

**Τμήμα
Μηχανικών
Πληροφορικής τ.ε.**

Τεχνολογικό Εκπαιδευτικό Ίδρυμα
Δυτικής Ελλάδας

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

**ΕΡΓΑΛΕΙΑ ΠΡΟΣΟΜΟΙΩΣΗΣ ΔΙΚΤΥΩΝ : ΕΠΙΣΚΟΠΗΣΗ ΚΑΙ
ΣΥΓΚΡΙΣΗ ΔΥΝΑΤΟΤΗΤΩΝ**

Επιβλέπων Καθηγητής : Ασημακόπουλος Γεώργιος

**Φοιτητές
Οικονομάκης Ιωάννης Α.Μ 806
Πάσσος Αντώνιος Α.Μ. 1082**

Αντίρριο 2015-2016

Σύνοψη Πτυχιακής

Από τα προγράμματα περιήγησης του ιστού, στα κινητά τηλέφωνα, μέχρι τα καφεενεία με δημόσια πρόσβαση στο Διαδίκτυο, από τον παραδοσιακό χώρο εργασίας με υποδομή τεχνολογίας των πληροφοριών σε κάθε γραφείο, μέχρι τους δικτυωμένους αισθητήρες περιβάλλοντος και μέχρι το διαπλανητικό Διαδίκτυο -και ενώ φαίνεται ότι τα δίκτυα υπολογιστών είναι πανταχού παρόντα- εντυπωσιακές νέες εφαρμογές αναπτύσσονται, οι οποίες επεκτείνουν την προσπέλαση των δικτύων υπολογιστών ακόμη περισσότερο. Φαίνεται ότι τα δίκτυα υπολογιστών βρίσκονται παντού!

Με τον ίδιο τρόπο αναπτύσσεται και η προσομοίωση, από ένα εξειδικευμένο εργαλείο λήψης αποφάσεων, σε μέσο οικονομικών αναλύσεων, εκπαιδευτικό βοήθημα για πολύπλοκες λειτουργίες και συστήματα, εργαλείο ανάλυσης μικροσκοπικών και μακροσκοπικών συστημάτων και υποστηρικτικό λογισμικό για υπολογιστικά συστήματα. Στις υπάρχουσες γλώσσες προσομοίωσης προστέθηκαν νέες, είτε με τη μορφή αυτόνομων γλωσσών (όπως η VHDL), είτε με τη μορφή εργαλείων μοντελοποίησης σε υπολογιστικά πακέτα (όπως το SIMULINK του MATLAB).

Η παρούσα πτυχιακή εργασία μελετά την προσομοίωση, τα πιο γνωστά εργαλεία της και τέλος συγκρίνει δυο από αυτά, τον Network Simulator 2 με τον πιο σύγχρονο του Network Simulator 3. Θα πραγματοποιηθούν δικτυακές προσομοιώσεις διαφορετικών αρχιτεκτονικών και πρωτοκόλλων με εξαγωγή αποτελεσμάτων και τέλος σύγκριση των δυο προσομοιωτών ως προς τον χρόνο εκτέλεσης, τον κώδικα και την απόδοση.

Συμπερασματικά ο NS-2 έχοντας σταματήσει η ανάπτυξή του έχει πετύχει απολύτως τους σκοπούς του. Ο NS-3 ως πιο εξελιγμένη έκδοση του NS-2 με περισσότερες δυνατότητες προσομοίωσης, δεν είναι ικανός να τον αντικαταστήσει πλήρως. Χρήζει αλλαγών σε ζητήματα που έχουν παρατηρηθεί, τα οποία μπορούν να διευθετηθούν μέσω μεγαλύτερης συνεισφοράς της ερευνητικής και ακαδημαϊκής κοινότητας.

Περιεχόμενα

| | |
|---|----|
| 1. Εισαγωγή..... | 6 |
| 2. Δίκτυα υπολογιστών, Ορισμοί και Πρωτόκολλα..... | 8 |
| 2.1 Γενικά περί δικτύων..... | 8 |
| 2.2 TCP/IP..... | 9 |
| 2.3 FTP..... | 9 |
| 2.4 UDP..... | 11 |
| 2.5 802.11..... | 12 |
| 3. Προσομοίωση Δικτύων..... | 13 |
| 3.1 Ορισμός, Διαχωρισμός προσομοίωσης-εξομοίωσης..... | 13 |
| 3.2 Λειτουργία προσομοίωσης..... | 14 |
| 3.3 Χρήσεις..... | 16 |
| 3.4 Είδη προσομοιωτών..... | 18 |
| 3.5 Πλεονεκτήματα..... | 22 |
| 3.6 Μειονεκτήματα..... | 22 |
| 3.7 Επισκόπηση εξελίξεων..... | 23 |
| 4. Εργαλεία προσομοίωσης | 24 |
| 4.1 OMNET..... | 24 |
| 4.2 J-SIM..... | 25 |
| 4.3 SSFNET..... | 26 |
| 4.4 NETSIM..... | 27 |

| | |
|--|----|
| 4.5 TED..... | 28 |
| 4.6 NEST..... | 29 |
| 4.7 COMNET III..... | 29 |
| 4.8 INSANE..... | 31 |
| 4.9 NIST..... | 31 |
| 4.10 TRAFFIC..... | 31 |
| 4.11 GLOMOSIM..... | 32 |
| 4.12 SHURA VIRTUAL ENTERPRISE..... | 32 |
| 4.13 CNET..... | 33 |
| 4.14 QUALNET DEVELOPER..... | 33 |
| 4.15 GTNETS..... | 35 |
| 4.16 NCTUNS..... | 36 |
| 4.17 PARSEC..... | 36 |
| 4.18 PERFORMANCE PROPHET..... | 36 |
| 4.19 REAL..... | 37 |
| 4.20 CPSIM..... | 37 |
| 4.21 OPTSIM..... | 37 |
| 4.22 PHYSIM..... | 39 |
| 4.23 OVERSIM..... | 39 |
| 5. Δικτυακός προσομοιωτής Network Simulator 2..... | 40 |
| 5.1 Ιστορία..... | 40 |

| | | |
|-----|---|----|
| 5.2 | Εγκατάσταση Network Simulator 2..... | 41 |
| 5.3 | TCL SCRIPT-NAM..... | 41 |
| 5.4 | X-GRAPH..... | 45 |
| 5.5 | Χαρακτηριστικά..... | 46 |
| 6. | Δικτυακός προσομοιωτής Network simulator 3..... | 48 |
| 6.1 | Ιστορία..... | 48 |
| 6.2 | Εγκατάσταση..... | 49 |
| 6.3 | Χρήση..... | 50 |
| 6.4 | NETANIM..... | 51 |
| 6.5 | WIRESHARK..... | 52 |
| 6.6 | Χαρακτηριστικά..... | 53 |
| 7. | Σύγκριση NS-2-NS-3..... | 55 |
| 7.1 | Υπολογιστικά περιβάλλοντα..... | 55 |
| 7.2 | Πακέτα..... | 55 |
| 7.3 | Εξαρτήσεις..... | 56 |
| 7.4 | Απόδοση..... | 57 |
| 7.5 | Αρχιτεκτονική..... | 58 |
| 7.6 | Κύκλος ζωής..... | 59 |
| 8. | Σενάρια προσομοίωσης..... | 60 |
| 8.1 | Σενάρια προσομοίωσης στον NS-2..... | 60 |
| 8.2 | Σενάρια προσομοίωσης στον NS-3..... | 67 |
| 9. | Αποτελέσματα Προσομοιώσεων..... | 75 |

| | |
|--|----|
| 9.1 Προγραμματιστικές γλώσσες..... | 75 |
| 9.2 Χρόνος εκτέλεσης..... | 76 |
| 9.3 Αποτελέσματα και οπτικοποίηση..... | 76 |
| 10. Επίλογος-Συμπεράσματα..... | 77 |
| Βιβλιογραφία..... | 80 |

1. Εισαγωγή

Καθ'όλη την εξέλιξη στην επιστήμη των υπολογιστών, τα δίκτυα υπολογιστών και ιδιαίτερα το διαδίκτυο, έχουν μπει για τα καλά στη ζωή μας, διαμορφώνοντας νέες συνήθειες ακόμη και στον τρόπο που δουλεύουμε. Μέσω του διαδικτύου, έχουμε πρόσβαση στην ενημέρωση, σε ιδέες και πληροφορίες, πραγματοποιούμε συναλλαγές, επικοινωνούμε και πολλά άλλα που επηρεάζουν στο μέγιστο την καθημερινότητά μας.

Στο παρελθόν η κύρια λειτουργία ενός δικτύου ήταν να δίνει τη δυνατότητα σε διαφορετικούς υπολογιστές να μοιράζονται και να στέλνουν έγγραφα σε έναν εκτυπωτή. Σήμερα, η λειτουργία των δικτύων έχει γίνει πιο σύνθετη, από την άποψη ότι στα σύγχρονα δίκτυα, μπορούμε να αποθηκεύουμε αρχεία στους servers και στο cloud ενώ μέσω του δικτύου έχουμε πρόσβαση στο διαδίκτυο όπου μπορούμε να κάνουμε χρήση email, video κλήσεων κ.τ.λ. .

Στην προσπάθεια κατανόησης της λειτουργίας των δικτύων, της σχεδιάσής τους και της λειτουργικής συμπεριφοράς τους έχουμε τη δυνατότητα να χρησιμοποιήσουμε μια μεγάλη ιδέα της επιστήμης των υπολογιστών, την προσομοίωση. Μέσω της προσομοίωσης μπορούμε να προσομοιώσουμε όλα τα στάδια της σχεδίασης ενός πραγματικού δικτύου, δημιουργώντας, προσομοιώνοντας και αναλύοντας τη λειτουργία ενός εικονικού πρωτοτύπου σε ποικίλα περιβάλλοντα προσομοίωσης που θα αναλυθούν στη συνέχεια.

Η προσομοίωση από την αρχή της ανάπτυξής της ως τεχνολογία έως και σήμερα καθίσταται σαν μια πολύ σημαντική και δημοφιλή μέθοδος μελέτης των συστημάτων σε πολλούς κλάδους και εφαρμογές. Σαν τεχνολογία την βρίσκουμε στις επιστήμες, την μηχανική και σε πλήθος πολλαπλών κλάδων και εφαρμογών όπως και στην Επιστήμη των υπολογιστών.

Μέσω ηλεκτρονικών υπολογιστών, η προσομοίωση δύναται να μοντελοποιήσει συστήματα του πραγματικού κόσμου σε ένα υπολογιστικό, με σκοπό να διαπιστωθεί η συμπεριφορά του εν λόγω συστήματος. Μπορούν να χρησιμοποιηθούν πολλές και

διαφορετικές μεταβλητές για την ανάλυση και την μοντελοποίηση ενός συστήματος έτσι ώστε να βρεθούν τυχόν δυσλειτουργίες σε αυτό. Η τεχνολογία της προσομοίωσης συναντάται πολύ συχνά και στις επιστήμες της Φυσικής και τις Χημείας ενώ την βρίσκουμε και στην Οικονομία. Τα μέσα του 80 υιοθετήθηκε και στην Επιστήμη των υπολογιστών με απώτερο σκοπό την μελέτη και τον σχεδιασμό δικτυακών συστημάτων με χρήση ποικίλων πρωτοκόλλων. Μια δικτυακή προσομοίωση μπορεί να ελέγξει ένα πρωτόκολλο ως προς την ορθότητα και την απόδοση ενώ πολλοί προσομοιωτές μπορούν και να οπτικοποιήσουν τα αποτελέσματα .

Είναι προφανές ότι οι δικτυακές τεχνολογίες και όχι μόνο, εξελίσσονται πολύ γρήγορα με αναρίθμητους φορείς να συνεισφέρουν στην ανάπτυξή τους. Η δικτυακή προσομοίωση έρχεται με την σειρά της να προσφέρει στην ανάπτυξη αυτή, με πλήθος από ανοικτές πλατφόρμες ικανές να μελετήσουν διαφορετικές προσπάθειες και πακέτα προσομοιώσεων στο συνολικό δίκτυο.

Ερευνητές, σχεδιαστές δικτύων και πολλοί άλλοι που εργάζονται για την διασφάλιση της ποιότητας ενός συστήματος, προσομοιώνουν και αναλύουν την αποδοτικότητα διαφορετικών δικτυακών πρωτοκόλλων μέσω της επίδρασης διαφορετικών τιμών παραμέτρων.

Ο σκοπός αυτής της εργασίας είναι η μελέτη της προσομοίωσης σαν τεχνική μελέτης δικτύων , εργαλείων προσομοίωσης ως προς την λειτουργία τους ενώ θα γίνει εκτεταμένη αναφορά σε δυο εξ αυτών , τον NS-2 και τον NS-3 με πραγματοποίηση πειραμάτων και εξαγωγή συμπερασμάτων.

2. Δίκτυα υπολογιστών, Ορισμοί και Πρωτόκολλα

2.1 Γενικά περί Δικτύων

Η προσομοίωση δικτύων μελετά πρωτόκολλα και αλγορίθμους δικτυακών συστημάτων. Σε αυτό το κεφάλαιο θα γίνει αναφορά στις πιο βασικές έννοιες των δικτύων όπως επίσης και σε μερικά από τα πιο βασικά δικτυακά πρωτόκολλα.

Με τον όρο δίκτυο υπολογιστών ορίζουμε ένα σύνολο από αυτόνομους ή και μη αυτόνομους διασυνδεδεμένους υπολογιστές. Τα δίκτυα υπολογιστών κατηγοριοποιούνται ως προς το φυσικό μέσο διασύνδεσής τους (ενσύρματα ή ασύρματα), την γεωγραφική τους κάλυψη (Lan,Wan,Man) όπως και ως προς τον τρόπο πρόσβασης σε αυτά (δημόσια, ιδιωτικά).

Ένα τοπικό δίκτυο ή αλλιώς Lan (local area network) είναι ένα δίκτυο που συνδέει μια ομάδα υπολογιστών που βρίσκονται στην ίδια περιοχή και συνδέονται σε κοινό κύκλωμα προσφέροντας ταχύτητες από 10 μέχρι 100 Mbps. Ένα δίκτυο κορμού (backbone network) είναι ένα μεγαλύτερο, κεντρικό δίκτυο που συνδέει LANS , BNS , MANS και WANS. Το μητροπολιτικό δίκτυο συνδέει μεταξύ τους LANS και BNS που βρίσκονται σε διαφορετική περιοχή με WANS με επέκταση έως και 40 μίλια και ταχύτητες από 100 μέχρι 1000 Mbps. Ως εκτεταμένο (wired area network) ορίζεται το δίκτυο που συνδέει BNS και MANS καλύπτοντας χιλιάδες μίλια με ταχύτητες έως 2 Gbps. Η διέλευση πραγματοποιείται από δημόσια δίκτυα ενώ η μεταφορά δεδομένων από κόμβο σε κόμβο.

Τα δίκτυα στην πλειοψηφία τους χρησιμοποιούν τηλεφωνικά δίκτυα, δορυφόρους και οπτικές ίνες. Ως διαδίκτυα ορίζονται δίκτυα ευρείας περιοχής διασυνδέοντας επιμέρους δίκτυα. Το πιο διάσιμο και μεγαλύτερο δίκτυο είναι το Διαδίκτυο (Internet). Το Διαδίκτυο και τα δίκτυα υπολογιστών γενικά, κάνουν εκτεταμένη

χρήση πρωτοκόλλων. Διαφορετικά πρωτόκολλα χρησιμοποιούνται ώστε να γίνουν διαφορετικές εργασίες επικοινωνίας. Ένα πρωτόκολλο ορίζει την μορφή και την σειρά των μηνυμάτων που ανταλλάσσονται ανάμεσα σε δύο ή περισσότερες επικοινωνούσες οντότητες, όπως και τις ενέργειες που γίνονται κατά την διάρκεια της μετάδοσης και της λήψης ενός μηνύματος ή άλλου γεγονότος.

2.2 TCP/IP

Η πιο σημαντική συλλογή πρωτοκόλλων επικοινωνίας, καθώς πάνω του βασίζεται το Διαδίκτυο, αλλά και ένα τεράστιο ποσοστό εμπορικών δικτύων. Η ονομασία του προέρχεται από τις συντομογραφίες των δυο σημαντικότερων πρωτοκόλλων που το απαρτίζουν, το TCP κοινώς Transmission Control Protocol και το IP ή και Internet Protocol. Τα πρωτόκολλα αυτά είναι οργανωμένα σε στρώματα ή layers το καθένα από τα οποία δίνει λύση σε πολλαπλά προβλήματα μεταφοράς δεδομένων παρέχοντας προκαθορισμένες υπηρεσίες στα παραπάνω στρώματα. Τα στρώματα κοντά στην κορυφή είναι πιο κοντά στη λογική του χρήστη ενώ αυτά των χαμηλότερων στρωμάτων μεταφράζουν δεδομένα που διαβιβάζονται με φυσικά μέσα.

2.3 FTP

Το File Transfer Protocol είναι ένα ευρέως χρησιμοποιούμενο πρωτόκολλο δικτύων τα οποία υποστηρίζονται από το πρωτόκολλο TCP/IP. Το τερματικό που τρέχει την FTP client εφαρμογή, μόλις συνδεθεί με τον server, μπορεί να εκτελέσει ένα πλήθος διεργασιών, όπως ανέβασμα και κατέβασμα αρχείων από και προς τον server, μετονομασία και διαγραφή αρχείων.

Ο τρόπος λειτουργίας του είναι ο εξής : ο FTP server ανοίγει την θύρα 21 περιμένοντας από τον αντίστοιχο client να πραγματοποιήσει σύνδεση. Εν συνεχεία ο client ξεκινά τη σύνδεση από μια τυχαία θύρα προς την θύρα του 21 του server. Μόλις η σύνδεση πραγματοποιηθεί με επιτυχία, παραμένει ανοιχτή για όλη τη διάρκεια και ονομάζεται σύνδεση ελέγχου. Μετά τη σύνδεση ελέγχου ακολουθεί η σύνδεση δεδομένων με την οποία ξεκινά η μεταφορά δεδομένων, η οποία μπορεί να είναι είτε ενεργητική είτε παθητική. Στην ενεργητική λειτουργία ο FTP client διαλέγει μια θύρα τυχαία στην οποία θα δεχθεί τα δεδομένα από τον FTP server ενώ όταν η μεταφορά ολοκληρωθεί, η σύνδεση κλείνει από τον server με την αποστολή ενός πακέτου FIN. Η διαδικασία επαναλαμβάνεται κάθε φορά που ο client ζητήσει μια σύνδεση δεδομένων. Στην παθητική λειτουργία ο client ζητά από τον server να διαλέξει μια τυχαία θύρα στην οποία θα πραγματοποιηθεί η σύνδεση δεδομένων. Ο server ενημερώνει τον client ποια θύρα έχει επιλέξει και ο client με τη σειρά του συνδέεται σε αυτή για να ξεκινήσει η διαδικασία της μεταφοράς δεδομένων. Η μεταφορά ολοκληρώνεται με τον ίδιο τρόπο που ολοκληρώνεται και στην ενεργητική λειτουργία καθώς η σύνδεση δεδομένων βασίζεται στο TCP.

Το FTP είναι ένα πρωτόκολλο πελάτη-εξυπηρετητή, ικανό να χειρίζεται οποιονδήποτε τύπο αρχείου χωρίς περαιτέρω επεξεργασία. Ωστόσο παρουσιάζει εξαιρετικά υψηλή καθυστέρηση, γεγονός που σημαίνει ότι ο χρόνος μεταξύ του αιτήματος και της διαδικασίας παραλαβής του είναι πολλές φορές μεγάλος και για αυτό μερικές φορές απαιτείται μεγάλη διαδικασία σύνδεσης.

2.4 UDP

Το User Datagram Protocol, ανήκει όπως και τα προηγούμενα, στα βασικά πρωτόκολλα που χρησιμοποιούνται στο Διαδίκτυο. Διάφορα προγράμματα χρησιμοποιούν το UDP για την αποστολή σύντομων μηνυμάτων (datagrams) μέσα σε ένα δίκτυο, από τον έναν υπολογιστή στον άλλον. Το UDP δεν εγγυάται αξιόπιστη επικοινωνία. Τα πακέτα που αποστέλλονται από έναν υπολογιστή μπορεί να φτάσουν στον παραλήπτη με λάθος σειρά ή και καθόλου. Αντιθέτως το TCP διαθέτει τους απαραίτητους μηχανισμούς ελέγχου και επιβολής της αξιοπιστίας με συνέπεια να μπορεί να εγγυηθεί την αξιόπιστη επικοινωνία μεταξύ των υπολογιστών. Η έλλειψη αυτών των μηχανισμών από το UDP το καθιστά αρκετά γρήγορο και αποτελεσματικό, για τις εφαρμογές που δεν απαιτούν αξιόπιστη επικοινωνία. Έτσι οι εφαρμογές audio και video χρησιμοποιούν πακέτα UDP. Είναι σημαντικό για τις προαναφερθέντες εφαρμογές η λήψη των πακέτων σε σύντομο χρονικό διάστημα ώστε να μην υπάρξει διακοπή στην ροή του ήχου ή της εικόνας του video. Σε περίπτωση που κάποιο πακέτο χαθεί, οι εφαρμογές διαθέτουν ειδικούς μηχανισμούς διόρθωσης και παρεμβολής, έτσι ώστε ο χρήστης να μην παρατηρήσει καμία αλλοίωση ή και διακοπή στη ροή του ήχου ή της εικόνας λόγω κάποιων χαμένων πακέτων.

Το UDP, αντίθετα με το TCP υποστηρίζει broadcasting, την αποστολή δηλαδή ενός πακέτου σε όλους τους υπολογιστές ενός δικτύου και multicasting, την αποστολή δηλαδή ενός πακέτου σε κάποιους συγκεκριμένους υπολογιστές ενός δικτύου. Η δυνατότητα αυτή, χρησιμοποιείται αρκετά συχνά στις εφαρμογές ήχου και βίντεο έτσι ώστε μια ροή ήχου και εικόνας να μπορεί να μεταδίδεται ταυτόχρονα σε πολλούς συνδρομητές. Μερικές από τις σημαντικότερες εφαρμογές που χρησιμοποιούν UDP είναι το Voice over IP (VoIP), Domain Name System (DNS) καθώς και τα παιχνίδια που παίζονται ζωντανά μέσω Διαδικτύου.

