



**ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ
ΙΔΡΥΜΑ ΗΠΕΙΡΟΥ**

EPIRUS INSTITUTE OF TECHNOLOGY
ARTA , ZIP CODE : 47100 , TEL NUMBER : +30 26810 50000

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Εφαρμογές Ανοιχτού Κώδικα για Μοντελοποίηση και
Μέτρηση Μεγεθών Απόδοσης
Τηλεπικοινωνιακών Δικτύων



Τερζόγλου Αποστολία

A.M.: 8294

Κατερέλος Άγγελος

A.M.:8317

Επιβλέπων καθηγητής

Στεργίου Ελευθέριος

Άρτα 2013

ΕΥΧΑΡΙΣΤΙΕΣ

Θα θέλαμε να ευχαριστήσουμε τους ακόλουθους:

Τον επιβλέποντα καθηγητή. Κ. Στεργίου Ελευθέριο, για τη βοήθεια και την υποστήριξη που μας πρόσφερε.

Τις οικογένειες μας για την υπομονή και τη διαρκή υποστήριξη που μας έδειξαν όλα αυτά τα χρόνια.

ΠΕΡΙΕΧΟΜΕΝΑ

ΕΥΧΑΡΙΣΤΙΕΣ.....	2
ΠΕΡΙΕΧΟΜΕΝΑ	3
ΜΕΡΟΣ Α.....	6
ΠΕΡΙΛΗΨΗ.....	6
ΚΕΦΑΛΑΙΟ 1	7
ΕΙΣΑΓΩΓΗ ΣΤΑ ΤΗΛΕΠΙΚΟΙΝΩΝΙΑΚΑ ΔΙΚΤΥΑ.....	7
Ορισμός.....	7
ΤΟΠΙΚΑ ΑΣΥΡΜΑΤΑ ΔΙΚΤΥΑ	8
1.1 Εισαγωγή.....	8
1.2 Σύγκριση WLAN - LAN	8
1.3 Εφαρμογές ασύρματων δικτύων.....	11
1.4 Πρωτόκολλα δρομολόγησης στα ad hoc δίκτυα	12
ΚΕΦΑΛΑΙΟ 2 ΠΡΟΤΥΠΟ 802.11.....	15
2.1 Εισαγωγή.....	15
2.2 Διαστρωμάτωση	16
2.3 Φυσικό στρώμα	16
2.3.1 MAC	17
ΜΕΡΟΣ Β	19
ΕΙΣΑΓΩΓΗ	19
ΚΕΦΑΛΑΙΟ 3.....	22
ΟΜΝΕΤ ++	22
3.1 Εισαγωγή.....	22
3.2 Χρησιμοποιώντας το ΟΜΝΕΤ++.....	22
3.3 Δημιουργία και Εκτέλεση μιας Προσομοίωσης	25
3.4 Εργαλεία Προσομοίωσης που Βασίζονται στο ΟΜΝΕΤ++.....	26
3.4.1 INET Framework	27
3.4.2 Mobility Framework	33
3.4.3 IPv6Suite	35
ΚΕΦΑΛΑΙΟ 4	37
CNET Network Simulator.....	37

4.1 Παρουσίαση του CNET.....	37
4.2 Το Μοντέλο Προσομοίωσης του CNET	46
ΚΕΦΑΛΑΙΟ 5	48
GloMoSim	48
5.1 Περιγραφή του GloMoSim	48
5.2 Χτίζοντας ένα Δίκτυο με το GloMoSim	49
ΚΕΦΑΛΑΙΟ 6	50
NIST Products	50
6.1 ATM/HFC Network Simulator	50
6.2 Private Network-Network Interface (PNNI) Simulator	52
ΚΕΦΑΛΑΙΟ 7	53
Network Simulator 2 (NS-2).....	53
7.1 Εισαγωγή.....	53
7.2 Εγκαθιστώντας το NS-2.....	53
7.3 Περιγραφή και Δυνατότητες του NS-2	55
7.4 Network Animator (NAM).....	59
ΚΕΦΑΛΑΙΟ 8	61
Η ΓΛΩΣΣΑ NED.....	61
8.1 Simple modules	61
8.2 Compound modules	62
8.3 Κανάλια (channels).....	63
Παράμετροι.....	63
Θύρες (Gates)	64
ΚΕΦΑΛΑΙΟ 9	66
ΠΕΙΡΑΜΑΤΙΚΟ ΜΕΡΟΣ.....	66
9.1 Εισαγωγή.....	66
9.2 Πρώτο Σενάριο	66
Αρχείο εξόδου	67
Επεξεργασία των αποτελεσμάτων	69
Επεξεργασία του cwnd	69
Επεξεργασία του throughput	70
Μετρήσεις.....	71
9.3 Δεύτερο Πείραμα	71

Αρχείο εξόδου	73
Επεξεργασία των αποτελεσμάτων	75
Επεξεργασία του cwnd	75
Επεξεργασία του throughput	77
Μετρήσεις	77
9.4 Τρίτο σενάριο.....	78
Επεξεργασία των αποτελεσμάτων	82
Επεξεργασία του cwnd	82
Επεξεργασία του throughput	85
Μετρήσεις.....	85
9.5 Παρατηρήσεις.....	86
ΠΑΡΑΡΤΗΜΑΤΑ.....	87
6.1 Κώδικας φίλτρου AWK	87
6.2 Κώδικας Πρώτου Σεναρίου.....	88
6.3 Κώδικας Δεύτερου Σεναρίου	92
6.4 Κώδικας Τρίτου Σεναρίου	95
Συμπεράσματα.....	100
ΒΙΒΛΙΟΓΡΑΦΙΑ.....	101

ΜΕΡΟΣ Α

ΠΕΡΙΛΗΨΗ

Τα τελευταία χρόνια οι τεχνικοί δικτύων ολοένα και περισσότερο στηρίζονται στην προσομοίωση για να σχεδιάσουν και να αναπτύξουν τα πολύπλοκα και επιρρεπή σε σφάλματα δίκτυα. Αντιστοίχως οι ερευνητές των δικτύων όλο και πιο πολύ στηρίζονται στην προσομοίωση για να εξερευνήσουν την συμπεριφορά και τις επιδόσεις των νέων πρωτοκόλλων που έχουν σχεδιαστεί. Πάντως, παρόλη αυτή την χρήση της προσομοίωσης, στις μέρες μας υπάρχει ελάχιστη κοινή αντίληψη του βαθμού του ελέγχου της αξιοπιστίας που απαιτείται για τις διάφορες εφαρμογές προσομοίωσης. Επίσης υπάρχει ελάχιστη γνώση που αφορά την αποτελεσματικότητα των τεχνικών του ελέγχου αξιοπιστίας.

Στην προσπάθεια να υλοποιηθεί η ανάγκη για προσομοίωση των πρωτοκόλλων και των υπηρεσιών των δικτύων δημιουργήθηκαν από πολλές ερευνητικές ομάδες προγράμματα τα οποία ασχολούνται με την προσομοίωση δικτύων, πρωτοκόλλων, υπηρεσιών και οτιδήποτε άλλο χρήζει ενδιαφέροντος προσομοίωσης με σκοπό την καλύτερη σχεδίαση και ανάπτυξη των δικτύων. Βέβαια από τα πολλά προγράμματα που έχουν δημιουργηθεί είναι λογικό κάποια από αυτά να απαιτούν κάποιο οικονομικό αντίτιμο ώστε να είναι δυνατή η χρήση τους από τους απλούς χρήστες, ενώ αρκετά από αυτά υπάρχουν ελεύθερα στο διαδίκτυο και μπορεί οποιοσδήποτε απλός χρήστης να τα «κατεβάσει» και να τα χρησιμοποιήσει στον προσωπικό του υπολογιστή.

Στην εργασία αυτή σκοπός είναι να περιγραφούν και να παρουσιαστούν κάποια από αυτά τα προγράμματα που προσφέρονται ελεύθερα στο ευρύ κοινό ώστε να μπορεί κανείς να επιλέξει το εργαλείο που του ταιριάζει ώστε να προσομοιώσει με το σωστό και αποτελεσματικό τρόπο τις δικτυακές λειτουργίες που επιθυμεί.

ΚΕΦΑΛΑΙΟ 1

ΕΙΣΑΓΩΓΗ ΣΤΑ ΤΗΛΕΠΙΚΟΙΝΩΝΙΑΚΑ ΔΙΚΤΥΑ



Ορισμός

Ένα τηλεπικοινωνιακό δίκτυο είναι ένα σύνολο από κόμβους που είναι διασυνδεδεμένοι έτσι ώστε να επιτρέπουν την ανταλλαγή πληροφορίας.

Ξεκινώντας την περιήγησή μας στο χώρο των τηλεπικοινωνιών αξίζει να σχολιάσουμε ότι στόχος των επικοινωνιών γενικά είναι η αποστολή ενός μηνύματος από ένα σημείο σε ένα άλλο καθώς και η επιβεβαίωση της πλήρους, ορθής και κατανοητής λήψης του από έναν παραλήπτη. Παρά την απλή αυτή αναφορά, πίσω από τις επικοινωνίες κρύβεται πλήθος από θεωρίες και τεχνικές που ασχολούνται με την αποστολή του παραπάνω μηνύματος.

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.

1.2 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:

> 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 επέκταση 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.

- > *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:

Destination-Sequenced Distance-Vector (DSDV)

Temporally Ordered Routing Algorithm (TORA)

Dynamic Source Routing (DSR)

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.

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

2.2 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

2.3 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.

2.3.1 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.

ΜΕΡΟΣ Β

ΕΙΣΑΓΩΓΗ

Τα δίκτυα συνεχίζουν να αναπτύσσονται όλο και πιο πολύπλοκα όσο η βιομηχανία παράγει τόσο ενσύρματες όσο και ασύρματες τεχνολογίες σε μεγάλης κλίμακας αρχιτεκτονικές δικτύων, αλλά και όσο οι εφαρμογές των χρηστών και η «κίνηση» σε αυτά συνεχίζουν να εξελίσσονται. Στις μέρες μας ελάχιστες εταιρίες μπορούν να ανεχτούν το κόστος για να χαραμίζουν επιπλέον megabits το δευτερόλεπτο σε ένα ερευνητικό έργο (project), όταν ο προϋπολογισμός της εταιρίας προστάζει για ακριβή, άρτια και οικονομικά σχέδια δικτύων από την αρχή.

Αντιμέτωποι με αυτή την τεράστια ανάπτυξη τα τελευταία χρόνια οι τεχνικοί και οι ερευνητές των δικτύων σχεδόν καθολικά χρησιμοποιούν την προσομοίωση με σκοπό να προβλέψουν την αναμενόμενη απόδοση πολύπλοκων δικτύων και να κατανοήσουν τη συμπεριφορά των υπαρχόντων πρωτοκόλλων δικτύων που δεν έχουν στηριχτεί σε πρότυπα πώς λειτουργούν στα σημερινά δίκτυα. Η προσομοίωση επίσης διαρκώς και πιο πολύ χρησιμοποιείται για να προβλέψει την ορθότητα και τις επιδόσεις των νέων σχεδίων πρωτοκόλλων για τα δίκτυα. Επιπλέον η χρησιμοποίηση της προσομοίωσης πλέον εμφανίζεται σαν μια αυστηρή απαίτηση στην διαδικασία που οδηγεί στην διεθνή πιστοποίηση, όπως το πρότυπο IMT-2000 για την τρίτη γενιά, ασύρματης και κινητής τηλεφωνίας.

Βέβαια αυτή η αναπτυσσόμενη εμπιστοσύνη στην προσομοίωση αυξάνει της απαιτήσεις σε ότι αφορά την εξασφάλιση της ορθότητας και της πρόβλεψης των σφαλμάτων σε συγκεκριμένα μοντέλα προσομοίωσης. Παρόλα αυτά όμως δεν υπάρχουν ευρέως αποδεκτές πρακτικές και τεχνικές για να βοηθήσουν στην αξιολόγηση των προσομοιώσεων των δικτύων και στην αποτίμηση εμπιστοσύνης στα αποτελέσματά τους.

Οι πρώτες εργασίες που αφορούσαν την τεχνική σχεδίαση και έρευνα των δικτύων περιείχε τόσο πειραματικά όσο και μαθηματικά μοντέλα για να αποδείξει την επίτευξη και την εξασφάλιση των ορίων για την προβλεπόμενη απόδοση. Τα τελευταία δέκα χρόνια καθώς τα δίκτυα έγιναν τόσο πολύ μεγάλα ώστε να επιτρέπουν εύκολο πειραματισμό, αλλά και τόσο πολύπλοκα ώστε να επιτρέπουν μια αρκετά εύκολη μαθηματική ανάλυση, η προσομοίωση των δικτύων ήρθε να συμπληρώσει ένα πολύ σημαντικό ρόλο, βοηθώντας τους ερευνητές και τους σχεδιαστές να αντιληφθούν την συμπεριφορά και την απόδοση των πρωτοκόλλων και των δικτύων. Σήμερα η προσομοίωση συχνά χρησιμοποιείται :

- για να την απόδοση των τωρινών δικτύων και πρωτοκόλλων με σκοπό να βοηθήσει στον καθορισμό της τεχνολογίας, να προγραμματίσει τις παραγόμενες δυνατότητες και να επιδείξει την εκπλήρωση των απαιτήσεων του πελάτη.
- για να προβλέψει την αναμενόμενη συμπεριφορά των νέων δικτυακών πρωτοκόλλων και σχεδίων μέσα από ποιοτικές και ποσοτικές εκτιμήσεις απόδοση ή ορθότητας και
- για να εξερευνήσει γρήγορα τα όρια και τις δυνατότητες ενδεχόμενων σχεδίων πρωτοκόλλων μέσα από σύντομη αποτίμηση και επανάληψη.

Λόγω αυτής της μεγάλης εξάπλωσης της προσομοίωσης των δικτύων έχει δημιουργηθεί ένας μεγάλος αριθμός εργαλείων με τα οποία μπορεί να γίνει η προσομοίωση σε διάφορες τοπολογίες ή σε διάφορα πρωτόκολλα και τα οποία μέσω των γραφικών που διαθέτουν να καταδείξουν τα αποτελέσματα και τις επιδόσεις που προκύπτουν από την λειτουργία ενός δικτύου κάτω από συγκεκριμένες συνθήκες και παραμέτρους. Τα εργαλεία αυτά που έχουν δημιουργηθεί αναφέρονται σε διάφορες πλατφόρμες και υποστηρίζονται από συγκεκριμένα λειτουργικά συστήματα. Πολλά αυτά λειτουργούν μόνο σε περιβάλλοντα Linux ή Unix, κάποια άλλα σε περιβάλλοντα Windows, μερικά σε Solaris, ενώ λίγα σχετικά προσφέρονται και είναι συμβατά σε όλα τα λειτουργικά συστήματα.

Όλα σχεδόν τα εργαλεία προσομοίωσης δικτύων απαιτούν αρκετά μεγάλες προγραμματιστικές ικανότητες σε συνδυασμό με άριστη γνώση των δικτύων και των πρωτοκόλλων τους, καθώς για να προκύψουν σωστά, αλλά κυρίως χρήσιμα αποτελέσματα τα οποία θα χρησιμοποιηθούν για την αποφυγή ανεπιθύμητων σφαλμάτων, θα πρέπει να μπορεί ο προγραμματιστής να εκτιμήσει και να δώσει τις σωστές παραμέτρους που θα εξαντλούν όλες τις πιθανές περιπτώσεις χρήσης ενός δικτύου. Είναι άσκοπο να ξοδεύεις άπειρες ώρες στην προσομοίωση μιας δικτυακής εφαρμογής αν στο τέλος δε μπορείς να βγάλεις ασφαλή και τεκμηριωμένα αποτελέσματα για την αξιοπιστία, τις επιδόσεις, την ασφάλεια και γενικά την άρτια λειτουργία της εφαρμογής. Όπως προαναφέρθηκε λοιπόν είναι απαραίτητο να δοθεί η ίδια έμφαση στην αξιολόγηση των αποτελεσμάτων που προκύπτουν από την προσομοίωση.

Το θέμα που θα μας απασχολήσει σε αυτή την εργασία αφορά στην παρουσίαση κάποιων προγραμμάτων προσομοίωσης τα οποία είναι ελεύθερα προς χρήση και δεν είναι απαραίτητο να πληρώσει κάποιος για να μπορέσει να τα χρησιμοποιήσει στον προσωπικό του υπολογιστή. Υπάρχουν όπως είπαμε πολλά προγράμματα προσομοίωσης, με αποτέλεσμα αρκετά από αυτά να απαιτούν κάποιο αντίτιμο ώστε να επιτρέψουν σε κάποιον να τα κατεβάσει μέσω του διαδικτύου και να τα χρησιμοποιήσει στον υπολογιστή του, όπως είναι το OPNET, το QUALNET και άλλα. Βέβαια από την άλλη μεριά υπάρχουν και πολλά προγράμματα προσομοίωσης τα

οποία μπορεί κάποιος ελεύθερα να τα κατεβάσει μέσω του διαδικτύου και να πειραματιστεί με τις υπηρεσίες που αυτά προσφέρουν, όπως είναι το NS-2, το OMNET++, το GloMoSim, το CNET και άλλα τα οποία θα παρουσιάσουμε πιο αναλυτικά στη συνέχεια. Όπως είπαμε τα προγράμματα αυτά δεν λειτουργούν σε όλες τις πλατφόρμες και όλα τα λειτουργικά συστήματα, οπότε για να μπορεί κάποιος να «τρέξει» το πρόγραμμα που επιθυμεί θα πρέπει να διαθέτει και το κατάλληλο λειτουργικό σύστημα ή το κατάλληλο λογισμικό ώστε να μπορέσει να λειτουργήσει σωστά το πρόγραμμα.

ΚΕΦΑΛΑΙΟ 3

OMNET ++

3.1 Εισαγωγή

Το OMNET++ είναι ένα πακέτο προσομοίωσης διακριτών γεγονότων βασισμένο στη γλώσσα C++ που αναπτύχθηκε στο Τεχνικό Πανεπιστήμιο της Βουδαπέστη από τον Andr'as Varga. Ο αρχικός τομέας εφαρμογών του OMNET++ είναι η προσομοίωση δικτύων υπολογιστών και άλλων κατανεμημένων συστημάτων. Είναι πρόγραμμα open-source, ελεύθερο για μη κερδοσκοπική χρήση, και έχει μια αρκετά μεγάλη και ενεργή κοινότητα χρηστών. Επιτρέπει επίσης τη σχεδίαση αρθρωτών μοντέλων προσομοίωσης, τα οποία μπορούν να συνδυαστούν και να επαναχρησιμοποιηθούν αρκετά εύκολα. Έχει αποδειχθεί ότι αυτό το πλαίσιο προσομοίωσης είναι κατάλληλο για την προσομοίωση σύνθετων συστημάτων όπως τους κόμβους Διαδικτύου και τη δυναμική των πρωτοκόλλων TCP/IP με τρόπο πολύ ρεαλιστικό. Το OMNET++ υποστηρίζει ένα πολύ καλά δομημένο και εξίσου βοηθητικό γραφικό περιβάλλον το GNED το οποίο υπάρχει ελεύθερο στο internet ώστε να μπορεί ο οποιοσδήποτε κοινός χρήστης να το κατεβάσει και να το τρέξει στον υπολογιστή του. Το OMNET++ αν και μπορεί από μόνο του να χρησιμοποιηθεί για εφαρμογές προσομοίωσης χρησιμοποιείται ευρέως ως πλατφόρμα πάνω στην οποία έχουν αναπτυχθεί άλλα προγράμματα προσομοίωσης που αφορούν σε συγκεκριμένες κατηγορίες και τύπους δικτύων και τα οποία θα αναφέρουμε στη συνέχεια.

Το OMNET++ τρέχει εξίσου καλά στα Linux και στα περισσότερα λειτουργικά συστήματα που βασίζονται ή μοιάζουν με Unix, αλλά και σε Win32 πλατφόρμες (Windows 2000, XP). Υπάρχει ελεύθερο στο internet και στο site <http://www.omnetpp.org/>.

3.2 Χρησιμοποιώντας το OMNET++

Ο προσομοιωτής OMNET++ λοιπόν μπορεί να χρησιμοποιηθεί για :

- Μοντελοποίηση κυκλοφορίας των δικτύων τηλεπικοινωνιών
- Μοντελοποίηση πρωτοκόλλων
- Μοντελοποίηση δικτύων ουρών αναμονής
- Μοντελοποίηση πολυεπεξεργαστών και άλλων κατανεμημένων hardware συστημάτων
- Επικύρωση αρχιτεκτονικών υλικού

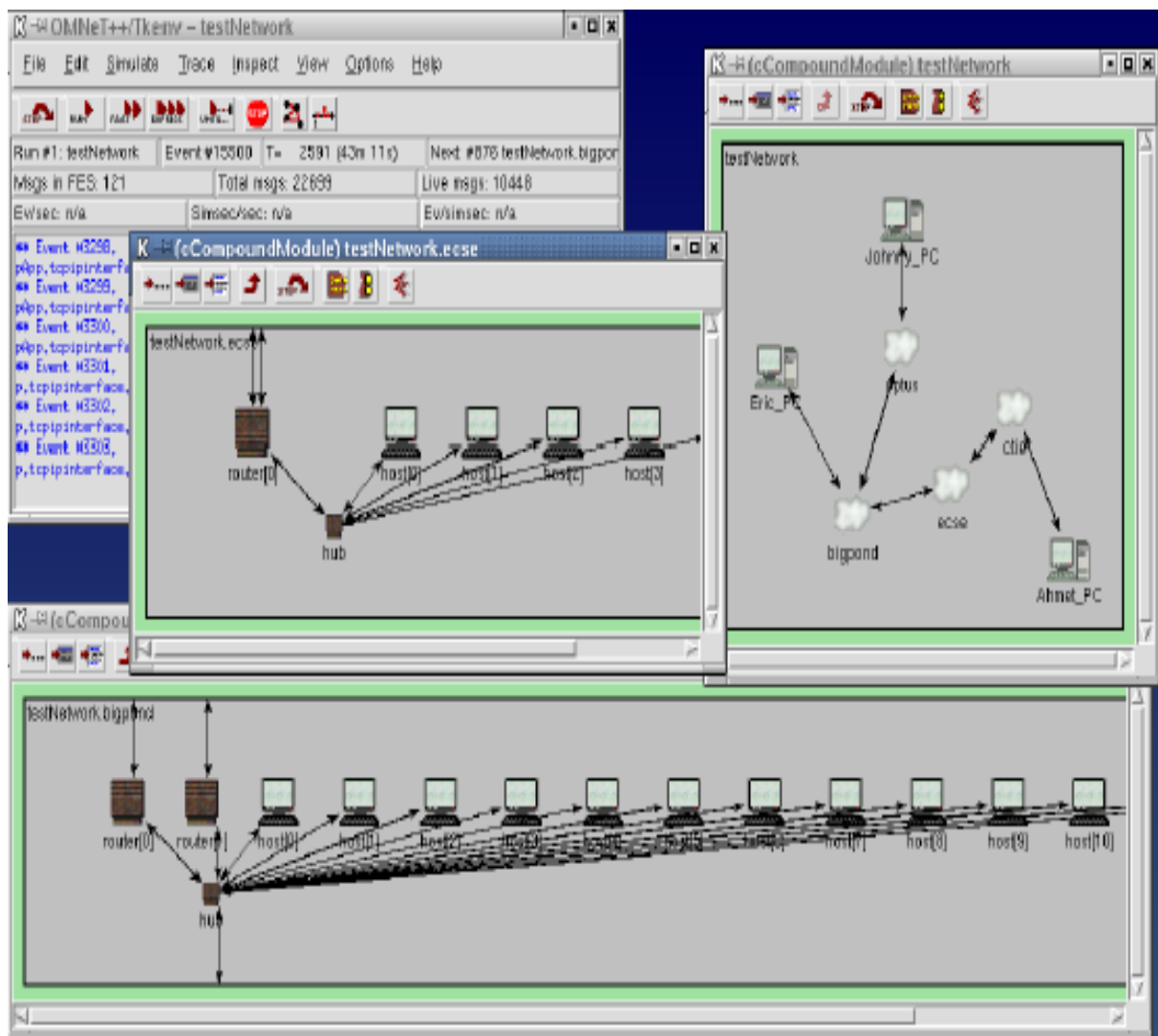
- Αξιολόγηση των πτυχών απόδοσης των σύνθετων συστημάτων λογισμικού <>
- Μοντελοποίηση οποιουδήποτε άλλου συστήματος όπου η προσέγγιση διακριτών γεγονότων είναι κατάλληλη.

Τα προσομοιωμένα μοντέλα αποτελούνται από ιεραρχικά τοποθετημένες ενότητες. Στο OMNET++ υπάρχουν δύο τύποι ενότητων: απλές και σύνθετες ενότητες. Οι απλές ενότητες διαμορφώνουν το χαμηλότερο επίπεδο ιεραρχίας και εφαρμόζουν την δραστηριότητα μιας ενότητας και μπορούν αυθαίρετα να συνδυαστούν για να διαμορφώσουν τις σύνθετες ενότητες. Οι ενότητες επικοινωνούν με τη μεταβίβαση μηνυμάτων. Τα μηνύματα μπορεί να σταλούν είτε μέσω των συνδέσεων που εκτείνονται μεταξύ των ενότητων είτε άμεσα στις ενότητες προορισμού τους. Ο χρήστης καθορίζει τη δομή του μοντέλου (τις ενότητες και τη διασύνδεσή τους) με τη χρησιμοποίηση της γλώσσας περιγραφής τοπολογίας (NED) από το OMNET++.

Οι απλές ενότητες εφαρμόζονται στην C++, χρησιμοποιώντας τις κλήσεις συστήματος του πυρήνα προσομοίωσης και την βιβλιοθήκη της κλάσης της προσομοίωσης. Για κάθε απλή ενότητα, είναι δυνατό να επιλεγεί προσομοίωση μεταξύ των τύπων process-style και protocol-style, ανάλογα με το τι επιθυμεί ο χρήστης. Επομένως, διαφορετικά μέρη υπολογισμού και συστημάτων επικοινωνιών μπορούν να προγραμματιστούν με φυσικό τρόπο και να συνδεθούν εύκολα. Η βιβλιοθήκη της κλάσης προσομοίωσης παρέχει μια καλά ορισμένη διεπαφή (API) στον προγραμματιστή εφαρμογής για τις πιο κοινές

εργασίες προσομοίωσης, που περιλαμβάνουν: τυχαία παραγωγή αριθμού, ουρές, πίνακες, μηνύματα, εξερεύνηση τοπολογίας και δρομολόγηση. Επίσης δημιουργία και καταστροφή ενότητας, δυναμικές τοπολογίες, στατιστικά στοιχεία, εκτίμηση πυκνότητας (που περιλαμβάνει ιστογράμματα, P2 και K-split) καθώς και καταγραφή των δεδομένων εξόδου. Η αντικειμενοστραφής προσέγγιση επιτρέπει την εύκολη επέκταση των κλάσεων της βάσης που παρέχονται από τον πυρήνα προσομοίωσης.

