύρματα έχει ως εξής :

f -t 33.195597851 -Hs 8 -Hd 9 -Ni 8 -Nx 750.00 -Ny 500.00 -Nz 0.00
Ne-1.000000 -NI RTR -Nw --- Ma 13a -Md 8 -Ms 7 -Mt 800 -Is 5.0 -
Id 9.0 -It cbr -Il 552 -If 1 -Ii 27 -Iv 29 -Pn cbr -Pi 6 -Pf 3 -Po 4

Η πρώτη στήλη δείχνει ποιο γεγονός συμβαίνει. Αν είναι «s», το πακέτο στέλνεται, αν είναι «r», παραλαμβάνεται και αν είναι «f», προωθείται.

Στην δεύτερη στήλη βλέπουμε την ώρα που γίνεται το γεγονός.

Η τρίτη και η τέταρτη στήλη περιέχουν τον αριθμό της επόμενης αναπήδησης.

Η επόμενη στήλη αναπαριστά έναν μοναδικό αριθμό για κάθε κόμβο.

Στην συνέχεια έχουμε τις X/Y/Z συντεταγμένες του κόμβου.

Στην ένατη στήλη βλέπουμε την ενέργεια του κόμβου.

Η δέκατη στήλη δείχνει το επίπεδο του δικτύου στο οποίο βρίσκεται το γεγονός.

Η ενδέκατη στήλη δείχνει τον λόγο που απορρίφθηκε ένα πακέτο.

Ακολουθεί η διάρκεια του γεγονότος.

Στις επόμενες δύο στήλες δίνετε η MAC διεύθυνση του αποστολέα και του παραλήπτη.

Στην δέκατη πέμπτη στήλη έχουμε τον τύπο του MAC.

Η δέκατη έκτη και έβδομη στήλη είναι η IP διεύθυνση του αποστολέα και του παραλήπτη.

Η δέκατη όγδοη στήλη δείχνει τον τύπο του πακέτου και η επόμενη το μήκος του πακέτου.

Ακολουθεί η ροή του-If Flow ID.

Η επόμενη στήλη έχει τον αριθμό που κάνει κάθε πακέτο μοναδικό.

Η εικοστή δεύτερη στήλη δείχνει την διάρκεια ζωής του πακέτου.

-Pn Packet Specific Trace Start

Στην εικοστή τέταρτη στήλη περιέχεται ο σειριακός αριθμός του πακέτου.

Στην προτελευταία στήλη βλέπουμε τον αριθμό προωθήσεων και τέλος βλέπουμε τον καταλληλότερο αριθμό προωθήσεων

Στα ad-hoc ασύρματα δίκτυα όμως η μορφή του διαφοροποιείται ως εξής :

Event	Time	Node	Layer	Flags	Pkt_id	Pkt_type	Pkt_size						
s	42,1175	_0_	RTR	---	2545	tcp	1060	[0 0 0 0]	-	[0:0 1:00 32 1]	[1250 0]	0	0
r	42,1197	_0_	AGT	---	2543	ack	60	[13a 0 1 800]	-	[1:0 0:00 32 0]	[1243 0]	1	0
s	42,1197	_0_	AGT	---	2546	tcp	1040	[0 0 0 0]	-	[0:0 1:00 32 0]	[1251 0]	0	0
r	42,1197	_0_	RTR	---	2546	tcp	1040	[0 0 0 0]	-	[0:0 1:00 32 0]	[1251 0]	0	0
s	42,1197	_0_	RTR	---	2546	tcp	1060	[0 0 0 0]	-	[0:0 1:00 32 1]	[1251 0]	0	0
r	42,1297	_1_	AGT	---	2534	tcp	1060	[13a 1 0 800]	-	[0:0 1:00 32 1]	[1245 0]	1	0
s	42,1297	_1_	AGT	---	2547	ack	40	[0 0 0 0]	-	[1:0 0:00 32 0]	[1245 0]	0	0
r	42,1297	_1_	RTR	---	2547	ack	40	[0 0 0 0]	-	[1:0 0:00 32 0]	[1245 0]	0	0
s	42,1297	_1_	RTR	---	2547	ack	60	[0 0 0 0]	-	[1:0 0:00 32 0]	[1245 0]	0	0
r	89,9926	_1_	RTR	---	10205	ack	40	[0 0 0 0]	-	[1:0 0:00 32 0]	[5013 0]	0	0
s	89,9926	_1_	RTR	---	10205	ack	60	[0 0 0 0]	-	[1:0 0:00 32 0]	[5013 0]	0	0
D	90	_0_	IFQ	---	10201	tcp	1060	[0 1 0 800]	-	[0:0 1:00 32 1]	[5015 0]	0	0

Η πρώτη στήλη είναι το "event", το οποίο μας δείχνει αν στέλνεται «s», αν παραλαμβάνεται «r» ή αν έχει χαθεί «D».

Στη συνέχεια έχουμε το «time», δηλαδή τη χρονική στιγμή που συνέβη το γεγονός.

Ακολουθεί το «node», ο κόμβος δηλαδή στον οποίο λαμβάνει μέρος το γεγονός.

Στην τέταρτη στήλη το «layer» μας δείχνει σε ποιο επίπεδο βρίσκεται το γεγονός.

Στην πέμπτη στήλη τα «flags» είναι σημαίες τις οποίες μπορούμε να έχουμε κάποιες από αυτές ή όλες ενεργοποιημένες ή απενεργοποιημένες.

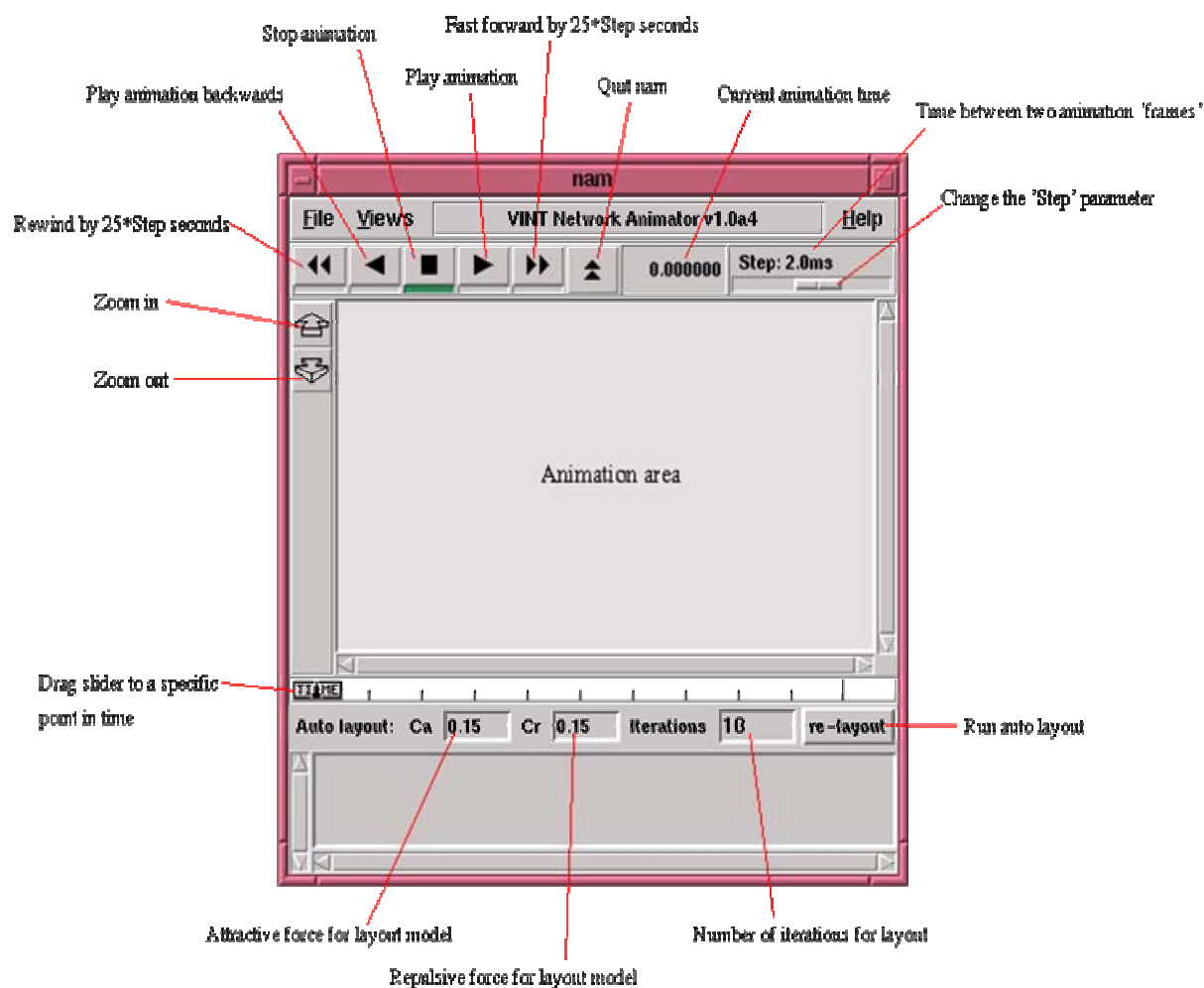
Στη έκτη στήλη έχουμε έναν δείκτη ο οποίος είναι μοναδικός για κάθε πακέτο και του αποδίδεται τη στιγμή της δημιουργίας του.

Στην επόμενη στήλη ακολουθεί το μέγεθος του πακέτου.

#### **4.7 NETWORK ANIMATOR (NAM)**

Ο network animator (NAM) είναι μία γραφική διεπαφή με την οποία μπορούμε να παρουσιάσουμε εικονικά την τοπολογία του δικτύου που δημιουργήσαμε καθώς και την ανταλλαγή και τον αριθμό πακέτων μεταξύ των κόμβων. Στην αρχή δημιουργείται το αρχείο nam\_trace, το οποίο διαβάζει εν συνεχεία ο nam, που δημιουργεί την δικτυακή τοπολογία και εμφανίζει το παρακάτω παράθυρο:





Σχήμα 4.4 Γραφικό περιβάλλον του NAM

Με τις παρακάτω εντολές καθορίζουμε τα γραφικά χαρακτηριστικά των κόμβων(π.χ. χρώμα, σχήμα)

Καθορίζει το χρώμα του κόμβου:

`$node color [color]`

Καθορίζει το σχήμα του κόμβου που μπορεί να έχει τρεις πιθανές τιμές κύκλος(που είναι η προκαθορισμένη), τετράγωνο και εξάγωνο:

`$node shape [shape]`

Η εντολή αυτή θέτει κάποια ετικέτα στον κόμβο:

`$node label [label]`

Η εντολή αυτή καθορίζει το χρώμα της ετικέτας:

`$node label-color [color]`

Η εντολή αυτή μαρκάρει τον συγκεκριμένο κόμβο. Μπορούμε να δώσουμε στο μαρκάρισμα συγκεκριμένο χρώμα, όνομα και σχήμα.

`$node add-mark [name] [color] [shape]`

Η εντολή αυτή αναφέρεται στα χαρακτηριστικά που μπορεί να έχει μια γραμμή:

`$ns duplex-link-op <attribute> <value>`

Έτσι η παράμετρος <attribute> παίρνει τις τιμές: orient, color, queuePos, label. Το <attribute> orient καθορίζει την γωνία ανάμεσα στη γραμμή με το οριζόντιο επίπεδο και οι τιμές του μπορεί να εκφράζονται είτε σε μοίρες είτε με κείμενο όπως right (0), right-up (45), left (180), left-down (135), up (90), down (-90). Το <attribute> queuePos καθορίζει την γωνία ανάμεσα στην ουρά ενός κόμβου με το οριζόντιο επίπεδο. Δίνονται κάποια σχετικά παραδείγματα για κάθε <attribute>:

`$ns duplex-link-op $n1 $n2 orient right`

`$ns duplex-link-op $n1 $n2 color green`

`$ns duplex-link-op $n1 $n2 queuePos 0.5`

`$ns duplex-link-op $n1 $n2 label`

## 4.8 Οδηγίες για την εγκατάσταση και εκτέλεση του Προσομοιωτή Network Simulator 2 σε περιβάλλον windows.

Απαραίτητα αρχεία:

Για την εγκατάσταση και εκτέλεση του Network Simulator είναι απαραίτητα τα ακόλουθα αρχεία:

- [tcl830.exe](#) (Ο installer της γλώσσας tcl)
- [ns-2.1b9a-win32.exe](#) (Το εκτελέσιμο αρχείο του Network Simulator2)
- [nam-1.0a11a-win32.exe](#) (Το εκτελέσιμο αρχείο του Network Animator)
- [awk95.exe](#) (Ο interpreter της γλώσσας awk)

### 4.8.1 Για την εγκατάσταση των εκτελέσιμων αρχείων:

1. Εγκατάσταση της γλώσσας tcl

Αρχικά πρέπει να εγκατασταθούν οι βιβλιοθήκες της γλώσσας tcl.

Για να γίνει αυτό εκτελούμε τον installer tcl830.exe

2. Αντιγραφή εκτελέσιμων αρχείων στην τοποθεσία εγκατάστασης.

Στην συνέχεια πρέπει να αντιγραφούν τα εκτελέσιμα αρχεία του Network Simulator 2 (**ns-2.1b9a-win32.exe**), του Network Animator (**nam-1.0a11a-win32.exe**) και της γλώσσας awk (**awk95.exe**) σε κάποιο φάκελο του τοπικού συστήματος. Ο φάκελος εγκατάστασης δεν πρέπει να περιέχει κενά και μεγάλα ονόματα. (π.χ c:\netsim\)

Για ευκολία στην εκτέλεση μετονομάζουμε τα αρχεία σε **ns.exe**, **nam.exe** και **awk.exe** αντίστοιχα.

### 3. Ενημέρωση της μεταβλητής PATH

Για να είναι δυνατή εκτέλεση των αρχείων του network simulator από οποιαδήποτε τοποθεσία του συστήματος αρχείων, πρέπει να ενημερωθεί η μεταβλητή PATH με την τοποθεσία εγκατάστασης των εκτελέσιμων αρχείων του Network Simulator. Η διαδικασία ενημέρωσης της μεταβλητής PATH εξαρτάται από το λειτουργικό σύστημα:

Σε windows 98 η μεταβλητή PATH καθορίζεται από το αρχείο c:\autoexec.bat. Αν τα εκτελέσιμα του Network Simulator έχουν εγκατασταθεί στον κατάλογο c:\netsim τότε η γραμμή που θα προστεθεί στο τέλος του αρχείου c:\autoexec.bat είναι:

**SET PATH = %PATH%;c:\netsim\**

Σε περιβάλλον windows 2000 και windows XP για την ενημέρωση της μεταβλητής PATH γίνεται η εξής διαδικασία: Κάνουμε δεξί κλικ στο εικονίδιο "My computer" και επιλέγουμε ιδιότητες. Από το tab "Advanced" επιλέγουμε "Environment Variables". Στη συνέχεια με "Edit" προσθέτουμε στην μεταβλητή PATH τον κατάλογο της εγκατάστασης του Network Simulator.

4. Η διαδικασία εγκατάστασης έχει ολοκληρωθεί

### 4.8.2 Εκτέλεση των προγραμμάτων προσομοίωσης

Η εκτέλεση των προγραμμάτων προσομοίωσης γίνεται από την γραμμή εντολών (command line). Για να ανοίξει η γραμμή εντολών σε windows 2000 και windows XP επιλέγουμε start -> run και στο

παράθυρο που ανοίγει πληκτρολογούμε **cmd**. Σε windows 98 Επιλέγουμε start->run και πληκτρολογούμε command.