2.5 802.11

Το IEEE 802.11 είναι μια οικογένεια προτύπων της IEEE για ασύρματα τοπικά δίκτυα (WLAN) που είχαν ως σκοπό να επεκτείνουν το 802.3 (Ethernet, το συνηθέστερο πρωτόκολλο ενσύρματης δικτύωσης υπολογιστών) στην ασύρματη περιοχή. Τα πρότυπα 802.11 είναι ευρύτερα γνωστά ως «WiFi» επειδή η WiFi Alliance, ένας οργανισμός ανεξάρτητος της IEEE, παρέχει την πιστοποίηση για τα προϊόντα που υπακούν στις προδιαγραφές του 802.11. Αυτή η οικογένεια πρωτοκόλλων αποτελεί το καθιερωμένο πρότυπο της βιομηχανίας στο χώρο των ασύρματων τοπικών δικτύων.

Ο όρος WiFi (Wireless Fidelity) χρησιμοποιείται για να προσδιορίσει τις συσκευές που βασίζονται στην προδιαγραφή IEEE 802.11 b/g/n και εκπέμπουν σε συχνότητες 2.4GHz. Ωστόσο το WiFi («ασύρματη πιστότητα» στα ελληνικά) έχει επικρατήσει και ως όρος αναφερόμενος συνολικά στα ασύρματα τοπικά δίκτυα. Συνήθεις εφαρμογές του είναι η παροχή ασύρματων δυνατοτήτων πρόσβασης στο Internet, τηλεφωνίας μέσω διαδικτύου (VoIP) και διασύνδεσης μεταξύ ηλεκτρονικών συσκευών όπως τηλεοράσεις, ψηφιακές κάμερες, DVD_ player και ηλεκτρονικοί υπολογιστές. Σε φορητές ηλεκτρονικές συσκευές το 802.11 βρίσκει εφαρμογές ασύρματης μετάδοσης, όπως π.χ. τη μεταφορά φωτογραφιών από ψηφιακές κάμερες και σε υπολογιστές για περαιτέρω επεξεργασία και εκτύπωση. Αν και αξίζει να σημειωθεί ότι στον τελευταίο τομέα έχει υποσκελιστεί από το πρωτόκολλο Bluetooth για τα πολύ μικρότερης εμβέλειας ασύρματα προσωπικά δίκτυα.

3. Προσομοίωση Δικτύων

3.1 Ορισμός, Διαχωρισμός Προσομοίωσης-Εξομοίωσης

Ο όρος προσομοίωση (Simulaton) συχνά συγχέεται με τον όρο εξομοίωση (Emulation), αν και οι όροι αυτοί υποδηλώνουν τελείως διαφορετικές μεθοδολογίες. Η προσομοίωση είναι μια μέθοδος μελέτης ενός συστήματος και εξοικείωσης με τα χαρακτηριστικά του με τη βοήθεια ενός άλλου συστήματος το οποίο στις περισσότερες περιπτώσεις είναι ηλεκτρονικός υπολογιστής. Ως εξομοίωση ορίζεται μια μέθοδος αναπαραγωγής ενός συστήματος εντός ή μέσω άλλου συστήματος παρόμοιου με το πρώτο. Είναι λοιπόν εμφανές ότι κατά την προσομοίωση δεν πρέπει να υπάρχει ούτε η εντύπωση ούτε η επιθυμία υλοποίησης του πραγματικού συστήματος, γιατί σκοπός είναι η μελέτη του συστήματος και όχι η χρήση του. Αντίθετα, κατά την εξομοίωση υπάρχει η εντύπωση υλοποίησης του πραγματικού συστήματος καθώς σκοπός είναι η χρήση του. Απλούστερα ένας εξομοιωτής ουσιαστικά μιμείται την λειτουργία ενός συστήματος αντιθέτως ένας προσομοιωτής την αναπαριστά.

Στην προσομοίωση, το υπό μελέτη δίκτυο μοντελοποιείται και μέσα από μαθηματικούς τύπους υπολογίζεται η αλληλεπίδραση των συστατικών του δικτύου. Η μοντελοποίηση είναι ικανή να αναπαράγει έναν πραγματικό φόρτο και μια πραγματική συμπεριφορά μέσα σε μια τοπολογία ενός πραγματικού δικτύου. Όλες οι μεταβλητές μπορούν να τροποποιηθούν έτσι ώστε να αναδειχθεί η συμπεριφορά του δικτύου κάτω από διαφορετικές συνθήκες.

Στην εξομοίωση, πραγματοποιείται μια μελέτη του δικτύου ως προς την απόδοση του έτσι ώστε να προβλεφθεί η επίδραση που θα έχει η όποια αλλαγή σε κάποια παράμετρο λειτουργίας του.

3.2 Λειτουργία προσομοίωσης

Στα δίκτυα υπολογιστών, η προσομοίωση είναι μια τεχνική όπου ένα πρόγραμμα μοντελοποιεί την συμπεριφορά ενός δικτύου είτε υπολογίζοντας την αλληλεπίδραση μεταξύ των οντοτήτων (host/πακέτα) χρησιμοποιώντας μαθηματικές φόρμουλες, είτε συλλαμβάνοντας και αναπαριστώντας τις παρατηρήσεις από ένα δίκτυο. Η συμπεριφορά ενός δικτύου και οι διαφορετικές εφαρμογές και υπηρεσίες που υποστηρίζει μπορούν να παρατηρηθούν μέσω οπτικοποίησης των αποτελεσμάτων της προσομοίωσης. Διαφορετικά χαρακτηριστικά του συστήματος μπορούν να τροποποιηθούν έτσι ώστε να προσδιοριστεί πως θα συμπεριφερόταν το δίκτυο κάτω από διαφορετικές συνθήκες.

Ένας δικτυακός προσομοιωτής είναι ένα λογισμικό το οποίο προβλέπει την συμπεριφορά ενός δικτύου υπολογιστών. Στους προσομοιωτές, ένα δίκτυο υπολογιστών μοντελοποιείται μέσω συσκευών, συνδέσεων, εφαρμογών με σκοπό να αναλυθεί η εκτέλεσή του. Στην συνέχεια, οι χρήστες έχουν την δυνατότητα να προσαρμόσουν την προσομοίωση για να ικανοποιήσουν τις ανάγκες τους. Οι προσομοιωτές υποστηρίζουν τα πιο γνωστά πρωτόκολλα και δίκτυα που χρησιμοποιούνται σήμερα, όπως WLAN, WI-MAX, TCP, WSN.

Οι περισσότεροι προσομοιωτές που κυκλοφορούν στο εμπόριο είναι GUI Driven και CLI Driven. Τα αρχικά G.U.I. σημαίνουν *graphical user interface* πράγμα που παραπέμπει σε προσομοιωτές οι οποίοι χρησιμοποιούν γραφική διεπαφή για την διεξαγωγή της προσομοίωσης. C.L.I σημαίνει *command-line interface* όπου ο χρήστης αλληλεπιδρά στο πρόγραμμα μέσω εντολών που πληκτρολογεί (*command lines*).

Το δικτυακό μοντέλο περιγράφει την κατάσταση ενός δικτύου (κόμβοι, δρομολογητές, συνδέσεις) και τα γεγονότα (μετάδοση δεδομένων, packet error). Μια

σημαντική έξοδος της προσομοίωσης είναι τα αρχεία trace. Τα trace files καταγράφουν το κάθε πακέτο , το κάθε γεγονός που συμβαίνει στην προσομοίωση και το χρησιμοποιούν για ανάλυση. Οι προσομοιωτές μπορούν επίσης να παρέχουν άλλα εργαλεία για να οπτικοποιήσουν την ανάλυση των τάσεων και τον πιθανόν προβληματικών σημείων.

Οι πιο λεπτομερείς (και αργοί) προσομοιωτές χρησιμοποιούν προσομοίωση της ροής των ψηφίων (κάθε byte) όπως αυτά αλλάζουν κατεύθυνση μέσα στο δίκτυο. Οι πιο γρήγοροι προσομοιωτές χρησιμοποιούν ρολόγια (logical clocks) τα οποία συλλέγουν χρονολογικά τυχαίες σχέσεις σε ένα κατανεμημένο σύστημα.

Οι περισσότεροι δικτυακοί προσομοιωτές χρησιμοποιούν προσομοίωση διακριτών γεγονότων, στην οποία μοντελοποιείται η λειτουργία ενός συστήματος ως μια διακριτή αλληλουχία γεγονότων στο χρόνο. Κάθε γεγονός λαμβάνει χώρα σε μια συγκεκριμένη χρονική στιγμή και σηματοδοτεί μια αλλαγή της κατάστασης του συστήματος. Αντίθετα στην συνεχή προσομοίωση παρακολουθείται συνεχώς η δυναμική του συστήματος στην πάροδο του χρόνου, πράγμα το οποίο καθιστά την συνεχή προσομοίωση πιο αργή συγκριτικά με αυτή των διακριτών γεγονότων.

Η προσομοίωση δικτύων είναι ένα πολύ σύνθετο έργο. Ο χρόνος που μπορεί να χρειάζεται για να πάρουμε μια ακριβή απάντηση μπορεί να είναι μεγάλος σε μια πιθανή υπερχείλιση σε ένα δίκτυο. Για αυτό το λόγο υπάρχουν εξειδικευμένες τεχνικές, όπως οι μεταβλητές ελέγχου και η δειγματοληψία ώστε να επιταχυνθεί η διαδικασία.

3.3 Χρήσεις

Οι δικτυακοί προσομοιωτές εξυπηρετούν ποικίλες ανάγκες. Σε σύγκριση με τον χρόνο και το κόστος που απαιτείται για την σύσταση ενός πραγματικού δικτύου με δικτυωμένους υπολογιστές, δρομολογητές και συνδέσεις, η προσομοίωση είναι μια σχετικά γρήγορη και ανέξοδη λύση. Δίνει την δυνατότητα σε μηχανικούς και ερευνητές να δοκιμάσουν σενάρια τα οποία θα μπορούσαν να είναι ιδιαίτερα δύσκολα και ακριβά στην υλοποίηση με πραγματικό hardware- για παράδειγμα η προσομοίωση εν μέσω πολλαπλών κόμβων και νέων πρωτοκόλλων σε ένα δίκτυο.

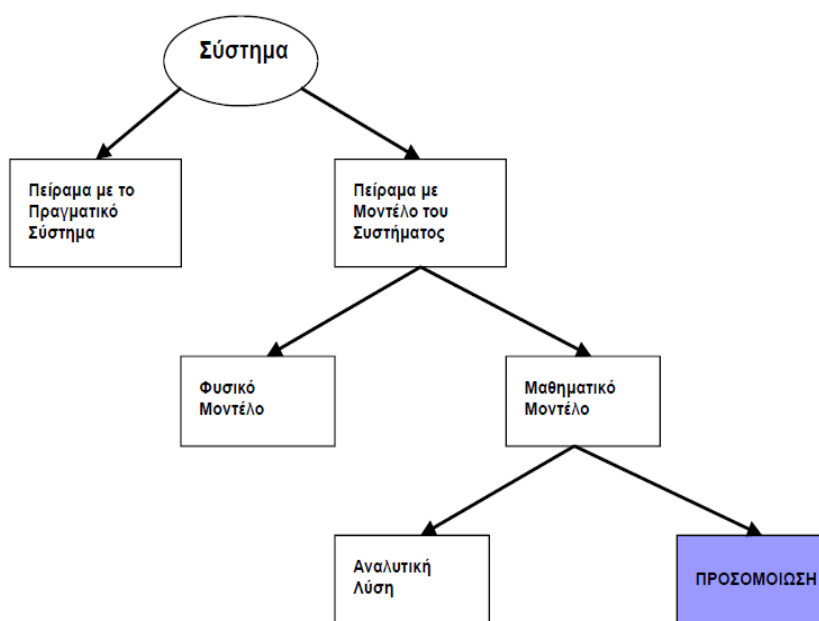
Έτσι οι προσομοιωτές δικτύων είναι ιδιαίτερα χρήσιμα εργαλεία επιτρέποντας στους χρήστες το τεστάρισμα νέων πρωτοκόλλων ή ακόμα και τις αλλαγές σε ήδη υπάρχοντα σε ένα ελεγχόμενο αναπαραγωγίμο περιβάλλον.

Ένας απλός δικτυακός προσομοιωτής καλύπτει ένα ευρύ φάσμα των τεχνολογιών δικτύωσης και μπορεί να βοηθήσει τους χρήστες στη δημιουργία πολύπλοκων δικτύων, από τα βασικά δομικά στοιχεία όπως διαφορετικούς κόμβους, γέφυρες, συνδέσεις, δρομολογητές, κινητές και σταθερές μονάδες κ.λ.π.. Διάφοροι τύποι δικτύων (ευρείας περιοχής, τοπικά δίκτυα), αλλά και διάφοροι τύποι δικτυακών πρωτοκόλλων (Tcp, Udp) μπορούν να προσομοιωθούν με έναν απλό προσομοιωτή, με τον χρήστη να έχει τη δυνατότητα να πειραματιστεί και να αναλύσει τα εξαχθέντα αποτελέσματα.

Υπάρχει μια μεγάλη ποικιλία δικτυακών προσομοιωτών, από τον πιο απλό στον πιο περίπλοκο. Ο χρήστης έχει την δυνατότητα να αναπαραστήσει μια τοπολογία ενός δικτύου καθορίζοντας τους κόμβους, τους δεσμούς μεταξύ αυτών όπως επίσης και την κίνηση που επιθυμεί. Τα πιο πολύπλοκα συστήματα επιτρέπουν στον χρήστη να καθορίσει τα πάντα σχετικά με τα πρωτόκολλα και την κυκλοφορία σε ένα δίκτυο ενώ με τις γραφικές αναπαραστάσεις δίνουν την δυνατότητα της απεικόνισης της

προσομοίωσης

Αξίζει να σημειωθεί, ότι ως σύστημα, ορίζουμε μία συλλογή οντοτήτων (π.χ. άνθρωποι ή μηχανές) που ενεργούν και αλληλεπιδρούν, με στόχο κάποιο λογικό τερματισμό.



Εικόνα 1 : Τρόποι μελέτης ενός συστήματος

3.4 Είδη Προσομοιωτών

Πριν αναλύσουμε τα είδη των δικτυακών προσομοιωτών, θα δώσουμε μερικούς βασικούς ορισμούς εννοιών που θα χρησιμοποιήσουμε στις επόμενες παραγράφους.

Οντότητες : Αυτοτελή στοιχεία του συστήματος, τα οποία προσομοιώνονται και η συμπεριφορά τους ορίζεται επακριβώς. Το πρόγραμμα που υλοποιεί το μοντέλο προσομοίωσης τον ηλεκτρονικό υπολογιστή, διατηρεί πληροφορίες σχετικά με τις οντότητες . Σε περίπτωση που συμβεί η οποιαδήποτε αλλαγή στην κατάσταση της οντότητας, το πρόγραμμα την καταγράφει.

Πόροι : Όπως και οι οντότητες, και οι πόροι είναι αυτοτελή στοιχεία του συστήματος αλλά δεν μοντελοποιούνται μεμονωμένα. Οι πόροι έχουν περιορισμένη διαθεσιμότητα. Για παράδειγμα σε ένα σύστημα, πόροι θα μπορούσαν να είναι τα διαθέσιμα κουτιά ενός προ'όντος σε μια αποθήκη ή οι ταμίες σε μια τράπεζα.

Ρολόι προσομοίωσης : Το ρολόι προσομοίωσης δείχνει πάντα τον τρέχων χρόνο προσομοίωσης, ο οποίος όμως είναι πάντα διαφορετικός από τον πραγματικό χρόνο. Συνήθως λίγα λεπτά πραγματικού χρόνου αντιστοιχούν σε μεγάλα προσομοιούμενα χρονικά διαστήματα.

Γεγονός : Ως γεγονός ορίζεται η συγκεκριμένη χρονική στιγμή κατά την οποία πραγματοποιείται αλλαγή στην κατάσταση μιας ή περισσότερων μεταβλητών.

Δραστηριότητα : Η δραστηριότητα καταναλώνει χρόνο και περιορίζεται από ένα αρχικό και ένα τελικό γεγονός. Παράδειγμα μια εξυπηρέτηση ενός πελάτη από έναν υπάλληλο η οποία χαρακτηρίζεται από το αρχικό γεγονός : αρχή εξυπηρέτησης.

Διαδικασία : Οι διαδικασίες περιέχουν διάφορα γεγονότα και δραστηριότητες ενώ

ουσιαστικά περιγράφουν το ιστορικό μιας οντότητας μέσα στο σύστημα που προσομοιώνεται.

Αφού αποσαφηνίσαμε τους παραπάνω σημαντικούς όρους, ας δούμε τώρα τα είδη των δικτυακών προσομοιωτών.

- **Discrete Event Simulators**

Στους προσομοιωτές διακριτών γεγονότων (DES), τα μοντέλα αποτελούνται από οντότητες, πόρους (στοιχεία που εξυπηρετούν τις οντότητες) καθώς επίσης και από στοιχεία ελέγχου (στοιχεία που καθορίζουν την κατάσταση των οντοτήτων και των πόρων). Οι προσομοιωτές αυτής της κατηγορίας έχουν σχεδιαστεί για την προσομοίωση διεργασιών όπως τηλεφωνικά κέντρα, λειτουργίες εργοστασίων και ναυτιλιακών εγκαταστάσεων όπου η πληροφορία που προσομοιώνεται μπορεί να περιγραφεί καθώς κινείται σε διακριτά βήματα ή πακέτα. Συνεπώς δεν προορίζονται για την μοντελοποίηση συνεχών συστημάτων (π.χ. κίνηση νερού).

Οι DES χρησιμοποιούνται συνήθως για την παρακολούθηση και την πρόβλεψη της συμπεριφοράς ενός συστήματος, όπως ένα δίκτυο κάτω από δύσκολες συνθήκες.

- **Agent based simulators**

Αυτή είναι η ειδική κατηγορία των προσομοιωτών διακριτών γεγονότων στην οποία οι κινητές οντότητες ορίζονται σαν *agents*. Ενώ σε ένα μοντέλο διακριτών γεγονότων οι οντότητες έχουν μόνο κάποιες ιδιότητες, που μπορούν να ελέγχουν τον τρόπο με τον οποίο αλληλεπιδρούν με τους πόρους και τα στοιχεία ελέγχου, οι *agents* έχουν και ιδιότητες όπως και μεθόδους (κανόνες που τους επιτρέπουν να αλληλεπιδρούν με άλλους *agents*). Για παράδειγμα ένα agent-based μοντέλο, θα μπορούσε να προσομοιώσει οικονομικές αναλύσεις και πειράματα στην βιολογία.

- **Continuous Simulators**

Αυτή η κατηγορία εργαλείων, λύνει προσομοιώνει ένα σύστημα το οποίο χρησιμοποιεί συνεχείς εξισώσεις. Το εν λόγω είδος προσομοιωτών είναι το πλέον κατάλληλο εάν η πληροφορία που προσομοιώνεται μπορεί να περιγραφεί ως εξελισσόμενη και κινούμενη ομαλά και συνεχώς. Οι συνεχείς προσομοιωτές μπορούν επίσης να χρησιμοποιηθούν για την προσομοίωση συστημάτων που αποτελούνται από διακριτές οντότητες, εάν ο αριθμός αυτών είναι μεγάλος, έτσι ώστε η κίνηση να μπορεί να θεωρηθεί ως μια ροή.

- **Hybrid Simulators**

Τα εργαλεία που συνδυάζουν τα χαρακτηριστικά των συνεχών προσομοιωτών και αυτών των διακριτών γεγονότων. Δηλαδή, λύνουν διαφορετικές εξισώσεις, αλλά μπορούν να θέσουν υπεράνω τα διακριτά γεγονότα σε ένα συνεχώς μεταβαλλόμενο σύστημα.

Από κει και πέρα ανεξαρτήτως σε ποια κατηγορία ανήκουν, κάποιοι από τους προσομοιωτές δικτύου είναι εμπορικοί, πράγμα που σημαίνει ότι δεν παρέχουν τον πηγαίο κώδικα του λογισμικού για τους χρήστες δωρεάν. Όλοι οι χρήστες πρέπει να πληρώσουν για να πάρουν την άδεια χρήσης του λογισμικού ή ακόμα και κάποιον συγκεκριμένων πακέτων για να καλύψουν ανάγκες τους. Ένα χαρακτηριστικό παράδειγμα προσομοιωτή σε αυτή την κατηγορία είναι ο OPNET.

Οι εμπορικοί προσομοιωτές έχουν τα θετικά τους όπως επίσης και τα αρνητικά τους.

Το σημαντικότερο πλεονέκτημα είναι ότι συντηρούνται και ενημερώνονται από εξειδικευμένο προσωπικό της κάθε εταιρίας. Αντιθέτως οι προσομοιωτές ανοικτού κώδικα μειονεκτούν σε αυτό τον τομέα καθώς δεν υπάρχουν εξειδικευμένα άτομα τα οποία ασχολούνται με τον παραπάνω τομέα. Αυτό το πρόβλημα μπορεί να είναι σημαντικό όταν δημοσιοποιούνται οι διαφορετικές νέες εκδόσεις καθώς θα καθίσταται δύσκολη η κατανόηση των παλαιότερων. Στους προσομοιωτές ανοικτού κώδικα υπάρχει το πλεονέκτημα ότι πάντα θα είναι προσβάσιμοι από τον οποιοδήποτε και ο καθένας μπορεί να συμβάλει στην εύρεση σφαλμάτων σε αυτούς. Το περιβάλλον είναι επίσης ανοιχτό για μελλοντικές βελτιώσεις, ενώ μπορούν να υποστηρίξουν τις εξελίξεις πάνω στον τομέα με πολύ γρήγορο τρόπο, σε σχέση με τους εμπορικούς προσομοιωτές. Συμπεραίνεται ότι τα πλεονεκτήματα των εμπορικών προσομοιωτών είναι μειονεκτήματα για αυτούς του ανοικτού κώδικα και αντίθετα. Οι πιο διάσημοι προσομοιωτές ανοικτού κώδικα είναι οι NS2 και NS3 οι οποίοι θα αναλυθούν με λεπτομέρεια στο επόμενο κεφάλαιο.