Τα στοιχεία του μοντέλου συντάσσονται και συνδέονται με τη βιβλιοθήκη προσομοίωσης και μια από τις βιβλιοθήκες διεπαφής με το χρήστη για να διαμορφώσουν ένα εκτελέσιμο πρόγραμμα. Μια βιβλιοθήκη διεπαφής με τον χρήστη βελτιστοποιείται για το commandline, ενώ η άλλη χρησιμοποιεί ένα γραφικό περιβάλλον διεπαφής με τον χρήστη (GUI) που μπορεί να χρησιμοποιείται για να εντοπίσει και να διορθώσει τα λάθη της προσομοίωσης (για παράδειγμα, το παρακάτω σχήμα παρουσιάζει την προσομοίωση της μορφής ενός δικτύου).



Σχήμα 1 : Οθόνη του OMNET++ που δείχνει ένα ιεραρχικό μοντέλο δικτύου. Το πάνω δεξιά παράθυρο δείχνει την τοπολογία του δικτύου προσομοίωσης που αποτελείται από τέσσερα υποδίκτυα. Οι τοπολογίες των δύο από αυτά τα υποδίκτυα, ecse και bigpond φαίνονται στο μεσαίο και στο κάτω παράθυρο αντίστοιχα.

Το GUI καθιστά το εσωτερικό ενός μοντέλου προσομοίωσης πλήρως ορατό, επιδεικνύει γραφικά το δίκτυο, δείχνει με κίνηση τη ροή των μηνυμάτων και αφήνει το χρήστη να επιλέξει τα αντικείμενα (μηνύματα, ουρές, κ.λ.π.) μέσα στο μοντέλο. Είναι επίσης δυνατό να αλλαχτούν οι παράμετροι και τα πεδία μηνυμάτων για τη διαδικασία της διόρθωσης. Τα χαρακτηριστικά της οπτικοποίησης καθιστούν το OMNET++ κατάλληλο ακόμη και για εκπαιδευτικούς ή λόγους επίδειξης. Λόγω της αρθρωτής σχεδίασης, είναι δυνατό να ενσωματωθεί η μηχανή προσομοίωσης (συμπεριλαμβανομένων των μοντέλων) σε άλλες εφαρμογές. Το OMNET++ επίσης υποστηρίζει την προσομοίωση παράλληλων διακριτών γεγονότων (PDES).

3.3 Δημιουργία και Εκτέλεση μιας Προσομοίωσης

Ένα μοντέλο στο OMNET++ αποτελείται από τα ακόλουθα μέρη :

- Τη γλώσσα NED (.ned αρχεία) που περιγράφει την τοπολογία, την δομή του μοντέλου με τις παραμέτρους, τις πύλες κ.τ.λ. Τα αρχεία NED μπορεί να γραφτούν σε οποιοδήποτε επεξεργαστή κειμένου ή με τον GNED γραφικό επεξεργαστή.
- Τους ορισμούς των μηνυμάτων (msg. αρχεία). Μπορεί ο προγραμματιστής να ορίσει διάφορους τύπους μηνυμάτων και να προσθέσει πεδία δεδομένων σε αυτά. Το OMNET++ θα μεταφράσει τους ορισμούς των μηνυμάτων σε πλήρως δομημένες κλάσεις της C++.
- Τα αρχεία από τις απλές ενότητες. Είναι αρχεία C++ με κατάληξη .h/.cc .

Το σύστημα προσομοίωσης παρέχει τα ακόλουθα μέρη :

- Ο πυρήνας προσομοίωσης. Αυτός περιέχει τον κώδικα που διαχειρίζεται την προσομοίωση και την βιβλιοθήκη της κλάσης προσομοίωσης. Είναι γραμμένος σε C++ και μαζί με την μεταγλώττιση δημιουργεί μια βιβλιοθήκη (ένα αρχείο με κατάληξη .a ή .lib).
- Διεπαφή με τον χρήστη. Το OMNET++ παρέχει διεπαφή με τον χρήστη που χρησιμοποιείτε κατά την εκτέλεση της προσομοίωσης για να εξηγήσει τον έλεγχο των λαθών, για επίδειξη των αποτελεσμάτων ή για παράλληλη εκτέλεση προσομοιώσεων. Είναι γραμμένη σε C++ και μαζί με την μεταγλώττιση δημιουργεί μια βιβλιοθήκη (ένα αρχείο με κατάληξη .a ή .lib).

Τα προγράμματα προσομοίωσης δημιουργούνται από τα παραπάνω συστατικά μέρη. Πρώτα τα αρχεία NED μεταγλωττίζονται σε C++ πηγαίο κώδικα, χρησιμοποιώντας τον NEDC μεταφραστή που είναι μέρος του OMNET++. Στη συνέχεια όλα τα αρχεία με τον πηγαίο κώδικα C++ μεταγλωττίζονται και συνδέονται με τον πυρήνα προσομοίωσης και την διεπαφή του χρήστη για να διαμορφώσουν ένα εκτελέσιμο αρχείο προσομοίωσης.

Το εκτελέσιμο αρχείο προσομοίωσης είναι ένα αυτόνομο πρόγραμμα, κατά συνέπεια μπορεί να τρέξει σε άλλες μηχανές χωρίς το OMNET++ ή τα αρχεία του μοντέλου να είναι παρόντα. Όταν το πρόγραμμα αρχίζει, διαβάζει ένα αρχείο σχηματισμού (configuration file συνήθως αποκαλούμενο omnetpp.ini). Αυτό το αρχείο περιέχει τις

ρυθμίσεις που ελέγχουν το πώς η εκτελείται προσομοίωση, τιμές των παραμέτρων του μοντέλου κ.λ.π. Το αρχείο σχηματισμού μπορεί επίσης να ορίσει διαφορετικούς τρόπους με τους οποίους μπορεί να τρέξει η προσομοίωση και στην απλούστερη περίπτωση, θα εκτελεσθούν από το πρόγραμμα προσομοίωσης το ένα μετά από άλλο.

Το αποτέλεσμα της προσομοίωσης γράφεται σε αρχεία εξόδου : σε ενδιάμεσα αρχεία εξόδου, σε κλιμακωτά αρχεία εξόδου, και ενδεχομένως στα αρχεία εξόδου του χρήστη. Το OMNET++ παρέχει ένα εργαλείο GUI που ονομάζεται Ploive για να βλέπει και να σχεδιάζει το περιεχόμενο των αρχείων εξόδου των αποτελεσμάτων της προσομοίωσης. Δεν αναμένεται ότι κάποιος θα επεξεργαστεί τα αρχεία αποτελέσματος χρησιμοποιώντας το OMNET++ μόνο: τα αρχεία εξόδου είναι αρχεία κειμένων που μπορούν να διαβαστούν από προγράμματα όπως το Matlab ή το Octave, ή να εισαχθεί στους υπολογισμούς με λογιστικό φύλλο (spreadsheet) όπως στο OpenOffice, το Calc, το Gnumeric ή το Excel. Όλα αυτά τα εξωτερικά προγράμματα παρέχουν την πλούσιες λειτουργίες για στατιστική ανάλυση και απεικόνιση των αποτελεσμάτων, και είναι έξω από το πεδίο του OMNET++ να αναπαράγει αυτές τις λειτουργίες.

Ο αρχικός σκοπός της διεπαφής με τον χρήστη είναι να κατασταθούν τα εσωτερικά στοιχεία του μοντέλου ορατά στο χρήστη, να ελεγχθεί η εκτέλεση προσομοίωσης και να επιτραπεί ενδεχομένως στον χρήστη να επέμβει με την αλλαγή των μεταβλητών και των αντικειμένων μέσα στο μοντέλο. Αυτό είναι πολύ σημαντικό στη φάση της ανάπτυξης και διόρθωσης του προγράμματος προσομοίωσης. Εξίσου σημαντικό είναι και το γεγονός ο χρήστης αποκτά μια οπτική εμπειρία που τον διευκολύνει να πάρει μια αίσθηση της συμπεριφοράς του μοντέλου. Το γραφικό ενδιάμεσο με τον χρήστη μπορεί επίσης να χρησιμοποιηθεί για να καταδείξει τη λειτουργία ενός προτύπου.

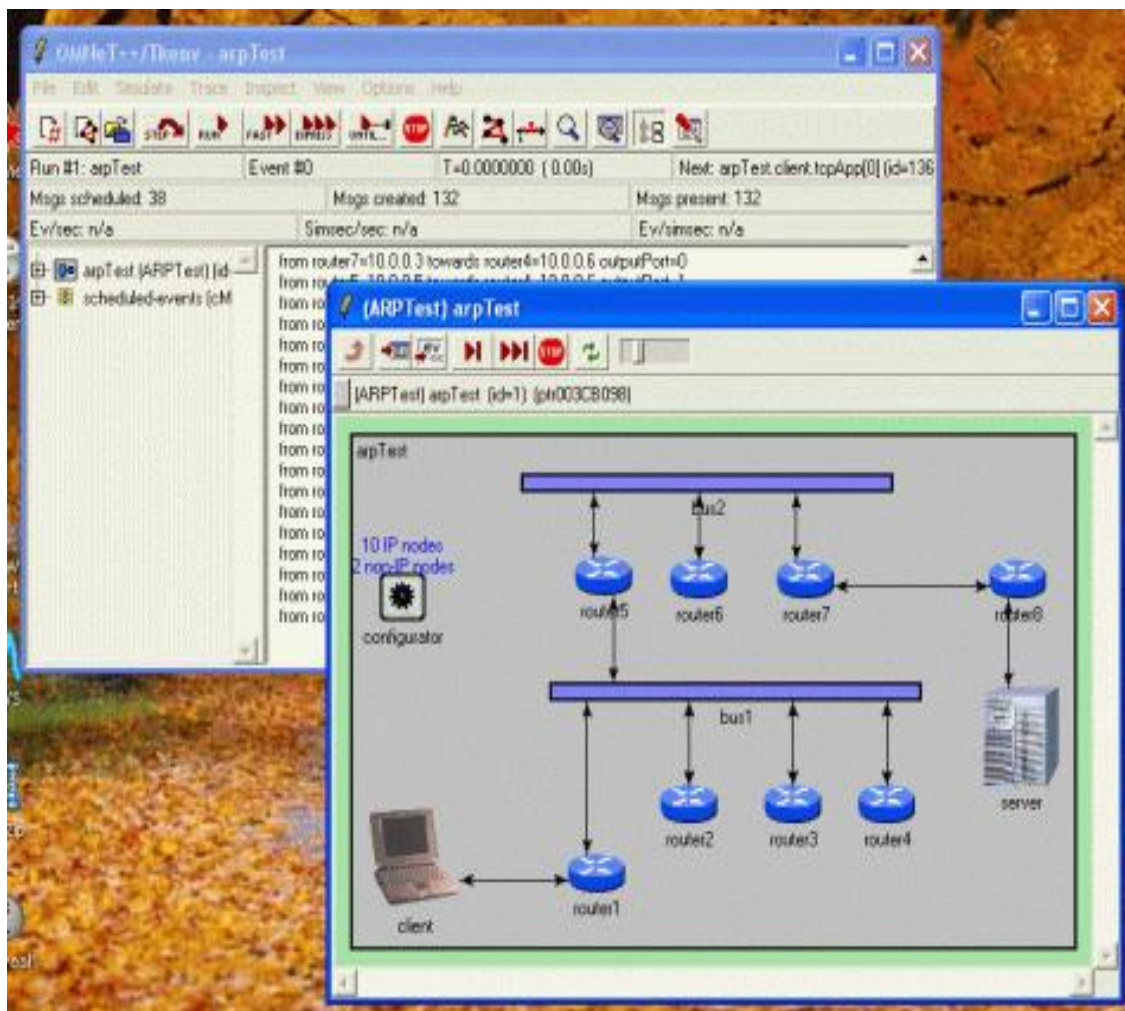
Το ίδιο μοντέλο προσομοίωσης μπορεί να εκτελεσθεί με διαφορετική διεπαφή με τον χρήστη, χωρίς οποιαδήποτε αλλαγή στα αρχεία του μοντέλου από τον ίδιο. Ο χρήστης θα μπορεί να εξετάζει και να διορθώνει την προσομοίωση με ένα ισχυρό γραφικό ενδιάμεσο με τον χρήστη, και τελικά να την τρέχει με ένα απλό και γρήγορο γραφικό περιβάλλον που υποστηρίζει εκτέλεση batch.

3.4 Εργαλεία Προσομοίωσης που Βασίζονται στο OMNET++

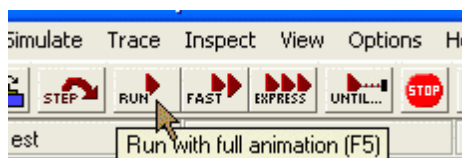
Η ανάπτυξη και χρησιμότητα του OMNET++ οδήγησε στην ανάγκη να δημιουργηθούν εργαλεία τα οποία να εξειδικεύουν την προσομοίωση σε συγκεκριμένους τύπους και σχηματισμούς δικτύων. Στη συνέχεια θα αναφερθούμε στα εργαλεία INET Framework, Mobility Framework και IPv6Suite.

3.4.1 INET Framework

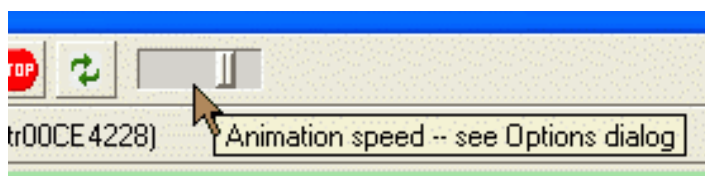
Το INET Framework είναι ένα πρόγραμμα προσομοίωσης το οποίο είναι κατάλληλο για την προσομοίωση του TCP/IP και άλλων πρωτοκόλλων που σχετίζονται με το internet, που είναι γραμμένο για το περιβάλλον του OMNET++. Αν κάποιος εγκαταστήσει το INET Framework στον υπολογιστή του και προσπαθήσει να το τρέξει αντικρίσει το εξής παράθυρο :



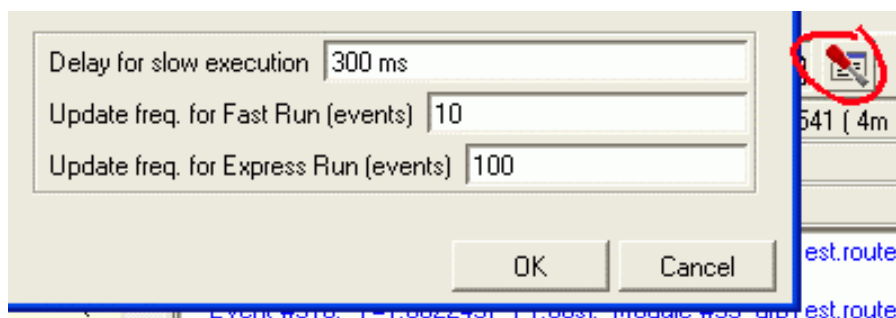
Στο πίσω μέρος της εικόνας φαίνεται η αρχική οθόνη του OMNET++ ενώ στο μπροστινό μέρος βλέπουμε το περιβάλλον του INET Framework. Για να τρέξουμε μια προσομοίωση θα πρέπει να πατήσουμε το κουμπί Run από κύριο μενού εργαλείων ή πατάμε το F5. Για να την σταματήσουμε αρκεί να πατήσουμε το κόκκινο STOP ή πατώντας το F8.



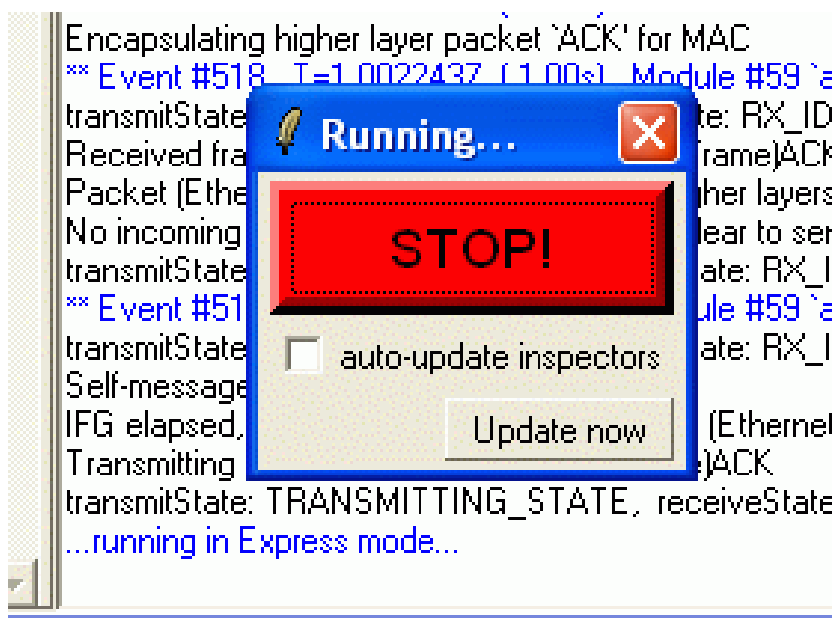
Στη συνέχεια η προσομοίωση ξεκινάει και ο χρήστης μπορεί να επιλέξει το πόσο γρήγορα θέλει να τρέχει αυτή απλά καθορίζοντας την από γραμμή κύλισης στο πάνω μέρος του παραθύρου του προγράμματος.



Το εικονίδιο Fast θα σβήνει το μήνυμα της προσομοίωσης και θα ανανεώνει την γραφική απεικόνιση μόνο ανάμεσα σε κάθε 10 γεγονότα. Αυτό βέβαια μπορούμε να το ρυθμίσουμε από το μενού των επιλογών. Το κουμπί Express παρέχει την υψηλότερη ταχύτητα εκτέλεσης και αυτόματα αν πατηθεί κλείνουν οι γραφικές απεικονίσεις προς τον χρήστη, ενώ ανανεώνει της ενδείξεις στην οθόνη κάθε 1000 γεγονότα κάτι που επίσης μπορεί να αλλάξει από το μενού επιλογών.



Στην κατάσταση Express μπορούμε να σταματήσουμε την προσομοίωση μόνο πατώντας το μεγάλο κόκκινο STOP που φαίνεται παρακάτω.

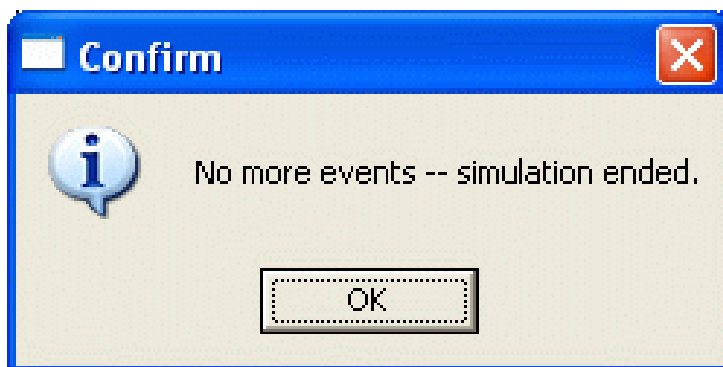


Στο status bar της εφαρμογής μπορούμε να δούμε την ταχύτητα της προσομοίωσης. Η ένδειξη Ev/sec μας λέει πόσα γεγονότα εκτελεί η CPU σε ένα δευτερόλεπτο. Αυτό εξαρτάται από τις δυνατότητες του υλικού του χρήστη, από το μέγεθος του κώδικα της C++ που χρησιμοποιείται στο μοντέλο και τον τρόπο εκτέλεσης της προσομοίωσης (Normal, Fast, Express). Είναι σύνηθες η Express προσομοίωση να είναι 100-200 φορές γρηγορότερη από την Fast προσομοίωση.

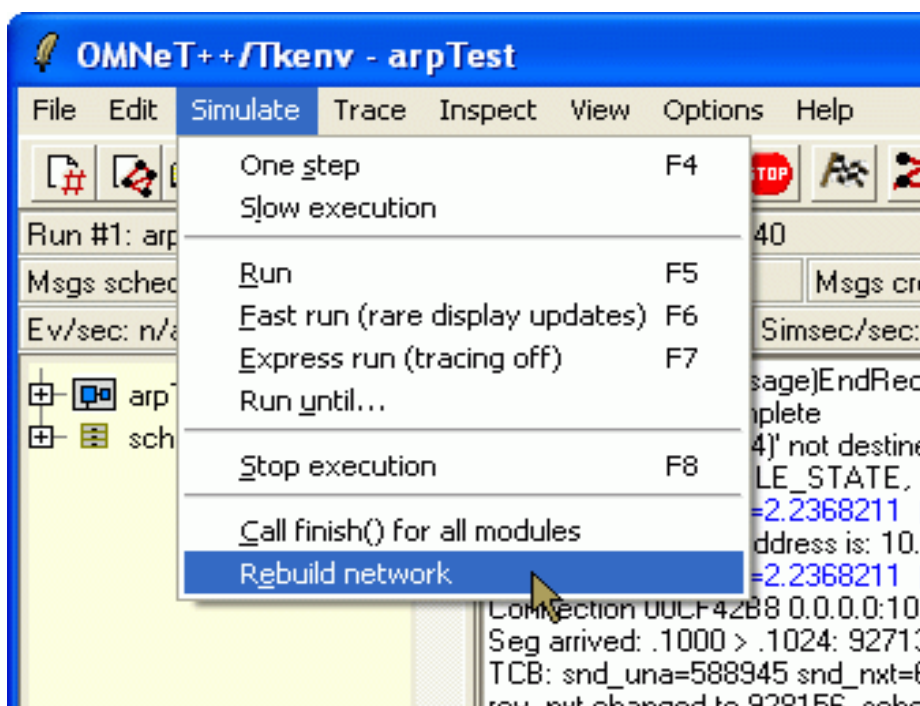
Msgs scheduled: 13	Msgs created: 701	Msgs present: 144
Ev/sec: 189.81	Simsec/sec: 0.00526841	Ev/simsec: 36028

Στην παραπάνω μπάρα μπορούμε επίσης μπορούμε να δούμε πόσα δευτερόλεπτα προσομοίωσης έχουν περάσει σε ένα πραγματικό δευτερόλεπτο εκτέλεσης (Simsec/sec) και πόσα γεγονότα συνέβησαν σε ένα δευτερόλεπτο προσομοίωσης (Ev/Simsec). Η ένδειξη Simsec/sec είναι χρήσιμη για να υπολογίσουμε πόσο θα κρατήσει η προσομοίωση. Η ένδειξη Ev/Simsec είναι ανεξάρτητη από το αν χρησιμοποιούμε Normal, Fast ή Express εκτέλεση και εξαρτάται μόνο από τη φύση και το μέγεθος του μοντέλου.

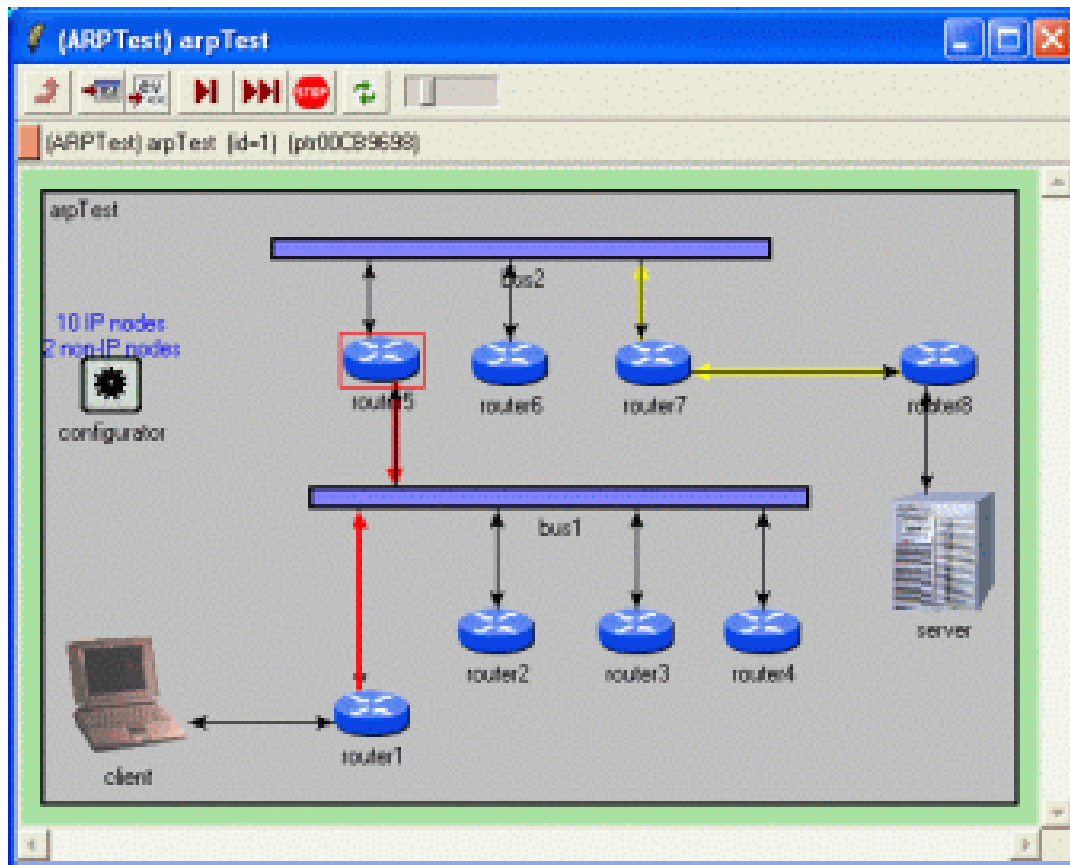
Αφού τρέξει η προσομοίωσή μετά από λίγο θα εμφανιστεί το επόμενο εικονίδιο πληροφόρησης :



Αυτό το παράθυρο μας πληροφορεί ότι η προσομοίωση τελείωσε και ότι δεν υπάρχουν άλλα γεγονότα τα οποία να μπορούν προσομοιωθούν. Αυτό στο επίπεδο του μοντέλου σημαίνει ότι ο χρήστης έκλεισε την TCP σύνδεση και τελείωσε τις ενέργειές του, ενώ δεν κυκλοφορούν άλλα πακέτα και δεν τρέχει κανένας χρονομετρητής σε ολόκληρο το μοντέλο. Μπορεί επίσης κάποια προσομοίωση να τελειώσει μετά από κάποιο συγκεκριμένο χρόνο προσομοίωσης ή αφού επιτεύχθηκε η απαιτούμενη στατιστική ακρίβεια που είχε καθοριστεί. Σε όλες αυτές τις περιπτώσεις μπορεί να ξαναξεκινήσει πατώντας το Rebuild network στο μενού Simulate. Στην πραγματικότητα αυτό μπορεί να γίνει σε οποιαδήποτε στιγμή της εκτέλεσης της προσομοίωσης και αυτή να επανεκκινήσει.

