



**ΠΑΝΕΠΙΣΤΗΜΙΟ
ΠΑΤΡΩΝ**
UNIVERSITY OF PATRAS



**Τμήμα Μηχανικών Ηλεκτρονικών Υπολογιστών
και Πληροφορικής**

**Ανάπτυξη εργαλείου αυτοματοποίησης και διαχείρισης
εκτέλεσης εξομοιώσεων**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΔΡΑΚΟΥΛΕΛΗ ΜΙΧΑΛΗ

Υπεύθυνος Καθηγητής: Μπούρας Χρήστος
Καθηγητής Πανεπιστημίου Πατρών

Επιβλέπων: Στάμος Κώστας

Πάτρα, Μάιος 2015

Πίνακας Περιεχομένων

ΠΕΡΙΛΗΨΗ	6
ABSTRACT	8
1. Εισαγωγή	11
1.1. Προσομοίωση Δικτύων	11
1.2. Αντικείμενο Διπλωματικής	13
1.3. Οργάνωση Κειμένου	14
1.4. Δημοσιεύσεις	15
2. Σχετικές Εργασίες	17
2.1. Εκτέλεση προσομοιώσεων.....	18
2.2. Σύνθεση πριν την προσομοίωση.....	19
2.3. Ανάλυση μετά την προσομοίωση.....	20
3. Θεωρητικό Υπόβαθρο.....	22
3.1. Η δομή ενός trace file	22
3.2. Metafiles: Η ιδέα του TRAFIL.....	30
3.2.1. Metafiles.....	31
3.2.2. Submetafiles.....	35
3.3. Η δομή ενός TCL αρχείου.....	37
3.3.1. Κόμβοι δικτύου	38
3.3.2. Σύνδεσμοι δικτύου	38
3.3.3. Γεννήτριες κυκλοφορίας/Εφαρμογές	39
3.3.4. Δικτυακά πρωτόκολλα	39
3.3.5. Παράμετροι ασύρματων κόμβων	40
4. Αρχιτεκτονική του TRAFIL.....	42
4.1. Οι λειτουργικές μονάδες του TRAFIL.....	42
4.2. Περιγραφή λειτουργιών του TRAFIL	44
4.2.1. Κύρια Υπολογιστική Μονάδα (Main Processing Module)	45
4.2.2. Μονάδα Διαχείρισης Metafiles & Submetafiles (Metafile & Submetafile Handler)	47
4.2.3. Μονάδα Παραγωγής Μετρήσεων (Metrics Module).....	50
4.2.4. Μονάδα Παραγωγής Γραφικών Παραστάσεων (Charts Module)	51
4.2.5. Μονάδα Εκτέλεσης Ερωτημάτων προς τη Βάση Δεδομένων (SQL Queries Module).....	52

4.2.6.	Μονάδα Προσομοίωσης (Simulation Module)	52
4.2.7.	Μονάδα Σχεδιασμού Προσομοίωσης (Simulation Plane Design Module)..	56
5.	Σχεδίαση Συστήματος	61
5.1.	Αρχιτεκτονική	61
5.2.	Ανάλυση λειτουργίας TCL Design	77
6.	Υλοποίηση	79
6.1.	Παραδείγματα χρήσης TRAFIL	79
6.1.1.	Είσοδος στο TRAFIL.....	79
6.1.2.	Γενικές πληροφορίες της προσομοίωσης	81
6.1.3.	Υπολογισμός μετρήσεων.....	82
6.1.4.	Εκτέλεση ερωτημάτων προς την βάση δεδομένων	83
6.1.5.	Εκτέλεση προσομοιώσεων.....	84
6.1.6.	Σχεδίαση προσομοιώσεων	87
6.1.7.	Σχεδίαση γραφικών παραστάσεων	91
6.2.	Υποστηριζόμενες πλατφόρμες και προαπαιτούμενα προγραμματιστικά εργαλεία	93
7.	Αξιολόγηση	93
8.	Επίλογος	96
8.1.	Μελλοντικές επεκτάσεις.....	96

Πίνακας Εικόνων

Εικόνα 1 Παράδειγμα ενός normal trace file	29
Εικόνα 2 Normal trace file με επιπλέον TCP κεφαλίδα.....	30
Εικόνα 3 Παράδειγμα New Wireless trace file	30
Εικόνα 4 Normal metafile	32
Εικόνα 5 Old Wireless metafile	32
Εικόνα 6 New Wireless metafile	34
Εικόνα 7 Normal TCP metafile	36
Εικόνα 8 Old Wireless AODV Request submetafile	36
Εικόνα 9 Old Wireless IP submetafile	36
Εικόνα 10 Παράδειγμα συσχετισμών εφαρμογής - agent.....	40
Εικόνα 11 Παράδειγμα δήλωσης agent	40
Εικόνα 12 Αρχιτεκτονική TRAFIL	44
Εικόνα 13 Κύρια υπολογιστική μονάδα	45
Εικόνα 14 Μονάδα διαχείρισης metafiles (Metafile Handler)	47
Εικόνα 15 Μονάδα διαχείρισης metafiles (Metafile Handler)	51
Εικόνα 16 Μονάδα Εκτέλεσης Ερωτημάτων προς τη Βάση Δεδομένων (SQL Queries Module)	52
Εικόνα 17 Παράδειγμα MP4.exe trace file.....	54
Εικόνα 18 Διαδικασία εκτέλεσης προσομοίωσης με Evalvid-RA και NS-2	55
Εικόνα 19 Μονάδα Προσομοίωσης (Simulation Module).....	56
Εικόνα 20 Μονάδα Σχεδιασμού Προσομοίωσης (Simulation Plane Design Module)	58
Εικόνα 21 Μονάδα Σχεδιασμού Προσομοίωσης (Simulation Plane Design Module)	60
Εικόνα 22 Η κλάση TRAFIL	63
Εικόνα 23 Η κλάση TraceFileHandler	63
Εικόνα 24 Η κλάση MetaFiles.....	64
Εικόνα 25 Οι κλάσεις MetaDataHandler, SubMetaDataHandler και MetaEntry, καθώς και οι μεταξύ τους σχέσεις	65
Εικόνα 26 Η κλάση TraceFileInfo.....	65
Εικόνα 27 Οι κλάσεις GeneralInformation και GeneralSimulationInformation.....	66
Εικόνα 28 Οι κλάσεις GeneralNodeInformation και Metrics	67
Εικόνα 29 Η κλάση Graph	68
Εικόνα 30 Η κλάση ThroughputChart.....	68
Εικόνα 31 Η κλάση ThroughputBitsChart	68
Εικόνα 32 Η κλάση DelayJitterChart	68
Εικόνα 33 Η κλάση PacketEndtoEndDelayChart.....	68
Εικόνα 34 Η κλάση Simulator	69
Εικόνα 35 Η κλάση NetworkSimulator	69
Εικόνα 36 Η κλάση VideoSimulator	69
Εικόνα 37 Η κλάση VideoPostSimulator	69
Εικόνα 38 Η κλάση SimulationResultPresenter	70
Εικόνα 39 Ενέργειες κατά τη φάση pre-simulation & simulation	71
Εικόνα 40 Ενέργειες κατά τη φάση post-simulation	72
Εικόνα 41 Η κλάση TclDesignerPanel	73
Εικόνα 42 Η κλάση NodePropertiesWindow	73

Εικόνα 43 Οι κλάσεις LinkListWindow, ConnectedAgentsWindow και SimulationPropertiesWindow.....	74
Εικόνα 44 Η κλάση TclDesignLink.....	75
Εικόνα 45 Οι κλάσεις TclDesignNode, TclDesignWiredNode και TclDesignWirelessNode	75
Εικόνα 46 Λειτουργική σχέση μεταξύ της κλάσης TclDesignerPanel και των βοηθητικών κλάσεων της.....	76
Εικόνα 47 Αναζήτηση νέων διασυνδεδεμένων κόμβων κατά τη σύνδεση δυο κόμβων μεταξύ τους.....	78
Εικόνα 48 Οθόνη εισόδου στο TRAFIL	79
Εικόνα 49 Κύρια οθόνη του TRAFIL	80
Εικόνα 50 Παράδειγμα απλού δικτύου τεσσάρων κόμβων	81
Εικόνα 51 Οθόνη γενικών πληροφοριών	82
Εικόνα 52 Οθόνη μετρήσεων	83
Εικόνα 53 Οθόνη ερωτημάτων προς τη βάση δεδομένων.....	84
Εικόνα 54 Η καρτέλα Evalvid-RA Simulation.....	85
Εικόνα 55 Οθόνη αποτελέσματος προσομοίωσης Evalvid-RA.....	86
Εικόνα 56 Οθόνη αποτελέσματος φάσης post-simulation του Evalvid-RA	87
Εικόνα 57 Η καρτέλα TCL Design του TRAFIL, με σχηματισμένο ενσύρματο δίκτυο....	88
Εικόνα 58 Παράμετροι κόμβου	89
Εικόνα 59 Οθόνη λίστας συνδέσμων	90
Εικόνα 60 Οθόνες συνδεδεμένων agents και επιλογών για ασύρματα δίκτυα.....	90
Εικόνα 61 Οθόνη δημιουργίας γραφικών παραστάσεων	92

ΠΕΡΙΛΗΨΗ

Ο στόχος της διπλωματικής αυτής εργασίας είναι η ανάπτυξη, βελτίωση και εξέλιξη ενός εργαλείου με σκοπό την διευκόλυνση της χρηστικότητας προσομοιωτών δικτύων. Στις υπηρεσίες που προσφέρονται περιλαμβάνονται η αυτοματοποίηση τετριμμένων διαδικασιών κατά τις προσομοιώσεις, η καλύτερη και εκτενέστερη αναπαράσταση αποτελεσμάτων και μετρήσεων σε προσομοιώσεις, καθώς και η ευκολότερη δημιουργία νέων προσομοιώσεων. Το εργαλείο αυτό ονομάστηκε TRAFIL (αρχικά των TRAcE FILE, αρχείων στα οποία παρουσιάζονται τα αποτελέσματα των προσομοιώσεων) και είναι γραμμένο κατά κύριο λόγο σε γλώσσα προγραμματισμού Java.

Το TRAFIL εκτελεί πολλαπλές λειτουργίες που βασίζονται στην ανάλυση και επεξεργασία των trace files του Network Simulator 2 (NS-2), αλλά έχει σχεδιαστεί με την προοπτική περαιτέρω επέκτασής του ώστε να υποστηρίζει trace files οποιουδήποτε προσομοιωτή. Αρχικός και κύριος σκοπός του είναι η επιτάχυνση των διαδικασιών ανάλυσης και παραγωγής μετρήσεων από τα αποτελέσματα προσομοιώσεων μεγάλου μεγέθους. Για το λόγο αυτό χρησιμοποιεί έναν καινοτόμο τρόπο διαχείρισης των trace files, παρουσιάζοντας την ιδέα των metafiles. Στα αρχεία αυτά κωδικοποιείται η δομή ενός τύπου trace file, ώστε αργότερα να μπορούν να χρησιμοποιηθούν για την αναγνώριση και ανάλυση οποιουδήποτε αποτελέσματος έχει προτυποποιηθεί με αυτή τη μορφή. Έτσι εμφανίζεται μια πιο αφηρημένη μέθοδος επεξεργασίας των trace files, η οποία μάλιστα προσφέρει και την επιθυμητή επεκτασιμότητα.

Ως φυσική συνέχεια της ανάλυσης ενός trace file, το TRAFIL εμφανίζει τα αποτελέσματα με φιλικό προς το χρήστη τρόπο, παρέχοντας επιπλέον μετρήσεις και δυνατότητες γραφικής αναπαράστασης, καθώς και προσαρμοσμένων συγκρίσεων. Επίσης αποθηκεύει όλα αυτά τα δεδομένα σε μια τοπική βάση δεδομένων, καθιστώντας τα έτσι γρήγορα και άμεσα διαθέσιμα, χωρίς να απαιτείται η επαναφόρτωση του trace file.

Εκτός αυτής της λειτουργίας, το TRAFIL έχει την δυνατότητα να εκτελέσει έμμεσα σενάρια προσομοιώσεων γραμμένα σε γλώσσα κατανοητή από τον NS-2 (OTcl). Απαραίτητη προϋπόθεση, φυσικά, να είναι εγκατεστημένος σε λειτουργικό σύστημα Linux, ώστε να λειτουργεί χωρίς υποστήριξη τρίτων βιβλιοθηκών.

Μια από τις πιο δημοφιλείς επεκτάσεις του NS-2 είναι το Evalvid-RA, το οποίο προσομοιώνει την μετάδοση αρχείων βίντεο μέσω streaming. Το TRAFIL, λαμβάνοντας υπόψιν την δυνατότητα εκτέλεσης σεναρίων που αναφέρθηκε παραπάνω, επεκτάθηκε

ώστε να διευκολύνει την διαδικασία χρήσης της επέκτασης αυτής, προσφέροντας ένα πλήρες γραφικό περιβάλλον πριν και μετά την προσομοίωση.

Μια τελευταία αλλά εξίσου σημαντική λειτουργία του TRAFIL είναι η δημιουργία σεναρίων προσομοιώσεων δικτύων μέσω γραφικού περιβάλλοντος. Συγκεκριμένα, έχει αναπτυχθεί ένα γραφικό περιβάλλον σχεδίασης στο οποίο δίνονται τα κατάλληλα εργαλεία ώστε να σχεδιαστεί σχεδόν οποιοσδήποτε τύπος δικτύου, καθώς και να οριστούν λεπτομέρειες που δεν είναι εμφανίσιμες σχηματικά. Το TRAFIL αναλαμβάνει να μοντελοποιήσει το σχεδιασμένο σενάριο βάσει των δομών που διέπουν τον NS-2, και να εξάγει ένα εκτελέσιμο script file σε γλώσσα Tcl. Η λειτουργία αυτή συνδυάζεται με τις προηγούμενες ώστε να μπορεί να γίνει άμεση δοκιμαστική εκτέλεση και ανάλυση αποτελεσμάτων ενός σεναρίου.

ABSTRACT

The goal of this thesis is the development, enhancement and evolution of a tool with the purpose of adding usability in common network simulators. The aforementioned tool provides services that include the automation of trivial tasks involved in the simulation process, better and wider result presentation as well as easier simulation creation. It was named TRAFIL (after TRAcE FILE, the type of result files that simulations produce) and is written mostly in Java.

TRAFIL has many functions that are based on the analysis and processing of trace files of the Network Simulator 2 (NS-2), but it is also designed with an expansive perspective so that it may process trace files of any simulator. Its first and most basic goal is to accelerate the procedures of trace file analysis and metrics production of large scale simulations. For that purpose it utilizes a novel way of managing trace files, the metafiles. These are special files that encode the structure of a trace file type so that it can later be used in order to recognise any trace file that has similar traits. Thus a level of abstraction is introduced in trace file processing, while also granting the desired expandability.

Furthermore, TRAFIL presents the simulation results in a user-friendly way, providing features such as custom metrics production, graph creation and metrics comparisons. In addition, it stores the data produced in a local database so that it can be retrieved quickly, avoiding the trace file parsing procedure whatsoever.

Besides that, TRAFIL may execute simulation scenarios written in OTcl, a language understandable by NS-2. However it requires that the simulator is installed in a Linux operating system, so that it may be executed without third party libraries.

One of the most popular NS-2 extensions is Evalvid-RA, which enables video streaming simulation. Given that, TRAFIL facilitates the entire procedure using a graphic interface before as well as after the simulation.

Last but not least, TRAFIL allows for the creation of simulation scenarios using a graphic user interface (GUI). Specifically, it provides a design interface where any type of network may be designed in detail. Then it produces an executable script file in OTcl language. This function is supplementary to the ones that are mentioned above so that the script file may be executed immediately, and the produced trace file may be analyzed through TRAFIL.

Keywords: Network Simulation, TRAFIL, trace file, metafile, NS-2, simulation design, Evalvid-RA,
simulation processing, TCL

ΠΡΟΛΟΓΟΣ

Πριν την παρουσίαση του έργου της διπλωματικής θα ήθελα να εκφράσω τις θερμές ευχαριστίες μου προς τον καθηγητή μου κ. Μπούρα για την εμπιστοσύνη και την παρότρυνση που μου έδειξε καθ' όλη τη διάρκεια της εκπόνησής της. Χωρίς την δική του συγκατάθεση και βοήθεια δεν θα ήταν δυνατή η ανάδειξή της σε διεθνή μέσα.

Θα ήθελα επίσης να ευχαριστήσω όλους όσους με βοήθησαν κατά την διάρκεια της με χρήσιμες συμβουλές και καθοδήγηση. Θα ήθελα ιδιαίτερα να ευχαριστήσω τον Δρ. Κώστα Στάμο για την συνεχή υποστήριξη και συμπαράσταση που μου έδειξε, καθώς και τον Σάββα Χαραλαμπίδη για την πολύτιμη βοήθειά του στην απαρχή της ενασχόλησής μου με το έργο αυτό. Άλλωστε χωρίς την δική του αρχική ιδέα, σκέψη και υλοποίηση το εργαλείο αυτό δεν θα υπήρχε.

Επίσης θέλω να ευχαριστήσω και τους υπόλοιπους συναδέλφους μου στο Εργαστήριο Τηλεματικής του Τμήματος Ηλεκτρονικών Υπολογιστών και Πληροφορικής του Πανεπιστημίου Πατρών, καθώς και στην Ερευνητική Μονάδα 6 του Ινστιτούτου Τεχνολογίας Υπολογιστών και Εκδόσεων (ITYE «Διόφαντος») για την ευχάριστη συνεργασία καθώς και την συνεχή αλληλοβοήθεια.

Τέλος θα ήθελα να ευχαριστήσω την οικογένεια και τους φίλους μου για την αμέριστη συμπαράστασή τους καθ' όλη τη διάρκεια του έργου, αφού χωρίς αυτήν δεν θα ήταν δυνατή η ομαλή ολοκλήρωση των πανεπιστημιακών σπουδών μου.

1. Εισαγωγή

1.1. Προσομοίωση Δικτύων

Η προσομοίωση δικτύων¹ είναι μια μέθοδος η οποία χρησιμοποιεί εργαλεία για να μοντελοποιήσει την συμπεριφορά ενός πραγματικού δικτύου, συνήθως τηλεπικοινωνιακού, ή δικτύου υπολογιστών. Η μοντελοποίηση συνήθως γίνεται βάσει συγκεκριμένων μαθηματικών προτύπων, τα οποία συνήθως καθορίζονται από τα ίδια τα πρωτόκολλα επικοινωνιών. Προσομοιώσεις δικτύων γίνονται κατά κόρον για ερευνητικούς σκοπούς, αφού δίνεται μεγάλη δυνατότητα πειραματισμού σε ένα συγκεκριμένο δίκτυο πριν αυτό εφαρμοστεί σε πραγματικές συνθήκες. Διαπιστώνεται έτσι εύκολα πως αυτό το δίκτυο θα συμπεριφέρεται σε διάφορες περιστάσεις και γίνονται οι κατάλληλες τροποποιήσεις πριν την δοκιμή στον πραγματικό κόσμο. Γίνεται προφανές ότι τέτοιου είδους τροποποιήσεις σε ένα δίκτυο παραγωγής, είτε σημαντικές, είτε όχι, κοστίζουν τόσο σε χρόνο όσο και σε χρήμα πολύ περισσότερο από την χρήση προσομοιώσεων, ενώ υπάρχει, φυσικά, και η περίπτωση να μην είναι υλοποιήσιμες.

Η διαδικασία προσομοίωσης συνήθως περιλαμβάνει την μοντελοποίηση ενός δικτύου κατά τις προτιμήσεις του ερευνητή, την εφαρμογή του με τη χρήση ειδικών εργαλείων, των προσομοιωτών δικτύων, και τέλος την παρατήρηση των αποτελεσμάτων.

Οι προσομοιωτές δικτύων είναι ένας πολύ ευέλικτος τύπος εργαλείων κατάλληλος για ερευνητές και μηχανικούς. Εξυπηρετούν εύκολα την ανάγκη για αποδοτική σχεδίαση και έλεγχο ενός μοντέλου δικτύου με ελάχιστο κόστος. Η χρησιμότητά τους είναι τόσο μεγάλη, ώστε έχει εμφανιστεί μια πληθώρα προσομοιωτών, οι οποίοι μπορεί να είναι είτε εμπορικοί, είτε ανοιχτού λογισμικού (open-source). Ένας από τους πιο ευρέως διαδεδομένους προσομοιωτές δικτύων ανοιχτού λογισμικού είναι ο Network Simulator 2 (NS-2)².

Ο NS-2 είναι ευρέως διαδεδομένος στην ακαδημαϊκή, την ερευνητική, αλλά και την βιομηχανική κοινότητα. Δημιουργήθηκε το 1997, βασίζεται στην ιδέα των διακριτών συμβάντων (discrete event simulation) και είναι υλοποιημένος σε C++ και OTcl³, μια αντικειμενοστραφή εκδοχή της Tcl σχεδιασμένη από το MIT. Παρότι είναι σχεδιασμένος για Unix συστήματα, μπορεί να τρέξει με τις κατάλληλες προσαρμογές σχεδόν σε όλα τα σύγχρονα λειτουργικά συστήματα.

Χαρακτηριστικό της δημοτικότητάς του είναι το γεγονός ότι, ενώ το κύριο πρόγραμμα αποτελείται περίπου από 300,000 γραμμές κώδικα, έχει τουλάχιστον άλλες

τόσες σε τρίτες συνεισφορές και παραλλαγές του. Αυτό οφείλεται εν μέρει στο πλεονέκτημα που έχει ως ανοιχτό λογισμικό, αλλά και στον προσεγμένο σχεδιασμό του ώστε να επεκτείνεται εύκολα. Η επέκταση του NS-2 μπορεί να γίνει εύκολα από οποιονδήποτε θέλει να σχεδιάσει ένα απλό πρωτόκολλο ή ακόμα και κάτι πολύ πιο πολύπλοκο, όπως ένα ολόκληρο framework. Έτσι ο προσομοιωτής μένει ενημερωμένος και αποκτά συνεχώς νέες λειτουργίες ανεξάρτητα από όσες προσφέρει εγγενώς.

Βέβαια, η πλειοψηφία των προσομοιώσεων που μπορεί να απαιτηθούν ικανοποιούνται από τις λειτουργίες που προσφέρει από μόνος του ο NS-2. Άλλωστε, υποστηρίζεται ήδη ένα μεγάλο πλήθος συνηθισμένων δικτυακών πρωτοκόλλων, όπως το Transmission Control Protocol (TCP), User Datagram Protocol (UDP), πρωτοκόλλων εφαρμογής, όπως το File Transfer Protocol και διάφορες προσομοιώσεις ουρών, όπως Drop Tail, Random Early Detection κλπ. Επίσης, υποστηρίζονται διάφοροι τύποι δικτύων, όπως ασύρματα, δορυφορικά, ενσύρματα, οπτικών ινών και τεχνολογίες δικτύου, όπως Ethernet, Token Ring, Wi-Fi κλπ.

Ωστόσο, η δημιουργία ενός σεναρίου προσομοίωσης απαιτεί την γνώση του τρόπου αναπαράστασης των συστατικών ενός δικτύου στην μορφή που κατανοεί ο NS-2, δηλαδή την OTcl. Ειδικά στην περίπτωση ενός πιο λεπτομερούς σεναρίου, χρειάζεται να καθοριστούν επιπλέον παράμετροι, πολλές από τις οποίες είναι αλληλένδετες. Πέραν αυτού, δεν υπάρχει τρόπος να αναπαρασταθεί γραφικά το δίκτυο πριν την εκτέλεση. Γι' αυτό το λόγο έχουν αναπτυχθεί μερικά εργαλεία τα οποία προσφέρουν αναπαράσταση της τοπολογίας του δικτύου.

Κατά τη διάρκεια της προσομοίωσης ο NS-2 καταγράφει όλα τα συμβάντα μεταξύ των συστατικών του δικτύου, καθώς και άλλες χρήσιμες πληροφορίες που παράγονται. Μετά την ολοκλήρωση της προσομοίωσης όλη αυτή η πληροφορία αποθηκεύεται κωδικοποιημένη με ειδική μορφή στα λεγόμενα trace files⁴. Ο NS-2 κατασκευάζει τρία διακριτά είδη trace files:

- Normal
- Old wireless
- New wireless

Η ανάλυση και κατανόηση του περιεχομένου των trace files είναι ουσιώδης για την χρήση των προσομοιωτών. Η δομή τους συνήθως δεν είναι ευνόητη και απαιτεί κάποιου είδους επεξεργασία πριν την εξαγωγή χρήσιμων συμπερασμάτων. Η επεξεργασία αυτή

δεν γίνεται αυτόματα από τον NS-2, αλλά έχουν δημιουργηθεί τρίτα εργαλεία για το σκοπό αυτό.

1.2. Αντικείμενο Διπλωματικής

Αφορμή για την εκπόνηση αυτής της διπλωματικής ήταν η έλλειψη ενός πλήρους εργαλείου που να καλύπτει τις ανάγκες τόσο μετά την προσομοίωση (post-simulation), με αποδοτική και επαρκή ανάλυση των trace files, όσο και πριν (pre-simulation), κατά την δημιουργία του σεναρίου. Παρ' ότι έχουν παρατηρηθεί αρκετά εργαλεία τα οποία αναλαμβάνουν να ικανοποιήσουν τις απαιτήσεις κάθε φάσης, δεν έχει υποπέσει στην αντίληψή μας κάποιο το οποίο να ενώνει όλες αυτές τις υπηρεσίες σε ένα πρόγραμμα.

Η ανάλυση των trace files και εξαγωγή των επιθυμητών μετρήσεων και συμπερασμάτων από αυτά είναι μια περίπλοκη διαδικασία, η οποία επαφίεται στον χρήστη. Όμως, ακόμα και μια μικρομεσαίας κλίμακας προσομοίωση παράγει μεγάλο μέγεθος trace files, δυσχεραίνοντας την ανάλυσή τους. Επιπλέον, τα trace files, όπως αναφέρθηκε και παραπάνω, είναι σε ειδική μορφή, αναλόγως του τύπου της προσομοίωσης που γίνεται. Κάθε ένα από τα είδη trace files αναφέρεται σε διαφορετικό τύπο δικτύου, ενώ πολλές φορές υπάρχουν υποπεριπτώσεις αυτών των ειδών με επιπρόσθετες παραλλαγές στην μορφή τους. Το γεγονός αυτό αυξάνει την δυσκολία της ανάλυσης, αφού ο χρήστης θα πρέπει να εξοικειωθεί με το είδος trace file που τον ενδιαφέρει κάθε φορά.

Ο αρχικός στόχος του TRAFIL⁵ ήταν να εστιάσει στην έλλειψη μιας απλής, εύχρηστης και αποδοτικής εφαρμογής, η οποία θα αναλαμβάνει την επεξεργασία και ανάλυση οποιουδήποτε τύπου trace file. Για το λόγο αυτό χρησιμοποιήθηκε η καινοτόμος ιδέα των metafiles, η οποία επεξηγείται περαιτέρω σε επόμενη ενότητα. Πέραν αυτού, η εφαρμογή αποθήκευε την πληροφορία που παρήγαγε σε τοπική βάση δεδομένων ώστε να είναι προσβάσιμη οποιαδήποτε στιγμή, ακόμα και αν το πρόγραμμα τερματιστεί ή ο χρήστης διαγράψει το trace file. Με βάση την παραγόμενη πληροφορία μπορούν να παραχθούν πολλά επιπλέον συμπεράσματα και απεικονίσεις, όπως γραφήματα αποδόσεων, στατιστικές, επιπλέον προσαρμοσμένες μετρήσεις κ.α. Επίσης, είναι δυνατόν να αποθηκευτούν τυχόν αλλαγές στα δεδομένα της βάσης δεδομένων.

Επιπλέον, με γνώμονα την αυτοματοποίηση όσο γίνεται περισσότερων διαδικασιών που αφορούν στις προσομοιώσεις με την χρήση του NS-2, το TRAFIL επεκτάθηκε ώστε να αυτοματοποιεί και την χρήση μιας πολύ δημοφιλούς επέκτασής του, του Evalvid-RA. Αναπτύχθηκε, επομένως, κατάλληλη πλήρης γραφική διεπαφή για την

χρήση του συγκεκριμένου framework, η οποία εξυπηρετεί πριν και μετά την προσομοίωση.

Στην ίδια κατεύθυνση, το TRAFIL πλέον αποτελεί και μια πλατφόρμα βοήθειας πριν την προσομοίωση (pre-simulation), έχοντας την δυνατότητα σχεδιασμού τοπολογιών δικτύου και την μετάφρασή τους σε μορφή σεναρίων κατανοητή στον NS-2. Όπως αναφέρθηκε νωρίτερα, η μορφή αυτή είναι η γλώσσα OTcl. Η τοπολογία του δικτύου, δηλαδή τα διάφορα συστατικά στοιχεία του, σχεδιάζεται γραφικά, ενώ οι λεπτομέρειες των συστατικών αυτών, καθορίζονται μέσα από μενού. Πέραν αυτού, ένα πολύ σημαντικό χαρακτηριστικό που προσφέρει αυτή η νέα λειτουργία του TRAFIL είναι η δυνατότητα αποθήκευσης ενός ημιτελούς σεναρίου και η φόρτωσή του αργότερα από το ίδιο το αρχείο του σεναρίου, χωρίς να χρησιμοποιείται κάποιο δευτερεύον βοηθητικό αρχείο. Όπως θα δούμε, αυτό γίνεται βάσει μιας ειδικής σήμανσης που ακολουθείται στην εκτύπωση του αρχείου του σεναρίου. Στόχος είναι η προτυποποίηση των αρχείων αυτών, ώστε να καταστεί εύκολη η επεξεργασία και γραφική αναπαράστασή τους, κατά κύριο λόγο μέσα από το TRAFIL.

1.3. Οργάνωση Κειμένου

Στην δεύτερη ενότητα αναφέρουμε σχετικές εργασίες στον τομέα του Pre-Simulation Assistance και του Post-Simulation Analysis, όσον αφορά τον NS-2. Περιγράφονται παρόμοια εργαλεία σε αυτούς τους τομείς, συγκρίνονται με το TRAFIL και τέλος καταδεικνύονται πιθανές ελλείψεις, αδυναμίες ή και πλεονεκτήματα που εμφανίζουν. Στην ενότητα τρία στοιχειοθετείται το θεωρητικό υπόβαθρο στο οποίο στηρίζεται το TRAFIL, περιγράφοντας αναλυτικά τη δομή ενός OTcl αρχείου, καθώς και την δομή ενός τυπικού σεναρίου προσομοίωσης δικτύου. Στην ίδια ενότητα αναλύεται και η μορφή των διαφόρων τύπων trace file που παράγει ο NS-2 και θεμελιώνεται η ιδέα των metafiles και submetafiles, την οποία εισάγει το TRAFIL για την επεξεργασία και ανάλυση των trace files.

Στην τέταρτη ενότητα παρουσιάζεται η αρχιτεκτονική του TRAFIL: αναφέρονται περιγραφικά οι διάφορες λειτουργικές μονάδες που το απαρτίζουν, η συσχέτισή τους και καθορίζονται τα τρία επίπεδα λειτουργιών στα οποία βασίζονται οι μονάδες αυτές.

Στην πέμπτη ενότητα αναλύονται οι κυριότερες κλάσεις του εργαλείου ως προς τον ρόλο τους στην προαναφερθείσα αρχιτεκτονική, την λειτουργία που επιτελούν και τις μεταξύ τους σχέσεις. Επιπλέον, αναλύονται οι σημαντικότερες διαδικασίες και δομές στις οποίες στηρίζεται, όπως η βάση δεδομένων. Τέλος, αναφέρονται οποιεσδήποτε

εξαρτήσεις έχει το TRAFIL από τρίτα εξωτερικά προγράμματα και αρχεία για να επιτελέσει κάποιες προχωρημένες λειτουργίες.

Στην έκτη ενότητα γίνεται παρουσίαση δοκιμαστικών εκτελέσεων και εργασιών του TRAFIL με όλες σχεδόν τις δυνατότητες που παρέχει, δίνονται οδηγίες χρήσης για την λειτουργία κάθε σεναρίου χρήσης του και εξηγούνται μερικές ειδικές περιπτώσεις μέσω παραδειγμάτων. Επίσης, δίνονται οδηγίες εγκατάστασης και προετοιμασίας του εργαλείου.

Στην έβδομη ενότητα γίνεται μια συνολική αξιολόγηση του TRAFIL όσον αφορά την επίτευξη των στόχων του. Επιπρόσθετα γίνεται και σύγκριση με παρόμοια εργαλεία, χρησιμοποιώντας μεταξύ άλλων και μετρικές επιδόσεων.

Τέλος, στην όγδοη ενότητα γίνεται μια προσπάθεια απόφασης συμπερασμάτων βάσει των στοιχείων που παρουσιάστηκαν, καταλήγοντας σε κάποιες πιθανές μελλοντικές επεκτάσεις και βελτιώσεις που μπορεί να αποδεχτεί το εργαλείο.

1.4. Δημοσιεύσεις

- Christos Bouras, Savvas Charalambides, Michalis Drakoulelis, Georgios Kioumourtzis, Kostas Stamos. “A tool for automating network simulation and processing tracing data files”. *Simulation Modelling Practice and Theory* 30 (2013), Elsevier. <<http://ru6.cti.gr/ru6/publications/tool-automatic-network-simulation-and-processing-tracing-data-files>>
- Christos Bouras, Savvas Charalambides, Michalis Drakoulelis, Georgios Kioumourtzis, Kostas Stamos. “Enhancing Simulation Environments with TRAFIL”. *Simulation Technologies in Networking and Communications: Selecting the Best Tool for the Test*, CRC Press. <<http://ru6.cti.gr/ru6/publications/enhancing-simulation-environment-trafil>>
- Christos Bouras, Savvas Charalambides, Michalis Drakoulelis, Georgios Kioumourtzis, Kostas Stamos. “Simulation Design and Execution: The case of TRAFIL”. 3rd International Conference on Simulation and Modeling Methodologies, Technologies and Applications, *Reykjavík, Iceland*. <<http://ru6.cti.gr/ru6/publications/simulation-design-and-execution-case-trafil>>

Προγενέστερες δημοσιεύσεις:

- Christos Bouras, Savvas Charalambides, Georgios Kioumourtzis, Kostas Stamos. “TRAFIL: A Tool for Enhancing Simulation TRAcE FILEs Processing”. International Conference on Data Communication Networking - DCNET 2012, *Rome, Italy*. <<http://ru6.cti.gr/ru6/publications/trafil-tool-enhancing-simulation-trace-files-processing>>

2. Σχετικές Εργασίες

Το TRAFIL είναι ένα εργαλείο που στοχεύει τελικά στην κάλυψη όλων των αναγκών ενός χρήστη του NS-2. Θεωρούμε ότι αυτές χωρίζονται διαδικαστικά σε τρεις θεματικές ενότητες. Η πρώτη είναι η σύνθεση πριν την προσομοίωση (pre-simulation composition), η βοήθεια κατά την εκτέλεση της προσομοίωσης (simulation assistance) και η ανάλυση μετά την προσομοίωση (post-simulation analysis). Θα αναφερθούμε σε γενικές γραμμές στις διαδικασίες που υποστηρίζονται από το TRAFIL σε κάθε θεματική ενότητα.