Επί του παρόντος υπάρχει μια μεγάλη ποικιλία από δικτυακούς προσομοιωτές εκ των οποίων υπάρχουν απλοί (λειτουργία, κώδικα, περιβάλλον) και πιο σύνθετοι. Ένας προσομοιωτής δικτύου πρέπει να επιτρέπει στους χρήστες να αναπαριστούν μια τοπολογία δικτύου, τον καθορισμό των σεναρίων, προσδιορίζοντας τους κόμβους στο δίκτυο, τους δεσμούς μεταξύ των κόμβων καθώς επίσης και την κίνηση μεταξύ αυτών. Τα πιο πολύπλοκα συστήματα επιτρέπουν στο χρήστη να καθορίσει τα πάντα γύρω από τα πρωτόκολλα που χρησιμοποιούνται για την επεξεργασία της κίνησης του δικτύου. Γραφικές εφαρμογές επιτρέπουν επίσης στους χρήστες να απεικονίσουν εύκολα το προσομοιωμένο σενάριό τους. Άλλοι προσομοιωτές μπορεί να χρησιμοποιούν αντικειμενοστραφή προγραμματισμό, παρέχοντας στον χρήστη τη δυνατότητα να προσαρμόσει και να δημιουργήσει μια εφαρμογή να που προσομοιώνει το περιβάλλον δικτύωσης για δοκιμή.

3.5 Πλεονεκτήματα

Με την χρήση της προσομοίωσης, οι χρήστες μπορούν να μελετήσουν τις σχέσεις μεταξύ των συστατικών ενός δικτύου με λεπτομέρεια και να προσομοιώσουν τις προβλεπόμενες συνέπειες των πολλαπλών επιλογών σχεδιασμού του, προτού χρειαστεί να εφαρμόσουν την εφαρμογή των αποτελεσμάτων σε πραγματικό χρόνο. Καθίσταται δυνατή η σύγκριση εναλλακτικών σχεδίων με σκοπό την επιλογή του βέλτιστου, ενώ τα αποτελέσματα των προσομοιώσεων είναι πολύ ακριβή.

Η διαδικασία της ανάπτυξης της προσομοίωσης μπορεί να παρέχει πολύτιμες πληροφορίες για τις εσωτερικές διεργασίες του δικτύου, οι οποίες μπορούν να χρησιμοποιηθούν και σε μεταγενέστερο στάδιο. Οι πληροφορίες αυτές, μπορούν να βοηθήσουν τους χρήστες στην εύρεση μιας μη θεμιτής συμπεριφοράς του συστήματος.

Σε γενικές γραμμές μπορούμε να ορίσουμε ότι παραμέτρους θέλουμε σε μια προσομοίωση, λαμβάνοντας πολύ λεπτομερή αποτελέσματα κατά την έξοδο τα οποία δεν μπορούμε να τα μελετήσουμε πειραματικά με άλλη μέθοδο με την μέχρι τώρα εξέλιξη της τεχνολογίας.

3.6 Μειονεκτήματα

Παρά τα σημαντικά πλεονεκτήματα που προσφέρει μια προσομοίωση υπάρχουν και σημαντικά μειονεκτήματα τα οποία μπορούν να δυσκολέψουν την διεξαγωγή των πειραμάτων. Για την ανάπτυξη ενός προσομοιωτή και την ακριβή και αξιόπιστη λειτουργία του απαιτούνται σημαντικοί πόροι, όπως χρήματα και εργατοώρες.

Επιπρόσθετα οι παράμετροι που επηρεάζουν ένα πραγματικό δίκτυο είναι πάρα πολλοί, πράγμα που καθιστά αδύνατο από ένα μοντέλο προσομοίωσης να τους συμπεριλάβει όλους, όσο καλό και αν αυτό είναι. Έτσι για να φτάσει ένα μοντέλο προσομοίωσης την βέλτιστη μορφή, απαιτείται πολύ προσπάθεια ενώ τα αποτελέσματα, όσο επιτυχημένη και αν είναι η προσομοίωση, θα παραμένουν μόνο εκτιμήσεις/προβλέψεις.

Είναι δύσκολο να συνηθίσει κάποιος μηχανικός τον εικονικό κόσμο στο σχεδιασμό ενός δικτύου, αλλά μετά την πρώτη επαφή ο καθένας έχει την αίσθηση ότι τα όρια στην προσομοίωση είναι τεράστια.

3.7 Επισκόπηση εξελίξεων

Σήμερα υπάρχουν πολλοί προσομοιωτές δικτύου που έχουν διαφορετικά χαρακτηριστικά όσον αφορά την λειτουργία τους. Μια σύντομη λίστα με τους πιο γνωστούς προσομοιωτές δικτύου περιλαμβάνει : OPNET, NS-2, NS-3, OMNET, REAL, SSFNET, J-SIM και πολλούς άλλους στους οποίους θα γίνει αναφορά στο επόμενο κεφάλαιο. Από τους προαναφερθέντες, ο OPNET είναι ο πιο γνωστός προσομοιωτής εμπορίου. Ο NS-2 είναι ο πιο δημοφιλής στον ακαδημαϊκό χώρο, λόγω του ότι είναι ανοικτού κώδικα και της αφθονίας των στοιχείων στη βιβλιοθήκη του. Πολλοί οργανισμοί συμβάλλουν στη ανάπτυξη της βιβλιοθήκης του και αυτό έχει αποδεχθεί ως ένας επιτυχής τρόπος ανάπτυξης. Ωστόσο, λόγω κάποιων φυσικών περιορισμών στη σχεδίαση του NS-2 έχουμε την εξέλιξη NS-3 ο οποίος σχεδιάστηκε λαμβάνοντας υπόψιν τα αρνητικά και τα θετικά του προκατόχου του NS-2.

4. Εργαλεία προσομοίωσης

Τα τηλεπικοινωνιακά συστήματα είναι σύνθετα συστήματα του πραγματικού κόσμου που περιλαμβάνουν διαφορετικά χαρακτηριστικά τα οποία αλληλεπιδρούν , σε σύνθετες αλληλεξαρτήσεις. Η ανάλυση τέτοιων συστημάτων μπορεί να γίνει πολύ δύσκολη : οι τεχνικές μοντελοποίησης τείνουν να αναλύουν το κάθε συστατικό ξεχωριστά παρά τις σχέσεις μεταξύ αυτών. Η προσομοίωση είναι μια προσέγγιση η οποία μπορεί να χρησιμοποιηθεί ώστε να μοντελοποιηθούν μεγάλα συστήματα, για προβλέψεις, μετρήσεις και αποδοτικούς σκοπούς.

Η επιλογή της προσομοίωσης ως εργαλείο μοντελοποίησης γίνεται συνήθως επειδή είναι λιγότερο περιοριστική.

Η δικτυακή προσομοίωση συνήθως ακολουθάει τα παρακάτω βήματα :

- μοντελοποίηση του συστήματος σαν μια στοχαστική διαδικασία (δηλαδή τυχαία).
- "γέννηση" των αποτελεσμάτων της στοχαστικής διαδικασίας
- μέτρηση των αποτελεσμάτων της προσομοίωσης
- ανάλυση των δεδομένων εξόδου

Στις υποενότητες του κεφαλαίου θα γίνει αναφορά σε μερικά από τα πιο γνωστά εργαλεία προσομοίωσης.

4.1 OMNET

Το Optical Micro-Networks είναι ένας αντικειμενοστραφής προσομοιωτής διακεκριμένων γεγονότων (Discrete Event Simulator – DES) με *G.U.I.* υποστήριξη. Διαθέτει μια «γενική» αρχιτεκτονική έτσι μπορεί να χρησιμοποιηθεί σε διάφορους τομείς όπως:

- Μοντελοποίηση ασύρματων και ενσύρματων δικτύων επικοινωνιών.

- Μοντελοποίηση πρωτοκόλλων.
- Μοντελοποίηση δικτύων ουρών.
- Μοντελοποίηση μικροεπεξεργαστών και άλλων συστημάτων hardware.

Γενικά μπορεί να χρησιμοποιηθεί για την προσομοίωση οποιουδήποτε συστήματος για το οποίο είναι κατάλληλη η προσέγγιση των διακεκριμένων γεγονότων και το οποίο μπορεί να αντιστοιχηθεί σε οντότητες που επικοινωνούν μεταξύ τους ανταλλάσσοντας μηνύματα. Η προσομοίωση υλοποιείται σε C++ .

4.2 J-SIM

Ο J-Sim είναι ένας Java-based προσομοιωτής με εικονικό περιβάλλον , ο οποίος υποστηρίζει Web-based προσομοιώσεις οι οποίες εξελίσσονται ταχύτατα στον αναδυόμενο τομέα της έρευνας και της ανάπτυξης των προσομοιώσεων . Για την δικτυακή προσομοίωση ορίζεται ένα δίκτυο μεταγωγής πακέτων. Το μοντέλο αυτό ορίζει μια γενική δομή για ένα κόμβο, είτε αυτός είναι ένας τελικός host είτε είναι δρομολογητής, καθώς και τα γενικά συστατικά του δικτύου, τα οποία μπορούν να χρησιμοποιηθούν σαν βασικές κλάσεις προκειμένου να υλοποιηθούν πρωτόκολλα πολλαπλών επιπέδων. Επιπρόσθετα, το μοντέλο αυτό είναι ικανό να εξυπηρετήσει πολλαπλές δικτυακές αρχιτεκτονικές, όπως είναι η IETF, η δικτυακή αρχιτεκτονική των κινητών ασύρματων δικτύων καθώς και η δικτυακή αρχιτεκτονική των WDM-based οπτικών δικτύων.

Τα πιο σημαντικά χαρακτηριστικά του J-Sim είναι τα ακόλουθα:

- Ένα προγραμματιστικό μοντέλο βασισμένο σε components.
- Μια προσομοίωση πραγματικού χρόνου που βασίζεται κατά κύριο λόγο σε διεργασίες.
- Ένα αποτελεσματικό σύνολο από Internet Integrated, Differentiated, και Best Effort Services πρωτόκολλα.
- Ένα δίγλωσσο περιβάλλον στο οποίο οι κλάσεις είναι γραμμένες σε Java και

δένονται μεταξύ τους μέσω Tcl/Java.

- Πρέπει να σημειωθεί πως ο δικτυακός αυτός προσομοιωτής έχει υλοποιηθεί εξολοκλήρου σε Java παρά το γεγονός ότι η τρέχουσα έκδοση του επαυξήθηκε με μια Java υλοποίηση του TCL διερμηνέα που αποκαλείται Jacl.

4.3 SSFNET

Το SSFNet αποτελεί ένα πλαίσιο προσομοίωσης το οποίο βασίστηκε στο Scalable Simulation Framework. Το SSF (Scalable Simulation Framework) αποτελεί ένα κοινό πεδίο ορισμού για γεγονότα διακριτού χρόνου μεγάλων, πολύπλοκων συστημάτων σε Java και C++.

Πιο συγκεκριμένα, το SSFNET αποτελεί μια συλλογή από μοντέλα βασισμένα στο SSF που χρησιμοποιείται για την προσομοίωση πρωτοκόλλων και δικτύων. Κάθε κλάση του SSFNET μπορεί να σχηματιστεί αυτόνομα ρωτώντας κάποια βάση, η οποία μπορεί να τρέχει είτε τοπικά είτε στο Web. Τα configuration files του δικτύου είναι σε DML μορφή (Dimensional Markup Language, μορφή XML και χρησιμοποιούνται για την σύνθεση ενός μοντέλου καθώς επίσης και για την αρχικοποίηση μιας προσομοίωσης με τη βοήθεια ενός επεκτάσιμου configuration package για την βάση.

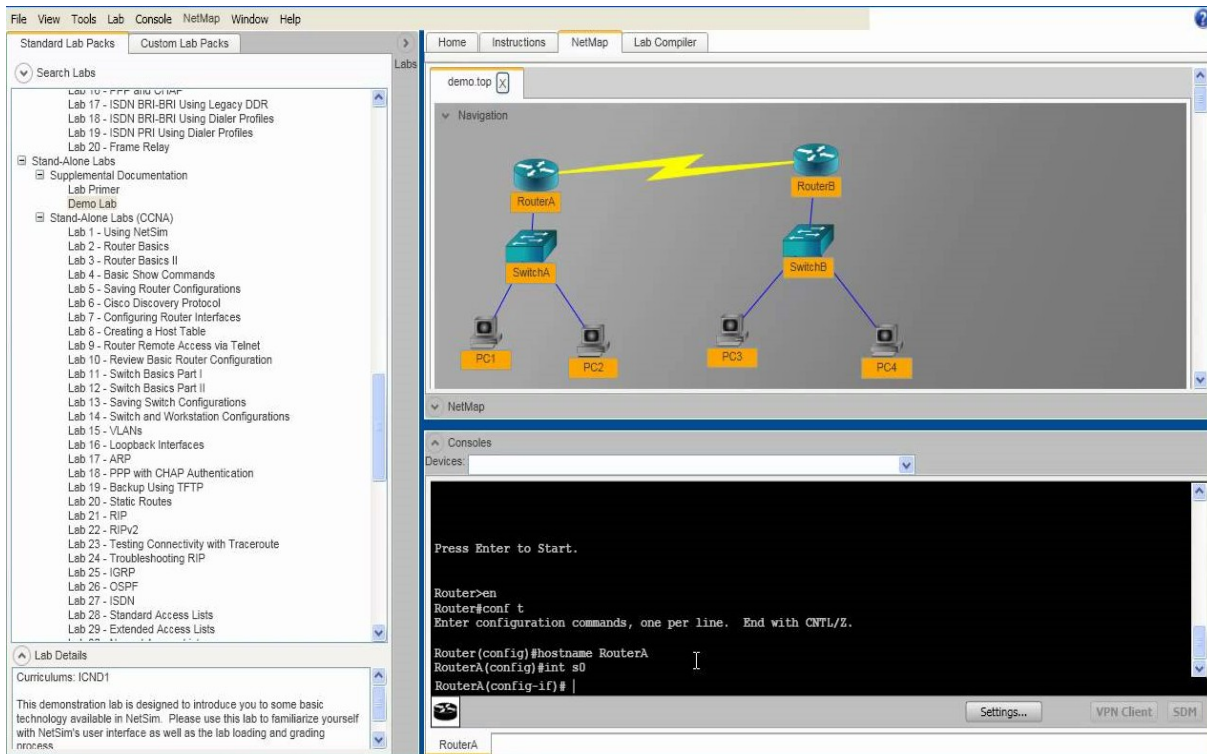
Προκειμένου να μελετηθεί η δυναμική της IP κυκλοφορίας σε μη ασύρματα δίκτυα, το SSFNET σχεδιάστηκε στο λογικό IP-δικτυακό επίπεδο, και το κβάντο της προσομοίωσης είναι ένα IP πακέτο. Στην περίπτωση αυτή, ένα SSFNET μοντέλο αποτελείται από δρομολογητές, απομονωμένα LAN, και συνδέσμους ευρείας περιοχής, αλλά αγνοεί τις λεπτομέρειες σε επίπεδο μετάδοσης συνδέσμων όταν υπάρχει μεγάλο εύρος ζώνης και καθυστερήσεις μετάδοσης πάνω στους συνδέσμους. Η μοντελοποίηση του επιπέδου των συνδέσμων και του φυσικού επιπέδου μπορούν να δοθούν σε διαφορετικά πακέτα. Παρά αυτές τις απλοποιήσεις, το SSFNET μπορεί

να προσομοιώσει την συμπεριφορά του TCP με μεγάλη λεπτομέρεια.

Πρέπει τέλος να αναφερθεί πως υπάρχουν πολλαπλές υλοποιήσεις του SSF και ένας μεγάλος αριθμός από open-source μοντέλα πρωτοκόλλων καθώς και άλλα components.

4.4 NETSIM

Ο Network Based Environment for Modelling and Simulation είναι ένας προσομοιωτής διακριτών γεγονότων που αναπτύχθηκε από την Tetcos, σε συνεργασία με το Ινδικό Ινστιτούτο Επιστημών, με την πρώτη έκδοση του τον Ιούνιο του 2002. Το WiMax, το Token Ring, το WLAN, το TCP και το UDP το GSM και το WSN είναι κάποια από τα πρωτόκολλα τα οποία υλοποιούνται στον Netsim μέσω της γλώσσας C. Χρησιμοποιείται για τα δίκτυα στα οποία υπάρχει ανταλλαγή μηνυμάτων ενώ λειτουργεί με βάση τα γεγονότα (events). Η προσομοίωση τρέχει αφού ο χρήστης προγραμματίσει σε C τα στοιχεία τα οποία συνδέονται μετά το compile. Η δημιουργία της προσομοίωσης πραγματοποιείται από το toolkit, μια βιβλιοθήκη με συναρτήσεις της C η οποία φορτώνει και εμφανίζει τις δικτυακές ρυθμίσεις. Για να τρέξει ο χρήστης μια προσομοίωση πρέπει να γράψει σε C καινούρια στοιχεία και αφού αλλάξει μερικά στοιχεία του προγράμματος να κάνει compile. Πρόκειται για μια εφαρμογή που προσομοιώνει δικτυακά συστήματα Cisco, με σκοπό την εκμάθηση των λειτουργιών τους στους χρήστες.



Εικόνα 2 : Δικτυακός προσομοιωτής NetSim

4.5 TED

Ένα μοντέλο στον TED αποτελείται από οντότητες που επικοινωνούν χρησιμοποιώντας γεγονότα . Κάθε οντότητα αντιπροσωπεύει κάποιο φυσικό αντικείμενο του δικτύου ενώ τα γεγονότα, τα μηνύματα μεταξύ των οντοτήτων, που φέρουν την πληροφορία για την συμπεριφορά της προσομοίωσης. Τα μηνύματα ταξιδεύουν μέσα από τα κανάλια. Η επικοινωνία πραγματοποιείται μέσω την ανταλλαγή events, πάνω από προκαθορισμένα από το χρήστη δρομολογημένα κανάλια. Τα εκτελέσιμα αρχεία του TED μπορούν να ενσωματωθούν σε κώδικα C++ μέσω ενός compiler που τα μεταφράζει.

4.6 NEST

Είναι ένα εργαλείο προσομοίωσης που αναπτύχθηκε το 2004 , γραμμένο σε C++ και Python το οποίο ασχολείται με νευρωνικά δίκτυα μεγάλης κλίμακας. Το νευρωνικό σύστημα ορίζεται από ένα μεγάλο αριθμό νευρώνων και των συνδέσεών τους. Ο χρήστης δημιουργεί ένα σύστημα μέσω νευρώνων τους οποίους συνδέει ώστε να σχηματίσει ένα δίκτυο κατασκευασμένο από πολλά υποδίκτυα. Ειδικοί κόμβοι που ονομάζονται συσκευές , είναι υπεύθυνοι για την λήψη μετρήσεων από το δίκτυο.

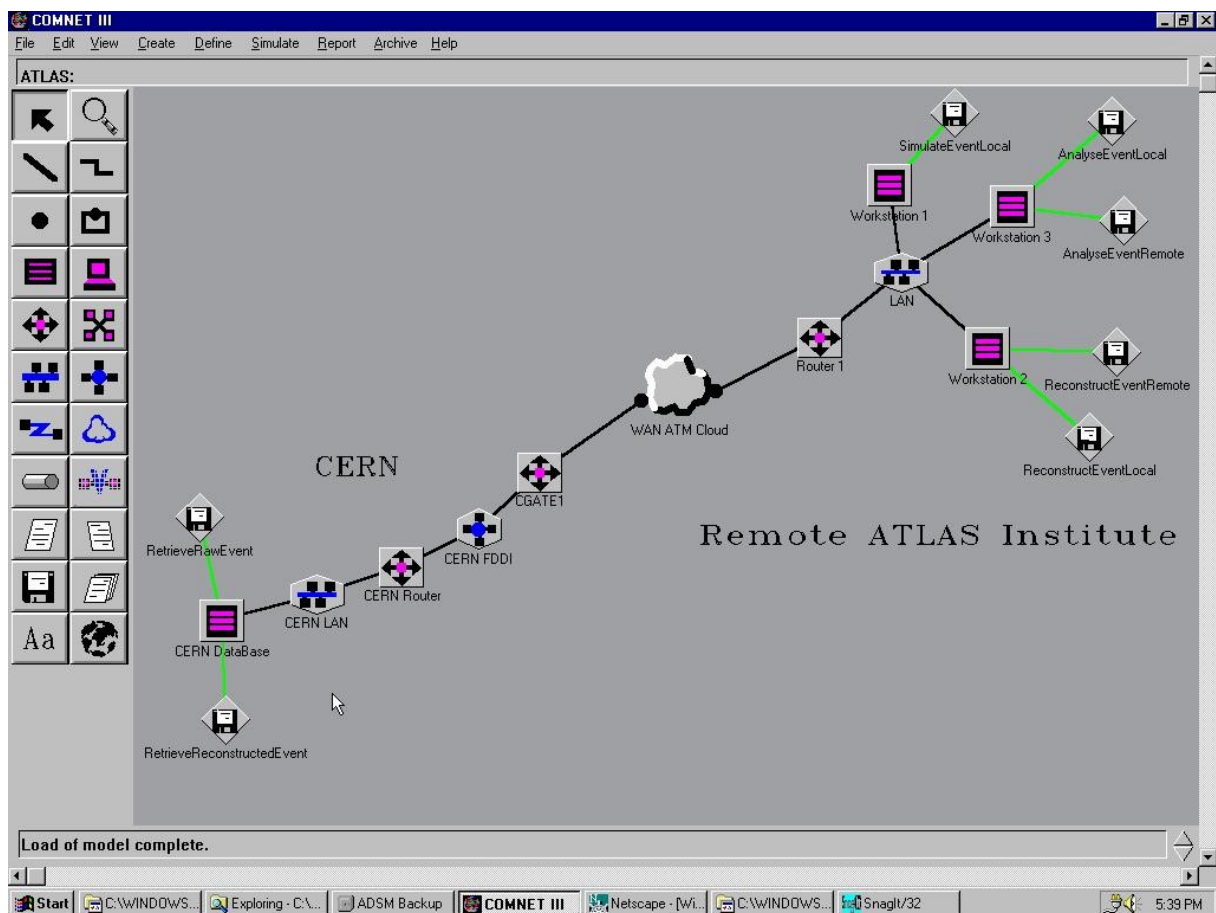
Οι μετρήσεις αυτές γράφονται σε αρχεία έτσι ώστε να μπορούν να αναλυθούν από εργαλεία όπως το MATLAB και το MATHEMATICA. Η προσομοίωση στον NEST μπορεί να επιταχυνθεί μέσω της χρήσης υπολογιστικών συστημάτων πολλαπλών επεξεργαστών και την αύξηση της διαθέσιμης μνήμης.