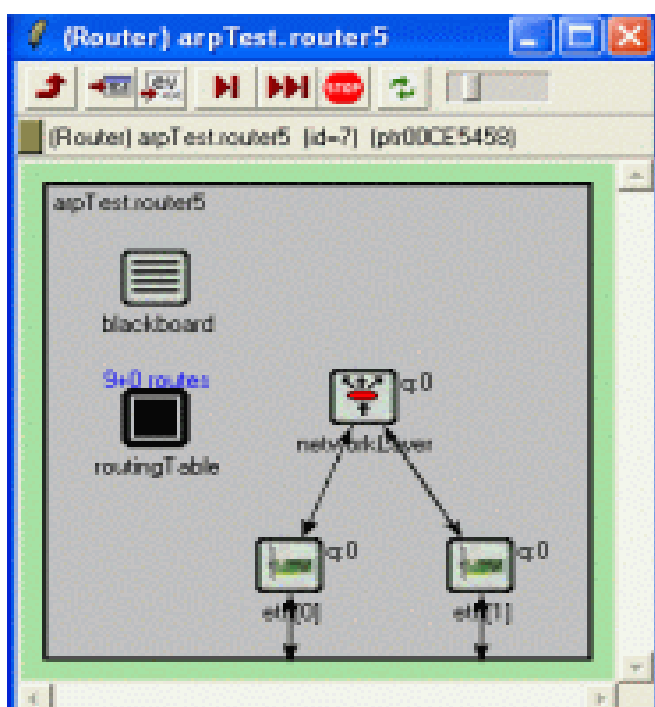
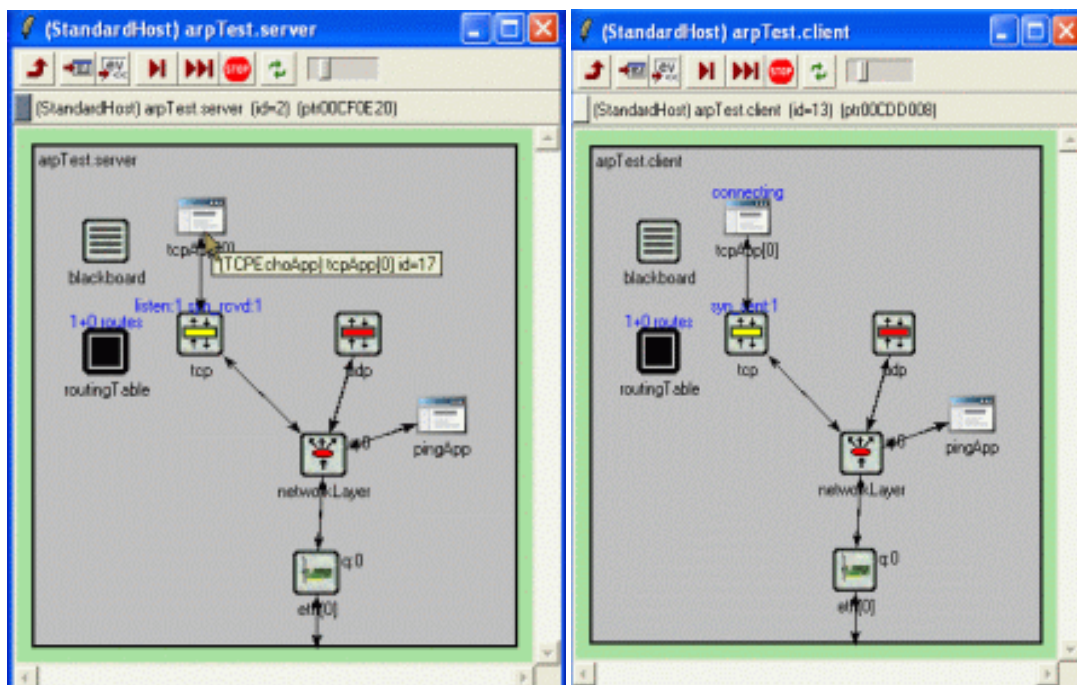


Κατά τη διάρκεια μιας προσομοίωσης βλέπουμε κάποια links στο μοντέλο να αναβοσβήνουν με συγκεκριμένα χρώματα, όπως φαίνεται στην εικόνα που ακολουθεί :

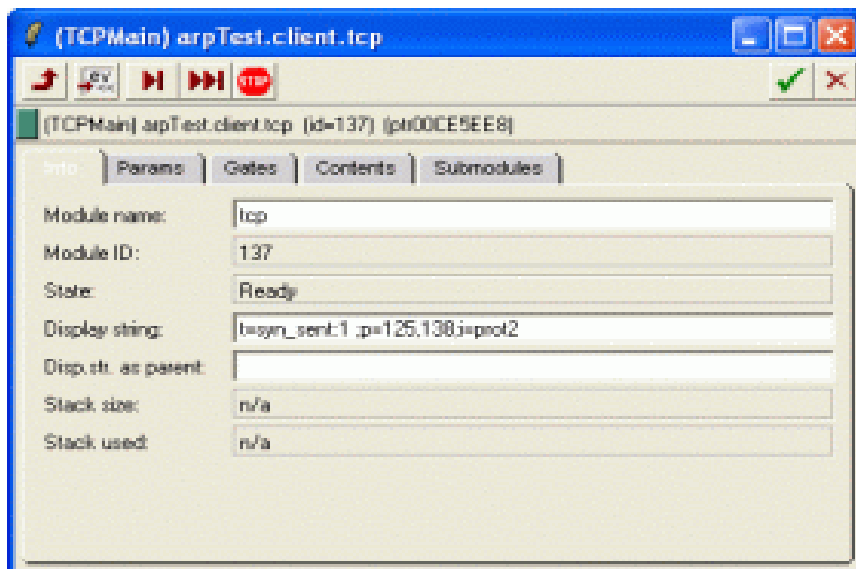


Αν ένα link είναι κίτρινο σημαίνει ότι ο κόμβος στέλνει στο link. Αν ένας κόμβος έχει υποστεί σύγκρουση και σταματήσει την αποστολή για λίγο τότε το link γίνεται κόκκινο. Το κόκκινο πλαίσιο που φαίνεται στο σχήμα γύρω από τον router 5 επισημαίνει την τοποθεσία στην οποία βρίσκεται το γεγονός που εκτελείται την συγκεκριμένη στιγμή.

Τέλος θα πρέπει να αναφερθεί ότι κάνοντας διπλό κλικ σε έναν κόμβο του σχεδιασμένου μοντέλου μπορούμε να δούμε τα εσωτερικά χαρακτηριστικά του.

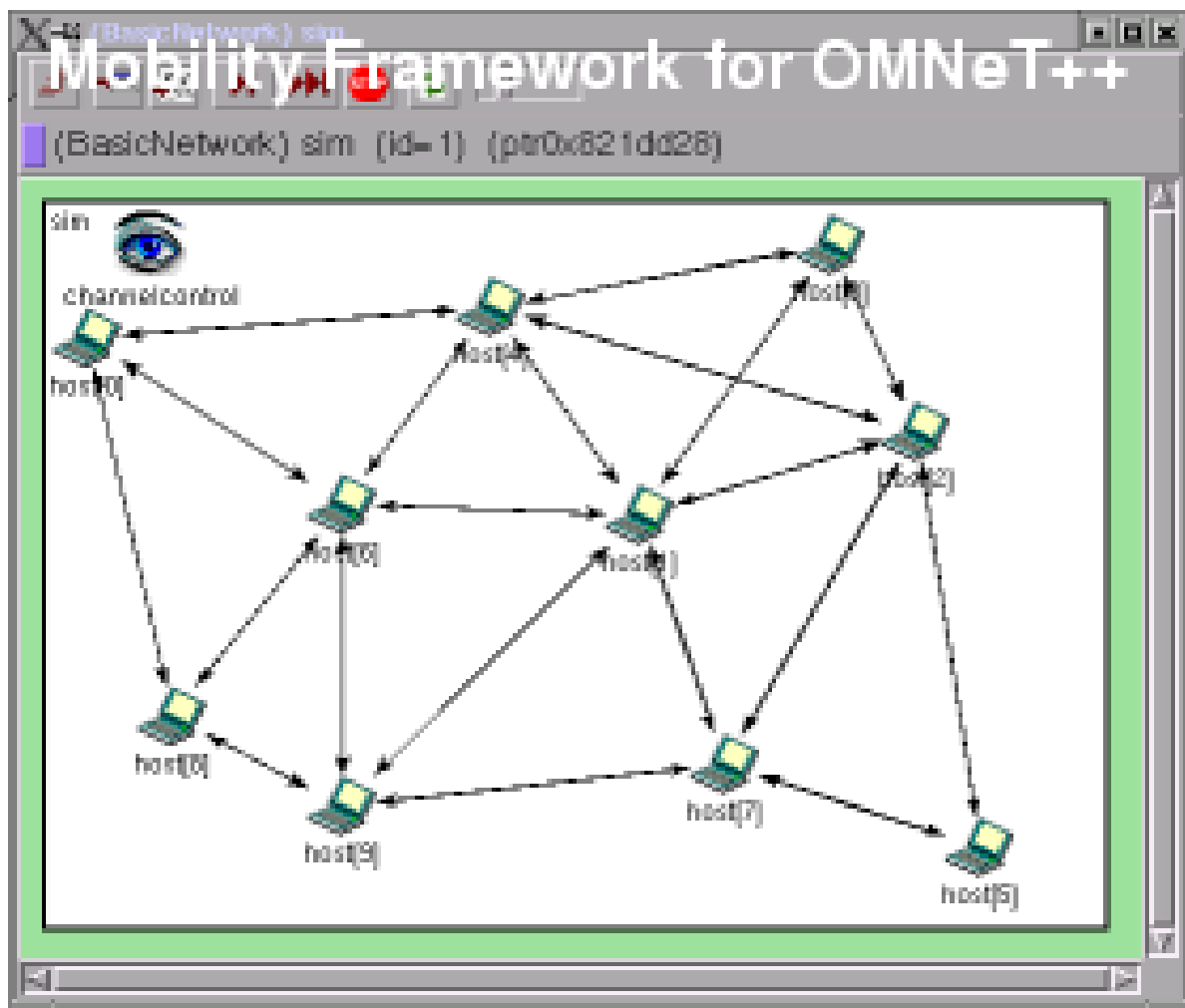


Αν στην συνέχεια ξαναπατήσουμε διπλό κλικ σε αυτά τα στοιχεία που βλέπουμε (TCP, UDP, etc.) τότε ανοίγει ένα παράθυρο με επιλογές όπου μπορούμε να ορίσουμε τα χαρακτηριστικά τους.



3.4.2 Mobility Framework

Αυτό το εργαλείο έχει τη δυνατότητα να υποστηρίξει ασύρματα και κινητά δίκτυα προσομοίωσης μέσα στα πλαίσια που προσφέρει το OMNET++. Το εργαλείο αυτό παρέχει υποστήριξη για κινητούς κόμβους, διαχείριση δυναμικής σύνδεσης και μοντέλα ασύρματων καναλιών. Επιπλέον προσφέρει απλές ενότητες υλοποίησης οι οποίες μπορούν να συνδυαστούν και να παράγουν σύνθετες ενότητες. Με αυτά τα δεδομένα είναι αρκετά εύκολο για έναν προγραμματιστή να αναπτύξει ένα δικό του πρωτόκολλο για το Mobility Framework (MF) χωρίς να χρειάζεται να ασχοληθεί με το interface άλλα και με τα στοιχεία διαλειτουργικότητας.



Το MF μπορεί να χρησιμοποιηθεί για την προσομοίωση :

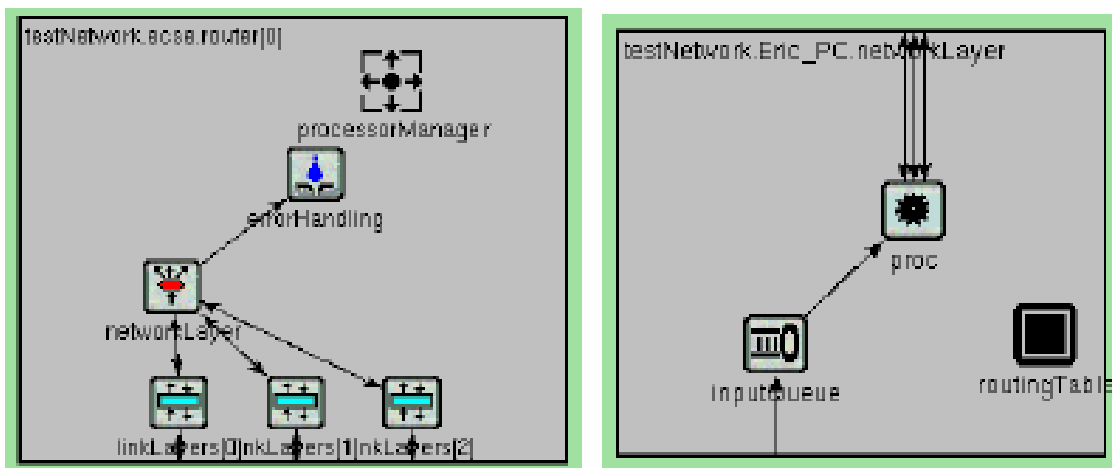
- Σταθερά ασύρματα δίκτυα
- Κινητά ασύρματα δίκτυα
- Κατανεμημένα (ad-hoc) και συγκεντρωτικά δίκτυα
- Δίκτυα αισθητήρων
- Πολυκάναλα ασύρματα δίκτυα
- Πολλές άλλες προσομοιώσεις που χρειάζονται κινητή υποστήριξη ή ασύρματο interface.

Αυτό τον καιρό αναπτύσσεται μια βιβλιοθήκη με τα πρότυπα πρωτόκολλα για το MF (802.11, AODV, ...), με τελικό σκοπό να δημιουργηθεί μια πλούσια βιβλιοθήκη για τέτοιου είδους πρωτόκολλα έτσι ώστε να γίνει εύκολη η προσομοίωση τους μέσα από

το εργαλείο. Υπάρχει εγχειρίδιο για την λειτουργία του προγράμματος το οποίο είναι προσβάσιμο μέσω internet σε μορφή pdf με το οποίο θα πρέπει να ξεκινήσει κάποιος που ενδιαφέρεται να υλοποιήσει προγράμματα προσομοίωσης με τη χρήση του FM. Για να εγκαταστήσει κάποιος το MF στον υπολογιστή του θα πρέπει να διαθέτει την έκδοση 3.0a4 ή παραπάνω του OMNET++ με όλες του τις λειτουργίες. Αν πληροί αυτές τις προϋποθέσεις αρκεί να κατεβάσει την τελευταία έκδοση το MF από το site <http://mobility-fw.sourceforge.net>.

3.4.3 IPv6Suite

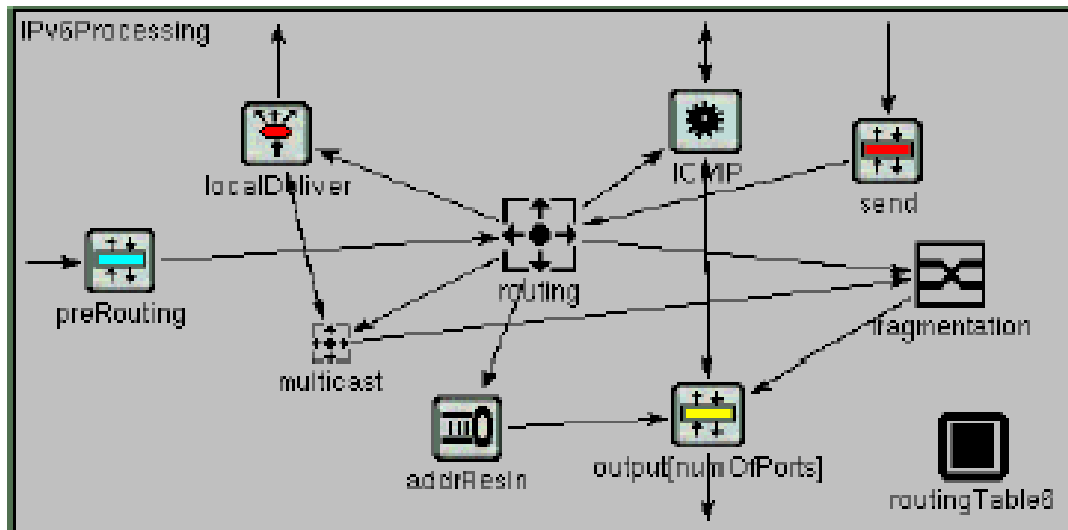
Το IPv6Suite εργαλείο προσομοίωσης αποτελείται από πολλά λειτουργικά μέρη και όπως είναι λογικό τα κυριότερα μέρη αφορούν τα επίπεδα δικτύωσης (network layer) και ελέγχου σύνδεσης δεδομένων (data link layer). Με τον συνδυασμό αυτών των μερών μπορούν να δημιουργηθούν προσομοιωμένοι hosts, routers, Ethernet, point-to-point συνδέσεις κ.α. Στην παρακάτω εικόνα φαίνεται το μοντέλο ενός δρομολογητή (router) σε τρεις δικτυακούς τύπους. Το εργαλείο αυτό παρέχει σημαντικές βελτιώσεις στην υλοποίηση προσομοιώσεων IP πρωτοκόλλων χρησιμοποιώντας διάφορους μηχανισμούς φυσικής επικοινωνίας (point-to-point συνδέσεις, Ethernet συνδέσεις κ.α.), όπως επίσης και στο σχεδιασμό κόμβων χρησιμοποιώντας οποιονδήποτε συνδυασμό με όλες αυτές τις φυσικές συνδέσεις.



Το μοντέλο προσομοίωσης ενός router σε τρεις διαφορετικούς τύπους δικτύων

Επιπλέον το IPv6Suite υποστηρίζει την μοντελοποίηση διπλής στοίβας δρομολογητές που επιτρέπουν την ροή πακέτων μέσω του IPv4 και του IPv6 ταυτόχρονα. Ακόμη επιτρέπει πολλά interfaces του επιπέδου ελέγχου σύνδεσης δεδομένων να είναι παρόντα σε ένα και μόνο κόμβο.

Για να τρέξουμε μια προσομοίωση με αυτό το εργαλείο θα πρέπει να δημιουργήσουμε τη σχεδίαση του μοντέλου μέσα από το οποίο θα φαίνεται ο προορισμός των πακέτων, ο αποστολέας και ο δέκτης καθώς και οι μηχανισμοί αναγνώρισης των γειτονικών κόμβων. Μια τέτοια σχεδίαση μέσω του εργαλείου IPv6Suite φαίνεται στο σχήμα που ακολουθεί :



Το IPv6Suite είναι διαθέσιμο στο internet και η εγκατάστασή του απαιτεί την ύπαρξη του OMNET++. Βρίσκεται ελεύθερο για κατέβασμα στην ιστοσελίδα <http://ctieware.eng.monash.edu.au/twiki/bin/view/Simulation/Documentation>

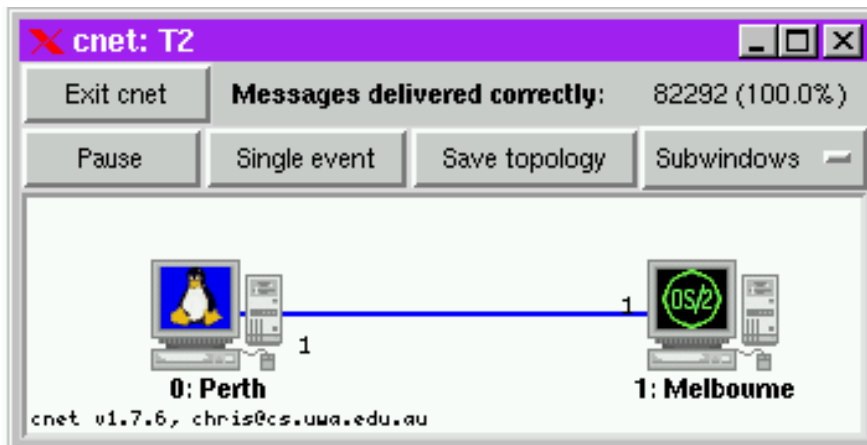
ΚΕΦΑΛΑΙΟ 4

CNET Network Simulator

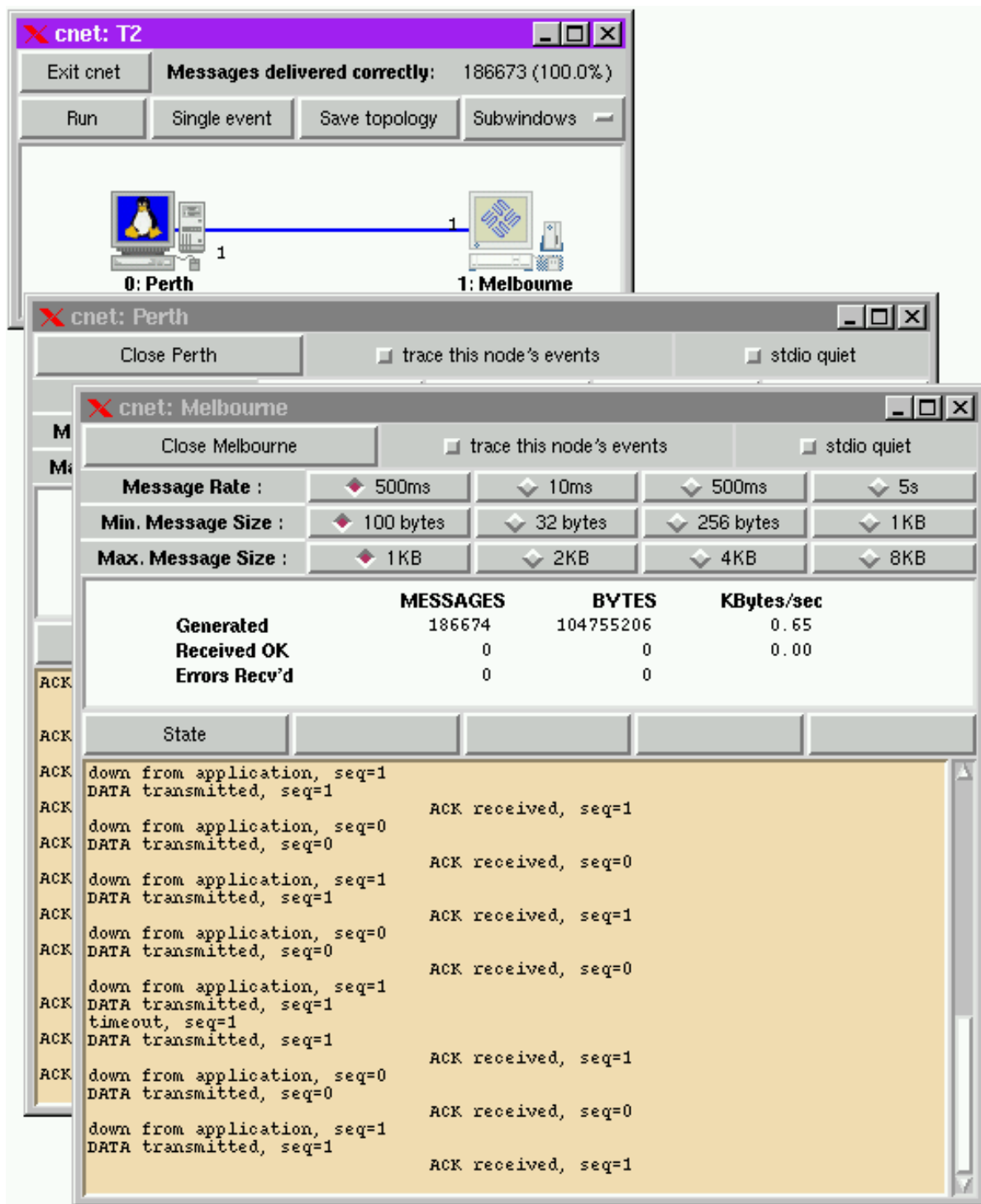
4.1 Παρουσίαση του CNET

Το CNET είναι εργαλείο προσομοίωσης δικτύων το οποίο επιτρέπει τον πειραματισμό με πολλά πρωτόκολλα των επιπέδων ελέγχου σύνδεσης δεδομένων (data-link layer), δικτύωσης (network layer), δρομολόγησης και επικοινωνίας, καθώς και σε δίκτυα που αποτελούνται από οποιοδήποτε συνδυασμό point-to-point συνδέσεων και IEEE 802.3 Ethernet στοιχείων. Με βάση το μοντέλο OSI το cnet παρέχει προσομοίωση στα επίπεδα εφαρμογών αλλά και στο φυσικό επίπεδο. Τα πρωτόκολλα που μπορούν να γραφτούν από τους χρήστες είναι απαραίτητο να συμπληρώσουν τα ενδιάμεσα επίπεδα και πιο συγκεκριμένα να υπερκεράσουν τα κατεστραμμένα ή χαμένα πλαίσια που το φυσικό επίπεδο του cnet τυχαία παράγει. Επιπλέον οι εξειδικευμένοι χρήστες μπορούν να αναπτύξουν διαφορετικά το επίπεδο εφαρμογών και το φυσικό επίπεδο και να παράγουν διαφορετικά στατιστικά χαρακτηριστικά της παραγωγής μηνυμάτων και της αποστολής δεδομένων. Το εύρος της προσομοίωσης μπορεί να ποικίλει από λίγους μέχρι λίγες εκατοντάδες κόμβους. Το cnet είναι ένα open-source εργαλείο το οποίο μπορεί κανείς να κατεβάσει από το site <http://www.csse.uwa.edu.au/cnet/>

Το cnet είτε εμφανίζει ολόκληρο το δίκτυο με τη βοήθεια της Tcl/Tk είτε τρέχει με λιγότερο γραφικό τρόπο σε ένα ASCII παράθυρο. Με τη βοήθεια της Tcl/Tk το cnet παρέχει μια γραφική απεικόνιση του δικτύου κατά την εκτέλεση και επιτρέπει ένα μεγάλο αριθμό ρυθμίσεων και χαρακτηριστικών του δικτύου να μπορούν να μετατραπούν κατά τη διάρκεια της εκτέλεσης της προσομοίωσης. Οι κόμβοι μπορούν να επιλεγούν με το ποντίκι και με διπλό κλικ να εμφανιστεί ένα παράθυρο το οποίο δείχνει την έξοδο και τα στατιστικά του πρωτοκόλλου του κόμβου. Μερικά από τα χαρακτηριστικά του κόμβου, όπως ο ρυθμός και το μέγεθος των μηνυμάτων, μπορούν να τροποποιηθούν κατά τη διάρκεια της εκτέλεσης της προσομοίωσης επιλέγοντας τα κουμπιά επιλογής. Ομοίως οι προεπιλεγμένες ρυθμίσεις για τα χαρακτηριστικά των κόμβων μπορεί ταυτόχρονα να αλλάξουν επιλέγοντας και αλλάζοντας τις καθολικές μεταβλητές. Από ένα άλλο μενού κάθε κόμβος μπορεί να αναγκαστεί να επανεκκινήσει, να διακοπεί, να τερματιστεί και να επανεκκινήσει, να σταματήσει προσωρινά και να αποτύχει.