Η σύνθεση πριν την προσομοίωση αναφέρεται σε όλες τις διαδικασίες που μπορούν να γίνουν πριν εκτελεστεί μια προσομοίωση στον NS-2. Συνήθως αυτό σημαίνει την σύνθεση ενός σεναρίου προσομοίωσης σε κατάλληλη μορφή, απ' όπου και η ονομασία που δόθηκε. Δυο λειτουργίες του TRAFIL έχουν σκέλη σε αυτή την θεματική ενότητα. Η πρώτη είναι η σχεδίαση σεναρίου προσομοίωσης μέσω γραφικού περιβάλλοντος, και μετατροπή του σε OTcl script file. Εδώ το TRAFIL εστιάζει στην όσο το δυνατόν μεγαλύτερη κάλυψη επιλογών σχεδίασης που προσφέρει ο NS-2, παρέχοντας όμως και την δυνατότητα αποθήκευσης σεναρίου σε μορφή OTcl script file οποιαδήποτε στιγμή, ώστε να μπορεί να ανακτηθεί και να επεκταθεί πάλι μέσω του εργαλείου αργότερα. Η δεύτερη είναι η λειτουργία προσομοίωσης του Evalvid-RA. Όπως αναφέρθηκε παραπάνω, το Evalvid-RA είναι ένα δημοφιλές πλαίσιο εργασίας (framework) το οποίο προσθέτει στον NS-2 τη δυνατότητα μετάδοσης ενός βίντεο (streaming). Στο TRAFIL το σκέλος που αφορά στη συγκεκριμένη θεματική ενότητα χρησιμοποιείται μια αλυσίδα τρίτων εργαλείων για να προετοιμάσει το βίντεο για μετάδοση, ενώ δέχεται μέσω γραφικού περιβάλλοντος τις υπόλοιπες παραμέτρους που χρειάζεται το framework για να λειτουργήσει.

Αντίστοιχα, η βοήθεια κατά την εκτέλεση της προσομοίωσης αφορά στις διαδικασίες που αυτοματοποιούν την εκτέλεση μιας προσομοίωσης και την ανάγνωση της εξόδου της. Συγκεκριμένα, το TRAFIL μπορεί να κληθεί να εκτελέσει μια προσομοίωση μέσω του NS-2 και να διαβάσει τα αποτελέσματα σε 3 περιπτώσεις: είτε μέσω του σχεδιασμού σεναρίου, είτε με την χρήση της μονάδας Evalvid-RA, είτε απευθείας με ανάθεση έτοιμου αρχείου σεναρίου. Και στις τρεις περιπτώσεις τα αποτελέσματα εμφανίζονται σε αντίστοιχη οθόνη, ενώ φορτώνονται και στην βάση δεδομένων για περαιτέρω ανάλυση. Είναι απαραίτητο να σημειωθεί ότι η χρήση του TRAFIL για αυτές τις δραστηριότητες υλοκρύπτει αρκετές διαδικασίες και λεπτομέρειες που θα έπρεπε σε

άλλη περίπτωση να κληθούν χειροκίνητα, συνήθως μέσω κονσόλας ή τερματικού, ενώ ο χρήστης θα έπρεπε να γνωρίζει εκ των προτέρων τις κατάλληλες εντολές.

Τέλος, η ανάλυση μετά την προσομοίωση εστιάζει στην ανάλυση και επεξεργασία των αποτελεσμάτων μιας προσομοίωσης, είτε μέσω έτοιμου trace file που καθορίζει ο χρήστης, είτε έπειτα από μια προσομοίωση που εκτελέστηκε από το ίδιο το TRAFIL. Σε κάθε περίπτωση σημαντικότερος στόχος είναι η συνολική διαδικασία χειρισμού ενός trace file, από την αρχική ανάλυση μέχρι και την παραγωγή προσαρμοσμένων μετρήσεων, να είναι αποδοτική, ευέλικτη, ακριβής και πλήρης.

2.1. *Εκτέλεση προσομοιώσεων*

Ένα αποκύημα των μεγάλων δυνατοτήτων επέκτασης του NS-2 αλλά και της μεγάλης ηλικίας του είναι η υψηλή πολυπλοκότητα που παρουσιάζει κατά την εγκατάσταση είτε αυτού είτε επιπρόσθετων μονάδων του. Στην θεματική ενότητα της εκτέλεσης προσομοιώσεων σε NS-2 έχουν γίνει κάποιες προσπάθειες απλοποίησης και αυτοματοποίησης των διαδικασιών εγκατάστασης του προσομοιωτή και των διαφόρων επεκτάσεών του. Συνήθως ο προσομοιωτής “πακετάρεται” με τις επεκτάσεις του ή τα τρίτα εργαλεία που απαιτούνται για ένα συγκεκριμένο είδος προσομοιώσεων.

Ένα ολοκληρωμένο παράδειγμα είναι το Liowsn Project^{6,7}, το οποίο αποτελεί μια παραλλαγμένη διανομή των Ubuntu 9.10 Linux με προεγκατεστημένο τον NS-2.34 και κάποιες επεκτάσεις του, κυρίως απαραίτητες στον τομέα των προσομοιώσεων δικτύων αισθητήρων (Wireless Sensor Networks - WSN). Συγκεκριμένα, περιέχει το Mannasim, το OMNeT++, το Castalia, το NAM και μερικά εξωτερικά εργαλεία, όπως το Xgraph και το Tracegraph. Το Mannasim είναι υπεύθυνο για την προσομοίωση δικτύων αισθητήρων, το Xgraph για την παραγωγή γραφημάτων των αποτελεσμάτων, και το Tracegraph για την ανάλυση trace files. Το NAM είναι ένα σημαντικό συνοδευτικό εργαλείο του NS-2 το οποίο απεικονίζει με κινούμενα γραφικά την προσομοίωση δικτύου, χρησιμοποιώντας ένα ειδικό επιπλέον trace file που παράγει ο προσομοιωτής, το οποίο ονομάζεται nam trace file. Το Liowsn Project αποτελεί μια καλή λύση για γρήγορη εγκατάσταση του NS-2 και την εκτέλεση των προσομοιώσεων, ωστόσο εστιάζει κυρίως στην προσομοίωση δικτύων αισθητήρων, ενώ δεν προσφέρει κάποιο αυτοματισμό στην ίδια την διαδικασία προσομοίωσης μέσω NS-2. Τέλος, να σημειωθεί ότι το εργαλείο αυτό έχει τη μορφή λειτουργικού συστήματος, κάτι που σημαίνει ότι ο χρήστης θα χρειαστεί να το εγκαταστήσει ξεχωριστά από όποιο λειτουργικό σύστημα έχει και χρησιμοποιεί για τις υπόλοιπες εργασίες του. Επίσης, η διανομή είναι πλέον αρκετά απαρχαιωμένη ώστε υπάρχουν κάποια προβλήματα συμβατότητας με νεότερο υλικό (hardware). Όπως θα

δειχθεί παρακάτω, το TRAFIL δεν απαιτεί ιδιαίτερη εγκατάσταση, ενώ προσφέρει πολύ ανώτερους αυτοματισμούς στην διαδικασία προσομοίωσης καθ'αυτή.

Ένα άλλο “πακέτο” του NS-2 είναι το PiccSIM⁸, το οποίο δημιουργήθηκε το 2009 στο Aalto University, και παραμένει ενεργό. Το συγκεκριμένο πακέτο χρησιμοποιείται για την προσομοίωση ασύρματου ελέγχου συστημάτων (Wireless Control Systems - WCS), χρησιμοποιώντας το λογισμικό Simulink⁹ αλλά και τον NS-2. Αργότερα δημιουργήθηκε το PiccSIM Toolchain¹⁰, μια γραφική διεπαφή χρήστη (Graphical User Interface - GUI) για την δημιουργία σεναρίων και τον χειρισμό του εργαλείου μέσα από αυτή, ενώ πρόσφατα προστέθηκε ένα ακόμα εργαλείο στο πακέτο, το TrueTime¹¹. Το πακέτο έχει την δυνατότητα μιας απλής ανάλυσης των παραγόμενων trace files παράγοντας κάποιες στατιστικές, όμως επαφίεται στον χρήστη η περαιτέρω ανάλυσή τους.

Πέρα από τον NS-2, στον τομέα των προσομοιώσεων δικτύων υπάρχουν και άλλες επιλογές. Μια αρκετά πλήρης από αυτές είναι το J-Sim^{12,13}, το οποίο είναι ένα περιβάλλον προσομοιώσεων ανοιχτού λογισμικού, υλοποιημένο στην γλώσσα προγραμματισμού Java. Βασίζεται στην ιδέα της Αρχιτεκτονικής Αυτόνομου Συστατικού (Autonomous Component Architecture - ACA), η οποία εκφράζει την δυνατότητα κάθε συστατικού του δικτύου να σχεδιάζεται, να υλοποιείται και να δοκιμάζεται ανεξάρτητα. Το J-Sim υποστηρίζει επιπλέον μερικές scripting γλώσσες για τον σχεδιασμό δικτύων, όπως η Tcl, η Python και η Perl. Οι γλώσσες αυτές συνδυάζονται με την Java για να συνδέσουν τα διάφορα συστατικά και να τα μεταφράσουν σε στοιχεία της προσομοίωσης, με τον ίδιο περίπου τρόπο που η OTcl λειτουργεί με την C++ στον NS-2. Το J-Sim διαθέτει επίσης μια γραφική διεπαφή σχεδίασης, το gEditor¹⁴, το οποίο αναλαμβάνει να συνδέσει τα αντικείμενα και τις παραμέτρους τους που έχουν σχεδιαστεί με τα συστατικά Java που καταλαβαίνει το κυρίως εργαλείο. Αυτό το χαρακτηριστικό είναι πολύ χρήσιμο, και θα δούμε ότι το TRAFIL διαθέτει πλέον αυτή την δυνατότητα μέσω της νέας μονάδας σχεδιασμού. Εν κατακλείδι, το J-Sim είναι ένα αρκετά χρήσιμο εργαλείο, παρατηρείται όμως ότι η υποστήριξή του έχει σταματήσει, ενώ οι συνεισφορές της κοινότητας για τον NS-2 τον έχουν εξελίξει πολύ περισσότερο.

2.2. *Σύνθεση πριν την προσομοίωση*

Στον τομέα της σύνθεσης σεναρίων δικτύου έχουν αναπτυχθεί αρκετά εργαλεία τοπολογιών. Τα εργαλεία αυτά συνήθως παρέχουν μια γραφική διεπαφή στην οποία μπορεί κάποιος να σχεδιάσει το δίκτυο και να υποβάλλει κάποιες παραμέτρους. Αναφέραμε ήδη το gEditor, την γραφική διεπαφή (front-end) του J-Sim, και το PiccSIM

Toolchain, τα οποία δίνουν τέτοιες δυνατότητες. Θα καταγραφούν παρακάτω μερικά επιπλέον εργαλεία τα οποία λειτουργούν αντίστοιχα.

Το NSG-2¹⁵ είναι ένα εργαλείο που χρησιμοποιείται αποκλειστικά για την παραγωγή σεναρίων από γραφική αναπαράσταση σε μορφή OTcl. Παρέχει αρκετές επιλογές στον χρήστη σε ένα αρκετά απλοποιημένο περιβάλλον. Είναι υλοποιημένο σε Java, επομένως είναι δυνατόν να εκτελεστεί σε οποιαδήποτε λειτουργικό σύστημα υποστηρίζει η Java. Το NSG-2 δίνει την δυνατότητα σχεδιασμού είτε ενσύρματου (wired) είτε ασύρματου (wireless) δικτύου. Η γραφική διεπαφή χρήστη του δίνει επίσης την δυνατότητα εμφάνισης πλέγματος και εστίασης (zoom). Προς αυτή την κατεύθυνση έχει κινηθεί και η υλοποίηση του TRAFIL, με κάποιες διαφοροποιήσεις.

Το Liowsn Project μεταξύ άλλων περιλαμβάνει και το MannaSim^{16,17} framework, μέρος του οποίου είναι και το Script Generator Tool (SGT), ένα εργαλείο σχεδιασμού TCL script file για NS-2. Ο σχεδιασμός, όμως, δεν είναι καθαρά γραφικός, παρά μόνο μέσω καθορισμού των συστατικών του δικτύου και των παραμέτρων μέσω μενού. Το εργαλείο έχει επίσης την δυνατότητα αποθήκευσης των παραμέτρων σεναρίου σε μορφή XML ώστε να μπορεί να δημιουργήσει αργότερα ξανά το TCL script file.

2.3. *Ανάλυση μετά την προσομοίωση*

Τα εργαλεία που έχουν αναπτυχθεί στον τομέα της ανάλυσης των παραγόμενων αποτελεσμάτων παράγουν αναλύσεις, στατιστικά στοιχεία, μετρήσεις και γραφικές παραστάσεις βασιζόμενα στα trace files και μπορούν να χωριστούν σε δυο κατηγορίες. Η μια κατηγορία αποτελείται από αυτά που ενσωματώνουν τον NS-2, όπως το Network Simulation Cradle^{18,19} και το NS2Measure²⁰. Τα εργαλεία της κατηγορίας αυτής προσφέρουν συνήθως εύκολη εγκατάσταση και υποστήριξη του NS-2 με κάποιες επεκτάσεις. Από την άλλη δεν είναι πάντα εύκολη η αλλαγή έκδοσης του προσομοιωτή ή η προσθήκη επιπλέον επεκτάσεων. Η δεύτερη κατηγορία περιλαμβάνει τα εργαλεία που δέχονται ως είσοδο απλά trace files, χωρίς να απαιτούν την ύπαρξη προσομοιωτή. Προγράμματα αυτής της κατηγορίας είναι το JTrana, το Trace Graph και το NS2 Trace Analyzer, ενώ και το ίδιο το TRAFIL ανήκει σε αυτή την κατηγορία.

Το JTrana²¹ είναι υλοποιημένο σε γλώσσα προγραμματισμού Java, επομένως μπορεί να εκτελεστεί ανεξαρτήτως πλατφόρμας. Υποστηρίζει την ανάλυση όλων των τύπων trace file από τα οποία παράγει γενικές πληροφορίες για την προσομοίωση αλλά και επιπλέον μετρήσεις ποιότητας υπηρεσιών (QoS) ενώ μπορεί να παράγει μια μεγάλη ποικιλία γραφικών παραστάσεων. Χρησιμοποιεί γραφική διεπαφή χρήστη για την εμφάνιση των αποτελεσμάτων και τον χειρισμό του. Επιπρόσθετα, αν και χρησιμοποιεί

βάση δεδομένων για την προσωρινή αποθήκευση του trace file που αναλύει, δεν το αποθηκεύει για επόμενη χρήση.

Το Tracegraph^{22,23} είναι εργαλείο για ανάλυση trace files υλοποιημένο σε Matlab 6.0, αλλά μπορεί να μεταγλωττιστεί ώστε να εκτελείται και αυτόνομα. Χρησιμοποιεί και αυτό γραφική διεπαφή χρήστη, υποστηρίζει όλα τα trace files του NS-2, και πέραν των τυπικών πληροφοριών και μετρήσεων μπορεί να παράγει δισδιάστατες αλλά και τρισδιάστατες γραφικές παραστάσεις για μετρήσεις όπως η ρυθμαπόδοση (throughput), ρυθμαπόδοση και καθυστέρηση (delay), jitter κλπ.

Το NS-2 Trace Analyzer²⁴ είναι εργαλείο χωρίς γραφική διεπαφή, επομένως ο χειρισμός του είναι αποκλειστικά μέσω γραμμής εντολών. Είναι υλοποιημένο σε C/C++ και μπορεί να εκτελεστεί σε όλες τις Unix πλατφόρμες, αλλά και σε Windows με την χρήση Cygwin. Όπως και τα προηγούμενα εργαλεία, μπορεί να αναλύσει όλα τα trace files και να εξάγει μετρήσεις από αυτά, αλλά δεν μπορεί να παράγει γραφικές παραστάσεις.

Τέλος, ένα σχετικά καινούργιο εργαλείο είναι το NS2 Visual Trace Analyzer²⁵, το οποίο είναι υλοποιημένο σε Java. Παρέχει ανάλυση trace file, μετρήσεις και γραφικές παραστάσεις, ενώ μπορεί επίσης να απεικονίσει και τη μορφή του δικτύου. Για τη λειτουργία αυτή απαιτεί σαν είσοδο το αρχικό αρχείο σεναρίου σε γλώσσα Tcl.

Όλα τα παραπάνω εργαλεία προσφέρουν είτε την δυνατότητα ανάλυσης trace files, είτε την δυνατότητα σχεδιασμού και εκτέλεσης κάποιων συγκεκριμένων προσομοιώσεων. Παρατηρούμε ότι κανένα από αυτά δεν συνδυάζει και τα δυο. Επίσης, παρατηρούμε ότι τα εργαλεία ανάλυσης δεν προσφέρουν αποθήκευση των αποτελεσμάτων τους σε κάποια βάση δεδομένων, με αποτέλεσμα ο χρήστης να πρέπει να επαναφορτώσει το trace file κάθε φορά που το χρειάζεται. Όμως, η διαδικασία αυτή, όπως προαναφέρθηκε και θα παρατηρηθεί στη συνέχεια, είναι επίπονη και χρονοβόρα για τυπικές προσομοιώσεις. Επιπλέον, ο μόνος τρόπος που προτείνεται για αποθήκευση των αποτελεσμάτων είναι μέσω αρχείων κειμένου, τα οποία επαφίενται στον χρήστη να διαχειριστεί. Ως αποτέλεσμα των παραπάνω, η χρησιμότητα των εργαλείων αυτών μειώνεται.

Το κενό σε όλα αυτά τα ζητήματα έρχεται να καλύψει το TRAFIL. Με λίγα λόγια, ο στόχος του είναι η εκτέλεση μιας πλήρους προσομοίωσης από τον αρχικό σχεδιασμό του, την εκτέλεση καθεαυτή και τέλος την ανάλυση, αποθήκευση και σύγκριση των αποτελεσμάτων μεταξύ τους.

3. Θεωρητικό Υπόβαθρο

Σε αυτή την ενότητα θα γίνει περιγραφή των μοντέλων, τεχνικών και άλλων υποκείμενων θεωρητικών λεπτομερειών πάνω στις οποίες στηρίζεται ο σχεδιασμός και η λειτουργία του εργαλείου.

3.1. Η δομή ενός trace file

Η βασικότερη λειτουργία του TRAFIL είναι η ανάλυση των αποτελεσμάτων μιας προσομοίωσης. Υποστηρίζει την ανάλυση trace file προερχόμενων από οποιαδήποτε έκδοση του NS-2, αλλά έχει την δυνατότητα επέκτασης ώστε να υποστηρίζει και επιπλέον αρχεία.

Τα τρία βασικά είδη trace file που παράγει ο NS-2, αναλόγως του δικτύου που προσομοιώθηκε, είναι τα εξής:

- Normal trace file
- Old wireless trace file
- New wireless trace file

Κάθε βασικό είδος αποτελείται από κάποια εξειδικευμένα υποείδη τα οποία περιέχουν επιπλέον στήλες πληροφορίας. Πολλές φορές χρησιμοποιούνται αυτές οι υποπεριπτώσεις ώστε να περιγραφεί καλύτερα το δίκτυο της προσομοίωσης. Επομένως χρειάζεται να αναγνωριστούν μεταξύ τους και να αναλυθούν κατάλληλα.

Κάθε γραμμή ενός trace file περιγράφει ένα γεγονός (event). Το πρώτο σύμβολο της γραμμής δείχνει το γενικό είδος του γεγονότος, το οποίο μπορεί να είναι αποστολή, λήψη, απόρριψη, ενώ ανάλογα με το είδος του δικτύου μπορεί να υπάρχουν και διάφορα άλλα, όπως σφάλμα, προώθηση κλπ. Οι επόμενες στήλες σημαίνουν διαφορετικά πράγματα για κάθε είδος trace file. Παρατίθενται οι δομές καθενός όπως ορίζονται από το εγχειρίδιο του NS-2²⁶, ώστε να μπορεί να γίνει άμεση σύγκριση των διαφορών τους. Οι υποπεριπτώσεις ενός είδους ακολουθούν αμέσως μετά το βασικό είδος.

Event	Abbreviation	Type	Value
Normal Event	r: Receive d: Drop e: Error	%g %d %d %s %d %s %d %d. %d %d. %d %d %d	

	+ : Enqueue -: Dequeue		
		double	Time
		int	(Link-layer) Source Node
		int	(Link-layer) Destination Node
		string	Packet Name
		int	Packet Size
		string	Flags
		int	Flow ID
		int	(Network-layer) Source Address
		int	Source Port
		int	(Network-layer) Destination Address
		int	Destination Port
		int	Sequence Number
		int	Unique Packet ID
TCP Trace		%d 0x%x %d %d	
		int	Ack number
		hexadecimal	Flags (Used by FullTCP) FIN=0x01, SYN=02, PUSH=08, ACK=10, ECE=40, CWR=80
		int	Header Length
		int	Socket Address Length
Satellite Trace		%.2f %.2f %.2f %.2f	
		double	Source Latitude
		double	Source Longitude
		double	Destination Latitude

		double	Destination Longitude
Old Wireless Event	s: Send r: Receive d: Drop f: Forward	%.9f %d (%6.2f %6.2f) %3s %4s %d %s %d [%x %x %x %x]	
		%.9f _%d_ %3s %4s %d %s %d [%x %x %x %x]	
		double	Time
		int	Node ID
		double	X Coordinate (If logging position)
		double	Y Coordinate (If logging position)
		string	Trace Name
		string	Reason
		int	Event Identifier
		string	Packet Type
		int	Packet Size
		hexadecimal	Time To Send Data
		hexadecimal	Destination MAC Address
		hexadecimal	Source MAC Address
		hexadecimal	Type (ARP, IP)
ARP Trace		----- [%s %d/%d %d/%d]	
		string	Request or Reply
		int	Source MAC Address
		int	Source Address
		int	Destination MAC Address

		int	Destination Address
DSR Trace		%d [%d %d] [%d %d %d %d->%d] [%d %d %d %d->%d]	
		int	Number of Nodes Traversed
		int	Routing Request Flag
		int	Route Request Sequence Number
		int	Routing Reply Flag
		int	Route Request Sequence Number
		int	Reply Length
		int	Source of Source Routing
		int	Destination of Source Routing
		int	Error Report Flag
		int	Number of Errors
		int	Report to Whom
		int	Link Error From
		int	Link Error To
AODV Trace		[0x%x %d %d [%d %d] [%d %d]] (REQUEST)	
		hexadecimal	Type
		int	Hop Count
		int	Broadcast ID
		int	Destination
		int	Destination Sequence Number
		int	Source

		int	Source Sequence Number
		[0x%x %d [%d %d] %f] (%s)	
		hexadecimal	Type
		int	Hope Count
		int	Destination
		int	Destination Sequence Number
		double	Lifetime
		string	Operation (REPLY, ERROR, HELLO)
TORA Trace		[0x%x %d] (QUERY)	
		hexadecimal	Type
		int	Destination
		0x%x %d (%f %d %d %d %d) (UPDATE)	
		hexadecimal	Type
		int	Destination
		double	Tau
		int	Oid
		int	R
		int	Delta
		int	ID
		[0x%x %d %f %d] (CLEAR)	
		hexadecimal	Type
		int	Destination

		double	Tau
		int	Oid
IP Trace		----- [%d:%d %d:%d %d %d]	
		int	Source IP Address
		int	Source Port Number
		int	Destination IP Address
		int	Destination Port Number
		int	TTL Value
		int	Next Hop Address, if any
TCP Trace		[%d %d] %d %d	
		int	Sequence Number
		int	Acknowledgement Number
		int	Number of Times Packet was forwarded
		int	Optimal Number of Forwards
CBR Trace		[%d] %d %d	
		int	Sequence Number
		int	Number of Times Packet was forwarded
		int	Optimal Number of Forwards
IMEP Trace		[%c %c %c 0x%04x]	
		char	Acknowledgement Flag
		char	Hello Flag
		char	Object Flag
		hexadecimal	Length

RCA Trace (from MIT Leach code)		----- [%c %d %d %d]	
		char	Operation (A, R, D)
		int	RCA Source
		int	RCA Link Destination
		int	RCA MAC Destination
New Wireless Event	s: Send r: Receive d: Drop f: Forward	double	Time (* For Global Setting)
		int	Node ID
		double	Node X Coordinate
		double	Node Y Coordinate
		double	Node Z Coordinate
		double	Node Energy Level
		string	Network trace level (AGR, RTR, MAC, etc)
		string	Drop Reason
		int	Hop source node ID
		int	Hope destination Node ID, -1, -2
		hexadecimal	Duration
		hexadecimal	Source Ethernet Address
		hexadecimal	Destination Ethernet Address
		hexadecimal	Ethernet Type
		string	Packet Type (arp, dsr, imep, tora, etc)
		string	Packet Type (cbr, tcp)

Οι επιπλέον στήλες που μπορεί να έχει ένα New Wireless trace file είναι περίπου ίδιες με αυτές του Old Wireless. Παρακάτω βλέπουμε μερικά δείγματα trace files:

```
1 + 1 0 2 tcp 40 ----- 0 0.0 3.0 0 0
2 - 1 0 2 tcp 40 ----- 0 0.0 3.0 0 0
3 + 1 1 2 tcp 40 ----- 0 1.0 3.1 0 1
4 - 1 1 2 tcp 40 ----- 0 1.0 3.1 0 1
5 r 1.03016 0 2 tcp 40 ----- 0 0.0 3.0 0 0
6 + 1.03016 2 3 tcp 40 ----- 0 0.0 3.0 0 0
7 - 1.03016 2 3 tcp 40 ----- 0 0.0 3.0 0 0
8 r 1.04032 1 2 tcp 40 ----- 0 1.0 3.1 0 1
9 + 1.04032 2 3 tcp 40 ----- 0 1.0 3.1 0 1
10 - 1.04032 2 3 tcp 40 ----- 0 1.0 3.1 0 1
11 r 1.0808 2 3 tcp 40 ----- 0 0.0 3.0 0 0
12 + 1.0808 3 2 ack 40 ----- 0 3.0 0.0 0 2
13 - 1.0808 3 2 ack 40 ----- 0 3.0 0.0 0 2
14 r 1.09096 2 3 tcp 40 ----- 0 1.0 3.1 0 1
15 + 1.09096 3 2 ack 40 ----- 0 3.1 1.0 0 3
16 - 1.09096 3 2 ack 40 ----- 0 3.1 1.0 0 3
17 r 1.13144 3 2 ack 40 ----- 0 3.0 0.0 0 2
```

Εικόνα 1 Παράδειγμα ενός normal trace file

```

1 + 0.005032 2 3 cbr 40 ----- 1 0.0 3.0 0 0 2 0x23 3 4
2 - 0.005032 2 3 cbr 40 ----- 1 0.0 3.0 0 0 2 0x23 3 4
3 + 0.0058 2 3 cbr 1000 ----- 2 1.0 3.1 0 1 2 0x23 3 4
4 - 0.0058 2 3 cbr 1000 ----- 2 1.0 3.1 0 1 2 0x23 3 4
5 + 0.005032 2 3 cbr 40 ----- 1 0.0 3.0 0 0
6 - 0.005032 2 3 cbr 40 ----- 1 0.0 3.0 0 0
7 + 0.0058 2 3 cbr 1000 ----- 2 1.0 3.1 0 1
8 - 0.0058 2 3 cbr 1000 ----- 2 1.0 3.1 0 1
9 + 0.007032 2 3 cbr 40 ----- 1 0.0 3.0 1 2
10 + 0.009032 2 3 cbr 40 ----- 1 0.0 3.0 2 3
11 r 0.010245 2 3 cbr 40 ----- 1 0.0 3.0 0 0
12 + 0.0108 2 3 cbr 1000 ----- 2 1.0 3.1 1 4
13 + 0.011032 2 3 cbr 40 ----- 1 0.0 3.0 3 5
14 - 0.011133 2 3 cbr 1000 ----- 2 1.0 3.1 1 4
15 + 0.013032 2 3 cbr 40 ----- 1 0.0 3.0 4 6
16 + 0.015032 2 3 cbr 40 ----- 1 0.0 3.0 5 8
17 + 0.0158 2 3 cbr 1000 ----- 2 1.0 3.1 2 7

```

Εικόνα 2 Normal trace file με επιπλέον TCP κεφαλίδα

```

s 20.000000000 _0_ AGT --- 6 cbr 512 [0 0 0 0] ----- [0:0 1:0 32 0] [0] 0 0
r 20.000000000 _0_ RTR --- 6 cbr 512 [0 0 0 0] ----- [0:0 1:0 32 0] [0] 0 0
s 20.000000000 _0_ RTR --- 6 cbr 532 [0 0 0 0] ----- [0:0 1:0 32 1] [0] 0 0
s 20.000275000 _0_ MAC --- 6 cbr 584 [13a 1 0 800] ----- [0:0 1:0 32 1] [0] 0 0
r 20.004947063 _1_ MAC --- 6 cbr 532 [13a 1 0 800] ----- [0:0 1:0 32 1] [0] 1 0
s 20.004957063 _1_ MAC --- 0 ACK 38 [0 0 0 0]
r 20.004972063 _1_ AGT --- 6 cbr 532 [13a 1 0 800] ----- [0:0 1:0 32 1] [0] 1 0
r 20.005261125 _0_ MAC --- 0 ACK 38 [0 0 0 0]

```

Εικόνα 3 Παράδειγμα New Wireless trace file

Παρατηρούμε ότι αν και υπάρχει μια ρητή δομή σε κάθε περίπτωση, είναι δυνατόν κάποια πεδία να λείπουν από κάποιο trace file. Αυτό δυσκολεύει ακόμα περισσότερο την αναγνώριση και προτυποποίηση ενός trace file.

3.2. *Metafiles: Η ιδέα του TRAFIL*

Μια αρχική ιδέα και συνηθής τρόπος αντιμετώπισης του προβλήματος από παρόμοιες λύσεις είναι η κωδικοποίηση των trace files μέσα στον κώδικα του προγράμματος. Κάτι τέτοιο όμως καθιστά το πρόγραμμα μη ευέλικτο, αφού δεν είναι εύκολη η προσθήκη νέων προτύπων.

3.2.1.

Metafiles

Γι' αυτό το TRAFIL εισήγαγε την ιδέα των metafiles. Πρόκειται για βοηθητικά αρχεία τα οποία περιγράφουν τη δομή ενός trace file εστιάζοντας στην χρήσιμη πληροφορία που παρέχουν. Τα metafiles έχουν χρησιμοποιηθεί και σε άλλες εφαρμογές προκειμένου να προσθέσουν ένα επίπεδο αφαίρεσης στην είσοδο του προγράμματος. Στην περίπτωση των trace files ως εισόδου προς ανάλυση στο TRAFIL διαπιστώνεται ότι πρώτο μέλημα είναι η ταυτοποίησή τους με κάποιο γνωστό είδος trace file, ώστε να μπορεί να επέλθει η ανάλυση και επεξεργασία τους. Για την ταυτοποίηση είναι απαραίτητο να ξέρουμε την δομή κάθε είδους, ώστε να γίνει σύγκριση μεταξύ τους. Αυτό το ρόλο αναλαμβάνουν τα metafiles, τα οποία κωδικοποιούν την δομή κάθε βασικού είδους trace file που παράγει ο NS-2. Περιέχουν πληροφορίες για κάθε πεδίο, όπως αυτά αναφέρονται στον παραπάνω πίνακα. Συγκεκριμένα για τα trace files του NS-2 υπάρχουν τρία metafiles τα οποία φαίνονται παρακάτω:

```
NumberOfFields 14
NumberOfColumns 12
UniqueCounter 1
-name event -type char(1) -index 0 -unique +
-name time -type double -index 1
-name SourceNode -type int -index 2
-name DestinationNode -type int -index 3
-name PacketName -type varchar(20) -index 4
-name PacketSize -type int -index 5
-name Flags -type varchar(7) -index 6
-name FlowID -type int -index 7
-name SourceAddress -type int -index 8 -delimiter .
-name SourcePort -type int -index 8
-name DestinationAddress -type int -index 9 -delimiter .
-name DestinationPort -type int -index 9
-name SequenceNumber -type int -index 10
-name UniquePacketID -type int -index 11
-TimeRelated -column time
-NodeRelated -column SourceNode -column DestinationNode
-PacketSize -column PacketSize
```

-SendingNodes -column SourceNode -column SourceAddress
-GeneratedPackets -column SourceNode -column SourceAddress -column UniquePacketID
-ReceivedPackets -column DestinationNode -column DestinationAddress -column UniquePacketID
-ForwardedPackets -column SourceNode -column SourceAddress -column DestinationAddress -column UniquePacketID
-SentPackets -column SourceNode -column SourceAddress -column UniquePacketID
-DroppedPackets -column UniquePacketID

Εικόνα 4 Normal metafile

NumberOfFields 12
NumberOfColumns 12
UniqueCounter 2
-name event -type char(1) -index 0
-name time -type double -index 1
-name NodeID -type integer -index 2 -startsWith _ -endsWith _
-name TraceName -type varchar(20) -index 3
-name Reason -type varchar(20) -index 4
-name EventIdentifier -type int -index 5
-name PacketType -type varchar(20) -index 6
-name PacketSize -type int -index 7
-name TimeToSendData -type varchar(20) -index 8 -startsWith [-unique [
-name DestinationMACAddress -type varchar(20) -index 9
-name SourceMACAddress -type varchar(20) -index 10
-name TypeArpIp -type varchar(20) -index 11 -endsWith] -unique]
-TimeRelated -column time
-NodeRelated -column NodeID
-PacketSize -column PacketSize
-SendingNodes -column EventIdentifier -column NodeID -column SourceIpAddress
-GeneratedPackets -column EventIdentifier -column NodeID -column SourceIpAddress
-ReceivedPackets -column EventIdentifier -column DestinationIpAddress -column NodeID
-ForwardedPackets -column EventIdentifier -column DestinationIpAddress -column NodeID

Εικόνα 5 Old Wireless metafile

NumberOfFields 25

NumberOfColumns 45

UniqueCounter 0

-name event -type char(1) -index 0

-name time -type double -index 2

-name HopSourceNodeID -type int -index 4

-name HopDestNodeID -type int -index 6

-name NodeID -type int -index 8

-name XCoordinate -type double -index 10

-name YCoordinate -type double -index 12

-name ZCoordinate -type double -index 14

-name NodeEnergyLevel -type double -index 16

-name NetworkTraceLevel -type varchar(6) -index 18

-name DropReason -type varchar(8) -index 20

-name Duration -type varchar(20) -index 22

-name DestinationEthernetAddress -type varchar(20) -index 24

-name SourceEthernetAddress -type varchar(20) -index 26

-name EthernetType -type varchar(20) -index 28 -startsWith ox

-name SourceAddress -type int -index 30 -delimiter .

-name SourcePort -type int -index 30

-name DestinationAddress -type int -index 32 -delimiter .

-name DestinationPort -type int -index 32

-name PacketTypeIP -type varchar(20) -index 34

-name PacketSize -type int -index 36

-name FlowID -type int -index 38

-name UniqueID -type int -index 40

-name TTLValue -type int -index 42

-name PacketType -type varchar(20) -index 44

-TimeRelated -column time

-NodeRelated -column NodeID -column NetworkTraceLevel

-PacketSize -column PacketSize

-SendingNodes -column UniqueID -column NodeID -column SourceAddress

-GeneratedPackets -column UniqueID -column NodeID -column SourceAddress

-ReceivedPackets -column UniqueID -column DestinationAddress -column NodeID
-ForwardedPackets -column UniqueID -column DestinationAddress -column NodeID

Εικόνα 6 New Wireless metafile

Παρατηρείται ότι και η δομή των metafiles είναι προκαθορισμένη. Αναλυτικότερα, τα τρία πρώτα πεδία είναι απαραίτητα σε κάθε metafile και χρησιμοποιούνται πέραν της ταυτοποίησης του trace file και για την επικύρωση του metafile. Τα πεδία αυτά είναι:

- *NumberOfFields*: Ο αριθμός αυτός δηλώνει το πλήθος των ξεχωριστών πεδίων που πρέπει να εξαχθούν από κάθε γραμμή του trace file εάν αυτό αναλυθεί με το παρόν metafile. Δεν είναι απαραίτητο κάθε γραμμή να περιέχει αυτό τον αριθμό πεδίων, αλλά είναι πιθανόν τα πεδία αυτά να εξάγονται μετά από επεξεργασία της γραμμής ώστε να διασπαστούν τυχόν ομαδοποιημένα πεδία.
- *NumberOfColumns*: Δηλώνει τον αριθμό των ξεχωριστών πεδίων που υπάρχουν σε κάθε γραμμή του trace file. Με την έννοια ξεχωριστών πεδίων εννοούνται τα πεδία που χωρίζονται με τον κενό χαρακτήρα μεταξύ τους.
- *UniqueCounter*: Δηλώνει τον αριθμό των χαρακτήρων που πρέπει να ψάξει το TRAFIL σε κάθε γραμμή του trace file ώστε να το ταιριάξει με το παρόν metafile.