4.7 COMNET III

Πρόκειται για μια εμπορική εφαρμογή, η οποία επιτρέπει στους χρήστες την εκτίμηση των χαρακτηριστικών και της απόδοσης ενός δικτύου υπολογιστών. Η περιγραφή του δικτύου δημιουργείται γραφικά, χρησιμοποιώντας μια διεπαφή , ενώ δεν απαιτείται προγραμματισμός από τον χρήστη.

Προτιμάται κυρίως από σχεδιαστές δικτύων ώστε να υπολογίσουν την επίδοση LAN, WAN, σύνθετων εσωτερικών δικτύων επιχειρήσεων μέσω της προσομοίωσης. Είναι ένας αρκετά καλός προσομοιωτής ο οποίος μοντελοποιεί την επίδοση των δικτύων ενώ μειώνει το ρίσκο λαθών. Ο COMNET III γράφτηκε σε MODSIM II χρησιμοποιώντας αντικειμενοστραφή προγραμματισμό ενώ χρησιμοποιεί δομικά στοιχεία, τα χαρακτηριστικά των οποίων ταιριάζουν με εκείνα του εξοπλισμού που

βρίσκονται σε ένα πραγματικό παγκόσμιο δίκτυο. Αυτά τα δομικά στοιχεία τα οποία περιλαμβάνουν υπολογιστές, δρομολογητές και συνδέσμους επικοινωνίας, μπορούν να διαμορφωθούν, έτσι ώστε να καθοριστούν τα χαρακτηριστικά του δικτύου που είναι επιθυμητό να προσομοιωθεί. Το βασικό βήμα οικοδόμησης ενός μοντέλου είναι να οριστεί η τοπολογία του δικτύου χρησιμοποιώντας κόμβους και συνδέσεις, τα οποία διατίθενται στην εφαρμογή. Στη συνέχεια καθορίζεται η κίνηση και τέλος η επαλήθευση της ορθότητας του δικτύου ώστε να τρέξει η προσομοίωση, κατά την ολοκλήρωση της οποίας δημιουργούνται αναφορές οι οποίες περιγράφουν την απόδοση του υπό μοντελοποίησης δικτύου. Ο προσομοιωτής τρέχει σε Windows.



Εικόνα 3 : Γραφικό περιβάλλον Comnet III

4.8 INSANE

Ο INSANE είναι ένας αντικειμενοστραφής προσομοιωτής δικτύου, διακριτών γεγονότων που σχεδιάστηκε για τη δοκιμή διαφόρων αλγορίθμων IP-over-ATM (asynchronous transfer mode). Το μεγαλύτερο μέρος του INSANE είναι γραμμένο σε C++, ενώ η παραμετροποίηση και η διαμόρφωση της προσομοίωσης γίνεται με Tcl scripts. Η μονάδα ATM χρησιμοποιεί ουρές όπως η FIFO και η RCSP (Rate-Controlled Static Priority). Τα υποστηριζόμενα πρωτόκολλα περιλαμβάνουν τα IP, TCP, UDP και άλλα. Ειδικότερα η προσομοίωση του TCP εκτελεί σύνδεση, έλεγχο ροής και συμφόρησης, αναμετάδοση και γρήγορη επαναμετάδοση. Όπως πολλοί προσομοιωτές, έτσι και ο INSANE μιμείται την συμπεριφορά εφαρμογών του Διαδικτύου, με σκοπό την παροχή ενός ρεαλιστικού φόρτου εργασίας, συμπεριλαμβανομένων των: telnet, ftp, WWW, real-time ήχου και real-time video. Ο INSANE τρέχει σε περιβάλλον UNIX.

4.9 NIST

Επηρεασμένος από τον NETSIM, ο NIST είναι ένας εξομοιωτής που μελετά την απόδοση των ATM δικτύων. Μέσω της ανάλυσης του ATM πρωτοκόλλου εξάγει στατιστικά όπως το throuput, το bandwidth και το link utilization. Όπως και ο INSANE, ο NIST τρέχει σε UNIX σε περιβάλλον γραμμένο σε C και παρέχει μοντελοποίηση της τοπολογίας του δικτύου και των παραμέτρων που συνδέονται με τα components που εξομοιώνονται.

4.10 TRAFFIC

Ο Traffic είναι ένα προ'ίόν προσομοίωσης το οποίο σχεδιάστηκε για την επίλυση σύνθετων προβλημάτων σε τηλεφωνικά δίκτυα (UMTS,GPRS,GSM,LTE), όπου δεν

μπορεί να βρεθεί λύση με τις παραδοσιακές εξισώσεις Erlang. Παρότι το Erlang λύνει πολλά προβλήματα, υπάρχουν πολλές περιπτώσεις στο σχεδιασμό δικτύων πραγματικού κόσμου όπου δεν μπορεί να βρεθεί λύση μέσω εξισώσεων. Σε αυτές τις περιπτώσεις, η πιο πρακτική λύση είναι να "προσομοιώσουμε" το πρόβλημα. Ο Traffic έχει ένα εύκολο στην χρήση γραφικό περιβάλλον και τρέχει σε Windows.

4.11 GLOMOSIM

Το πλήρες του όνομα είναι Global Mobile Information System Simulator και είναι ένας δικτυακός προσομοιωτής ενσύρματων και ασύρματων δικτύων. Σχεδιάστηκε χρησιμοποιώντας την προσομοίωση παράλληλων διακριτών γεγονότων από την Parsec (γλώσσα παράλληλου προγραμματισμού σε C). Ο προσομοιωτής παρέχει ένα εργαλείο οπτικοποίησης σε μια ανεξάρτητη πλατφόρμα επειδή είναι γραμμένος σε C++. Αυτό το εργαλείο επιτρέπει τον εντοπισμό σφαλμάτων και την επαλήθευση μοντέλων και σεναρίων , εμφανίζει τη διαδρομή των πακέτων, την κινητικότητα σε διάφορα χρώματα ενώ εξάγει και στατιστικά στοιχεία της προσομοίωσης. Η ανάπτυξη του Glosim δεν είναι πλέον ενεργή.

4.12 SHUNRA VIRTUAL ENTERPRISE

Ο Shunra V.E. είναι ένα εργαλείο προσομοιωτής που μπορεί να χρησιμοποιηθεί για το τεστάρισμα του σχεδιασμού και της απόδοσης ενός δικτύου πριν αυτό σχεδιαστεί σε πραγματικό χρόνο. Σύμφωνα με τον ιστιότοπο του, ο Shunra V.E. υποστηρίζει τη χρήση του από πάνω από 2000 πελάτες εντός των χρηματοπιστωτικών υπηρεσιών, των κατασκευών, της τεχνολογίας των τηλεπικοινωνιών, την υγειονομική περίθαλψη καθώς ακόμα και σε κυβερνητικούς οργανισμούς αλλά και στο στρατό. Στον τομέα των δικτύων μπορεί να δώσει λύσεις σε δίκτυα απομακρυσμένων περιοχών και τελικών χρηστών. Μέσω του Shunra VE μπορεί να δοκιμασθεί η λειτουργικότητα , η απόδοση και η επεκτασιμότητα ενός δικτύου υπό της τρέχουσες αλλά και

μελλοντικές συνθήκες του πραγματικού κόσμου.

4.13 CNET

Ένας προσομοιωτής δικτύων ο οποίος τρέχει σε Linux και Mac ενώ επιτρέπει πειράματα σε διάφορα επίπεδα δικτύου (data-link layer, network layer). Αναπτύχθηκε και χρησιμοποιείται ειδικά από φοιτητές σε προπτυχιακά μαθήματα σε όλο τον κόσμο. Παρέχει γραφική αναπαράσταση του υπό προσομοίωση δικτύου και επιτρέπει τις τροποποιήσεις ενώ η προσομοίωση τρέχει. Για την σωστή λειτουργία του CNET απαιτείται τα πρωτόκολλα να είναι γραμμένα σε C ενώ ένας compiler τρέχει την προσομοίωση.



Εικόνα 4 : Ο Cnet σε περιβάλλον Mac.

4.14 QUALNET DEVELOPER

Ισχυρίζεται ότι είναι ο ταχύτερος προσομοιωτής πραγματικού χρόνου, ο οποίος μοντελοποιεί την κυκλοφορία ασύρματων και ενσύρματων δικτύων σε πραγματικό

χρόνο. Μέσω του QualNet Animator είναι δυνατή η γραφική σχεδίαση του δικτύου (χρησιμοποιώντας μια βιβλιοθήκη συστατικών) και η εμφάνιση των αποτελεσμάτων της προσομοίωσης.

Ο QualNet αποτελείται από τα ακόλουθα στοιχεία:

Qualnet Architect – Ένα εργαλείο για την γραφική οπτικοποίηση . Στην σχεδίαση μπορεί να ρυθμιστεί το έδαφος, τις συνδέσεις δικτύου, τα υποδίκτυα και άλλες παραμέτρους λειτουργίας των κόμβων του δικτύου.

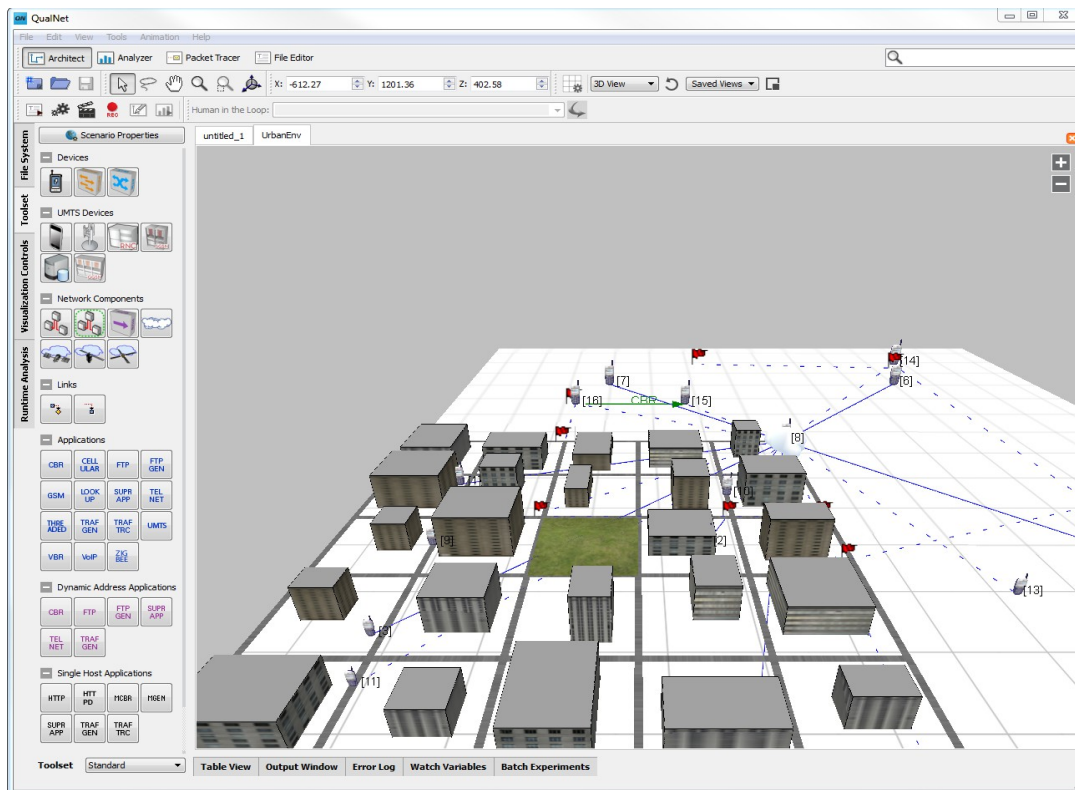
Qualnet Analyzer- Ένα εργαλείο το οποίο εμφανίζει γραφικές παραστάσεις με στατιστικά στοιχεία εκατοντάδων μετρήσεων που συλλέγονται κατά τη διάρκεια ενός σεναρίου.

Qualnet Packet Tracer- Ένα γραφικό εργαλείο το οποίο παρέχει οπτική αναπαράσταση των πακέτων που δημιουργούνται κατά τη διάρκεια της προσομοίωσης ενός δικτύου.

Qualnet File Editor- Ένα εργαλείο επεξεργασίας κειμένου.

Qualnet Command Line- Command Line για πρόσβαση στην προσομοίωση.

Το Qualnet επιτρέπει στον χρήστη να σχεδιάσει νέα πρωτόκολλα, να βελτιστοποιήσει τα ήδη υπάρχοντα , να σχεδιάσει μεγάλα ενσύρματα και ασύρματα δίκτυα καθώς και να ανάλυση την απόδοση δικτύων χρησιμοποιώντας what-if ανάλυση για να τα βελτιστοποιήσει. Για την εφαρμογή νέων πρωτοκόλλων ο προσομοιωτής χρησιμοποιεί C/C++ ακολουθώντας ένα διαδικαστικό πρότυπο.



Εικόνα 5 : Στιγμιότυπο προσομοίωσης στον Qualnet Developer

4.15 GTNETS

Είναι ένα πλήρως εξοπλισμένο περιβάλλον προσομοίωσης δικτύων που επιτρέπει στους χρήστες να μελετήσουν την απόδοση δικτύων μέτριας έως μεγάλης κλίμακας. Η σχεδιαστική του φιλοσοφία είναι να δημιουργήσει ένα περιβάλλον προσομοίωσης, δομημένο σαν ένα πραγματικό δίκτυο. Τα αντικείμενα προς προσομοίωση, οι κόμβοι του δικτύου έχουν μια ή περισσότερες διασυνδέσεις κάθε μια από τις οποίες μπορεί να έχει μια σχετική IP διεύθυνση, όπως ακριβώς γίνεται στο transport layer στις TCP συνδέσεις.

4.16 NCTUNS

Είναι ένας αρκετά πιστός προσομοιωτής ασύρματων και ενσύρματων δικτύων, διαφόρων πρωτοκόλλων. Μπορεί εύκολα να χρησιμοποιηθεί και σαν εξομοιωτής, χαρακτηριστικό πολύ χρήσιμο για την λειτουργία και την απόδοση ενός δικτύου πραγματικού κόσμου. Ένα δίκτυο αποτελούμενο από πολλούς κόμβους και πολλές εφαρμογές να τρέχουν σε αυτούς απαιτεί χρήση αρκετής ενέργειας της CPU και της κύριας μνήμης. Σε μια τέτοια περίπτωση ο NCTUns μπορεί να χωρίσει το δίκτυο σε πολλά μικρότερα τμήματα προς εξομοίωση. Υποστηρίζει πολλαπλά σημαντικά δίκτυα και πρωτόκολλα όπως : wireless, ad hoc, GPRS, WiMax και CSMA/CD, OSPF, UDP, RTP, RTCP, HTTP, FTP, TELNET και πολλά άλλα.

4.17 PARSEC

Ένας προσομοιωτής βασισμένος στην C που αναπτύχθηκε στο U.C.L.A. από το Parallel Computing Laboratory για διαδοχική και παράλληλη εκτέλεση μοντέλων προσομοίωσης διακριτών γεγονότων. Μπορεί να χρησιμοποιηθεί και σαν παράλληλη γλώσσα προγραμματισμού. Είναι διαθέσιμο μόνο σε ακαδημαϊκά ιδρύματα ενώ οι υπόλοιποι χρήστες μπορούν να χρησιμοποιήσουν την εμπορική του εφαρμογή, το Qualnet.

4.18 PERFORMANCE PROPHET

Ένα εργαλείο για την πρόβλεψη της απόδοσης παράλληλων και κατανεμημένων συστημάτων βασισμένο στον CpSim. Με την χρήση προσομοίωσης διακριτών γεγονότων και μαθηματικών συναρτήσεων αποσκοπεί στην μείωση του χρόνου προσομοίωσης ενός σεναρίου. Βασισμένο σε ένα μοντέλο *UML* το εργαλείο μπορεί να προβλέψει την εκτελεστική συμπεριφορά του εκτελέσιμου και των

αρχιτεκτονικών του. Σχεδιάστηκε στην Αυστρία και υποστηρίχθηκε από την Austrian Science Fund.

4.19 REAL

Ο REalistic And Large είναι ένας προσομοιωτής δικτύου που προορίζεται για την μελέτη της δυναμικής συμπεριφοράς της ροής και της συμφόρησης σε δίκτυα μεταγωγής πακέτων. Γραμμένο σε C μπορεί να μιμηθεί τις ενέργειες πολλών γνωστών πρωτοκόλλων ελέγχου ροής ενώ υποστηρίζει αλγόριθμους όπως τους Fair Queuing και Hierarchical Round Robin. Ο πηγαίος κώδικας παρέχεται μέσω 30 στοιχείων γραμμένων σε C και έτσι οι χρήστες μπορούν να τροποποιήσουν τον προσομοιωτή για να ικανοποιήσουν τις ανάγκες τους. Στην τελευταία έκδοσή του συμπεριλαμβάνει ένα GUI γραμμένο σε JAVA όπου η τοπολογία δημιουργείται γραφικά.

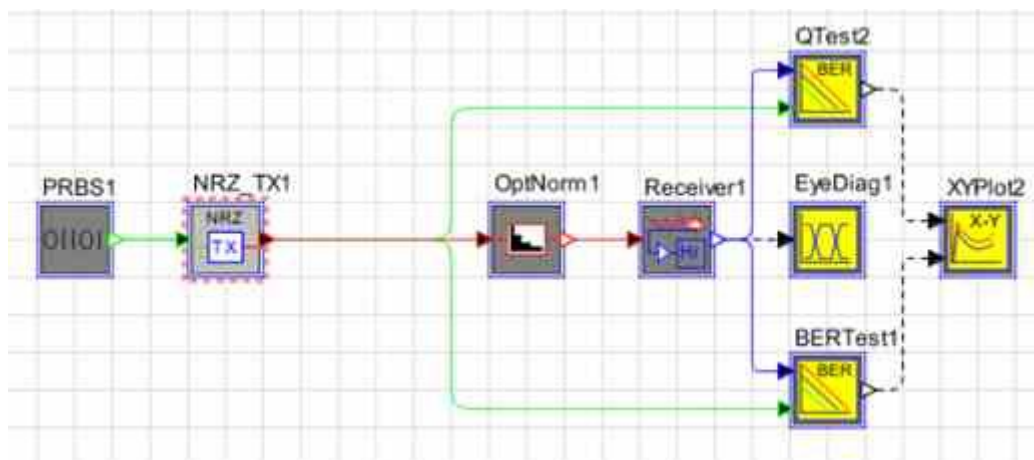
4.20 CPSIM

Είναι ένα εργαλείο παράλληλης και γενικού σκοπού προσομοίωσης αφού χρησιμοποιείται για την προσομοίωση παράλληλων διακριτών γεγονότων, ενώ είναι κατάλληλος για μεγάλες προσομοιώσεις δικτύων υπολογιστών. Στόχος των σχεδιαστών του είναι η βέλτιστη απόδοση. Οι χρήστες έχουν απόλυτη ελευθερία στη δημιουργία μοντέλων ενώ δεν υπάρχει γραφικό περιβάλλον. Ο CpSim έχει σχεδιαστεί κυρίως για ειδικούς στην προσομοίωση χρήστες.

4.21 OPTSIM

Είναι το βραβευμένο λογισμικό της Rsoft για το σχεδιασμό και την προσομοίωση συστημάτων οπτικής επικοινωνίας. Διατίθεται στο εμπόριο από το 1998 και είναι σε

χρήση από τους κορυφαίους μηχανικούς σε ακαδημαϊκούς και βιομηχανικούς οργανισμούς σε όλο τον κόσμο. Εύκολο στη χρήση με γραφικό περιβάλλον και όργανα μέτρησης σαν σε πραγματικό εργαστήριο. Μέσω του Optsim αναλύονται συστήματα επικοινωνιών με σκοπό την αύξηση της παραγωγικότητάς τους αλλά και τη μείωση του χρόνου , βελτιστοποιώντας τον σχεδιασμό για καλύτερη απόδοση και το κόστος . Ένα πολύ σημαντικό προτέρημα του προσομοιωτή είναι οι διασυνδέσεις που παρέχει με εργαλεία τρίτων κατασκευαστών, όπως το Matlab και το Luna Optical Vector Analyzer. Η διεπαφή Matlab καθιστά εύκολη την ανάπτυξη προσαρμοσμένων μοντέλων που χρησιμοποιούν τη γλώσσα m-file ή το περιβάλλον SimulinkModeling environment. Σαν προσομοιωτής οπτικών επικοινωνιών, υποστηρίζει διαμορφώσεις όπως QPSK, BPSK, QAM, OFDM PSK κ.τ.λ.. Το Optsim πέρα από το drag-and-drop περιβάλλον παρέχει και μια διεπαφή προγραμματισμού, για γλώσσες προγραμματισμού, όπως η C/C++ για την ανάπτυξη προσαρμοσμένων μοντέλων του χρήστη.



Εικόνα 6 : Γραφικό περιβάλλον στον OptSim

4.22 PHYSIM

Ένα λογισμικό προσομοίωσης για την κατανόηση του σχεδιασμού και της λειτουργίας των συστημάτων επικοινωνίας. Υλοποιήθηκε από την Tetcos, όπως και ο πιο γνωστός NetSim. Ο PhySim διαθέτει παλμοσκόπιο, αναλυτή φάσματος ενώ επιτρέπει την δημιουργία ραδιοσυχνοτήτων. Καλύπτει τεχνικές μερικές εκ των οποίων είναι : ASK, FSK, PSK, DPSK, QPSK, PAM, PPM, TDM-PAM, PCM , RF.