Αυτή είναι μια τυπική απεικόνιση μιας προσομοίωσης με δύο κόμβους, όπου εδώ συγκεκριμένα χρησιμοποιείται το πρωτόκολλο *stop-and-wait*. Η προσομοίωση βρίσκεται κατά τη διάρκεια της εκτέλεσης αλλά υπάρχει δυνατότητα να σταματήσει και έπειτα να πραγματοποιηθεί με την εκτέλεση ενός βήματος τη φορά χρησιμοποιώντας τα κουμπιά. Ένα σύνολο από 82292 μηνύματα έχουν εμφανιστεί έως τώρα και το επίπεδο εφαρμογών δεν έχει παρουσιάσει εσφαλμένα μηνύματα. Άρα το πρωτόκολλο αναφέρεται από το πρόγραμμα σαν να είναι 100% σωστό.



Εδώ κάναμε κλικ σε έναν από τους κόμβους έτσι ώστε να φανεί το παράθυρο εξόδου. Ένα παράθυρο εξόδου είναι διαθέσιμο για κάθε κόμβο ακόμα και σε μεγάλες προσομοιώσεις.

Επιλέγοντας έναν σύνδεσμο έχει ως αποτέλεσμα να εμφανίζεται ένα υποπαράθυρο το οποίο δείχνει στατιστικά στοιχεία της μετάδοσης για το συγκεκριμένο link. Οι σύνδεσμοι είναι και προς τις δύο κατευθύνσεις έτσι επιλέγοντας έναν από αυτούς σημαίνει ότι κάνουμε κλικ σε σημείο κοντά στον κόμβο πηγή. Τα χαρακτηριστικά της σύνδεσης για το κόστος αποστολής και τις πιθανότητες λάθους μπορούν να τροποποιηθούν ενώ η προσομοίωση βρίσκεται σε εκτέλεση μετακινώντας τις μπάρες κύλισης. Ομοίως τα χαρακτηριστικά όλων των συνδέσεων του δικτύου μπορούν ταυτόχρονα να τροποποιηθούν κατά την διάρκεια εκτέλεσης της προσομοίωσης επιλέγοντας και αλλάζοντας τα συνολικά χαρακτηριστικά των συνδέσεων.

The screenshot displays the 'cnet: T2' simulation interface. The main window shows a network topology with two nodes: 0: Perth (Linux) and 1: Melbourne (Windows). A status bar indicates 'Messages delivered correctly: 1517 (100.0%)'. Below the topology is a detailed window for the 'Perth -> Melbourne link' showing transmission statistics and adjustable parameters.

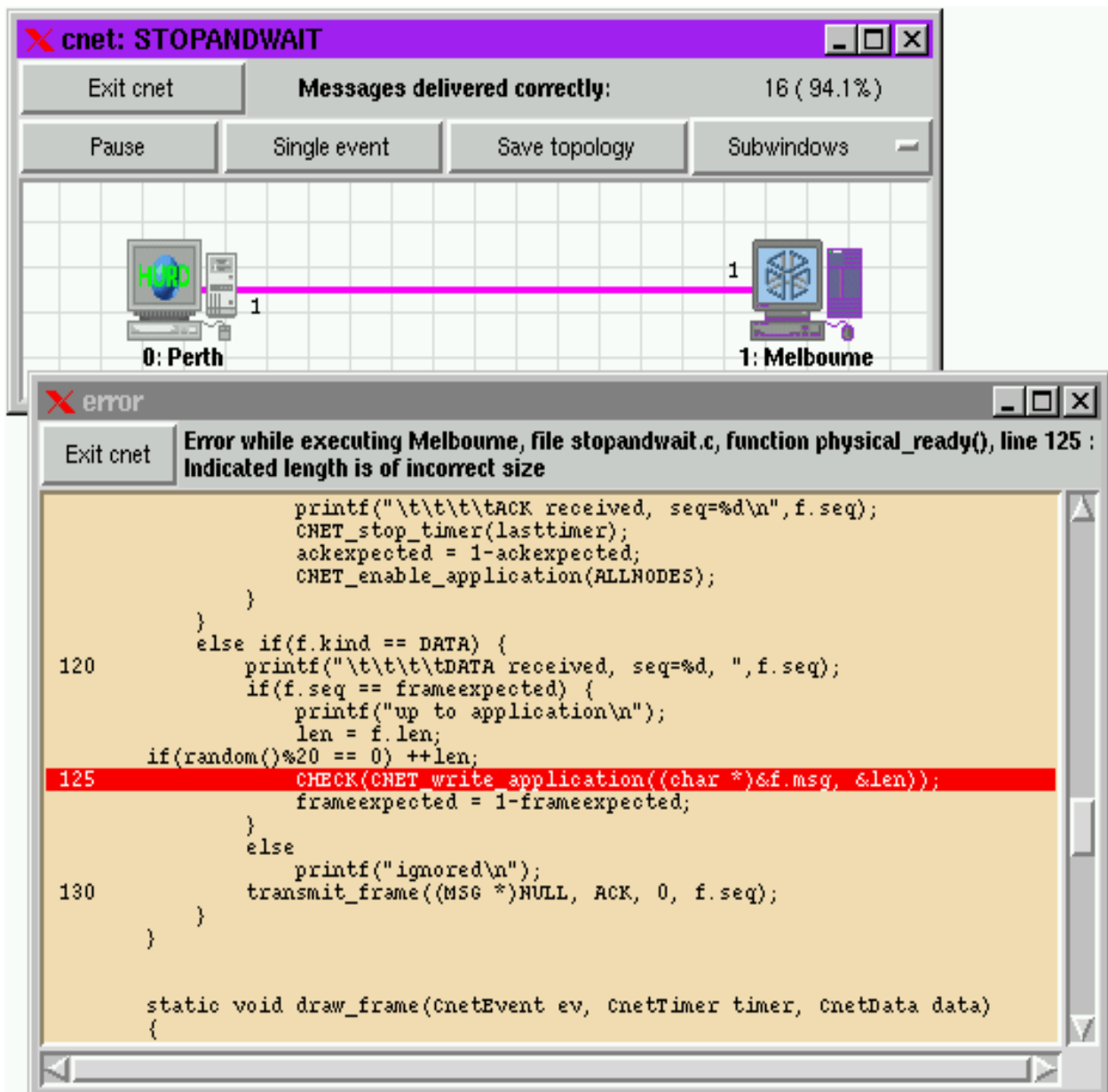
	FRAMES	BYTES	KBytes/ sec
Transmitted	5245	1506279	2.32
Received	4608	2217572	3.42
Errors Introduced	3282	950498	

Adjustable parameters for the link:

- Cost(per byte): 1
- Cost(per frame): 5
- Prob(frame corruption) $1:2^N$: 1
- Prob(frame loss) $1:2^N$: 2

Το cnet είναι ικανό αυτόματα να ανιχνεύει και αναφέρει έναν μεγάλο αριθμό σφαλμάτων τα οποία είναι πιθανόν να γίνουν κατά την ανάπτυξη ενός πρωτοκόλλου. Αυτά περιέχουν λάθος ορισμό των παραμέτρων σε λειτουργίες, λάθος ορισμός των

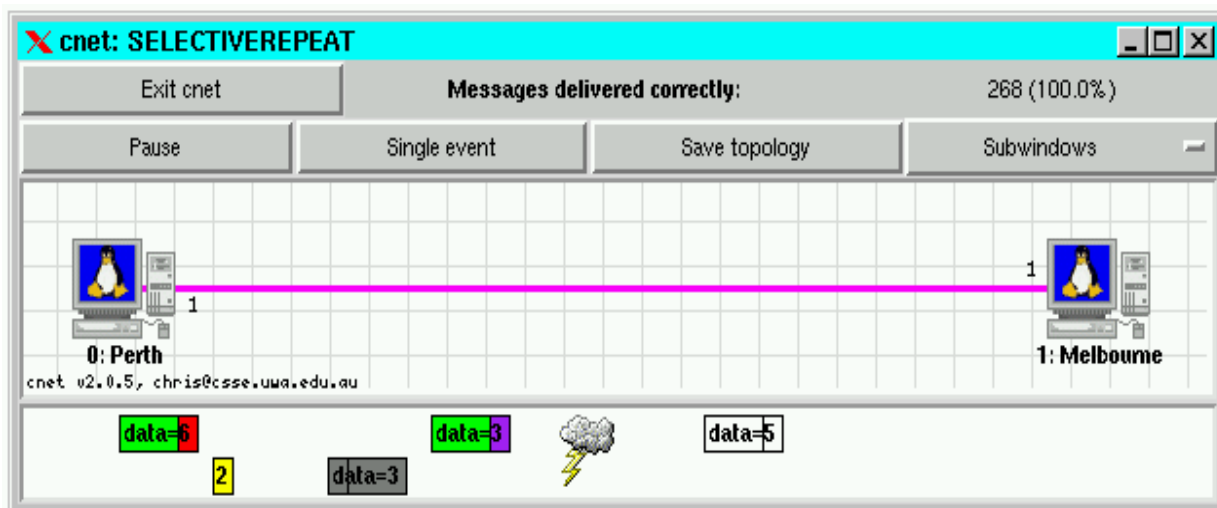
συνδέσεων, διευθύνσεις και χρονομετρητές, μηνύματα που πέρασαν σε λάθος προορισμούς, εκτός σειράς ή διπλά μηνύματα λάθους. Όπως φαίνεται στην ακόλουθη εικόνα, τα λάθη αναφέρονται σε ένα pop-up παράθυρο περιγράφοντας το λάθος και τη θέση του στον κώδικα πηγής του χρήστη.



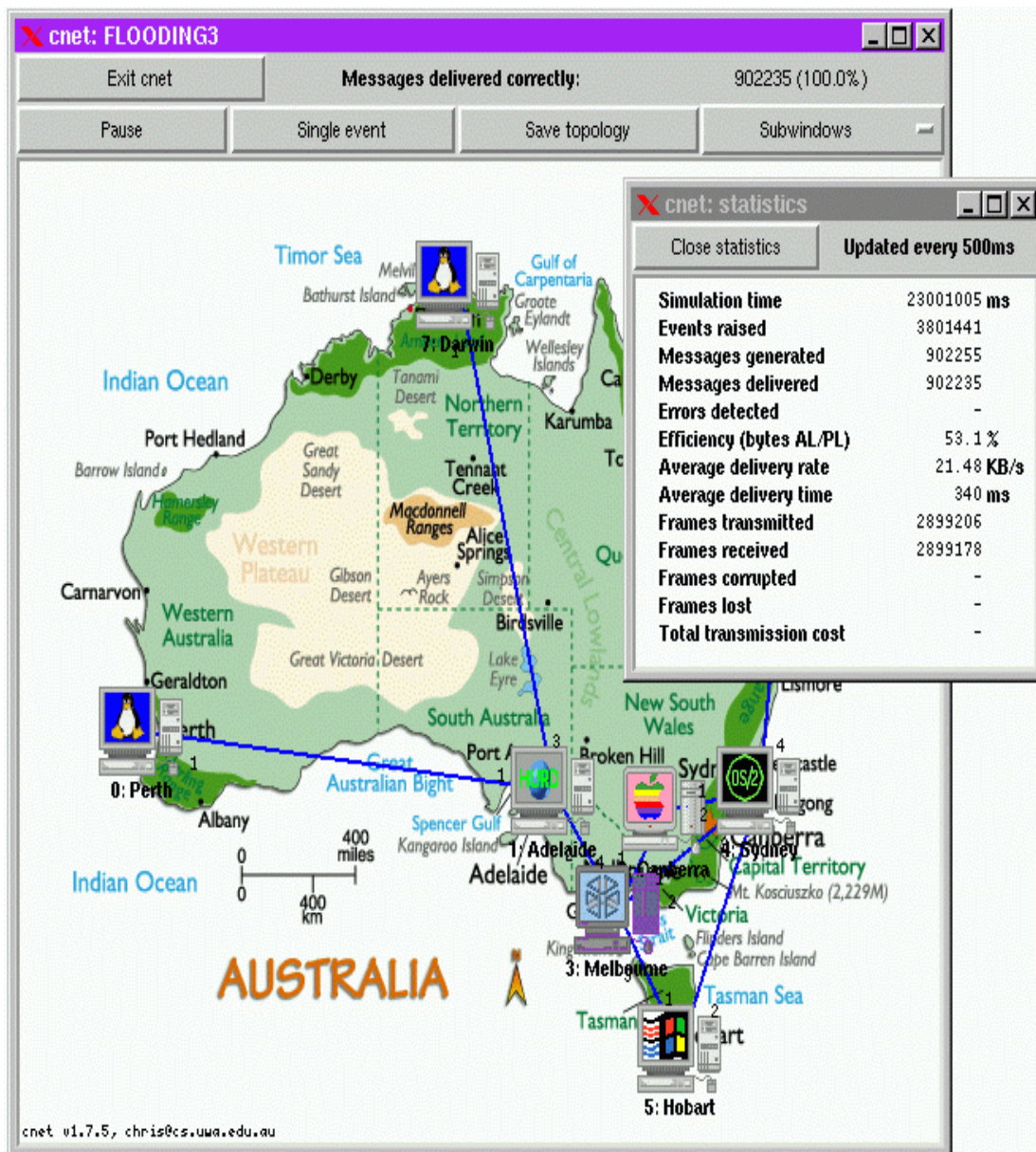
Το cnet μπορεί να παρουσιάσει μια περιορισμένη απεικόνιση των πλαισίων δεδομένων που διαπερνούν το φυσικό επίπεδο. Χρησιμοποίηση απλών ορθογωνίων, χρωμάτων και σύντομων string είναι δυνατό να επιδειχθούν τα δεδομένα και τα πλαίσια αναγνώρισης, καθώς και το περιεχόμενο μερικών από τους τομείς τους. Σε

συνδυασμό, αυτά τα χαρακτηριστικά γνωρίσματα μπορούν να χρησιμοποιηθούν για να διορθώσουν τις εφαρμογές του επιπέδου σύνδεσης δεδομένων. Παρακάτω, βλέπουμε σε ένα selective-repeat πρωτόκολλο την απώλεια

Πλαισίων (το άσπρο πλαίσιο δεδομένων αριθμός 5) και την καταστροφή των πλαισίων (το γκριζο πλαίσιο δεδομένων αριθμός 3), ενώ άλλα πλαίσια δεδομένων και αναγνώρισης ανταλλάσσονται επίσης.

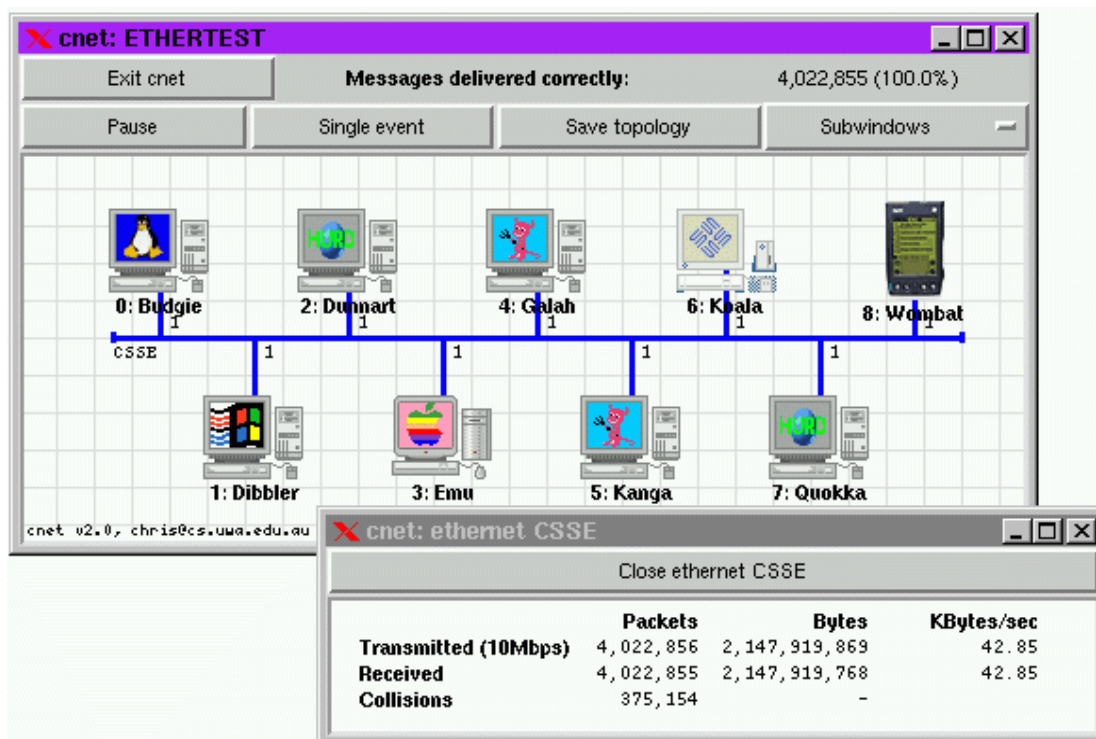


Το cnet απαιτεί τα πρωτόκολλα δικτύων να γράφονται στη γλώσσα προγραμματισμού Ansi-C και υποστηρίζει την εκτέλεσή τους μέσα σε μια Unix ή Linux διαδικασία. Ένας πρότυπος μεταγλωττιστής, κατά προτίμηση GCC, χρησιμοποιείται για να μεταγλωττίσει ο χρήστης τον γραπτό κώδικα του πρωτοκόλλου. Ο μεταγλωττισμένος κώδικας συνδέεται έπειτα δυναμικά στο χρόνο εκτέλεσης με τον προσομοιωτή cnet. Τα πρωτόκολλα καλούνται άμεσα από το cnet το ίδιο και δεν ερμηνεύονται. Από τη σχεδίαση, τα πρωτόκολλα δεν είναι απαραίτητο να περιέχουν οποιοδήποτε windowing κώδικα.

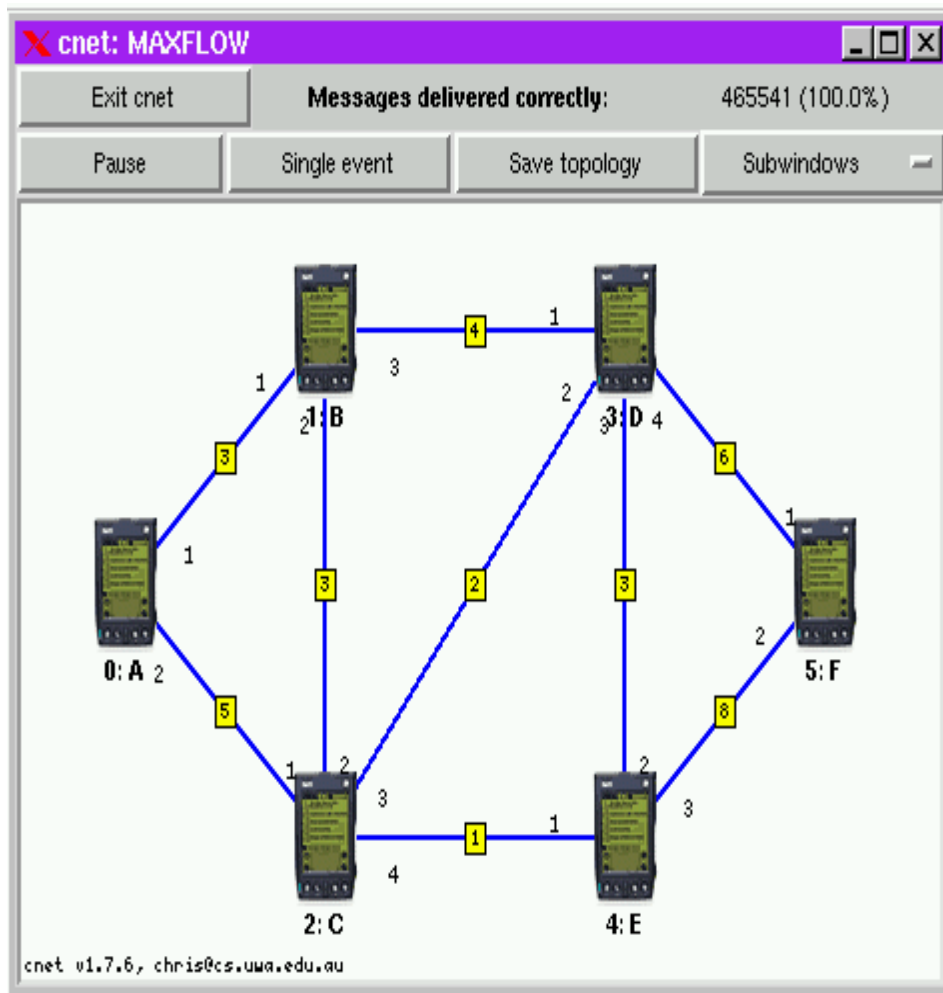


Όσον αφορά την έκδοση 2.0, το cnet υποστηρίζει το IEEE 802.3, τμήματα του Ethernet, καθώς επίσης και οι παραδοσιακές από σημείο σε σημείο WAN συνδέσεις. Κάθε τμήμα μνήμης αποτελείται από δύο ή περισσότερες κάρτες διεπαφής δικτύων (NICs) που συνδέουν τους κόμβους σταθμών ή δρομολογητών με το τμήμα του δικτύου. Τα μεγάλα δίκτυα μπορούν να κατασκευαστούν με την ένωση των πολλαπλάσιων τμημάτων του Ethernet μέσω κόμβων πυλών και συνδέσεων ευρείας ζώνης από σημείο σε σημείο. Η διεύθυνση του κάθε NIC καθορίζεται αρχικά στο αρχείο τοπολογίας, με τη δεκαεξαδική μορφή 6-οκτάδων παραδοσιακού Ethernet, και το cnet παρέχει τις λειτουργίες για να διαμορφώσει τη διεύθυνση κάθε NIC, και για να θέσει τα χαρακτηριστικά ενός NIC με ένα κατάλληλο τρόπο. Η προσομοίωση Ethernet

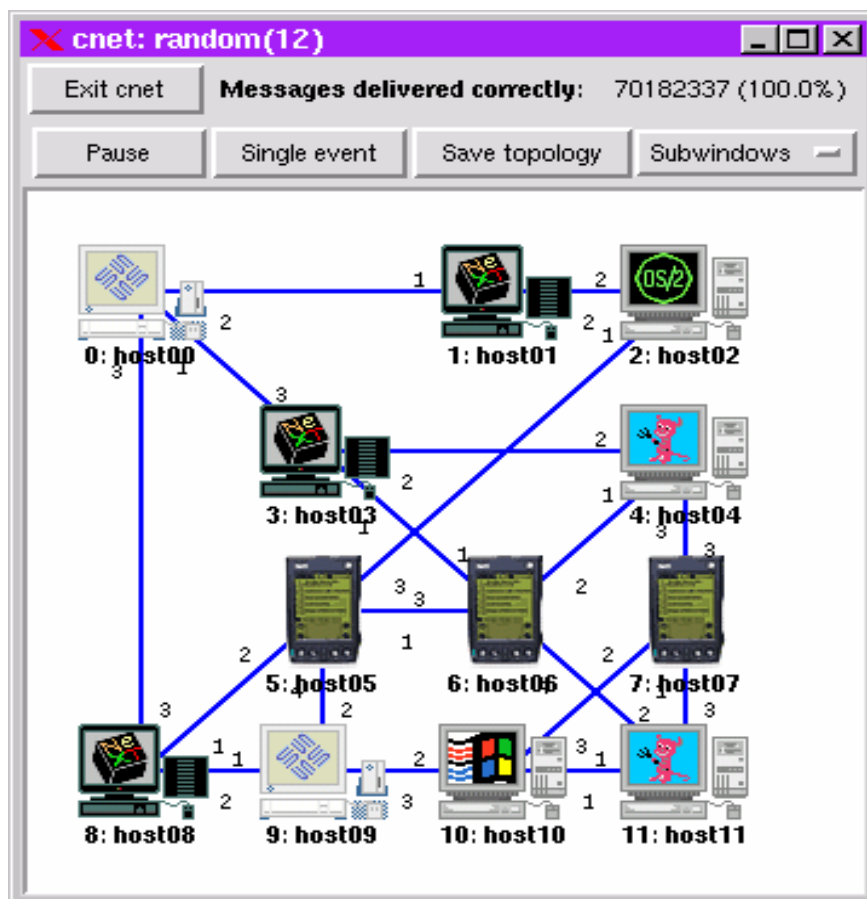
είναι λογική, αλλά όχι τελειοποιημένη. Το cnet υποστηρίζει ένα σταθερό ποσοστό μετάδοσης του 10Mbps, ένα slot-time με 52 χρήστες, broadcast μετάδοση, ανίχνευση σύγκρουσης, μπλοκάρισμα, και δυαδικό εκθετικό backoff, αλλά δεν υποστηρίζει έλεγχο jitter ούτε την multicasting διευθυνσιοδότηση.



Οι παρούσες τιμές των ιδιοτήτων των κόμβων και των συνδέσεων (όπως ενδεχομένως τροποποιήθηκαν μέσω της διαδικασίας windowing) είναι διαθέσιμες στον κώδικα του πρωτοκόλλου και κάθε κόμβος στις δομές δεδομένων και τις μεταβλητές της C. Αυτές οι δομές και οι μεταβλητές αρχικοποιούνται όταν κάθε κόμβος επανεκκινεί και ενημερώνεται έτσι ώστε κάθε κόμβος να σχεδιάζεται για εκτέλεση. Οι τιμές τους θεωρούνται ως μόνο ανάγνωσης και δεν μπορούν (επιτυχώς) να τροποποιηθούν άμεσα από τα πρωτόκολλα. Μπορούν, φυσικά, να τροποποιούνται μέσω της διαδικασίας windowing.