Οι επόμενες γραμμές περιγράφουν τα πεδία που αντλούνται από το trace file μέσω σημαίων (flags). Κάθε γραμμή απαιτείται να περιέχει τουλάχιστον τις σημαίες -name, -type και -index, αφού χρησιμοποιούνται για την κατασκευή πίνακα στη βάση δεδομένων του TRAFIL και την ονοματοδοσία της στήλης στον πίνακα στατιστικών που εμφανίζονται στον χρήστη. Οι σημαίες που μπορεί να χρησιμοποιηθούν είναι οι παρακάτω:

- *-name*: Δηλώνει το όνομα του πεδίου και μπορεί να είναι ο,τιδήποτε θέλει ο χρήστης. Το όνομα αυτό θα χρησιμοποιηθεί σαν όνομα στήλης στο TRAFIL.
- *-type*: Δηλώνει τον τύπο δεδομένων του πεδίου. Επειδή θα χρησιμοποιηθεί σαν όρισμα κατά την δημιουργία του πίνακα στη βάση δεδομένων του TRAFIL, είναι απαραίτητο να είναι κατανοητός σαν τύπος δεδομένων από την MySQL.

- *-index*: Δείχνει την θέση του πεδίου στη γραμμή του trace file, και ακολουθεί την αρίθμηση που υποδεικνύει η παράμετρος *NumberOfColumns* που είδαμε νωρίτερα.
- *-unique*: Όλα τα πεδία που έχουν αυτή τη σημαία πρέπει να βρουν την ακολουθία χαρακτήρων που ακολουθεί τη σημαία σε κάθε γραμμή προκειμένου να ταιριάξει το παρόν metafile με το trace file. Αφού ταιριάξει το metafile μπορεί να ξεκινήσει η κανονική ανάλυση των αποτελεσμάτων της προσομοίωσης.
- *-delimiter*: Πολλές φορές κάποια πεδία στις γραμμές του trace file μπορεί να είναι ενωμένα με κάποιον ειδικό χαρακτήρα. Αυτός είναι και ο λόγος που μετράμε ξεχωριστά στο metafile το *NumberOfFields* και το *NumberOfColumns*. Για τον διαχωρισμό τους χρησιμοποιούμε τον χαρακτήρα που ορίζεται από τη σημαία *-delimiter*.
- *-startsWith/endsWith*: Κάποια πεδία περιέχουν ειδικούς χαρακτήρες πριν ή και μετά. Οι χαρακτήρες αυτοί δεν είναι χρήσιμοι και πρέπει να παραβλεφθούν στην τελική ανάλυση. Με αυτές τις σημαίες σημειώνονται ώστε το TRAFIL να τις αφαιρέσει.

Οι υπόλοιπες γραμμές σηματοδοτούν κάποιες επιπρόσθετες μετρήσεις που παράγονται, τις στήλες που απαιτούνται για τον υπολογισμό τους αλλά και το όνομα της μέτρησης. Η μόνη σημαία εδώ είναι το *-column* η οποία δέχεται σαν όρισμα το όνομα ενός πεδίου, όπως αυτό καθορίστηκε στις παραπάνω γραμμές από την σημαία *-flag*. Οι σημαίες αυτές είναι απαραίτητες σε κάθε metafile ώστε να μπορεί να παράγει μετρήσεις το TRAFIL.

3.2.2. Submetafiles

Όπως προαναφέρθηκε, τα βασικά είδη trace file του NS-2 περιγράφουν μόνο κάποια γενικά δίκτυα. Στην πράξη πολλές φορές χρησιμοποιούνται εξειδικεύσεις, οι οποίες παράγουν επιπλέον πληροφορία σε μορφή επιπρόσθετων στηλών στα trace files. Για το λόγο αυτό δημιουργήθηκαν συμπληρωματικά αρχεία στα metafiles, τα submetafiles. Όπως υποδηλώνεται και από το όνομά τους, καθένα είναι ξεχωριστό και ανήκει σε ένα metafile. Περιέχουν μόνο πεδία για την επιπλέον πληροφορία που καλούνται να αποκωδικοποιήσουν, ενώ δεν συμμετέχουν στην ταυτοποίηση του trace file αλλά μόνο στην ανάλυσή του. Παραθέτουμε μερικά submetafiles ως παράδειγμα.

```

NumberOfFields 4
NumberOfColumns 4
UniqueCounter 1
-name AckNumber -type int -index 1
-name FlagsTCP -type varchar(7) -index 2 -unique ox
-name HeaderLength -type int -index 3
-name SocketAddressLength -type int -index 4

```

Εικόνα 7 Normal TCP metafile

```

NumberOfFields 8
NumberOfColumns 8
UniqueCounter 6
-name AODVReuest_Type -type varchar(20) -index 1 -startsWith [ -unique ox
-name AODVRequest_HopCount -type int -index 2
-name AODVRequest_BroadcastID -type int -index 3
-name AODVRequest_Destination -type int -index 4 -unique [ -startsWith [
-name AODVRequest_DestinationSeenceNumber -type int -index 5 -unique ] -endsWith ]
-name AODVRequest_Source -type int -index 6 -startsWith [ -unique [
-name AODVRequest_SourceSequenceNumber -type int -index 7 -endsWith ]] -unique ]]
-name AODVRequest_Request -type varchar(20) -index 8 -startsWith ( -endsWith ) -unique )

```

Εικόνα 8 Old Wireless AODV Request submetafile

```

NumberOfFields 6
NumberOfColumns 5
UniqueCounter 3
-name SourceIpAddress -type int -index 2 -startsWith [ -unique [ -delimiter :
-name SourcePortNumber -type int -index 2
-name DestinationIpAddress -type int -index 3 -delimiter : -unique :
-name DestinationPortNumber -type int -index 3
-name TTLValue -type int -index 4
-name NextHopAddress -type int -index 5 -endsWith ] -unique ]

```

Εικόνα 9 Old Wireless IP submetafile

Παρατηρείται ότι η δομή των submetafiles ακολουθεί τους ίδιους κανόνες με αυτούς των metafiles. Υπάρχουν δυο μόνο διαφορές σε σχέση με τα metafiles. Η πρώτη είναι ότι δεν υπάρχουν πεδία μετρήσεων, αφού αυτά αναφέρονται στο κυρίως metafile.

Η δεύτερη είναι ότι η σημαία `-index` ξεκινάει την αρίθμηση από το 1, κάτι το οποίο γίνεται για τεχνικούς λόγους στην υλοποίηση του TRAFIL.

3.3. *Η δομή ενός TCL αρχείου*

Πέρα από την έξοδο μιας προσομοίωσης NS-2 θα πρέπει να εξετάσουμε και την μορφή της εισόδου. Όπως προαναφέρθηκε, ο NS-2 δέχεται σενάρια προσομοιώσεων γραμμένα σε γλώσσα προγραμματισμού OTcl, μια αντικειμενοστραφή έκδοση της Tcl. Σε κάθε περίπτωση απαιτεί κάποια συγκεκριμένα αντικείμενα να έχουν οριστεί ώστε να μπορέσει να αναγνωρίσει την προσομοίωση. Συγκεκριμένα, είναι απαραίτητο να οριστεί ένα αντικείμενο προσομοίωσης:

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

Επίσης, είναι απαραίτητο να οριστεί το πρόγραμμα της προσομοίωσης, δηλαδή τα χρονικά περιθώρια που θα γίνει η αποστολή των δεδομένων. Τέλος, αναλόγως του δικτύου που θέλουμε να προσομοιώσουμε είναι δυνατόν να χρειαστούν επιπλέον παράμετροι. Κατά κύριο λόγο η δομή ενός σεναρίου που περιγράφει ένα ενσύρματο δίκτυο θεωρείται απλούστερη από τη δομή ενός που περιγράφει ένα ασύρματο δίκτυο. Βασικό ρόλο και στις δυο περιπτώσεις όμως έχουν τα ίδια συστατικά στοιχεία δικτύων. Τα στοιχεία αυτά είναι τα εξής:

- *Κόμβοι δικτύου (nodes)*: Οι κόμβοι αποτελούν το κυριότερο συστατικό ενός δικτύου και είναι απαραίτητοι. Τουλάχιστον δυο κόμβοι χρειάζονται ώστε να χαρακτηριστεί μια τοπολογία “δίκτυο”.
- *Σύνδεσμοι δικτύου (links)*: Εδώ υπάρχει ένας μικρός διαχωρισμός μεταξύ των δυο τύπων δικτύου. Παρ’ όλα αυτά σαν έννοια ο σύνδεσμος (link) λειτουργεί με τον ίδιο τρόπο. Αναλαμβάνει να ενώσει δικτυακά δυο κόμβους, επιτρέποντας την μετάδοση δεδομένων μεταξύ τους.
- *Δικτυακά πρωτόκολλα (agents)*: Τα πρωτόκολλα αυτά λειτουργούν στους κόμβους που εμπλέκονται με τις μεταφορές δεδομένων. Αφορούν στο επίπεδο μεταφοράς των κόμβων και είναι απαραίτητο να οριστούν μόνο στον κόμβο του αποστολέα και του δέκτη.
- *Γεννήτριες κυκλοφορίας/Εφαρμογές (traffic generators/applications)*: Για την παραγωγή δεδομένων προς αποστολή χρησιμοποιούνται προσομοιώσεις διαφόρων πρωτοκόλλων επιπέδου εφαρμογής. Έτσι μπορούν να δημιουργηθούν

ρεαλιστικά σενάρια προσομοιώσεων, αφού οι γεννήτριες ή εφαρμογές που χρησιμοποιούνται ανταποκρίνονται σε ρεαλιστικές καταστάσεις.

Κάθε συστατικό στοιχείο του δικτύου έχει τις δικές του παραμέτρους, οι οποίες είτε δηλώνονται έμμεσα, λόγω της υπόλοιπης διάταξης του δικτύου, είτε άμεσα, μέσα από απευθείας ανάθεση στο Tcl αρχείο. Θα παρουσιάσουμε κάθε στοιχείο ξεχωριστά, ώστε να δούμε και τις παραμέτρους που υποστηρίζει το TRAFIL.

3.3.1. Κόμβοι δικτύου

Οι κόμβοι ενός δικτύου μπορεί να είναι είτε ενσύρματοι (wired) είτε ασύρματοι (wireless). Φυσικά σε ένα δίκτυο μπορεί να υπάρξει και συνδυασμός αυτών των δυο (wired-cum-wireless scenario²⁷). Η βασική εντολή εισαγωγής ενός νέου κόμβου στον NS-2 ακολουθεί το μοτίβο:

```
$ns \fcref{\textbf{node}}{../ns-2/ns-lib.tcl}{Simulator::node}
```

Μια τυπική δήλωση κόμβου όμως συνήθως είναι της μορφής :

```
set no [$ns node]
```

Στα βασικά χαρακτηριστικά των κόμβων περιλαμβάνεται η διεύθυνσή τους, η οποία συνήθως καθορίζεται αυτόματα από τον NS-2, μια λίστα από γειτονικούς κόμβους και μια λίστα από agents. Επίσης από προεπιλογή κάθε κόμβος θεωρείται ότι συμμετέχει σε ένα προς ένα μετάδοση (unicast transmission). Για να επιτραπεί η μετάδοση προς πολλά σημεία (multicast transmission) είναι απαραίτητο κατά τη δημιουργία της προσομοίωσης, δηλαδή η προσομοίωση να δηλωθεί με την σημαία “-multicast on”. Οι κόμβοι σε ένα δίκτυο μπορεί να έχουν και συντεταγμένες τοποθεσίας. Κάτι τέτοιο έχει νόημα κυρίως σε ασύρματα δίκτυα, αφού αντικαθιστά την έννοια των συνδέσμων. Κάθε κόμβος που είναι στο βεληνικές ενός άλλου θεωρείται ότι μπορεί να συνδεθεί με αυτόν αν ζητηθεί από το πρωτόκολλο δικτύου του.

3.3.2. Σύνδεσμοι δικτύου

Οι σύνδεσμοι σε ένα ενσύρματο δίκτυο φέρνουν σε επικοινωνία δυο κόμβους. Ο NS-2 υποστηρίζει πολλών ειδών συνδέσμους, οι πιο συνηθισμένοι όμως είναι οι απλοί (simplex), οι οποίοι είναι μονής κατεύθυνσης, και διπλοί (duplex), οι οποίοι είναι διπλής κατεύθυνσης. Οι σύνδεσμοι επίσης δέχονται διάφορες επιπλέον παραμέτρους όπως το εύρος ζώνης (bandwidth), η καθυστέρηση που εισάγουν (delay) και ο τύπος ουράς που χρησιμοποιούν. Συνηθισμένοι τύποι ουρών, τις οποίες υποστηρίζει και το TRAFIL, είναι η DropTail, Fair Queueing (FQ), Stochastic Fair Queueing (SFQ), Deficit Round Robin

(DRR) κλπ. Ο τρόπος με τον οποίο δηλώνεται π.χ. ένας απλός σύνδεσμος μονής κατεύθυνσης μεταξύ των κόμβων n0 και n4, με DropTail ουρά, εύρος ζώνης 100 Mbit/sec και 10ms καθυστέρηση είναι ο παρακάτω:

```
$ns duplex-link $no $n4 100.0Mb 10ms DropTail
```

3.3.3.

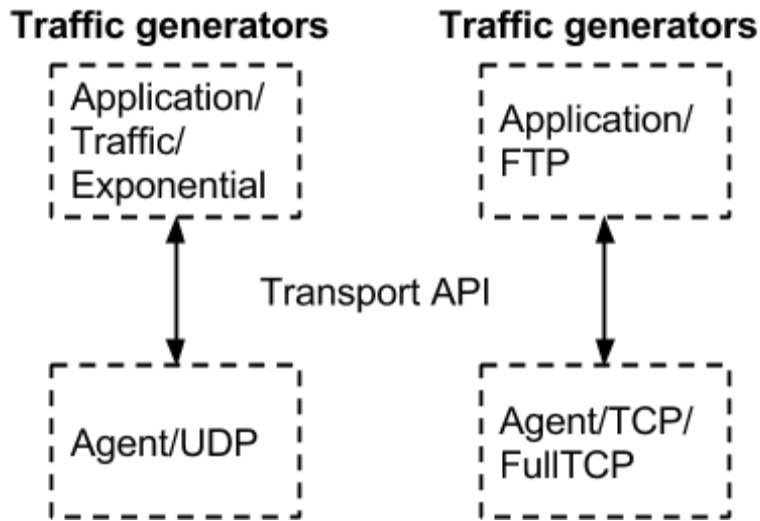
Γεννήτριες κυκλοφορίας/Εφαρμογές

Για την λειτουργία της προσομοίωσης χρειάζεται να παραχθούν δεδομένα, έστω και πλαστά. Παρ' ότι δεν έχει σημασία το περιεχόμενο των πακέτων δεδομένων που αποστέλλονται, είναι αρκετά σημαντικός ως προς την συμπεριφορά της προσομοίωσης ο τρόπος με τον οποίο παράγονται. Για παράδειγμα μια εφαρμογή μπορεί να παράγει δεδομένα σε σταθερό ρυθμό (constant bit rate - CBR) ή κατά ριπές. Για τον λόγο αυτό ο NS-2 προσφέρει μια ποικιλία γεννητριών δεδομένων προς αποστολή, ενώ έχουν επίσης προσομοιωθεί δυο γνωστά πρωτόκολλα επιπέδου εφαρμογής, το FTP και το Telnet²⁸. Οι παράμετροι που εφαρμόζονται στις γεννήτριες μπορεί να διαφέρουν μεταξύ τους, καθώς αλλάζει πολύ ο τρόπος που λειτουργούν.

3.3.4.

Δικτυακά πρωτόκολλα

Αναφερόμενα στον NS-2 ως agents, τα πρωτόκολλα αυτά λειτουργούν στο επίπεδο μεταφοράς και αναλαμβάνουν να δρομολογήσουν τα δεδομένα που παράγονται από τις γεννήτριες κυκλοφορίας προς τον προορισμό τους ή να τα δεχτούν. Συνδέονται άμεσα με έναν κόμβο, ενώ ένας κόμβος μπορεί να περιέχει περισσότερους του ενός agents. Κάθε εφαρμογή που παράγει δεδομένα επισυνάπτεται σε έναν agent πριν μπορέσει να γίνει η οποιαδήποτε μετάδοση. Παράδειγμα της σχέσης ενός agent με την γεννήτρια δεδομένων φαίνεται παρακάτω:



Εικόνα 10 Παράδειγμα συσχετισμών εφαρμογής - agent

Τυπικοί agents που υποστηρίζει ο NS-2 και το TRAFIL είναι ο TCP, ο οποίος εφαρμόζει τις συνθήκες του Transmission Control Protocol²⁹, και ο UDP ο οποίος εφαρμόζει το User Datagram Protocol³⁰ αντίστοιχα. Προσφέρεται επίσης ως καταβόθρα (sink) ένας ειδικός agent, ο Null. Λόγω της απλότητάς τους, πέραν του μεγέθους πακέτου ενός UDP agent, οι UDP και Null agents δεν έχουν επιπλέον παραμέτρους. Παρατίθεται παράδειγμα δήλωσης ενός UDP agent με γεννήτρια κυκλοφορίας σταθερού ρυθμού:

```
#Create a UDP agent and attach it to node no
set udpo [new Agent/UDP]
$ns attach-agent $no $udpo

# Create a CBR traffic source and attach it to udpo
set cbro [new Application/Traffic/CBR]
$cbro set packetSize_ 500
$cbro set interval_ 0.005
$cbro attach-agent $udpo
```

Εικόνα 11 Παράδειγμα δήλωσης agent

3.3.5.

Παράμετροι ασύρματων κόμβων

Στην περίπτωση ασύρματων δικτύων ένα πλήθος από επιπλέον παραμέτρους για τους κόμβους είναι απαραίτητο να προσδιοριστούν. Οι παράμετροι αυτοί αποθηκεύονται στο λεγόμενο “god object” του αρχείου σεναρίου και έχουν καθοριστικό ρόλο στη συμπεριφορά του συστήματος, καθώς ορίζουν σε μεγάλο βαθμό το φυσικό επίπεδο. Είναι οι εξής:

Παράμετροι	Παραδείγματα τιμών
------------	--------------------

-addressingType	flat or hierarchical or expanded
-ad hocRouting	DSDV or DSR or TORA
-llType	LL
-macType	Mac/802_11
-propType	"Propagation/TwoRayGround"
-ifqType	"Queue/DropTail/PriQueue"
-ifqLen	50
-phyType	"Phy/WirelessPhy"
-antType	"Antenna/OmniAntenna"
-channelType	"Channel/WirelessChannel"
-topoInstance	\$topo
-energyModel	"EnergyModel"
-initialEnergy	(in Joules)
-rxPower	(in W)
-txPower	(in W)
-agentTrace	ON or OFF
-routerTrace	ON or OFF
-macTrace	ON or OFF
-movementTrace	ON or OFF

Πίνακας 1 Παράμετροι κόμβων ασύρματου δικτύου

4. Αρχιτεκτονική του TRAFIL

Στην ενότητα αυτή θα γίνει παρουσίαση των διαφόρων λειτουργικών μονάδων του TRAFIL. Κάθε μονάδα στο TRAFIL έχει ένα δικό της ξεχωριστό στόχο, όμως συσχετίζεται και με τις υπόλοιπες μονάδες, έτσι ώστε να μπορεί να πάρει τα δεδομένα που χρειάζεται για να τον πραγματοποιήσει.

4.1. Οι λειτουργικές μονάδες του TRAFIL

Το TRAFIL πλέον εξυπηρετεί πολλούς σκοπούς, οι οποίοι εκτείνονται πέραν της φάσης του post-simulation και της ανάλυσης των αποτελεσμάτων μιας προσομοίωσης. Έτσι οι λειτουργικές μονάδες του έχουν αυξηθεί ώστε να καλύπτουν τις ανάγκες αυτές. Προσπαθώντας να τις κατηγοριοποιήσουμε βάσει των αναγκών που καλύπτουν έχουμε τις εξής μονάδες όσον αφορά την παραγωγή αποτελεσμάτων και μετρήσεων μετά την προσομοίωση:

- Κύρια Υπολογιστική Μονάδα (Main Processing Module)
- Μονάδα Παραγωγής Μετρήσεων (Metrics Module)
- Μονάδα Παραγωγής Γραφικών Παραστάσεων (Charts Module)
- Μονάδα Εκτέλεσης Ερωτημάτων προς τη Βάση Δεδομένων (SQL Queries Module)

Όμως, όπως είδαμε, πριν γίνει η χρήση των παραπάνω μονάδων, σημαντικό κομμάτι είναι η εξακρίβωση του τύπου ενός trace file. Η διαδικασία αυτή λαμβάνει υπόψιν της όλα τα metafiles τα οποία παρέχονται στο TRAFIL, ενώ χρειάζεται αμέσως μετά και η χρήση των κατάλληλων submetafiles για την σωστή ανάλυση των αρχείων. Για την ανάλυση των trace files και την αναγνώρισή τους το TRAFIL διαθέτει τις εξής μονάδες:

- Μονάδα Διαχείρισης Metafiles (Metafile Handler)
- Μονάδα Διαχείρισης Submetafiles (Submetafile Handler)

Όσον αφορά στην δυνατότητα του TRAFIL για εκτέλεση σεναρίων μέσω του NS-2, είτε αυτά είναι προκαθορισμένα είτε σχεδιασμένα μέσα από το TRAFIL, αλλά και την εκτέλεση προσομοιώσεων μετάδοσης βίντεο χρησιμοποιώντας το Evalvid-RA, υπεύθυνη είναι η εξής μονάδα:

- Μονάδα Προσομοίωσης (Simulation Module)

Τέλος, η μονάδα η οποία αναλαμβάνει τον σχεδιασμό ενός σεναρίου και την κωδικοποίησή αλλά και αποκωδικοποίηση σε Tcl script file είναι η εξής:

- Μονάδα Σχεδιασμού Προσομοίωσης (Simulation Plane Design Module)

Η δομή του TRAFIL χωρίζεται σε τρία επίπεδα αρχιτεκτονικής (3-layer architecture). Τα τρία αυτά επίπεδα είναι:

- Επίπεδο Παρουσίασης (Presentation Layer)
- Επίπεδο Εφαρμογής (Application Layer)
- Επίπεδο Πρόσβασης στη Βάση Δεδομένων (Database Access Layer)

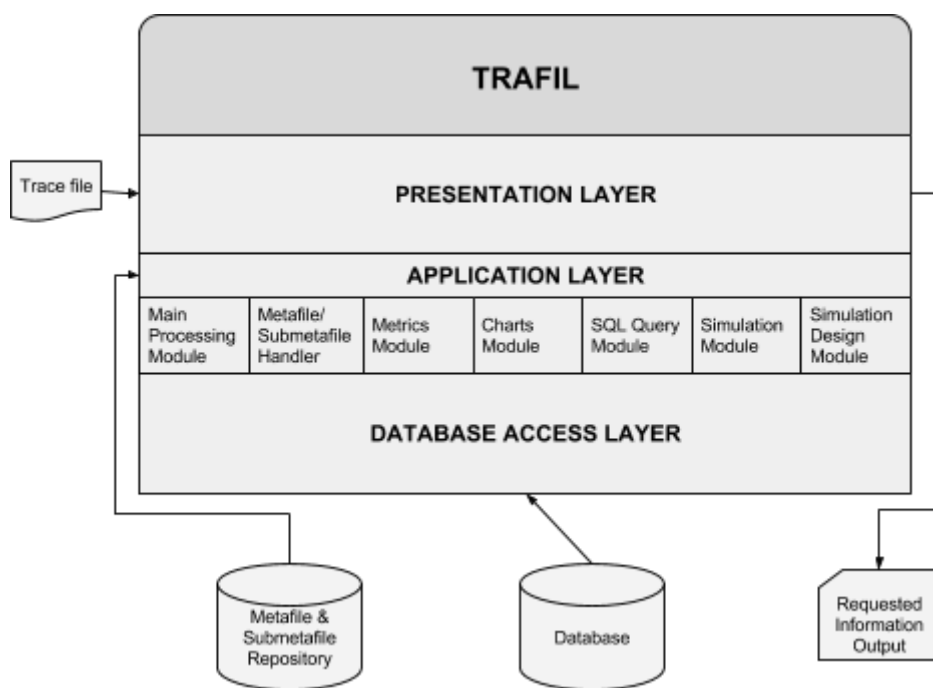
Το επίπεδο παρουσίασης αποτελείται από οτιδήποτε αλληλεπιδρά άμεσα με τον χρήστη. Περιλαμβάνει την γραφική διεπαφή με το χρήστη και όλες τις επιλογές που κάνει ο χρήστης ως είσοδο στο σύστημα. Έτσι το επίπεδο αυτό λειτουργεί ως μεσάζοντας μεταξύ του χρήστη και του επιπέδου εφαρμογής. Έτσι οι εσωτερικές λειτουργίες του TRAFIL δεν είναι ορατές στο χρήστη. Ο διαχωρισμός αυτός επιτρέπει την εύκολη συντήρηση και επέκταση του εργαλείου, είτε σε χαμηλότερα επίπεδα είτε στο επίπεδο παρουσίασης.

Το επίπεδο εφαρμογής παρεμβάλλεται μεταξύ του επιπέδου παρουσίασης και του επιπέδου πρόσβασης στη βάση δεδομένων. Αναλαμβάνει να εκτελέσει όλες τις λειτουργίες που ζητάει ο χρήστης μέσω του επιπέδου παρουσίασης, όπως μετρήσεις, αναλύσεις και διάφορα αποτελέσματα, και για αυτό το λόγο μπορεί να απαιτήσει αποθηκευμένα δεδομένα από το επίπεδο πρόσβασης στη βάση δεδομένων. Γι' αυτό το λόγο οι υπολογιστικές μονάδες που αναφέρθηκαν ανήκουν σε αυτό το επίπεδο ως επί το πλείστον. Παραμένουν, όμως, ανεξάρτητες μεταξύ τους, έτσι ώστε να μην επηρεάζει η καθεμία την λειτουργία της άλλης.

Το επίπεδο πρόσβασης στη βάση δεδομένων αναλαμβάνει να παρέχει οποιαδήποτε δεδομένα απαιτούνται από τις μονάδες του επιπέδου εφαρμογής ώστε να επιτελέσουν τη λειτουργία τους. Διαχειρίζεται την βάση δεδομένων του προγράμματος, όπου αποθηκεύει όλα τα trace files μετά την ανάλυσή τους. Έτσι δεν απαιτείται ξανά η ανάλυση ενός trace file, γλυτώνοντας σημαντικό χρόνο. Μάλιστα, κατά την έναρξη του προγράμματος, το επίπεδο αυτό ενημερώνει το επίπεδο παρουσίασης για τα διαθέσιμα αποθηκευμένα trace files, ώστε να φορτώσει όποιο επιθυμεί. Επίσης, προσφέρεται η δυνατότητα επεξεργασίας αυτών των δεδομένων και αποθήκευσης των αλλαγών, έτσι ώστε ο χρήστης να μπορεί να πειραματιστεί με αυτά και να παρατηρήσει τις αλλαγές στις

υπόλοιπες μετρήσεις. Τα χαρακτηριστικά αυτού του επιπέδου είναι μοναδικό γνώρισμα του TRAFIL σε σχέση με παρεμφερή εργαλεία.

Παράλληλα σε αυτά τα τρία επίπεδα αρχιτεκτονικής λειτουργούν διάφορα βοηθητικά εξωτερικά εργαλεία και φάκελοι, όπως το αποθετήριο metafiles και submetafiles (Metafile and Submetafile Repository), το οποίο ανήκει στην ιεραρχία φακέλων του εργαλείου. Εκεί τοποθετούνται ιεραρχημένα τα metafiles και submetafiles, είτε αυτά είναι τα προκαθορισμένα του TRAFIL είτε νέα από το χρήστη. Τα submetafiles ενός metafile πρέπει να τοποθετούνται βάσει κανόνα σε υποφάκελο του αποθετηρίου με όνομα υποφακέλου την ονομασία του metafile. Έτσι συσχετίζονται άμεσα και εύκολα με το συγκεκριμένο metafile. Επίσης, για την λειτουργία της προσομοίωσης μετάδοσης βίντεο μέσω Evalvid-RA χρησιμοποιούνται αρκετά τρίτα εργαλεία, επομένως και αυτά είναι αποθηκευμένα εσωτερικά στην ιεραρχία φακέλων του TRAFIL. Παρακάτω φαίνεται σχηματικά η αρχιτεκτονική του TRAFIL ως σύστημα.



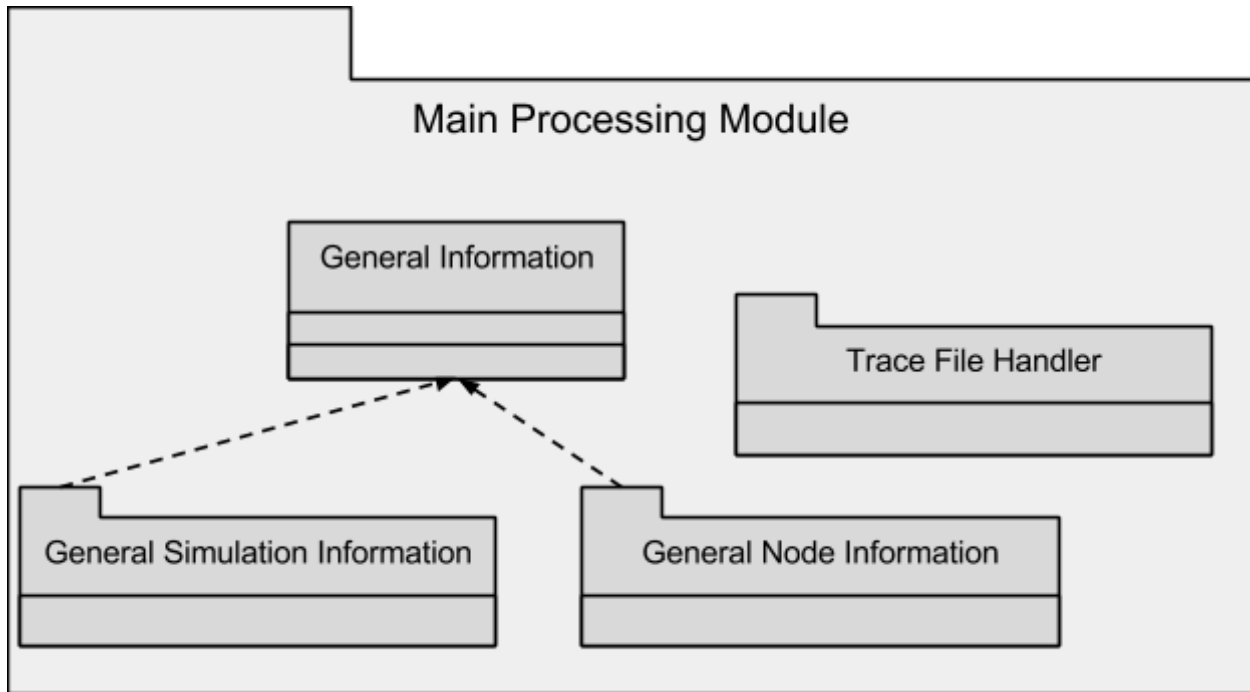
Εικόνα 12 Αρχιτεκτονική TRAFIL

4.2. Περιγραφή λειτουργιών του TRAFIL

Σε αυτή την υποενότητα περιγράφουμε αναλυτικά τις υπηρεσίες που προσφέρει κάθε λειτουργική μονάδα που αναφέρθηκε. Επίσης θα περιγραφούν τυχόν υποσυστήματα ή διασυνδέσεις που υπάρχουν μεταξύ των μονάδων αυτών.

4.2.1.

Κύρια Υπολογιστική Μονάδα (Main Processing Module)



Εικόνα 13 Κύρια υπολογιστική μονάδα

Η Κύρια Υπολογιστική Μονάδα είναι υπεύθυνη για δυο λειτουργίες:

- Επεξεργασία, ανάλυση και αποθήκευση του trace file
- Παραγωγή των γενικών πληροφοριών της προσομοίωσης

Γι' αυτό το λόγο βλέπουμε ότι αποτελείται από μερικά υποσυστήματα. Το πρώτο είναι το Trace File Handler, το οποίο αναλαμβάνει να ταιριάζει το trace file εισόδου με ένα metafile. Είναι πιθανόν ένα trace file να μην ταιριάζει με κανένα metafile. Στην περίπτωση αυτή η μονάδα αναλαμβάνει να ενημερώσει τον χρήστη μέσω του επιπέδου παρουσίασης. Σε αντίθετη περίπτωση το προωθεί για ανάλυση και επεξεργασία των δεδομένων του, ενώ αργότερα αυτά μπορεί να αποθηκευτούν στη βάση δεδομένων. Είναι αρκετά σημαντική η σωστή λειτουργία του υποσυστήματος αυτού, αφού από αυτό εξαρτώνται οι επόμενες. Οι ενέργειες του μπορούν να συνοψιστούν στις εξής:

- Ταυτοποίηση trace file
- Αντιστοίχιση metafile
- Επεξεργασία και ανάκτηση πληροφορίας

- Αποθήκευση στη βάση δεδομένων

Έχοντας τα δεδομένα της προσομοίωσης το TRAFIL παράγει τις γενικές πληροφορίες που την αφορούν, αλλά και γενικές πληροφορίες για τον κάθε κόμβο. Οι μετρήσεις αυτές είναι κατά κύριο λόγο κοινές μεταξύ τους, επομένως, όπως βλέπουμε και στο παραπάνω σχήμα, υλοποιούνται από την ίδια κλάση, την GeneralInformation. Οι πληροφορίες που παράγονται είναι οι παρακάτω:

1. Αριθμός απεσταλμένων πακέτων
2. Αριθμός απεσταλμένων bytes
3. Αριθμός παραγόμενων πακέτων
4. Αριθμός παραγόμενων bytes
5. Αριθμός προωθημένων πακέτων
6. Αριθμός προωθημένων bytes
7. Αριθμός πακέτων που απορρίφθηκαν
8. Αριθμός bytes που απορρίφθηκαν
9. Αριθμός πακέτων που λήφθηκαν
10. Αριθμός bytes που λήφθηκαν
11. Μέγιστο μέγεθος πακέτου
12. Ελάχιστο μέγεθος πακέτου
13. Μέσο μέγεθος πακέτου
14. Αριθμός πακέτων που έχουν παραχθεί στο AGT επίπεδο
15. Αριθμός bytes που έχουν παραχθεί στο AGT επίπεδο
16. Αριθμός πακέτων που έχουν ληφθεί στο AGT επίπεδο
17. Αριθμός bytes που έχουν ληφθεί στο AGT επίπεδο
18. Αριθμός πακέτων που έχουν παραχθεί στο RTR επίπεδο
19. Αριθμός bytes που έχουν παραχθεί στο RTR επίπεδο
20. Αριθμός πακέτων που έχουν ληφθεί στο RTR επίπεδο

21. Αριθμός bytes που έχουν ληφθεί στο RTR επίπεδο
22. Αριθμός πακέτων που έχουν παραχθεί στο MAC επίπεδο
23. Αριθμός bytes που έχουν παραχθεί στο MAC επίπεδο
24. Αριθμός πακέτων που έχουν ληφθεί στο MAC επίπεδο
25. Αριθμός bytes που έχουν ληφθεί στο MAC επίπεδο

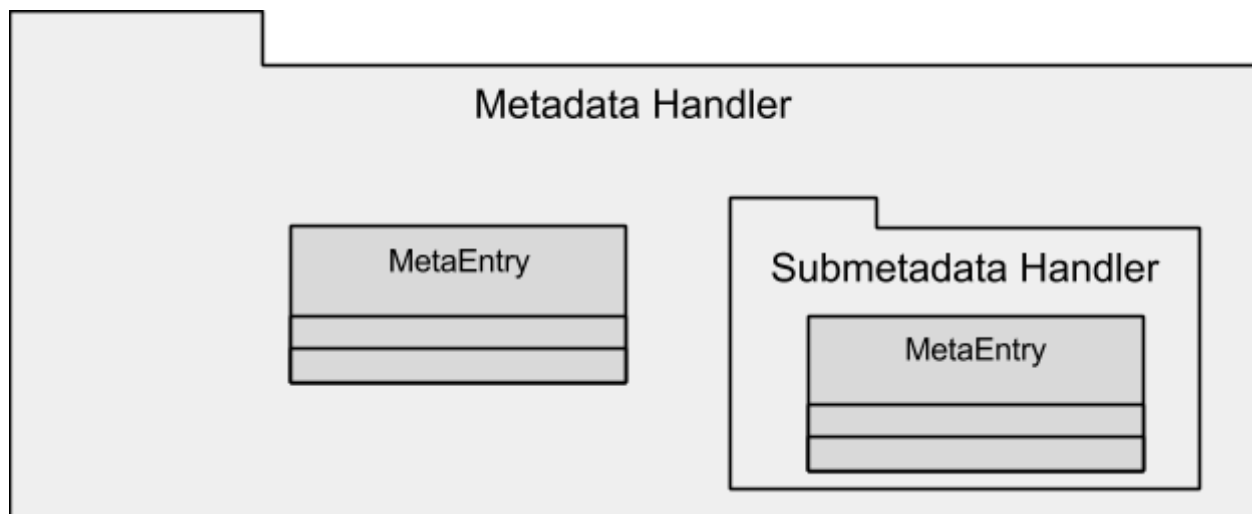
Ο χρήστης μπορεί να δει είτε τις μετρήσεις ενός συγκεκριμένου κόμβου μέσω μιας λίστας, είτε γενικές μετρήσεις της προσομοίωσης. Στην περίπτωση της προσομοίωσης υπάρχουν κάποιες επιπλέον πληροφορίες που δεν έχουν νόημα να υπολογιστούν για έναν κόμβο, οι εξής:

1. Χρόνος έναρξης
2. Χρόνος λήξης
3. Συνολικός χρόνος
4. Αριθμός κόμβων

Επίσης, οι μετρήσεις από το 14 έως το 25 αναφέρονται αποκλειστικά σε σενάρια με ασύρματα δίκτυα. Τα αποτελέσματα όλων των μετρήσεων εμφανίζονται στην καρτέλα “Simulation Information” του TRAFIL.