Για να δημιουργήσουμε ή να επεξεργαστούμε ένα αρχείο κώδικα tcl για προσομοίωση στο Network Simulator πληκτρολογούμε από τη γραμμή εντολών:

`notepad script.tcl`

όπου script.tcl είναι το όνομα του αρχείου που θα επεξεργαστούμε. Αφού δημιουργηθεί το αρχείο προσομοίωσης σύμφωνα με τις οδηγίες της κάθε άσκησης, εκτελούμε το πρόγραμμα με την εντολή

`ns script.tcl`

Αν δεν υπάρχουν λάθη στον κώδικα και η προσομοίωση ολοκληρωθεί επιτυχώς τότε θα έχουν δημιουργηθεί τα ανάλογα trace αρχεία σύμφωνα με τον κώδικα προσομοίωσης.

Εφόσον έχει δημιουργηθεί κάποιο animation της προσομοίωσης, αυτό μπορεί να εκτελεστεί με την εντολή

`nam out.nam`

όπου out.nam είναι το αρχείο που περιλαμβάνει το animation. (καθορίζεται με την εντολή namtrace-all στο αρχείο tcl.

Σε κάποιες προσομοιώσεις δημιουργούμε και αρχεία out.tr που περιλαμβάνουν κάποια γραφική παράσταση. Για ευκολία προτείνεται η χρήση του excel. Αφού ανοίξουμε το πρόγραμμα excel, επιλέγουμε File->Open, και από το παράθυρο που εμφανίζεται επιλέγουμε το αρχείο out.tr που δημιουργήθηκε από την προσομοίωση (Αν αυτό δεν εμφανίζεται επιλέγουμε στο πεδίο "File of type"→All Files). Στη συνέχεια, με τη βοήθεια του "Text import wizard" που εμφανίζεται επιλέγουμε Delimited με space για delimitator. Επίσης, πατάμε το

κουμπι advanced ώστε να διαλέξουμε την τελεία ως υποδιαστολή. Αφού ολοκληρωθεί ο wizard, θα πρέπει να δεδομένα από το αρχείο out.tr να εμφανίζονται σε δύο στήλες του excel. Επιλέγοντας αυτές τις δύο στήλες, και με την βοήθεια των εντολών Insert→Graph→XY Scatter, μπορούμε να εμφανίσουμε την γραφική παράσταση.

Αν τα αρχεία .tr αντί να περιλαμβάνουν μόνο μια γραφική παράσταση, περιλαμβάνουν περισσότερες πληροφορίες σχετικά την αλληλουχία γεγονότων που συνέβησαν κατά τη διάρκεια της πρωσομοίωσης. Τότε για την επεξεργασία αυτών των αρχείων πρέπει να δημιουργηθεί πρόγραμμα σε γλώσσα awk, σύμφωνα με τις οδηγίες της κάθε άσκησης. Στη συνέχεια το πρόγραμμα εκτελείται με την εντολή

```
awk -f trace.awk <out.tr
```

όπου trace.awk είναι το αρχείο με τις οδηγίες επεξεργασίας και out.tr είναι το trace αρχείο της προσομοίωσης. Σε περίπτωση που θέλουμε να αποθηκεύσουμε τα outputs της ανάλυσης των δεδομένων σε κάποιο αρχείο (πχ results.txt) εκτελούμε την εντολή

```
awk -f analyze.awk <out.tr >results.txt
```

## **ΚΕΦΑΛΑΙΟ 5**

### **ΠΕΙΡΑΜΑΤΙΚΟ ΜΕΡΟΣ**

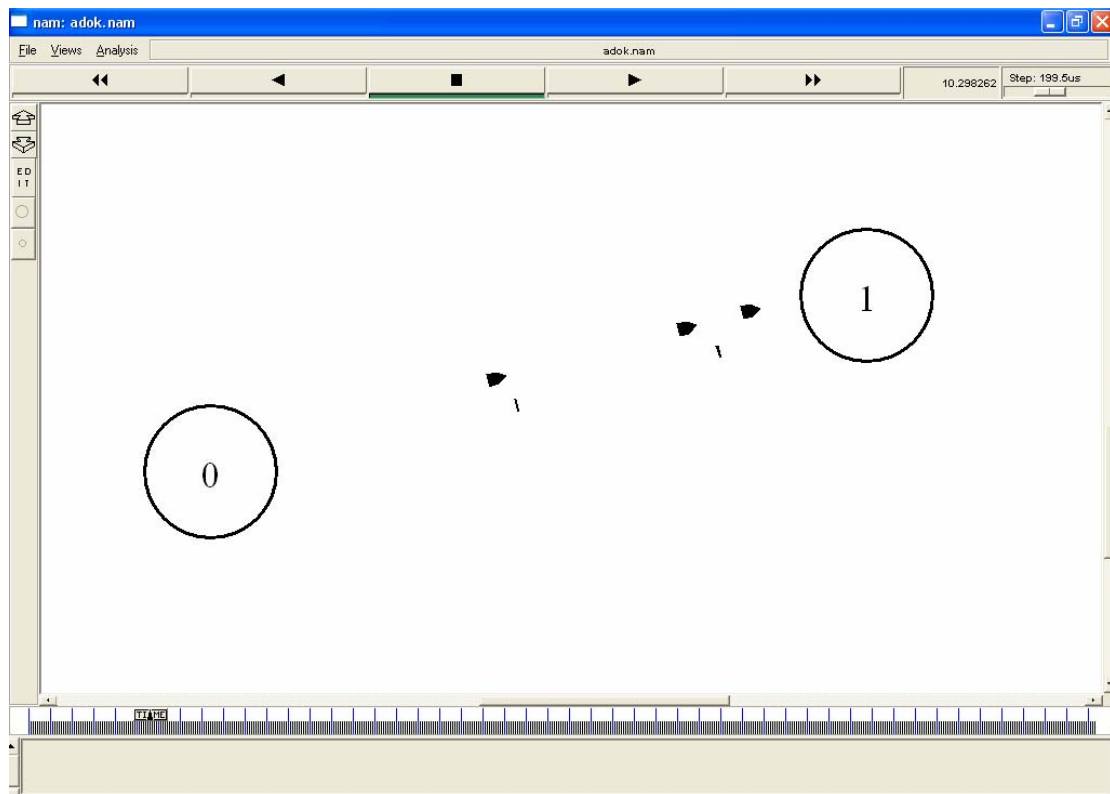
#### **5.1 Εισαγωγή**

Το πειραματικό μέρος της εργασίας συνίσταται στην εξομοίωση διάφορων τοπολογιών 802.11 και στη μελέτη των επιδόσεών τους. Η έκδοση του NS που χρησιμοποιήθηκε είναι η πιο πρόσφατη (2.26), και συνοδεύεται από την τελευταία έκδοση του προγράμματος απεικόνισης NAM. Στην έκδοση αυτή, το NAM έχει πλέον τη δυνατότητα απεικόνισης ασύρματων δεδομένων. Για την επεξεργασία των αποτελεσμάτων του NS, χρησιμοποιήθηκε η γλώσσα προγραμματισμού AWK. Τέλος, κάναμε ευρεία χρήση του προγράμματος Microsoft Excel για την απεικόνιση των γραφημάτων.

#### **5.2 Πρώτο Σενάριο**

##### **5.2.1 Γενική Περιγραφή**

Στο πρώτο παράδειγμα εξομοιώθηκε ένα απλούστατο δίκτυο ad-hoc, αποτελούμενο από δύο ασύρματους κόμβους που επικοινωνούν μεταξύ τους. Στην ουσία ο ένας κόμβος είναι ο αποστολέας και ο άλλος ο παραλήπτης. Στο πρώτο σκέλος του πειράματος οι κόμβοι είναι ακίνητοι και σε σχετικά μικρή απόσταση μεταξύ τους. Η προσομοίωση διαρκεί για 90.0 sec. Η αποστολή δεδομένων ξεκινά στα 10.0 sec και ολοκληρώνεται στα 90.0 sec. Ο κώδικας του σεναρίου βρίσκεται στο Παράρτημα Α. Παρακάτω φαίνεται μια εικονική αναπαράσταση στο NAM της τοπολογίας του δικτύου.



Σχήμα 5.1 Αναπαράσταση του πρώτου σεναρίου στο NAM

## 5.2.2 Αρχείο εξόδου

Μετά την εκτέλεση του κώδικα, σειρά έχει η κατάλληλη επεξεργασία του προκύπτοντος trace file. Αυτό έχει την εξής μορφή:

Event	Time	Node	Layer	Flags	Pkt_id	Pkt_type	Pkt_size	
s	10,00378	_0_	AGT	---	4	tcp	1000	
r	10,00378	_0_	RTR	---	4	tcp	1000	
.....								
s	10,01428	_1_	AGT	---	7	ack	40	
r	10,01428	_1_	RTR	---	7	ack	40	
s	10,01428	_1_	RTR	---	7	ack	60	
r	10,01543	_0_	AGT	---	6	ack	60	
s	10,01543	_0_	AGT	---	8	tcp	1000	Αποστολή πακέτου από τον κόμβο _0
r	10,01543	_0_	RTR	---	8	tcp	1000	Μεταβίβαση του πακέτου στον router
s	10,01543	_0_	RTR	---	8	tcp	1020	Αποστολή πακέτου από τον router
s	10,01543	_0_	AGT	---	9	tcp	1000	
r	10,01543	_0_	RTR	---	9	tcp	1000	
s	10,01543	_0_	RTR	---	9	tcp	1020	



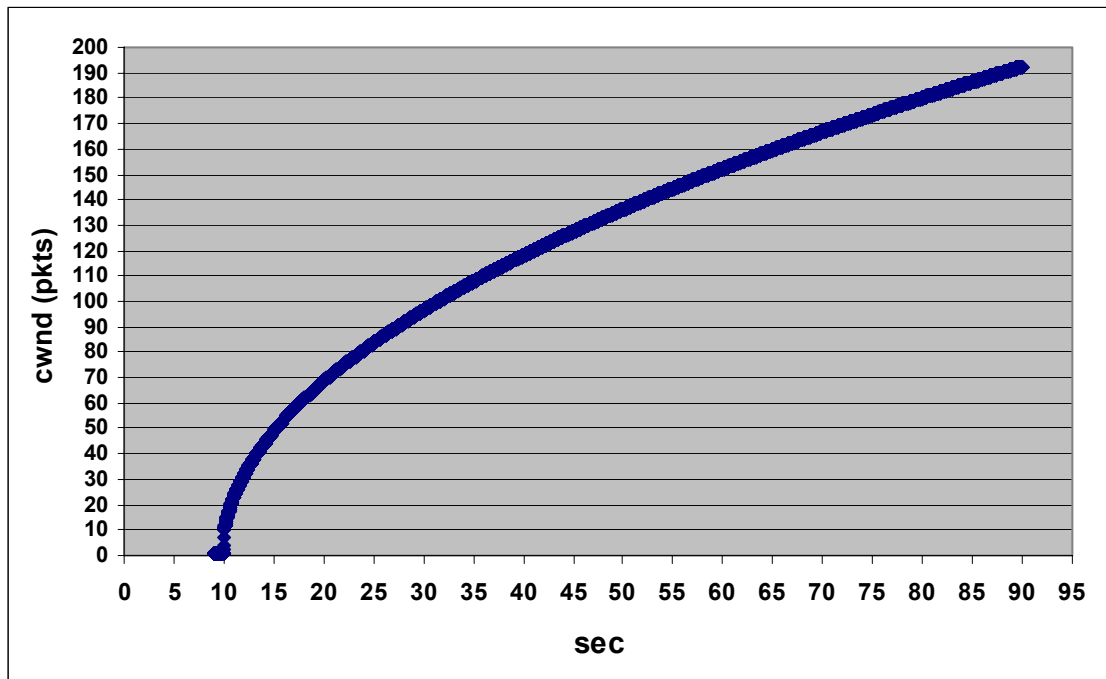
r	10,02042	_1_	AGT	---	8	tcp	1020	Λήψη του πακέτου από τον κόμβο_1
s	10,02042	_1_	AGT	---	10	ack	40	
r	10,02042	_1_	RTR	---	10	ack	40	
s	10,02042	_1_	RTR	---	10	ack	60	
r	10,02175	_0_	AGT	---	7	ack	60	
s	10,02175	_0_	AGT	---	11	tcp	1000	
r	10,02175	_0_	RTR	---	11	tcp	1000	
s	10,02175	_0_	RTR	---	12	tcp	1020	
r	10,02682	_1_	AGT	---	9	tcp	1020	
r	10,02821	_0_	AGT	---	10	ack	60	

Ας μελετήσουμε την πορεία ενός πακέτου TCP που στέλνεται από τον κόμβο 0 και λαμβάνεται από τον κόμβο 1. Ένα τέτοιο πακέτο είναι π.χ. το πακέτο 8 (packet id, δηλαδή ο αριθμός μετά τις τρεις παύλες). Βλέπουμε λοιπόν ότι το πακέτο αυτό στέλνεται αρχικά από τον κόμβο 0 (χρόνος 10.0154) και έχει μέγεθος 1000. Ταυτόχρονα μεταβιβάζεται στο RTR, οπότε και το μέγεθός του αυξάνεται στα 1020. Κατόπιν λαμβάνεται από το AGT επίπεδο του κόμβου 1 με μέγεθος 1020 (χρόνος 10.0204). Πακέτα με την αντίστροφη πορεία (πχ. το 10) ακολουθούν εντελώς αντίστοιχη λογική, κάτι που ισχύει και για τις επιβεβαιώσεις (ACK).

### 5.2.3 Επεξεργασία των αποτελεσμάτων

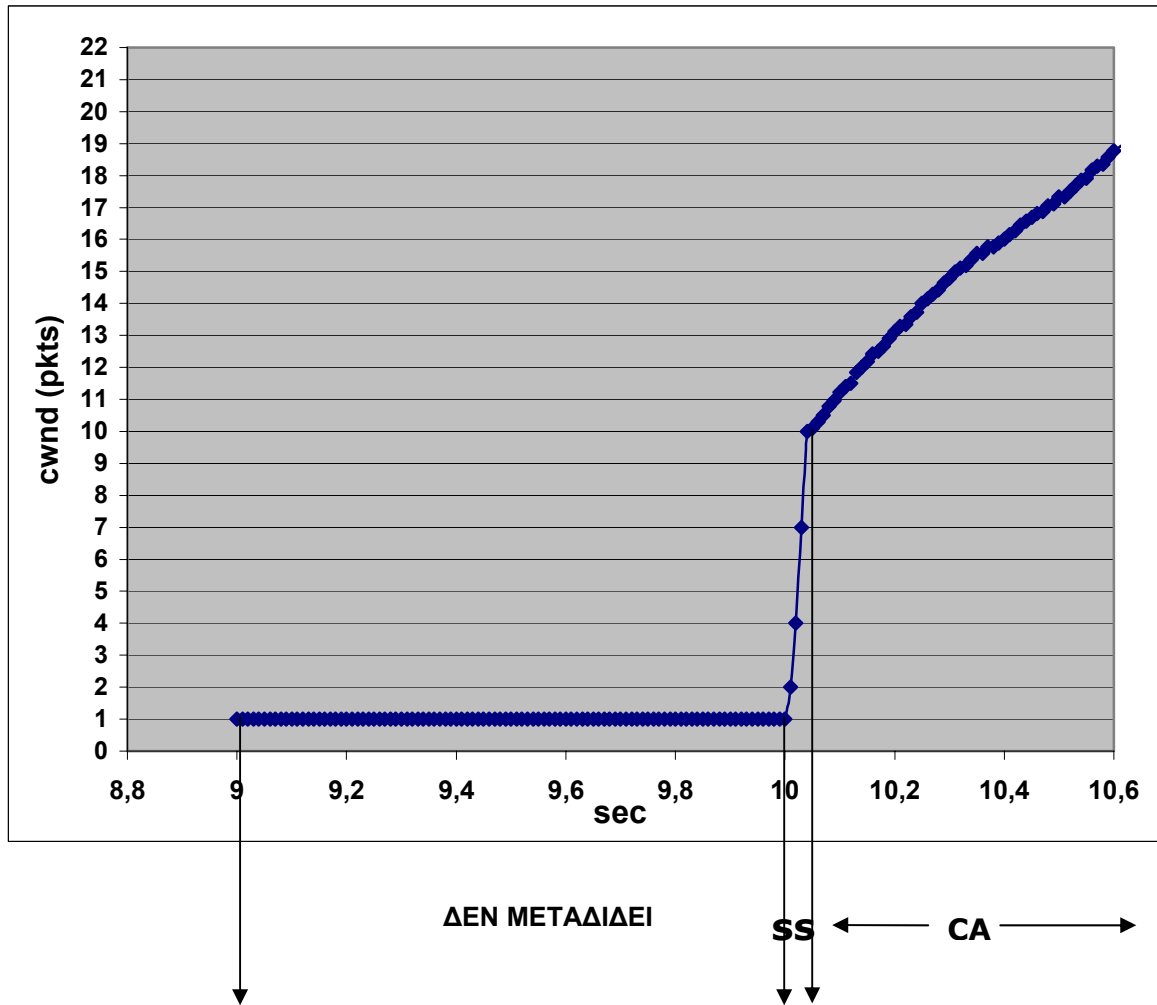
#### 5.2.3 .1 Επεξεργασία του cwnd

Στα πρώτα 10 δευτερόλεπτα δεν υπάρχει επικοινωνία μεταξύ των δύο κόμβων οπότε το cwnd είναι ίσο με ένα πακέτο. Από την χρονική στιγμή που ξεκινά η αποστολή δεδομένων, δηλαδή τα 10.0 sec, μέχρι τα 10.049 sec όπου το cwnd γίνεται ίσο με το ssthresh, στην προκειμένη περίπτωση τα 10 πακέτα, και εκτελείται η διαδικασία Slow Start. Έπειτα μπαίνει στη διαδικασία Congestion Avoidance και διατηρείται σε αυτήν μέχρι να δημιουργηθεί συμφόρηση ή να τελειώσει η επικοινωνία μεταξύ των κόμβων.