Οι δαπάνες ή τα βάρη μπορούν να ανατεθούν σε κάθε σύνδεση, για να αντιπροσωπεύσει το κόστος κάθε πλαισίου ή τον αριθμός των οκτάδων κάθε πλαισίου. Τα πρωτόκολλα μπορούν έπειτα να αναπτυχθούν για να παραδώσουν την κυκλοφορία δικτύων όπου κάθε ένα ελαχιστοποιεί το συνολικό κόστος, ή μεγιστοποιεί τη συνολική ροή.

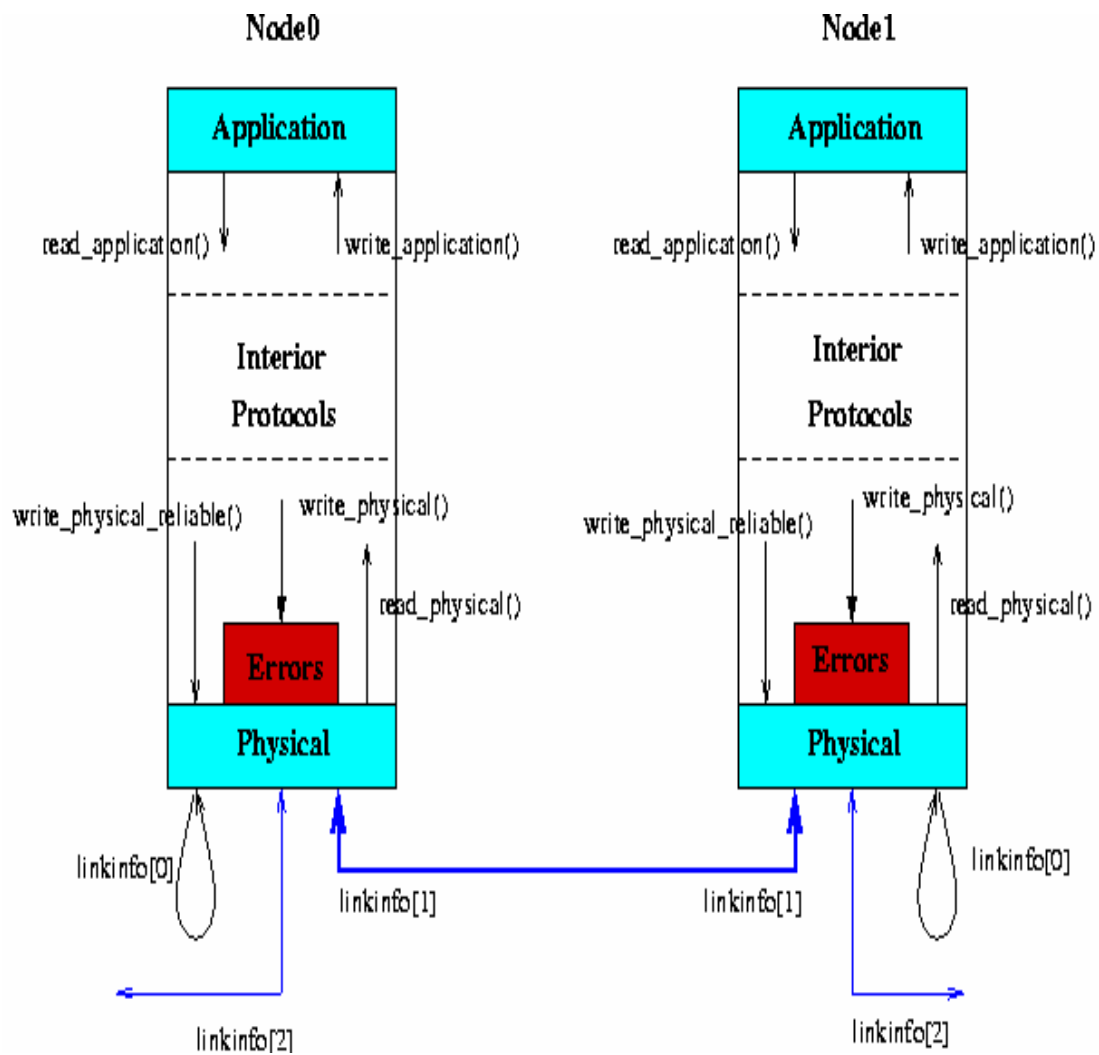


Και όταν θεωρείτε ότι έχετε την εργασία πρωτοκόλλων σας, τρέξτε την με την cnet – r προαιρετική δυνατότητα ώστε να παραχθεί μια τυχαία τοπολογία.

4.2 Το Μοντέλο Προσομοίωσης του CNET

Το cnet υποστηρίζει ένα μοντέλο δικτύωσης στο οποίο οι κόμβοι συνδέονται είτε από σημείο σε σημείο συνδέσεις είτε σε τμήματα Ethernet. Οι κόμβοι μπορούν να είναι είτε σταθμοί είτε δρομολογητές, αλλά τα εισαγωγικά πρωτόκολλα θα χρησιμοποιήσουν μόνο τους σταθμούς και τις από σημείο σε σημείο συνδέσεις. Οι σταθμοί είναι όπως τους τερματικούς σταθμούς, με κάθε έναν που έχει το επίπεδο εφαρμογών να παράγει τα μηνύματα για την παράδοση μέσω του επιπέδου εφαρμογών σε άλλους σταθμούς. Οι σταθμοί δεν παράγουν ποτέ τα μηνύματα για τους εαυτούς τους, και οι δρομολογητές δεν παράγουν μηνύματα καθόλου.

Το μοντέλο προσομοίωσης του cnet εμφανίζεται στο ακόλουθο διάγραμμα:



ΚΕΦΑΛΑΙΟ 5

GloMoSim

5.1 Περιγραφή του GloMoSim

Σε GloMoSim χτίζετε ένα εξελικτικό περιβάλλον προσομοίωσης για τα ασύρματα και συνδεδεμένα με καλώδιο συστήματα δικτύων. Σχεδιάζεται χρησιμοποιώντας την παράλληλη ικανότητα προσομοίωσης διακριτών γεγονότων που παρέχεται από την Parsec. Το GloMoSim υποστηρίζει αυτήν την περίοδο τα πρωτόκολλα για ένα καθαρώς ασύρματο δίκτυο. Στο μέλλον, υπάρχει η προσδοκία της λειτουργίας προσομοίωσης για συνδεδεμένο με καλώδιο καθώς επίσης και υβριδικό δίκτυο (με τις συνδεδεμένες με καλώδιο και ασύρματες ικανότητες).

Τα περισσότερα συστήματα δικτύων χτίζονται αυτήν την περίοδο χρησιμοποιώντας την προσέγγιση των επιπέδων που είναι παρόμοια με την OSI δικτυακή αρχιτεκτονική επτά επιπέδων. Το σχέδιο είναι να χτιστεί το GloMoSim χρησιμοποιώντας μια παρόμοια προσέγγιση βασιζόμενη σε επίπεδα. Πρότυπα API's θα χρησιμοποιηθούν μεταξύ των διαφορετικών επιπέδων προσομοίωσης. Αυτό θα επιτρέψει τη γρήγορη ολοκλήρωση των μοντέλων που αναπτύσσονται στα διαφορετικά επίπεδα από διαφορετικούς χρήστες. Τα πρωτόκολλα που στέλνονται με την παρούσα βιβλιοθήκη περιλαμβάνουν τα εξής :

Επίπεδα	Πρωτόκολλα
Mobility	Random waypoint, Random drunken, Trace based
Radio Propagation	Two ray and Free space
Radio Model	Noise Accumulating
Packet Reception Models	SNR bounded, BER based with BPSK/QPSK modulation
Data Link (MAC)	CSMA, IEEE 802.11 and MACA
Network (Routing)	IP with AODV, Bellman-Ford, DSR, Fisheye, LAR scheme 1, ODMRP, WRP
Transport	TCP and UDP
Application	CBR, FTP, HTTP and Telnet

Για να τρέξει το GloMoSim, θα χρειαστείτε τον πιο πρόσφατο μεταγλωττιστή της Parsec (που περιλαμβάνεται τώρα με τη διανομή του GloMoSim). Εάν θέλει κανείς να αναπτύξετε τα πρωτόκολλά σας σε GloMoSim, πρέπει να υπάρχει κάποια οικειότητα με την Parsec, αλλά δεν χρειάζεται να είναι εμπειρογνώμονας. Οι περισσότεροι υπεύθυνοι για την ανάπτυξη ενός πρωτοκόλλου θα πρέπει γράφουν καθαρά τον κώδικα σε C με μερικές Parsec λειτουργίες για τη χρονική διαχείριση. Ως εκ τούτου, θα πρέπει να χρησιμοποιήσετε τον Parsec μεταγλωττιστή. Ο Parsec κώδικας χρησιμοποιείται εκτενώς στον πυρήνα του GloMoSim. Οι περισσότεροι χρήστες δε χρειάζεται να ξέρουν πώς λειτουργεί ο πυρήνας. Εάν ενδιαφέρετε κάποιος για τον πυρήνα του GloMoSim, θα πρέπει να έχει εκτενή γνώση της Parsec. Το GloMoSim μπορεί ελεύθερα να το αποκτήσει κάποιος και να το εγκαταστήσει στον υπολογιστή του αρκεί να το κατεβάσει από το site <http://pcl.cs.ucla.edu/projects/glomosim/>.

5.2 Χτίζοντας ένα Δίκτυο με το GloMoSim

Η απλή προσέγγιση στο σχεδιασμό μιας προσομοίωσης δικτύων θα ήταν να αρχικοποιηθεί κάθε κόμβος του δικτύου στην προσομοίωση ως μια Parsec οντότητα. Μπορούμε να εμφανίσουμε τις διαφορετικές εκκινήσεις οντοτήτων ως χωριστές λογικές διαδικασίες στο σύστημα. Ως εκ τούτου κάθε εκκίνηση οντοτήτων απαιτεί το διάστημα στοιβών της στο χρόνο εκτέλεσης. Σε GloMoSim, προσπαθούμε να χτίσουμε μια προσομοίωση που θα επεκτείνεται σε χιλιάδες κόμβους. Εάν πρέπει να αρχικοποιήσουμε μια οντότητα για κάθε κόμβο στο χρόνο εκτέλεσης, οι απαιτήσεις μνήμης θα αυξάνονταν εντυπωσιακά. Η απόδοση του συστήματος θα έπεφτε επίσης γρήγορα. Δεδομένου ότι υπάρχουν τόσες πολλές οντότητες στην προσομοίωση, ο χρόνος εκτέλεσης θα χρειαζόταν συνεχώς στο διακόπτη πλαισίου μεταξύ των διαφορετικών οντοτήτων στο σύστημα. Αυτό θα προκαλέσει τη σημαντική υποβάθμιση κατά την εκτέλεση της προσομοίωσης. Ως εκ τούτου η αρχικοποίηση κάθε κόμβου ως χωριστή οντότητα θα περιορίσει εγγενώς την εξελισιμότητα και την απόδοση της προσομοίωσης.

Για να παρακαμφθούν αυτά τα προβλήματα στο χτίσιμο των δικτύων εισήχθη στην προσομοίωση η τεχνική του gridding των δικτύων. Με gridding δικτύων, μια ενιαία οντότητα μπορεί να προσομοιώσει διάφορους κόμβους δικτύων στο σύστημα. Μια χωριστή δομή δεδομένων που αντιπροσωπεύει την πλήρη κατάσταση κάθε κόμβου διατηρείται μέσα στην οντότητα. Ομοίως πρέπει να διατηρήσουμε το σωστό επίπεδο αφαιρέτηκότητας. Όταν ο κώδικας προσομοίωσης ενός ιδιαίτερου κόμβου εκτελείται δεν πρέπει να έχει πρόσβαση στις δομές δεδομένων των άλλων κόμβων στην προσομοίωση. Η gridding δικτύων τεχνική σημαίνει ότι μπορούμε να αυξήσουμε τον αριθμό κόμβων στο σύστημα διατηρώντας τον ίδιο αριθμό οντοτήτων στην

προσομοίωση. Στην πραγματικότητα, η μόνη απαίτηση είναι ότι χρειαζόμαστε μόνο τόσες οντότητες όσες ο αριθμός των επεξεργαστών στον οποίο η προσομοίωση εκτελείται. Ως εκ τούτου εάν τρέχουμε μια διαδοχική προσομοίωση πρέπει να

αρχικοποιήσουμε μόνο μια οντότητα στο σύστημα. Επίσης δεν συναντάμε τα προβλήματα μετατροπής μνήμης ή πλαισίου που περιορίζουν την προσομοίωση.

Στο GloMoSim, κάθε οντότητα αντιπροσωπεύει έναν γεωγραφικό τομέα της προσομοίωσης. Ως εκ τούτου οι κόμβοι των δικτύων που αντιπροσωπεύουν μια ιδιαίτερη οντότητα καθορίζονται από τη φυσική θέση των κόμβων.

ΚΕΦΑΛΑΙΟ 6

NIST Products

6.1 ATM/HFC Network Simulator

Ένας προσομοιωτής δικτύων ασύγχρονης μεταφοράς δεδομένων (ATM)έχει αναπτυχθεί για να παρέχει τα κατάλληλα μέσα στους ερευνητές και τους αρμόδιους για το σχεδιασμό δικτύων, ώστε να αναλύσουν τη συμπεριφορά των δικτύων ATM χωρίς τη δαπάνη της οικοδόμησης ενός πραγματικού δικτύου.

Ο προσομοιωτής χρησιμοποιείται ως εργαλείο προγραμματισμού δικτύων με τη προσομοίωση των διάφορων διαμορφώσεων του δικτύου και των φορτίων κυκλοφορίας, και λαμβάνοντας στατιστικά στοιχεία όπως η χρησιμοποίηση των συνδέσεων δικτύων και τα ποσοστά ρυθμό-απόδοσης εικονικών κυκλωμάτων. Μπορεί να χρησιμοποιηθεί για να απαντήσει σε ερωτήσεις όπως : πού είναι οι δυσκολίες στα δίκτυα; ποια είναι η επίδραση της αλλαγής της ταχύτητας μιας σύνδεσης; αν η προσθήκη μιας νέας εφαρμογής θα προκαλέσει συμφόρηση;

Ο προσομοιωτής μπορεί επίσης να χρησιμοποιηθεί ως εργαλείο ανάλυσης πρωτοκόλλου για να μελετήσει τη συνολική επίδραση συστημάτων ενός ιδιαίτερου πρωτοκόλλου δικτύων του ATM. Παραδείγματος χάριν, κάποιος μπορεί να ερευνήσει την αποτελεσματικότητα των διάφορων μηχανισμών ελέγχου ροής δεδομένων για τα δίκτυα ATM και να αντιμετωπίσει τέτοια ζητήματα όπως τους μηχανισμούς για τη δίκαιη κατανομή εύρους ζώνης, υπερχρησιμοποίηση πρωτοκόλλου, και χρησιμοποίηση εύρους ζώνης. Ο προσομοιωτής σχεδιάζεται κατά τέτοιο τρόπο ώστε τις ενότητες που προσομοιώνουν τα συστατικά ενός δικτύου ATM να μπορούν να αλλάξουν εύκολα, να προστίθενται ή να διαγράφονται..

Ο προσομοιωτής δικτύων NIST ATM δίνει στο χρήστη ένα διαλογικό περιβάλλον διαμόρφωσης με ένα γραφικό ενδιάμεσο με τον χρήστη. Η NIST έχει αναπτύξει το εργαλείο χρησιμοποιώντας και τη γλώσσα C και το σύστημα Windows X που τρέχει σε μια Unix βασισμένη πλατφόρμα. Το εργαλείο είναι βασισμένο σε έναν προσομοιωτή δικτύων που

αναπτύσσεται στο MIT και παρέχει υποστήριξη για τις ιδιαίτερες τεχνικές προσομοίωσης γεγονότος.

Το δίκτυο που προσομοιώνεται αποτελείται από διάφορα συστατικά που στέλνουν μηνύματα το ένα στο άλλο. Ο ευρυζωνικός τερματικός εξοπλισμός (B-TE) προσομοιώνει τους κόμβους και τις εφαρμογές σταθμού του ATM. Οι διακόπτες και τα συστατικά B-TE διασυνδέονται με τις φυσικές συνδέσεις, οι οποίες θεωρούνται επίσης συστατικά. Οι εφαρμογές του ATM είναι λογικές οντότητες που τρέχουν στους σταθμούς B-TE και μπορούν να θεωρηθούν ως γεννήτριες κυκλοφορίας ικανές για μεταβλητές ή σταθερές πηγές κυκλοφορίας ποσοστού δυαδικών ψηφίων.

Για να εκτελέσει μια προσομοίωση, ο χρήστης δημιουργεί ένα δίκτυο που αποτελείται από οποιοδήποτε αριθμό διακοπών και σταθμών του ATM που διασυνδέονται από τις συνδέσεις

των μεταβλητών ταχυτήτων. Οι εικονικές συνδέσεις μεταξύ οποιωνδήποτε δύο σταθμών καθιερώνονται χειροκίνητα, και οι εφαρμογές στους σταθμούς επιλέγονται και αρχίζουν. Ενώ μια προσομοίωση τρέχει, τα διάφορα στιγμιαία μέτρα απόδοσης μπορούν να παρουσιαστούν σε γραφική μορφή κειμένων στην οθόνη ή να σωθούν σε αρχείο για την επόμενη ανάλυση.

Το εργαλείο αυτό είναι διαθέσιμο μέσω του internet και στην ιστοσελίδα http://w3.antd.nist.gov/Hsntg/prd_atm-sim.html.

6.2 Private Network-Network Interface (PNNI) Simulator

Ο Private Network-Network Interface (PNNI) προσομοιωτής αναπτύχθηκε στο εθνικό ίδρυμα προτύπων και τεχνολογίας (NIST) για να μελετήσει την απόδοση των πρωτοκόλλων δρομολόγησης PNNI σε ένα περιβάλλον δικτύων πολλών κόμβων.

Ο προσομοιωτής PNNI γράφεται σε C. Ο κώδικας αποτελείται από δύο μέρη: τη διεπαφή ενδιάμεσο με τον χρήστη και τα πρωτόκολλα δρομολόγησης PNNI. Η διεπαφή με τον χρήστη επιτρέπει στο χρήστη για να εισαγάγει τις διάφορες φυσικές τοπολογίες δικτύων και τις παραμέτρους του PNNI. Μερικές από τις παραμέτρους τοπολογίας δικτύων είναι αριθμός διακοπών PNNI ATM, ο αριθμός φυσικών συνδέσεων και καθυστέρησής τους. Μερικές από τις παραμέτρους PNNI είναι η όμοια ομάδα, η όμοια προτεραιότητα ηγεσίας ομάδας, οι τιμές χρονομέτρων, και οι όμοιες ομάδες πύο υψηλού επιπέδου. Τα πρωτόκολλα δρομολόγησης PNNI εκτελούν τις απαραίτητες λειτουργίες των μεμονωμένων πρωτοκόλλων όπως προσδιορίζονται στην ιδιωτική έκδοση προδιαγραφών στο φόρουμ του ATM Private Network-Network Interface Specification Version 1.0.

Τα αποτελέσματα για οποιοδήποτε ιδιαίτερο τρέξιμο προσομοίωσης παρέχουν το χρόνο και τα στοιχεία που απαιτούνται για να φθάσουν σε μία σταθερή κατάσταση για τη δεδομένη τοπολογία δικτύων. Η έξοδος μπορεί επίσης να παρέχει το ποσό των δεδομένων που απαιτείται για να διατηρήσει αυτή τη σταθερή κατάσταση. Με τη μελέτη των πολλαπλάσιων τρεξιμάτων της προσομοίωσης με την ποικιλία των παραμέτρων δρομολόγησης PNNI, μπορεί κάποιος να καθορίσει μια βέλτιστη διαμόρφωση για μια δεδομένη τοπολογία δικτύων.

Το εργαλείο αυτό είναι διαθέσιμο μέσω του internet και στην ιστοσελίδα http://w3.antd.nist.gov/Hsntg/prd_aprops.html.

ΚΕΦΑΛΑΙΟ 7

Network Simulator 2 (NS-2)

7.1 Εισαγωγή

Το περισσότερο διαδεδομένο εργαλείο προσομοίωσης είναι κατά πολλούς το Network Simulator 2 ή αλλιώς NS-2. Το NS-2 είναι ένα open source εργαλείο προσομοίωσης που τρέχει κυρίως σε περιβάλλον Linux. Είναι ένας προσομοιωτής διακριτών γεγονότων που στοχεύει στην έρευνα δικτύωσης. Παρέχει ουσιαστική υποστήριξη για την προσομοίωση της δρομολόγησης, των multicast πρωτοκόλλων και πρωτοκόλλων IP, όπως τα UDP, TCP, RTP και SRM πάνω από συνδεδεμένα με καλώδιο αλλά και ασύρματα (τοπικά και δορυφορικά) δίκτυα. Το NS-2 έχει πολλά πλεονεκτήματα που το κατατάσσουν ως ένα χρήσιμο εργαλείο για την υποστήριξη πολλαπλών πρωτοκόλλων και τη δυνατότητα της γραφικής λεπτομερειακής απεικόνισης της δικτυακής κίνησης. Επιπλέον υποστηρίζει αρκετούς αλγόριθμους που αφορούν στη δρομολόγηση και στις ουρές αναμονής.

Το NS άρχισε ως παραλλαγή του REAL network simulator το 1989 και έχει εξελιχθεί σημαντικά τα τελευταία χρόνια. Το 1995 η ανάπτυξη του NS υποστηρίχθηκε από την DAPRA μέσω του VINT project στις LBL, Xerox PARC, UCB, και USC/ISI. Αυτή την περίοδο η ανάπτυξη του NS υποστηρίζεται μέσω της DAPRA με την SAMAN και μέσω της NSF με την CONSER, και οι δύο σε συνεργασία με άλλους ερευνητές συμπεριλαμβανομένου της ACIRI. Το NS λαμβάνει συνεχώς σημαντική βοήθεια και από άλλους ερευνητές, συμπεριλαμβανομένου του ασύρματου κώδικα από τα UCB Daedalus και CMU Monarch projects και από την Sun Microsystems.

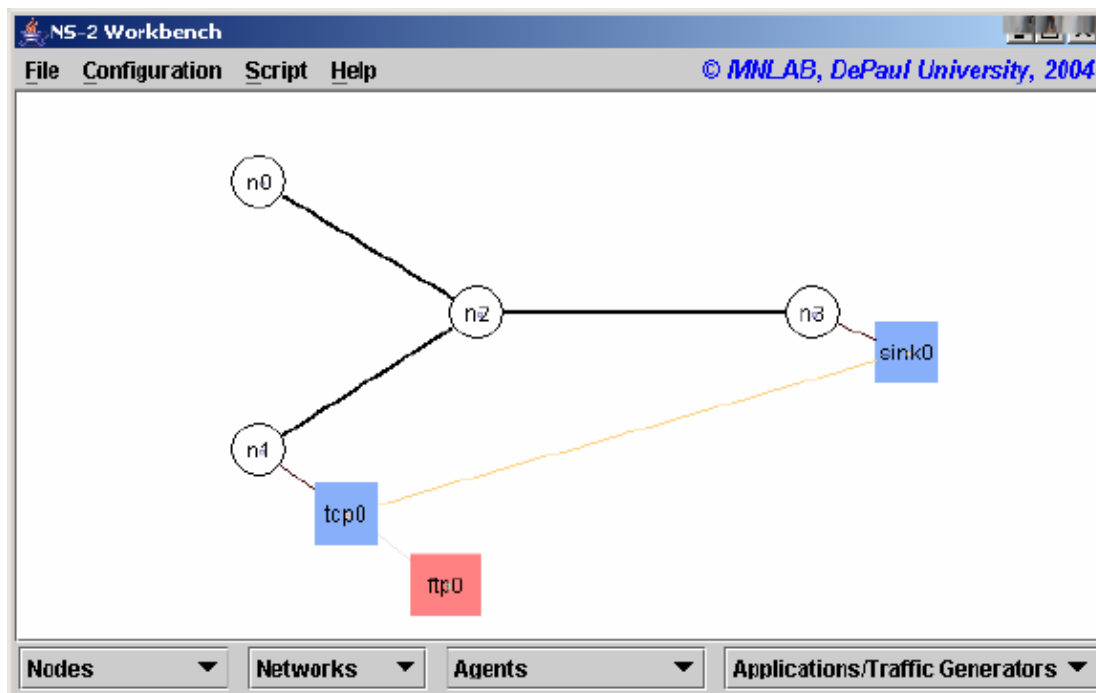
7.2 Εγκαθιστώντας το NS-2

Το NS-2 είναι διαθέσιμο σε αρκετές πλατφόρμες όπως είναι οι : FreeBSD, Linux, SunOS και Solaris. Το NS-2 επίσης μπορεί να εγκατασταθεί και να τρέξει σε περιβάλλον Windows. Τα απλά σενάρια θα πρέπει λογικά να τρέχουν σε οποιοδήποτε απλό μηχάνημα, παρόλα αυτά όμως τα πολύ μεγάλα σενάρια απαιτούν και επωφελούνται από την ύπαρξη μεγάλης κεντρικής μνήμης από το υπολογιστικό σύστημα. Επιπλέον το NS-2 χρειάζεται τα ακόλουθα πακέτα να είναι εγκατεστημένα επίσης ώστε να μπορεί να τρέξει : Tcl release 8.3.2, Tk release 8.3.2, OTcl release 1.0a7 and TclCL release 1.0b11.

Το NS-2 είναι διαθέσιμο τόσο μέσα από HTTP σελίδες (<http://www.isi.edu/nsnam/ns/>), όσο και σε anonymous FTP στο <ftp://ftp.isi.edu/nsnam>, αλλά και σε anonymous CVS στην *anon-cvs page*. Το ftp για το NS-2 υπάρχει αντίστοιχα και στο ευρωπαϊκό mirror <ftp://ftp.ee.surrey.ac.uk/pub/Mirrors/ftp.isi.edu/nsnam>

Ενώ υπάρχει ιδιαίτερη εμπιστοσύνη στο NS, το NS δεν είναι ένα καλογουαλισμένο και ολοκληρωμένο προϊόν, αλλά το αποτέλεσμα μιας συνεχούς προσπάθειας έρευνας και ανάπτυξης. Ειδικότερα, λάθη στο λογισμικό του προγράμματος ακόμα ανακαλύπτονται και διορθώνονται. Οι χρήστες του NS είναι αρμόδιοι για να επαληθεύσουν ότι οι προσομοιώσεις τους δεν καταλήγουν να είναι άκυρες λόγω των λαθών. Για να βοηθηθεί ο χρήστης σε αυτό το πρόβλημα καταβάλλεται μεγάλη προσπάθεια με τη επέκταση και την αυτοματοποίηση των προγραμμάτων ελέγχου της αξιοπιστίας των παραγόμενων αποτελεσμάτων.