4.2.2.

Μονάδα Διαχείρισης Metafiles & Submetafiles (Metafile & Submetafile Handler)



Εικόνα 14 Μονάδα διαχείρισης metafiles (Metafile Handler)

Η μονάδα αυτή αποτελείται ουσιαστικά από δυο μονάδες, την Μονάδα Διαχείρισης Metafiles (Metafile Handler) και την Μονάδα Διαχείρισης Submetafiles (Submetafile Handler). Και οι δυο μονάδες έχουν την ίδια δομή, η οποία φαίνεται στο παραπάνω σχήμα, με την διαφορά ότι η δεύτερη είναι ουσιαστικά υπομονάδα της πρώτης, δηλαδή λειτουργεί εσωτερικά σε αυτήν.

Στην παραπάνω εικόνα φαίνεται η Μονάδα Διαχείρισης Metafiles και Submetafiles. Τα metafiles είναι ύψιστης σημασίας για τις υπηρεσίες post-simulation που προσφέρει το TRAFIL, επομένως η μονάδα αυτή είναι αρκετά σημαντική. Συγκεκριμένα, οι ενέργειες για τις οποίες είναι υπεύθυνη είναι οι εξής:

- Έλεγχος δομής κάθε metafile για να εξασφαλιστεί η ορθότητά του.
- Φόρτωση του κάθε metafile από το αποθετήριο κατά τη διαδικασία ταυτοποίησης, ανάλυσης και αποθήκευσης ενός trace file.
- Παροχή του σχήματος (schema) του πίνακα της βάσης δεδομένων στον οποίο θα αποθηκευτούν οι πληροφορίες που θα προκύψουν από το trace file.
- Παροχή όλων των πληροφοριών που είναι απαραίτητες για την παραγωγή των μετρήσεων για ένα τύπο trace file βάσει της ταυτοποίησης που έγινε με κάποιο metafile.
- Φόρτωση των αντίστοιχων submetafiles.
- Φόρτωση ενός trace file από τη βάση δεδομένων.

Η υπομονάδα διαχείρισης submetafiles είναι υπεύθυνη για τις εξής ενέργειες:

- Φόρτωση όλων των submetafiles
- Έλεγχος της ορθότητας της δομής των submetafiles
- Παροχή πληροφοριών που χρειάζονται για την κατασκευή του σχήματος (schema) του πίνακα στον οποίο θα αποθηκευτούν στη βάση δεδομένων οι πληροφορίες που θα προκύψουν από το trace file.

Η διαδικασία που ακολουθείται για κάθε καινούργιο trace file που εισάγεται είναι τυποποιημένη. Για κάθε metafile που υπάρχει αποθηκευμένο στο αποθετήριο δημιουργείται ένας ξεχωριστός χειριστής, ο λεγόμενος MetaDataHandler. Πρώτη εργασία ενός MetaDataHandler είναι ο έλεγχος της ορθότητας του metafile στο οποίο αντιστοιχεί. Συγκεκριμένα, ελέγχεται η δομή του metafile ώστε να ακολουθεί τους κανόνες που περιγράφηκαν στην ενότητα 3.2.1.

Εφόσον ο έλεγχος ορθότητας είναι επιτυχής, φορτώνεται στον `MetaDataHandler` κάθε εγγραφή του `metafile`, η λεγόμενη `MetaEntry` στην ορολογία του TRAFIL. Στην δομή του `metafile` οι εγγραφές αυτές περιγράφουν ένα πεδίο του `trace file`, δηλαδή περιέχουν τις σημαίες `-name`, `-type` και `-index`. Έτσι, ο αριθμός των `MetaEntries` καθορίζεται αποκλειστικά από τον αριθμό εγγραφών του `metafile` και δεν είναι αυστηρά καθορισμένος από το εργαλείο.

Όταν ετοιμαστούν όλοι οι `MetaDataHandlers` μπορεί να ξεκινήσει η διαδικασία της ταυτοποίησης του `trace file`. Όπως περιγράφηκε και στην ενότητα 3.2.1, το TRAFIL προσπαθεί να ταιριάξει το `trace file` με κάποιο `metafile` μέσω του `MetaDataHandler` του. Αυτό θα επιτρέψει στο TRAFIL να προχωρήσει στο επόμενο βήμα, που είναι η ανάλυση του `trace file` και η ετοιμασία του πίνακα της βάσης δεδομένων για τις πληροφορίες που θα εξαχθούν από αυτό.

Σε αυτό το σημείο, πριν δηλαδή ξεκινήσει η ανάλυση, αναλαμβάνουν ρόλο και οι `SubMetaDataHandlers`. Όπως ακριβώς και οι `MetaDataHandlers`, ελέγχουν την δομή των `submetafiles` στους οποίους έχουν αντιστοιχηθεί. Επίσης, ως άμεση επέκταση της εξάρτησης των `submetafiles` από τα `metafiles`, κάθε `SubMetaDataHandler` ανήκει σε κάποιον `MetaDataHandler`. Επομένως, εφόσον έχει ήδη αντιστοιχηθεί ένας `MetaDataHandler` με το `trace file` το οποίο πρόκειται να αναλυθεί, φορτώνονται μόνο οι `SubMetaDataHandlers` που ανήκουν σε αυτόν τον `MetaDataHandler`. Έπειτα, χρησιμοποιούνται κατά τη διάρκεια της ανάλυσης για την αναγνώριση των επιπλέον κεφαλίδων που μπορεί να υπάρχουν ανά γραμμή.

Με βάση αυτά, η πληροφορία του `trace file` μπορεί να αποθηκευτεί στη βάση δεδομένων. Για το σχήμα (`schema`) του πίνακα που θα δημιουργηθεί συνεισφέρουν τόσο ο `MetaDataHandler`, όσο και οι `SubMetaDataHandlers` που χρειάστηκαν κατά τη διάρκεια της ανάλυσης.

Μετά την αποθήκευση στη βάση δεδομένων ακολουθεί η παραγωγή μετρήσεων και πληροφοριών σχετικά με την προσομοίωση. Υπεύθυνη για αυτό είναι η Μονάδα Παραγωγής Μετρήσεων. Παρόλα αυτά, στην διαδικασία παραμένει απαραίτητος ο `MetaDataHandler`, αφού οι τιμές που χρειάζονται για κάθε μέτρηση αναφέρονται στο `metafile` που αντιστοιχήθηκε. Επομένως, ο `MetaDataHandler` παραμένει φορτωμένος καθ' όλη τη διάρκεια που είναι φορτωμένο και το `trace file` στο εργαλείο και φορτώνεται κάθε φορά που θα ζητηθεί πάλι το συγκεκριμένο `trace file`, αφού δεν μπορεί να γίνει οποιαδήποτε επεξεργασία του χωρίς αυτό.

4.2.3.

Μονάδα Παραγωγής Μετρήσεων (Metrics Module)

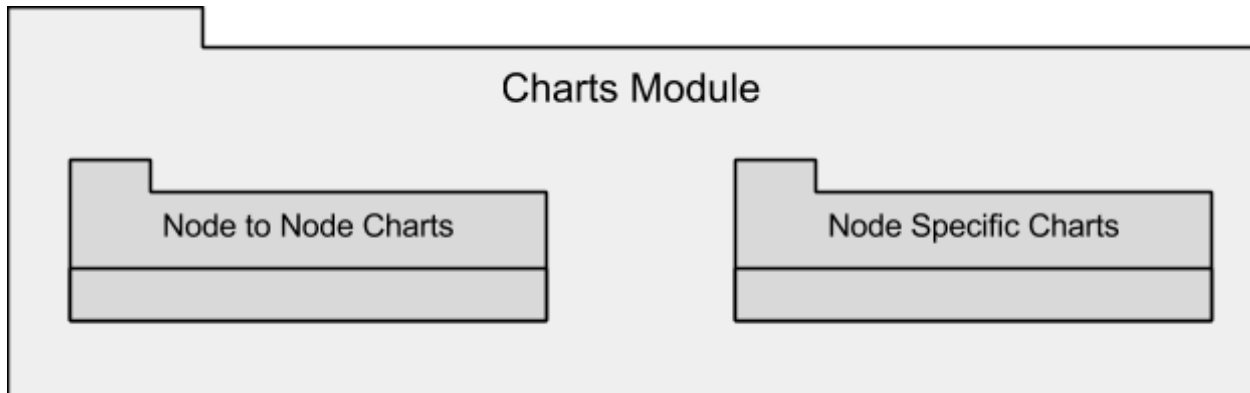
Η μονάδα αυτή είναι υπεύθυνη για την παραγωγή μετρήσεων. Πέρα από τις γενικές μετρήσεις που παράγονται, γίνονται και κάποιες μετρήσεις που αναφέρονται στο metafile που έχει αντιστοιχηθεί, όπως εξηγήθηκε στην προηγούμενη ενότητα. Οι μετρήσεις που παράγονται είναι οι εξής:

- Packet Delivery Rate (packets/second)
- Throughput (bits/second)
- Minimum End to End Delay (milliseconds)
- Maximum End to End Delay (milliseconds)
- Average End to End Delay (milliseconds)
- Delay Jitter (milliseconds)
- Average Delay Jitter (milliseconds)
- Minimum Delay Jitter (milliseconds)
- Maximum Delay Jitter (milliseconds)
- Packet Loss Ratio (packets/second)

Όλες οι μετρήσεις γίνονται είτε για ενσύρματες είτε για ασύρματες προσομοιώσεις και γι' αυτό το λόγο δίνεται και η επιλογή επιπέδου από το οποίο θα ληφθούν οι πληροφορίες. Δηλαδή, για ενσύρματες προσομοιώσεις υπάρχουν οι επιλογές Link Layer ή Network Layer, ενώ για ασύρματες οι επιλογές AGT, RTR ή MAC. Οι μετρήσεις επιλέγονται και εμφανίζονται στην καρτέλα Metrics του TRAFIL.

4.2.4.

Μονάδα Παραγωγής Γραφικών Παραστάσεων (Charts Module)



Εικόνα 15 Μονάδα διαχείρισης metafiles (Metafile Handler)

Με τη χρήση αυτής της μονάδας δημιουργούνται οι γραφικές παραστάσεις για το trace file που έχει επιλεγεί. Αυτές χωρίζονται σε δυο κατηγορίες:

- Node to Node Charts
- Node Specific Charts

Όπως υποδηλώνει και η ονομασία της, η πρώτη κατηγορία δημιουργεί γραφικές παραστάσεις μεταξύ δύο κόμβων, ενώ η δεύτερη αναφέρεται σε ένα μόνο κόμβο. Και στις δύο περιπτώσεις ο χρήστης μπορεί να δηλώσει το επίπεδο από το οποίο θα συγκεντρωθούν τα δεδομένα για τη γραφική παράσταση. Το επίπεδο αυτό μπορεί να είναι είτε Link Layer είτε Network Layer για ενσύρματη προσομοίωση, ενώ για ασύρματη οι επιλογές είναι AGT, RTR ή MAC.

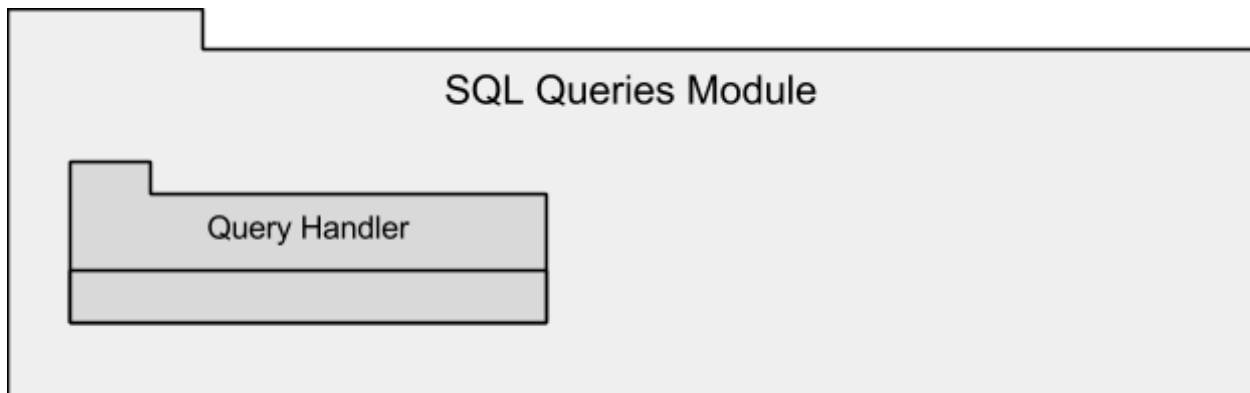
Τα είδη των γραφικών παραστάσεων που μπορούν να παραχθούν είναι τα εξής:

- Packet Delivery Rate (packets/second)
- Throughput (bits/second)
- Delay Jitter (milliseconds)
- Packet End to End Delay (packets/second)

Μια άλλη επιλογή που προσφέρεται σε κάθε γραφική παράσταση είναι ο ρυθμός δειγματοληψίας. Συγκεκριμένα, μπορεί να γίνει δειγματοληψία ανά δευτερόλεπτο ή ανά πέντε δευτερόλεπτα. Οι επιλογές αυτές θεωρήθηκαν υπεραρκετές ώστε να μπορεί ο χρήστης να έχει ικανοποιητικό επίπεδο λεπτομέρειας από τις γραφικές παραστάσεις.

4.2.5.

Μονάδα Εκτέλεσης Ερωτημάτων προς τη Βάση Δεδομένων (SQL Queries Module)



Εικόνα 16 Μονάδα Εκτέλεσης Ερωτημάτων προς τη Βάση Δεδομένων (SQL Queries Module)

Η μονάδα αυτή προσφέρει την απαραίτητη διεπαφή στο χρήστη για την άμεση αλληλεπίδραση με την βάση δεδομένων του TRAFIL. Όπως εξετάσαμε στα προηγούμενα modules, το TRAFIL παράγει τις περισσότερες πληροφορίες και μετρήσεις που χρειάζεται να πάρει ένας χρήστης από την προσομοίωση. Παρόλα αυτά είναι πάντα πιθανόν ο χρήστης να χρειάζεται κάποια επιπλέον πληροφορία που δεν προσφέρεται άμεσα. Αυτή την ανάγκη καλύπτει η συγκεκριμένη μονάδα. Εμφανίζει το schema του πίνακα στη βάση δεδομένων ώστε ο χρήστης να κάνει τα κατάλληλα ερωτήματα και να αποσπάσει την επιθυμητή πληροφορία.

Η δυνατότητα αυτή όμως μπορεί να προκαλέσει ανεπανόρθωτη ζημιά στη βάση δεδομένων. Γι' αυτό το λόγο τα ερωτήματα ελέγχονται μέσα από το υποσύστημα Query Handler, ως προς την ορθότητά τους, πριν περάσουν στην βάση. Ως ορθά επιλέγονται τα ερωτήματα που περιορίζονται στο να διαβάζουν δεδομένα, αλλά δεν αλλάζουν τις τιμές τους ή τη δομή της βάσης.

4.2.6.

Μονάδα Προσομοίωσης (Simulation Module)

Όπως έχει αναφερθεί ήδη, το TRAFIL προσφέρει βοήθεια και για την ίδια τη διαδικασία της προσομοίωσης. Συγκεκριμένα, προσφέρονται τρεις διαφορετικές περιπτώσεις εκτέλεσης μιας προσομοίωσης μέσα από το TRAFIL. Η απλούστερη είναι η απευθείας εκτέλεση ενός έτοιμου σεναρίου και η αυτόματη μεταφορά των αποτελεσμάτων στις υπηρεσίες post-simulation του εργαλείου. Η δεύτερη περίπτωση είναι η εκτέλεση ενός σεναρίου σχεδιασμένου στο ίδιο το TRAFIL, και θα αναλυθεί περισσότερο στην επόμενη ενότητα. Η τρίτη περίπτωση είναι η εκτέλεση μιας

προσομοίωσης μετάδοσης βίντεο με την βοήθεια μιας δημοφιλούς επέκτασης του NS-2, του Evalvid-RA^{31,32}.

Το Evalvid-RA βασίζεται στο Evalvid³³ και δίνει την δυνατότητα προεπεξεργασίας ενός βίντεο για να το προσαρμόσει στις ανάγκες της προσομοίωσης. Έπειτα, γίνεται μετάδοση μέσα από την τοπολογία που έχει ορίσει ο χρήστης χρησιμοποιώντας για την προσομοίωση τον NS-2 και παράγονται αποτελέσματα που αξιολογούν την μετάδοση που έγινε. Τα αποτελέσματα αυτά περιέχουν πληροφορίες, τις οποίες χρησιμοποιεί το Evalvid-RA ώστε να ανακατασκευάσει το μεταδιδόμενο βίντεο και να υπολογίσει κάποιες μετρήσεις, όπως θόρυβος (PSNR σε dB).

Ένα σημαντικό χαρακτηριστικό του Evalvid-RA είναι το γεγονός ότι δεν χρησιμοποιεί την δυαδική πληροφορία του αρχείου που μεταδίδεται κατά την προσομοίωση, καθώς κάτι τέτοιο είναι αρκετά χρονοβόρο. Αντίθετα, χρησιμοποιεί μεθόδους προεπεξεργασίας ώστε να παράξει ειδικά trace files πριν την προσομοίωση τα οποία αφορούν αποκλειστικά στο συγκεκριμένο βίντεο. Συνήθως η δομή τους περιέχει πληροφορίες για τον τύπο των frame, το μέγεθός τους, την κατάτμησή τους σε τμήματα (segments), τον αριθμό των frames και τον χρόνο μετάδοσης κάθε frame. Με αυτές τις πληροφορίες, οι οποίες ονομάζονται metadata του αρχείου, η μετάδοση μπορεί να προσομοιωθεί επακριβώς χωρίς να είναι απαραίτητη η χρήση της ίδιας της πληροφορίας του βίντεο.

Για την εκτέλεση της διαδικασίας επεξεργασίας, μετάδοσης και ανάλυσης αποτελεσμάτων το Evalvid-RA χρησιμοποιεί τρίτα εργαλεία. Τα εργαλεία αυτά χωρίζονται σε δύο κατηγορίες: αυτά που χρησιμοποιούνται πριν την προσομοίωση (pre-simulation) και αυτά που χρησιμοποιούνται μετά την προσομοίωση (post-simulation).

Τα εργαλεία που χρησιμοποιούνται πριν την προσομοίωση είναι τα εξής:

- *FFmpeg*³⁴: Το εργαλείο αυτό είναι ανοιχτού κώδικα (open-source), cross-platform, δηλαδή μπορεί να χρησιμοποιηθεί ανεξαρτήτως λειτουργικού συστήματος, και αναλαμβάνει την συμπίεση και αποσυμπίεση αρχείων βίντεο ή ήχου σε έναν μεγάλο αριθμό τύπων (format). Το Evalvid-RA το χρησιμοποιεί κυρίως για την μετατροπή .yuv αρχείων βίντεο, δηλαδή raw video files, σε άλλο τύπο, συνήθως .mp4.
- *MP4.exe*: Είναι ένα εργαλείο επίσης ανοιχτού κώδικα (open-source) που συνοδεύει το Evalvid-RA και χρησιμοποιείται για τη δημιουργία των trace files σχετικά με τα αρχεία βίντεο, όπως αναφέρθηκε προηγουμένως. Η είσοδος του Mp4.exe είναι ένα αρχείο βίντεο, συνήθως αυτό που παράγεται από το FFmpeg. Η

έξοδος του χρησιμοποιείται για την προσομοίωση της μετάδοσης. Παράδειγμα της εξόδου του Mp4.exe φαίνεται παρακάτω.

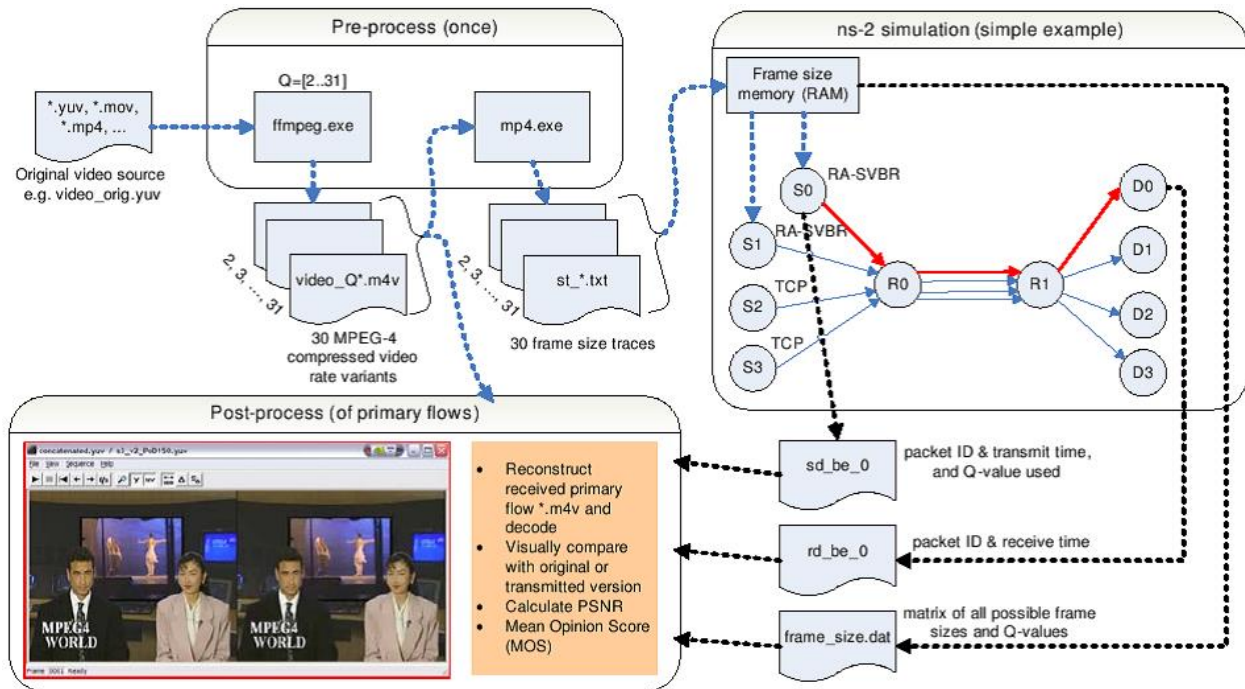
1	1	I	15876	16	0.002
2	2	P	1223	2	0.049
3	3	P	1116	2	0.084
4	4	P	937 1	0.103	
5	5	P	1163	2	0.140
6	6	P	1168	2	0.178
7	7	P	1079	2	0.215
8	8	P	1234	2	0.234
9	9	P	1290	2	0.272
10	10	P	1388	2	0.310
11	11	P	1230	2	0.347
12	12	P	1383	2	0.384
13	13	P	1727	2	0.403
14	14	P	2016	2	0.440
15	15	P	2387	3	0.478
16	16	P	3010	3	0.516

Εικόνα 17 Παράδειγμα MP4.exe trace file

Η επόμενη φάση, μετά την προσομοίωση με χρήση των παραπάνω αρχείων, αφορά στην ανακατασκευή του βίντεο και στην παραγωγή των μετρήσεων ώστε να αξιολογηθεί η μετάδοση. Κατά τη φάση αυτή χρησιμοποιούνται τα εξής εργαλεία:

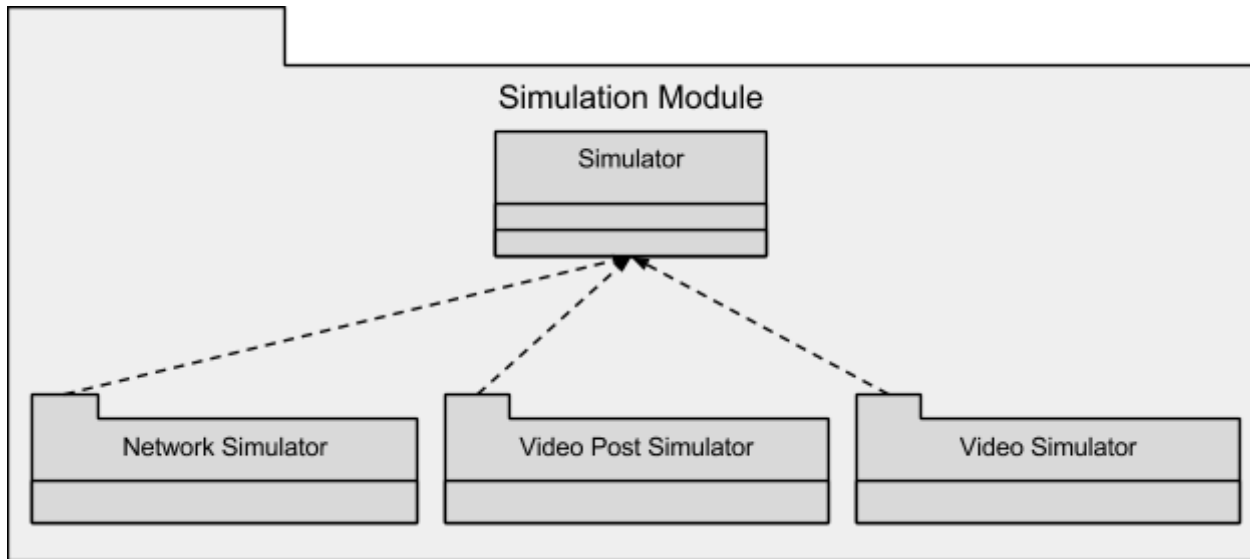
- *et_ra*: Το εργαλείο αυτό είναι ένα πρόγραμμα που κάνει τον υπολογισμό του jitter και του packet loss. Επιπλέον συνδυάζει τα δεδομένα εισόδου του, δηλαδή τα trace file που παράγονται από το MP4.exe και το mp4 video που εξήγαγε το FFmpeg, με τα αρχεία εξόδου, δηλαδή τα αρχεία που παράγονται από την προσομοίωση του NS-2. Ως αποτέλεσμα εξάγει ένα βίντεο τύπου m4v.
- *psnr.exe*: Το εργαλείο αυτό είναι το τελευταίο της διαδικασίας. Χρησιμοποιεί το αρχικό .yuv αρχείο και το ανακατασκευασμένο αρχείο βίντεο μετά την μετάδοση ώστε να παράγει το PSNR.

Το συνολικό διάγραμμα μιας προσομοίωσης είναι:



Εικόνα 18 Διαδικασία εκτέλεσης προσομοίωσης με Evalvid-RA και NS-2

Το TRAFIL προσφέρει την δυνατότητα αυτοματοποίησης όλων αυτών των διαδικασιών. Μέσω της γραφικής διεπαφής ο χρήστης μπορεί να εισάγει το βίντεο που θέλει να μεταδοθεί σε οποιαδήποτε μορφή καθώς και το σενάριο της προσομοίωσης και να πάρει απευθείας τα αποτελέσματα της εξόδου. Επίσης, δίνεται η δυνατότητα αλλαγής κάποιων παραμέτρων των εργαλείων που χρησιμοποιούνται, σε περίπτωση που υπάρχει αυτή η επιθυμία. Τη λειτουργία αυτή προσφέρει η Μονάδα Προσομοίωσης, όπως φαίνεται στην παρακάτω εικόνα.



Εικόνα 19 Μονάδα Προσομοίωσης (Simulation Module)

Όπως φαίνεται, η μονάδα αποτελείται από τρία υποσυστήματα:

- *Network Simulator*: Είναι υπεύθυνο για την εκτέλεση των απλών προσομοιώσεων που δεν εμπεριέχουν μετάδοση video.
- *Video Simulator*: Αναλαμβάνει την προεπεξεργασία του βίντεο και την εκτέλεση της προσομοίωσης. Για την προεπεξεργασία, όπως είδαμε, χρησιμοποιούνται τα προγράμματα FFmpeg και MP4.exe.
- *Video Post Simulator*: Αναλαμβάνει όλη τη διαδικασία post-simulation. Συγκεκριμένα, επιλέγει τις παραμέτρους για τα προγράμματα et_ra και FFmpeg, ενώ χρησιμοποιεί τελικά και το psnr.exe για να παράξει το PSNR. Η διαδικασία αυτή μπορεί να εκτελεστεί επανειλημμένα με διαφορετικούς συνδυασμούς παραμέτρων και αρχείων που δημιουργήθηκαν στις προηγούμενες φάσεις.

Όλες αυτές οι δυνατότητες δίνονται στο TRAFIL μέσα από την καρτέλα Evalvid-RA. Τέλος, όλα τα αρχεία που παράγονται είναι διαθέσιμα σε φάκελο που δηλώνεται μέσα από το TRAFIL, ώστε να μπορεί να τα εξετάσει και μόνος του ο χρήστης.

4.2.7. Μονάδα Σχεδιασμού Προσομοίωσης (Simulation Plane Design Module)

Οι δυνατότητες του TRAFIL κατά τη φάση pre-simulation εκφράζονται κυρίως μέσα από τη Μονάδα Σχεδιασμού Προσομοίωσης. Η μονάδα αυτή προσφέρει την απαραίτητη γραφική διεπαφή καθώς και τα υποκείμενα συστήματα ώστε να μεταφράζει μια τοπολογία σχηματισμένη γραφικά σε λειτουργικό σενάριο προσομοίωσης για NS-2.

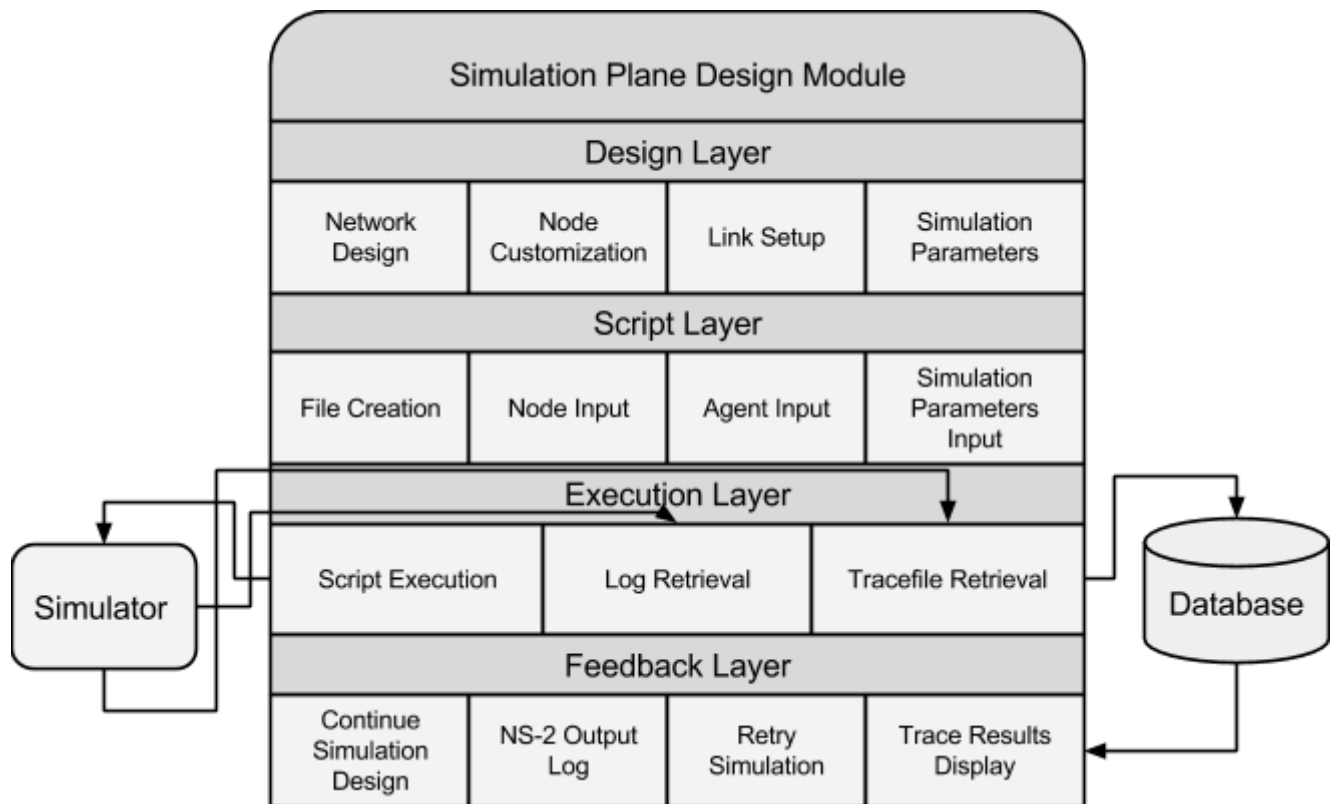
Ο χρήστης μπορεί να επιλέξει από την γραφική διεπαφή συστατικά του δικτύου (κόμβους, συνδέσμους κλπ) και να τα τοποθετήσει συνδυάζοντάς τα στην επιθυμητή διάταξη. Οποιοδήποτε στοιχείο μπορεί να παραμετροποιηθεί ή να αλλάξει μέσα από το ίδιο γραφικό περιβάλλον ή επιπλέον μενού που εμφανίζονται. Με αυτό τον τρόπο περιορίζεται η προγραμματιστική εργασία που χρειάζεται για την δημιουργία ενός σεναρίου, ενώ η διαδικασία μετατρέπεται κυρίως σε σχεδιαστική. Επιπλέον το σενάριο γίνεται προσφιλέστερο στο μάτι, ενώ γίνεται δυσκολότερο να παραβλεφθούν λεπτομέρειες ή λάθη που μπορεί να γίνουν σε ένα μεγάλο σενάριο προσομοίωσης.