Σχήμα 5.2 Γράφημα του cwnd (pkts) προς χρόνο (sec)

Στο παραπάνω σχήμα (5.2) παρατηρούμε ότι δεν υπάρχει συμφόρηση γι' αυτό και το cwnd αυξάνεται εκθετικά μέχρι να γίνει ίσο με το κατώφλι και γραμμικά μέχρι να συμβεί συμφόρηση. Όπως αναφέρουμε και στην θεωρία, το μέγεθος του παραθύρου του αποστολέα (window) ισούται με  $window = \min \{cwnd, awnd\}$ . Από την στιγμή που το cwnd εξισωθεί με την τιμή του awnd, το window θα πάρει, ως μέγιστο όριο του, την τιμή του awnd αφού είναι η ελάχιστη. Εφόσον στο παράδειγμά μας δεν υπάρχει συμφόρηση, το window θα στέλνει το μέγιστο αριθμό πακέτων τον οποίο του έχει ορίσει ο παραλήπτης.



**SS : Slow – Start**  
**CA : Congestion Avoidance**

Σχήμα 5.3 Ανάλυση της αλλαγής μεταξύ Slow-Start και Congestion Avoidance

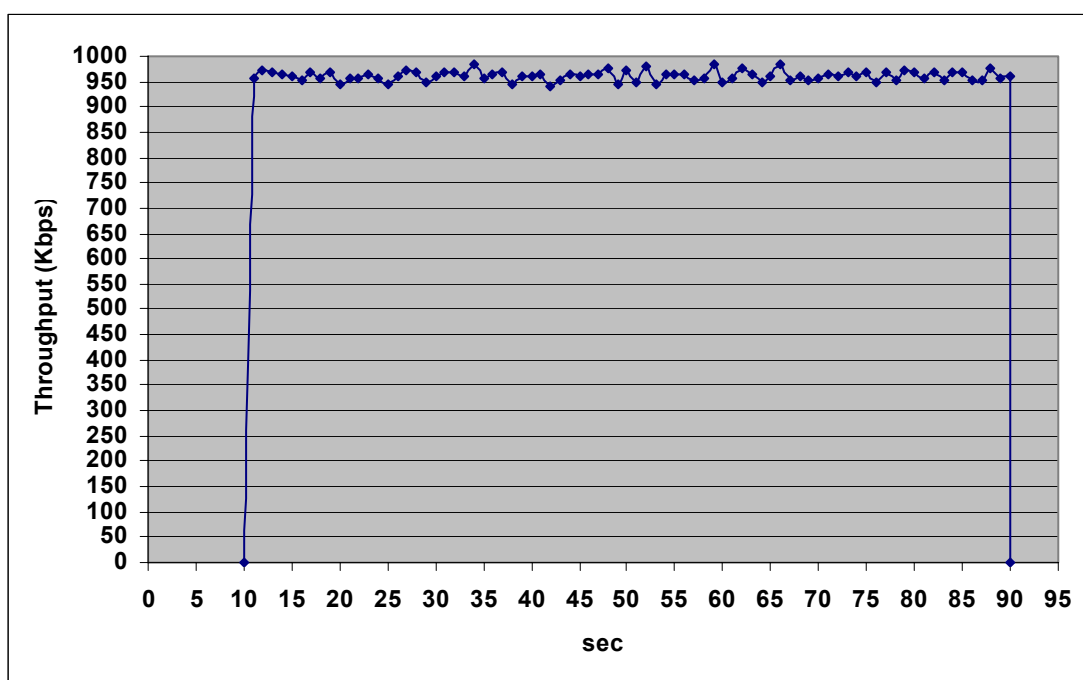
### 5.2.3 .2 Επεξεργασία του throughput

Για τον υπολογισμό του throughput του παραλήπτη χρειάζεται να απομονώσουμε τις γραμμές του trace file στις οποίες γίνεται λήψη από το AGT επίπεδο από του κόμβου\_1 και αφού υπολογίσουμε τον συνολικό όγκο δεδομένων να το διαιρέσουμε δια τον χρόνο που διαρκεί η αποστολή δεδομένων. Στην προκειμένη περίπτωση είναι τα 80 sec. Όλες αυτές οι λειτουργίες επιτελούνται από το πρόγραμμα,

γραμμένο στην (ειδική για φίλτρα) γλώσσα AWK, που βρίσκεται στο Παράρτημα Α (6.1).

### 5.2.3.3 Μετρήσεις

Εφαρμογή του φίλτρου AWK στο trace file δίνει τα εξής αποτελέσματα:



Σχήμα 5.4 Γράφημα του Throughput σε Kbps προς χρόνο σε sec

## 5.3 Δεύτερο Πείραμα

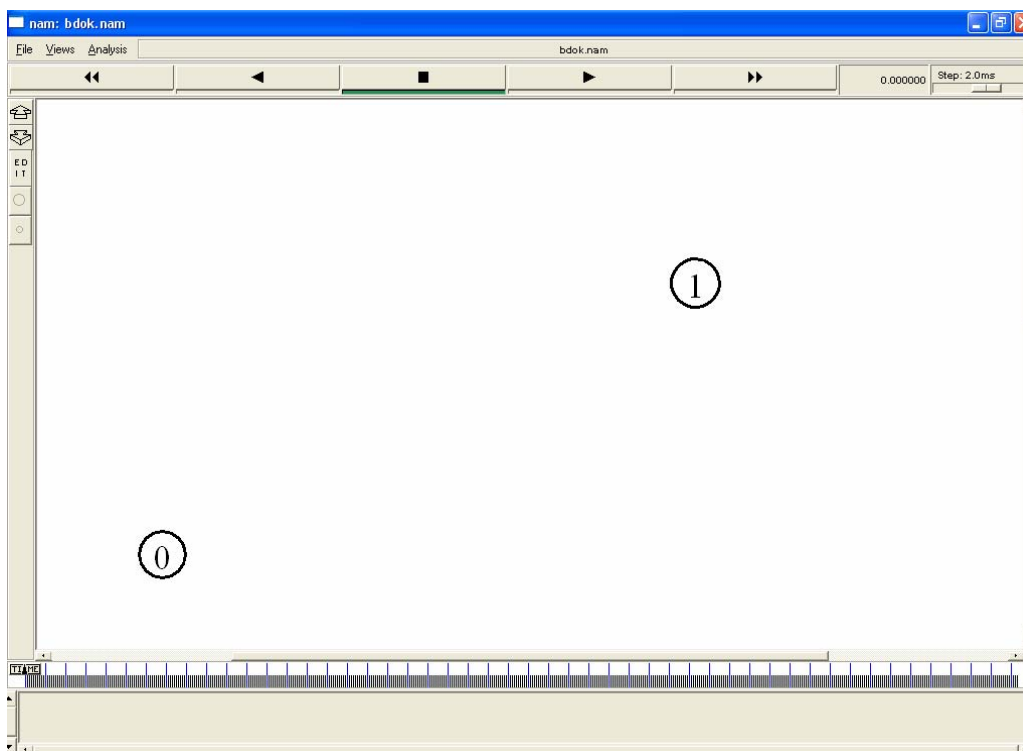
### 5.3.1 Γενική Περιγραφή

Το σενάριο μας στο δεύτερο παράδειγμα αποτελείται από δύο ασύρματους κινητούς κόμβους που επικοινωνούν μεταξύ τους. Ο ένας κόμβος είναι ο αποστολέας και ο άλλος ο παραλήπτης. Οι κόμβοι αρχικά κινούνται πλησιάζοντας ο ένας τον άλλον και από

κάποια χρονική στιγμή και μετά αρχίζουν να απομακρύνονται μέχρι να χαθεί η εμβέλεια τους. Η προσομοίωση διαρκεί για 90.0 sec. Η αποστολή δεδομένων ξεκινά στα 10.0 sec και ολοκληρώνεται στα 90.0 sec. Ολόκληρος ο κώδικας του σεναρίου βρίσκεται στο Παράρτημα Α (6.3). Να τονίσουμε ότι η κίνηση των κόμβων γίνεται με τις παρακάτω σειρές του κώδικα :

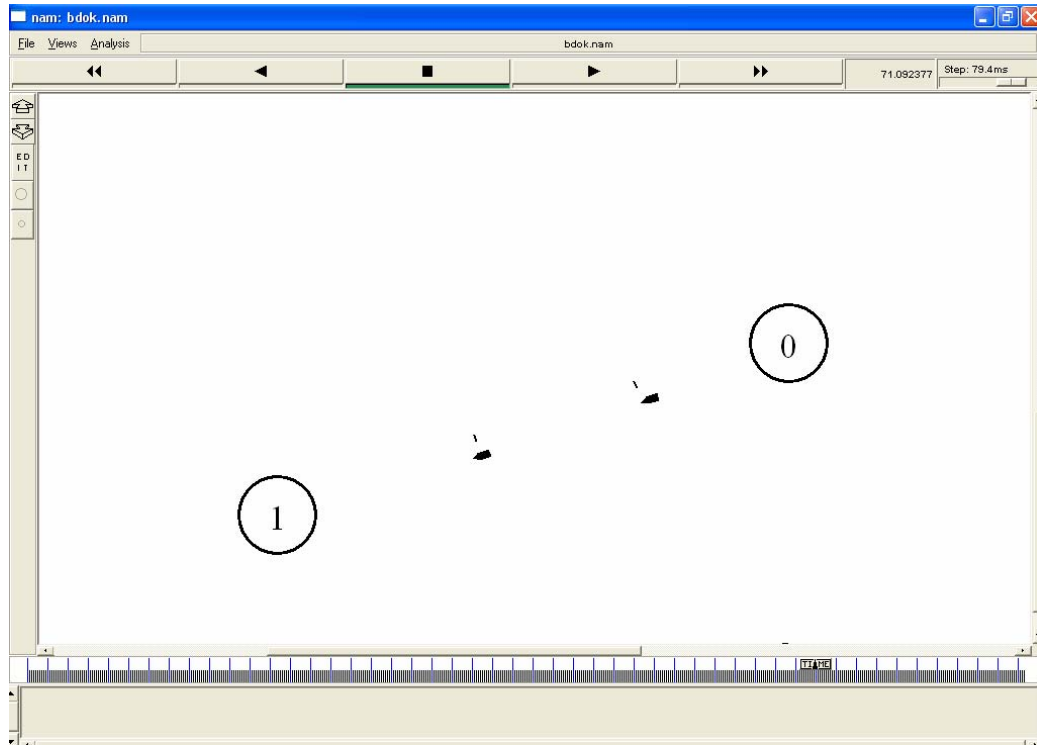
```
$ns at 50.0 "$node_(1) setdest 25.0 20.0 15.0"  
$ns at 10.0 "$node_(0) setdest 170.0 35.0 25.0"  
$ns at 100.0 "$node_(1) setdest 490.0 480.0 15.0"  
$ns at 85.0 "$node_(0) setdest 390.0 200.0 20.0"
```

Στο παρακάτω σχήμα βλέπουμε, στο NAM, τις αρχικές θέσεις των κόμβων:



Σχήμα 5.5 Αρχικές θέσεις των κόμβων του 2<sup>ου</sup> σεναρίου

Στο σχήμα που ακολουθεί βλέπουμε, στο NAM, τις θέσεις των κόμβων μετά την κίνηση :



Σχήμα 5.6 Οι θέσεις των κόμβων του 2<sup>ου</sup> σεναρίου κατά την κίνησή τους

### 5.3.2 Αρχείο εξόδου

Μετά την εκτέλεση του κώδικα, σειρά έχει η κατάλληλη επεξεργασία του προκύπτοντος trace file. Αυτό έχει την εξής μορφή:

Event	Time	Node	Layer	Flags	Pkt_id	Pkt_type	Pkt_size	
r	27,72309	_0_	AGT	---	173	ack	60	
s	27,72309	_0_	AGT	---	176	tcp	1040	
r	27,72309	_0_	RTR	---	176	tcp	1040	
s	27,72309	_0_	RTR	---	176	tcp	1060	
s	27,72309	_0_	AGT	---	177	tcp	1040	Αποστολή πακέτου από τον κομβο_0
r	27,72309	_0_	RTR	---	177	tcp	1040	Μεταβίβαση του πακέτου στον router
s	27,72309	_0_	RTR	---	177	tcp	1060	Αποστολή πακέτου από τον router
D	27,72311	_0_	IFQ	---	177	tcp	1060	Απόρριψη πακέτου λόγω γεμάτης ουράς
r	27,73312	_1_	AGT	---	161	tcp	1060	
s	27,73312	_1_	AGT	---	178	ack	40	
r	27,73312	_1_	RTR	---	178	ack	40	
s	27,73312	_1_	RTR	---	178	ack	60	

r	27,7431	_1_	AGT	---	162	tcp	1060	
s	27,7431	_1_	AGT	---	179	ack	40	
r	27,7431	_1_	RTR	---	179	ack	40	
s	27,7431	_1_	RTR	---	179	ack	60	
r	27,7452	_0_	AGT	---	174	ack	60	
.....								
r	60,82555	_0_	AGT	---	5652	ack	60	
s	60,82555	_0_	AGT	---	5653	tcp	1040	
r	60,82555	_0_	RTR	---	5653	tcp	1040	
s	60,82555	_0_	RTR	---	5653	tcp	1060	
r	60,8355	_1_	AGT	---	5632	tcp	1060	
s	60,8355	_1_	AGT	---	5654	ack	40	
r	60,8355	_1_	RTR	---	5654	ack	40	
s	60,8355	_1_	RTR	---	5654	ack	60	
r	60,83794	_0_	AGT	---	5654	ack	60	
s	60,83794	_0_	AGT	---	5655	tcp	1040	
r	60,83794	_0_	RTR	---	5655	tcp	1040	