Εικόνα 7 : Διάγραμμα κεραίας στον PhySim

4.23 OVERSIM

Ένα θεμελιώδες πρόβλημα στη μελέτη των *peer-to-peer* δικτύων είναι η αξιολόγηση των νέων πρωτοκόλλων. Ο OverSim, ένα *open source* και ευέλικτο πρόγραμμα προσομοίωσης, γραμμένο σε C++ και βασισμένο στον OmNet++ , σχεδιάστηκε για να ικανοποιήσει ορισμένες απαιτήσεις που είχαν παραμεληθεί εν μέρει, από υφιστάμενα προγράμματα προσομοίωσης. Ο προσομοιωτής περιλαμβάνει πολλά δομημένα και αδόμητα *peer-to-peer* πρωτόκολλα, όπως το Chord, Kademlia και το Gira, τα οποία μπορούν να χρησιμοποιηθούν για την υλοποίηση μιας προσομοίωσης καθώς και σε δίκτυα του πραγματικού κόσμου. Για να διευκολυνθεί η εφαρμογή πρόσθετων πρωτοκόλλων ο OverSim παρέχει πολλές κοινές λειτουργίες όπως ένα

γενικό μηχανισμό αναζήτησης για τα δομημένα *peer-to-peer* δίκτυα και μια διασύνδεση *RPC (remote procedure call)*. Ο προσομοιωτής συλλέγει διάφορα στατιστικά στοιχεία όπως αποστολές και λήψεις δεδομένων, την κίνηση των κόμβων και τις επιτυχείς και ανεπιτυχείς παραδόσεις πακέτων ενώ μπορεί να υποστηρίξει την προσομοίωση δικτύων αποτελούμενα με έως και 100.000 κόμβους.

5. Δικτυακός προσομοιωτής Network Simulator 2

Ο NS2 είναι ένας προσομοιωτής διακριτών γεγονότων που απευθύνεται στην δικτυακή έρευνα. Ακόμη και σήμερα παραμένει ίσως ο δημοφιλέστερος προσομοιωτής ανοιχτού κώδικα. Υποστηρίζει προσομοίωση του TCP, δρομολόγηση καθώς και multicast πρωτόκολλα για ενσύρματα και ασύρματα (τοπικά και δορυφορικά) δίκτυα. Αναπτύχθηκε από το Information Science Institute (I.S.I) .

Ο δικτυακός τύπος της ομάδας εξέλιξης του NS-2 είναι <http://www.isi.edu/nsnam/ns/> . Η ιστοσελίδα παρέχει πληροφορίες σχετικά με τον NS όπως επίσης και τον πηγαίο κώδικά του.

5.1 Ιστορία

Πριν από τον NS, το 1989 αναπτύχθηκε από τον *Srinivasan Keshav* ο προσομοιωτής REAL, πάνω στον οποίο βασίστηκε ο NS. Έτσι 6 χρόνια αργότερα έχουμε την πρώτη έκδοση του NS, γνωστή σαν NS-1, η οποία αναπτύχθηκε στο Lawrence Berkeley National Laboratory (L.B.N.L.) στο διάστημα 1996-1997 από τους Steve McCanne, Sally Floyd Kevin Fall και άλλους συνεισφέροντες. Αυτή η έκδοση ήταν γνωστή σαν L.B.N.L. simulator και προερχόταν από τον προσομοιωτή REAL του Keshav. Ο πυρήνας του προσομοιωτή γράφτηκε σε C++ με σενάρια βασισμένα σε Tcl (γλώσσα προγραμματισμού σεναρίων) .

5.2 Εγκατάσταση Network Simulator 2

Ο δικτυακός προσομοιωτής NS-2 μπορεί να εγκατασταθεί σε συστήματα UNIX (Linux, FreeBSD, Solaris) αλλά και σε Windows μέσω του Cygwin, το οποίο είναι ένα περιβάλλον Linux, αλλά και μέσω του VMWare, το οποίο εξομοιώνει έναν προσωπικό υπολογιστή μέσα από ένα ήδη υπάρχον λειτουργικό σύστημα (Windows, Linux) .

Σε ένα υπολογιστικό σύστημα με λειτουργικό Ubuntu Linux η εγκατάσταση ξεκινά με τις ακόλουθες εντολές :

```
sudo apt-get install build-essential autoconf automake libxmu-dev
sudo apt-get install ns nam xgraph
```

Η πρώτη εντολή εγκαθιστά βασικές εξαρτήσεις του λογισμικού (κάποιες μπορεί να μην χρησιμοποιηθούν καθόλου αλλά η διαδικασία συνιστάται για μια ασφαλή εγκατάσταση). Η δεύτερη εντολή εγκαθιστά σημαντικά συνοδευτικά πακέτα του NS-2, το *Nam* και το *Xgraph* τα οποία θα αναλυθούν παρακάτω.

Σε περίπτωση που θέλουμε να εκτελέσουμε ένα σενάριο προσομοίωσης, αυτό γίνεται με την εντολή *ns*, όπως φαίνεται παρακάτω :

```
ns script_name
```

5.3 TCL SCRIPT-NAM

Η διάταξη του δικτύου που θέλουμε να προσομοιώσουμε καθώς και τα πάντα γύρω από αυτό πρέπει να οριστούν σε ένα αρχείο κειμένου. Η γλώσσα που

χρησιμοποιείται είναι η "Tcl" . Μπορείτε να ξεκινήσετε τον NS από ένα παράθυρο εντολών (command prompt) και να τρέξετε αρχεία με την εντολή "ns filename_tcl" . Αυτό προϋποθέτει να είσαστε στον κατάλογο (directory), που βρίσκεται το αρχείο όπως επίσης ότι το αρχείο "filename_Tcl" είναι το όνομα του κειμένου σε Tcl γλώσσα που περιγράφει την προσομοίωση.

Τα αποτελέσματα της προσομοίωσης θα είναι αρχεία τα οποία θα προορίζονται για χρήστη στο NAM και το όνομα τους θα είναι της μορφής "filename.nam", είτε αρχεία τα οποία προορίζονται για χρήση στο Xgraph.

Η αρχή δημιουργίας ενός Tcl αρχείου γίνεται γράφοντας ένα βασικό template, το οποίο μπορούμε να χρησιμοποιήσουμε για όλα τα αρχεία Tcl scripts. Με την εντολή :

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

δημιουργούμε ένα αντικείμενο (object) προσομοίωσης. Στη συνέχεια πρέπει να ανοίξουμε ένα αρχείο για να γραφτούν τα δεδομένα τα οποία θα χρησιμοποιηθούν για το NAM, τα λεγόμενα "trace data" :

```
set nf [open out.nam w]
$ns namtrace-all $nf
```

Η πρώτη γραμμή ανοίγει ένα αρχείο με το όνομα "out.nam" για εγγραφή και του προσδίδει το file handle "nf". Στη δεύτερη γραμμη καλούμε το αντικείμενο που δημιουργήσαμε προηγουμένως να γράφει σε αυτό όλα τα αρχεία τα οποία έχουν

σχέση με τον NAM. Το επόμενο βήμα έχει ως στόχο να κλείνει το αρχείο δεδομένων trace με την προσθήκη μιας διαδικασίας trace.

```
proc finish {} {  
    global ns nf  
    $ns flush-trace  
    close $nf  
    exit 0  
}
```

Η επόμενη εντολή λέει στο αντικείμενο της προσομοίωσης να εκτελέσει την διαδικασία "finish" μετά από 10.0 sec προσομοίωσης. Τέλος η τελευταία γραμμή αρχίζει την προσομοίωση.

```
$ns at 10.0 "finish"  
$ns run
```

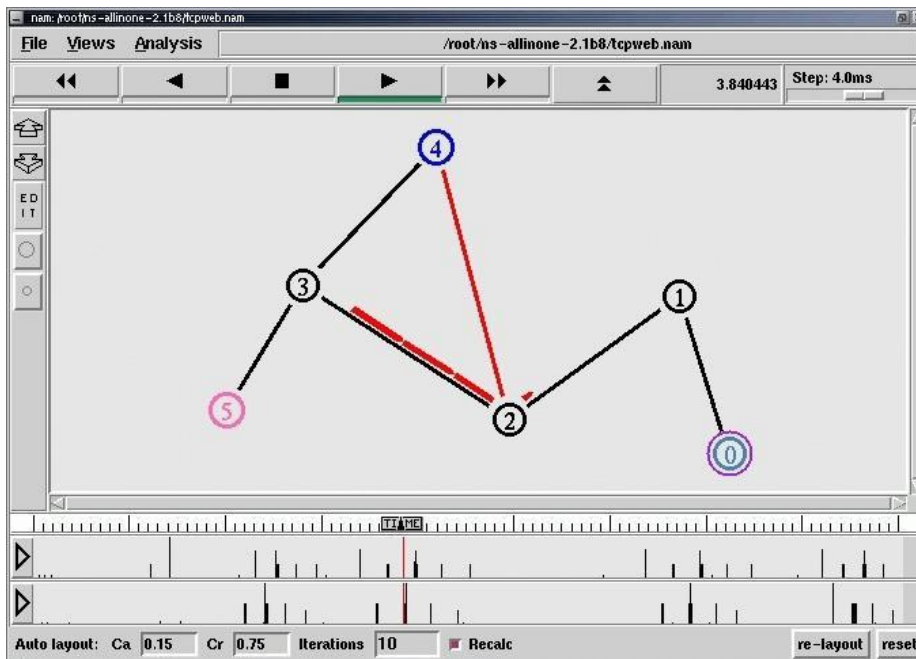
Αφού σώσουμε το αρχείο, το τρέχουμε με την εντολή που αναφέρθηκε και παραπάνω γράφοντας "ns filename.tcl" σε ένα παράθυρο εντολών. Αφού πραγματοποιηθεί η προσομοίωση, θα πρέπει να δημιουργηθεί ένα αρχείο ".nam", στην προκειμένη περίπτωση μας θα έχει το όνομα "out.nam", που θα έχει αποθηκευμένα τα δεδομένα της προσομοίωσης που είναι απαραίτητα για την γραφική απεικόνισή της. Τη γραφική απεικόνιση μπορούμε να την δούμε πληκτρολογώντας την εντολή "nam out.nam" στο παράθυρο εντολών

Ο *Network Animator* είναι ένα πρόγραμμα απεικόνισης βασισμένο στην γλώσσα Tcl που χρησιμοποιείται από τον NS-2 και είναι υπεύθυνο για την παρακολούθηση των

αρχείων της δικτυακής προσομοίωσης. Είναι μια γραφική διεπαφή μέσω της οποίας μπορεί να οπτικοπο'ηθεί η τοπολογία του δικτύου και η απεικόνιση της ροής των πακέτων μεταξύ των κόμβων .Αρχικά δημιουργείται το αρχείο `nam_trace` με την ακόλουθη εντολή :

```
nam nam-file
```

όπου `nam file` είναι το όνομα του αρχείου `nam` το οποίο δημιουργείται μετά από κάθε προσομοίωση του NS, το οποίο διαβάζει ο NAM για να μας εμφανίσει το παρακάτω παράθυρο με την απεικόνιση της τοπολογίας, των πακέτων και πολλά άλλα εργαλεία παρακολούθησης των δεδομένων.



Εικόνα 8 : Τοπολογία σεναρίου στον NAM του NS-2

Εάν λοιπόν κάποιος προτιμά ένα γραφικό περιβάλλον όπου θα μπορεί να υλοποιεί τις

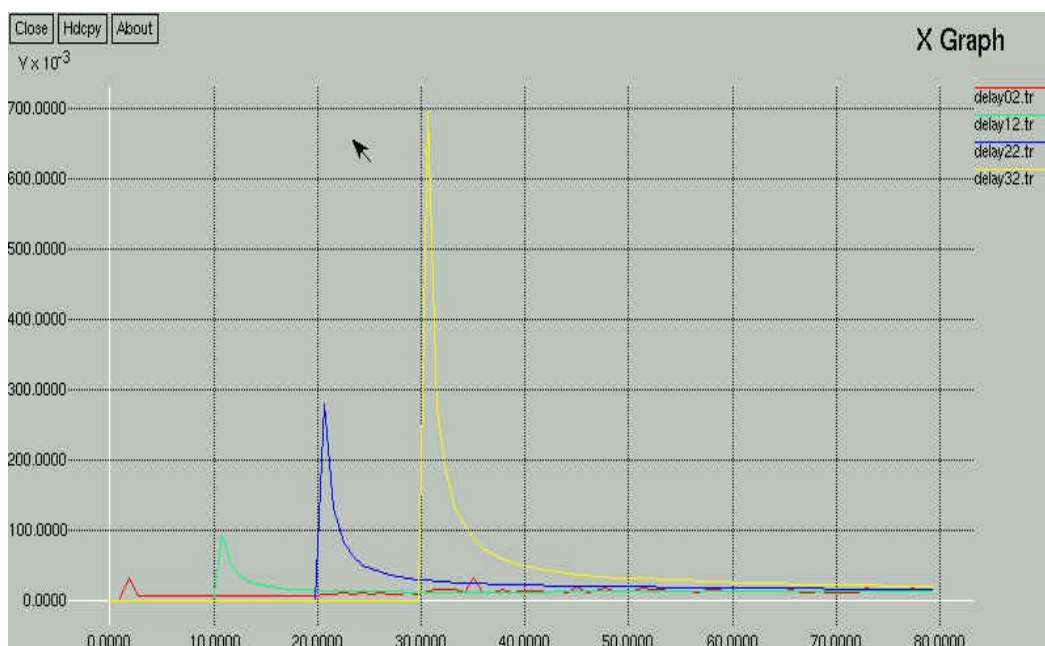
προσομοιώσεις που δημιουργούνται μέσω του NS-2, το NAM υποστηρίζει ένα εύχρηστο περιβάλλον στο χρήστη. Σε αυτό τοποθετούνται οι κόμβοι, ενώνονται μεταξύ τους και στη συνέχεια ορίζονται οι παράγοντες που επιθυμεί ο χρήστης. Στο παραπάνω παράδειγμα φαίνεται μια τοπολογία μεταξύ έξι κόμβων όπως επίσης και η κίνηση μεταξύ αυτών.

Μέσω του Network Animator καθίσταται δυνατή η οπτικοποίηση της τοπολογίας του υπό προσομοίωσης δικτύου, καθώς και η ανταλλαγή και ο αριθμός πακέτων μεταξύ των κόμβων. Ο Nam διαβάζει το αρχείο `nam_trace` και στη συνέχεια δημιουργεί τη δικτυακή τοπολογία εμφανίζοντας το παραπάνω παράθυρο.

5.4 X-GRAPH

Το Xgraph είναι ένα πρόγραμμα, το οποίο μπορεί να χρησιμοποιηθεί για τη δημιουργία γραφικών παραστάσεων των αποτελεσμάτων της προσομοίωσης. Το `xgraph` με τα αποτελέσματά του καλείται με την ακόλουθη εντολή :

```
xgraph <data-file>
```



Εικόνα 8 : Παράδειγμα γραφικής παράστασης, τοπολογίας στον NS-2

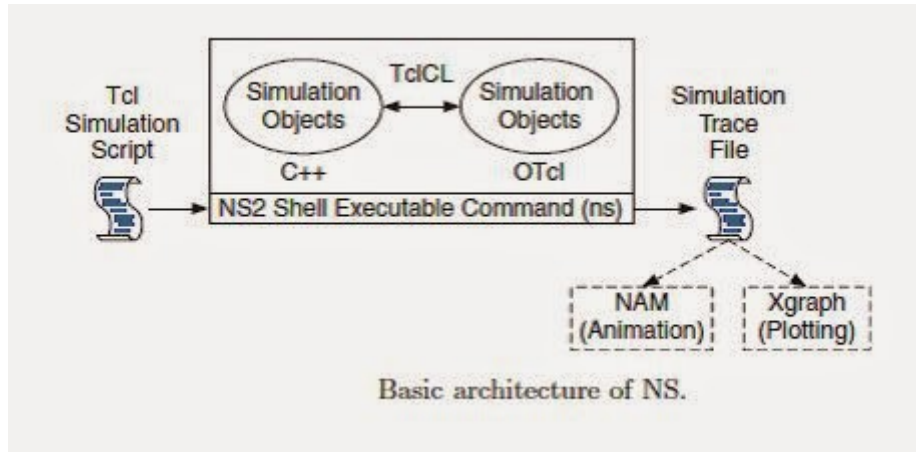
5.5 Χαρακτηριστικά

Ο NS-2 ανήκει στην κατηγορία των προσομοιωτών διακριτών γεγονότων, πράγμα που σημαίνει ότι ο χρόνος προσομοίωσης δεν είναι συνεχής, αλλά αποτελείται από δείγματα στο χρόνο, τα λεγόμενα *events*. Τα γεγονότα δημιουργούν γεγονότα, ενώ η προσομοίωση τερματίζει όταν δεν υπάρχουν άλλα υπό επεξεργασία γεγονότα. Ο NS-2 σχεδιάστηκε με σκοπό να μιμείται την ροή των πακέτων σε ένα δίκτυο. Το υπό προσομοίωση περιβάλλον περιλαμβάνει κόμβους επικοινωνίας ενώ κάθε ένας από αυτούς περιλαμβάνει ένα σύνολο από δικτυακά αντικείμενα, που αναπαριστούν δικτυακά στρώματα ISO/PSO και εφαρμογές. Η αποστολή και η λήψη πακέτων πραγματοποιείται μέσω της αλληλεπίδρασης των αντικειμένων. Τα πακέτα εμπεριέχουν κεφαλίδες (*header*) και ουρές (*trailer*), (όπως συμβαίνει και στα πραγματικά δίκτυα) ανάλογες με τα πρωτόκολλα που χρησιμοποιούνται.

Στον προσομοιωτή χρησιμοποιούνται δύο γλώσσες προγραμματισμού, η C++ και η OTcl. Η χρήση αποκλειστικά αυτών των γλωσσών προγραμματισμού έχει τους λόγους της. Ο μεγαλύτερος είναι στα εσωτερικά χαρακτηριστικά αυτών των δυο. Η C++ είναι αποδοτική στην εφαρμογή σεναρίων αλλά δεν βοηθά στην οπτική και γραφική απεικόνιση της προσομοίωσης. Η OTcl αντιθέτως, συμβαίνει να έχει το χαρακτηριστικό το οποίο στερείται η C++. Έτσι ο συνδιασμός αυτών των δυο γλωσσών αποδεικνύεται να είναι πολύ αποτελεσματικός. Η C++ χρησιμοποιείται για να εκτελέσει το υπό ανάλυση πρωτόκολλο και η OTcl χρησιμοποιείται από τους χρήστες για να ελέγχουν το σενάριο της προσομοίωσης και να προγραμματίζουν τα γεγονότα.

Μια απλοποιημένη οπτική από την μεριά του χρήστη για τον NS2 φαίνεται στο παρακάτω σχήμα. Το σενάριο OTcl χρησιμοποιείται για να ξεκινήσει ο προγραμματιστής γεγονότων, τη σύσταση της δικτυακής τοπολογίας και να σημάνει την αρχή και το τέλος της κίνησης, στέλνοντας τα πακέτα μέσω του προγραμματιστή

των γεγονότων. Αξίζει να σημειωθεί ότι ο προγραμματιστής γεγονότων παρακολουθεί το χρόνο της προσομοίωσης και ελευθερώνει όλα τα γεγονότα σε σειρά. Το σενάριο μπορεί να αλλάξει με μεγάλη ευκολία προγραμματίζοντας το script OTcl .



Εικόνα 9 : Αρχιτεκτονική Ns-2

Το περιβάλλον, το οποίο μπορεί εύκολα να απεικονιστεί μέσω του *NAM*, αποτελείται από κόμβους επικοινωνίας, όπου καθένας από αυτούς περιλαμβάνει δικτυακά αντικείμενα τα οποία αναπαριστούν εφαρμογές. Τα αντικείμενα αυτά αλληλεπιδρούν μεταξύ τους μέσω της αποστολής και της λήψης πακέτων. Όταν τα πακέτα φτάσουν στον προορισμό που έχει οριστεί από τον χρήστη, αποσυναρμολογούνται και αναλύεται η πληροφορία.

Το ένα τμήμα του προσομοιωτή, είναι γραμμένο σε Otcl ώστε να μη χρειάζεται ο χρήστης να γνωρίζει τη δυσκολότερη αλλά πανίσχυρη C++ με σκοπό να τον χρησιμοποιήσει. Ο υπόλοιπος κώδικας του NS-2 είναι γραμμένος σε C++ ώστε τα αποτελέσματα να είναι υψηλού επιπέδου και ο προσομοιωτής αποδοτικός και ταχύς.

Πιο συγκεκριμένα ο χρήστης δημιουργεί το αρχείο το οποίο θέλει να προσομοιώσει,

σε *OTcl*. Η προσομοίωση υλοποιείται τόσο σε C++ όσο και σε *OTcl* (*TclCL*) και το αποτέλεσμα είναι ένα αρχείο που ονομάζεται *trace file*, το οποίο μπορεί να αναλυθεί από τον χρήστη έχοντας και την βοήθεια των *Nam* και *Xgraph*.

6. Δικτυακός προσομοιωτής Network Simulator 3

Η εξέλιξη του NS-2, ο NS-3 στηρίζεται στην C++ για την δημιουργία των μοντέλων προσομοίωσης. Η *OTcl* που χρησιμοποιείται στον NS-2 για τον έλεγχο της προσομοίωσης δεν χρησιμοποιείται από τον NS-3. Η προσομοίωση μπορεί να επιτευχθεί μέσω της C++ και κάποιες φορές μέσω της Python.

6.1 Ιστορία

Το 2006 μια ομάδα ερευνητών ζήτησε και έλαβε χρηματοδότηση από το αμερικανικό Εθνικό Ίδρυμα Επιστημών (N.S.F.) για να οικοδομήσει τον αντικαταστάτη του NS-2. Έτσι τον Ιούνιο του 2008 έχουμε την πρώτη έκδοση του NS-3 ενός προσομοιωτή ανοικτού κώδικα, ο οποίος δεν είναι συμβατός με τον προκάτοχό του NS-2. Ο προσομοιωτής θα δημιουργηθεί από την αρχή, χρησιμοποιώντας την C++. Σήμερα ο NS-3 αριθμεί 21 εκδόσεις (NS-3.21), αναπτύσσονται δηλαδή περίπου τρεις το χρόνο.