Η μονάδα χωρίζεται σε 4 επίπεδα:

- *Επίπεδο Σχεδίασης:* Αποτελείται από την παλέτα, από όπου μπορούν να επιλεγούν τα διάφορα συστατικά ενός δικτύου, το πλαίσιο προσομοίωσης και μερικά αναδυόμενα μενού (pop-up menus). Το πλαίσιο προσομοίωσης (simulation panel) είναι ο διαδραστικός χώρος στον οποίο δημιουργείται και εμφανίζεται η τοπολογία του δικτύου και είναι παρεμφερής σε εμφάνιση με άλλα σχεδιαστικά εργαλεία που υπάρχουν. Τα αναδυόμενα μενού είναι ξεχωριστά παράθυρα τα οποία εμφανίζονται και περιέχουν προσαρμόσιμες παραμέτρους για συγκεκριμένα στοιχεία του σεναρίου, όπως οι σύνδεσμοι ή οι κόμβοι. Οι παράμετροι αυτοί, όπως το εύρος ζώνης ενός συνδέσμου ή οι παράμετροι ενός πρωτοκόλλου δικτύου, δεν θα μπορούσαν να εμφανιστούν γραφικά σε ευνόητη μορφή. Το επίπεδο αυτό ανήκει πλήρως στο Presentation Layer του TRAFIL.
- *Επίπεδο Σεναρίου:* Το επίπεδο αυτό ακολουθεί το επίπεδο σχεδίασης διαδικαστικά. Λαμβάνει όλα τα δεδομένα του σεναρίου, όπως αυτά ορίστηκαν στο προηγούμενο επίπεδο, και τα επεξεργάζεται ώστε να δημιουργήσει το αρχείο προσομοίωσης στη γλώσσα προγραμματισμού OTcl. Το αρχείο αυτό είναι δομημένο με συγκεκριμένο τρόπο, ενώ προστίθενται και επιπλέον πληροφορία με τη μορφή σχολίων, ώστε το TRAFIL να μπορεί να φορτώσει αργότερα το ίδιο αρχείο και να το αναπαραστήσει στην ίδια ακριβώς μορφή.
- *Επίπεδο Εκτέλεσης:* Το επίπεδο αυτό είναι προαιρετικό. Σε περίπτωση που επιλέξει ο χρήστης, το αρχείο που δημιουργήθηκε στο επίπεδο σεναρίου αποστέλλεται στο σύστημα προσομοίωσης δικτύου που είδαμε στην προηγούμενη υποενότητα. Το αποτέλεσμα της προσομοίωσης αποστέλλεται κατευθείαν στις μονάδες του TRAFIL που ασχολούνται με την φάση post-simulation, και φορτώνεται στη βάση δεδομένων. Το αποτέλεσμα προωθείται όμως και στο επόμενο επίπεδο.

- *Επίπεδο ανατροφοδότησης:* Κατά τον σχεδιασμό ενός σεναρίου είναι πιθανόν πολλές μικροαλλαγές να χρειαστούν. Το επίπεδο αυτό παρουσιάζει τα αποτελέσματα της προσομοίωσης και τυχόν σφάλματα που εμφανίστηκαν, ενώ παράλληλα δίνει την δυνατότητα επανεξέτασης του σεναρίου με επιστροφή στο πρώτο επίπεδο, επανεκτέλεση της προσομοίωσης ή και έξοδο από τη διαδικασία.

Η δομή της μονάδας φαίνεται καλύτερα στην παρακάτω εικόνα:



Εικόνα 20 Μονάδα Σχεδιασμού Προσομοίωσης (Simulation Plane Design Module)

Κατά τον σχεδιασμό του δικτύου πρωταρχικό ρόλο παίζει η τοποθέτηση των κόμβων. Κάθε κόμβος, όταν τοποθετηθεί, αποκτά ένα δικό του μενού παραμέτρων, οι οποίες έχουν κάποιες συνήθεις προκαθορισμένες τιμές, ώστε να μην μειωθεί ο χρόνος που σπαταλά ο χρήστης σε κάθε νέο κόμβο. Οι κόμβοι μπορεί να είναι είτε ενσύρματοι είτε ασύρματοι. Αναλόγως του είδους τους εμφανίζονται και διαφορετικές παράμετροι στο μενού. Προκειμένου να μην γίνει λάθος κατά την δήλωση τιμής μιας παραμέτρου, το TRAFIL περιορίζει τις επιλογές στις μοναδικές δυνατές κάθε στιγμή. Επίσης, με αυτόν τον τρόπο δεν υποχρεώνεται ο χρήστης να γνωρίζει από μνήμης τις τιμές κάθε παραμέτρου.

Σημαντικότερες παράμετροι σε έναν κόμβο είναι τα πρωτόκολλα δικτύου (agents) που χρησιμοποιεί καθώς και οι γεννήτριες κυκλοφορίας ή εφαρμογές με τις οποίες έχουν συνδεθεί. Το TRAFIL υποστηρίζει όλα τα συνηθισμένα πρωτόκολλα του NS-2, όπως TCP, UDP, Null. Το επιλεγμένο πρωτόκολλο εμφανίζει αυτόματα τις δυνατές γεννήτριες κυκλοφορίας ή εφαρμογές που μπορεί να δεχθεί, ενώ κάθε τέτοια γεννήτρια εμφανίζει τις δικές τις παραμέτρους όταν επιλεγεί.

Στα ενσύρματα δίκτυα το χαρακτηριστικότερο στοιχείο είναι οι σύνδεσμοι (links). Για να εγκαταστήσει ένα σύνδεσμο μεταξύ δυο κόμβων στο TRAFIL, ο χρήστης μπορεί να επιλέξει το αντίστοιχο στοιχείο από την παλέτα. Τότε ενεργοποιείται η λειτουργία σύνδεσης, όπου ο χρήστης μπορεί να επιλέξει δυο κόμβους ώστε να συνδεθούν. Ο νέος σύνδεσμος πέρα από την τοπολογία καταγράφεται και στο ξεχωριστό παράθυρο λίστας συνδέσμων, όπου αναφέρονται και οι παράμετροι όλων των συνδέσμων. Οι παράμετροι αυτές είναι τύπος συνδέσμου (μονής ή διπλής κατεύθυνσης), τύπος ουράς (DropTail, FQ, SFQ, DRR), εύρος ζώνης και καθυστέρηση μετάδοσης. Μέσα από το ίδιο παράθυρο ο χρήστης μπορεί να αλλάξει τις παραμέτρους αυτές.

Όταν τελειώσει ο σχεδιασμός του δικτύου, ο χρήστης μπορεί να καθορίσει ορισμένες παραμέτρους σημαντικές μόνο για την ίδια τη διαδικασία της προσομοίωσης, όπως τα ονόματα των αρχείων εξόδου και το χρονοδιάγραμμα της προσομοίωσης. Έπειτα αποφασίζει εάν θέλει να αποθηκεύσει το αρχείο ή να το εκτελέσει μέσω της Μονάδας Προσομοίωσης του TRAFIL.

Κατά τη δημιουργία του αρχείου σεναρίου σε μορφή OTcl πρώτα καταγράφονται οι γενικές πληροφορίες της προσομοίωσης, οι οποίες ελάχιστα διαφέρουν μεταξύ σεναρίων. Σε περίπτωση ασύρματης προσομοίωσης, καταγράφονται επίσης οι παράμετροι που θα ισχύουν για τους κόμβους. Έπειτα καταγράφεται η λίστα των κόμβων, οι σύνδεσμοι μεταξύ τους και οι ιδιότητες των συνδέσμων. Τέλος καταγράφονται οι agents και οι εφαρμογές που ανήκουν σε αυτούς, καθώς και οι παράμετροί τους, αφού ελεγχθούν για την ορθότητά τους. Τέλος καταγράφεται το χρονοδιάγραμμα της προσομοίωσης. Παράδειγμα σεναρίου σχεδιασμένου με τη χρήση TRAFIL φαίνεται στην παρακάτω εικόνα.

```

85 #Create agent udp7 and attach them to node n4
86 set udp7 [new Agent/UDP]
87 $ns attach-agent $n4 $udp7
88
89 #Create traffic sources and attach them to agent udp7
90 set cbr4 [new Application/Traffic/CBR]
91 $cbr4 set packetSize_ 200
92 $cbr4 set interval_ 0.3
93 $cbr4 attach-agent $udp7
94
95 #Schedule events for the cbr4 source
96 $ns at 0.5 "$cbr4 start"
97 $ns at 3.0 "$cbr4 stop"
98
99 #Create agent null8 and attach them to node n5
100 set null8 [new Agent/Null]
101 $ns attach-agent $n5 $null8
102
103 #Connect the traffic source with the traffic sink
104 $ns connect $tcp4 $null1
105 $ns connect $udp7 $null8
106
107 #Call the finish procedure after seconds of simulation time
108 $ns at 4.0 "finish"

```

Εικόνα 21 Μονάδα Σχεδιασμού Προσομοίωσης (Simulation Plane Design Module)

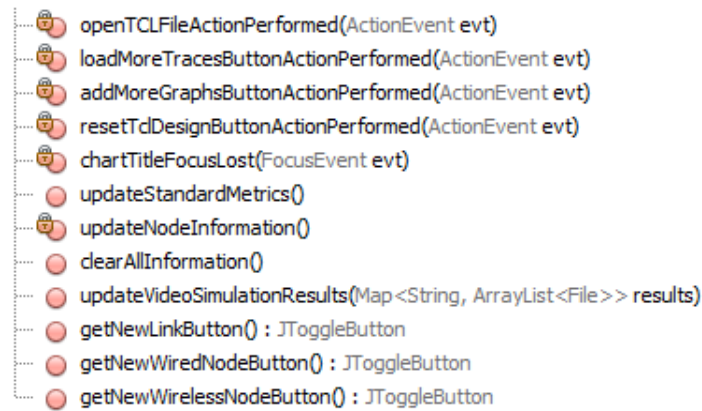
5. Σχεδίαση Συστήματος

Στην ενότητα αυτή παρουσιάζεται η σχεδίαση του συστήματος και πιο συγκεκριμένα τις σημαντικότερες κλάσεις που απαρτίζουν το TRAFIL. Αυτές ανήκουν στις μονάδες που αναφέραμε στην προηγούμενη ενότητα. Το TRAFIL υλοποιήθηκε σε γλώσσα προγραμματισμού Java, λαμβάνοντας υπόψη την cross-platform ιδιότητά της.

5.1. *Αρχιτεκτονική*

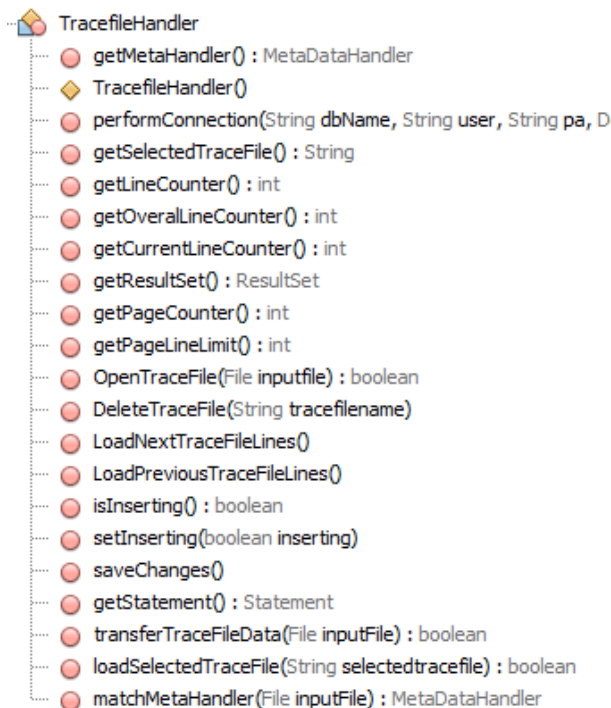
Εδώ παρουσιάζουμε τις κλάσεις του συστήματος αναφέροντας τις μεθόδους καθεμίας και δίνοντας μια σύντομη περιγραφή για την λειτουργία κάθε κλάσης. Σε περιπτώσεις κλάσεων που έχουν έντονη διάδραση μεταξύ τους θα παρουσιάζονται οι σχέσεις αυτές και σχηματικά.

- TRAFIL :: JFrame
 - TRAFIL()
 - getConnectionedAgents() : ConnectedAgentsWindow
 - getTdDesigner() : TdDesignerPanel
 - getLinkWindow() : LinkListWindow
 - initComponents()
 - connectActionPerformed(ActionEvent evt)
 - cancelActionPerformed(ActionEvent evt)
 - ExitActionPerformed(ActionEvent evt)
 - OpenTraceFileActionPerformed(ActionEvent evt)
 - previousActionPerformed(ActionEvent evt)
 - nextActionPerformed(ActionEvent evt)
 - DeleteActionPerformed(ActionEvent evt)
 - savechangesActionPerformed(ActionEvent evt)
 - selecttracefileActionPerformed(ActionEvent evt)
 - nodeSelectorActionPerformed(ActionEvent evt)
 - calculateMetricsActionPerformed(ActionEvent evt)
 - exportToExcellActionPerformed(ActionEvent evt)
 - exportToTxtFileActionPerformed(ActionEvent evt)
 - exportSimultaionInfoActionPerformed(ActionEvent evt)
 - exportMetricsActionPerformed(ActionEvent evt)
 - createChartActionPerformed(ActionEvent evt)
 - exportChartActionPerformed(ActionEvent evt)
 - openOTCLScriptActionPer formed(ActionEvent evt)
 - executeNS2Simulation(String path)
 - enableNodeSpecificActionPerformed(ActionEvent evt)
 - executeSQLActionPerformed(ActionEvent evt)
 - dearSQLActionPerformed(ActionEvent evt)
 - enableNodetoNodeActionPerformed(ActionEvent evt)
 - DesignQueryButtonActionPerformed(ActionEvent evt)
 - browseRawVideoActionPerformed(ActionEvent evt)
 - browseSimulationScriptActionPerformed(ActionEvent evt)
 - executePreAndSimulationActionPerformed(ActionEvent evt)
 - executepostSimulationActionPerformed(ActionEvent evt)
 - dbnameKeyReleased(KeyEvent evt)
 - connect_panelKeyReleased(KeyEvent evt)
 - usernameKeyReleased(KeyEvent evt)
 - passwordKeyReleased(KeyEvent evt)
 - senderFileSelectActionPerformed(ActionEvent evt)
 - newWiredNodeButtonActionPerformed(ActionEvent evt)
 - chartStartNodeActionPerformed(ActionEvent evt)
 - simParamButtonActionPerformed(ActionEvent evt)
 - linkListButtonActionPerformed(ActionEvent evt)
 - newWirelessNodeButtonActionPer formed(ActionEvent evt)
 - newLinkButtonActionPerformed(ActionEvent evt)
 - connectedAgentsButtonActionPerformed(ActionEvent evt)
 - wirelessSettingsButtonActionPer formed(ActionEvent evt)



Εικόνα 22 Η κλάση TRAFIL

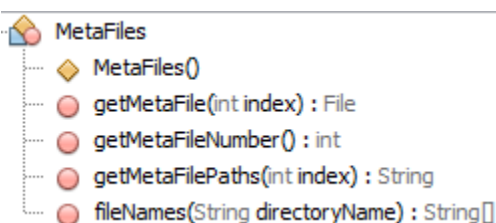
Είναι η κύρια κλάση του TRAFIL. Περιέχει μεταξύ άλλων την μέθοδο main(), δηλαδή το πρόγραμμα ξεκινάει την εκτέλεσή του από αυτή την κλάση. Η κλάση αυτή περιέχει το μεγαλύτερο μέρος της γραφικής διεπαφής χρήστη του TRAFIL, έχοντας όχι μόνο όλα τα στοιχεία της αλλά και τον χειρισμό των δράσεων (action) καθενός. Στην ουσία η κλάση αυτή είναι το Presentation Layer, όπως περιγράφηκε στην ενότητα 4.1, και δέχεται τις εντολές του χρήστη πριν τις περάσει στο Application Layer και εν τέλει επιστρέφει τα αποτελέσματα σε αυτόν.



Εικόνα 23 Η κλάση TraceFileHandler

Η κλάση `TraceFileHandler` είναι η σημαντικότερη στη διαδικασία αναγνώρισης και ταυτοποίησης ενός `trace file`. Χρησιμοποιεί ένα `MetaDataHandler` για την ταυτοποίηση, επεξεργασία και τέλος αποθήκευση του `trace file`. Επίσης, η κλάση αυτή δημιουργεί τις απαραίτητες συνδέσεις προς τη βάση δεδομένων, φορτώνει ένα αποθηκευμένο `trace file`, αποθηκεύει πιθανές αλλαγές του χρήστη ή και το διαγράφει από τη βάση αυτή.

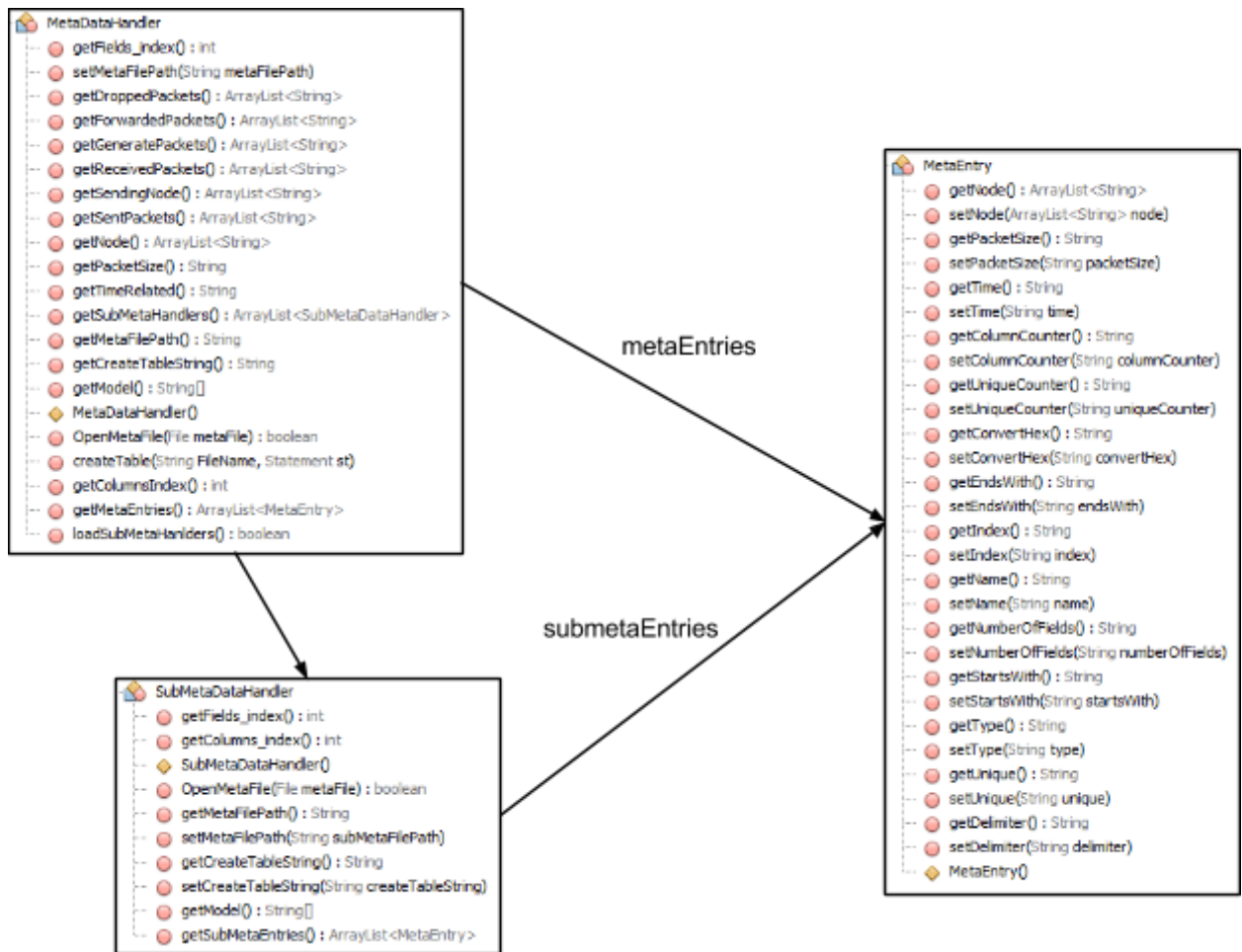
Η κλάση `MetaDataHandler` περιέχει τις πληροφορίες της δομής ενός `trace file`, όπως αυτό περιγράφεται από το `metafile` που αντιστοιχεί σε αυτόν. Επίσης, έχει μεθόδους που αφορούν στο σχήμα του πίνακα της βάσης δεδομένων αλλά και άλλες για να φορτώσει όλους τους `SubMetaDataHandlers` που του αντιστοιχούν.



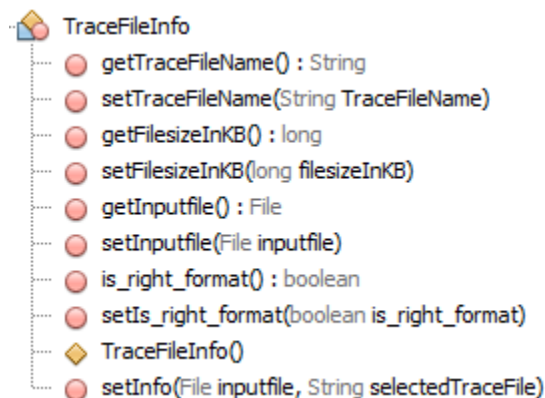
Εικόνα 24 Η κλάση `MetaFiles`

Η κλάση αυτή διαχειρίζεται τα `metafiles` που υπάρχουν στο αποθετήριο. Ανιχνεύει, δηλαδή, τα `metafiles` που υπάρχουν εκεί και επιστρέφει το μονοπάτι (`path`) τους σε όσες λειτουργικές μονάδες το χρειαστούν.

Παρόμοια με τις κλάσεις `MetaDataHandler` και `MetaFiles` υπάρχουν οι κλάσεις `SubMetaDataHandler` και `SubMetaFiles`. Όπως είπαμε σε προηγούμενη ενότητα, κάθε `MetaDataHandler` έχει μια σειρά από `SubMetaDataHandlers` οι οποίοι περιέχουν την πληροφορία που υπάρχει στα συγκεκριμένα `submetafiles`. Επίσης, και οι δυο `handlers` έχουν έναν αυθαίρετο αριθμό από `MetaEntries` που περιγράφουν την πληροφορία αυτή. Η σχέση μεταξύ των κλάσεων αυτών φαίνεται παρακάτω:



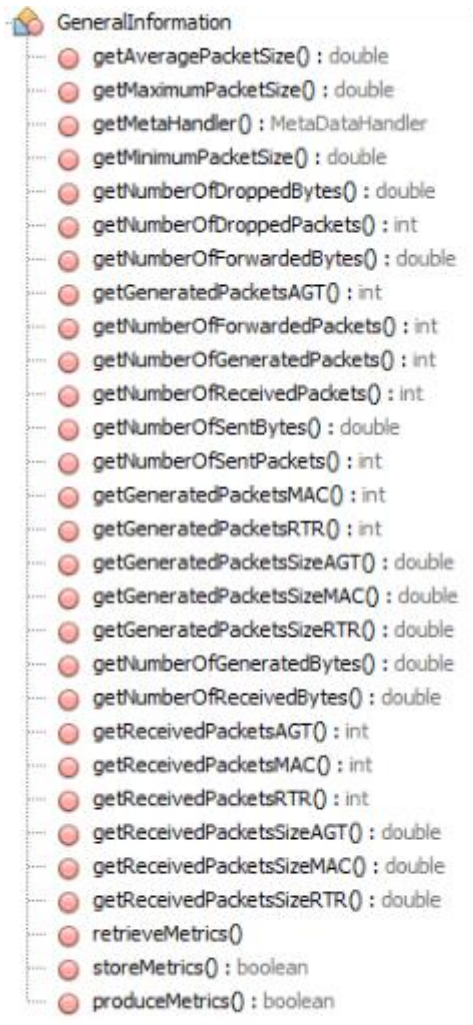
Εικόνα 25 Οι κλάσεις MetaDataHandler, SubMetaDataHandler και MetaEntry, καθώς και οι μεταξύ τους σχέσεις



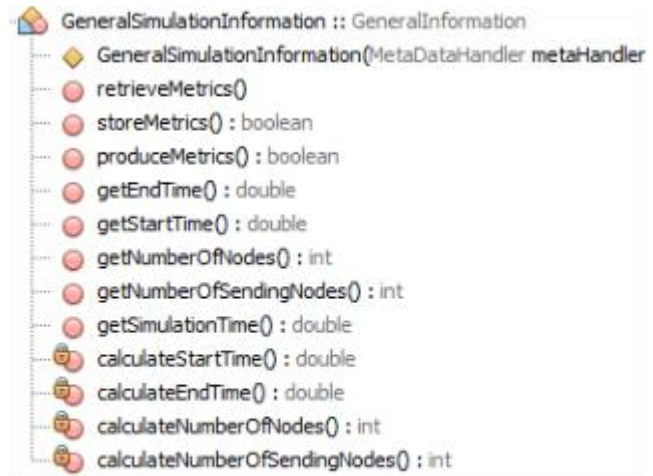
Εικόνα 26 Η κλάση TraceFileInfo

Στην κλάση TraceFileInfo αποθηκεύονται κάποιες γενικές πληροφορίες για το trace file που είναι φορτωμένο κάθε στιγμή στο TRAFIL. Οι πληροφορίες αυτές, όπως

φαίνεται στην εικόνα που δείχνει την δομή της κλάσης, δεν είναι πλήρεις αλλά χρησιμοποιούνται για γενικούς σκοπούς μόνο από διάφορες άλλες κλάσεις.

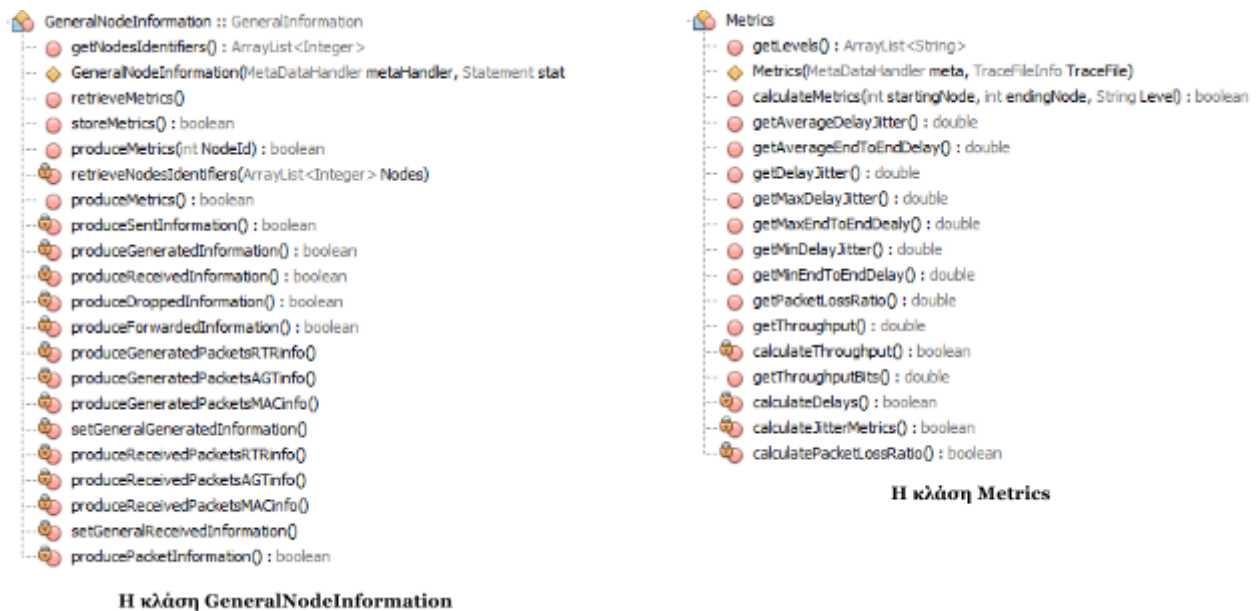


Η κλάση GeneralInformation



Η κλάση GeneralSimulationInformation

Εικόνα 27 Οι κλάσεις GeneralInformation και GeneralSimulationInformation



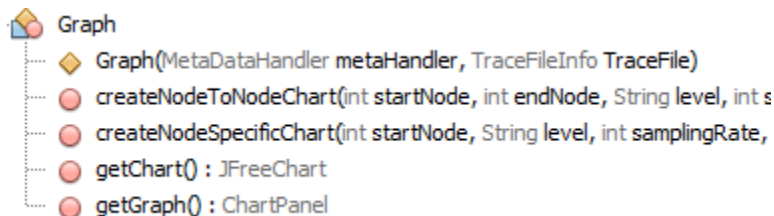
Εικόνα 28 Οι κλάσεις GeneralNodeInformation και Metrics

Η κλάση GeneralInformation είναι abstract κλάση που ορίζει τα χαρακτηριστικά που πρέπει να καλύπτουν οι κλάσεις που την υλοποιούν. Τα χαρακτηριστικά αυτά είναι είτε πληροφορίες, είτε μετρήσεις που κάθε προσομοίωση πρέπει να παρέχει.

Οι δυο κλάσεις που υλοποιούν την GeneralInformation είναι οι GeneralSimulationInformation και GeneralNodeInformation. Η GeneralSimulationInformation είναι υπεύθυνη για τις γενικές πληροφορίες της προσομοίωσης. Ορίζει, επίσης, μερικές πιο ειδικές ιδιότητες που αφορούν μόνο στις γενικές πληροφορίες. Η GeneralNodeInformation είναι υπεύθυνη για την παραγωγή πληροφοριών που αφορούν σε ένα συγκεκριμένο κόμβο τον οποίο ορίζει ο χρήστης. Και οι δυο κλάσεις δέχονται ως όρισμα τον MetaDataHandler, ο οποίος και θα δίνει πληροφορίες για την δομή του επιλεγμένου trace file. Έτσι μπορούν να αντιστοιχηθούν οι τιμές που χρειάζονται για την παραγωγή των μετρήσεων. Οι κλάσεις αυτές ουσιαστικά υλοποιούν την Κεντρική Υπολογιστική Μονάδα.

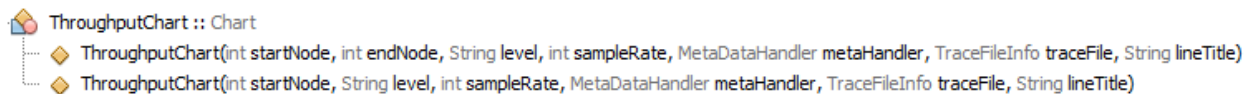
Η κλάση Metrics υλοποιεί την Μονάδα Παραγωγής Μετρήσεων. Η διαδικασία παραγωγής μετρήσεων είναι από τις πιο απαιτητικές υπολογιστικά στο TRAFIL και θεωρητικά επηρεάζεται αρκετά η επίδοση σε σχέση με το μέγεθος του trace file. Παρ' όλα αυτά, το TRAFIL καταφέρνει να κρατάει την ταχύτητα υπολογισμού σε αρκετά υψηλά επίπεδα, χρησιμοποιώντας πολυνηματικό προγραμματισμό. Έτσι, κάθε εκτέλεση υπολογισμού μετρήσεων χωρίζεται σε 4 κατηγορίες υπολογισμών, καθεμία από τις οποίες υπολογίζεται σε ξεχωριστό νήμα. Ο σχεδιασμός αυτός εκμεταλλεύεται το γεγονός ότι οι

σημερινοί επεξεργαστές είναι πολυπύρηντοι και προσφέρουν δυνατότητες παραλληλίας. Συνδυάζοντας το πλεονέκτημα αυτό με κατάλληλο σχεδιασμό της βάσης δεδομένων και τοποθετώντας κατάλληλους δείκτες στους πίνακες, μπορεί να βελτιώσει δραματικά τον χρόνο υπολογισμού.

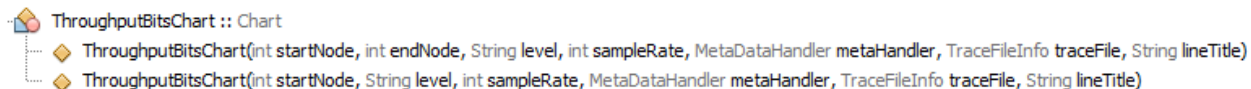


Εικόνα 29 Η κλάση Graph

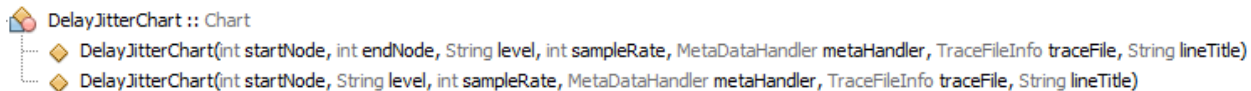
Η πιο πάνω κλάση αποτελεί το σύνδεσμο μεταξύ Presentation Layer, δηλαδή της κλάσης TRAFIL, και της δημιουργίας μιας γραφικής παράστασης. Η εντολή δημιουργίας ενός γραφήματος από τη γραφική διεπαφή μεταβιβάζεται σε αυτή την κλάση, η οποία έχει τις κατάλληλες βοηθητικές μεθόδους για την παραγωγή οποιαδήποτε γραφικής παράστασης. Χρησιμοποιείται η κλάση που αναλαμβάνει να φτιάξει την παράσταση σύμφωνα με τον τύπο γραφήματος που ζητήθηκε. Οι κλάσεις που δημιουργούν τις παραστάσεις είναι οι εξής:



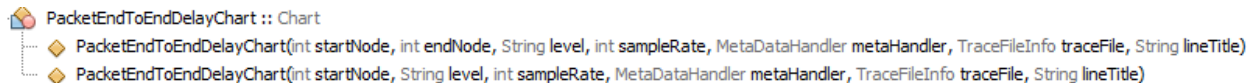
Εικόνα 30 Η κλάση ThroughputChart



Εικόνα 31 Η κλάση ThroughputBitsChart

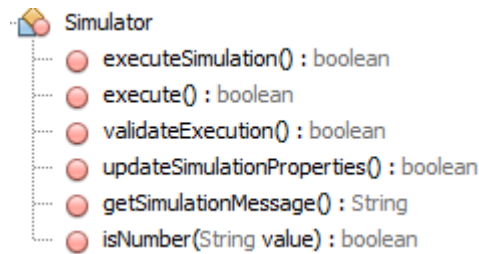


Εικόνα 32 Η κλάση DelayJitterChart

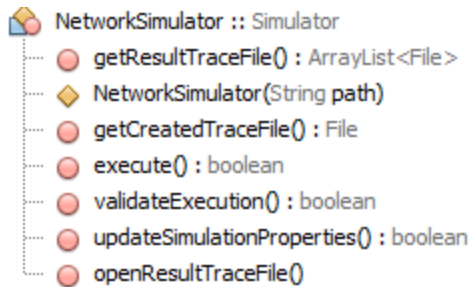


Εικόνα 33 Η κλάση PacketEndtoEndDelayChart

Η κλάση Simulator που φαίνεται στην παρακάτω εικόνα είναι και αυτή μια abstract κλάση η οποία ορίζει κάποιες ιδιότητες και μεθόδους που πρέπει να ικανοποιούν όλες οι κλάσεις που την υλοποιούν.