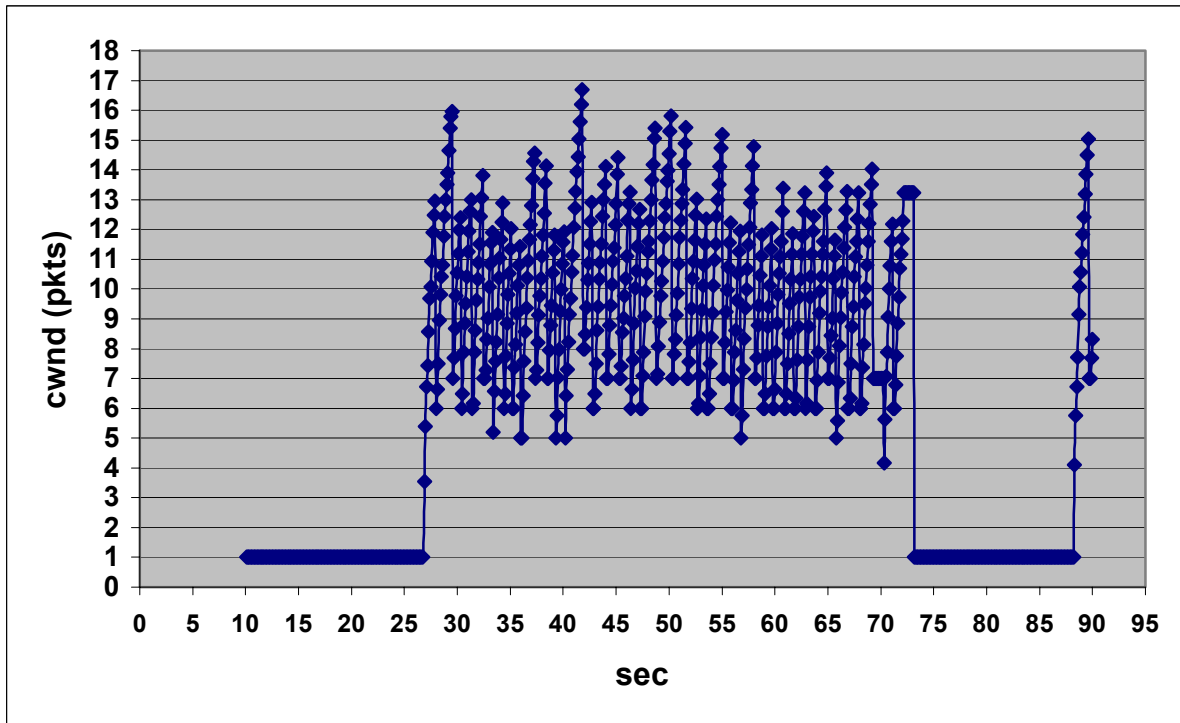
Ας μελετήσουμε την πορεία ενός πακέτου tcp που στέλνεται από τον κόμβο 0 και λαμβάνεται από τον κόμβο 1. Ένα τέτοιο πακέτο είναι πχ. το πακέτο 177 (packet id, δηλαδή ο αριθμός μετά τις τρεις παύλες). Βλέπουμε λοιπόν ότι το πακέτο αυτό στέλνεται αρχικά από τον κόμβο 0 (χρόνος 27,72309) και έχει μέγεθος 1040. Ταυτόχρονα μεταβιβάζεται στο RTR, οπότε και το μέγεθός του αυξάνεται στα 1060. Όμως την χρονική στιγμή 27,72311 το πακέτο χάνεται λόγω ότι η ουρά είναι γεμάτη (IFQ).

### 5.3.3 Επεξεργασία των αποτελεσμάτων

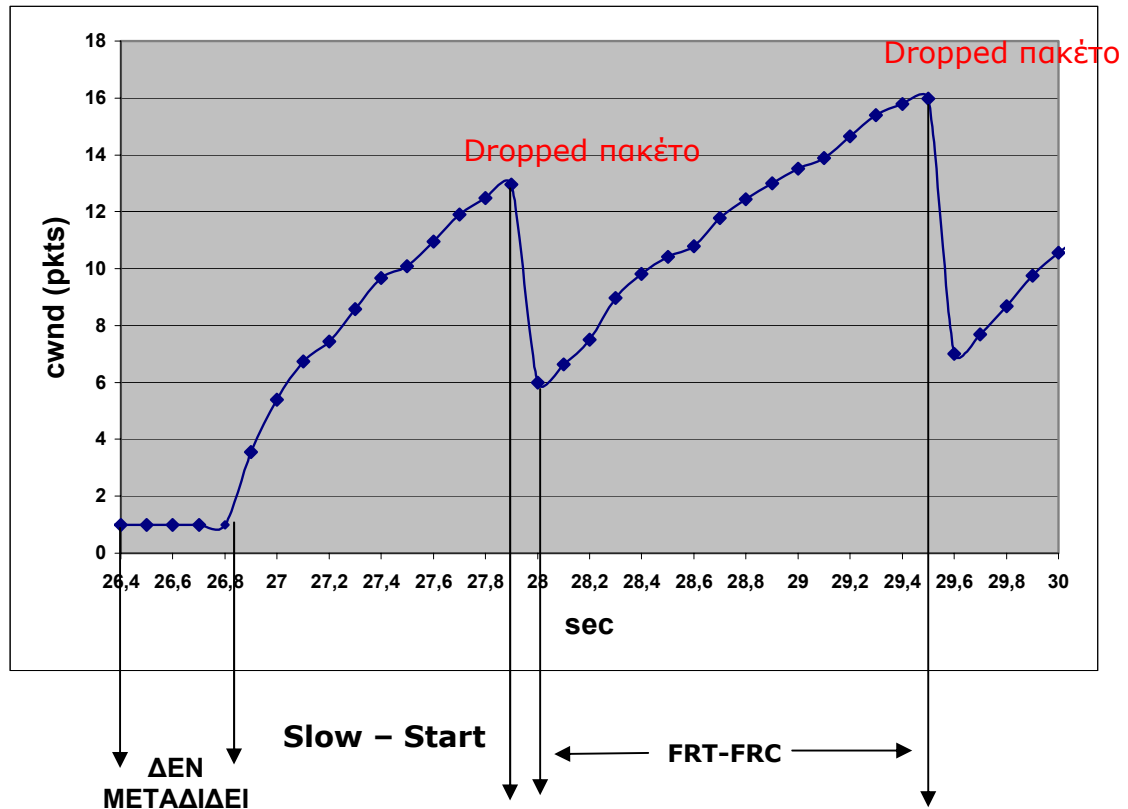
#### 5.3.3 .1 Επεξεργασία του cwnd

Μέχρι το δέκατο δευτερόλεπτο, που ξεκινάει η επικοινωνία μεταξύ των κόμβων, το cwnd είναι ίσο με ένα πακέτο. Την στιγμή που ξεκινάει η αποστολή δεδομένων και μέχρι τα 26,6 sec τα πακέτα χάνονται γιατί οι κόμβοι είναι εκτός εμβέλειας. Από εκείνη την στιγμή και μετά αρχίζει ο κόμβος 1 να λαμβάνει πακέτα οπότε και αρχίζει η διαδικασία Slow Start μέχρι την χρονική στιγμή 27,7 sec. Εκεί παρατηρούμε ότι έχει χαθεί ένα πακέτο άρα η τιμή του cwnd

κατεβαίνει στο μισό της τιμής του. Από την στιγμή 27,9 sec αρχίζει η διαδικασία fast retransmit - fast recovery.



Σχήμα 5.7 Γράφημα του cwnd (pkts) προς χρόνο (sec)





## FRT – FRC : Fast Retransmit – Fast Recovery

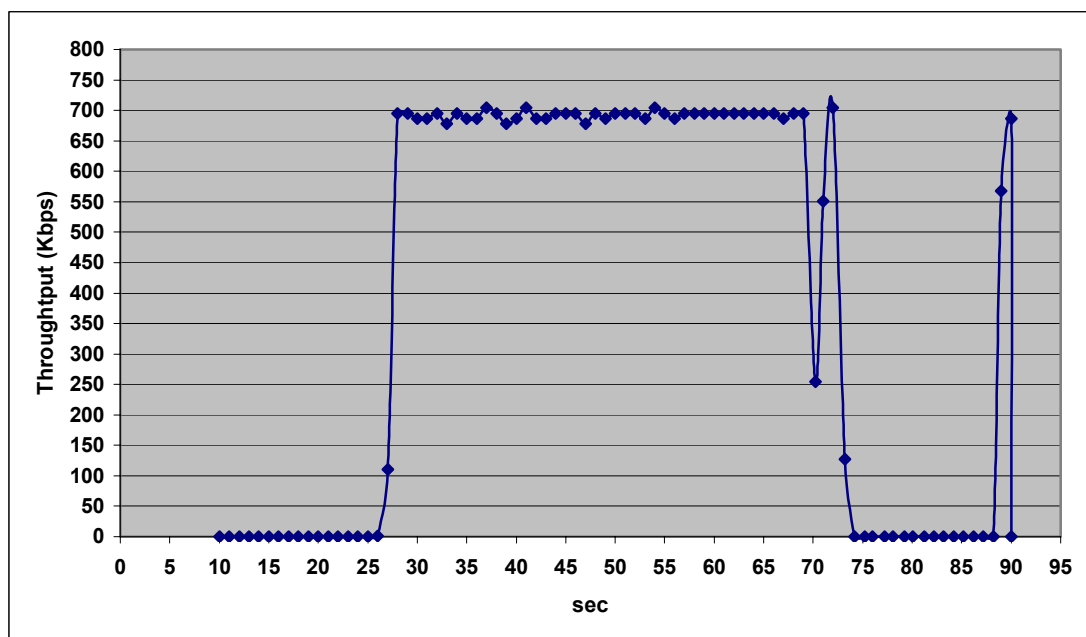
Σχήμα 5.8 Ανάλυση της αλλαγής μεταξύ Slow-Start και Fast Retransmit-Fast Recovery

### 5.3.3.2 Επεξεργασία του throughput

Για την επεξεργασία του throughput χρησιμοποιούμε τον ίδιο κώδικα φίλτρου awk με το προηγούμενο παράδειγμα στην παράγραφο 5.1.4.2, που βρίσκεται στο Παράρτημα Α (6.1).

### 5.3.3.3 Μετρήσεις

Εφαρμογή του παραπάνω φίλτρου AWK στο trace file δίνει τα εξής αποτελέσματα:



Σχήμα 5.9 Γράφημα του Throughput σε Kbps προς χρόνο σε sec

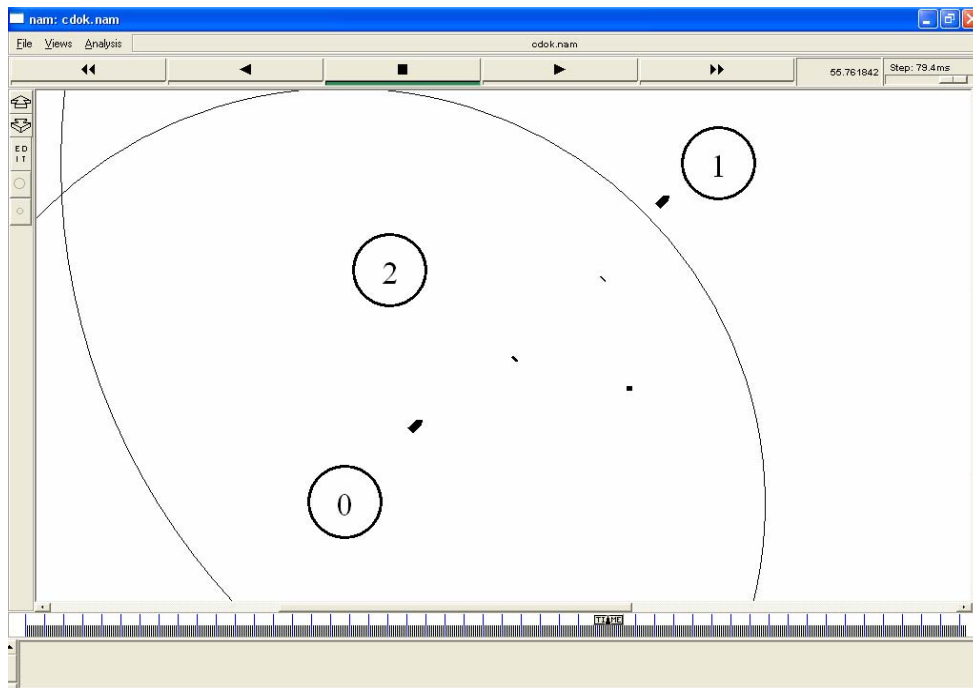
## 5.4 Τρίτο σενάριο

### 5.4.1 Γενική Περιγραφή

Το τρίτο σενάριο είναι μια επέκταση του δεύτερου σεναρίου. Αυτή τη φορά, το σενάριο, αποτελείται από τρεις ασύρματους κόμβους, εκ των οποίων οι δύο είναι κινητοί και ο ένας παραμένει σταθερός. Η επικοινωνία ξεκινάει με τον κόμβο\_0 να είναι αποστολέας και παραλήπτης ο κόμβος\_1. Στην συνέχεια η επικοινωνία «παρεμβάλλεται» από τον κόμβο\_2 ο οποίος επικοινωνεί, ταυτόχρονα με τον κόμβο\_0, με τον κόμβο\_1. Επίσης οι κόμβοι 0 και 1 κινούνται ο ένας προς τον άλλον και κάποια χρονική στιγμή απομακρύνονται. Η προσομοίωση διαρκεί για 90.0 sec. Η αποστολή δεδομένων, από τον ftp Agent του κόμβου\_0, ξεκινά στα 10.0 sec ενώ του κόμβου\_2 στα 60.0 sec και ολοκληρώνονται στα 90.0 sec. Ο κώδικας παρατίθεται στο Παράρτημα Α (6.4). Ο ορισμός του κόμβου\_2 ως αποστολέα και του κόμβου\_1 ως παραλήπτη καθώς και η δημιουργία της επικοινωνίας μεταξύ τους γίνεται με το παρακάτω κομμάτι του κώδικα :

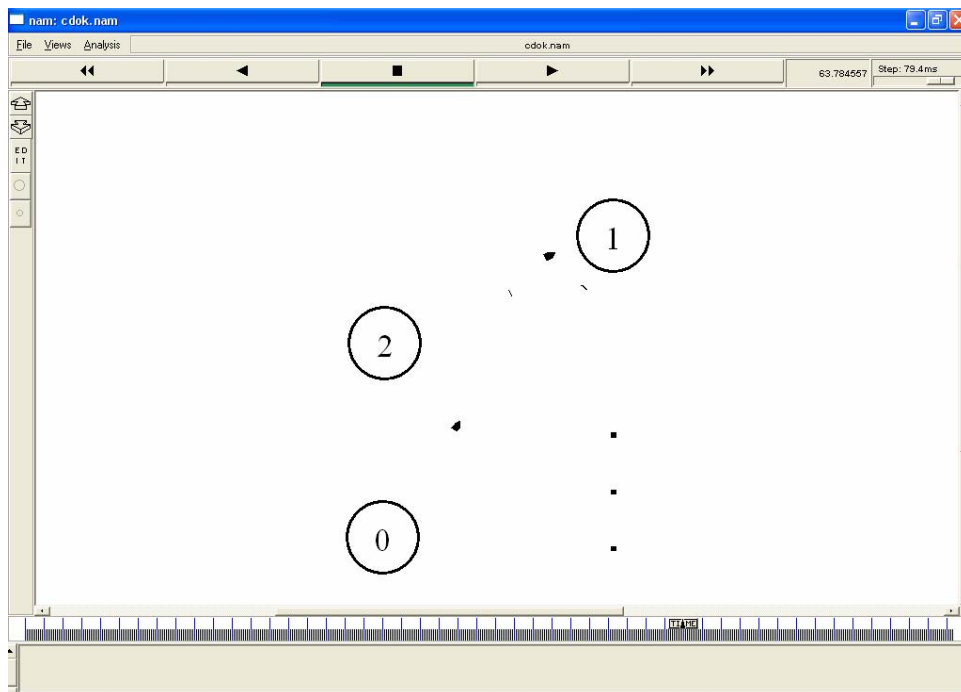
```
set tcp1 [new Agent/TCP/Newreno]
$tcp set class_ 2
$tcp set maxrto_ 1
$tcp set ssthresh_ 20
$tcp set packetSize_ 1000
$ns attach-agent $node_(2) $tcp1
set sink1 [new Agent/TCPSink]
$ns attach-agent $node_(1) $sink1
$ns connect $tcp1 $sink1
set ftp1 [new Application/FTP]
$ftp1 attach-agent $tcp1
$ns at 60.0 "$ftp1 start"
```

Ακολουθεί η εικονική αναπαράσταση του τρίτου σεναρίου, κατά την διάρκεια αποστολής δεδομένων από τον κόμβο\_0 προς τον κόμβο\_1.