Η κατασκευή του NS-3 ενός εξ ολοκλήρου νέου προσομοιωτή, στοχεύει στην ανταπόκρισή του στις σύγχρονες ερευνητικές και μη ανάγκες αναπτυσσόμενος από μια κοινότητα ανοικτού κώδικα. Η συμβατότητα με τον NS-2 ήταν επιθυμητή αλλά τελικά δεν επιτεύχθηκε, γεγονός που απελευθέρωσε τον NS-3 από τυχόν περιορισμούς που μπορεί να έθετε ο προκάτοχός του.

Έτσι ο νέος NS-3 αναπτύσσεται από μια κοινότητα ανοικτού κώδικα με κύριο στόχο τη δημιουργία ενός ολότελα νέου προσομοιωτή, ο οποίος έχει ως απώτερο στόχο να ανταποκρίνεται στις σύγχρονες ερευνητικές ανάγκες.

6.2 Εγκατάσταση

Ο κώδικας του NS-3 διανέμεται βάση της άδειας χρήσης GNU Public Licence. Η εγκατάστασή του απαιτεί την απόκτηση του πηγαίου κώδικα και την μεταγλώττισή του. Παρόμοια με τον NS-2 , ο NS-3 για την μεταγλώττισή του απαιτεί ένα λειτουργικό UNIX/Linux ή Microsoft Windows (με χρήση Cygwin).

Για να αρχίσει η εγκατάσταση, πρέπει να φορτωθούν και να εκτελεστούν στο σύστημα σενάρια χρήσης υλοποιημένα σε Python προκειμένου να μεταφορτωθεί ο πηγαίος κώδικας του προσομοιωτή. Έστω ότι επιθυμούμε να πραγματοποιήσουμε εγκατάσταση του NS-3 σε έναν φάκελο με όνομα *workspace*.

```
$ cd\  
  
$ mkdir workspace  
  
$ cd workspace  
  
$ wget http://www.nsnam.org/release/ns-allinome-3.22.tar.bz2  
  
$ tar xjf ns-allinome-3.22.tar.bz2
```

Η τελευταία εντολή ξεκινά την μεταγλώττιση , διαδικασία η οποία θα διαρκέσει αρκετή ώρα.

Στη συνέχεια θα βρούμε ένα αρχείο με όνομα *build.py* στο φάκελο με τον κώδικα του

NS-3. Για να εγκατασταθεί ο *build* τρέχουμε την ακόλουθη εντολή :

```
./build.py
```

Αν η εγκατάστασή του είναι επιτυχής θα μας εμφανιστεί και σχετικό μήνυμα.

Τέλος τρέχουμε τις ακόλουθες εντολές, για να ρυθμίσουμε το *waf* ένα εργαλείο του *build* :

```
cd ns-<3.22> (ο αριθμός έκδοσης του NS-3 που προτιμήθηκε)
./waf configure
./waf
./test
```

Με την τελευταία εντολή "τεστάρουμε" αν η εγκατάσταση πραγματοποιήθηκε με επιτυχία και κατόπιν το λογισμικό είναι έτοιμο προς χρήση.

6.3 Χρηση

Το σενάριο προσομοίωσης του NS-3 είναι ένα πρόγραμμα γραμμένο σε γλώσσα C++. Η πιο απλή δομή σε μια προσομοίωση του NS-3 είναι η εξής : ορίζουμε έναν φάκελο μέσα στον οποίο δημιουργούμε τρία αρχεία. Ένα από αυτά είναι το αρχείο, το οποίο περιέχει τον κώδικα της προσομοίωσης γραμμένο σε C++ και έχει την κατάληξη cc. Το δεύτερο αρχείο είναι ένα waf το οποίο μέσω μιας εντολής μας δείχνει την θέση του εκτελέσιμου αρχείου του NS-3. Το τρίτο αρχείο είναι ένα python script, το οποίο στην ουσία υποδεικνύει στον NS-3 ότι πρέπει να γίνει μεταγλώττιση και στην συνέχεια εκτέλεση του σεναρίου που δημιουργήσαμε. Αν το σενάριό μας έχει λάθη, είτε συντακτικά είτε λογικά, αυτά θα φανούν στην οθόνη

όπως και γίνεται στα περισσότερα προγραμματιστικά περιβάλλοντα.

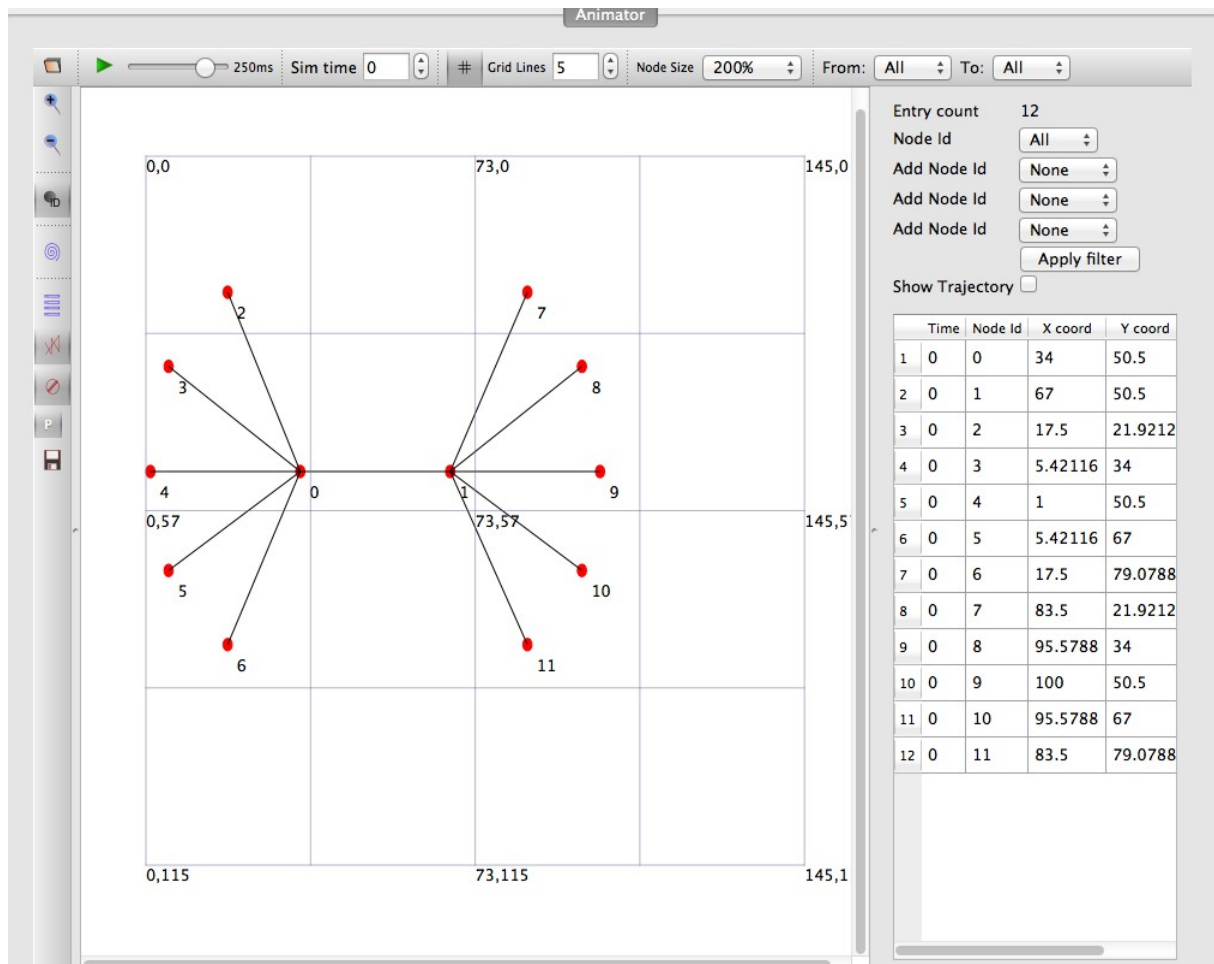
6.4 NETANIM

Όπως ο NS-2, έτσι και ο NS-3 χρησιμοποιεί επιπρόσθετο λογισμικό οπτικοποίησης της προσομοίωσης. Αυτή τη στιγμή το πιο ευρέως διαδεδομένο λογισμικό είναι ο *NetworkAnimator* (*NetAnim*). Το *NetAnim* όπως και το *Xgraph* διατίθεται σε μορφή κώδικα και απαιτεί εγκατάσταση. Για την εγκατάστασή του πάμε στον φάκελο του *NetAnim* ο οποίος μπορεί να βρεθεί στο φάκελο του NS-3.

```
cd NetAnim
qmake-qt4 (GUIToolkits προαπαιτούμενα για την εγκατάσταση του
Animator)
make
```

Για την πραγματοποίηση της οπτικοποίησης το *NetAnim* δέχεται αρχεία της μορφής *.tr*. Δηλαδή πρέπει μετά από κάθε προσομοίωση στον NS-3 να παραχθεί ένα αρχείο τέτοιας μορφής το οποίο θα εκτελεστεί στον *NetAnim*. Η προσομοίωση εκτελείται στον *Animator* με την ακόλουθη εντολή:

```
./NetAnim <file_name>.tr
```



Εικόνα 10 : Τοπολογία δικτύου στον NetAnim του NS-3

Ο NetAnim στην τρέχουσα μορφή του δυστυχώς παρουσιάζει ένα σημαντικό μειονέκτημα για τους χρήστες του καθώς παρουσιάζει σφάλματα στην οπτικοποίηση της κίνησης ενός ασύρματου δικτύου επικοινωνιών. Αυτό έχει σαν αποτέλεσμα την χρήση ενός άλλου λογισμικού για την ανάλυση και επεξεργασία των δεδομένων, το *Wireshark*.

6.5 WIRESHARK

Το *Wireshark* είναι ένα ελεύθερο λογισμικό ανάλυσης πακέτων. Χρησιμοποιείται για την παρακολούθηση και την ανάλυση του δικτύου το εντοπισμό και την

αντιμετώπιση προβλημάτων. Επιτρέπει στον χρήστη να παρακολουθεί όλη την κίνηση που πραγματοποιείται στο δίκτυο δίνοντάς του τη δυνατότητα να φιλτράρει τα αποτελέσματα που επιθυμεί να δει, ως προς το πρωτόκολλα τον χρόνο κ.τ.λ.. Ενδεικτικά στο παράθυρο εργασίας του *Wireshark* μπορούμε να διακρίνουμε πληροφορίες για τα πακέτα, τον χρόνο, τον αποστολέα και τον παραλήπτη, το πρωτόκολλο μεταφοράς, όπως και πληροφορίες σχετικά με κατώτερου επιπέδου πρωτόκολλα (Ip, Ethernet) .

6.6 Χαρακτηριστικά

Ο διακριτών γεγονότων NS-3, είναι ένας πρακτικός προσομοιωτής ανοικτού κώδικα ο οποίος ακόμα αναπτύσσεται σε σύγκριση με τον προκάτοχό του NS-2. Η βασική ιδέα του προσομοιωτή, προέρχεται από πολλούς διαφορετικούς προσομοιωτές πέραν του NS2, συμπεριλαμβανομένων των Yans και του GTNets . Ο NS3 είναι δομημένος σαν μια βιβλιοθήκη η οποία συνδέεται στατικά ή δυναμικά με ένα πρόγραμμα C++ που καθορίζει την τοπολογία προσομοίωσης και ξεκινά τον προσομοιωτή. Για οποιοδήποτε κομμάτι γράφεται σε Python και Tcl πρέπει να κατασκευάζεται μέσω ειδικών γεννητριών και μια διεπαφή σε C++. Συνοπτικά ο NS-3 είναι μια βιβλιοθήκη της C++ η οποία παρέχει ένα σύνολο μοντέλων προσομοίωσης δικτύου που υλοποιούνται σε C++ "τυλιγμένα" με Python. Εν συγκρίσει με τον NS-2, ο NS-3 κάνει εκτεταμένη χρήση της C++ . Η χρήστες συνήθως αλληλεπιδρούν με αυτήν τη βιβλιοθήκη με την εγγραφή μιας εφαρμογής σε C++ ή Python η οποία ενσαρκώνει μια σειρά από μοντέλα τα οποία στήνουν το σενάριο προσομοίωσης. Ο προσομοιωτής συνδέει την C++ με την Python χάρη στη βιβλιοθήκη pybindgen η οποία δημιουργεί αυτόματα τα αρχεία C++ μέσω των οποίων πραγματοποιείται η αλληλεπίδραση από τον χρήστη.

Η κατανάλωση της μνήμης, η διαχείριση πακέτων και οι αλγόριθμοι προσομοίωσης λειτουργούν όλα σε πολύ ικανοποιητικό επίπεδο εύκολα αντιληπτό από τους χρήστες και των δυο προσομοιωτών.

Η ενσωμάτωση των μοντέλων στον προσομοιωτή πραγματοποιείται με τις ακόλουθες τεχνικές. Η πιο δύσκολη είναι η μετατροπή μοντέλων άλλων προσομοιωτών με την προσαρμογή τους σε κώδικα αποδεκτό από τον NS-3 , τεχνική η οποία συνήθως έρχεται εις πέρας από ειδικούς. Μια δεύτερη προσέγγιση μπορεί να θεωρηθεί η ενσωμάτωση ήδη υπαρχόντων βιβλιοθηκών υλοποίησης δικτυακών πρωτοκόλλων στον προσομοιωτή . Τέλος η ενσωμάτωση κώδικα ενός αληθινού λειτουργικού συστήματος από τον NS-3.

Ο προσομοιωτής παρέχει ενημερωμένα μοντέλα , όσον αφορά τα πρωτόκολλα και τις λειτουργίες τους , ενώ η λειτουργία του είναι συμβατή με αυτή των πραγματικών συστημάτων.

Η χρήση του NS-3 είναι εκτεταμένη στον εκπαιδευτικό προπτυχιακό τομέα μέσω *animation* και εκπαιδευτικού υλικού σειρές, τομέας στον οποίο ο NS-2 δε γνώρισε μεγάλη επιτυχία.

Ωστόσο για τον λόγο ότι ο NS-3 βρίσκεται ακόμη υπό ανάπτυξη, παραμένουν κάποιες σημαντικές προκλήσεις υπό επεξεργασία . Το σημαντικότερο, είναι να βελτιωθεί όσο το δυνατόν περισσότερο η αξιοπιστία του προσομοιωτή . Είναι γνωστό ότι ένας από τους μεγαλύτερους περιορισμούς των προσομοιώσεων παραμένει η έλλειψη αξιοπιστίας .

7. Σύγκριση NS-2-NS-3

Ο NS-2 όπως και ο διάδοχός του NS-3 είναι και οι δυο προσομοιωτές διακριτών γεγονότων. Ο NS-2 εξακολουθεί να χρησιμοποιείται από περισσότερους χρήστες ενώ ο NS-3 είναι σε κατάσταση ανάπτυξης, δημοσιοποιώντας κάθε χρόνο νέες εκδόσεις διορθώνοντας πιθανόν λάθη δικά του, αλλά και του NS-2. Στις υποενότητες του κεφαλαίου, θα πραγματοποιηθεί μια συγκριτική μελέτη των δυο προσομοιωτών που αναλύθηκαν στα προηγούμενα κεφάλαια.

7.1 Υπολογιστικά περιβάλλοντα

Η οικογένεια προσομοιωτών NS είναι στενά συνδεδεμένη με τα υπολογιστικά περιβάλλοντα Unix, καθώς η ανάπτυξή τους γινόταν από την αρχή σε αυτά, πράγμα που συνεχίζεται έως και σήμερα. Το λειτουργικό σύστημα GNU/Linux είναι η βασική πλατφόρμα ανάπτυξης των δυο NS. Αυτό δε σημαίνει ότι δεν μπορεί να γίνει η ανάπτυξή τους σε άλλα λειτουργικά, καθότι έχουν κυκλοφορήσει μετατροπές, όπως για Windows (με το Cygwin) αλλά και σε Mac. Παρόλα αυτά είναι αποδεδειγμένο ότι η εγκατάσταση, η συντήρηση και η χρήση των εν λόγω προσομοιωτών είναι προτιμότερη σε Linux.

Και οι δυο προσομοιωτές διανέμονται σε μορφή πηγαίου κώδικα από τους κατασκευαστές τους. Έτσι οι χρήστες έχουν πρόσβαση στον κώδικα με αποτέλεσμα να έχουν την δυνατότητα να διορθώσουν τυχόν σφάλματα αλλά και να προσθέσουν χρήσιμα προς αυτούς χαρακτηριστικά με σκοπό την βέλτιστη απόδοση του προσομοιωτή.

7.2 Πακέτα

Στον NS-2 το κάθε πακέτο αποτελείται από δυο διαφορετικές περιοχές : μια για τα

Headings και το δεύτερο τμήμα το οποίο περιέχει στοιχεία του ωφέλιμου φορτίου. Η μνήμη δεν ελευθερώνεται μέχρι η προσομοίωση να ολοκληρωθεί. Αυτό που γίνεται είναι η επαναχρησιμοποίηση των κατανεμημένων πακέτων επανειλημμένα με αποτέλεσμα η περιοχή των *Headings* του κάθε πακέτου, να περιλαμβάνει όλες τις κεφαλίδες που ορίστηκαν σε περίπτωση επαναχρησιμοποίησής του. Στον NS-3 ένα πακέτο αποτελείται από έναν *buffer*, ο οποίος ανταποκρίνεται επακριβώς στη ροή των *bits* που θα μπορούσαν να αποστέλλονται σε ένα πραγματικό δίκτυο. Η πληροφορία προστίθεται στο πακέτο χρησιμοποιώντας τις υποκλάσεις – το *Heading*, το οποίο προσθέτει πληροφορία στην αρχή του *buffer* και το *Trailer* που προσθέτει στο τέλος – .

7.3 Εξαρτήσεις

Ο NS-2, προκειμένου να εγκατασταθεί και να εκτελεσθεί σε ένα σύστημα, εξαρτάται από άλλα λογισμικά. Αρκετές από αυτές τις εξαρτήσεις, δίνονται από τα αποθετήρια λογισμικού των πλέον σύγχρονων διανομών των Linux, ενώ άλλες οφείλουν να εγκατασταθούν χειροκίνητα από τους χρήστες.

Η εγκατάσταση του NS-2 μπορεί να χαρακτηριστεί ως προβληματική διαδικασία, σε περίπτωση που οι εξαρτήσεις είναι πολύ συγκεκριμένες. Σχετικά πρόσφατα, το παραπάνω πρόβλημα ξεπεράστηκε με την λύση της πακετοποίησης και δημοσίευσης του λογισμικού σαν ενιαίο σύνολο στα Linux.

Αντιθέτως, στον NS-3 υπήρξε από την αρχή η πακετοποίηση του κώδικα καθώς και αυτόματη ρύθμιση των παραμέτρων κατά την εγκατάσταση. Μπορεί κατά την χρήση του να απαιτούνται εξωτερικές βιβλιοθήκες, έχει όμως καταβληθεί προσπάθεια έτσι ώστε να είναι εύκολο να βρεθούν και να εγκατασταθούν. Οι βιβλιοθήκες ανάπτυξης όπως η *python* και η *libxml*, είναι υποχρεωτικές ενώ όλες οι υπόλοιπες εγκαθίστανται προαιρετικά από τον χρήστη προκειμένου να "ξεκλειδωθούν" συγκεκριμένα

χαρακτηριστικά του προσομοιωτή.

Εν κατακλείδι, ο προσομοιωτής NS-3 ανταποκρίνεται θετικά με τα λειτουργικά συστήματα και τις σημαντικές διανομές GNU/Linux. Η εξάρτησή του με αυτά τα λογισμικά, σε σχέση με αυτή του NS-2, είναι λιγότερο στενή και έτσι η διαδικασία εγκατάστασης, ρύθμισης και αντιμετώπισης των προβλημάτων είναι μια διαδικασία φιλική προς τον χρήστη. Αυτό αποτελεί ένα πολύ σημαντικό πλεονέκτημα για ένα οποιοδήποτε λογισμικό, το οποίο βρίσκεται ακόμα σε φάση ανάπτυξης και έχει μεγάλες φιλοδοξίες για το μέλλον. Αντίθετα ο NS-2 έχει σταματήσει την ανάπτυξή του δείχνοντας τα όριά του, μη ικανός πλέον να σπαταλήσει πόρους στην ανανέωση των εξαρτήσεων του.

7.4 Απόδοση

Μια προσομοίωση οφείλει να εξάγει αποτελέσματα και την παρατήρηση αυτών καθώς μεταβάλλονται οι παράμετροι της λειτουργίας της. Ο απαιτούμενος χρόνος μιας προσομοίωσης πρέπει να είναι μικρός και να μην αυξάνεται με την αύξηση του μεγέθους και της πολυπλοκότητας του υπό μελέτη συστήματος.

Στο τομέα αυτό, η μελέτη και η εκτέλεση σεναρίων και στους δυο προσομοιωτές έδειξε πως ο συνολικός χρόνος που απαιτείται για την ολοκλήρωση σεναρίων στον NS-3 είναι γρηγορότερος από αυτόν του NS-2.

Θεωρείται απαραίτητη η δημιουργία ίδιων συνθηκών εκτέλεσης όπως επίσης και των παραμέτρων της προσομοίωσης και στους δυο προσομοιωτές.

Ο NS-2 υστερεί σε αυτό τον τομέα λόγω της διασύνδεσης που πρέπει να πραγματοποιηθεί μεταξύ Tcl και C++. ενώ τα σενάρια στον NS-3 είναι προγράμματα σε γλώσσα C++ που μεταγλωττίζονται και εκτελούνται. Μπορεί σε όποια αλλαγή στον κώδικα του NS-3 να χρειάζεται εκ νέου μεταγλώττιση, ο χρόνος όμως αυτός

θεωρείται αμελητέος. Ο NS-3 επίσης, διαχειρίζεται καλύτερα και την μνήμη του καθώς αποτρέπει την συγκέντρωση αχρείαστων παραμέτρων, με αποτέλεσμα τα πακέτα να μην περιέχουν αχρησιμοποίητη πληροφορία.