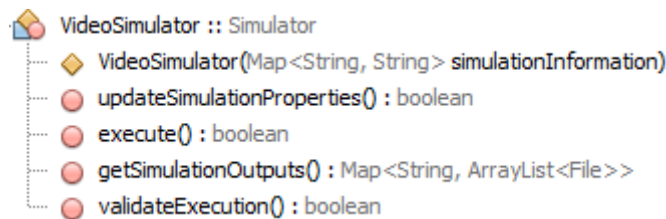


Εικόνα 34 Η κλάση Simulator

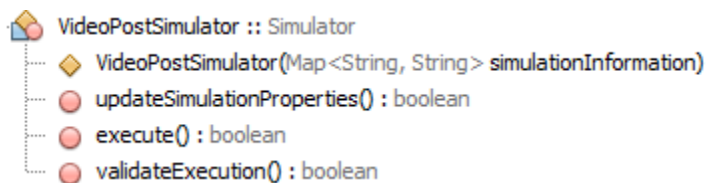


Εικόνα 35 Η κλάση NetworkSimulator

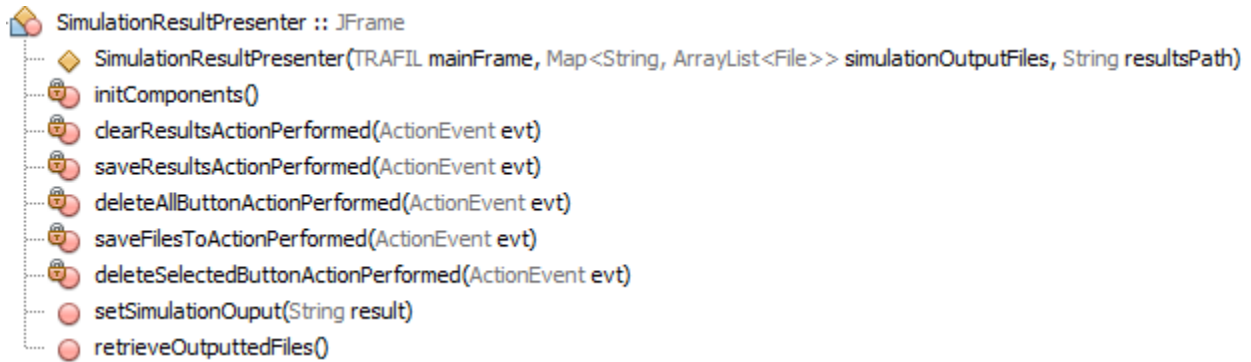
Η κλάση NetworkSimulator αναλαμβάνει τις απλές προσομοιώσεις, οι οποίες τροφοδοτούνται απευθείας στον NS-2. Η κλάση θεωρεί ένα αρχείο Tcl ως είσοδο στη μεταβλητή path, το εκτελεί μέσω του NS-2 και στέλνει απευθείας το παραγόμενο trace file για ανάλυση. Σε περίπτωση που παράγονται περισσότερα του ενός trace files ο χρήστης επιλέγει αυτό που θα αναλυθεί.



Εικόνα 36 Η κλάση VideoSimulator



Εικόνα 37 Η κλάση VideoPostSimulator



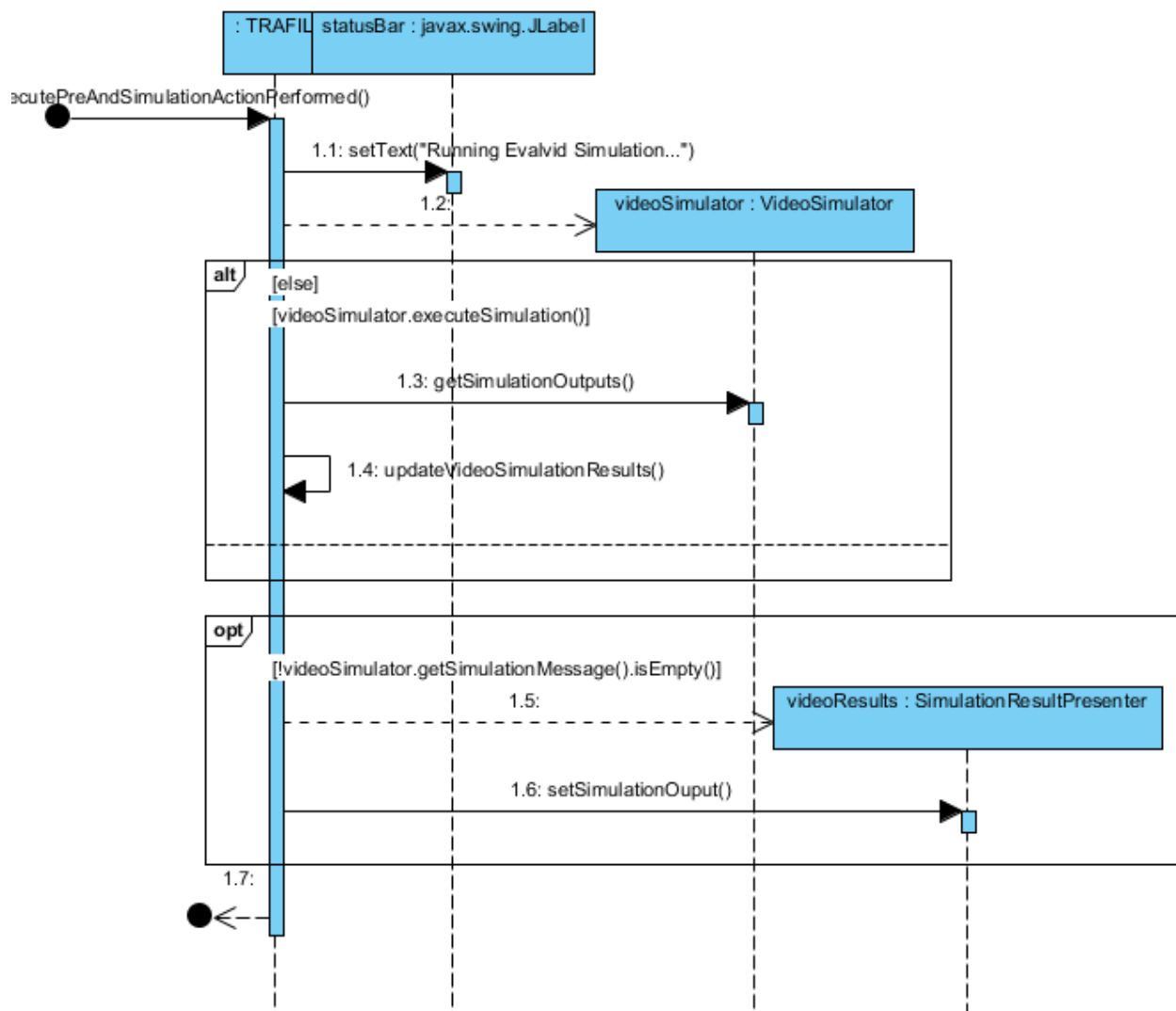
Εικόνα 38 Η κλάση **SimulationResultPresenter**

Οι πρώτες δυο κλάσεις είναι αντίστοιχες των υποσυστημάτων που είδαμε στην ενότητα 4.2.6. Η κλάση **VideoSimulator** είναι η κλάση που ασχολείται με το κομμάτι *pre-simulation* και καλεί όλα τα βοηθητικά προγράμματα να προεπεξεργαστούν την είσοδο του προσομοιωτή. Τα βοηθητικά προγράμματα του *pre-simulation*, τα οποία αναφέρθηκαν στην ενότητα 4.2.6, καλούνται μέσω εξωτερικού *bash script* αρχείου (.sh). Το *script* αυτό λαμβάνει τις παραμέτρους των προγραμμάτων μέσω ενός *properties* αρχείου, ενώ η έξοδός του διαβάζεται μέσω ειδικής κλάσης που διατηρεί το **TRAFIL** σε ξεχωριστό νήμα.

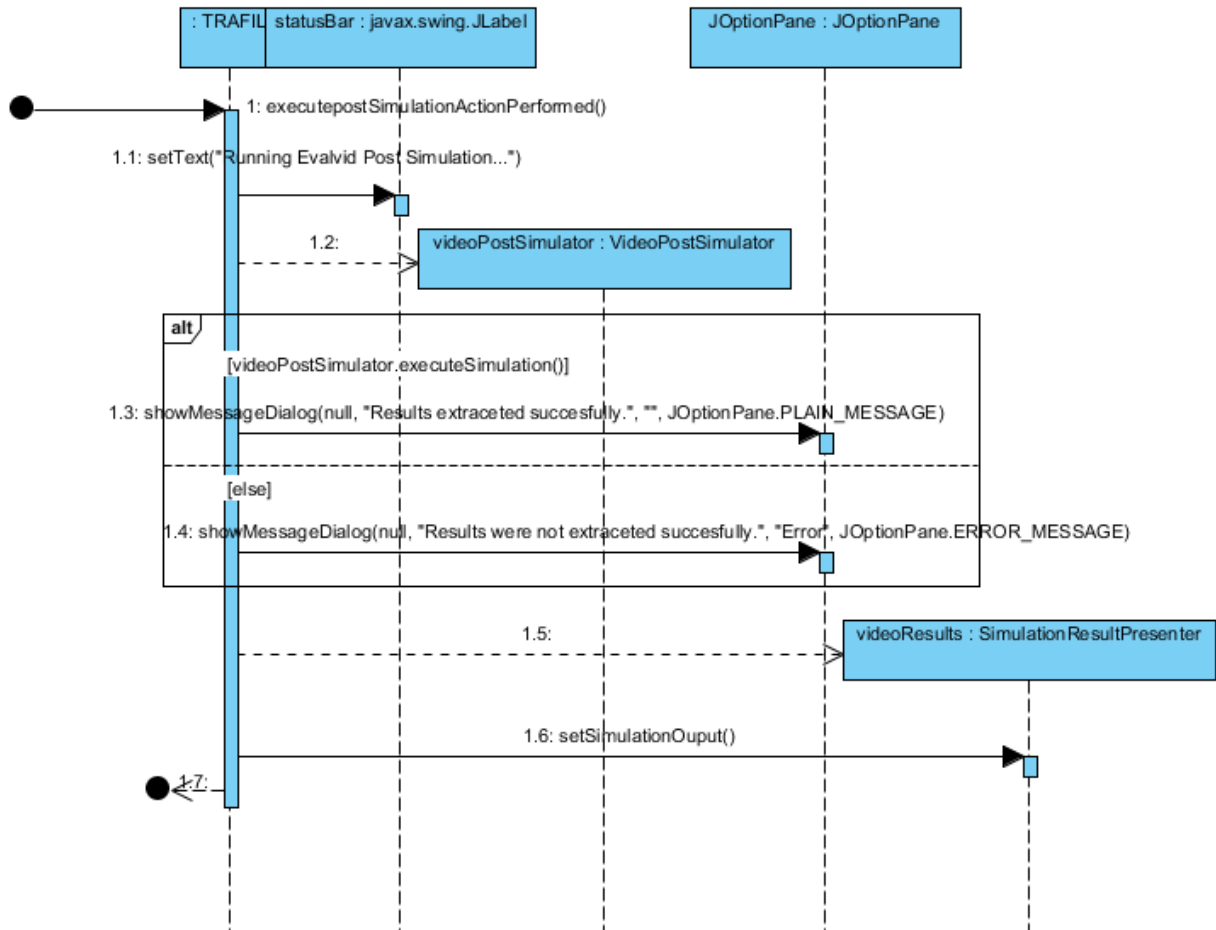
Τα αποτελέσματα της προσομοίωσης παρουσιάζονται στον χρήστη μέσω της κλάσης **SimulationResultPresenter**. Η κλάση αυτή παρουσιάζει, επίσης, μια λίστα των αρχείων που έχουν παραχθεί από την προσομοίωση και επιτρέπει την αποθήκευση σε άλλη τοποθεσία ή διαγραφή τους.

Μετά την εκτέλεση της προσομοίωσης έρχεται η φάση *post-simulation*, στην οποία γίνεται επεξεργασία των αποτελεσμάτων. Ο χρήστης επιλέγει τα αρχεία που θέλει να επεξεργαστεί και η κλάση **VideoPostSimulator** αναλαμβάνει αυτή την εργασία. Για το σκοπό αυτό χρησιμοποιεί, όπως και η κλάση **VideoSimulator**, εξωτερικά εργαλεία, ενώ ο τρόπος επικοινωνίας με αυτά είναι ο ίδιος.

sd preSimulation_Simulation

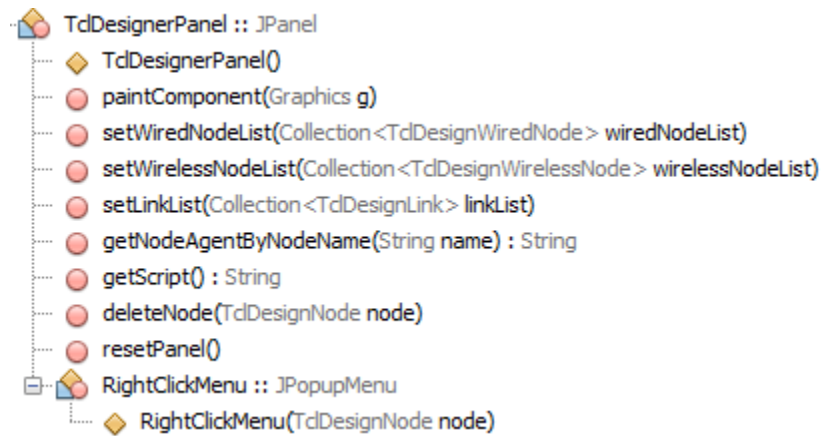


Εικόνα 39 Ενέργειες κατά τη φάση pre-simulation & simulation

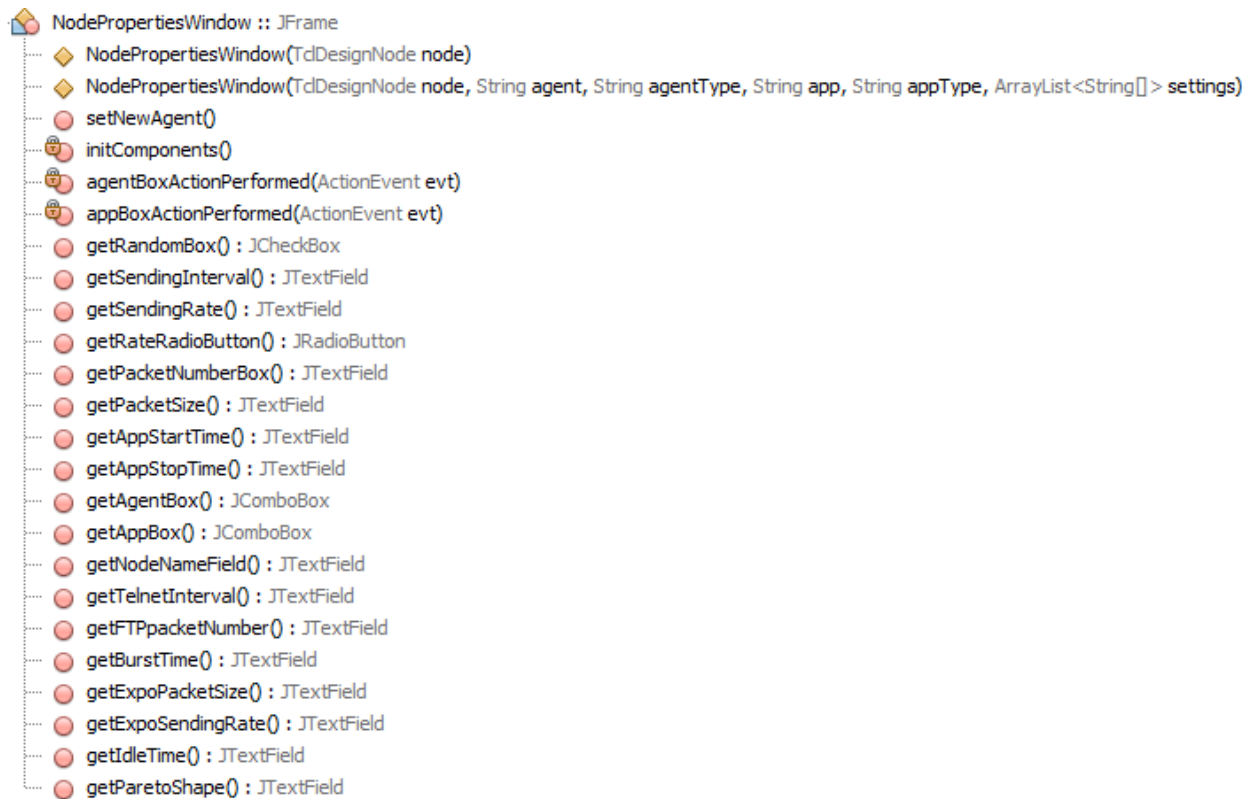


Εικόνα 40 Ενέργειες κατά τη φάση post-simulation

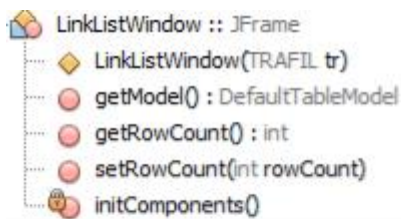
Οι επόμενες κλάσεις αποτελούν την Μονάδα Σχεδιασμού Προσομοίωσης. Κύρια κλάση είναι η TcdDesignerPanel, η οποία συγκεντρώνει τη λογική της μονάδας. Επεκτείνει την κλάση JPanel της Java, εμφανίζοντας το πλαίσιο προσομοίωσης. Επομένως η κλάση ανήκει στο Presentation Layer. Ανήκει, όμως, και στο Application Layer της εφαρμογής, καθώς αναλαμβάνει επίσης κομμάτι του Script Layer της μονάδας. Συγκεκριμένα, σε κάθε στιγμή η μέθοδος getScript() επιστρέφει το σχεδιασμένο δίκτυο σε μορφή OTcl, παραλείποντας μόνο κάποιες δευτερεύοντες παραμέτρους. Επίσης, η κλάση υλοποιεί ένα μενού που εμφανίζεται με δεξί κλικ, ώστε ο χρήστης να μπορεί να βλέπει ποιες ενέργειες μπορεί να εκτελέσει σε κάθε κόμβο.



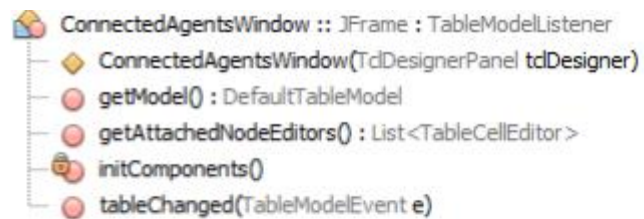
Εικόνα 41 Η κλάση `TcdDesignerPanel`



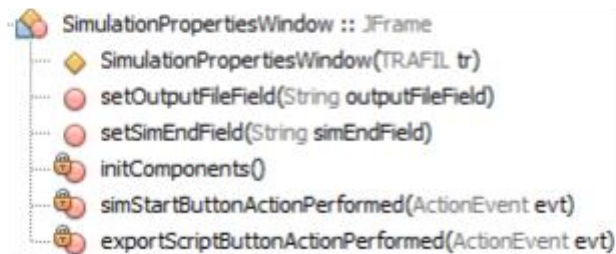
Εικόνα 42 Η κλάση `NodePropertiesWindow`



Η κλάση LinkListWindow



Η κλάση ConnectedAgentsWindow

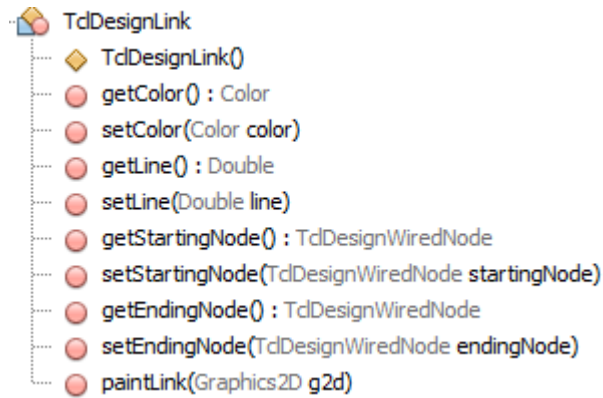


Η κλάση SimulationPropertiesWindow

Εικόνα 43 Οι κλάσεις LinkListWindow, ConnectedAgentsWindow και SimulationPropertiesWindow

Οι παραπάνω κλάσεις συμπληρώνουν το Presentation Layer της Μονάδας Σχεδιασμού Προσομοίωσης. Χρησιμοποιούνται για την παραγωγή μενού σε διάφορα στοιχεία της διεπαφής. Συγκεκριμένα, η κλάση NodePropertiesWindow εμφανίζει το μενού παραμέτρων του επιλεγμένου κόμβου, η κλάση LinkListWindow εμφανίζει έναν πίνακα όλων των συνδέσμων μαζί με τις παραμέτρους τους και η κλάση ConnectedAgentsWindow εμφανίζει αντίστοιχα έναν πίνακα που δείχνει ποιοι agents είναι συνδεδεμένοι μεταξύ τους. Τέλος η κλάση SimulationPropertiesWindow εμφανίζει ένα ξεχωριστό παράθυρο, όπου μπορούν να συμπληρωθούν οι δευτερεύουσες πληροφορίες ενός σεναρίου, όπως όνομα τελικού αρχείου και χρονοδιάγραμμα προσομοίωσης, ενώ υπάρχουν και οι επιλογές αποθήκευσης ή αποθήκευσης και εκτέλεσης του αρχείου.

Για την αποτελεσματική αναπαράσταση ενός δικτύου καθώς και των πολυάριθμων παραμέτρων των συστατικών του το TRAFIL έχει κωδικοποιήσει σε ξεχωριστές κλάσεις τα στοιχεία ενός δικτύου σε σενάριο προσομοίωσης NS-2. Συγκεκριμένα, έχουν δημιουργηθεί οι κλάσεις TclDesignLink, TclDesignNode, TclDesignWiredNode και TclDesignWirelessNode. Οι κλάσεις αυτές, καθώς και η σχέση τους με την TclDesignerPanel εμφανίζονται παρακάτω.

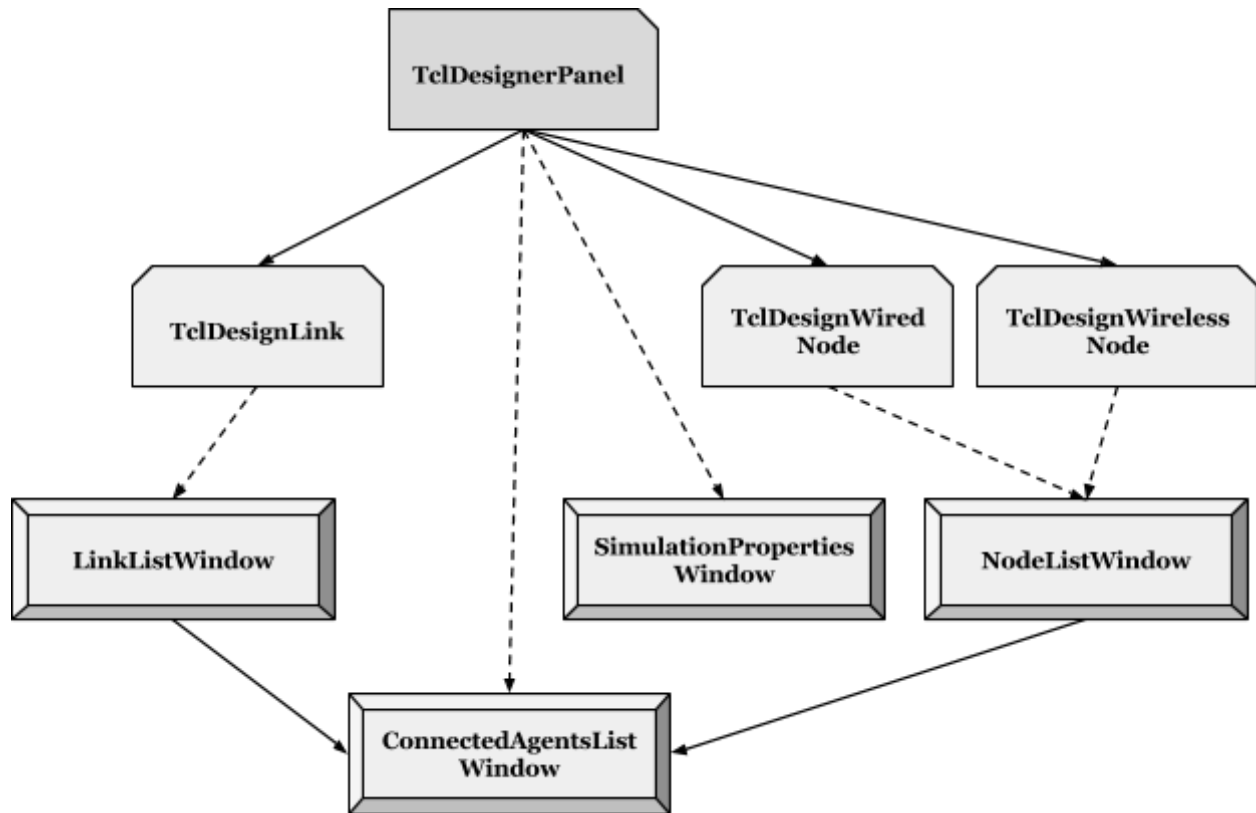


Εικόνα 44 Η κλάση TcdDesignLink



Εικόνα 45 Οι κλάσεις TcdDesignNode, TcdDesignWiredNode και TcdDesignWirelessNode

Η κλάση `TclDesignNode` είναι abstract κλάση, δηλαδή παρέχει κάποιες προδιαγραφές όσον αφορά τις μεθόδους και ιδιότητες που πρέπει να καλύπτουν όλες οι κλάσεις που την υλοποιούν και αφορούν όλους τους κόμβους ενός δικτύου. Οι κλάσεις `TclDesignWiredNode` και `TclDesignWirelessNode` υλοποιούν την `TclDesignNode`, παρέχοντας επίσης έναν επιπλέον βαθμό εξειδίκευσης, ο οποίος αφορά και το σχεδιαστικό κομμάτι. Η σχέση όλων των κλάσεων με την κύρια κλάση της μονάδας, δηλαδή την `TclDesignerPanel`, φαίνεται παρακάτω:



Εικόνα 46 Λειτουργική σχέση μεταξύ της κλάσης `TclDesignerPanel` και των βοηθητικών κλάσεών της

Στο σχήμα οι διακεκομμένες γραμμές δηλώνουν το άνοιγμα ενός καινούργιου παραθύρου, ενώ οι έντονες γραμμές την προγραμματιστική εξάρτηση. Παρατηρούμε ότι για την συμπλήρωση της λίστας των συνδεδεμένων agents χρησιμοποιούνται πληροφορίες τόσο από την κλάση `NodeListWindow` όσο και από την `LinkListWindow`. Αυτό γίνεται ώστε οι επιλογές που εμφανίζονται στο παράθυρο των συνδεδεμένων agents να είναι οι μόνες έγκυρες κάθε στιγμή. Αναλυτικότερα, εξετάζεται συνεχώς εάν δυο κόμβοι είναι συνδεδεμένοι, είτε μέσω συνδέσμου σε ενσύρματο δίκτυο, είτε ασύρματα σε ασύρματο δίκτυο. Έπειτα, αντλούνται πληροφορίες για τους υπάρχοντες agents ενός κόμβου ώστε να δοθούν ως επιλογή.

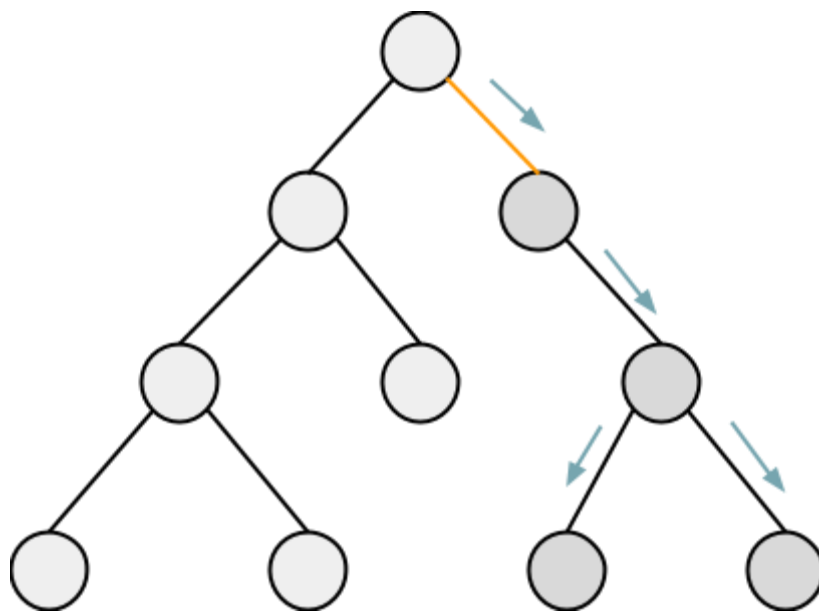
5.2.

Ανάλυση λειτουργίας TCL Design

Η διαδικασία λειτουργίας της μονάδας Σχεδιασμού Προσομοίωσης που θα αναλύσουμε περιγράφει τα δυο πρώτα επίπεδά της. Η καρτέλα TCL Design όπως είδαμε αποτελείται από διάφορα μέρη, το κυριότερο όμως είναι το πλαίσιο σχεδίασης. Αυτό υλοποιείται πλήρως από την κλάση `TCLDesignerPanel`, η οποία επεκτείνει την εγγενή κλάση της Java, `JPanel`. Έτσι δίνεται η δυνατότητα δυναμικής τοποθέτησης είτε στατικών είτε μη στατικών αντικειμένων πάνω στο πλαίσιο αυτό χωρίς να αλλάζει η θέση τους κάθε φορά που κινείται το παράθυρο. Επίσης παρέχει listeners γεγονότων στην περιοχή του πλαισίου ώστε κάθε χτύπημα του ποντικιού του χρήστη να αναγνωρίζεται και έπειτα να λειτουργεί ανάλογα με τις επιλογές που έχουν γίνει στην παλέτα.

Στην περίπτωση ενός ασύρματου κόμβου γίνεται αυτόματα συσχέτιση με οποιοδήποτε γειτονικό κόμβο. Στην περίπτωση ενός ενσύρματου κόμβου η διαδικασία εκτελείται μόνο εάν εγκατασταθεί σύνδεσμος σε αυτόν τον κόμβο. Η διαδικασία που εκτελείται όμως είναι παρόμοια και βασίζεται στην αναζήτηση δέντρων. Συγκεκριμένα, οι κόμβοι δημιουργούν ένα δέντρο με κορυφή τον κόμβο που τοποθετήθηκε ή συνδέθηκε μόλις.

Η προσθήκη των νέων διασυνδεδεμένων κόμβων γίνεται από την μέθοδο `addAdjacentNodes()` της κλάσης `TclDesignNode`, η οποία όπως αναφέρθηκε στην προηγούμενη ενότητα κωδικοποιεί όλες τις ιδιότητες και μεθόδους ενός οποιουδήποτε κόμβου. Η μέθοδος αυτή δέχεται σαν ορίσματα μορφής `ArrayList` δυο πίνακες. Ο πρώτος πίνακας περιέχει τον κόμβο-ρίζα του δέντρου καθώς και όλους τους ήδη γειτονικούς του κόμβους. Ο δεύτερος πίνακας περιέχει τους κόμβους που έχουν ήδη εξερευνηθεί στο δέντρο, ώστε να μην γίνονται πολλαπλές επισκέψεις στον ίδιο κόμβο. Η εξερεύνηση του δέντρου γίνεται σε μορφή `Depth-First Search`³⁵ (DFS), δηλαδή εξερευνούνται πρώτα όλοι οι άμεσοι γείτονες ενός κόμβου πριν προχωρήσουμε στους κόμβους του αμέσως επόμενου επιπέδου. Τέλος, μετά την αναζήτηση γίνεται ανανέωση όλων των στηλών του παραθύρου “`Connected Agents`” ώστε να εμφανιστούν οι νέες δυνατές επιλογές. Παρακάτω φαίνεται σχηματικά η ανακάλυψη των νέων διασυνδέσεων μεταξύ κόμβων.



Εικόνα 47 Αναζήτηση νέων διασυνδεδεμένων κόμβων κατά τη σύνδεση δυο κόμβων μεταξύ τους

Το αριστερά δέντρο απεικονίζει το ένα άκρο του δικτύου, το οποίο όταν συνδεθεί με ένα νέο υποδίκτυο μέσω ενός κόμβου του ξεκινά την διαδικασία εξερεύνησης νέων διασυνδεδεμένων κόμβων. Η ίδια διαδικασία εκτελείται και από την πλευρά του νέου κόμβου, καθώς και των κόμβων του υποδικτύου στο οποίο ήταν αποκλειστικά συνδεδεμένος. Η περίπτωση που εξετάζεται στη συγκεκριμένη εικόνα είναι η πιο γενική, αφού σε αυτή την περίπτωση ο νέος κόμβος είναι ήδη συνδεδεμένος με μερικούς ακόμη κόμβους, χωρίς όμως να έχει συνδεθεί με κάποιον τρόπο με τον κόμβο-ρίζα.

Επιπλέον, κατά την τοποθέτηση ενός καινούργιου κόμβου σημειώνεται η τοποθεσία του σε σχέση με το πλαίσιο σχεδιασμού ώστε να αποθηκευτεί στο αρχείο σεναρίου που θα δημιουργηθεί. Η αποθήκευση αυτή γίνεται μέσω σχολίων της γλώσσας TCL, τα οποία αναγνωρίζονται από το TRAFIL κατά την λειτουργία ανάγνωσης αρχείου TCL. Γι' αυτό το λόγο η λειτουργία αυτή προτείνεται μόνο για αρχεία δημιουργημένα από το TRAFIL, αφού μόνο αυτά περιέχουν την συγκεκριμένη κωδικοποίηση.

6. Υλοποίηση

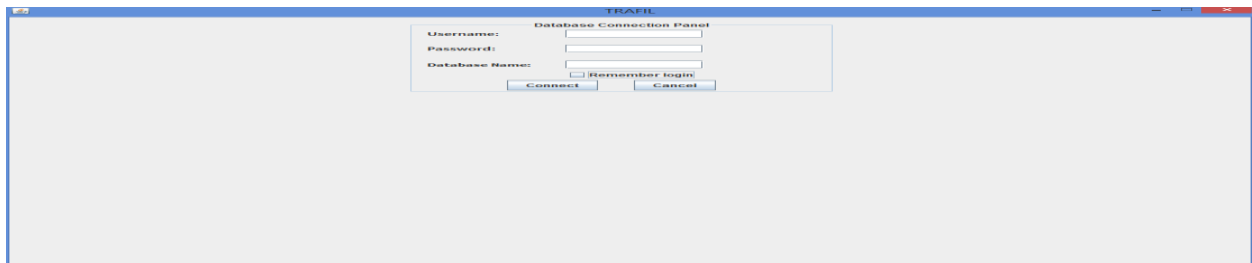
Σε αυτή την ενότητα θα αναφέρουμε παραδείγματα χρήσης του TRAFIL. Δίνονται παραδείγματα εκτελέσεων διαφόρων λειτουργιών βήμα-βήμα, ενώ στο τέλος της ενότητας υπάρχει οδηγός εγκατάστασης του εργαλείου και των προαπαιτούμενών του.