Σχήμα 5.10 Αναπαράσταση αποστολής δεδομένων του κόμβου\_0

Στο παρακάτω σχήμα βλέπουμε την ταυτόχρονη επικοινωνία των κόμβων 0 και 2 με τον κόμβο\_1.



Σχήμα 5.11 Ταυτόχρονη αποστολή δεδομένων προς τον κόμβο\_1

## 5.4.2 Αρχείο εξόδου

Μετά την εκτέλεση του κώδικα, σειρά έχει η κατάλληλη επεξεργασία του προκύπτοντος trace file. Αυτό έχει την εξής μορφή:

Event	Time	Node	Layer	Flags	Pkt_id	Pkt_type	Pkt_size	
s	52,53014	_1_	RTR	---	137	ack	60	
r	52,53234	_0_	AGT	---	135	ack	60	
s	52,53234	_0_	AGT	---	138	tcp	1040	Αποστολή πακέτου από τον κόμβο_0
r	52,53234	_0_	RTR	---	138	tcp	1040	Μεταβίβαση του πακέτου στον router
s	52,53234	_0_	RTR	---	138	tcp	1060	Αποστολή πακέτου από τον router
r	52,53444	_0_	AGT	---	137	ack	60	
s	52,53444	_0_	AGT	---	139	tcp	1040	
r	52,53444	_0_	RTR	---	139	tcp	1040	
s	52,53444	_0_	RTR	---	139	tcp	1060	
s	52,53444	_0_	AGT	---	140	tcp	1040	
r	52,53444	_0_	RTR	---	140	tcp	1040	
s	52,53444	_0_	RTR	---	140	tcp	1060	
r	52,54455	_1_	AGT	---	138	tcp	1060	Λήψη του πακέτου από τον κόμβο_1
.....								
r	71,27412	_1_	RTR	---	3235	ack	40	
s	71,27412	_1_	RTR	---	3235	ack	60	
S	71,28145	_2_	AGT	---	3236	tcp	1040	Αποστολή πακέτου από τον κόμβο_0
R	71,28145	_2_	RTR	---	3236	tcp	1040	Μεταβίβαση του πακέτου στον router
S	71,28145	_2_	RTR	---	3236	tcp	1060	Αποστολή πακέτου από τον router
R	71,28424	_1_	AGT	---	3234	tcp	1060	
S	71,28424	_1_	AGT	---	3237	ack	40	
R	71,28424	_1_	RTR	---	3237	ack	40	
S	71,28424	_1_	RTR	---	3237	ack	60	
R	71,28636	_0_	AGT	---	3228	ack	60	
S	71,28636	_0_	AGT	---	3238	tcp	1040	
R	71,28636	_0_	RTR	---	3238	tcp	1040	
S	71,28636	_0_	RTR	---	3238	tcp	1060	
s	71,28636	_0_	AGT	---	3239	tcp	1040	
.....								
r	71,29654	_1_	AGT	---	3238	tcp	1060	
s	71,29654	_1_	AGT	---	3241	ack	40	
r	71,29654	_1_	RTR	---	3241	ack	40	
s	71,29654	_1_	RTR	---	3241	ack	60	
r	71,29853	_0_	AGT	---	3231	ack	60	
s	71,29853	_0_	AGT	---	3242	tcp	1040	
r	71,29853	_0_	RTR	---	3242	tcp	1040	

s	71,29853	_0_	RTR	---	3242	tcp	1060	
r	71,30053	_0_	AGT	---	3232	ack	60	
s	71,30053	_0_	AGT	---	3243	tcp	1040	
r	71,30053	_0_	RTR	---	3243	tcp	1040	
s	71,30053	_0_	RTR	---	3243	tcp	1060	
r	71,31051	_1_	AGT	---	3236	tcp	1060	Λήψη του πακέτου από τον κόμβο_1
s	71,31051	_1_	AGT	---	3244	ack	40	

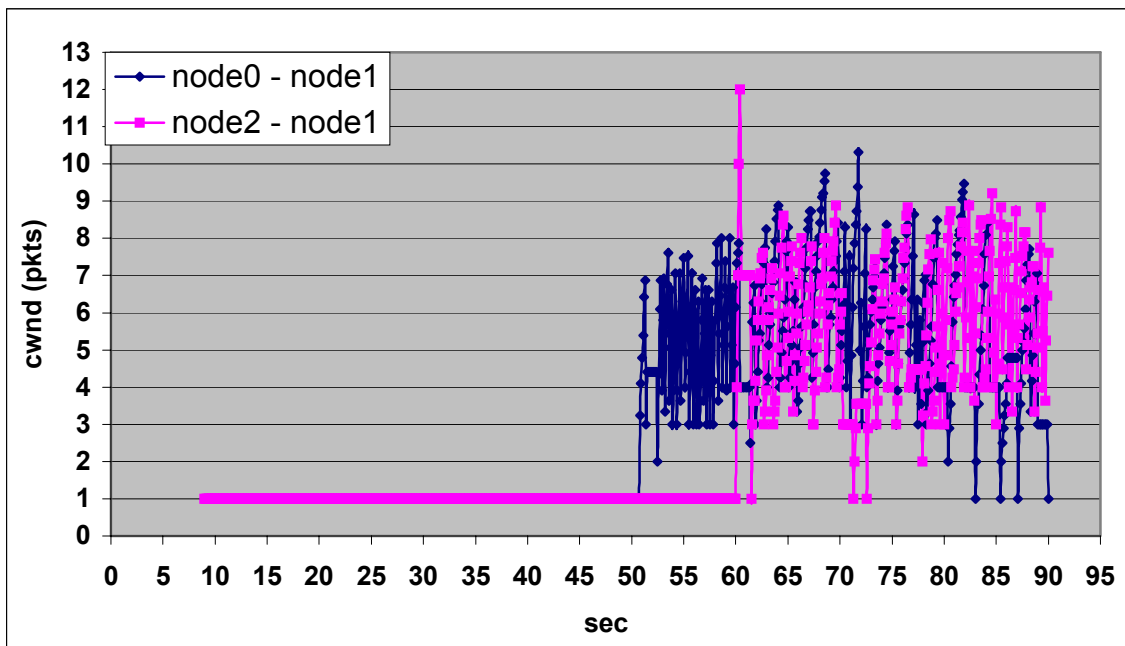
Θα αναλύσουμε την πορεία ενός πακέτου κατά την διάρκεια που επικοινωνούν μόνο ο κόμβος\_0 με τον κόμβο\_1. Η πορεία του είναι ίδια ακριβώς με την πορεία των πακέτων που μελετήσαμε στα παραπάνω παραδείγματα. Παρατηρούμε ότι έχει πολύ μικρή χρονική καθυστέρηση, μόλις 0.01 sec, από την στιγμή της αποστολής του από τον κόμβο\_0 μέχρι την λήψη του από τον κόμβο\_1. Σε αντίθεση με την πορεία του δεύτερου πακέτου, το οποίο στέλνεται από τον κόμβο\_2 στον κόμβο\_1, και παρατηρούμε ότι έχει μια χρονική καθυστέρηση της τάξης των 0.03 sec, λόγω της ταυτόχρονης επικοινωνίας του κόμβου 0 και 2 με τον κόμβο 1.

### 5.4.3 Επεξεργασία των αποτελεσμάτων

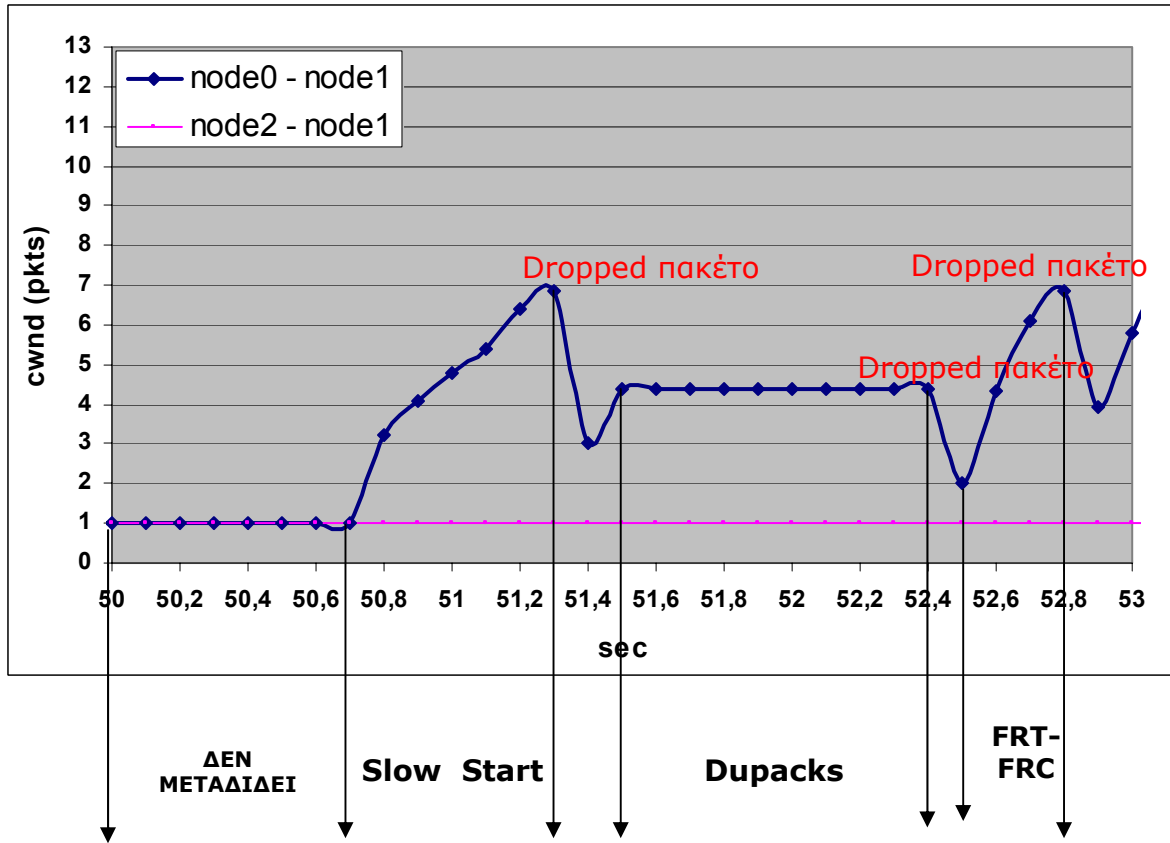
#### 5.4.3 .1 Επεξεργασία του cwnd

Κατά την διάρκεια των πρώτων δέκα δευτερολέπτων το cwnd είναι ίσο με ένα πακέτο, γιατί δεν υπάρχει επικοινωνία μεταξύ των κόμβων. Από την χρονική στιγμή που ξεκινά η αποστολή δεδομένων από τον κόμβο\_0 στον κόμβο\_1, δηλαδή τα 10.0 sec μέχρι τα 50.7 sec όμως τα πακέτα χάνονται γιατί ο ένας κόμβος δεν είναι μέσα στην εμβέλεια του άλλου. Από εκείνη την στιγμή και μετά αρχίζει ο κόμβος 1 να λαμβάνει πακέτα οπότε και αρχίζει η διαδικασία Slow Start μέχρι την χρονική στιγμή 51.3 sec. Εκεί παρατηρούμε ότι έχει χαθεί ένα πακέτο άρα η τιμή του cwnd κατεβαίνει στο μισό της τιμής

του. Από τα 51.5 sec μέχρι και τα 52.4 sec το cwnd παραμένει σταθερό λόγω των dupAcks. Εκείνη την χρονική στιγμή έχουμε όμως ένα χαμένο πακέτο οπότε αρχίζει η διαδικασία fast retransmit - fast recovery μέχρι τα 52.8 sec, όπου πάλι έχουμε χαμένο πακέτο. Όσον αφορά τον κόμβο\_2 ξεκινάει την επικοινωνία του με τον κόμβο\_1 στα 60.0 sec. Μέχρι εκείνη την στιγμή δεν υπάρχει επικοινωνία, επομένως και το cwnd είναι ίσο με ένα πακέτο. Μετά βλέπουμε μια εκθετική αύξηση του cwnd γιατί έχει μπει στην διαδικασία Slow Start μέχρι και την στιγμή 60,4 sec όπου χάνεται ένα πακέτο. Έπειτα από τα 60,5 sec μέχρι τα 61,4 sec έχουμε dupAcks επομένως και σταθερό cwnd, μέχρι που χάνεται ένα πακέτο και έχουμε την πτώση του στο μισό. Στην συνέχεια από τα 61,6 sec μέχρι και τα 62,6 μπαίνει σε διαδικασία ανάκαμψης με τους αλγόριθμους Fast Retransmit – Fast Recovery.