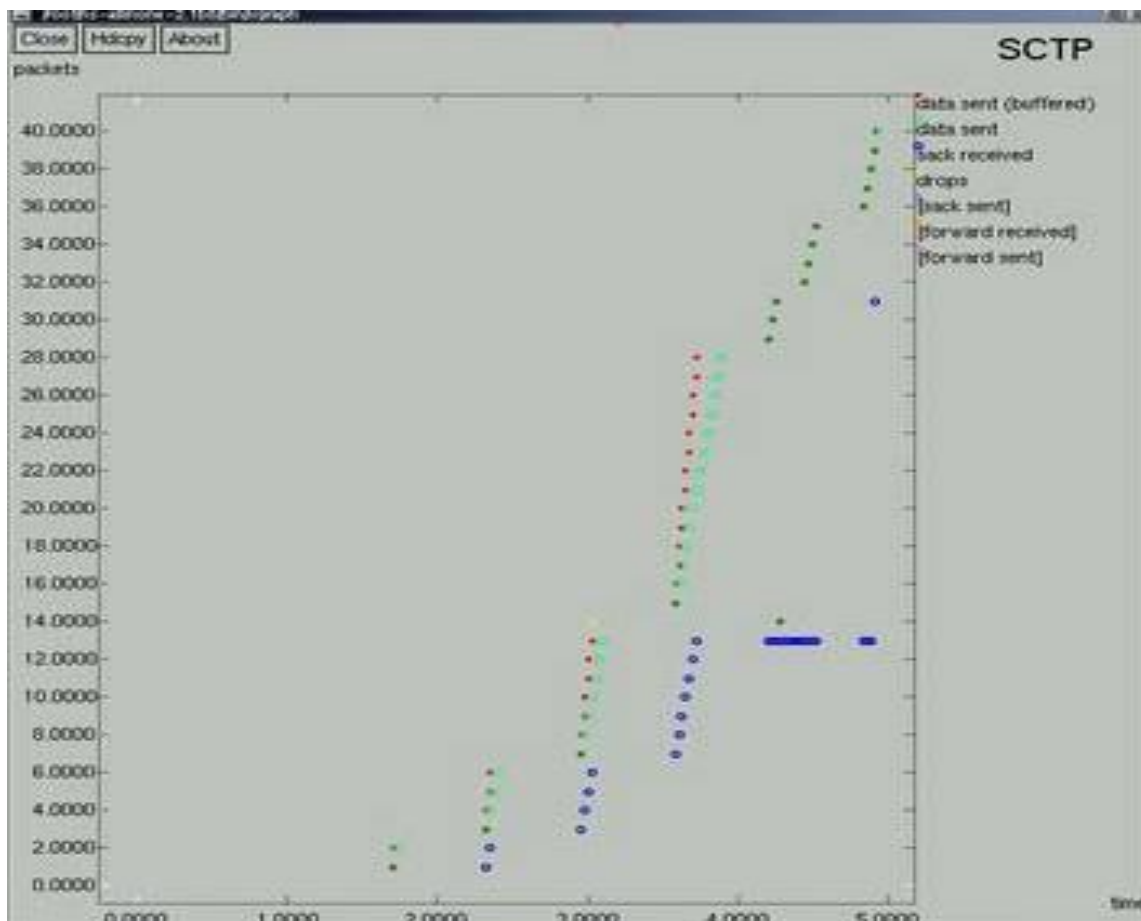
Ομοίως, οι χρήστες είναι αρμόδιοι για να επαληθεύσουν ότι οι προσομοιώσεις τους δεν ακυρώνονται επειδή το πρότυπο που εφαρμόζεται στον προσομοιωτή δεν είναι το πρότυπο που ανέμεναν. Το τρέχον εγχειρίδιο του NS θα πρέπει να τους βοηθήσει σε αυτήν την διαδικασία.



7.3 Περιγραφή και Δυνατότητες του NS-2

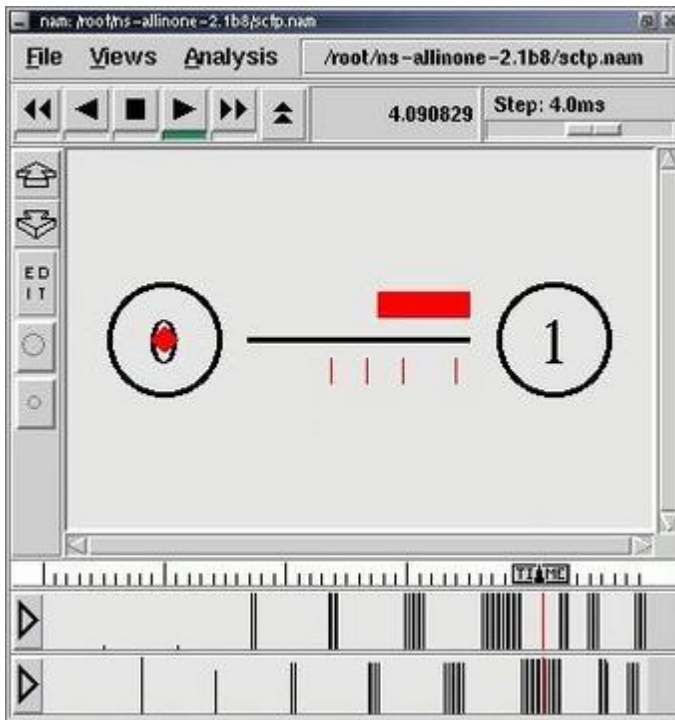
Ο NS-2 είναι ένας αντικειμενοστραφής προσομοιωτής (*Object Oriented Simulator*), ο οποίος χρησιμοποιεί στο προσκήνιο την γλώσσα προγραμματισμού OTcl (*Object Tool Command Language*) και στο παρασκήνιο την C++. Ο προσομοιωτής διαθέτει μια ιεραρχία τάξεων σε C++ (που αναφέρεται ως *compiled hierarchy*) και μια παρόμοια ιεραρχία στην OTcl (που αναφέρεται ως *interpreted hierarchy*, μιας και η Tcl διαθέτει interpreter και όχι compiler). Από την πλευρά του χρήστη υπάρχει μια αντιστοιχία 1-1 ανάμεσα στις τάξεις που ανήκουν στην *compiled hierarchy* (C++) και σε εκείνες που ανήκουν στην *interpreted hierarchy*

(OTcl). Η ρίζα σε αυτή την ιεραρχία είναι η τάξη TclObject. Οι χρήστες μπορούν να δημιουργούν νέα αντικείμενα προσομοίωσης μέσα από τον OTcl interpreter. Η αρχικοποίηση αυτών των αντικειμένων γίνεται από τον OTcl interpreter, κατά την οποία αντιστοιχίζονται και με το ανάλογο αντικείμενο που υπάρχει στην class hierarchy (C++).



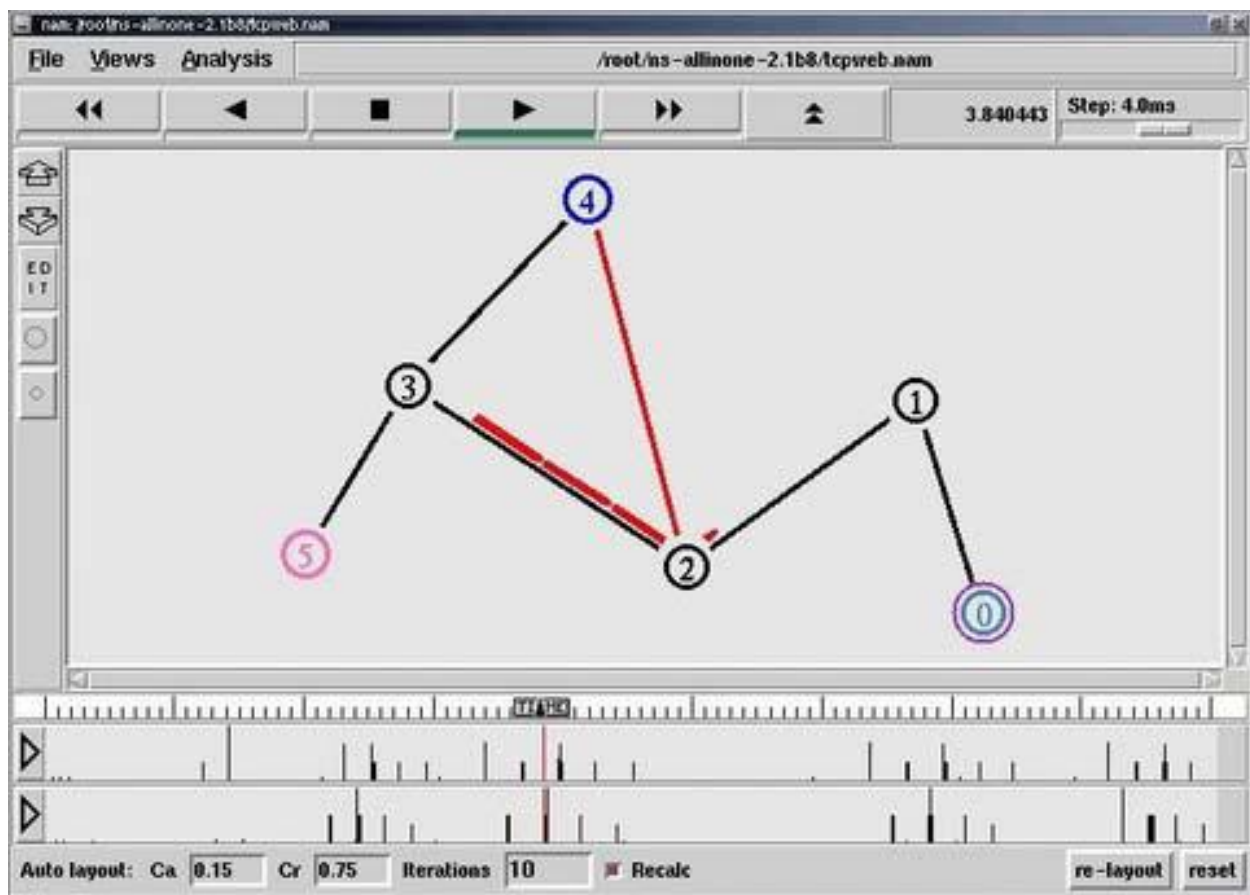
Η φιλοσοφία που κρύβεται πίσω από τη χρήση δύο γλωσσών προγραμματισμού και δύο παρόμοιων ιεραρχιών τάξεων, είναι ότι κατά τη διάρκεια μιας προσομοίωσης υπάρχουν δύο διαφορετικά πράγματα που πρέπει να εκτελεστούν από τον προσομοιωτή. Από τη μια πλευρά, η λεπτομερής προσομοίωση πρωτοκόλλων και δικτύων απαιτεί μια γλώσσα προγραμματισμού, η οποία να μπορεί να διαχειρίζεται με αποδοτικό τρόπο bytes, επικεφαλίδες πακέτων (packet headers), καθώς και να τρέχει αλγόριθμους, οι οποίοι να διαχειρίζονται μεγάλα σύνολα δεδομένων. Για αυτές τις ενέργειες, ο χρόνος εκτέλεσης (run-time) αποτελεί το σημαντικότερο στοιχείο, ενώ ο χρόνος δόρθωσης & επαναπροσδιορισμού της προσομοίωσης (turn-around time), ο οποίος περιλαμβάνει τη διαδικασία εύρεσης και δόρθωσης λαθών στον κώδικα και επανάληψη της μεταγλώττισης και της εκτέλεσης, είναι λιγότερο σημαντικός.

Από την άλλη ένα μεγάλο κομμάτι της έρευνας σε δίκτυα υπολογιστών περιλαμβάνει τις διαρκείς δοκιμές και εκτελέσεις της προσομοίωσης μεταβάλλοντας ίσως κατά πολύ λίγο ορισμένες παραμέτρους (όπως π.χ. την ταχύτητα των ζεύξεων, τον ρυθμό λαθών στο κανάλι, τον αριθμό των κόμβων, κλπ), έτσι ώστε να εξερευνηθεί ένας όσο το δυνατόν μεγαλύτερος αριθμός «σεναρίων» σχετικά με την εξεταζόμενη δικτυακή τοπολογία. Για τις ενέργειες αυτές, ο χρόνος κατά τον οποίο μεταβάλλουμε τις παραμέτρους *ρύθμισης* της προσομοίωσης είναι πολύ σημαντικός και η interpreted γλώσσα προγραμματισμού Tcl αποτελεί μια πολύ καλή λύση, μιας και η ρύθμιση μιας δικτυακής τοπολογίας γίνεται στην αρχή και μετά παραμένει η ίδια σε όλη τη διάρκεια της προσομοίωσης.



Αν και θα μπορούσαμε να υλοποιήσουμε τα πάντα σε όποια γλώσσα προγραμματισμού θέλουμε από τις δύο (αν και σε C++ είναι λίγο δυσκολότερο μιας και στο προσκήνιο χρησιμοποιείται ο OTcl interpreter), καλό είναι την OTcl να τη χρησιμοποιούμε κυρίως για να ρυθμίζουμε τη δικτυακή τοπολογία μας και τις διάφορες παραμέτρους της, ενώ τη C++ για να υλοποιούμε πολύπλοκους και απαιτητικούς αλγόριθμους δρομολόγησης, ελέγχου λαθών, πρωτοκόλλων εφαρμογών, δικτύου, κλπ. Ωστόσο, αξίζει να σημειωθεί ότι ο NS-2 παρέχει τη δυνατότητα να «σπάμε» την υλοποίηση ενός αντικειμένου ανάμεσα στις 2 γλώσσες, με μερικές από τις ιδιότητες και μεθόδους να βρίσκονται στο OTcl αντικείμενο (ίσως αυτές της ρύθμισης της προσομοίωσης) και τις περισσότερο πολύπλοκες στο αντίστοιχο C++ αντικείμενο που ανήκει στην compiled hierarchy. Στη συνέχεια, μπορούμε (μέσα από κατάλληλους μηχανισμούς που παρέχει ο NS-2), να **καλούμε** μια μέθοδο του αντικειμένου που είναι γραμμένη σε C++, **μέσα** από το OTcl αντικείμενο, ελέγχοντας και εκτελώντας την προσομοίωσή μας κατευθείαν από την OTcl, η οποία άλλωστε χρησιμοποιείται και στο προσκήνιο από τον NS-2. Ο NS-2 είναι ένας Object-Oriented Simulator και κατά συνέπεια τα πάντα γίνονται με τη χρήση αντικειμένων. Ο NS διαθέτει ένα ευρύ σύνολο αντικειμένων για τη δημιουργία και τη διαχείριση των προσομοιώσεών μας. Το κάθε αντικείμενο διαθέτει ορισμένες **ιδιότητες** (χαρακτηριστικά, τα οποία αντιπροσωπεύονται ως **μεταβλητές** που είναι ενσωματωμένες στο αντικείμενο) και **μεθόδους** (δυνατότητες εκτέλεσης ορισμένων εργασιών από το αντικείμενο, οι οποίες αντιπροσωπεύονται ως **διαδικασίες** που είναι ενσωματωμένες κι αυτές στο αντικείμενο).

Κάθε προσομοίωση που δημιουργούμε στον NS-2, εξαρτάται από ένα κεντρικό αντικείμενο, το *αντικείμενο ελέγχου της προσομοίωσης Simulator*. Με τις ιδιότητες και τις μεθόδους του αντικειμένου αυτού μπορούμε να ελέγχουμε την προσομοίωσή μας, ρυθμίζοντας την τοπολογία της, τα πρωτόκολλα που θα «τρέχουν» οι κόμβοι του δικτύου, τις συνδέσεις ανάμεσα στους κόμβους, το πρωτόκολλο δρομολόγησης που θα χρησιμοποιεί η προσομοίωση, τη *δυναμικότητα* του δικτύου (απενεργοποίηση & ενεργοποίηση συνδέσεων και κόμβων), κλπ. Το αντικείμενο Simulator δημιουργείται στην αρχή και το χρησιμοποιούμε σε όλη τη διάρκεια της προσομοίωσης.



Αφού δημιουργήσουμε το αντικείμενο ελέγχου της προσομοίωσης, σειρά έχει η δημιουργία της δικτυακής μας τοπολογίας, η οποία ξεκινά με τη δημιουργία των *κόμβων*. Ένας κόμβος μπορεί να αντιπροσωπεύει τον υπολογιστή ενός χρήστη, έναν δρομολογητή, ή ακόμη και ένα ολόκληρο εργαστήριο υπολογιστών, ανάλογα ίσως και με το ρυθμό με τον οποίο μεταδίδει δεδομένα. Επίσης, μπορεί να αντιπροσωπεύει και έναν ασύρματο κόμβο (κινητό ή ακίνητο), ή έναν δορυφόρο, κλπ, ανάλογα πάντα με το τι θέλουμε να προσομοιώσουμε.

Αφού δημιουργήσουμε τους κόμβους και τις συνδέσεις τους, θα πρέπει να δημιουργήσουμε και κάποια κυκλοφορία δεδομένων ανάμεσά τους. Αυτό απαιτεί δύο πράγματα. Καταρχήν θα πρέπει να εξοπλίσουμε τους **κόμβους** μας με τα κατάλληλα **πρωτόκολλα** μετάδοσης δεδομένων, όπως είναι το TCP, το UDP, το SCTP, κλπ. Ωστόσο, επειδή αυτά τα πρωτόκολλα δεν είναι ικανά να αποστέλλουν και να λαμβάνουν δεδομένα από μόνα τους, χρειάζεται να προσδιορίσουμε και κάποιες **εφαρμογές** που να τα χρησιμοποιούν (π.χ. μια εφαρμογή FTP).

Όπως προαναφέραμε, θα πρέπει να δημιουργήσουμε και τις κατάλληλες **εφαρμογές** που θα χρησιμοποιούν τα πρωτόκολλα αυτά για την αποστολή και λήψη των δεδομένων τους. Υπάρχουν διαφορετικές εφαρμογές που χρησιμοποιούν το πρωτόκολλο TCP και άλλες που χρησιμοποιούν το πρωτόκολλο UDP. Ένα παράδειγμα εφαρμογής για το TCP πρωτόκολλο είναι η μεταφορά αρχείων FTP Application, ενώ παραδείγματα εφαρμογών για το UDP πρωτόκολλο είναι : μία εφαρμογή που να αποστέλλει δεδομένα με σταθερό ρυθμό (Constant Bit Rate – CBR Application) και μία που αποστέλλει δεδομένα σε τυχαίες χρονικές στιγμές (exponential application).

Με τον NS-2 έχουμε τη δυνατότητα να εισάγουμε μοντέλα δυναμικής διαχείρισης της δικτυακής τοπολογίας μας. Τα μοντέλα αυτά έχουν να κάνουν με την ενεργοποίηση και την απενεργοποίηση των ζεύξεων μεταξύ των κόμβων του δικτύου, έτσι ώστε να προσομοιωθούν περιπτώσεις όπου μια ζεύξη ή ένας κόμβος είναι ελαττωματικός και προκαλεί κατά συνέπεια απώλεια δεδομένων.

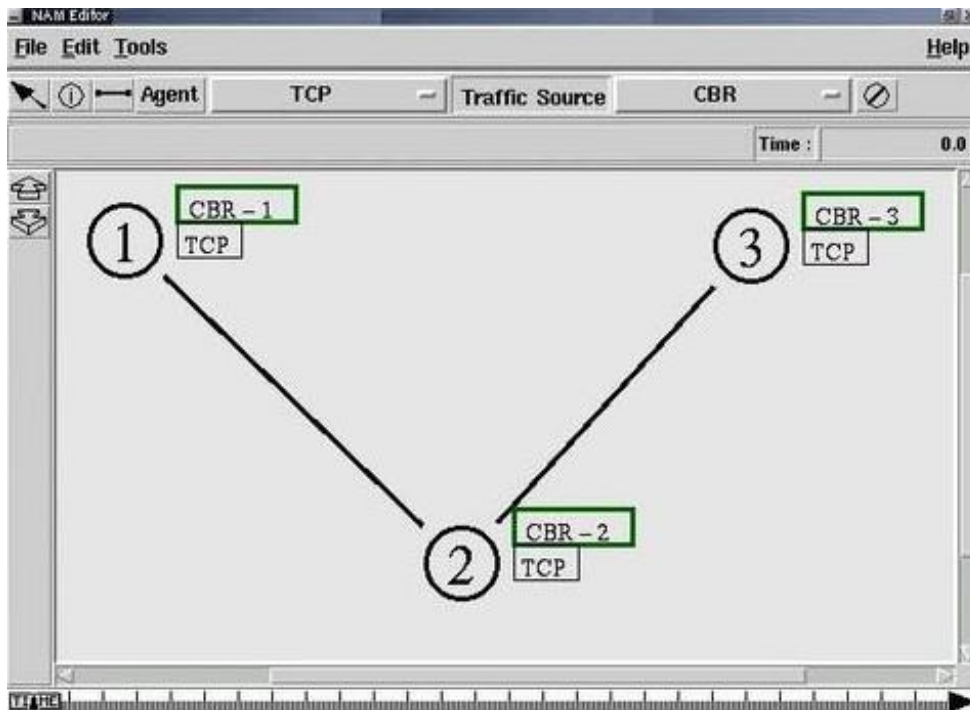
Επιπλέον με τον NS-2 έχουμε τη δυνατότητα να **παρακολουθούμε** μία ή περισσότερες ουρές της δικτυακής μας τοπολογίας. Αυτό γίνεται με τη χρήση των αντικειμένων τύπου **Monitor**, τα οποία μπορούν να συσχετίζονται με τις ουρές της δικτυακής μας τοπολογίας και να καταγράφουν την κίνηση σε αυτές μέσα στις ιδιότητές τους. Στο τέλος της προσομοίωσης (ή όποτε άλλοτε θέλουμε) μπορούμε απλά να **ανακτήσουμε** τις τιμές αυτών των ιδιοτήτων και είτε να τις αποθηκεύσουμε σε αρχεία, ή να τις εμφανίσουμε στην οθόνη.

Ο προσομοιωτής NS-2 συνοδεύεται από ένα βοηθητικό εργαλείο οπτικής απεικόνισης της εκτέλεσης της προσομοίωσής μας, τον **Network Animator (NAM)**. Όταν απεικονίζουμε την προσομοίωσή του NS-2 σε NAM μπορούμε να χρωματίσουμε με διαφορετικό χρώμα τις ροές των Agents πρωτοκόλλου, έτσι ώστε να μπορούμε να τις ξεχωρίσουμε και να παρακολουθούμε με μεγαλύτερη ευκρίνεια τη ροή των αποτελεσμάτων.

7.4 Network Animator (NAM)

Το NAM είναι ένα πρόγραμμα απεικόνισης βασισμένο στην γλώσσα προγραμματισμού Tcl/Tk για την παρακολούθηση αρχείων δικτυακών προσομοιώσεων που έχουν δημιουργηθεί από εργαλεία προσομοίωσης όπως είναι το NS-2. Υποστηρίζει την σχεδίαση τοπολογιών, την απεικόνιση του επιπέδου των πακέτων και πολλά εργαλεία παρακολούθησης των δεδομένων. Το NAM ξεκίνησε στην LBL και εξελίχθηκε σημαντικά τα τελευταία χρόνια. Έτσι λοιπόν αν κανείς προτιμά ένα γραφικό περιβάλλον στο οποίο θα μπορεί να υλοποιεί τις προσομοιώσεις που δημιουργεί μέσω του NS-2 το NAM υποστηρίζει ένα drag and drop περιβάλλον χρήστη. Σε αυτό μπορούν να τοποθετηθούν οι δικτυακοί κόμβοι, να ενωθούν μεταξύ

τους και στη συνέχεια να οριστούν οι παράγοντες που επιθυμεί ο χρήστης καθώς και η αντίστοιχη εφαρμογή ή η γεννήτρια κίνησης. Στην εικόνα που ακολουθεί φαίνεται ένα παράδειγμα μιας απλής χρήσης του NAM. Για να υλοποιηθεί την τοπολογία που φαίνεται στην παρακάτω εικόνα θα πρέπει να κάνουμε τα εξής :



NAM Editor

Στην μπάρα εργαλείων κάνουμε κλικ στο κουμπί Add node και τοποθετούμε τρεις κόμβους στο παράθυρο του προγράμματος κάνοντας δεξί κλικ στις σωστές θέσεις. Για να ενώσουμε τους κόμβους κάνουμε κλικ στο Add link. Επιλέγουμε τον έναν κόμβο και κρατώντας πατημένο το κουμπί του ποντικιού επιλέγουμε τον άλλο κόμβο για να δημιουργήσουμε έναν σύνδεσμο. Στη συνέχεια ποια μέσα θέλουμε να χρησιμοποιήσουμε στο δίκτυο από μενού των agents. Για να προστεθεί το μέσο θα πρέπει να κάνουμε κλικ στον κατάλληλο κόμβο. Τέλος επιλέγουμε ποια εφαρμογή θέλουμε να προσομοιώσουμε, δηλαδή ή FTP ή CBR πηγή. Για να προσθέσουμε την εφαρμογή θα πρέπει να κάνουμε κλικ στο επιλεγμένο μέσο. Σε αυτό το σημείο με δεξί κλικ στα διάφορα στοιχεία της τοπολογίας μπορούμε να διαμορφώσουμε τις ιδιότητές τους, όπως το χρώμα ή ο χρόνος έναρξης και λήξης της εφαρμογής. Θα πρέπει να σημειωθεί ότι η προσομοίωση πρέπει πρώτα να μεταγλωττιστεί από το NS-2 και στη συνέχεια να τρέξει στο περιβάλλον του NAM.

ΚΕΦΑΛΑΙΟ 8

Η ΓΛΩΣΣΑ NED

Ακολουθεί μια περιγραφή της γλώσσας NED και των κύριων χαρακτηριστικών της. Ο χρήστης περιγράφει την δομή του μοντέλου της προσομοίωσης με την γλώσσα NED (NEtwork Description). Μας επιτρέπει να δηλώσουμε `simple modules`, να τα συνδέσουμε και να τα συναρμολογήσουμε σχηματίζοντας `compound modules`.