6.1. Παραδείγματα χρήσης TRAFIL

6.1.1. Είσοδος στο TRAFIL

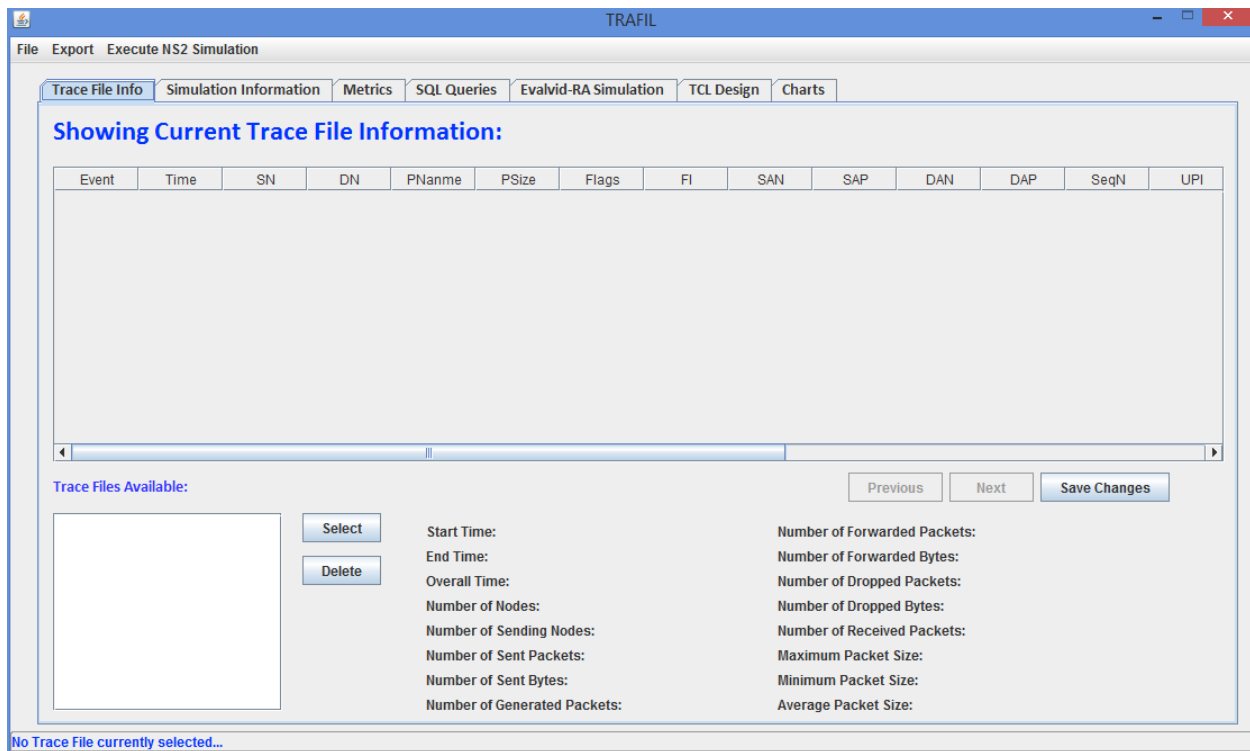
Το TRAFIL χρησιμοποιεί μια εξωτερική βάση δεδομένων τύπου MySQL για την αποθήκευση των δεδομένων των trace file. Γι' αυτό το λόγο το πρώτο πράγμα που ζητείται από το χρήστη όταν εκτελεστεί το εργαλείο είναι τα στοιχεία της βάσης δεδομένων στην οποία θα συνδεθεί. Τα στοιχεία αυτά είναι το όνομα χρήστη, ο αντίστοιχος κωδικός, καθώς και το όνομα της βάσης δεδομένων που θα χρησιμοποιήσει. Σημειώνεται ότι η βάση αυτή θα πρέπει να είναι κενή κατά την πρώτη εκτέλεση του TRAFIL, ώστε να μπορούν να δημιουργηθούν οι πίνακες του εργαλείου.

Εφ' όσον ο χρήστης το επιθυμεί, μπορεί να αποθηκεύσει τα στοιχεία αυτά στο TRAFIL, έτσι ώστε να μην χρειάζεται να τα πληκτρολογεί ξανά σε επόμενη εκτέλεση. Η επιλογή αυτή γίνεται μέσω ενός checkbox. Η οθόνη σύνδεσης φαίνεται συνολικά στην παρακάτω εικόνα.



Εικόνα 48 Οθόνη εισόδου στο TRAFIL

Σε περίπτωση επιτυχημένης σύνδεσης ο χρήστης κατευθύνεται στην κεντρική οθόνη του TRAFIL, η οποία είναι η εξής:



Εικόνα 49 Κύρια οθόνη του TRAFIL

Όπως παρατηρούμε στην παραπάνω εικόνα, δεν υπάρχει φορτωμένο κάποιο trace file. Αφού φορτώσουμε κάποιο trace file ή εφόσον υπάρχουν trace files στην βάση δεδομένων του TRAFIL θα εμφανίζονται ως λίστα στο κάτω αριστερά μέρος του παραθύρου. Επίσης, εφόσον φορτωθεί κάποιο trace file, ο πίνακας θα εμφανίσει τα περιεχόμενά του στον κεντρικό πίνακα του TRAFIL καθώς και τις γενικές πληροφορίες του στο κάτω μέρος. Επειδή το TRAFIL φορτώνει μόνο τις πρώτες 10,000 γραμμές ενός trace file, με τα κουμπιά “Previous” και “Next” μπορούμε να περιηγηθούμε σε περισσότερες γραμμές. Ο χρήστης μπορεί να αλλάξει τα δεδομένα του trace file και να αποθηκεύσει τις αλλαγές στη βάση μέσω του κουμπιού “Save Changes”. Τέλος, μπορεί να φορτώσει ή να διαγράψει ένα trace file από τη βάση δεδομένων μέσω των κουμπιών “Select” και “Delete”.

Στο πάνω μέρος του παραθύρου υπάρχει ένα οριζόντιο μενού επιλογών με τρεις επιλογές:

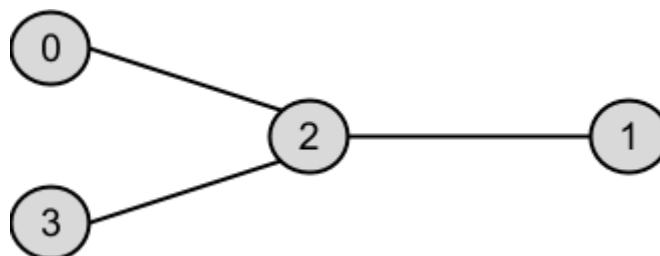
- *File*: Δίνει τη δυνατότητα ο χρήστης να φορτώσει ένα νέο trace file, να φορτώσει ένα TCL script file δημιουργημένο από το TRAFIL ή να τερματίσει την εφαρμογή.

- *Export*: Δίνει την δυνατότητα στο χρήστη να εξαγάγει πληροφορία από το TRAFIL σε διάφορες μορφές. Μπορεί να εξαγάγει την πληροφορία σε μορφή επεξεργασμένη από το TRAFIL είτε σε απλή μορφή .txt, είτε σε αρχείο .xls, ώστε να είναι αναγνωρίσιμη από spreadsheet εφαρμογές όπως το Microsoft Excel. Μπορεί επίσης, να εξαγάγει τις γραφικές παραστάσεις που έχει δημιουργήσει ο χρήστης σε .jpeg μορφή αλλά και γενικές πληροφορίες για την προσομοίωση ή συγκεκριμένες μετρήσεις που παρήχθησαν σε .txt αρχείο.
- *Execute NS2 Simulation*: Η επιλογή αυτή δίνει στο χρήστη τη δυνατότητα να εκτελέσει μια προσομοίωση παρέχοντας ο ίδιος το σενάριο. Τα αποτελέσματα παρουσιάζονται στον χρήστη και φορτώνονται στην βάση δεδομένων, όπως περιγράφηκε στην ενότητα 4.2.6.

Τέλος, παρατηρούμε τις υπόλοιπες καρτέλες που απαρτίζουν το TRAFIL: Simulation Information, Metrics, SQL Queries, Evalvid-RA Simulations, TCL Design και Charts.

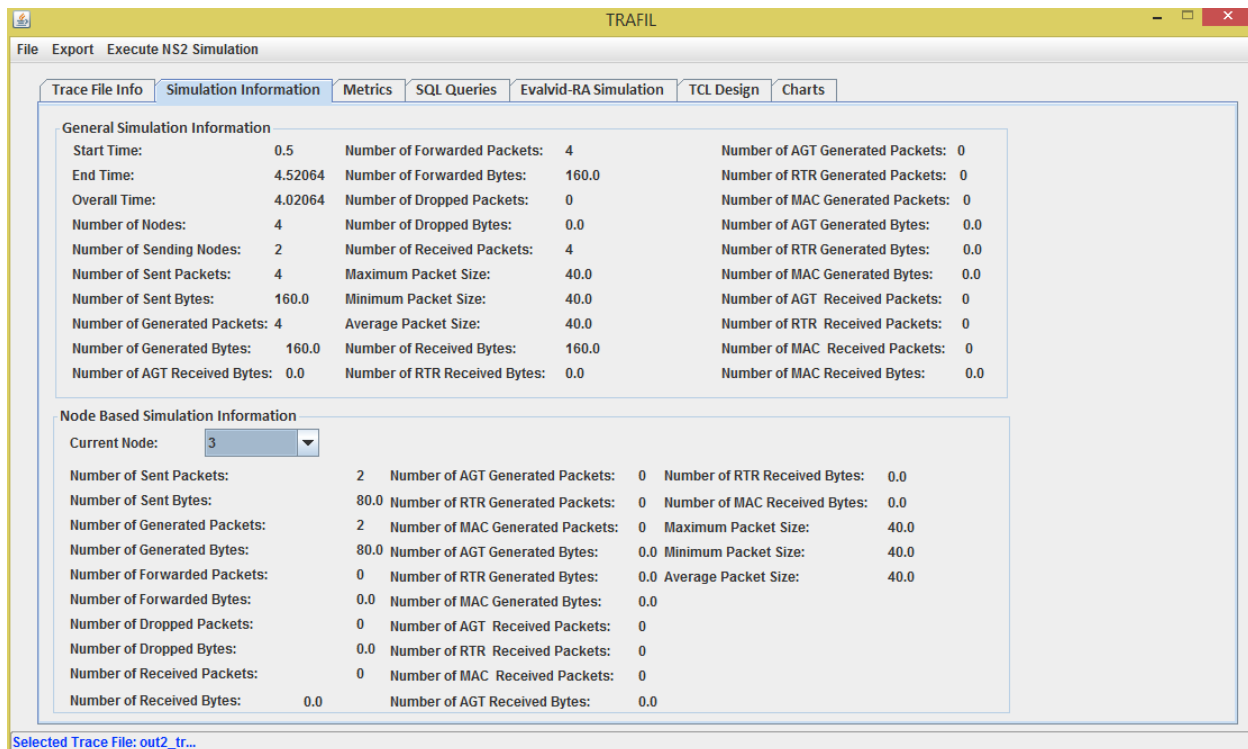
6.1.2. Γενικές πληροφορίες της προσομοίωσης

Όπως είδαμε, η φόρτωση ενός νέου trace file γίνεται από την επιλογή “Open Trace File” του υπομενού “File” του μενού επιλογών στο πάνω μέρος του TRAFIL. Αφού φορτωθεί το trace file, συμπληρώνονται τόσο τα πεδία και ο πίνακας της κεντρικής καρτέλας του TRAFIL, όσο και η καρτέλα “Simulation Information”, στην οποία υπάρχουν οι γενικές πληροφορίες της προσομοίωσης. Οι πληροφορίες αυτές αφορούν τόσο την προσομοίωση συνολικά, όσο και κάθε κόμβο ξεχωριστά. Στα παραδείγματα που θα εξετάσουμε παρακάτω χρησιμοποιείται ένα απλό δίκτυο με τέσσερις κόμβους, δυο εκ των οποίων είναι κόμβοι αφετηρίας και ένας κόμβος-καταβόθρα. Το δίκτυο έχει την εξής δομή:



Εικόνα 50 Παράδειγμα απλού δικτύου τεσσάρων κόμβων

Στην παρακάτω εικόνα βλέπουμε τις πληροφορίες ενός απλού σεναρίου ενσύρματου δικτύου με τέσσερις κόμβους:

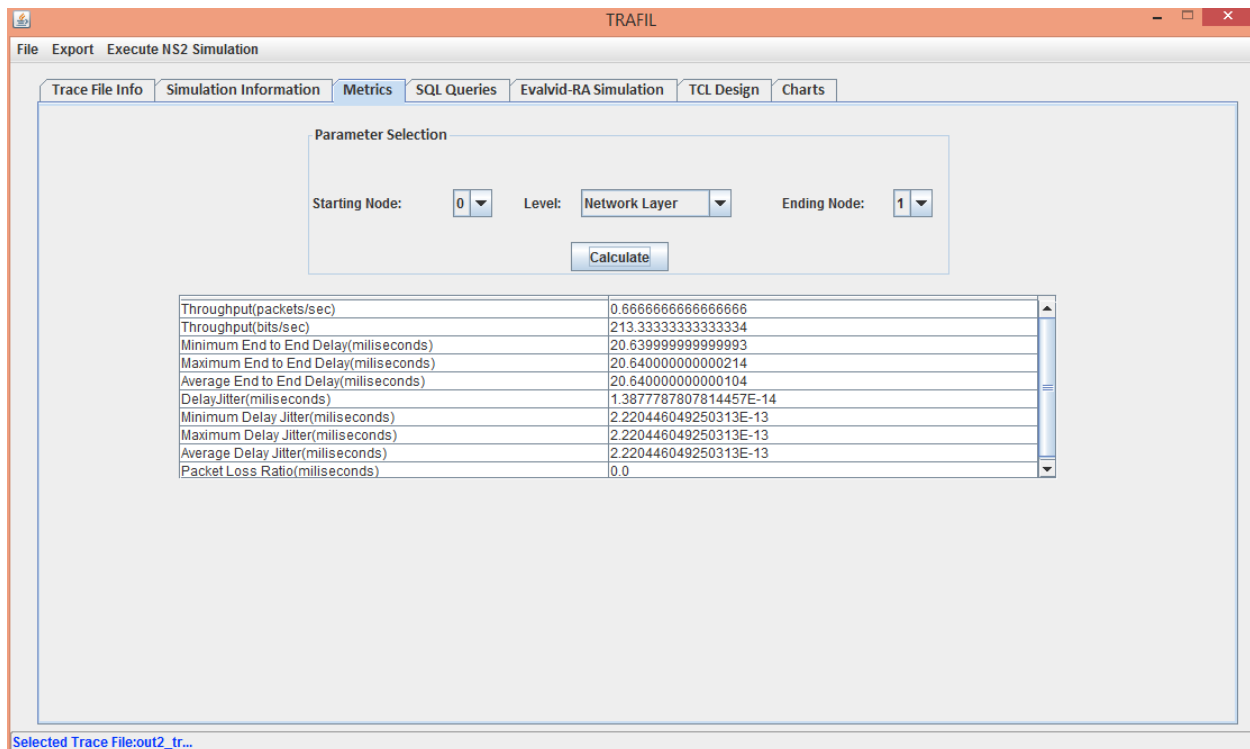


Εικόνα 51 Οθόνη γενικών πληροφοριών

6.1.3.

Υπολογισμός μετρήσεων

Δεδομένου ότι έχει επιλεγθεί ένα trace file (στην περίπτωσή μας το ίδιο trace file με την προηγούμενη υποενότητα) οι υπόλοιπες καρτέλες του TRAFIL συσχετίζονται με αυτό. Επομένως, τόσο η καρτέλα “Simulation Information” που είδαμε νωρίτερα, όσο και η “Metrics” που θα εξετάσουμε τώρα επιτρέπουν διάφορες ενέργειες πάνω στα δεδομένα του επιλεγμένου trace file. Η καρτέλα “Metrics” παρέχει επιλογές για διάφορες μετρήσεις μεταξύ δυο κόμβων, ανεξαρτήτως δικτύου. Οι μετρήσεις μπορούν να γίνουν σε δυο επίπεδα επικοινωνίας μεταξύ των κόμβων, είτε στο χαμηλό επίπεδο συνδέσμου (Link Layer), είτε στο επίπεδο δικτύου (Network Layer). Οι τιμές των μετρήσεων αυτών δείχνουν σε μεγάλο βαθμό την ποιότητα του δικτύου που προσομοιώνεται, γι’ αυτό και αποκαλούνται μετρήσεις QoS (Quality of Service). Στην εικόνα βλέπουμε τις μετρήσεις που έγιναν στο δεύτερο επίπεδο μεταξύ ενός από τους δυο κόμβους αφετηρίας του δικτύου μας και του κόμβου-καταβόθρα.



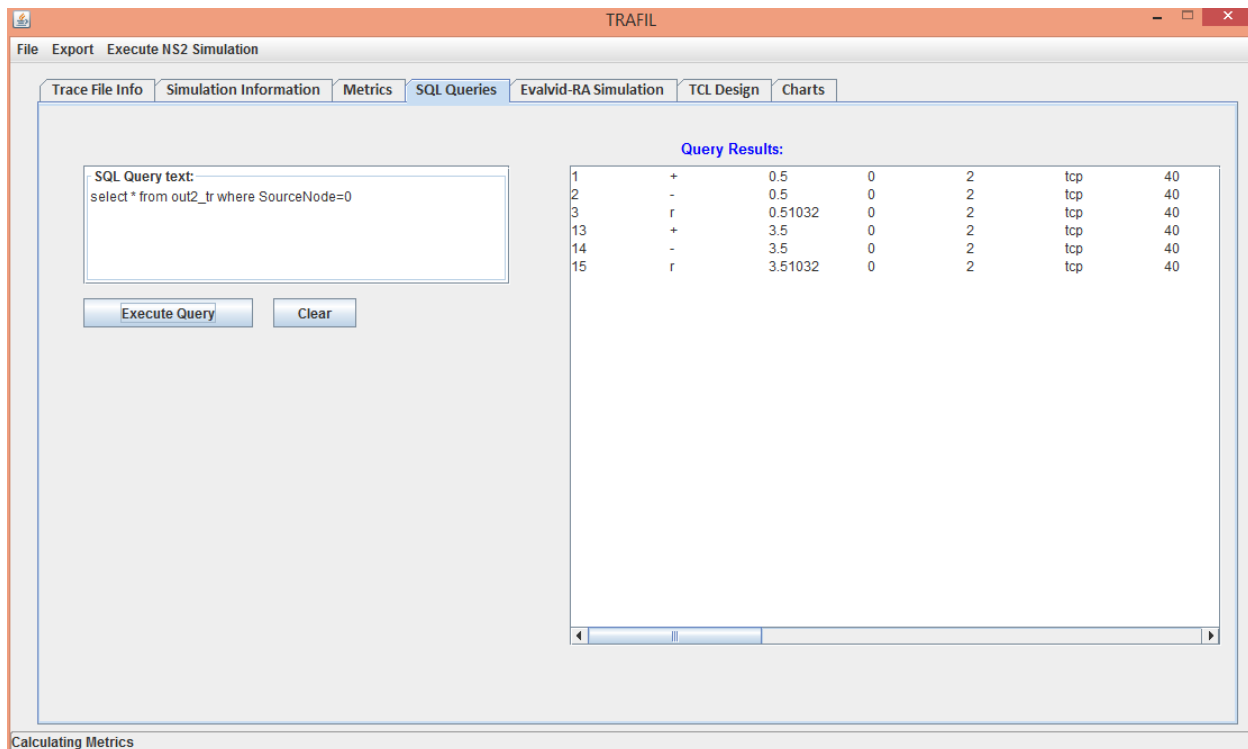
Εικόνα 52 Οθόνη μετρήσεων

6.1.4.

Εκτέλεση ερωτημάτων προς την βάση δεδομένων

Η επόμενη καρτέλα δίνει την δυνατότητα εκτέλεσης ερωτημάτων SQL προς την βάση δεδομένων. Τα μόνα ερωτήματα που γίνονται δεκτά από το πρόγραμμα είναι ερωτήματα ανάγνωσης πληροφοριών. Δεν επιτρέπονται ερωτήματα που επηρεάζουν την δομή ή τα περιεχόμενά της. Σε περίπτωση που ο χρήστης στείλει μη έγκυρο ερώτημα θα εμφανιστεί μήνυμα λάθους.

Φυσικά, ο χρήστης θα πρέπει να γνωρίζει την δομή του πίνακα στη βάση δεδομένων πριν μπορέσει να σχηματίσει ένα έγκυρο ερώτημα. Η δομή του πίνακα μπορεί να παρατεθεί με την εκτέλεση της εντολής “desc <όνομα_πίνακα>”. Τα ονόματα των πινάκων μπορούν να βρεθούν στην αρχική οθόνη του TRAFIL, στο πεδίο όπου αναφέρονται τα trace file της βάσης δεδομένων. Επίσης, τα πεδία του πίνακα στη βάση δεδομένων είναι ταυτόσημα κάθε φορά με τα πεδία του πίνακα στην κεντρική καρτέλα του TRAFIL, όπου αναγράφονται τα περιεχόμενα του trace file. Τέλος, υπάρχει ένα ανώτατο όριο στον αριθμό στοιχείων που μπορούν να επιστραφούν από την βάση δεδομένων κι επομένως ο χρήστης πρέπει να προσέξει τον όγκο δεδομένων που ζητάει. Παρακάτω βλέπουμε την καρτέλα εκτελέσεων ερωτημάτων στην βάση δεδομένων καθώς και ένα παράδειγμα ερωτήματος.



Εικόνα 53 Οθόνη ερωτημάτων προς τη βάση δεδομένων

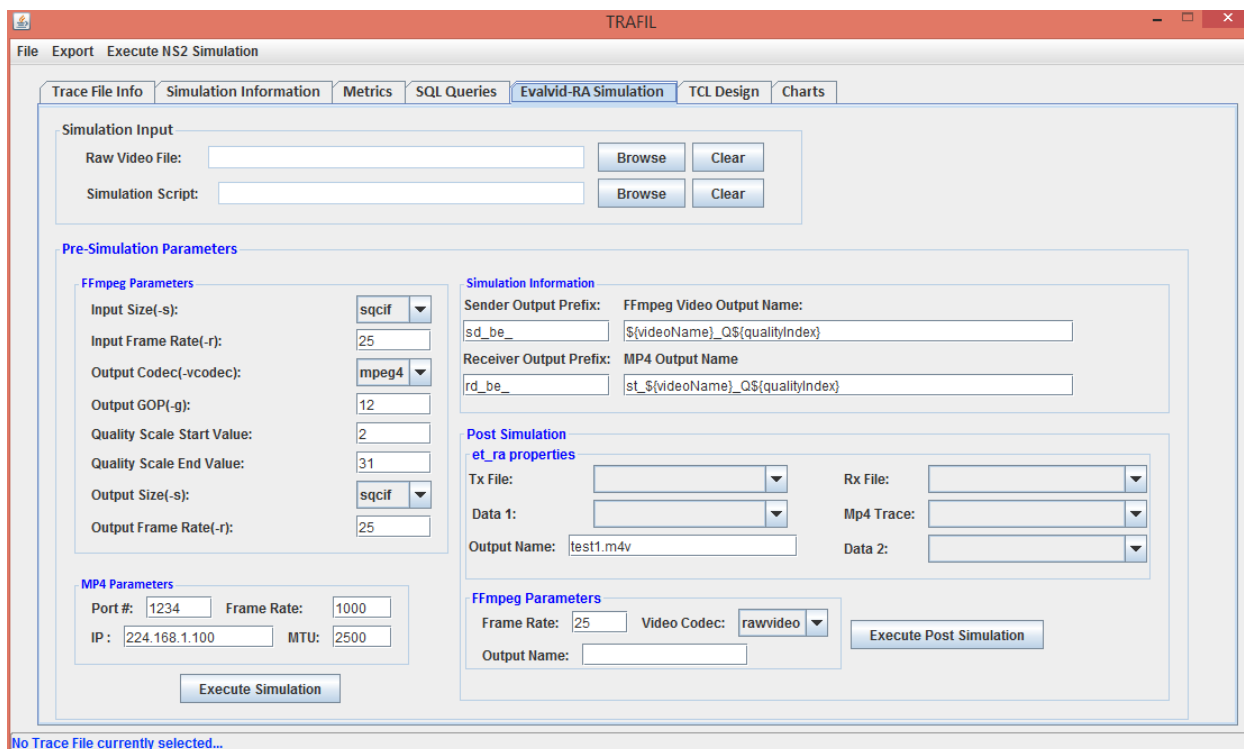
6.1.5.

Εκτέλεση προσομοιώσεων

Η διαδικασία προσομοίωσης είναι ένα μοναδικό χαρακτηριστικό του TRAFIL, το οποίο, όπως είδαμε σε προηγούμενες ενότητες, έχει τρεις πτυχές: Την απλή προσομοίωση με έτοιμο από το χρήστη σενάριο, την προσομοίωση μετά από δημιουργία σεναρίου μέσω της μονάδας σχεδίασης του TRAFIL και τέλος την προσομοίωση μετάδοσης βίντεο με χρήση της προσθήκης Evalvid-RA για NS-2.

Για την εκτέλεση απλής προσομοίωσης ο χρήστης μπορεί να καθορίσει το αρχείο σεναρίου μέσα από την επιλογή “Execute NS-2 Simulation” από το οριζόντιο μενού στο πάνω μέρος του TRAFIL. Σε περίπτωση που υπάρχουν ήδη αρχεία αποτελεσμάτων προσομοίωσης (trace files) από προηγούμενη προσομοίωση, το TRAFIL θα ρωτήσει τον χρήστη εάν θέλει να διαγραφούν, ώστε να μην τα μπερδέψει με επόμενα. Έπειτα εκτελείται η προσομοίωση και, σε περίπτωση επιτυχίας, το αποτέλεσμα φορτώνεται αυτόματα στην βάση δεδομένων του TRAFIL. Σε περίπτωση παραγωγής περισσότερων του ενός αρχείων εμφανίζεται παράθυρο επιλογής, έτσι ώστε ο χρήστης να επιλέξει ποιο trace file θέλει να επεξεργαστεί.

Η διαδικασία προσομοίωσης μετάδοσης βίντεο με χρήση του Evalvid-RA είναι πιο πολύπλοκη και γι' αυτό το λόγο δίνεται σε ξεχωριστή καρτέλα με την ονομασία “Evalvid-RA Simulation”. Παρακάτω φαίνεται η καρτέλα αυτή.



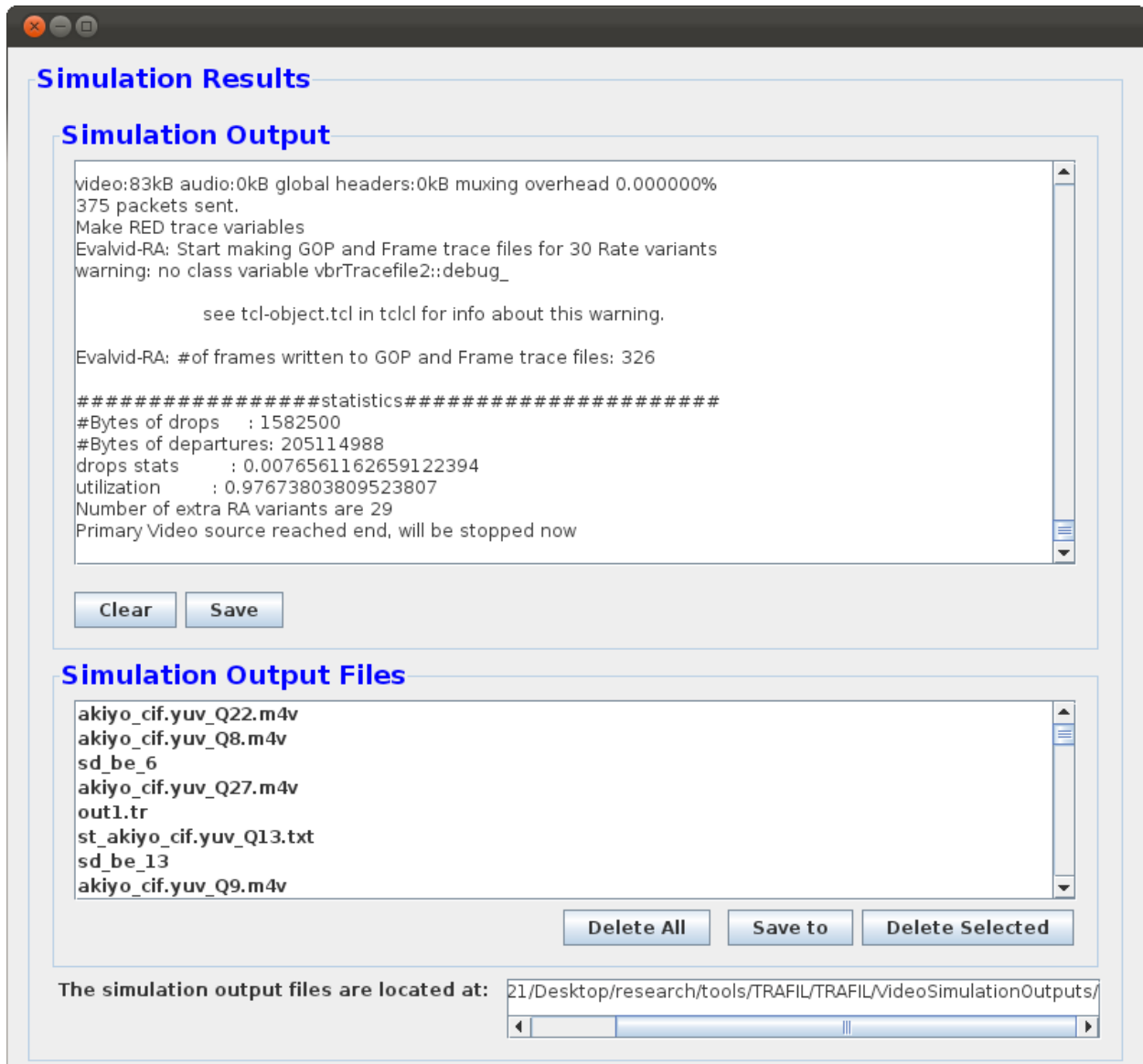
Εικόνα 54 Η καρτέλα Evalvid-RA Simulation

Όπως έχουμε ήδη αναφέρει η διαδικασία χωρίζεται σε δυο φάσεις, την φάση pre-simulation και την φάση post-simulation. Στην πρώτη φάση ο χρήστης χρειάζεται μόνο να καθορίσει το αρχείο βίντεο που θα μεταδοθεί (σε οποιαδήποτε μορφή) καθώς και το σενάριο της προσομοίωσης. Τα πεδία αυτά βρίσκονται στην κορυφή της καρτέλας. Εάν επιθυμεί, μπορεί να καθορίσει τις παραμέτρους του FFmpeg και του MP4, των βοηθητικών δηλαδή προγραμμάτων της διαδικασίας. Οι παράμετροι αυτοί είναι στο πλαίσιο “Pre-Simulation Parameters”.

Στην προσομοίωση με χρήση Evalvid-RA, όπως είδαμε στην ενότητα 4.2.6, το αρχείο βίντεο κωδικοποιείται μέσω του εργαλείου MP4 σε trace files. Τα ονόματα των trace files έχουν ιδιαίτερη σημασία, αφού καθορίζονται στο σενάριο προσομοίωσης της μετάδοσης. Για αυτό το λόγο, στο πλαίσιο “Simulation Information” δίνεται η δυνατότητα να αλλάξει ο χρήστης την ονομασία των αρχείων που παράγουν τα προγράμματα FFmpeg και MP4, ώστε να μπορεί να τα εντοπίσει ο προσομοιωτής. Ο μόνος περιορισμός στις τιμές που παίρνουν τα πεδία αυτά είναι να περιέχουν τα πεδία $\{\text{videoName}\}$ και $_Q\{\text{qualityIndex}\}$ και το τελευταίο να είναι στο τέλος του πεδίου. Σε κάθε περίπτωση,

το TRAFIL έχει ήδη συμπληρωμένα τα πεδία αυτά με έγκυρες τιμές, επομένως συνίσταται να μην αλλάζονται από τον χρήστη χωρίς ιδιαίτερο λόγο.

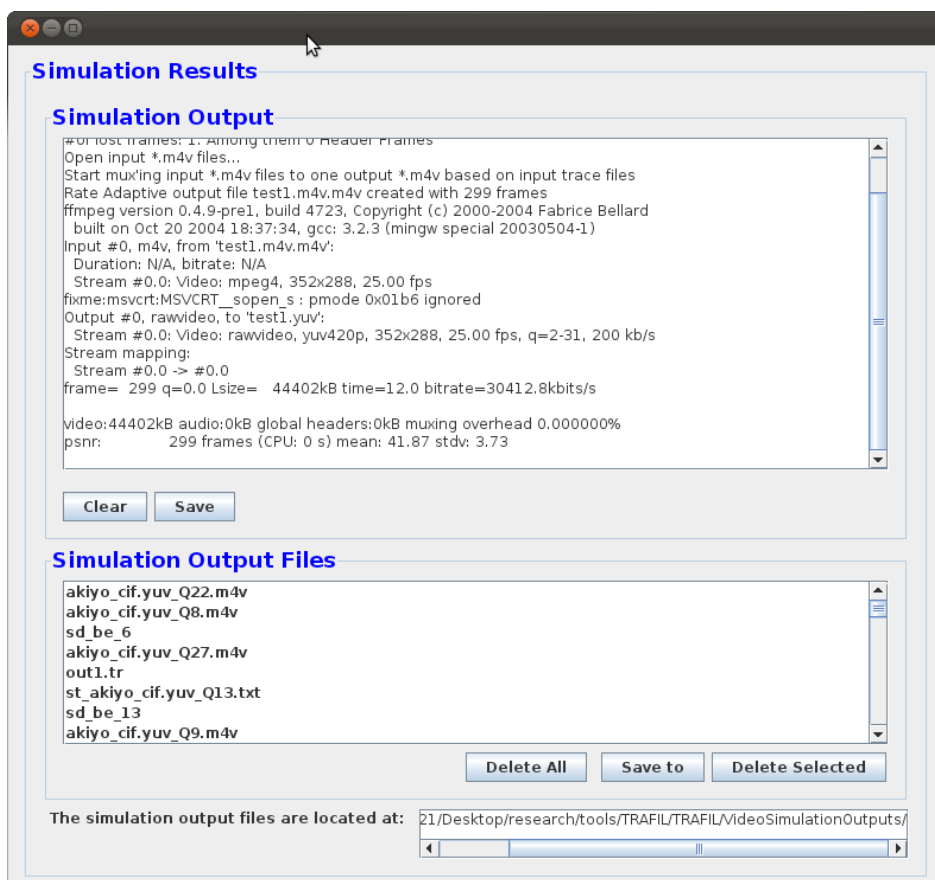
Η έξοδος μιας εκτέλεσης προσομοίωσης μετάδοσης βίντεο φαίνεται στην παρακάτω εικόνα:



Εικόνα 55 Οθόνη αποτελέσματος προσομοίωσης Evalvid-RA

Όπως φαίνεται, όλα τα αρχεία που παράγονται κατά τη διάρκεια της προσομοίωσης καθώς και η έξοδος όλων των προγραμμάτων που χρησιμοποιούνται παρουσιάζονται αναλυτικά στον χρήστη. Επίσης, δίνεται η δυνατότητα αποθήκευσης της εξόδου σε αρχείο .txt αλλά και η διαχείριση των αρχείων εξόδου, μεταφέροντάς τα σε άλλο φάκελο ή διαγράφοντάς τα, ώστε να μην κατακρατούν χώρο στους φακέλους του TRAFIL.

Για την φάση post-simulation ο χρήστης χρειάζεται μόνο να επιλέξει τα κατάλληλα αρχεία τα οποία απαιτεί το et_ra στο πλαίσιο “Post Simulation”. Επίσης, μπορεί να ορίσει τις παραμέτρους του FFmpeg, ώστε να δημιουργηθεί το ανακατασκευασμένο βίντεο στην αρχική του μορφή και να υπολογίσουμε έπειτα το PSNR. Έπειτα, γίνεται η φάση του post-simulation και αποθηκεύονται τα νέα αρχεία στον ίδιο ειδικό φάκελο με τα προηγούμενα. Το μονοπάτι αποθήκευσης εμφανίζεται και τις δυο φορές στα αποτελέσματα των διαδικασιών. Η επόμενη εικόνα δείχνει αποτέλεσμα εκτέλεσης της φάσης post-simulation στο ίδιο παράδειγμα με πριν.