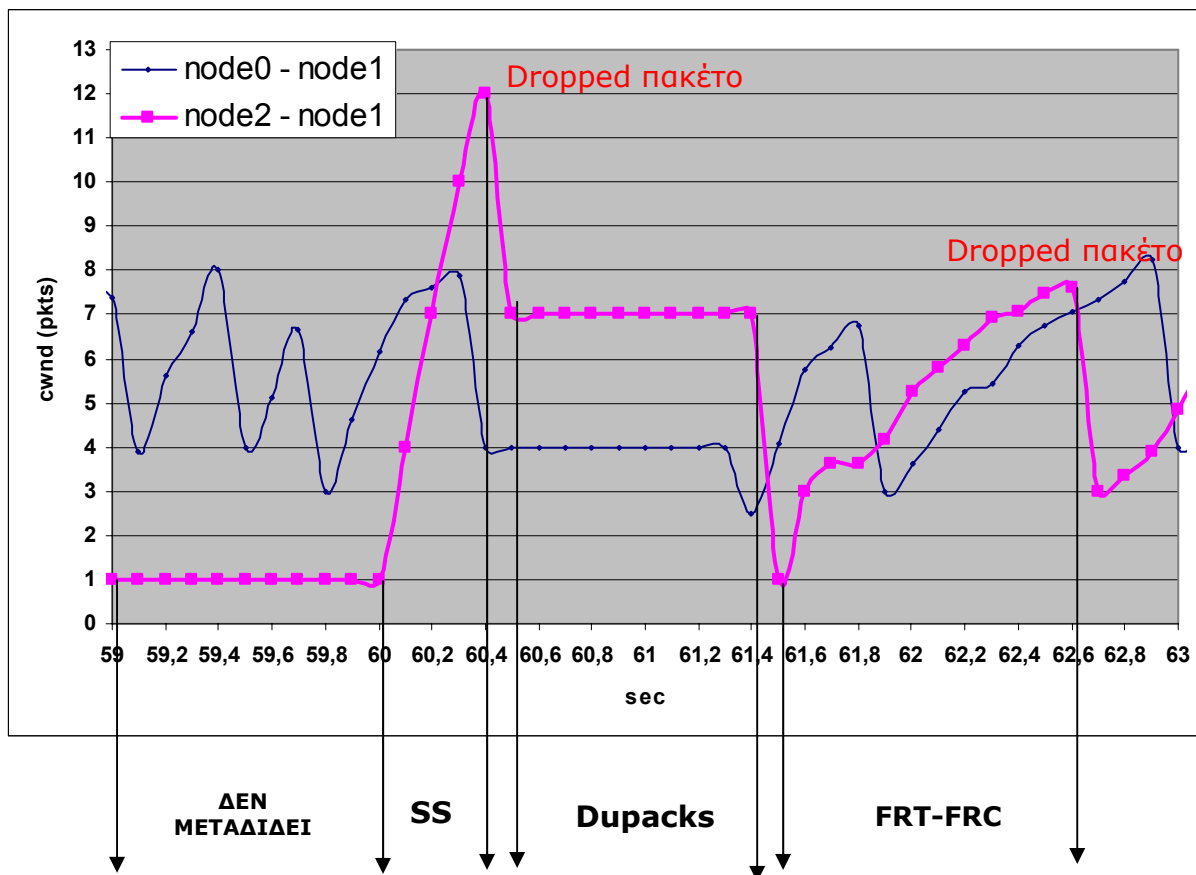


Σχήμα 5.12 Γράφημα του cwnd (pkts) προς χρόνο (sec)



**FRT – FRC : Fast Retransmit – Fast Recovery**

Σχήμα 5.13 Ανάλυση της αλλαγής μεταξύ Slow-Start και Fast Retransmit-Fast Recovery



**SS : Slow Start**

**FRT – FRC : Fast Retransmit – Fast Recovery**

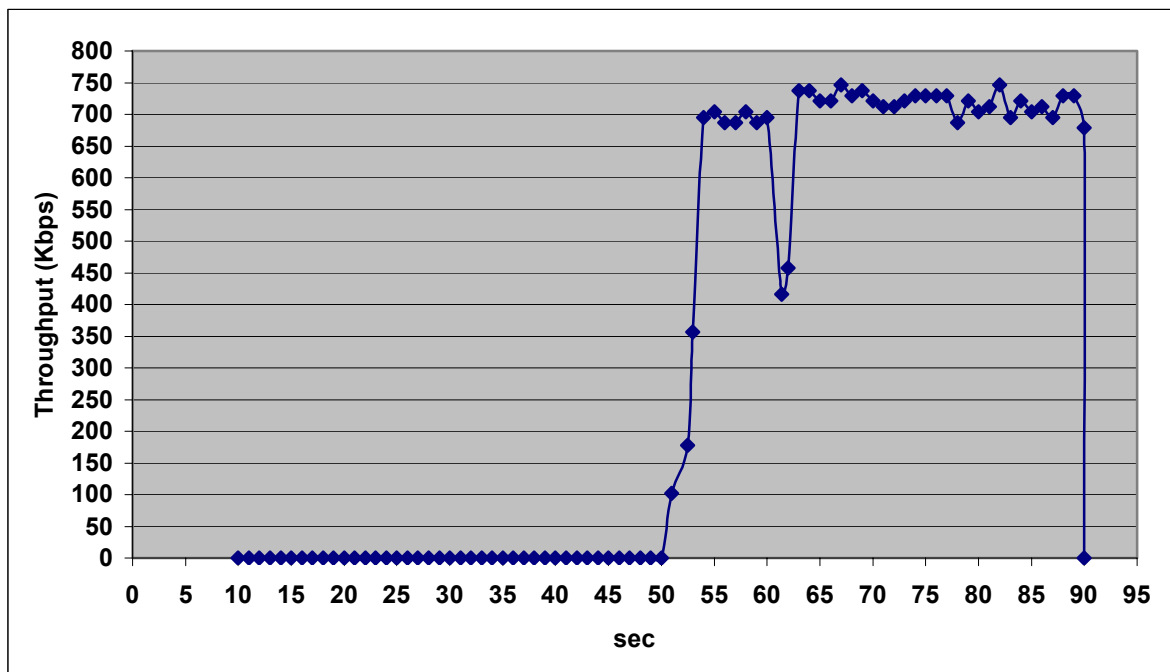
Σχήμα 5.14 Ανάλυση της εναλλαγής μεταξύ Slow-Start, Congestion Avoidance και Fast Retransmit-Fast Recovery

**5.4.3.2 Επεξεργασία του throughput**

Για την επεξεργασία του throughput χρησιμοποιούμε τον ίδιο κώδικα φίλτρου awk που βρίσκεται στο Παράρτημα Α (6.1).

**5.4.3.3 Μετρήσεις**

Εφαρμογή του παραπάνω φίλτρου AWK στο trace file δίνει τα εξής αποτελέσματα:



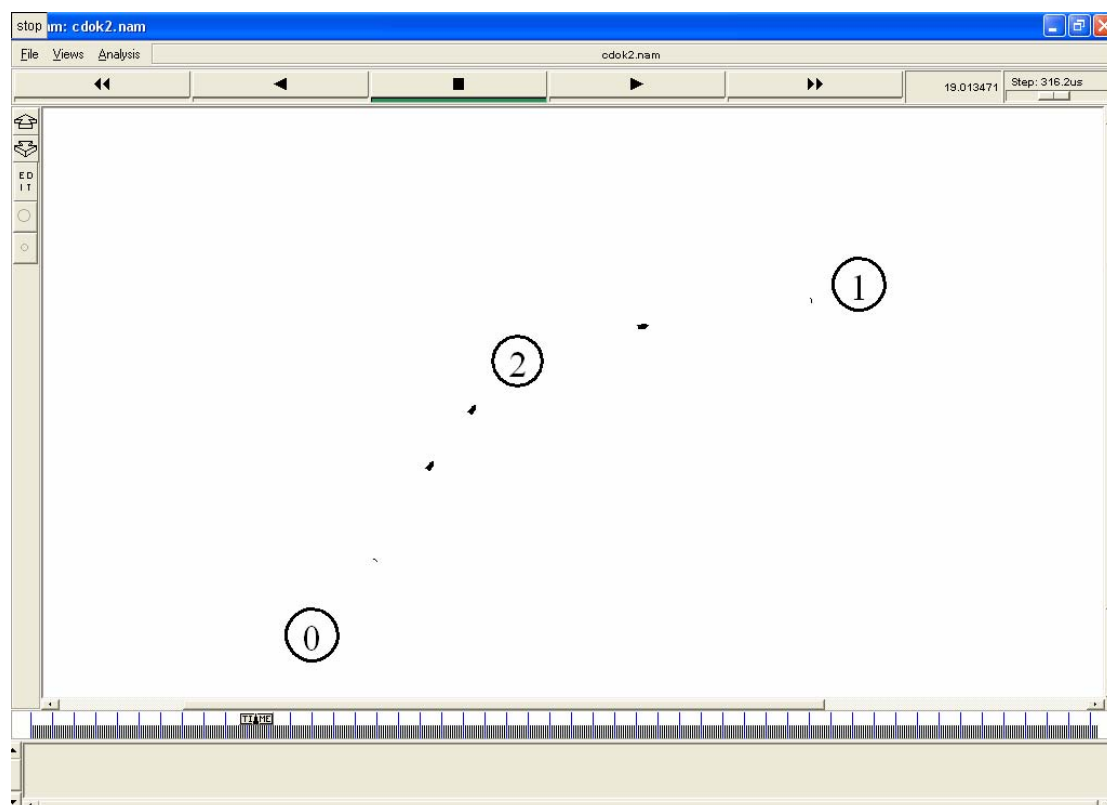
Σχήμα 5.15 Γράφημα του Throughput σε Kbps προς χρόνο σε sec



## 5.5 Τέταρτο Σενάριο

### 5.5.1 Γενική Περιγραφή

Στο τέταρτο και τελευταίο σενάριο μας έχουμε μια διαφοροποίηση της προηγούμενης προσομοίωσης. Σε αυτήν την περίπτωση έχουμε δύο ακίνητους κόμβους απομακρυσμένους που επικοινωνούν με την βοήθεια του κόμβου\_2 ο οποίος βρίσκεται ανάμεσα τους και εκτελεί χρέη δρομολογητή. Ο κόμβος\_0 στέλνει τα δεδομένα στον κόμβο\_1 αλλά επειδή δεν είναι στην εμβέλεια του τα λαμβάνει ο κόμβος\_2. Ο οποίος είναι στην εμβέλεια και των δύο κόμβων οπότε προωθεί τα δεδομένα στον κόμβο\_1. Η προσομοίωση διαρκεί για 90.0 sec.



Σχήμα 5.16 τοπολογία 4<sup>ου</sup> σεναρίου

## 5.5.2 Αρχείο εξόδου

Μετά την εκτέλεση του κώδικα, σειρά έχει η κατάλληλη επεξεργασία του προκύπτοντος trace file. Αυτό έχει την εξής μορφή:

Event	Time	Node	Layer	Flags	Pkt_id	Pkt_type	Pkt_size	
s	47,94093	_0_	AGT	---	5375	tcp	1000	Αποστολή πακέτου από τον κόμβο_0
r	47,94093	_0_	RTR	---	5375	tcp	1000	Μεταβίβαση του πακέτου στον router
s	47,94093	_0_	RTR	---	5375	tcp	1020	Αποστολή πακέτου από τον router
r	47,94586	_2_	RTR	---	5367	tcp	1020	
f	47,94586	_2_	RTR	---	5367	tcp	1020	
r	47,95085	_2_	RTR	---	5368	tcp	1020	
f	47,95085	_2_	RTR	---	5368	tcp	1020	
r	47,95194	_0_	AGT	---	5371	ack	60	
s	47,95194	_0_	AGT	---	5376	tcp	1000	
r	47,95194	_0_	RTR	---	5376	tcp	1000	
s	47,95194	_0_	RTR	---	5376	tcp	1020	
r	47,95729	_2_	RTR	---	5369	tcp	1020	
f	47,95729	_2_	RTR	---	5369	tcp	1020	
r	47,95842	_0_	AGT	---	5372	ack	60	
s	47,95842	_0_	AGT	---	5377	tcp	1000	
r	47,95842	_0_	RTR	---	5377	tcp	1000	
s	47,95842	_0_	RTR	---	5377	tcp	1020	
r	47,96341	_2_	RTR	---	5374	tcp	1020	
f	47,96341	_2_	RTR	---	5374	tcp	1020	
r	47,96462	_0_	AGT	---	5373	ack	60	
s	47,96462	_0_	AGT	---	5378	tcp	1000	
r	47,96462	_0_	RTR	---	5378	tcp	1000	
s	47,96462	_0_	RTR	---	5378	tcp	1020	
r	47,96963	_2_	RTR	---	5375	tcp	1020	Παραλαβή του πακέτου από τον κόμβο_2
f	47,96963	_2_	RTR	---	5375	tcp	1020	Πρωώθηση του πακέτου στον κόμβο_1
r	47,97468	_1_	AGT	---	5365	tcp	1020	
s	47,97468	_1_	AGT	---	5379	ack	40	
r	47,97468	_1_	RTR	---	5379	ack	40	
s	47,97468	_1_	RTR	---	5379	ack	60	
r	47,97585	_2_	RTR	---	5379	ack	60	
f	47,97585	_2_	RTR	---	5379	ack	60	
r	47,98084	_1_	AGT	---	5367	tcp	1020	
s	47,98084	_1_	AGT	---	5380	ack	40	
r	47,98084	_1_	RTR	---	5380	ack	40	
s	47,98084	_1_	RTR	---	5380	ack	60	
r	47,98585	_1_	AGT	---	5368	tcp	1020	
s	47,98585	_1_	AGT	---	5381	ack	40	

r	47,98585	_1_	RTR	---	5381	ack	40	
s	47,98585	_1_	RTR	---	5381	ack	60	
r	47,98696	_2_	RTR	---	5380	ack	60	
f	47,98696	_2_	RTR	---	5380	ack	60	
r	47,99219	_1_	AGT	---	5369	tcp	1020	
s	47,99219	_1_	AGT	---	5382	ack	40	
r	47,99219	_1_	RTR	---	5382	ack	40	
s	47,99219	_1_	RTR	---	5382	ack	60	
r	47,9971	_1_	AGT	---	5374	tcp	1020	
s	47,9971	_1_	AGT	---	5383	ack	40	
r	47,9971	_1_	RTR	---	5383	ack	40	
s	47,9971	_1_	RTR	---	5383	ack	60	
r	47,99817	_2_	RTR	---	5381	ack	60	
f	47,99817	_2_	RTR	---	5381	ack	60	
r	48,00337	_2_	RTR	---	5376	tcp	1020	
f	48,00337	_2_	RTR	---	5376	tcp	1020	
r	48,00832	_2_	RTR	---	5377	tcp	1020	
f	48,00832	_2_	RTR	---	5377	tcp	1020	
r	48,01335	_1_	AGT	---	5375	tcp	1020	
s	48,01335	_1_	AGT	---	5384	ack	40	

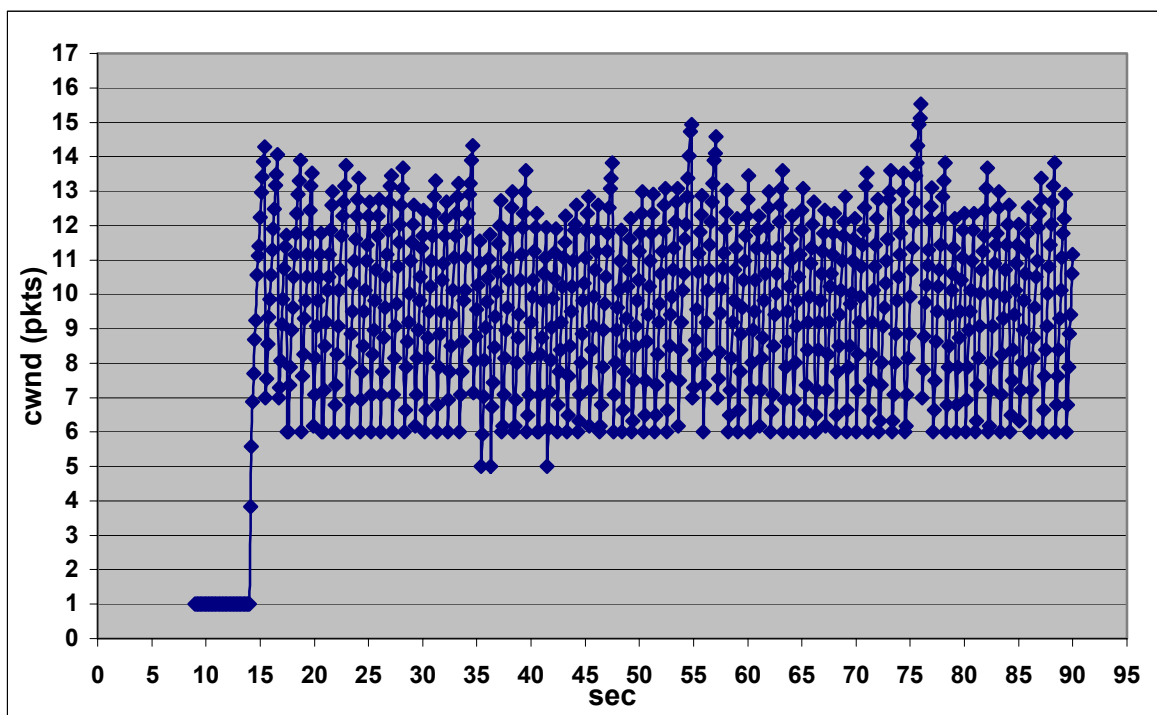
Με τη βοήθεια του παραπάνω πίνακα, μπορούμε να δούμε την πορεία που ακολουθεί ένα πακέτο, συγκεκριμένα το 5375, του τέταρτου σεναρίου. Κατά την χρονική στιγμή 47,94093 αποστέλλεται από τον κόμβο\_0, μεταβιβάζεται στο επίπεδο του router ο οποίος με τη σειρά του το στέλνει στον κόμβο\_2. Στον κόμβο\_2 παραλαμβάνεται από το επίπεδο του router και αφού ελέγξει ότι δεν είναι για κοινών το προωθεί αμέσως το πακέτο στον κόμβο\_1. Τελικά παραλαμβάνεται από τον κόμβο\_1 την χρονική στιγμή 48,01335, δηλαδή 0,08 δευτερόλεπτα αργότερα από την αποστολή του.