8.1 Simple modules

Όπως ξέρουμε τα `simple modules` είναι τα ενεργά συστατικά μέρη του μοντέλου και τα ορίζουμε με την δεσμευμένη λέξη `simple`. Ακολουθεί ένα παράδειγμα δήλωσης `Simple module`:

```
simple Module_name
{
parameters:
int x;
@display("i=block/queue)
gates:
input in;
output out;
}
```

Και ο τομέας `parameters` και ο τομέας `gates` είναι προαιρετικός. Είναι πολύ σημαντικό να προσέξουμε ότι η δήλωση ενός `module` στην γλώσσα NED (η οποία περιέχεται σε ένα αρχείο με κατάληξη `.ned`) δεν περιέχει καθόλου κώδικα για να ορίσει την λειτουργία του `module`. Αυτός ο κώδικας περιέχεται σε μια κλάση C++.

Αρχικά το OMNeT++ ψάχνει να εντοπίσει μια κλάση με το ίδιο όνομα όπως ο τύπος NED που δηλώσαμε (στο παράδειγμα παραπάνω ως: «`Module_name`»). Μπορούμε

όμως να δηλώσουμε ρητά την κλάση η οποία περιέχει τον κώδικα του module με την ιδιότητα @class.

Επίσης μπορούμε να επεκτείνουμε τις δυνατότητες μιας κλάσης δημιουργώντας υποκλάσεις από αυτήν. Μια υποκλάση της Module_name που ορίσαμε παραπάνω είναι η εξής:

```
simple specialModule_name extends Module_name  
  
{  
  
x = 10;  
  
}
```

8.2 Compound modules

Ένα compound module ομαδοποιεί άλλα modules σε μια μεγαλύτερη μονάδα, αλλά δεν έχει κώδικα C++ που να σχετίζεται με αυτό. Η συμπεριφορά του είναι το αποτέλεσμα της συνεργασίας των simple modules που το αποτελούν. Η δήλωση ενός τέτοιου module έχει την παρακάτω μορφή:

```
module Comp_mod  
  
{  
  
types:  
  
parameters: submodules : connections:  
  
}
```

Τα modules που αποτελούν το compound module γράφονται κάτω από την δεσμευμένη λέξη submodules. Οι τύποι των καναλιών και των modules που χρησιμοποιούνται μόνο τοπικά μπορούν να γραφούν κάτω από το "types:". Για τα κανάλια θα μιλήσουμε παρακάτω.

Φυσικά και στα compound modules μπορούμε να δημιουργήσουμε άλλα modules που να κληρονομούν και να επεκτείνουν τις δυνατότητες τους με την βοήθεια του "extends".

8.3 Κανάλια (channels)

Τα κανάλια ενσωματώνουν παραμέτρους και συμπεριφορά για τις συνδέσεις (connections). Όπως και στα simple modules έτσι και στα κανάλια υπάρχει από πίσω τους μια κλάση C++. Ένα παράδειγμα είναι το εξής:

```
channel CustomChannel // χρειάζεται μια κλάση με το όνομα CustomChannel
{}
```

Συνήθως δεν χρειάζεται να γράψουμε τις δικές μας κλάσεις για κανάλια γιατί χρησιμοποιούμε τις έτοιμες που μας προσφέρει το OMNeT++. Αυτές είναι :

ned.IdealChannel

ned.DelayChannel

ned.DatarateChannel

Για τις παραμέτρους του κάθε τύπου μπορείτε να ανατρέξετε στο manual του OMNeT++. Ακολουθεί ένα παράδειγμα:

```
channel C extends ned.DatarateChannel
{
    datarate = 100Mbps; delay =
    100us; ber = 1e -10;
}
```

Παράμετροι

Όπως είπαμε και προηγουμένως οι παράμετροι είναι μεταβλητές που ανήκουν σε ένα module. Μπορεί να είναι του τύπου int, double, bool, string και xml. Παίρνουν τις τιμές τους είτε από τα αρχεία NED, είτε από τα αρχεία διαμόρφωσης (.ini), είτε κατά τη διάρκεια της προσομοίωσης από τον χρήστη.

Αν η τιμή μιας παραμέτρου οριστεί με ρητό τρόπο σε ένα αρχείο NED, τότε δεν μπορεί να αλλάξει από ένα αρχείο διαμόρφωσης. Μπορούμε επίσης να ορίσουμε μια default τιμή η οποία θα εφαρμόζεται αν δεν δοθεί τιμή στην παράμετρο με άλλο τρόπο. (= default(...)).

Αν ορίσουμε μια παράμετρο με τον τροποποιητή volatile, τότε η παράμετρος θα υπολογίζεται κάθε φορά που διαβάζεται.

Θύρες (Gates)

Οι θύρες είναι τα σημεία επικοινωνίας των modules. Εκτός από απλές θύρες μπορούμε να δημιουργήσουμε και gate vectors (διανύσματα από θύρες -ουσιαστικά έναν μονοδιάστατο πίνακα). Το μέγεθος των διανυσμάτων μπορούμε να το δηλώσουμε μέσα στις αγκύλες ή να το αφήσουμε κενό. Στην συνέχεια μπορούμε να επεκτείνουμε το gate vector και να προσθέσουμε νέες θύρες με τον τελεστή gate++. Π.χ.

```
simple Treenode
{
Gates:
inout parent; inout children[];
}
```

Συνδέσεις (Connections)

Οι θύρες input και output συνδέονται με απλό βέλος ενώ οι inout με διπλό. Για να συνδέσουμε δύο θύρες με ένα κανάλι χρησιμοποιούμε τα βέλη και βάζουμε το όνομα του καναλιού ανάμεσα. Μερικά παραδείγματα είναι τα εξής: <--> {delay=10ms} <--> <--> <--> C <-->

Ιδιότητες (Properties)

Οι ιδιότητες μας επιτρέπουν να προσθέσουμε metadata σε οποιοδήποτε αντικείμενο της NED. Το @display και το @class είναι κάποια παραδείγματα που έχουμε δει μέχρι τώρα.

Κληρονομικότητα

Μια περίπτωση που πρέπει να τονίσουμε είναι ότι, όταν κάνουμε `extend` ένα `module` τόσο στην NED όσο και στην C++, τότε πρέπει να χρησιμοποιήσουμε την ιδιότητα `@class` υποχρεωτικά για να χρησιμοποιηθεί η νέα κλάση στη C++.

ΚΕΦΑΛΑΙΟ 9

ΠΕΙΡΑΜΑΤΙΚΟ ΜΕΡΟΣ

9.1 Εισαγωγή

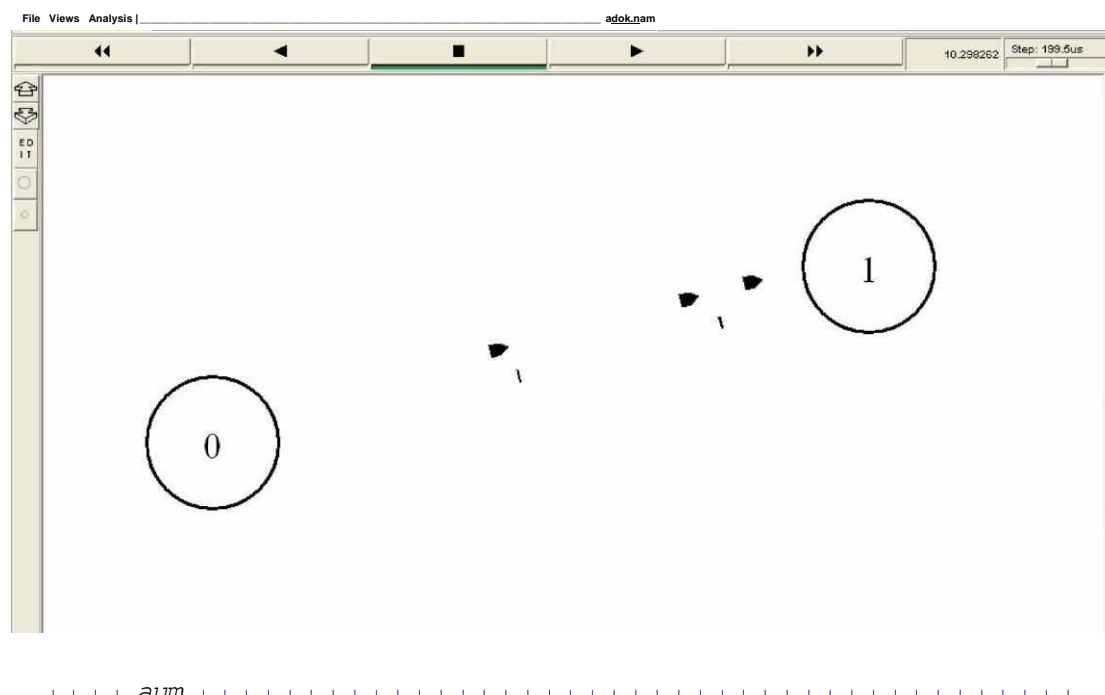
Το πειραματικό μέρος της εργασίας συνίσταται στην εξομοίωση διάφορων τοπολογιών 802.11 και στη μελέτη των επιδόσεών τους. Η έκδοση του NS που χρησιμοποιήθηκε είναι η πιο πρόσφατη (2.26), και συνοδεύεται από την τελευταία έκδοση του προγράμματος απεικόνισης NAM. Στην έκδοση αυτή, το NAM έχει πλέον τη 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.

9.2 Πρώτο Σενάριο

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.

Φαίνεται μια εικονική αναπαράσταση στο NAM της τοπολογίας του δικτύου.



Σχήμα 5.1 Αναπαράσταση του πρώτου σεναρίου στο NAM

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	10,0154 3	_0_	AGT	.	1000	Αποστολή πακέτου από τον κόμβο_0	
r	10,0154 3	_0_	RTR	.	8	tcp	1000 Μεταβίβαση του πακέτου στον router
s	10,0154 3	_0_	RTR	.	8	tcp	1020 Αποστολή πακέτου από τον router
s	10,0154 3	_0_	AGT	.	9	tcp	1000
r	10,0154 3	_0_	RTR	.	9	tcp	1000
s	10,0154 3	_0_	RTR	.	9	tcp	1020

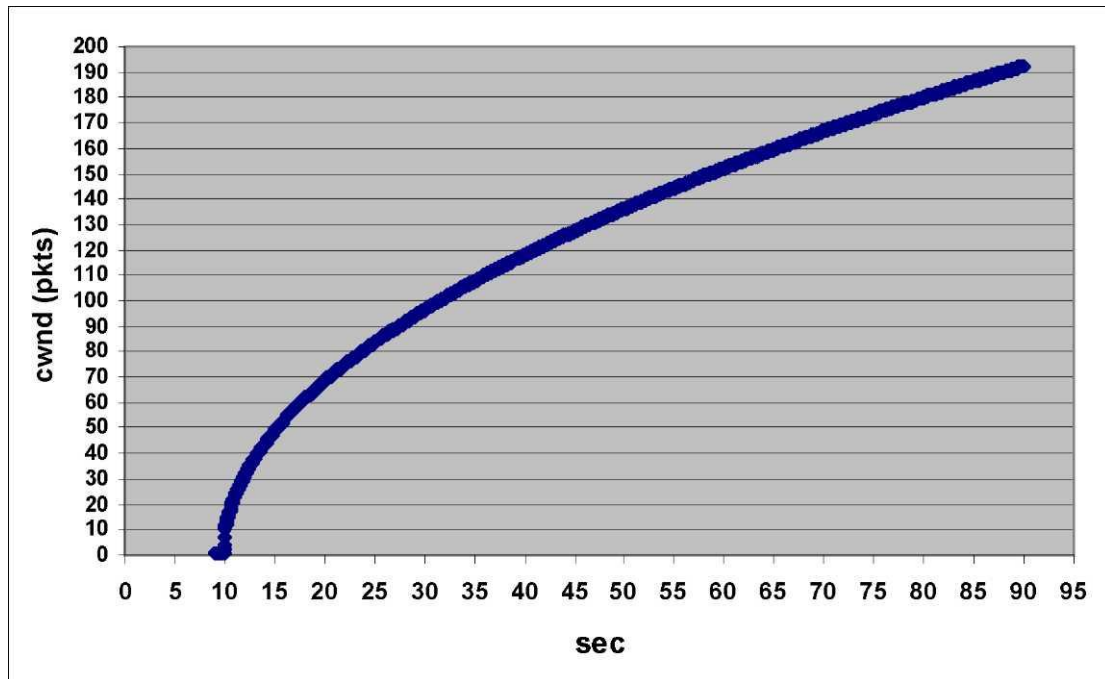
r	10,02042_1_	AGT	---	8	tcp	1020	Λήψη του πακέτου από τον κόμβο_1
s	10,02042	_1_	AGT	.	10	ack	40
r	10,02042	_1_	RTR	.	10	ack	40
s	10,02042	_1_	RTR	.	10	ack	60
r	10,02175	_0_	AGT	.	7	ack	60
s	10,02175	_0_	AGT	.	11	tcp	1000
r	10,02175	_0_	RTR	.	11	tcp	1000
s	10,02175	_0_	RTR	.	12	tcp	1020
r	10,02682	_1_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

Επεξεργασία των αποτελεσμάτων

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

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.

Επεξεργασία του throughput

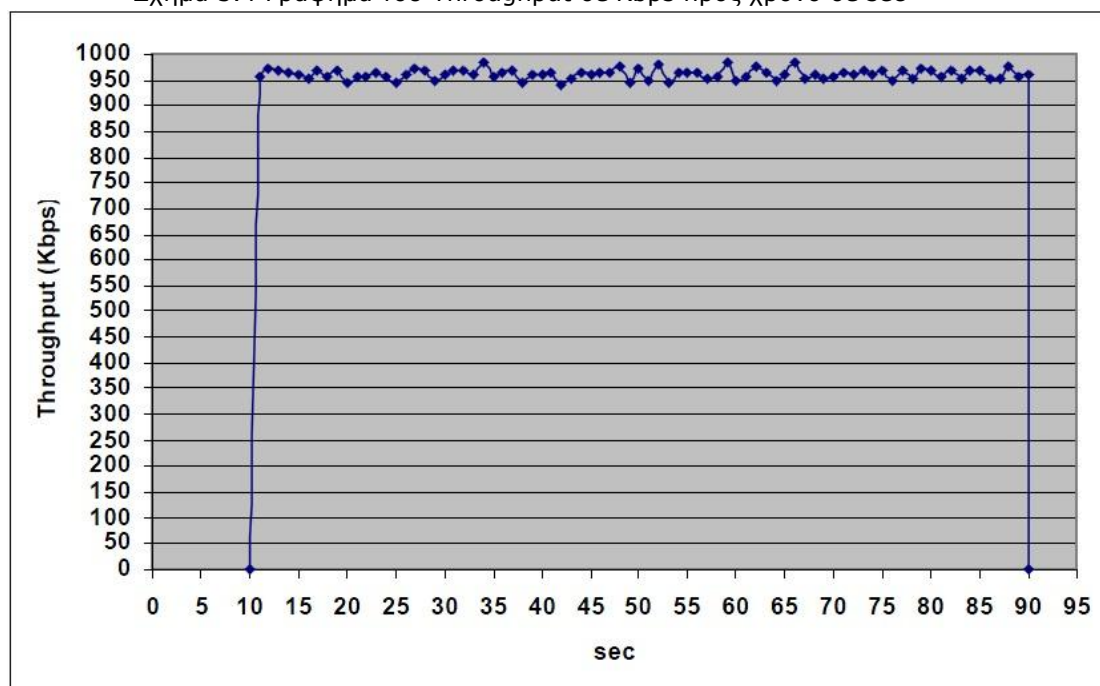
Για τον υπολογισμό του throughput του παραλήπτη χρειάζεται να απομονώσουμε τις γραμμές του trace file στις οποίες γίνεται λήψη από το AGT επίπεδο από του κόμβου_1 και αφού υπολογίσουμε τον συνολικό όγκο δεδομένων να το διαιρέσουμε δια τον χρόνο που διαρκεί η αποστολή δεδομένων. Στην προκειμένη περίπτωση 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).

Μετρήσεις

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

Σχήμα 5.4 Γράφημα του Throughput σε Kbps προς χρόνο σε sec



9.3 Δεύτερο Πείραμα

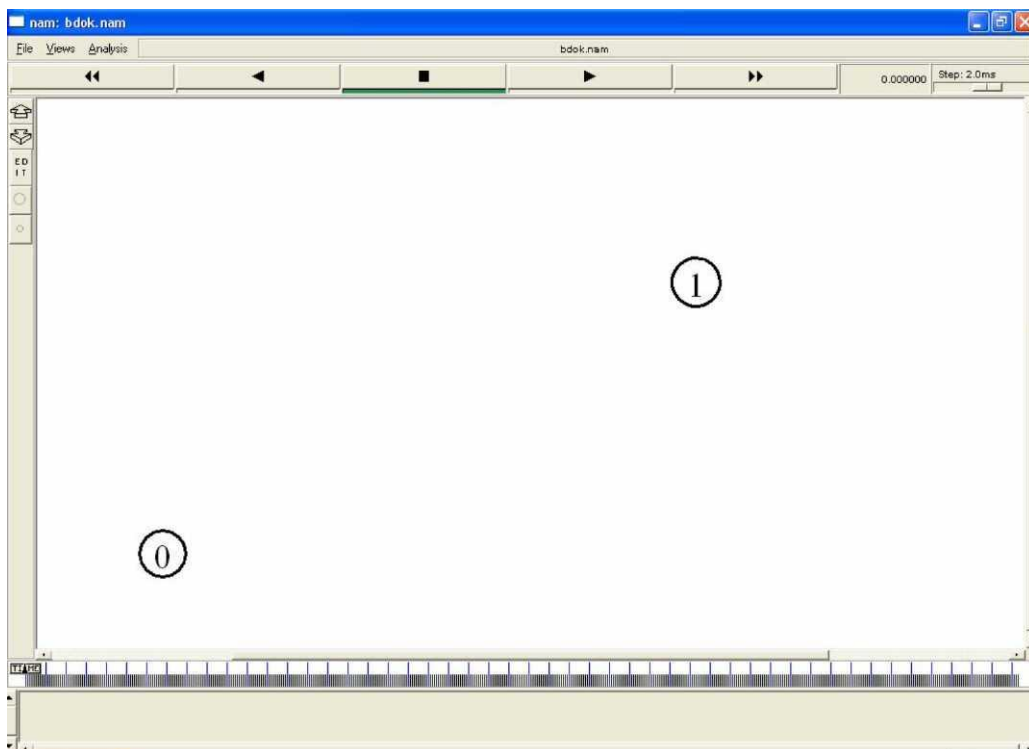
Γενική Περιγραφή

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 :

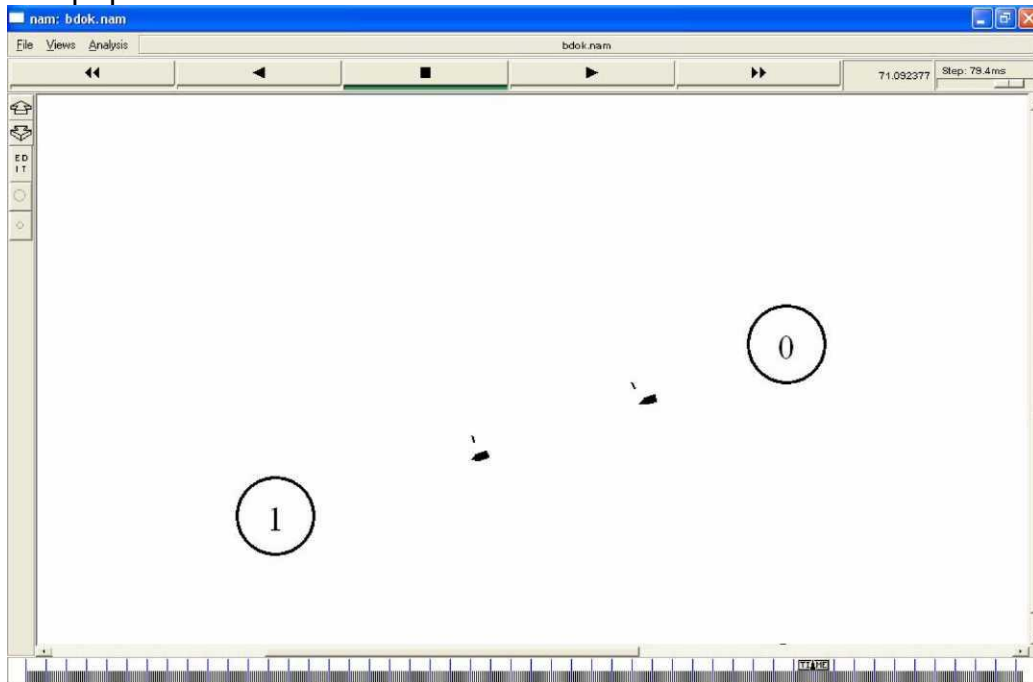
```
$ns at 50.0 "$node_(1) setdest 25.0 20.0 15.0" $ns at 10.0  
"$node_(0) setdest 170.0 35.0 25.0" $ns at 100.0  
"$node_(1) setdest 490.0 480.0 15.0" $ns at 85.0  
"$node_(0) setdest 390.0 200.0 20.0"
```

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



Σχήμα 5.5 Αρχικές θέσεις των κόμβων του 2^{ου} σεναρίου

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



Σχήμα 5.6 Οι θέσεις των κόμβων του 2^{ου} σεναρίου κατά την κίνηση τους

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

Τερζόγλου Αποστολία – Κατερέλος Άγγελος

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	

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

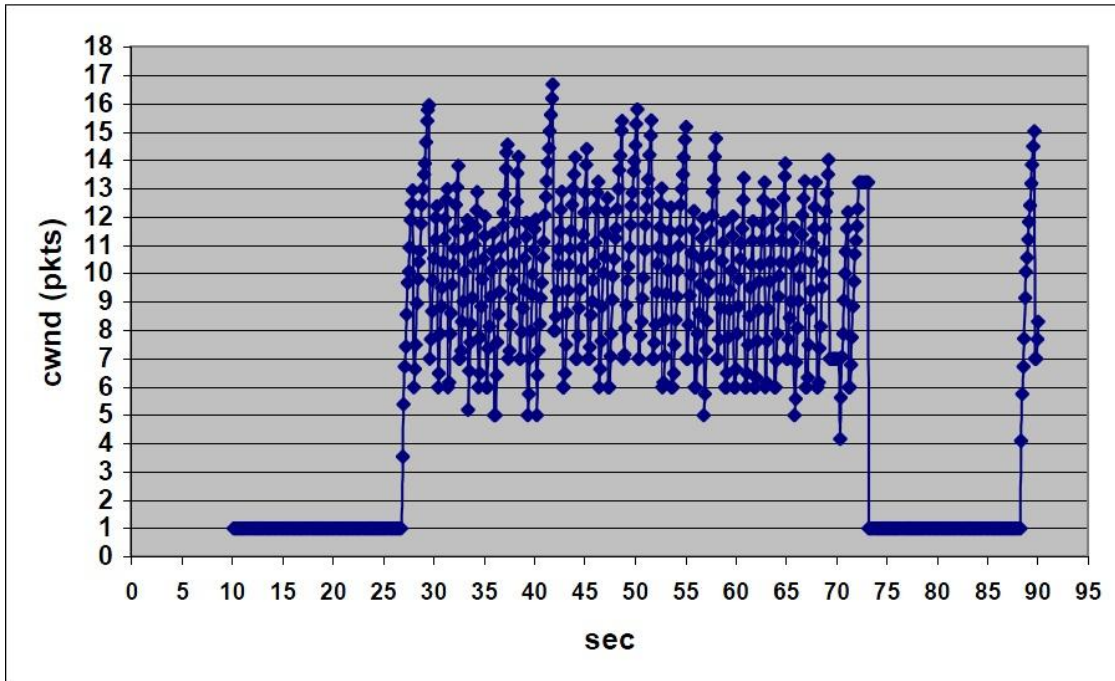
Ας μελετήσουμε την πορεία ενός πακέτου tcp που στέλνεται από τον κόμβο 0 και λαμβάνεται από τον κόμβο 1. Ένα τέτοιο πακέτο είναι πχ. το πακέτο 177 (packet id, δηλαδή ο αριθμός μετά τις τρεις παύλες). Βλέπουμε λοιπόν ότι το πακέτο αυτό στέλνεται αρχικά από τον κόμβο 0 (χρόνος 27,72309) και έχει μέγεθος 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).