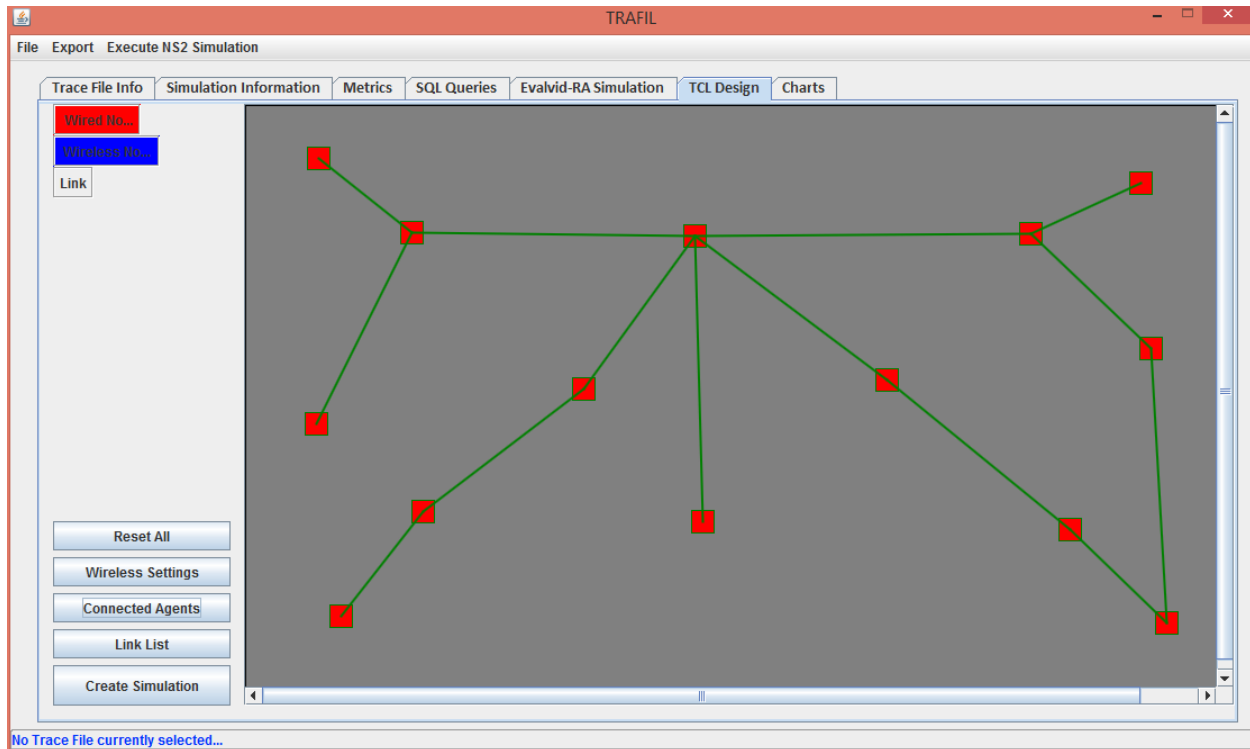


Εικόνα 56 Οθόνη αποτελέσματος φάσης post-simulation του Evalvid-RA

6.1.6.

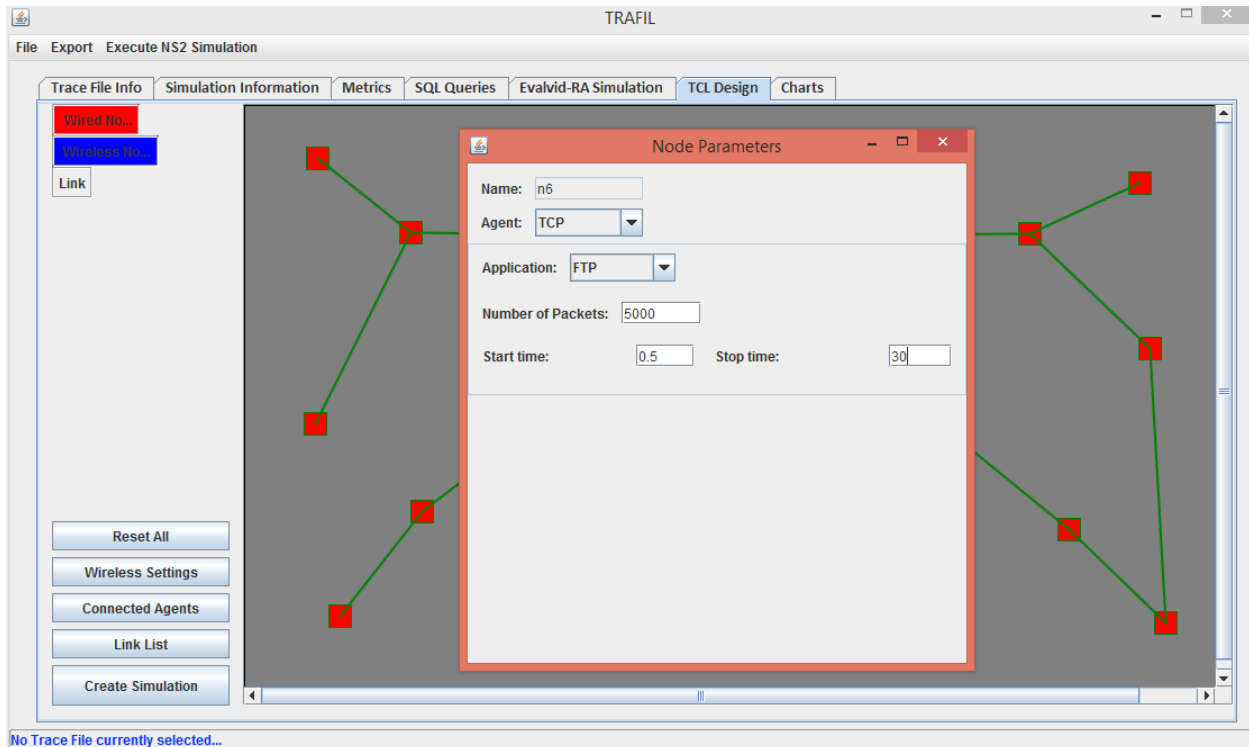
Σχεδίαση προσομοιώσεων

Η σχεδίαση σεναρίων προσομοιώσεων για NS-2 γίνεται στην καρτέλα “TCL Design” του TRAFIL, η οποία φαίνεται παρακάτω.



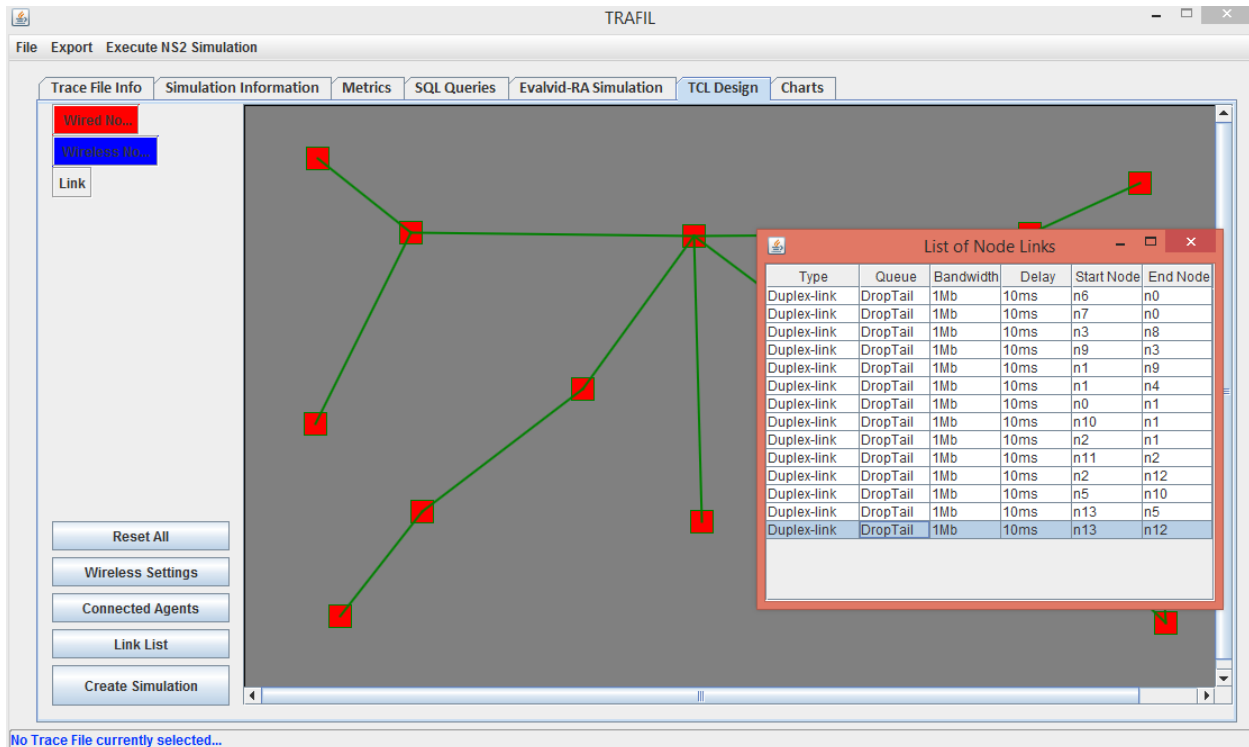
Εικόνα 57 Η καρτέλα TCL Design του TRAFIL, με σχηματισμένο ενσύρματο δίκτυο

Στο παράδειγμα που φαίνεται έχει σχεδιαστεί ένα ενσύρματο δίκτυο. Για την σχεδίαση ενός δικτύου ο χρήστης μπορεί να εναποθέσει πολλαπλούς κόμβους, αφού επιλέξει τον τύπο κόμβου που επιθυμεί από την παλέτα στο αριστερό μέρος. Οι ενσύρματοι κόμβοι συμβολίζονται με κόκκινα τετράγωνα, ενώ οι ασύρματοι με μπλε κύκλους. Αφού τοποθετηθεί ένας κόμβος είναι πάντα δυνατόν να αλλάξει η θέση του με απλό τράβηγμα. Επίσης, μπορούν να καθοριστούν οι παράμετροί του εμφανίζοντας το σχετικό μενού με διπλό κλικ στον κόμβο. Το μενού παραμέτρων ενός κόμβου είναι ξεχωριστό παράθυρο ενώ επιτρέπονται πολλά μενού ταυτόχρονα, ώστε ο χρήστης να μπορεί να αλλάξει γρήγορα παραμέτρους. Παράδειγμα μενού παραμέτρων για κόμβο με TCP agent και FTP application φαίνεται στην επόμενη εικόνα.

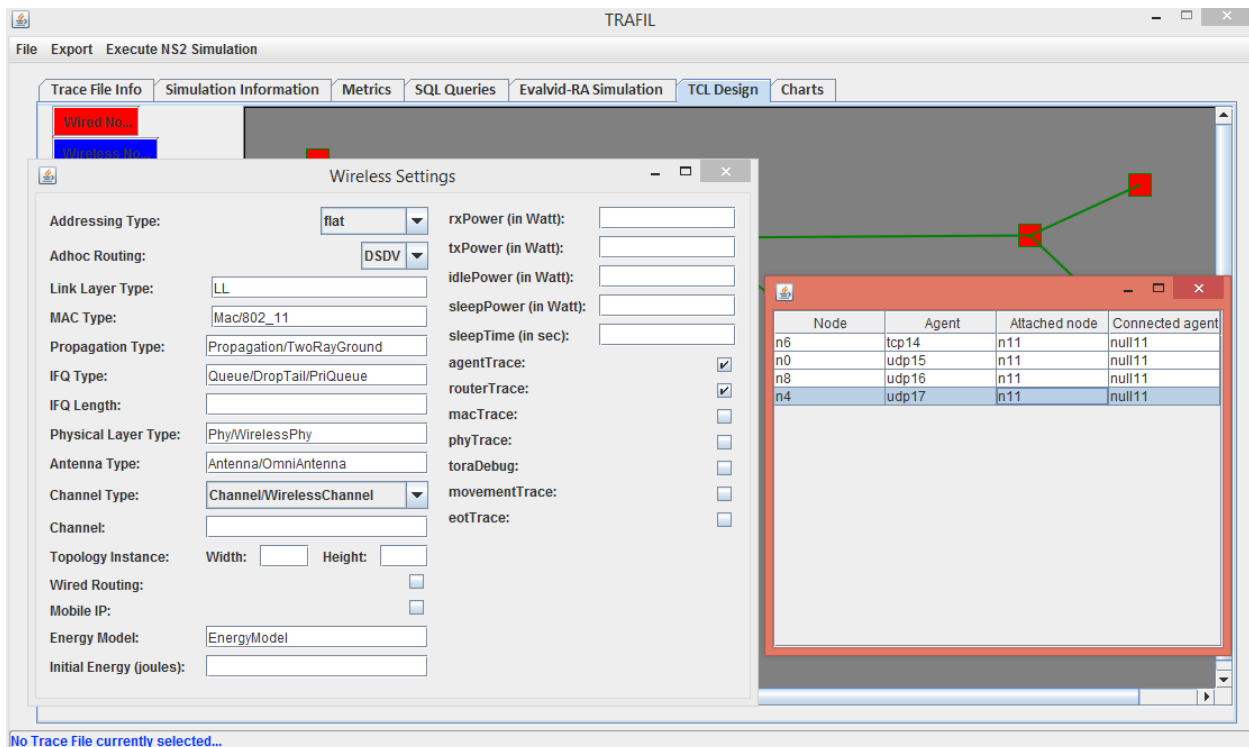


Εικόνα 58 Παράμετροι κόμβου

Η σχεδίαση των συνδέσμων (links) του δικτύου γίνεται επιλέγοντας το “Link” από την παλέτα και κλικάροντας στους δυο κόμβους που θέλουμε να συνδεθούν. Οι σύνδεσμοι συμβολίζονται με πράσινη γραμμή. Κάθε φορά που εγκαθίσταται ένας νέος σύνδεσμος σημειώνεται η νέα σύνδεση σε ένα ξεχωριστό μενού, το οποίο μπορεί να εμφανιστεί επιλέγοντας το κουμπί “Link List”. Εκεί καταγράφονται όλοι οι σύνδεσμοι, μαζί με τις ιδιότητές τους, ενώ μπορούν να γίνουν αλλαγές στους τύπους τους. Όλοι οι σύνδεσμοι έχουν κάποιες προκαθορισμένες τιμές, ώστε να μην χρειάζεται να τους συμπληρώσει ο χρήστης σε τετριμμένες περιπτώσεις. Παράδειγμα του παραθύρου βλέπουμε στην παρακάτω εικόνα.



Εικόνα 59 Οθόνη λίστας συνδέσεων



Εικόνα 60 Οθόνες συνδεδεμένων agents και επιλογών για ασύρματα δίκτυα

Τελευταία μενού που αφορούν τη σχεδίαση του δικτύου και την παραγωγή των παραμέτρων του είναι τα παράθυρα “Connected Agents” και “Wireless Settings”. Το πρώτο εμφανίζει όλους τους συνδεδεμένους agents, δηλαδή καθορίζουν τον αποστολέα και παραλήπτη κάθε μετάδοσης. Στην εικόνα φαίνεται ότι όλοι οι αποστολείς στέλνουν στον ίδιο παραλήπτη, ο οποίος έχει έναν Null agent. Η επιλογή του παραλήπτη γίνεται από αυτό το παράθυρο, το οποίο αρχικά έχει κενά τα πεδία “Attached node” και “Connected agent” κάθε αποστολέα. Επιλέγοντας τον παραλήπτη node αυτόματα επιλέγεται και ο παραλήπτης agent.

Το παράθυρο “Wireless Settings” εμφανίζει τις ρυθμίσεις που αναφέραμε στην ενότητα 3.3.5 και αφορούν στην συμπεριφορά των ασύρματων κόμβων του δικτύου. Δεν έχει καμία επίδραση σε περίπτωση ενσύρματου δικτύου. Περιέχει προκαθορισμένες τιμές σε όλα τα απαιτούμενα πεδία, ενώ αρκετά πεδία έχουν συγκεκριμένο αριθμό επιλογών ή checkbox.

Τέλος, το μενού “Create Simulation” δίνει την επιλογή αποθήκευσης του σεναρίου σε αρχείο .tcl και καθορίζει τον χρόνο τερματισμού του, ενώ το κουμπί “Reset All” διαγράφει όλη την πρόοδο και επαναφέρει το πλαίσιο σχεδιασμού στην αρχική του κατάσταση. Επίσης, δίνεται και η επιλογή άμεσης εκτέλεσης του δημιουργημένου σεναρίου.

Σημειώνεται ότι το μενού “File” προσφέρει την επιλογή φορτώματος ενός .tcl αρχείου δημιουργημένου από το TRAFIL. Σε αυτή την περίπτωση οι κόμβοι λαμβάνουν την αρχική τους διάταξη στο πλαίσιο σχεδιασμού, ενώ όλες οι παράμετροι είναι ήδη καθορισμένες.

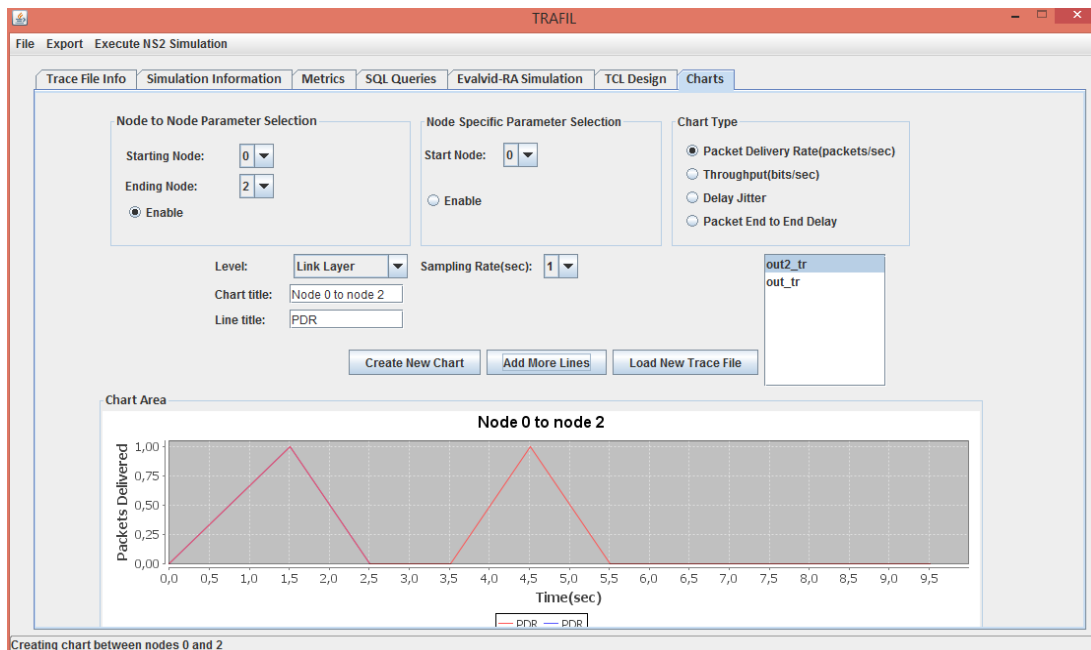
6.1.7.

Σχεδίαση γραφικών παραστάσεων

Στην καρτέλα “Charts” δίνεται η δυνατότητα παραγωγής γραφημάτων από τα δεδομένα και τις μετρήσεις που γίνονται στο TRAFIL. Η καρτέλα περιέχει τρεις κύριες περιοχές:

- *Επιλογή κόμβων γραφήματος:* Το TRAFIL μπορεί να σχεδιάσει γραφική παράσταση είτε για την μετάδοση μεταξύ δυο κόμβων, είτε για έναν κόμβο. Μόνο μια επιλογή είναι δυνατή σε κάθε παράσταση.
- *Επιλογή τύπου γραφήματος:* Όπως προαναφέρθηκε, υποστηρίζονται τέσσερις τύποι γραφημάτων: Packet Delivery Rate, Throughput, Delay Jitter και Packet End to End Delay.

- Επιλογή επιπέδου και λεπτομερειών:* Εδώ επιλέγεται το επίπεδο από το οποίο θα αντληθούν οι πληροφορίες για την γραφική παράσταση, ο ρυθμός δειγματοληψίας, ο τίτλος της και ο τίτλος κάθε γραφήματος σε αυτή. Επίσης, σε περίπτωση που ο χρήστης επιθυμεί να συγκρίνει τιμές δυο διαφορετικών προσομοιώσεων που βρίσκονται στην βάση δεδομένων, μπορεί να επιλέξει το trace file που επιθυμεί, το οποίο φορτώνεται μέσα από το κουμπί “Load New Trace File” στο TRAFIL. Έπειτα είναι δυνατόν να προσθέσει νέα γραφήματα στη γραφική παράσταση καθορίζοντας τις παραμέτρους του νέου γραφήματος και πατώντας στη συνέχεια το κουμπί “Add More Lines”. Σε περίπτωση που χρειάζεται να σχεδιαστεί εξ αρχής μια νέα γραφική παράσταση, αρκεί να πατηθεί το κουμπί “Create New Chart” για να δημιουργηθεί εξ αρχής η γραφική παράσταση με το τελευταίο μόνο γράφημα.
- Περιοχή γραφικής παράστασης:* Εδώ σχηματίζεται η γραφική παράσταση που ζητήθηκε. Το εμφανιζόμενο αποτέλεσμα μπορεί να εξαχθεί σε μορφή εικόνας από το μενού “Export” στο πάνω μέρος του TRAFIL.



Εικόνα 61 Οθόνη δημιουργίας γραφικών παραστάσεων

6.2.

Υποστηριζόμενες πλατφόρμες και προαπαιτούμενα προγραμματιστικά εργαλεία

Το TRAFIL είναι υλοποιημένο στη γλώσσα προγραμματισμού Java, κληρονομώντας την ιδιότητα cross-platform που προσφέρει. Μόνη απαίτηση για να μπορέσει ένα σύστημα να εκτελέσει το TRAFIL είναι να υποστηρίζει και να έχει εγκατεστημένη την Java³⁶, καθώς και τον εξυπηρετητή βάσεων δεδομένων MySQL (MySQL server³⁷). Με αυτά τα δυο εργαλεία είναι δυνατά τα χαρακτηριστικά pre-simulation και post-simulation.

Για την εξυπηρέτηση των simulation χαρακτηριστικών του TRAFIL, όμως, απαιτείται περιβάλλον Linux, καθώς τόσο ο NS-2 όσο και τα εξωτερικά script files που χρησιμοποιεί το TRAFIL εκτελούνται μόνο εκεί. Επιπλέον, στην λειτουργία προσομοίωσης με χρήση Evalvid-RA χρησιμοποιούνται βοηθητικά εργαλεία μερικά από τα οποία είναι εκτελέσιμα αρχεία Windows. Σε περίπτωση που κάποιο προαπαιτούμενο εργαλείο δεν είναι σωστά εγκατεστημένο, ειδοποιείται ο χρήστης κατά τη διαδικασία της προσομοίωσης. Επομένως, για την σωστή υποστήριξη των προσομοιώσεων και προσομοιώσεων με χρήση Evalvid-RA είναι απαιτούμενα τα εξής:

- NS-2³⁸: Ορθή εγκατάσταση του NS-2 και αντιγραφή του εκτελέσιμου ~/ns-2.35/ns στο φάκελο /usr/bin.
- Evalvid-RA³⁹: Ορθή εγκατάσταση στον NS-2, όπως περιγράφεται στις οδηγίες του ιστότοπου.
- Wine⁴⁰: Ορθή εγκατάσταση στο λειτουργικό σύστημα για υποστήριξη των εκτελέσιμων Windows.

Τέλος, το TRAFIL είναι διαθέσιμο εδώ⁴¹.

7. Αξιολόγηση

Σε αυτή την ενότητα θα προσπαθήσουμε να αναδείξουμε το πλεονέκτημα του TRAFIL έναντι παρεμφερών εργαλείων όσον αφορά στην ταχύτητά του σε σχέση με χρονοβόρα διαδικασία ταυτοποίησης, επεξεργασίας και αποθήκευσης ενός trace file. Όπως θα δούμε, το TRAFIL καταφέρνει να διατηρήσει το χρόνο ανάλυσης ενός trace file σε χαμηλά επίπεδα, ακόμα και για αρχεία της τάξης των 100MB.

Στον παρακάτω πίνακα βλέπουμε τους χρόνους επεξεργασίας αρχείων trace file διαφορετικού τύπου και μεγέθους.

Μέγεθος trace file (MB)	Χρόνος εκτέλεσης για Old Wireless trace file (ms)	Χρόνος εκτέλεσης για Normal trace file (ms)
2	1.967	1.456
6	2.499	2.215
13	4.973	3.635
24	8.699	5.613
35	13.095	8.406
47	17.816	11.767
62	20.906	16.048

Πίνακας 2 Συγκριτικός πίνακας χρόνων επεξεργασίας TRAFIL

Παρατηρούμε ότι τα μεγέθη διαφέρουν μεταξύ των τύπων trace file. Αυτό συμβαίνει λόγω της διαφορετικής αντιμετώπισης που κάνει το TRAFIL σε κάθε ένα. Συγκεκριμένα, σε ασύρματα σενάρια χρειάζεται μεγαλύτερος αριθμός submetafiles για την ανάλυση των trace file, επομένως απαιτείται περισσότερος χρόνος για την επεξεργασία κάθε γραμμής.

Τέλος, παρουσιάζουμε μια συγκριτική μελέτη μεταξύ του TRAFIL, του Tracegraph και του jTrana, τα οποία αναλύθηκαν στην ενότητα 2.3. Και τα δυο εργαλεία, όπως αναφέρθηκε, είναι πολύ δημοφιλή στον τομέα της ανάλυσης trace file, αλλά παραμένουν αρκετά χρονοβόρα και με λιγότερες δυνατότητες, καθώς δεν έχουν τρόπο αποθήκευσης και ανάκτησης των αποτελεσμάτων αργότερα.

Μέγεθος trace file (MB)	Χρόνος εκτέλεσης TRAFIL (ms)	Χρόνος εκτέλεσης Tracegraph (ms)	Χρόνος εκτέλεσης jTrana (ms)
2	1,967	27,000	29,700
6	2,499	75,400	68,000
13	4,973	197,600	135,200
24	8,699	495,100	177,700
35	13,095	949,300	272,000

47	17,816	1,090,700	362,700
62	20,906	2,097,100	486,800

Πίνακας 3 Συγκριτικός πίνακας χρόνων επεξεργασίας διαφόρων εργαλείων

Όπως φαίνεται, το TRAFIL διατηρεί σταθερά χρόνους 10 έως και 100 φορές πιο γρήγορους σε σχέση με τα υπόλοιπα προγράμματα, ενώ και η περίπτωση μεγάλων αρχείων (62MB) διατηρείται κάτω του μισού λεπτού.

Σε κάθε περίπτωση χρησιμοποιήθηκαν τα ίδια trace files, τα οποία προέκυψαν από προσομοίωση ασύρματου δικτύου και, όπως παρατηρήσαμε στον προηγούμενο πίνακα, το TRAFIL χρειαζόταν τον περισσότερο χρόνο. Οι μετρήσεις έγιναν σε Windows 7 64-bit, επεξεργαστή Intel Core i7-2670 @2.2Ghz και μνήμη RAM 4GB. Ο υπολογισμός στο Tracegraph και στο jTrana έγινε χειροκίνητα υπολογίζοντας τον μέσο χρόνο πολλαπλών εκτελέσεων.

Πέραν των τεράστιων δυνατοτήτων του στην ανάλυση, το TRAFIL έχει εξελιχθεί σε μια πλήρη σουίτα λειτουργιών όσον αφορά την προσομοίωση δικτύων σε NS-2. Έως τώρα κανένα παρεμφερές εργαλείο δεν έχει τόσο ευρεία ποικιλία σε αυτόν τον τομέα.

8. Επίλογος

Το TRAFIL δημιουργήθηκε με σκοπό να προτυποποιήσει και να επιταχύνει την διαδικασία της ανάλυσης και επεξεργασίας των trace files, κάτι που κατόρθωσε με την εισαγωγή των metafiles. Εκτός αυτού, παρείχε πολλά επιπλέον χαρακτηριστικά όπως αποθήκευση αποτελεσμάτων, δημιουργία και εξαγωγή μετρήσεων και γραφημάτων, κ.α., για να εξελιχθεί εν τέλει καλύπτοντας την εκτέλεση προσομοιώσεων αλλά και την δημιουργία τους. Σχεδιάστηκε ώστε να είναι εύκολο στη χρήση τόσο από αρχάριο χρήστη, δίνοντας πολλές έτοιμες επιλογές και αυτοματοποιώντας τετριμμένες διαδικασίες, όσο και από έμπειρο, δίνοντας πάντα την δυνατότητα για χειροκίνητη εν τέλει επεξεργασία είτε στην ανάλυση trace file, είτε στην δημιουργία TCL file, ώστε να καλύψει όλες τις περιπτώσεις.

8.1. *Μελλοντικές επεκτάσεις*

Παρ' όλη την εξέλιξη του TRAFIL υπάρχουν ακόμα πολλοί τομείς στους οποίους θα μπορούσε να βελτιωθεί ή και να επεκταθεί. Συγκεκριμένα, στον τομέα του σχεδιασμού σεναρίων μπορεί να γίνει πιο πλήρης υποστήριξη των προσθηκών και λειτουργιών του NS-2.

Επίσης, παρότι ο NS-2 εξακολουθεί να χρησιμοποιείται σε μεγάλο τμήμα της κοινότητας, δεδομένης της εξέλιξης του NS-3 και της συνέχειας του NS-2, είναι αναμενόμενη η επικράτησή του τελικά. Επομένως μια πολύ χρήσιμη μελλοντική προσθήκη για το TRAFIL είναι η υποστήριξη του NS-3 σε όλες τις φάσεις λειτουργίας του.

ΒΙΒΛΙΟΓΡΑΦΙΑ

-
- ¹ "Network simulation - Wikipedia, the free encyclopedia." 2005. 27 Apr. 2015 <http://en.wikipedia.org/wiki/Network_simulation>
- ² "The Network Simulator - ns-2." 27 Apr. 2015 <<http://www.isi.edu/nsnam/ns/>>
- ³ "OTcl - Wikipedia, the free encyclopedia." 2007. 27 Apr. 2015 <<http://en.wikipedia.org/wiki/OTcl>>
- ⁴ "NS-2 Trace Formats - Nsnam." 2006. 27 Apr. 2015 <http://nsnam.isi.edu/nsnam/index.php/NS-2_Trace_Formats>
- ⁵ "Ανάπτυξη εργαλείου αυτοματοποίησης και διαχείρισης εκτέλεσης εξομοιώσεων" 2012. June, Patras. 2015 <http://ru6.cti.gr/ru6/system/files/bouras_site/ergasies/diplwmatikes/128_charalampi_o.doc>
- ⁶ de Melo Lustosa, Marllus, and Shivjay Singh. "Liowasn Project: An Operating System Remastered for Works with Simulation of Wireless Sensor Networks." *International Journal of Computer Applications* 52 (2012).
- ⁷ "Liowasn Project | SourceForge.net." 2012. 2 May. 2015 <<http://sourceforge.net/projects/liowasn/>>
- ⁸ "PiccSIM Simulation of wireless control systems - Aalto University" 2013. 2 May. 2015 <<http://wsn.aalto.fi/en/tools/piccsim/>>
- ⁹ "Simulink - Simulation and Model-Based Design - MathWorks." 2 May. 2015 <<http://www.mathworks.com/products/simulink/>>
- ¹⁰ Kohtamaki, T et al. "Piccsim toolchain-design, simulation and automatic implementation of wireless networked control systems." *Networking, Sensing and Control, 2009. ICNSC'09. International Conference on* 26 Mar. 2009: 49-54.
- ¹¹ Björkbom, Mikael, and Henri Öhman. "Integration of piccsim and truetime." (2012).
- ¹² Sobeih, Ahmed et al. "J-sim: A simulation environment for wireless sensor networks." *Proceedings of the 38th annual Symposium on Simulation* 4 Apr. 2005: 175-187.
- ¹³ "J-Sim Official - Google Sites." 2011. 2 May. 2015 <<https://sites.google.com/site/jsimofficial/>>
- ¹⁴ "gEditor, the J-Sim Graphical Editor - J-Sim Home Page." 2008. 2 May. 2015 <<http://j-sim.cs.uiuc.edu/geditor/vo.5/>>
- ¹⁵ "NSG2 - Peng-Jung Wu - Google Sites." 2010. 1 May. 2015 <<https://sites.google.com/site/pengjungwu/nsg>>
- ¹⁶ Braga, Thais RM et al. "Mannasim: a framework to the simulation of wireless sensors networks." *Electronics Magazine of Undergraduate Scientific Research of the Brazilian Computer Science Society (REIC)* (2004): 50.
- ¹⁷ "MannaSim Framework - Main Page." 2004. 2 May. 2015 <<http://www.mannasim.dcc.ufmg.br/>>
- ¹⁸ Jansen, Sam. "Network Simulation Cradle." 2008.
- ¹⁹ "Network Simulation Cradle: index - WAND Group." 2005. 2 May. 2015 <<http://www.wand.net.nz/~stj2/nsc/>>
- ²⁰ "Ns2measure - Cng1wiki." 2008. 2 May. 2015 <<http://cng1.iet.unipi.it/wiki/index.php/Ns2measure>>
- ²¹ "JTrana - Google Sites." 2012. 2 May. 2015 <<https://sites.google.com/site/ns2trana/>>
- ²² Malek, Jaroslaw, and Kamil Nowak. "Trace graph-data presentation system for network simulator ns." *Proceedings of the Information Systems—Concepts, Tools and Applications (ISAT 2003), Poland* (2003).
- ²³ "Network Simulators: Tracegraph - a graphing software to ..." 2012. 2 May. 2015 <<http://www.nsnam.com/2012/09/tracegraph-graphing-software-to-plot.html>>

-
- 24 "Ns-2 Trace Analyzer" 2005. 3 May. 2015 <<http://trace-analyzer.sourceforge.net/>>
- 25 "NS2 Visual Trace Analyzer 0.2.72 Released." 2013. 3 May. 2015
<<https://nsvisualtraceanalyzer.wordpress.com/2012/07/03/ns2-trace-analyzer-download/>>
- 26 "NS-2 Trace Formats - Nsnam." 2006. 3 May. 2015
<http://nsnam.isi.edu/nsnam/index.php/NS-2_Trace_Formats>
- 27 "X. Creating Wired-cum-Wireless and MobileIP Simulations ..." 6 May. 2015
<<http://www.isi.edu/nsnam/ns/tutorial/nsscript6.html>>
- 28 "40. Applications and transport agent API." 6 May. 2015
<<http://www.isi.edu/nsnam/ns/doc/node499.html>>
- 29 "TCP - Wikipedia." 2003. 6 May. 2015
<http://en.wikipedia.org/wiki/Transmission_Control_Protocol>
- 30 "User Datagram Protocol - Wikipedia, the free encyclopedia." 2004. 6 May. 2015
<http://en.wikipedia.org/wiki/User_Datagram_Protocol>
- 31 Lie, Arne, and Jirka Klaue. "Evalvid-RA: trace driven simulation of rate adaptive MPEG-4 VBR video." *Multimedia Systems* 14.1 (2008): 33-50.
- 32 "EvalVid-RA - The Rate Adaptive Evalvid tool-set." 2006. 10 May. 2015
<<http://www.item.ntnu.no/~arnelie/Evalvid-RA.htm>>
- 33 Klaue, Jirka, Berthold Rathke, and Adam Wolisz. "Evalvid—A framework for video transmission and quality evaluation." *Computer Performance Evaluation. Modelling Techniques and Tools* (2003): 255-272.
- 34 "FFmpeg." 2010. 10 May. 2015 <<https://www.ffmpeg.org/>>
- 35 "Depth-first search - Wikipedia, the free encyclopedia." 2004. 22 May. 2015
<http://en.wikipedia.org/wiki/Depth-first_search>
- 36 "Download Free Java Software." 2012. 13 May. 2015 <<https://java.com/download>>
- 37 "MySQL :: Download MySQL Community Server." 2010. 13 May. 2015
<<https://dev.mysql.com/downloads/mysql/>>
- 38 "nsnam - Browse /allinone/ns-allinone-2.34 at SourceForge.net." 2009. 13 May. 2015
<<http://sourceforge.net/projects/nsnam/files/allinone/ns-allinone-2.34/>>
- 39 "Download EvalVid-RA - The Rate Adaptive Evalvid tool-set." 2006. 13 May. 2015
<http://www.item.ntnu.no/~arnelie/evalvid_test/evalvid_ra_v1.04_2.zip>
- 40 "WineHQ - Wine Binary Downloads." 2013. 13 May. 2015
<<https://www.winehq.org/download/>>
- 41 "TRAFIL download" <<http://ru6.cti.gr/ru6/research-areas/network-simulations#TRAFIL>>.