### 5.5.3 Επεξεργασία των αποτελεσμάτων

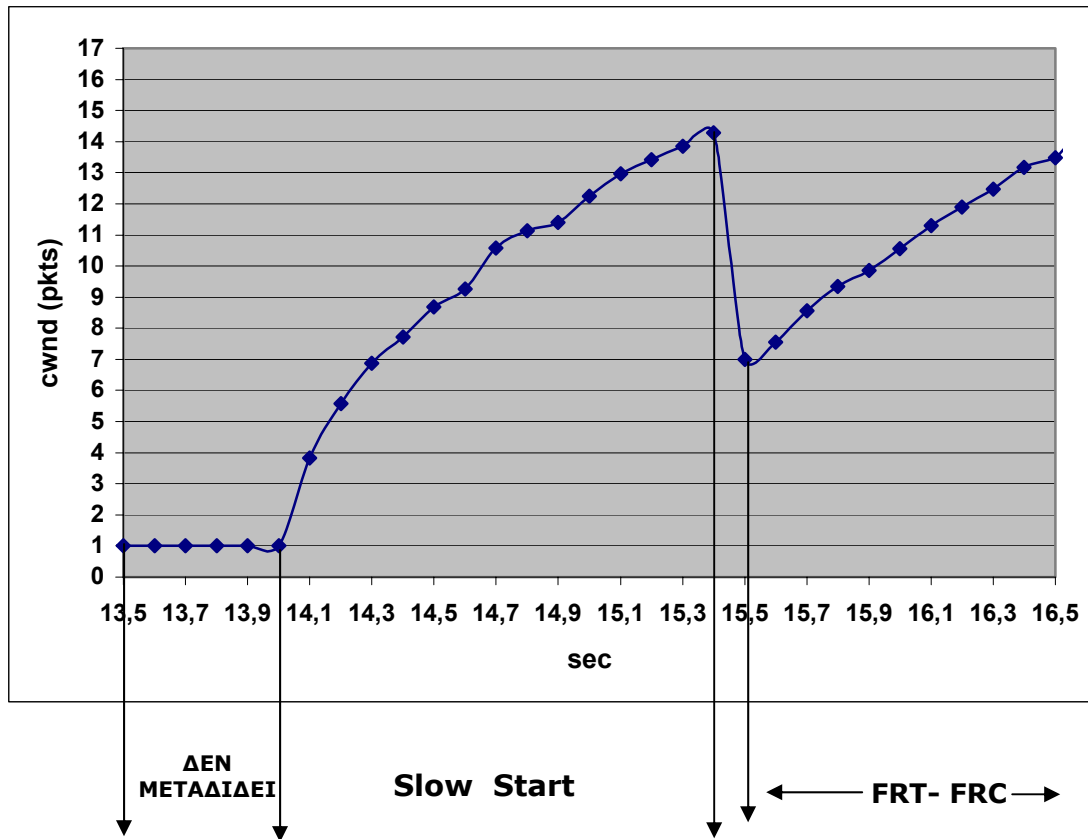
#### 5.5.3.1 Επεξεργασία του cwnd

Η αποστολή δεδομένων του κόμβου\_0, ξεκινά στα 10.0 sec. Πρώτα τα παραλαμβάνει ο κόμβος\_2 και στην συνέχεια τα στέλνει στον κόμβο\_1. Λογικό είναι να παρατηρείται μια μεγαλύτερη χρονική

καθυστέρηση στην παραλαβή των δεδομένων από τον κόμβο\_1 σε σχέση με τα προηγούμενα σενάρια. Από την χρονική στιγμή 13,9 sec έως και τα 15,4 sec έχουμε τη διαδικασία Slow Start. Στα 15,4 sec χάνετε ένα πακέτο οπότε και το cwnd πέφτει στο μισό. Και από τα 15,5 sec και μετά ξεκινάει η διαδικασία Fast Retransmit - Fast Recovery. Ο κώδικας παρατίθεται στο Παράρτημα Α (6.6). Στο σχήμα που ακολουθεί βλέπουμε την επικοινωνία του κόμβου\_0 με τον κόμβο\_1 μέσω του κόμβου\_2.



Σχήμα 5.17 Γράφημα του cwnd (pkts) προς χρόνο (sec)



**FRT – FRC : Fast Retransmit – Fast Recovery**

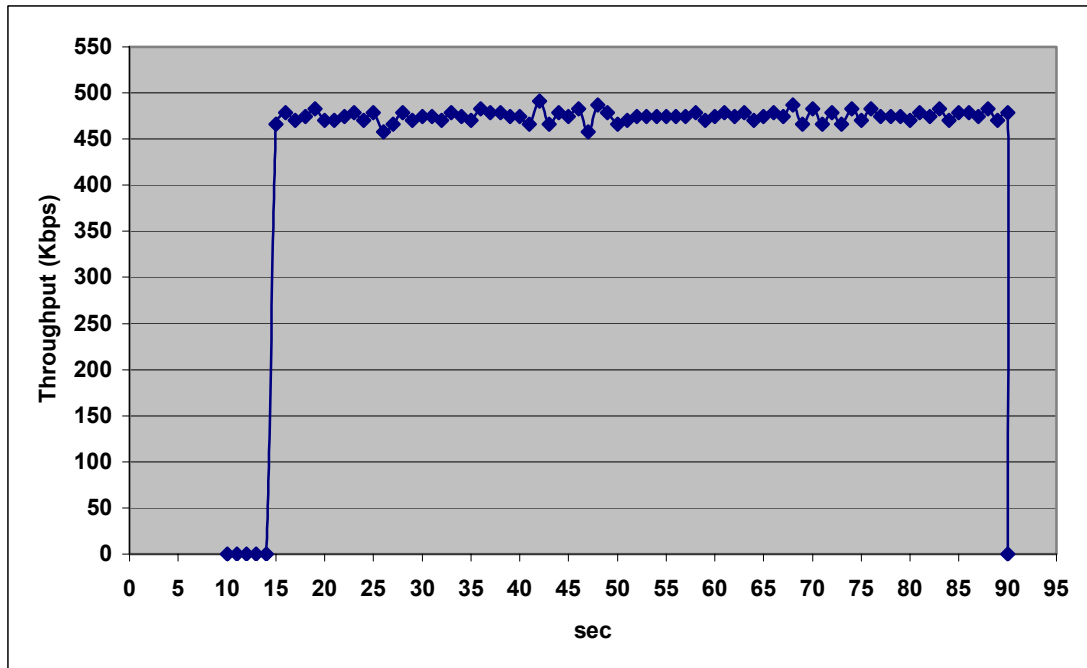
Σχήμα 5.19 Ανάλυση της εναλλαγής μεταξύ Slow-Start και Fast Retransmit-Fast Recovery

**5.5.3.2 Επεξεργασία του throughput**

Για την επεξεργασία του throughput χρησιμοποιούμε τον ίδιο κώδικα awk με το προηγούμενο παράδειγμα στην παράγραφο 5.1.4.2.

**5.5.3.3 Μετρήσεις**

Εφαρμογή του παραπάνω φίλτρου AWK στο trace file δίνει τα εξής αποτελέσματα:



Σχήμα 5.20 Γράφημα του Throughput σε Kbps προς χρόνο σε sec

## 5.6 Παρατηρήσεις

Στο πρώτο σενάριο έχουμε την ιδανική περίπτωση. Οι κόμβοι είναι σταθεροί, η αποστολή διαρκεί συνολικά για ογδόντα δευτερόλεπτα και γίνεται χωρίς παρεμβολές. Για αυτόν τον λόγο παρατηρούμε το cwnd να αυξάνεται εκθετικά μέχρι το κατώφλι του slow start (ssthreshold) και μετά συνεχίζει να αυξάνεται γραμμικά ( $1/cwnd$ ). Τις ιδανικές συνθήκες του σεναρίου τις αποδεικνύει και η ρυθμοροή του (throughput), στο σχήμα 5.4, στο οποίο τα αποτελέσματα είναι πολύ ικανοποιητικά, αφού είναι πολύ κοντά στην αναμενόμενη θεωρητική τιμή (1Mbps). Γενικά λοιπόν ο NS πετυχαίνει στο σενάριο αυτό επαρκέστατη εξομοίωση.

Στο δεύτερο σενάριο προσθέτουμε την κίνηση στους κόμβους. Η αποστολή δεδομένων, από τα συνολικά ογδόντα δευτερόλεπτα, διαρκεί για πενήντα δευτερόλεπτα αφού οι κόμβοι σε κάποια χρονική

στιγμή τίθονται εκτός εμβέλειας. Από το γράφημα 5.7 του cwnd διαπιστώνουμε ότι υπάρχουν πολλά χαμένα πακέτα. Αυτός είναι ένας σημαντικός παράγοντας για την μείωση της ρυθμοροής. Παρατηρούμε και στο σχήμα 5.9 ότι στην καλύτερη περίπτωση της φτάνει περίπου στα 700 Kbps ενώ κατά την διάρκεια έχουμε πολλές σημαντικές αυξομειώσεις της.

Στο τρίτο σενάριο, εκτός από την κίνηση, προσθέσαμε και έναν τρίτο κόμβο, ο οποίος εκτός από το να είναι ένας αποστολέας, δημιουργεί και παρεμβολές μεταξύ της ζεύξης των κόμβων 0 και 1. Κατά την διάρκεια που επικοινωνούν ταυτόχρονα οι κόμβοι 0 και 2 με τον 1 έχουμε ακόμη περισσότερα χαμένα πακέτα, δηλαδή ακόμη περισσότερες αυξομειώσεις του cwnd. Όσο για την ρυθμοροή παρατηρούμε στο σχήμα 5.15 ότι μετά βίας φτάνει στα 700 Kbps, και τα ξεπερνάει λίγο όταν αρχίσει να στέλνει ο άλλος κόμβος, και σε κάποια χρονική στιγμή μάλιστα πέφτει στα 250 Kbps. Τιμή η οποία είναι πολύ χαμηλή αφού οι δύο κόμβοι διεκδικούν την ίδια χρονική σχισμή.

Στο τέταρτο σενάριο, ο τρίτος κόμβος που προσθέσαμε έχει πάρει τον ρόλο του δρομολογητή. Δεν κάνει τίποτα παραπάνω από το να παραλαμβάνει τα πακέτα από τον κόμβο\_0 και να τα προωθεί στον κόμβο\_1. Δημιουργεί έτσι, εκτός από τα χαμένα πακέτα, όπως βλέπουμε στο σχήμα του cwnd 5.17 και μεγάλη καθυστέρηση. Κατά συνέπεια έχουμε πολύ χαμηλή ρυθμοροή, όπως παρατηρούμε στο σχήμα 5.20, η οποία δεν φτάνει τα 500 Kbps!

Μετά την μελέτη που κάναμε διαπιστώνουμε ότι το TCP, συγκεκριμένα New Reno, δεν καλύπτει ικανοποιητικά τις ανάγκες των ad-hoc δικτύων. Γιατί, δεν μπορεί να αντιληφθεί τον ανθρωπογενή θόρυβο και ότι η απώλεια ή ακόμα και η αλλοίωση ενός μόνο πακέτου δεν προκαλούν συμφόρηση στο δίκτυο. Και για

αυτούς τους λόγους δεν είναι απαραίτητο το TCP να μειώνει την τιμή του *παραθύρου (window)* και συνεπώς την ρυθμοροή. Σε αντίθεση με τα ενσύρματα δίκτυα, τα οποία έχουν ένα βασικό πλεονέκτημα απέναντι στα ασύρματα, την καλή και σωστή λειτουργία του TCP πρωτοκόλλου.



## ΚΕΦΑΛΑΙΟ 6 ΠΑΡΑΡΤΗΜΑ Α

### 6.1 Κώδικας φίλτρου AWK

```
BEGIN{
time = 10.0000
pkts = 0
st = 1.0000
}
{
if ($2 < time) {
if($1 == "r" && $4 == "AGT" && $3 == "_1_" && $7 == "tcp" )
{pkts = pkts + $8
}
}
else
{
if($1 == "r" && $4 == "AGT" && $3 == "_1_" && $7 == "tcp"){
pkts = pkts + $8
}
print $2
time=time+st
print "\t"(pkts/st)*8
pkts=0.0
}
}
END{
print $2, (pkts/st)*8
}
```

## 6.2 Κώδικας Πρώτου Σεναρίου

```

#senario 2-nodes

#=====
# Καθορισμός παραμέτρων
#=====

set val(chan) Channel/WirelessChannel ;
set val(prop) Propagation/TwoRayGround ;
set val(netif) Phy/WirelessPhy ;
set val(mac) Mac/802_11 ;
set val(ifq) Queue/DropTail ;
set val(ll) LL ;
set val(ant) Antenna/OmniAntenna ;
set val(ifqlen) 50 ;
set val(nn) 2 ;
set val(rp) DSDV;
set val(x) 500;
set val(y) 500;
set val(stop) 90;

#=====
# Κυρίως πρόγραμμα
#=====

set ns [new Simulator]

set tracefd [open adok.tr w]
set windowVsTime [open adokw.nam w]
set namtrace [open adok.nam w]
$ns trace-all $tracefd
$ns namtrace-all-wireless $namtrace $val(x) $val(y)

set topo [new Topography]