7.5 Αρχιτεκτονική

Στον NS-2 η γλώσσα Tcl επιλέχθηκε για τον εξής λόγο : η C++ όπως και οι υπόλοιπες μεταγλωτισμένες γλώσσες ήταν κοστοβόρες ως προς τον χρόνο μεταγλώττισης και εκτέλεσης. Γινόταν χρήση της C++ σε βασικά συστατικά, τα οποία όφειλαν να είναι σταθερά και αξιόπιστα. Έτσι επέλεξαν μια scripting γλώσσα όπως η TCL και οι επεκτάσεις της, η Otel, για την συγγραφή των προσομοιώσεων, με τους χρήστες να είναι σε θέση να γράφουν και να τεστάρουν τα σενάρια τους χωρίς καθυστέρηση σε χρονοβόρες μεταγλωττίσεις.

Παρ' όλα αυτά το παραπάνω χαρακτηριστικό του NS-2 τον καθιστά πολύπλοκο για νέους προγραμματιστές καθώς είναι δύσκολο να διακρίνουν ποια σημεία ενός νέου μοντέλου πρέπει να υλοποιηθούν σε C++ και ποια σε Otel. Πλέον, σήμερα τα υπολογιστικά συστήματα είναι σε θέση να διακρίνουν και να διαχειριστούν τον φόρτο μιας C++ μεταγλώττισης και να την ολοκληρώνουν σε πολυ σύντομο χρόνο. Αυτό έχει ως αποτέλεσμα τα πλεονεκτήματα της scripting γλώσσας να αχρηστεύονται.

Αντιθέτως ο NS-3 επιλέγοντας C++ απλοποίησε τον σχεδιασμό του. Οι προσομοιώσεις μπορούν να υλοποιηθούν σε C++ ενώ οι χρόνοι μεταγλώττισης δεν αποτελούν κανένα πρόβλημα, με τους χρήστες να μπορούν να επιλέξουν την Python σαν εναλλακτική γλώσσα προσομοίωσης.

7.6 Κύκλος ζωής

Ο πιο σύγχρονος NS-3 έχει όπως είναι λογικό μικρότερο ιστορικό ανάπτυξης από τον NS-2. Πλεονεκτεί όμως στο γεγονός ότι δεν έχουμε να κάνουμε με ένα λογισμικό το οποίο σχεδιάστηκε από την αρχή, αντιθέτως οι ιδέες και η εμπειρία που αποκτήθηκε από τον NS-2 χρησιμοποιήθηκε στον NS-3.

Από την άλλη ο NS-2 έχοντας περάσει από πολλά στάδια μπορεί να θεωρηθεί ότι πλέον ο κύκλος ζωής του έχει φτάσει στο τέλος του. Όλα αυτά τα χρόνια διάφορες εκδόσεις των δυο προσομοιωτών δοκιμάστηκαν με σκοπό την αξιολόγηση των εξελίξεων και της δυναμικής του κάθε προσομοιωτή. Και στις δυο περιπτώσεις χρησιμοποιήθηκε ο κώδικας από τα αντίστοιχα αποθετήρια.

Όπως αναφέρθηκε στα παραπάνω κεφάλαια ο NS-2 ξεκίνησε σαν μια απλή παραλλαγή του REAL το 1989. Ο NS-3 ξεκίνησε το 2006 και υπήρξε η εκτίμηση ότι η ανάπτυξή του θα διαρκούσε τέσσερα χρόνια . Παρ'όλα αυτά είναι σε ενεργή ανάπτυξη ακόμα και σήμερα με την ομάδα ανάπτυξής του να διαθέτει αρκετούς προγραμματιστές.

8. Σενάρια προσομοίωσης

Στις παρακάτω ενότητες του κεφαλαίου θα πραγματοποιηθούν τέσσερις δικτυακές προσομοιώσεις στον κάθε προσομοιωτή, οι οποίες αντλήθηκαν από τους φακέλους εγκατάστασης του κάθε προσομοιωτή.

8.1 Σενάρια προσομοίωσης στον NS-2

Όπως αναφέρθηκε παραπάνω τα σενάρια προσομοίωσης στον NS-2 αποτελούνται από αρχεία με κώδικα γραμμένο σε οTcl . Στη γραμμή εντολών ο χρήστης μπορεί να δει την έξοδο της προσομοίωσης, ενώ κατά την εκτέλεσή της ενημερώνεται για τα γεγονότα με διάφορα μηνύματα και παράλληλα παράγονται και τα trace files.

Τα trace files αποτελούνται από τα αρχεία .nam, τα οποία μας εξυπηρετούν στην οπτικοποίηση της τοπολογίας στο NAM που αναλύθηκε στην προηγούμενη ενότητα, όπως επίσης και από τα αρχεία .tr μέσω των οποίων περιγράφεται κωδικοποιημένα η κίνηση των πακέτων καθώς και η θέση του κάθε κόμβου.

Σενάριο Routing

Σε αυτό το σενάριο θα μελετηθεί η δρομολόγηση των πακέτων μεταξύ των κόμβων σε μια απλή συνδεσμολογία όπου θα οριστούν ένας κόμβος αποστολέας και ένας κόμβος παραλήπτης . Βασική προϋπόθεση ώστε να ξεκινήσει το σενάριο να τρέχει, είναι να ορίσουμε την τοπολογία καθώς και την θέση των κόμβων.

Με τις παρακάτω γραμμές κώδικα ορίζουμε το κανάλι επικοινωνίας και τους συνδέσμους μεταξύ των κόμβων.

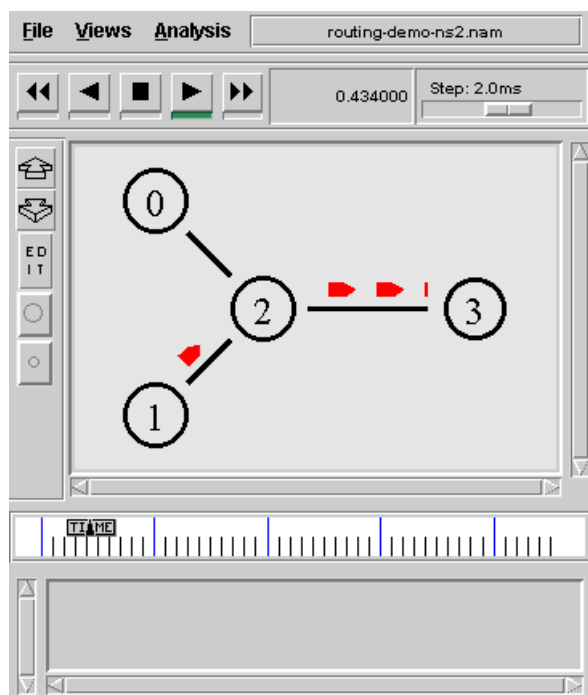
```
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]

$ns duplex-link $n0 $n2 2Mb 10ms DropTail
$ns duplex-link $n1 $n2 2Mb 10ms DropTail
$ns duplex-link $n2 $n3 1.7Mb 20ms DropTail
$ns queue-limit $n2 $n3 10
$ns duplex-link-op $n0 $n2 orient right-down
$ns duplex-link-op $n1 $n2 orient right-up
$ns duplex-link-op $n2 $n3 orient right
$ns duplex-link-op $n2 $n3 queuePos 0.5
```

Τέλος, ορίζεται η χρονική σειρά των γεγονότων που θα πραγματοποιηθούν και ξεκινά η διαδικασία της προσομοίωσης.

```
$ns at 0.1 "$cbr start"
$ns at 1.0 "ftp start"
$ns at 4.0 "ftp stop"
$ns at 4.5 "cbr stop"
$ns at 4.5 "$ns detach-agent $n0 $tcp ; $ns detach-agent $n3 $sink"
$ns at 8.0 "finish"
$ns run
```

Στο παρακάτω στιγμιότυπο μπορούμε να δούμε την οπτικοποίηση της παραπάνω προσομοίωσης μέσω του NAM.



Σενάριο TCP

Στο σενάριο αυτό θα μελετηθεί μια δικτυακή τοπολογία ενσύρματων συνδεδεμένων κόμβων οι οποίοι ανταλλάσσουν πακέτα TCP.

Με έναν βρόγχο "for" ο οποίος διατρέχει όλους τους κόμβους ορίζεται η συνδεσιμότητα μεταξύ τους όπως φαίνεται παρακάτω , ενώ η προσομοίωση διέπεται από δυο διαδικασίες, την main και την finish.


```

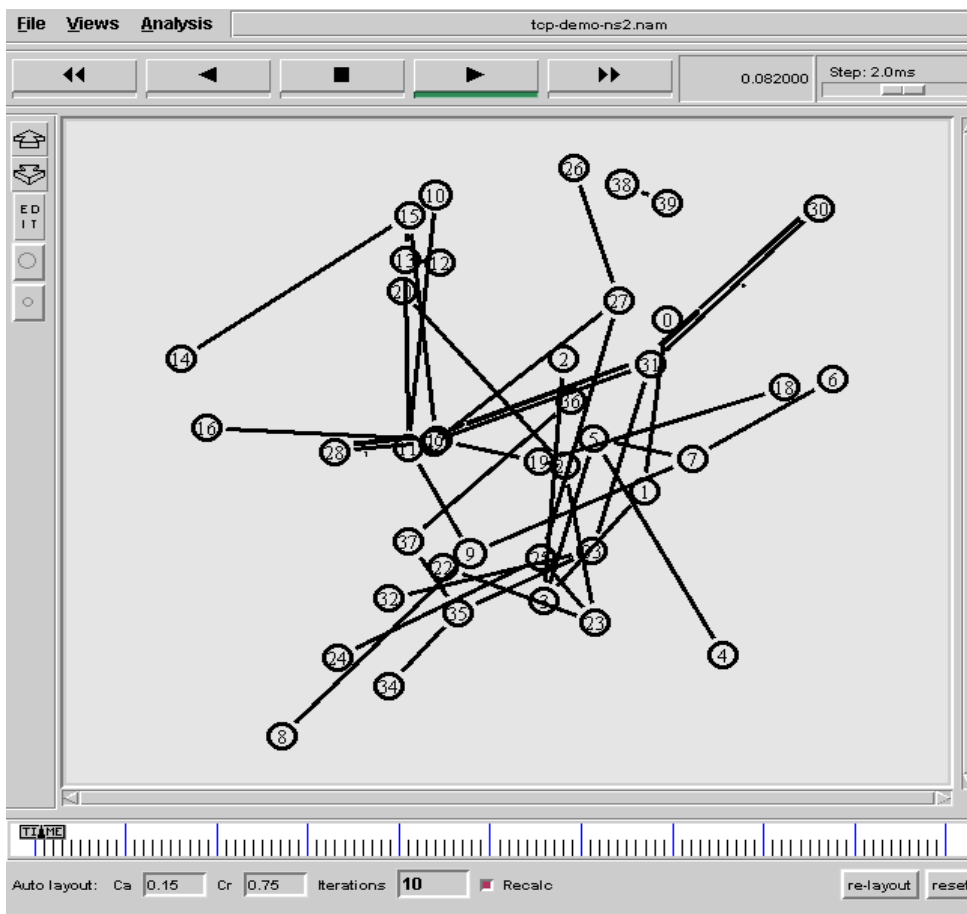
#Create bottleneck and dest nodes
set n2 [$ns node]
set n3 [$ns node]

#Create links between these nodes
$ns duplex-link $n2 $n3 0.7Mb 20ms DropTail

set NumbSrc 5
set Duration 10

#Source nodes
for {set j 1} {$j<=$NumbSrc} { incr j } {
set S($j) [$ns node]
}

```



Στην παραπάνω εικόνα φαίνεται ο ορισμός και η συνδεσιμότητα των κόμβων, η οποία πραγματοποιείται με μια for ενώ στη συνέχεια φαίνεται ένα στιγμιότυπο της οπτικοποίησης της προσομοίωσης μέσω του Nam.

Σενάριο UDP

Στο UDP σενάριο θα οριστεί μια τοπολογία, στην οποία οι κόμβοι θα ανταλλάσσουν UDP πακέτα. Η δομή του σεναρίου είναι παρόμοια με τα παραπάνω σενάρια με τη μόνη διαφορά ότι οι agents που θα ορίσουμε τώρα θα είναι τύπου UDP, όπως ακριβώς φαίνεται παρακάτω.

```
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
$ns duplex-link $n0 $n2 2Mb 5ms DropTail
$ns duplex-link $n2 $n1 1.5Mb 10ms DropTail
$ns duplex-link-op $n0 $n2 orient right
$ns duplex-link-op $n2 $n1 orient right
set udp0 [new Agent/UDP]
$ns attach-agent $n0 $udp0
set udp1 [new Agent/UDP]
$ns attach-agent $n1 $udp1
$ns connect $udp0 $udp1
```

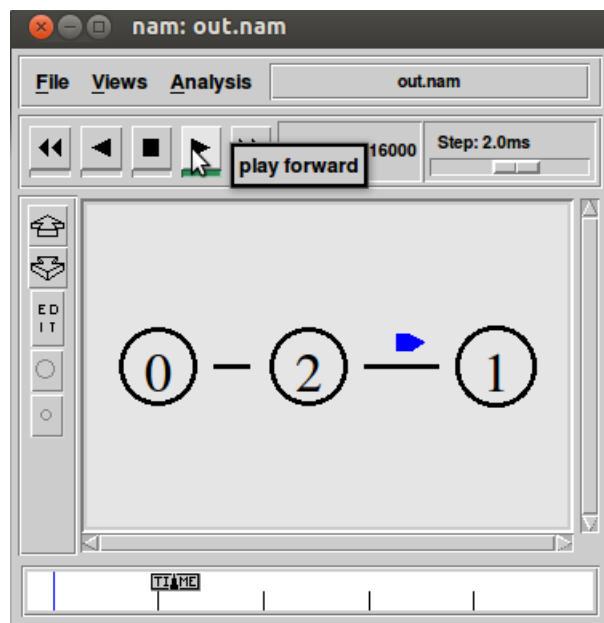
Όπως και στο σενάριο Tcp, έτσι και δω καλούνται δυο διαδικασίες. Η διαδικασία finish η οποία καλείται σε όλα τα σενάρια του NS-2 για τον τερματισμό της προσομοίωσης, καθώς και η διαδικασία process_data η οποία καλείται από κάθε κόμβο όταν αυτός λαμβάνει από κάποιον agent άγνωστο πακέτο, στην παρακάτω περίπτωση, πακέτο τύπου ping.

```

Agent/UDP instproc process_data {size data} {
  global ns
  $self instvar node_
  $ns trace-annotate "[${node_} node-addr] received {$data}"
  if {[regexp {ping *\(([0-9]+\)} $data entirematch number]} {
    $self send 100 "pong($number)"
  } elseif {[regexp {countdown *\(([0-9]+\)} $data entirematch number]
  && $number > 0 } {
    incr number -1
    $self send 100 "countdown($number)"
  }
}

```

Στην παρακάτω εικόνα φαίνεται ένα στιγμιότυπο του UDP σεναρίου μέσω του NAM.



Σενάριο Wireless

Στο ασύρματο σενάριο προσομοίωσης ορίζεται μια τοπολογία όπου οι κόμβοι διέπονται από το πρωτόκολλο 802.11, ενώ μεταξύ τους στέλνουν πακέτα των 1000 bytes . Η λήψη των πακέτων ρυθμίζεται σταθερή, ανεξάρτητα από την απόσταση των

κόμβων και την ισχύ μετάδοσης. Όπως και στα παραπάνω σενάρια έτσι και σε αυτό υπάρχει η διαδικασία finish η οποία καλείται για τον τερματισμό της προσομοίωσης, η οποία ξεκινάει αφού οριστούν οι παράμετροι , για τον τύπο της κεραίας, του καναλιού τις διευθύνσεις και άλλα όπως φαίνεται παρακάτω.

```
set val(chan) Channel/WirelessChannel ;# channel type
set val(prop) Propagation/TwoRayGround ;# radio-propagation model
set val(ant) Antenna/OmniAntenna ;# Antenna type
set val(ll) LL ;# Link layer type
set val(ifq) Queue/DropTail/PriQueue ;# Interface queue type
set val(ifqlen) 50 ;# max packet in ifq
set val(netif) Phy/WirelessPhy ;# network interface type
set val(mac) Mac/802_11 ;# MAC type
set val(rp) DSDV ;# ad-hoc routing protocol
set val(nn) 5 ;# number of mobilenodes
```

Παρακάτω ορίζεται η μορφή την οποία θα έχουν στην έξοδο τα αποτελέσματα της προσομοίωσης.

```
set tracefd [open simple.tr w]
$ns_ trace-all $tracefd
create-god $val(nn)
set topo [new Topography]
$topo load_flatgrid 1000 1000
$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) \
    -channel [new $val(chan)] \
    -topoInstance $topo \
    -agentTrace ON \
    -routerTrace ON \
    -macTrace OFF \
    -movementTrace OFF
```

Στη συνέχεια όπως και στα προηγούμενα σενάρια, ορίζουμε την κίνηση των κόμβων, την αρχική τους θέση και την κίνησή τους κατά τη διάρκεια της προσομοίωσης.

```
for {set i 0} {$i < $val(nn) } {incr i} {  
    set node_($i) [$ns_ node]  
    $node_($i) random-motion 0  
    $node_($i) set Y_ 0.0  
    $node_($i) set Z_ 0.0  
}  
  
$node_(0) set X_ 0.0  
$node_(1) set X_ 200.0  
$node_(2) set X_ 400.0  
$node_(3) set X_ 600.0  
$node_(4) set X_ 800.0
```

8.2 Σενάρια προσομοίωσης στον NS-3

Στην ενότητα που ακολουθεί θα παρουσιαστούν τα σενάρια προσομοίωσης που υλοποιήθηκαν στο δικτυακό προσομοιωτή NS-3.

Σενάριο Routing

Στο σενάριο Routing του NS-3 μελετάται η δρομολόγηση των πακέτων σε ένα ενσύρματο δίκτυο. Οι συνδέσεις μεταξύ των τεσσάρων κόμβων που έχουν οριστεί από την ομάδα ανάπτυξης του προσομοιωτή, είναι point-to-point, ενώ χρησιμοποιούνται FTP/TCP και CBR/UDP πακέτα σαν δικτυακός φόρτος. Ο πηγαίος

κώδικας όλων των σεναρίων προσομοίωσης, αποτελείται από ένα αρχείο με κατάληξη .cc, το οποίο είναι το σενάριο μας γραμμένο σε C++.

Το αρχείο routing.cc ξεκινάει με γραμμές σχολίων που παρέχουν πληροφορίες για το περιεχόμενο του αρχείου.

Όπως σε κάθε C++ αρχείο, στην αρχή του κώδικα βρίσκονται οι εντολές που καθορίζουν τις βιβλιοθήκες. Στην συγκεκριμένη περίπτωση εκτός από κάποιες συνηθισμένες βιβλιοθήκες της C++, βλέπουμε και νέες οι οποίες παρέχονται από τον προσομοιωτή.

Στο σενάριο routing, η λειτουργικότητα της προσομοίωσης υλοποιείται εντός της main() καθώς δεν υπάρχουν εξωτερικές συναρτήσεις, όπως μπορεί να συμβεί με άλλα σενάρια χρήσης.

Παρακάτω, φαίνεται η διαδικασία σύμφωνα με την οποία ορίζουμε κόμβους στον NS-3. Στην προκειμένη περίπτωση, ορίζονται τέσσερις κόμβοι.

```
NS_LOG_INFO ("Create nodes.");  
NodeContainer c;  
c.Create (4);  
NodeContainer n0n2 = NodeContainer (c.Get (0), c.Get (2));  
NodeContainer n1n2 = NodeContainer (c.Get (1), c.Get (2));  
NodeContainer n3n2 = NodeContainer (c.Get (3), c.Get (2));
```

Σενάριο TCP

Στο σενάριο TCP του NS-3, ορίζεται μια ενσύρματη δικτυακή τοπολογία μεταξύ δυο κόμβων. Όπως και στο παραπάνω σενάριο, ο πηγαίος κώδικας της προσομοίωσης

που όρισε η ομάδα ανάπτυξης, αποτελείται από ένα πρόγραμμα με κατάληξη .cc, γραμμένο σε C++.

Το αρχείο ξεκινάει με γραμμές σχολίων οι οποίες περιλαμβάνουν πληροφορίες για το περιεχόμενο του αρχείου. Στη συνέχεια μπορούμε να δούμε και ένα διάγραμμα της τοπολογίας.

Εν συνεχεία βρίσκουμε τις εντολές συμπερίληψης των βιβλιοθηκών όπως συμβαίνει σε κάθε υποδειγματικό αρχείο C++. Οι εντολές περιλαμβάνουν συνηθισμένες βιβλιοθήκες της C++, αλλά και άλλες που παρέχονται από τον προσομοιωτή.

Η λειτουργικότητα της προσομοίωσης υλοποιείται εντός της main() αλλά και σε εξωτερικές συναρτήσεις που ορίζονται έξω από αυτήν και καλούνται στην συνέχεια. Σε αυτό το σενάριο, δεν υπάρχουν εξωτερικές συναρτήσεις, έτσι όλη η λειτουργικότητα υλοποιείται εντός της main().

Παρακάτω φαίνεται η δημιουργία των δυο κόμβων στο TCP σενάριο.

```
NS_LOG_INFO ("Create nodes.");  
NodeContainer nodes;  
nodes.Create (2);  
  
NS_LOG_INFO ("Create channels.");
```

Μεταξύ των κόμβων υλοποιούνται point-to-point συνδέσεις όπως φαίνεται και στο παρακάτω στιγμιότυπο.

```
PointToPointHelper pointToPoint;  
pointToPoint.SetDeviceAttribute ("DataRate", StringValue ("500Kbps"));  
pointToPoint.SetChannelAttribute ("Delay", StringValue ("5ms"));
```

Σενάριο UDP

Σε αυτό το σενάριο, ορίζεται μια ενσύρματη τοπολογία μεταξύ τεσσάρων κόμβων, οι οποίοι ανταλλάσσουν UDP πακέτα. Ο κώδικας του σεναρίου αποτελείται από ένα αρχείο με κατάληξη .cc γραμμένο σε C++, το οποίο ξεκινάει με γραμμές σχολίων οι οποίες περιλαμβάνουν πληροφορίες για το περιεχόμενο του αρχείου.