Επεξεργασία των αποτελεσμάτων

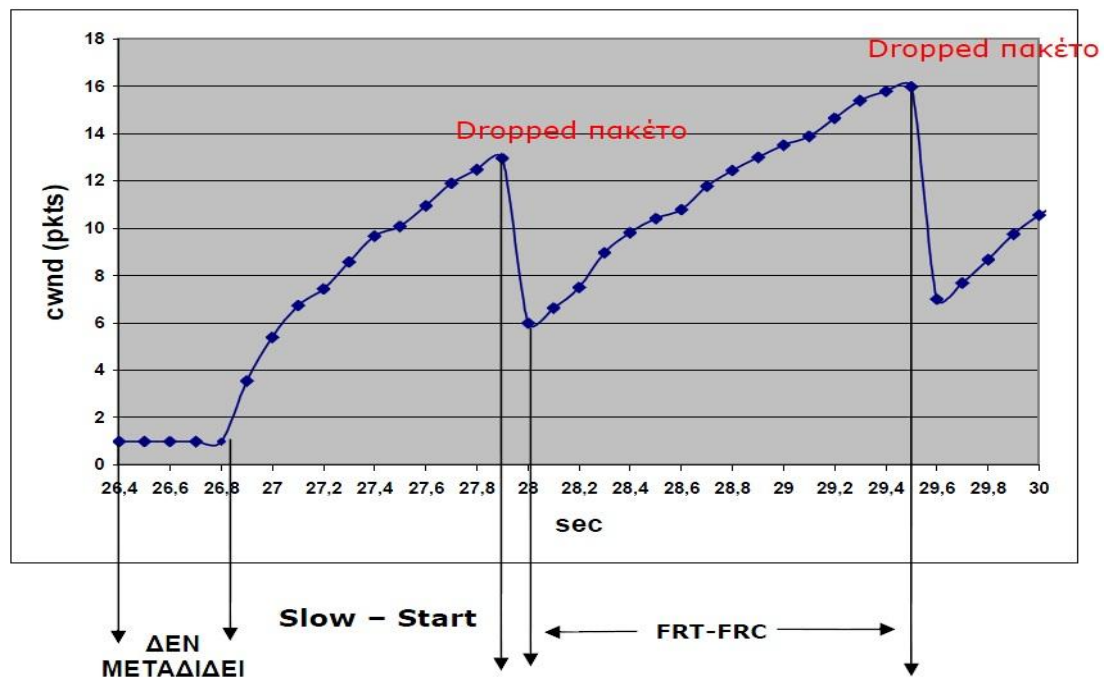
Επεξεργασία του cwnd

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

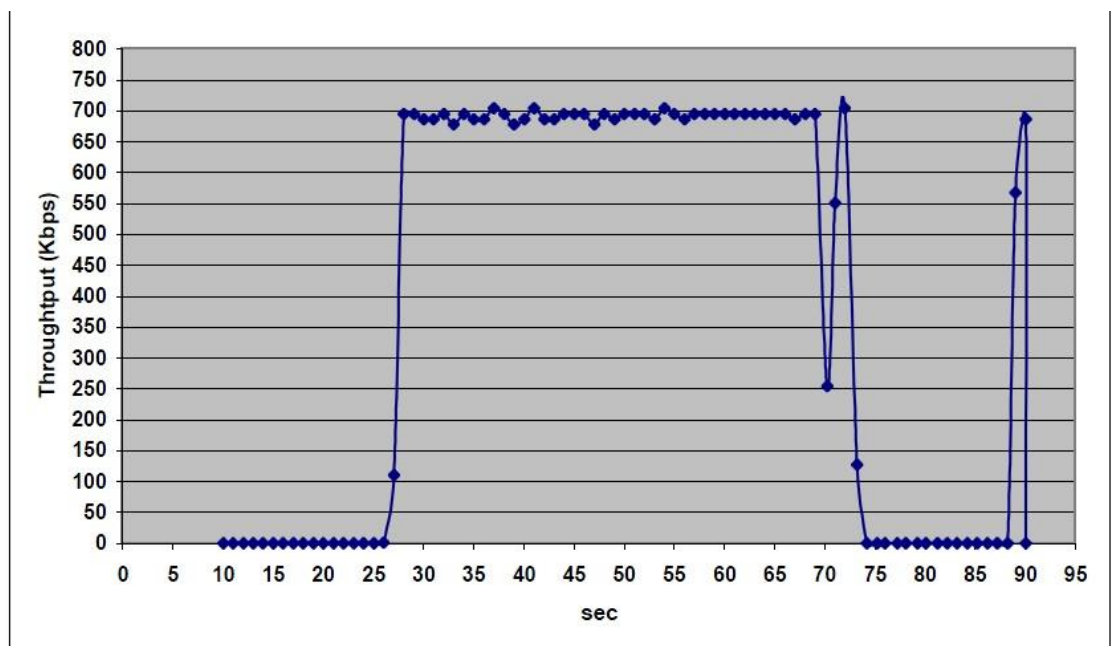
Σχήμα 5.8 Ανάλυση της αλλαγής μεταξύ Slow-Start και Fast Retransmit-Fast Recovery

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).

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

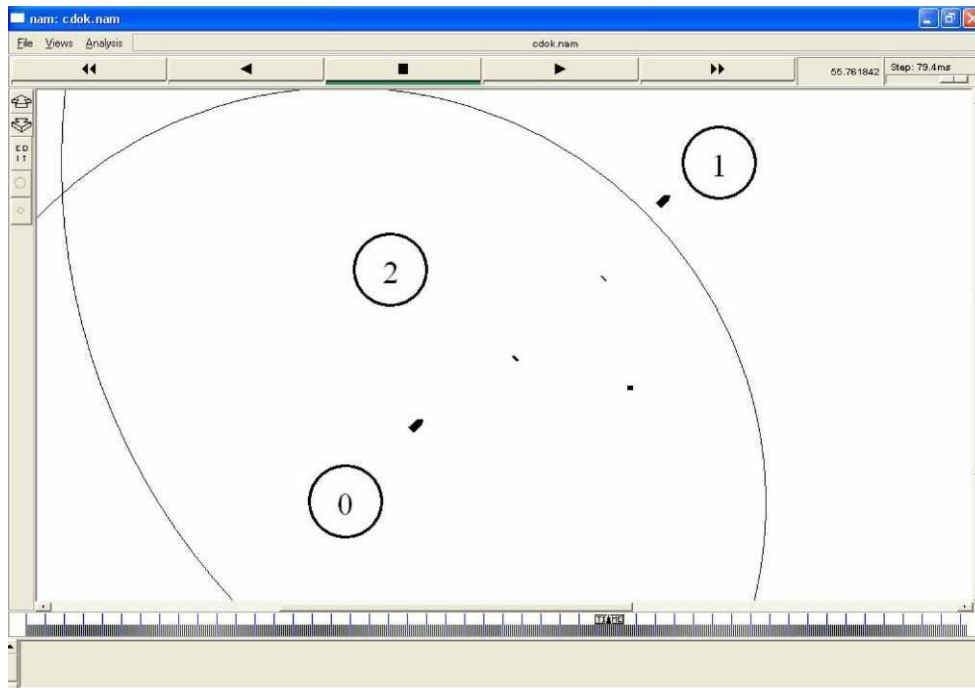
9.4 Τρίτο σενάριο

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 :

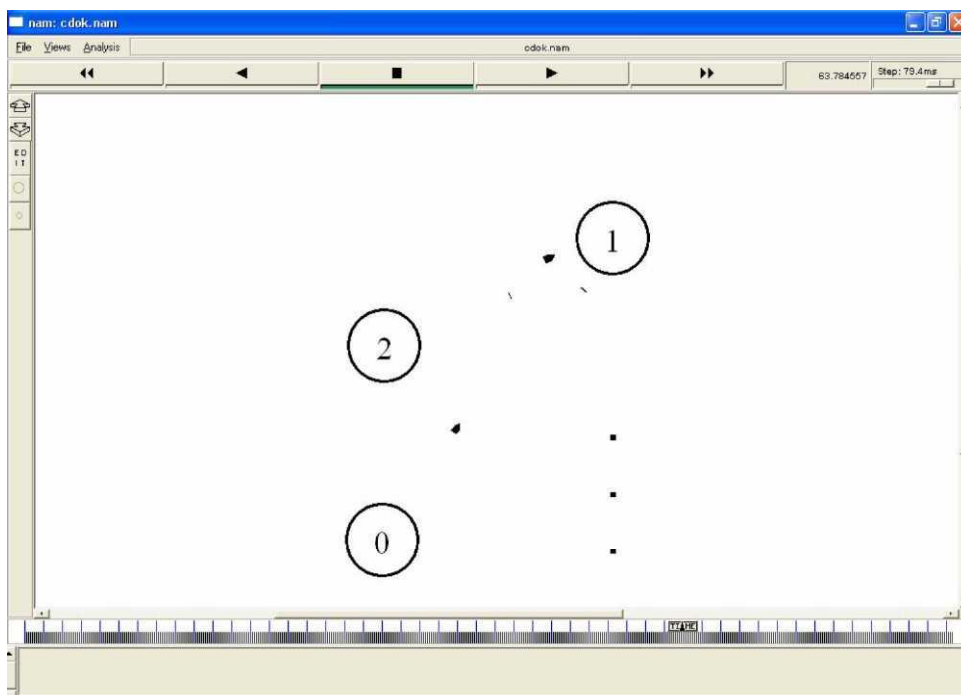
```
set tcpl [new Agent/TCP/Newreno]
$tcp set class_ 2
$tcp set maxrto_ 1
$tcp set ssthresh_ 20
$tcp set packetSize_ 1000
$ns attach-agent $node_(2) $tcp1
set sink1 [new Agent/TCPSink]
$ns attach-agent $node_(1) $sink1
$ns connect $tcp1 $sink1
set ftp1 [new Application/FTP]
$ftp1 attach-agent $tcp1
$ns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

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	

Τερζόγλου Αποστολία – Κατερέλος Άγγελος

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	

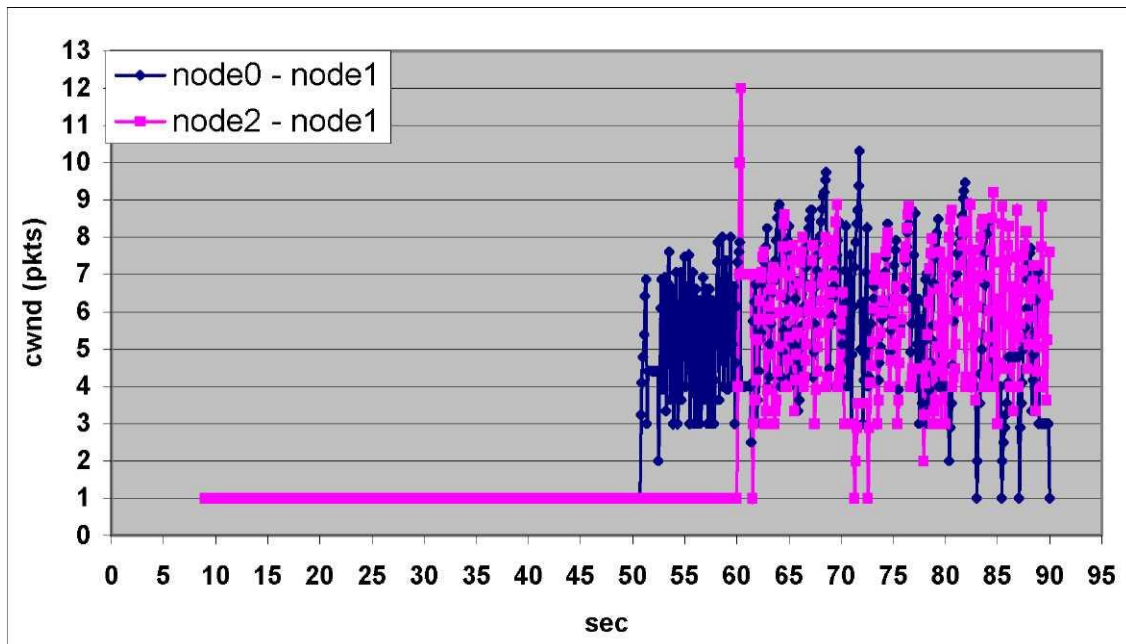
Θα αναλύσουμε την πορεία ενός πακέτου κατά την διάρκεια που επικοινωνούν μόνο ο κόμβος_0 με τον κόμβο_1. Η πορεία του είναι ίδια ακριβώς με την πορεία των πακέτων που μελετήσαμε στα παραπάνω παραδείγματα. Παρατηρούμε ότι έχει πολύ μικρή χρονική καθυστέρηση, μόλις 0.01 sec, από την στιγμή της αποστολής 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.

Επεξεργασία των αποτελεσμάτων

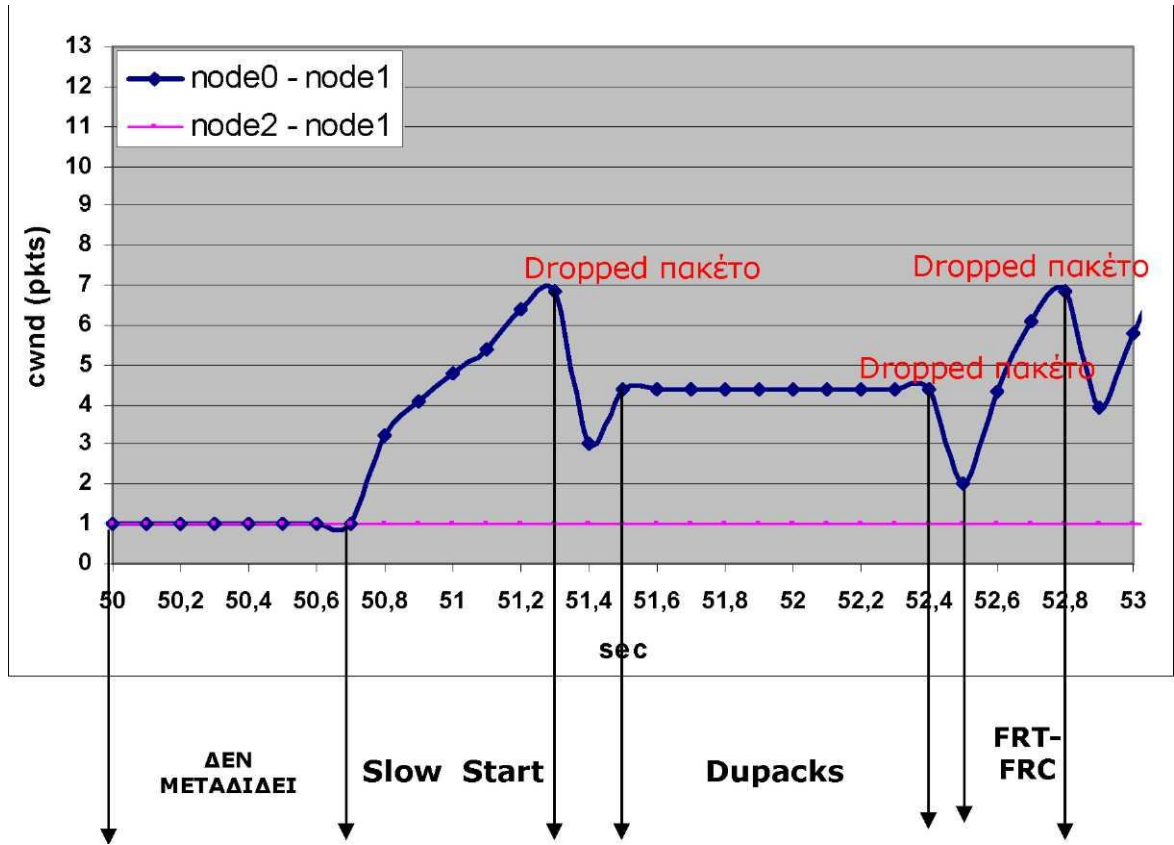
Επεξεργασία του cwnd

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 λόγο 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.

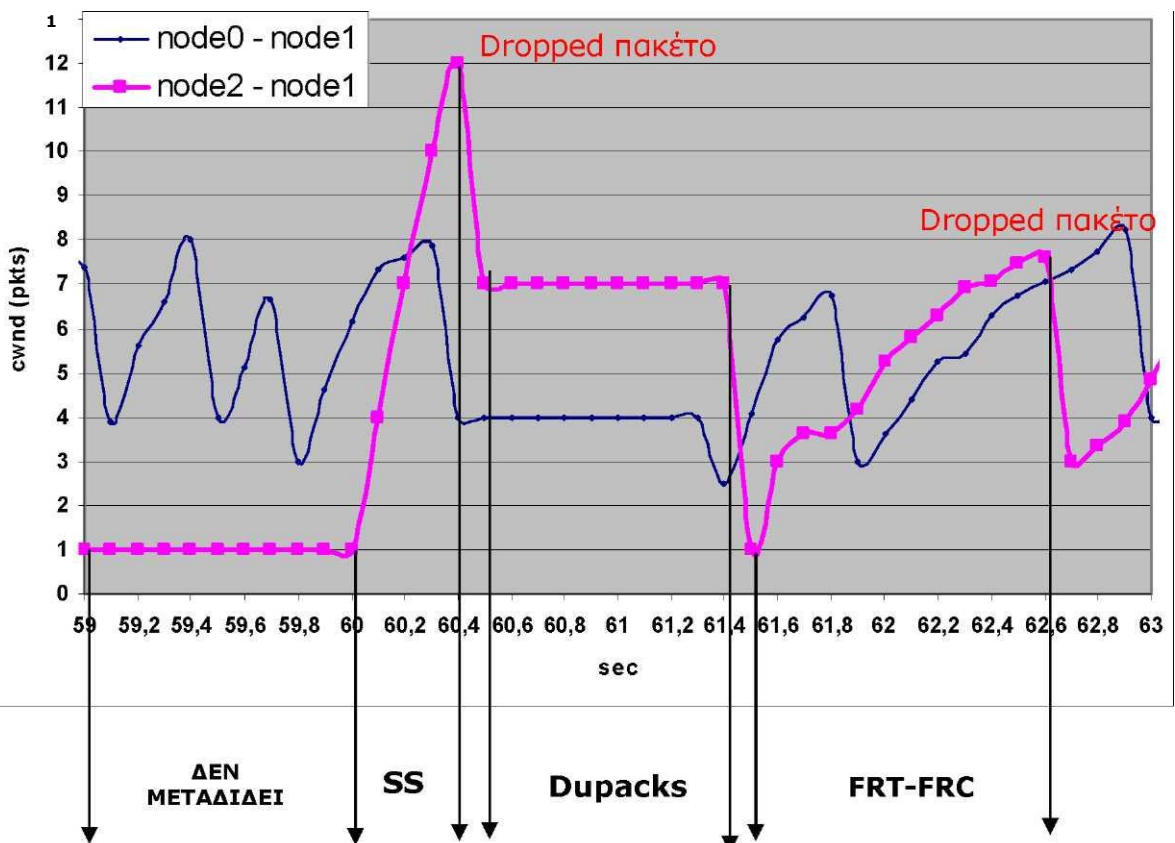


Σχήμα 5.12 Γράφημα του cwnd (pkts) προς χρόνο (sec)



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

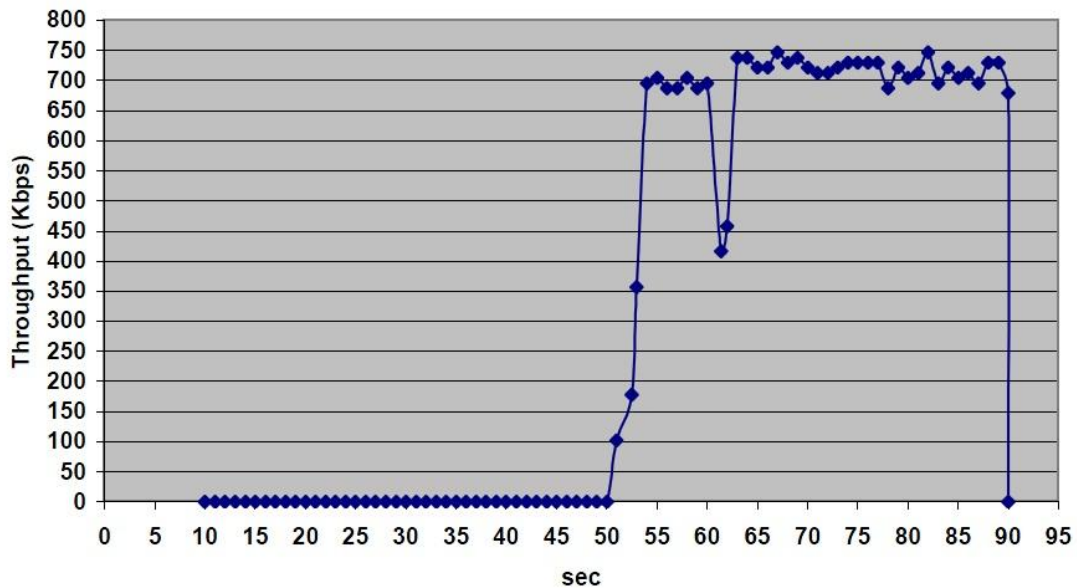
Σχήμα 5.14 Ανάλυση της εναλλαγής μεταξύ Slow-Start, Congestion Avoidance και Fast Retransmit-Fast Recovery

Επεξεργασία του throughput

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

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

9.5 Παρατηρήσεις

Στο πρώτο σενάριο έχουμε την ιδανική περίπτωση. Οι κόμβοι είναι σταθεροί, οι 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 τίθενται 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.

ΠΑΡΑΡΤΗΜΑΤΑ

6.1 Κώδικας φίλτρου AWK

```
BEGIN{
time = 10.0000 pkts
= 0 st = 1.0000
}
{

if ($2 < time) {
if($1 == "r" && $4 == "AGT" && $3 ==          && $7 == "tcp" )
{pkts = pkts + $8
}
}
else
{
if($1 == "r" && $4 == "AGT" && $3 ==          && $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)
set val(ant) ,
set val(ifqlen)
set val(nn)
set val(rp)
set val(x) 500;
set val(y) 500;
set val(stop) 90;

LL ;
Antenna/OmniAnten
na
50 ;

2

DSDV;

#=====
# 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]
```

Τερζόγλου Αποστολία – Κατερέλος Άγγελος

```
$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
```

```
# Δημίουργία 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)
set val(ant)            ,
set val(ifqlen)
set val(rp)
set val(nn)
set val(x) 500;
set val(y) 500;
set val(stop) 90;
LL ;
Antenna/OmniAntenna
5 ;
DSDV
    3

#=====
```

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
```

Συμπεράσματα

Όπως γίνεται κατανοητό υπάρχουν αρκετά ελεύθερα και open-source εργαλεία προσομοίωσης δικτύων τα οποία μπορούν και λειτουργούν σε όλες τις πιθανές πλατφόρμες που διατίθενται στην αγορά. Από εκεί και πέρα τον ποιον προσομοιωτή θα επιλέξει ο κάθε προγραμματιστής αφορά κυρίως στο περιβάλλον στο οποίο αυτός είναι συνηθισμένος να δουλεύει οπότε και γνωρίζει καλύτερα, αλλά και με ποια γλώσσα προγραμματισμού είναι κανείς περισσότερο εξοικειωμένος. Ίσως το περισσότερο διαδεδομένο εργαλείο προσομοίωσης, από αυτά που περιγράψαμε, να θεωρείται από πολλούς το NS-2 το οποίο παρέχει μια μεγάλη γκάμα τοπολογιών, πρωτοκόλλων και εφαρμογών που μπορεί να προσομοιώσει. Εξίσου διαδεδομένου είναι και το OMNET++ διότι εκτός του από μόνο μπορεί να χρησιμοποιηθεί για οποιαδήποτε προσομοίωση, έχουν αναπτυχθεί με βάση αυτό και άλλα εργαλεία τα οποία εξειδικεύονται σε τοπολογίες ή συγκεκριμένα πρωτόκολλα δικτύων και τα οποία στη συνέχεια μπορούν να συνδυαστούν άψογα με οποιαδήποτε άλλη εφαρμογή έχει αναπτυχθεί σε OMNET++. Ένα εργαλείο που ίσως δε χρειάζεται ιδιαίτερα μεγάλη εξοικείωση με γλώσσες προγραμματισμού είναι το Cnet το οποίο προσφέρει ένα πολύ εύχρηστο και βοηθητικό γραφικό περιβάλλον και διευκολύνει κατά πολύ την εκτέλεση των προσομοιώσεων. Τέλος παρουσιάστηκαν και κάποια άλλα εργαλεία λιγότερα διαδεδομένα, όπως το GloMoSim και τα δύο προϊόντα της NIST τα οποία όμως μπορούν αποβούν εξίσου αποτελεσματικά στη διαδικασία της προσομοίωσης των δικτύων. Σίγουρα υπάρχουν και άλλα εργαλεία τα οποία έχουν αναπτυχθεί και για τα οποία δε συγκεντρώσαμε τις απαραίτητες πληροφορίες για να τα παρουσιάσουμε, όπως είναι το Qualnet, το ONE simulator, το 3LS κ.α.

ΒΙΒΛΙΟΓΡΑΦΙΑ

Βιβλία

[1]. Peter Piedad, Jeremy Ethridge, Mandeep Baines, and Farhan Shallwani. *A Network Simulator, Differentiated Services Implementation*. Open IP, Nortel Networks, 2000.

[2]. David B. Johnson. Validation of Wireless and Mobile Network Models and Simulation. In Proceedings of the DARPA/NIST Network Simulation Validation Workshop, Fairfax, Virginia, USA, May, 1999.

[3] Rajive Bagrodia, Richard Meyer, Mineo Takai, Yu-an Chen, Xian Zeng, Jay Martin, and Ha Yoon Song. PARSEC: A Parallel Simulation Environment for Complex Systems. IEEE Computer, 31(10), 77-85, October, 1998.

<http://dlib.computer.org/co/books/co1998/pdf/rx077.pdf>

[4] X. Zeng, R. Bagrodia and M. Gerla, "GloMoSim: A Library for Parallel Simulation of Large-Scale Wireless Networks," 12th Workshop on Parallel and Distributed Simulation, May 1998.

[5] A. Varga. OMNeT++ User Manual. Department of Telecommunications, Technical University of Budapest, 1997.

URL reference: <ftp://ftp.hit.bme.hu/sys/anonftp/omnetpp/doc/usman.pdf>.

Συνέδρια

[1] Peter Piedad, Jeremy Ethridge, Mandeep Baines, and Farhan Shallwani. *A Network Simulator, Differentiated Services Implementation*. Open IP, Nortel Networks, 2000.

[2] A. Varga. The OMNeT++ discrete event simulation system. In Proceedings of the European Simulation Multiconference (ESM'2001)

[3] U. Kaage, V. Kahmann, and F. Jondral. An OMNeT++ TCP model. In Proceedings of the European Simulation Multiconference (ESM'2001)

[4] Pierre L'Ecuyer. Software for uniform random number generation: Distinguishing the good and the bad. In *Proceedings of the 2001 Winter Simulation Conference*, December 2001.

Sites

[1] NS2 Homepage : <http://www.isi.edu/nsnam/ns/>

[2] NS2 Complete Tutorial: <http://www.isi.edu/nsnam/ns/tutorial/index.html>

- [3] <http://www.omnetpp.org/>
- [4] <http://mobility-fw.sourceforge.net>
- [5] <http://ctieware.eng.monash.edu.au/twiki/bin/view/Simulation/Documentation>
- [6] <http://www.csse.uwa.edu.au/cnet/>
- [7] <http://pcl.cs.ucla.edu/projects/glomosim/>
- [8] http://w3.antd.nist.gov/Hsntg/prd_atm-sim.html
- [9] http://w3.antd.nist.gov/Hsntg/prd_aprops.html
- [10] <http://www.isi.edu/nsnam/nam/index.html>
- [11] <http://www.linuxjournal.com/article/5929>
- [12] <http://community.omnest.com/doc/INET-Tutorial/>
- [13] <http://pcl.cs.ucla.edu/projects/parsec/>
- [14] <http://www.cs.cornell.edu/skeshav/real/overview.html>
- [15] <http://www.nwfusion.com/reviews/0524rev.html>
- [16] <http://www.idsia.ch/~andrea/simtools.html>
- [17] <http://www.topology.org/soft/sim.html>
- [18] http://www.brothersoft.com/Utilities_Miscellaneous_AdventNet_Simulation_To_olkit
- [19] <http://www.scalable-networks.com/products/qualnet.php>
- [20] <http://pcl.cs.ucla.edu/projects/glomosim/GloMoSimManual.html>