```

```
$topo load_flatgrid $val(x) $val(y)
create-god $val(nn)
```

```
# ΠΑΡΑΜΕΤΡΟΠΟΙΗΣΗ ΤΩΝ ΚΟΜΒΩΝ
$ns node-config -adhocRouting $val(rp) \
    -llType $val(ll) \
    -macType $val(mac) \
    -ifqType $val(ifq) \
    -ifqLen $val(ifqlen) \
    -antType $val(ant) \
    -propType $val(prop) \
    -phyType $val(netif) \
    -channelType $val(chan) \
    -topoInstance $topo \
    -agentTrace ON \
    -routerTrace ON \
    -macTrace OFF \
    -movementTrace OFF
```

```
# ΔΗΜΙΟΥΡΓΙΑ ΤΩΝ ΚΟΜΒΩΝ
for {set i 0} {$i < $val(nn)} {incr i} {
    set node_($i) [$ns node]
}
```

```
$node_(0) set X_ 200.0
$node_(0) set Y_ 140.0
$node_(0) set Z_ 0.0
```

```
$node_(1) set X_ 350.0
$node_(1) set Y_ 180.0
$node_(1) set Z_ 0.0
```

```
$ns set buff_ 5

#Δημιουργία ενός TCP agent
set tcp [new Agent/TCP/Newreno]
$tcp set class_ 2
set sink [new Agent/TCPSink]

# ΚΑΘΟΡΙΣΜΟΣ ΡΟΗΣ
$ns attach-agent $node_(0) $tcp
$ns attach-agent $node_(1) $sink

# ΚΑΘΟΡΙΣΜΟΣ ΠΑΡΑΜΕΤΡΩΝ
$tcp set maxrto_ 1
$tcp set ssthresh_ 10
$tcp set packetSize_ 500
$ns connect $tcp $sink
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns at 10.0 "$ftp start"

# ΜΕΤΡΗΣΗ ΤΟΥ CWND ΠΡΟΣ ΧΡΟΝΟ
proc plotWindow {tcpSource file} {
global ns
set time 0.01
set now [$ns now]
set cwnd [$tcpSource set cwnd_]
puts $file "$now $cwnd"
$ns at [expr $now+$time] "plotWindow $tcpSource $file"
}
$ns at 9.0 "plotWindow $tcp $windowVsTime"

for {set i 0} {$i < $val(nn)} {incr i} {
```

```

    $ns initial_node_pos $node_($i) 30";
}

for {set i 0} {$i < $val(nn)} {incr i} {
    $ns at 90.0 "$node_($i) reset";
}
$ns at $val(stop) "$ns nam-end-wireless $val(stop)"
$ns at 90.0 "stop"
$ns at 90.01 "puts \"NS EXITING...\" ; $ns halt"

proc stop {} {
    global ns tracefd namtrace
    $ns flush-trace
    close $tracefd
    close $namtrace
}

puts "Starting Simulation..."
$ns run

```

### 6.3 Κώδικας Δεύτερου Σεναρίου

```

#senario 2mobile-nodes
#=====
# Καθορισμός παραμέτρων
#=====

set val(chan)Channel/WirelessChannel ;
set val(prop) Propagation/TwoRayGround ;
set val(netif) Phy/WirelessPhy ;
set val(mac) Mac/802_11 ;
set val(ifq) Queue/DropTail ;
set val(ll) LL ;
set val(ant) Antenna/OmniAntenna ;
set val(ifqlen) 10 ;

```

```

set val(nn)                2    ;
set val(rp)                DSDV;
set val(x)                 500;
set val(y)                 500;
set val(stop)              90;

#=====
# Κυρίως πρόγραμμα
#=====

set ns [new Simulator]

set tracefd [open bdok.tr w]
set windowVsTime [open bdokw.nam w]
set namtrace [open bdok.nam w]
$ns trace-all $tracefd
$ns namtrace-all-wireless $namtrace $val(x) $val(y)

set topo [new Topography]
$topo load_flatgrid $val(x) $val(y)
create-god $val(nn)

# ΠΑΡΑΜΕΤΡΟΠΟΙΗΣΗ ΤΩΝ ΚΟΜΒΩΝ
$ns node-config -adhocRouting $val(rp) \
                -llType $val(ll) \
                -macType $val(mac) \
                -ifqType $val(ifq) \
                -ifqLen $val(ifqlen) \
                -antType $val(ant) \
                -propType $val(prop) \
                -phyType $val(netif) \
                -channelType $val(chan) \
                -topoInstance $topo \
                -agentTrace ON \
                -routerTrace ON \
                -macTrace OFF \
                -movementTrace OFF

# ΔΗΜΙΟΥΡΓΙΑ ΤΩΝ ΚΟΜΒΩΝ
for {set i 0} {$i < $val(nn)} {incr i} {
    set node_($i) [$ns node]
}

$node_(0) set X_ 5.0
$node_(0) set Y_ 5.0

```

```

$node_(0) set Z_ 0.0

$node_(1) set X_ 350.0
$node_(1) set Y_ 180.0
$node_(1) set Z_ 0.0

$ns at 50.0 "$node_(1) setdest 25.0 20.0 15.0"
$ns at 10.0 "$node_(0) setdest 170.0 35.0 25.0"
$ns at 100.0 "$node_(1) setdest 490.0 480.0 15.0"
$ns at 85.0 "$node_(0) setdest 390.0 200.0 20.0"
$ns set buff_ 5

#Δημιουργία ενός TCP agent
set tcp [new Agent/TCP/Newreno]
$tcp set class_ 2

# ΚΑΘΟΡΙΣΜΟΣ ΡΟΗΣ
set sink [new Agent/TCPSink]
$ns attach-agent $node_(0) $tcp
$ns attach-agent $node_(1) $sink
$ns connect $tcp $sink

# ΚΑΘΟΡΙΣΜΟΣ ΠΑΡΑΜΕΤΡΩΝ
$tcp set maxrto_ 1
$tcp set ssthreshold_ 20
$tcp set packetSize_ 1000
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns at 10.0 "$ftp start"

# ΜΕΤΡΗΣΗ ΤΟΥ CWND ΠΡΟΣ ΧΡΟΝΟ
proc plotWindow {tcpSource file} {
  global ns
  set time 0.1
  set now [$ns now]
  set cwnd [$tcpSource set cwnd_]
  puts $file "$now $cwnd"
  $ns at [expr $now+$time] "plotWindow $tcpSource $file"
}

$ns at 10.1 "plotWindow $tcp $windowVsTime"

for {set i 0} {$i < $val(nn)} {incr i} {
  $ns initial_node_pos $node_($i) 30";
}

```

```

for {set i 0} {$i < $val(nn)} {incr i} {
    $ns at 90.0 "$node_($i) reset";
}
$ns at $val(stop) "$ns nam-end-wireless $val(stop)"
$ns at 90.0 "stop"
$ns at 90.01 "puts \"NS EXITING...\" ; $ns halt"

proc stop {} {
    global ns tracefd namtrace
    $ns flush-trace
    close $tracefd
    close $namtrace
}

puts "Starting Simulation..."
$ns run

```

## 6.4 Κώδικας Τρίτου Σεναρίου

```

#senario 2mboile nodes - 1 node
#=====
# Καθορισμός παραμέτρων
#=====

set val(chan) Channel/WirelessChannel ;
set val(prop) Propagation/TwoRayGround ;
set val(netif) Phy/WirelessPhy ;
set val(mac) Mac/802_11 ;
set val(ifq) Queue/DropTail ;
set val(ll) LL ;
set val(ant) Antenna/OmniAntenna ;
set val(ifqlen) 5 ;
set val(rp) DSDV ;
set val(nn) 3 ;
set val(x) 500;
set val(y) 500;
set val(stop) 90;

#=====
# Κυρίως πρόγραμμα
#=====

```



```

set ns [new Simulator]
set tracefd [open cdok.tr w]
set windowVsTime [open cdokw.nam w]
set windowVsTime1 [open cdokw1.nam w]
set namtrace [open cdok.nam w]
$ns trace-all $tracefd
$ns namtrace-all-wireless $namtrace $val(x) $val(y)

```

```

set topo [new Topography]
$topo load_flatgrid $val(x) $val(y)
create-god $val(nn)

```

```
# ΠΑΡΑΜΕΤΡΟΠΟΙΗΣΗ ΤΩΝ ΚΟΜΒΩΝ
```

```

$ns node-config -adhocRouting $val(rp) \
                -llType $val(ll) \
                -macType $val(mac) \
                -ifqType $val(ifq) \
                -ifqLen $val(ifqlen) \
                -antType $val(ant) \
                -propType $val(prop) \
                -phyType $val(netif) \
                -channelType $val(chan) \
                -topoInstance $topo \
                -agentTrace ON \
                -routerTrace ON \
                -macTrace OFF \
                -movementTrace OFF

```

```
# ΔΗΜΙΟΥΡΓΙΑ ΤΩΝ ΚΟΜΒΩΝ
```

```

for {set i 0} {$i < $val(nn)} {incr i} {
    set node_($i) [$ns node]
}

```

```

$node_(0) set X_ 30.0
$node_(0) set Y_ 50.0
$node_(0) set Z_ 0.0

```

```

$node_(1) set X_ 490.0
$node_(1) set Y_ 285.0
$node_(1) set Z_ 0.0

```

```

$node_(2) set X_ 150.0
$node_(2) set Y_ 240.0
$node_(2) set Z_ 0.0

```

```

$ns at 10.0 "$node_(0) setdest 250.0 250.0 3.0"

```

```
$ns at 15.0 "$node_(1) setdest 45.0 285.0 5.0"
$ns at 70.0 "$node_(0) setdest 480.0 300.0 5.0"
$ns set buff_ 5
```

```
# Δημιουργία ενός TCP agent
set tcp [new Agent/TCP/Newreno]
$tcp set class_ 2
```

```
# ΚΑΘΟΡΙΣΜΟΣ ΠΑΡΑΜΕΤΡΩΝ
$tcp set maxrto_ 1
$tcp set ssthresh_ 20
$tcp set packetSize_ 1000
$ns attach-agent $node_(0) $tcp
```

```
# ΚΑΘΟΡΙΣΜΟΣ ΡΟΗΣ
set sink [new Agent/TCPSink]
$ns attach-agent $node_(1) $sink
$ns connect $tcp $sink
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns at 10.0 "$ftp start"
```

```
set tcp1 [new Agent/TCP/Newreno]
$tcp set class_ 2
$tcp set maxrto_ 1
$tcp set ssthresh_ 20
$tcp set packetSize_ 1000
$ns attach-agent $node_(2) $tcp1
set sink1 [new Agent/TCPSink]
$ns attach-agent $node_(1) $sink1
$ns connect $tcp1 $sink1
set ftp1 [new Application/FTP]
$ftp1 attach-agent $tcp1
$ns at 60.0 "$ftp1 start"
```

```
# ΜΕΤΡΗΣΗ ΤΟΥ CWND ΠΡΟΣ ΧΡΟΝΟ
proc plotWindow {tcpSource file} {
  global ns
  set time 0.1
  set now [$ns now]
  set cwnd [$tcpSource set cwnd_]
  puts $file "$now $cwnd"
  $ns at [expr $now+$time] "plotWindow $tcpSource $file"
}
```

```
$ns at 9.0 "plotWindow $tcp $windowVsTime"
```

```
proc plotWindow {tcpSource1 file1} {
  global ns
  set time 0.1
  set now [$ns now]
  set cwnd [$tcpSource1 set cwnd_]
  puts $file1 "$now $cwnd"
  $ns at [expr $now+$time] "plotWindow $tcpSource1 $file1"
}

$ns at 9.0 "plotWindow $tcp1 $windowVsTime1"

for {set i 0} {$i < $val(nn)} {incr i} {
  $ns initial_node_pos $node_($i) 30";
}

for {set i 0} {$i < $val(nn)} {incr i} {
  $ns at 90.0 "$node_($i) reset";
}
$ns at $val(stop) "$ns nam-end-wireless $val(stop)"
$ns at 90.0 "stop"
$ns at 90.01 "puts \"NS EXITING...\" ; $ns halt"

proc stop {} {
  global ns tracefd namtrace
  $ns flush-trace
  close $tracefd
  close $namtrace
}

puts "Starting Simulation..."
$ns run
```

## 6.5 Κώδικας τέταρτου σεναρίου

```

#senario 2nodes - 1 node_router
#=====
# Καθορισμός παραμέτρων
#=====

set val(chan) Channel/WirelessChannel ;
set val(prop) Propagation/TwoRayGround ;
set val(netif) Phy/WirelessPhy ;
set val(mac) Mac/802_11 ;
set val(ifq) Queue/DropTail ;
set val(ll) LL ;
set val(ant) Antenna/OmniAntenna ;
set val(ifqlen) 10 ;
set val(nn) 3 ;
set val(rp) DSDV;
set val(x) 500;
set val(y) 500;
set val(stop) 90;

#=====
# Κυρίως πρόγραμμα
#=====

set ns [new Simulator]
set tracefd [open cdok2.tr w]
set windowVsTime [open cdokw2.nam w]
set namtrace [open cdok2.nam w]
$ns trace-all $tracefd
$ns namtrace-all-wireless $namtrace $val(x) $val(y)

set topo [new Topography]
$topo load_flatgrid $val(x) $val(y)
create-god $val(nn)

# ΠΑΡΑΜΕΤΡΟΠΟΙΗΣΗ ΤΩΝ ΚΟΜΒΩΝ
$ns node-config -adhocRouting $val(rp) \
                -llType $val(ll) \
                -macType $val(mac) \
                -ifqType $val(ifq) \
                -ifqLen $val(ifqlen) \
                -antType $val(ant) \
                -propType $val(prop) \

```

```

-phyType $val(netif) \
-channelType $val(chan) \
-topoInstance $topo \
-agentTrace ON\
-routerTrace ON\
-macTrace OFF \
-movementTrace OFF

```

```

# ΔΗΜΙΟΥΡΓΙΑ ΤΩΝ ΚΟΜΒΩΝ
for {set i 0} {$i < $val(nn)} {incr i} {
    set node_($i) [$ns node]
}

```

```

$node_(0) set X_ 30.0
$node_(0) set Y_ 80.0
$node_(0) set Z_ 0.0

```

```

$node_(1) set X_ 350.0
$node_(1) set Y_ 285.0
$node_(1) set Z_ 0.0

```

```

$node_(2) set X_ 150.0
$node_(2) set Y_ 240.0
$node_(2) set Z_ 0.0

```

```

#Δημιουργία ενός TCP agent
set tcp [new Agent/TCP/Newreno]
$tcp set class_ 2

```

```

# ΚΑΘΟΡΙΣΜΟΣ ΠΑΡΑΜΕΤΡΩΝ
$tcp set maxrto_ 1
$tcp set ssthresh_ 20
$tcp set packetSize_ 1000
$ns attach-agent $node_(0) $tcp

```

```

# ΚΑΘΟΡΙΣΜΟΣ ΡΟΗΣ
set sink [new Agent/TCPSink]
$ns attach-agent $node_(1) $sink
$ns connect $tcp $sink
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns at 10.0 "$ftp start"

```

```

# ΜΕΤΡΗΣΗ ΤΟΥ CWND ΠΡΟΣ ΧΡΟΝΟ
proc plotWindow {tcpSource file} {
  global ns
  set time 0.1
  set now [$ns now]
  set cwnd [$tcpSource set cwnd_]
  puts $file "$now $cwnd"
  $ns at [expr $now+$time] "plotWindow $tcpSource $file"
}

$ns at 9.0 "plotWindow $tcp $windowVsTime"

for {set i 0} {$i < $val(nn)} {incr i} {
  $ns initial_node_pos $node_($i) 30";
}

for {set i 0} {$i < $val(nn)} {incr i} {
  $ns at 90.0 "$node_($i) reset";
}
$ns at $val(stop) "$ns nam-end-wireless $val(stop)"
$ns at 90.0 "stop"
$ns at 90.01 "puts \"NS EXITING...\" ; $ns halt"

proc stop {} {
  global ns tracefd namtrace
  $ns flush-trace
  close $tracefd
  close $namtrace
}

puts "Starting Simulation..."
$ns run

```

## Βιβλιογραφία

[1]: [www.papaki.panteion.gr/teuxos18/diktua.htm](http://www.papaki.panteion.gr/teuxos18/diktua.htm)

[2]: Jean Walrand, "Δίκτυα Επικοινωνιών", 1997

[3]: Douglas E. Comer, "Internetworking with TCP/IP Principles, protocols, and architectures"

[4]: W. Richard Stevens "TCP/IP Illustrated Volume 1, The protocols" 2002

[5]: Douglas E. Comer, "Δίκτυα και Διαδίκτυα υπολογιστών, και οι εφαρμογές τους στο Internet"

[6]: <http://en.wikipedia.org/wiki/TCP> Westwood

<http://www.ntua.gr/>

<http://www.isi.edu/nsnam/ns/>

Tanenbaum, Andrew S., «Computer Networks, 4<sup>th</sup> edition», Prentice Hall, 2003