Στη συνέχεια βρίσκεται ένα διάγραμμα του δικτύου, ενώ μετά ξεκινάει ο πηγαίος κώδικας με τον ορισμό των βιβλιοθηκών, όπως γίνεται σε κάθε ορθό C++ πρόγραμμα. Οι εντολές περιλαμβάνουν γνώριμες βιβλιοθήκες της C++, σε συνδιασμό με νέες που παρέχει ο προσομοιωτής.

Πανομοιότυπα με τα προηγούμενα σενάρια του προσομοιωτή, η λειτουργικότητα της προσομοίωσης υλοποιείται εντός της main().

Στο παρακάτω στιγμιότυπο πραγματοποιείται ο ορισμός των τεσσάρων κόμβων του σεναρίου.

```
NS_LOG_INFO ("Create nodes.");  
NodeContainer n;  
n.Create (4);  
  
InternetStackHelper internet;  
internet.Install (n);
```


Σειρά παίρνει η δημιουργία του καναλιού επικοινωνίας.

```
CsmaHelper csma;  
csma.SetChannelAttribute ("DataRate", DataRateValue (DataRate (5000000)));  
csma.SetChannelAttribute ("Delay", TimeValue (Milliseconds (2)));  
csma.SetDeviceAttribute ("Mtu", UIntegerValue (1400));  
NetDeviceContainer d = csma.Install (n);
```

Στη συνέχεια πραγματοποιείται ανάθεση IP διευθύνσεων σε κάθε κόμβο.

```
Ipv4AddressHelper ipv4;  
ipv4.SetBase ("10.1.1.0", "255.255.255.0");  
Ipv4InterfaceContainer i = ipv4.Assign (d);  
serverAddress = Address (i.GetAddress (1));
```

Στη συνέχεια ορίζεται ο UDP server καθώς και ο UDP client, οι οποίοι βρίσκονται σε κάθε κόμβο και αποστέλλουν πακέτα. Παρακάτω φαίνεται ο ορισμός του client.

```
uint32_t packetSize = 1024;  
uint32_t maxPacketCount = 1;  
Time interPacketInterval = Seconds (1.);  
UdpEchoClientHelper client (serverAddress, port);  
client.SetAttribute ("MaxPackets", UIntegerValue (maxPacketCount));  
client.SetAttribute ("Interval", TimeValue (interPacketInterval));  
client.SetAttribute ("PacketSize", UIntegerValue (packetSize));  
apps = client.Install (n.Get (0));  
apps.Start (Seconds (2.0));  
apps.Stop (Seconds (10.0));
```

Τέλος η προσομοίωση πραγματοποιείται μέσω ενός βρόγχου, ο οποίος έχει διάρκεια, όσο και η χρονική διάρκεια της προσομοίωσης.

Σενάριο Wireless

Στο Wireless σενάριο του NS-3 υλοποιείται μια ασύρματη δικτυακή τοπολογία όπου ρυθμίζονται οι κόμβοι στο επίπεδο 802.11b, σε ad hoc mode. Μεταξύ των κόμβων αποστέλλονται πακέτα των 1000 bytes. Τέλος ο ρυθμός λήψης των πακέτων ορίζεται ως σταθερός, ανεξάρτητα από την απόσταση των κόμβων αλλά και την ισχύ μετάδοσης.

Ο πηγαίος κώδικας της προσομοίωσης αποτελείται από ένα αρχείο το οποίο φέρει την κατάληξη .cc και εμπεριέχει το πρόγραμμα γραμμένο σε C++. Σύμφωνα με το πρότυπο συγγραφής κώδικα το οποίο έχει ορίσει η ομάδα ανάπτυξης του NS-3 το αρχείο, όπως και τα προηγούμενα, ξεκινάει με γραμμές σχολίων και πληροφορίες για το περιεχόμενο του αρχείου.

Το πρόγραμμα -όπως και όλα τα προγράμματα στην C++- ξεκινάει με τον ορισμό των βιβλιοθηκών, οι οποίες πέρα από κάποιες συνηθισμένες βιβλιοθήκες της C++ περιλαμβάνει και βιβλιοθήκες που παρέχει ο προσομοιωτής.

Όλη η λειτουργικότητα της προσομοίωσης υλοποιείται εντός της main(), αλλά και σε εξωτερικές συναρτήσεις, οι οποίες καλούνται εντός της main(). Στην προκειμένη περίπτωση ορίζεται η συνάρτηση ReceivePacket(), η οποία εμφανίζει ένα μήνυμα κάθε φορά που ένας κόμβος λαμβάνει κάποιο πακέτο καθώς και η GenerateTraffic(), που καλείται σε περίπτωση που κάποιος κόμβος θέλει να αποστείλλει ένα πακέτο.

Από τις μεταβλητές που ορίζονται στην συνάρτηση main() αξίζει να αναφερθούμε στις παρακάτω :

- packetSize - ορίζει το μέγεθος του πακέτου.
- numPackets - μετράει το πακέτα που ανταλλάσσονται στο δίκτυο.
- phyMode -ορίζει το είδος του δικτύου σε ad-hoc, ασύρματο δίκτυο.
- nodes -διατηρεί το πλήθος των κόμβων.
- verbose -καθορίζει την εμφάνιση των μηνυμάτων.

Η εντολή AddValue η οποία καλείται επί της cmd ορίζει σαν παραμέτρους χρήσης τα παρακάτω.

```
int main (int argc, char *argv[])
{
    std::string phyMode ("DsssRate1Mbps");
    double rss = -80; // -dBm
    uint32_t packetSize = 1000; // bytes
    uint32_t numPackets = 1;
    double interval = 1.0; // seconds
    bool verbose = false;

    CommandLine cmd;

    cmd.AddValue ("phyMode", "wifi Phy mode", phyMode);
    cmd.AddValue ("rss", "received signal strength", rss);
    cmd.AddValue ("packetSize", "size of application packet sent", packetSize);
    cmd.AddValue ("numPackets", "number of packets generated", numPackets);
    cmd.AddValue ("interval", "interval (seconds) between packets", interval);
    cmd.AddValue ("verbose", "turn on all wifiNetDevice log components", verbose);
}
```

Στην συνέχεια πριν δημιουργήσουμε τους κόμβους, ορίζουμε το χρονικό διάστημα μεταξύ της αποστολής των πακέτων και απενεργοποιούμε το fragmentation για όσα πακέτα είναι μικρότερα των 2200 bytes ενώ δημιουργούμε και τις δικτυακές διεπαφές, καθώς και το ασύρματο κανάλι επικοινωνίας.

```
WifiHelper wifi;
if (verbose)
{
    wifi.EnableLogComponents (); // Turn on all Wifi logging
}
wifi.SetStandard (WIFI_PHY_STANDARD_80211b);
```

Παρακάτω ορίζεται το μοντέλο, βάσει του οποίου κινούνται εντός δικτύου οι κόμβοι.

```
MobilityHelper mobility;
Ptr<ListPositionAllocator> positionAlloc = CreateObject<ListPositionAllocator> ();
positionAlloc->Add (Vector (0.0, 0.0, 0.0));
positionAlloc->Add (Vector (5.0, 0.0, 0.0));
mobility.SetPositionAllocator (positionAlloc);
mobility.SetMobilityModel ("ns3::ConstantPositionMobilityModel");
mobility.Install (c);
```

Στην πορεία του σεναρίου και αφού ορίσουμε το δίκτυο μας να λειτουργεί ad-hoc, ορίζουμε το TCP/IP πρωτόκολλο που θα χρησιμοποιηθεί για την κατανομή IP διευθύνσεων στους κόμβους. Τέλος ζητάμε να ξεκινήσει η προσομοίωση.

Αξίζει να σημειωθεί ότι το εν λόγω δίκτυο, καθώς είναι ασύρματο δεν μπορεί να οπτικοποιηθεί στον NetAnim. Προσωρινή λύση μας δίνει το wireshark όπου μπορούμε να εισάγουμε τα παραγόμενα pcap αρχεία της προσομοίωσης για μελέτη.

9. Αποτελέσματα Προσομοιώσεων

Στο παρακάτω κεφάλαιο θα διαμορφωθούν οι τελικές απόψεις για τους δυο προσομοιωτές μέσα από την θεωρητική και πρακτική σύγκριση που πραγματοποιήθηκε παραπάνω.

9.1 Προγραμματιστικές γλώσσες

Όσον αφορά τον κώδικα των δυο προσομοιωτών, η C++ είναι πιο σύνθετη και δυσανάγνωστη για τους χρήστες ενώ η Tcl θεωρείται πιο απλή και προσιτή.

Ο NS-2 υλοποιείται χρησιμοποιώντας έναν συνδυασμό Tcl (για τα σενάρια τα οποία περιγράφουν την τοπολογία του δικτύου) και C++ (ο πυρήνας του προσομοιωτή). Αυτό το σύστημα επιλέχθηκε στις αρχές της δεκαετίας του 1990 με σκοπό να αποφευχθεί η επαναμεταγλώττιση της C++, διαδικασία αρκετά χρονοβόρα για την εποχή σε σύγκριση με την μεταγλώττιση της Tcl, της μόνης διαθέσιμης *scripting* γλώσσας τότε. Ο NS-3 από την άλλη υλοποιείται με την χρήση C++ καθώς με τις σύγχρονες δυνατότητες, ο χρόνος μεταγλώττισης δεν αποτέλεσε πρόβλημα ενώ υποστηρίζει και την Python κατά την εκτέλεση και την οπτικοποίηση των αποτελεσμάτων.

Η C++ του NS-3, ως προς το συντακτικό είναι πιο σύνθετη για τους χρήστες ενώ η Tcl θεωρείται απλή και προσιτή. Αυτό δεν σημαίνει ότι τα σενάρια στον NS-2 είναι πιο ευανάγνωστα και αυτό διότι τα σενάρια του NS-3 είναι μικρότερα ως προς το μέγεθος του κώδικα με λιγότερες δομές ελέγχου. Η ώριμη σχεδίαση του NS-3 οδηγεί σε καλύτερη ενθυλάκωση, ενώ ο ενιαίος τρόπος γραφής του κώδικα εξάγει πιο απλά προς τον χρήστη προγράμματα.

9.2 Χρόνος εκτέλεσης

Όπως αναφέρθηκε και στα αρχικά κεφάλαια για να είναι αξιοποιήσιμη μια προσομοίωση πρέπει να επιτρέπει την λήψη και την παρατήρηση των αποτελεσμάτων. Απαραίτητη προϋπόθεση σε έναν προσομοιωτή, είναι ο απαιτούμενος χρόνος για το πέρας μια προσομοίωσης να είναι μικρός και να μην αυξάνεται όσο αυξάνεται το μέγεθος του υπό εκτέλεσης αρχείου.

Σε αυτό τον τομέα η μελέτη αλλά και η εκτέλεση των σεναρίων της προσομοίωσης, έδειξε πως τα σεναρία του σύγχρονου NS-3 "τρέχουν" πιο γρήγορα σε σχέση με τα αντίστοιχα του NS-2.

Κάτω από όμοιες συνθήκες ο NS-3 πλεονεκτεί σε ταχύτητα, αποτέλεσμα αναμενόμενο, καθότι, τα σεναρία προσομοίωσης του είναι προγράμματα γραμμένα σε C++ ενώ τα σεναρία του NS-2 διερμηνεύονται σε μια μίξη C/C++ πριν εκτελεστούν. Στον NS-3 υπάρχει το μειονέκτημα της μεταγλώττισης κάθε φορά που πραγματοποιείται μια αλλαγή στον κώδικα, χρόνος ο οποίος μπορεί να χαρακτηριστεί και ως αμελητέος.

9.3 Αποτελέσματα και οπτικοποίηση

Με την ολοκλήρωση μιας προσομοίωσης στον NS-2 εξάγονται τα αποτελέσματα σε tracefiles, τα οποία περιέχουν αποτελέσματα της προσομοίωσης και σε αρχεία .nam, ένα σύστημα απεικόνισης βασισμένο σε Tcl, το οποίο παράγει γραφική αναπαράσταση του υπό μελέτη δικτύου. Τα εξαχθέντα αποτελέσματα του προσομοιωτή, είναι έως και τρεις φορές μεγαλύτερα από αυτά του NS-3 . Αυτό έχει ως αποτέλεσμα , σε περίπτωση που το υπό προσομοίωση δίκτυο είναι μεγάλο, να υπάρχουν δυσκολίες μεταφοράς του καθώς και ανάγνωσής του από κάποιο άλλο

λογισμικό. Στον NS-3 τα αποτελέσματα εξάγονται σε Xml, Wireshark (.pcap) , καθώς και σε tracefiles που αναφέρθηκαν προηγουμένως. Τα αρχεία Xml χρησιμοποιούνται από το λογισμικό NetAnim για την οπτικοποίηση των αποτελεσμάτων ενώ τα αρχεία pcap από το λογισμικό Wireshark.

Στον τομέα της οπτικοποίησης των σεναρίων, μέχρι στιγμής ο Nam του NS-2 έχει αποδεχθεί ένας εξαιρετικός animator δικτύου παρά την αρχέγονη σχεδίασή του. Στον NS-3 ο αξιόπιστος NetAnim και ο PyViz χρήζουν βελτίωσης στο άμεσο μέλλον καθώς δεν αποτελούν κομμάτι της διανομής του NS-3 με την ξεχωριστή εγκατάστασή τους ενώ τους υπολείπονται πολλά σημαντικά χαρακτηριστικά. Άξιο αναφοράς είναι το γεγονός ότι ο NetAnim αδυνατεί να οπτικοποιήσει ασύρματα δίκτυα.

10. Επίλογος-Συμπεράσματα

Υπάρχουν τόσες πολλές προκλήσεις που αντιμετωπίζει το πεδίο της προσομοίωσης σήμερα. Κανένας προσομοιωτής δεν ικανοποιεί πλήρως τις ανάγκες των σημερινών χρηστών. Πρέπει να πραγματοποιηθούν τόσες πολλές έρευνες και συγκρίσεις πριν τη σχεδίαση ενός προσομοιωτή. Σαν τεχνική, η προσομοίωση μας δίνει τη δυνατότητα να εξετάσουμε πιθανά ενδεχόμενα/αποτελέσματα από πιθανές βελτιώσεις των υπό μελέτη σεναρίων. Κάθε αλλαγή που πραγματοποιείται σε ένα δίκτυο, ή ακόμα και η σχεδίαση ενός νέου δικτύου μπορεί να αναλυθεί μέσα από τη σχεδίαση και την προσομοίωση της τοπολογίας του δικτύου μέσω των ποικίλων περιβαλλόντων προσομοίωσης.

Αλλάζοντας τους αλγόριθμους δρομολόγησης και τα δίκτυα, καταγράφοντας τα αποτελέσματα οδηγούμαστε στον βέλτιστο αλγόριθμο. Στο βέλτιστο αλγόριθμο του κάθε δικτύου, μας οδηγεί η ανάλυση των αποτελεσμάτων της κάθε προσομοίωσης. Τα μικρότερα δίκτυα μπορούν να αναλυθούν και τα αποτελέσματα αυτών να

χρησιμοποιηθούν σε μεγαλύτερα δίκτυα κάνοντας τη δρομολόγηση πιο οικονομική και αποτελεσματική.

Ο δικτυακός προσομοιωτής Ns-2 έχοντας επιτύχει τους σκοπούς του από τα πρώτα χρόνια λειτουργίας του, πλέον δεν αναπτύσσεται. Ο καινούριος Ns-3 ήρθε για να εκμηδενίσει τους περιορισμούς του προκατόχου του δημιουργώντας ενδιαφέρον και νέα πλεονεκτήματα για την ανάπτυξη νέων μοντέλων από τους χρήστες. Ο Yans, ο GTNets και ο Ns-2, συνθέτουν την σημερινή εικόνα του Ns-3 ο οποίος σε αντίθεση με την OTel του Ns-2, είναι γραμμένος σε C++ χρησιμοποιώντας διεπαφές Python. Ως προς τον ρεαλισμό των αποτελεσμάτων, τα πρωτόκολλα που υλοποιούνται στον Ns-3, είναι σχεδιασμένα έτσι ώστε να αναπαριστούν τις πραγματικές υλοποιήσεις όσο το δυνατόν πιστότερα.

Ωστόσο, υπάρχουν σημαντικές αλλαγές που αναμένεται να πραγματοποιηθούν στον υπό ανάπτυξη Ns-3 από την ερευνητική και ακαδημαϊκή κοινότητα. Σημαντικότερες εξ αυτών είναι η δημιουργία tutorials έτσι ώστε και οι πιο απλοί χρήστες να έχουν την δυνατότητα κατανόησης και χρήσης του προσομοιωτή, όπως επίσης και η υποστήριξη κώδικα ο οποίος έχει μετατραπεί από άλλη δικτυακή πλατφόρμα.

Απαραίτητη προϋπόθεση για να φθάσει ο Ns-3 την επιτυχία του Ns-2 είναι η συμβολή με μοντέλα και κώδικα επιστημονικών φορέων και οργανισμών. Έτσι έχοντας την απαραίτητη συντήρηση και ανάπτυξη από ειδικευμένους, ο Ns-3 μπορεί να αποκτήσει τα πλεονεκτήματα που έχουν πολλοί από τους καλύτερους δικτυακούς-εμπορικούς προσομοιωτές, όπως ο OPNET. Όντας ένα ενεργό εργαλείο ανοικτού κώδικα με πλεονεκτήματα αλλά και ελλείψεις, ο Ns-3 διαθέτει χαρακτηριστικά μέσω των οποίων μπορεί να αναδείξει τις ικανότητές του, έχοντας όμως την συμβολή οργανισμών και εξειδικευμένων ανθρώπων.

Συνοψίζοντας, η σχεδίαση και η λειτουργία ενός πραγματικού δικτύου με την

βοήθεια των εργαλείων προσομοίωσης, αποτελεί όχι μόνο ένα εκπαιδευτικό βοήθημα κατανόησης των βασικών εννοιών, της λειτουργίας και των αρχών σχεδίασης ενός δικτύου αλλά και ένα πολύ σημαντικό εργαλείο μέσα από το οποίο οι χρήστες έχουν τη δυνατότητα να εξετάσουν και να αναλύσουν τυχόν επεκτάσεις ενός δικτύου, (ασύρματες συνδέσεις, προσθήκη αισθητήρων) αλλά και τα πολλαπλά περιθώρια βελτίωσης της λειτουργίας του, σαν αποτέλεσμα τροποποιήσεων σε αυτό.

Βιβλιογραφίες-Πηγές

- Modeling and Tools for Network Simulaion (Wehrle, Klaus, Gunes, Mesut, 2010)
- Network Modelling and Simulation : A Practical Perspective (Mohsen Guizani, Ammar Rayaes, Bilal Khan, Ala Al-Fuqaha 2010)
- Simulation in Computer Network Design and Modeling : Use and Analysis (Hussein Al-Bahadili 2012)
- Introduction to Network Simulator NS2 (Teerawat Issariyakul , Ekram Hossain 2010)
- Advances in network simulation (IEEE Computer Society 2012)
- A Comprehensive Overview on Different Network Simulators (Department of Information Technology, Karunya University 2013)
- Key Features and Optimum Performance of Network Simulators (Dr. Ahsan Latif, Muddasar Ahmad 2013)
- Τεχνικές Προσομοίωσης (Εκδόσεις Τζιόλα 2012)
- <https://networksimulationtools.com/>
- <http://www.opnet.org>
- <http://www.physiome.org/jsim/>
- <http://www.ssfnet.org/>
- <http://www.boson.com/netsim-cisco-network-simulator>
- <http://www.nest-simulator.org/>
- <http://www-x.antd.nist.gov/nistnet/>
- <http://www.csse.uwa.edu.au/cnet/>
- <http://www.ssfnet.org/homePage.html>
- <http://www.oversim.org/>
- <http://tetcos.com/physim.html>
- <https://optics.synopsys.com/rsoft/rsoft-system-network-optsim.html>
- http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=478802&url=http%3A%2F%2Fieeexplore.ieee.org%2Fexpls%2Fabs_all.jsp%3Farnumber%3D478802
- <https://archive.org/details/computernetworks00sull>
- <http://www.kt.agh.edu.pl/~brus/atm/Comnet-Tutorial.pdf>
- <http://www.kitchenlab.org/www/bmah/Software/Insane/>
- <http://ccr.sigcomm.org/online/files/new.pdf>
- <http://www2.ece.gatech.edu/research/labs/MANIACS/GTNetS/>
- <http://nsl.cs.nctu.edu.tw/NSL/nctuns.html>
- <http://parsec.cs.princeton.edu/parsec3-doc.htm>
- <http://realsimulator.com/>
- <http://web.scalable-networks.com/content/qualnet>

- <http://www.ijarce.com/upload/june/3-Network%20Simulation%20Tools%20Survey.pdf>
- <http://arxiv.org/ftp/arxiv/papers/1307/1307.4129.pdf> (Cornell University Library)
- <http://www.ijcsit.com/docs/Volum%203/Vol3Issue1/ijcsit2012030129.pdf> (*IIMT Engineering College 2012*)
- <http://www.inrialpes.fr/planete/people/ernst/Documents/simulator.html#NETSIM>
- <http://www2.ece.gatech.edu/research/labs/MANIACS/GTNetS/>
- <https://optics.synopsys.com/rsoft/rsoft-system-network-optsim.html>
- <http://www.isi.edu/home>
- <https://www.nsnam.org/>
- <https://www.nsnam.org/ns-3-19/download/>
- <http://www.isi.edu/nsnam/ns/ns-build.html>
- <http://otcl-tclcl.sourceforge.net/otcl/>
- http://www.sv.vt.edu/classes/ESM4714/Student_Proj/class03/stillwater/prj/project/sim_pandc.htm
- <http://www.concentricabm.com/blog//advantages-and-disadvantages-of-simulation>
- <http://www.cse.wustl.edu/~jain/cse567-08/ftp/simtools/>
- <http://vlaurie.com/computers2/Articles/protocol.htm>
- <http://iitkgp.vlab.co.in/?sub=38&brch=121&sim=561&cnt=1>
- <http://www.wisegeek.com/what-is-network-simulation.htm>
- <https://www.wireshark.org/>
- <http://arxiv.org/ftp/arxiv/papers/1307/1307.4129.pdf>

