



Τεχνολογικό Εκπαιδευτικό Ίδρυμα Πελοποννήσου
Σχολή Τεχνολογικών Εφαρμογών
Τμήμα Μηχανικών Πληροφορικής Τ.Ε.

Μοντελοποίηση και Προσομοίωση Δικτύων
Ανάλυση από την άποψη του σχεδιαστή και του
προγραμματιστή

Τώρου Περσεφώνη Δέσποινα & Καρκούλης Αντώνης

Επιβλέπων Καθηγητής: Πικραμμένος Ιωάννης

Σπάρτη Νοέμβριος 2017

ΔΗΛΩΣΗ ΜΗ ΛΟΓΟΚΛΟΠΗΣ ΚΑΙ ΑΝΑΛΗΨΗΣ ΠΡΟΣΩΠΙΚΗΣ ΕΥΘΥΝΗΣ

"Με πλήρη επίγνωση των συνεπειών του νόμου περί πνευματικών δικαιωμάτων, δηλώνω ενυπογράφως ότι είμαι αποκλειστικός συγγραφέας της παρούσας Πτυχιακής Εργασίας, για την ολοκλήρωση της οποίας κάθε βοήθεια είναι πλήρως αναγνωρισμένη και αναφέρεται λεπτομερώς στην εργασία αυτή. Έχω αναφέρει πλήρως και με σαφείς αναφορές, όλες τις πηγές χρήσης δεδομένων, απόψεων, θέσεων και προτάσεων, ιδεών και λεκτικών αναφορών, είτε κατά κυριολεξία είτε βάση επιστημονικής παράφρασης. Αναλαμβάνω την προσωπική και ατομική ευθύνη ότι σε περίπτωση αποτυχίας στην υλοποίηση των ανωτέρω δηλωθέντων στοιχείων, είμαι υπόλογος έναντι λογοκλοπής, γεγονός που σημαίνει αποτυχία στην Πτυχιακή μου Εργασία και κατά συνέπεια αποτυχία απόκτησης του Τίτλου Σπουδών, πέραν των λοιπών συνεπειών του νόμου περί πνευματικών δικαιωμάτων. Δηλώνω, συνεπώς, ότι αυτή η Πτυχιακή Εργασία προετοιμάστηκε και ολοκληρώθηκε από εμένα προσωπικά και αποκλειστικά και ότι, αναλαμβάνω πλήρως όλες τις συνέπειες του νόμου στην περίπτωση κατά την οποία αποδειχθεί, διαχρονικά, ότι η εργασία αυτή ή τμήμα της δε μου ανήκει διότι είναι προϊόν λογοκλοπής άλλης πνευματικής ιδιοκτησίας."

Όνομα και Επώνυμο Συγγραφέα (Με Κεφαλαία): ΤΩΡΟΥ ΠΕΡΣΕΦΩΝΗ ΔΕΣΠΟΙΝΑ

Υπογραφή (Ολογράφως, χωρίς μονογραφή):

Ημερομηνία (Ημέρα – Μήνας – Έτος): 22 ΝΟΕΜΒΡΙΟΥ 2017

ΔΗΛΩΣΗ ΜΗ ΛΟΓΟΚΛΟΠΗΣ ΚΑΙ ΑΝΑΛΗΨΗΣ ΠΡΟΣΩΠΙΚΗΣ ΕΥΘΥΝΗΣ

"Με πλήρη επίγνωση των συνεπειών του νόμου περί πνευματικών δικαιωμάτων, δηλώνω ενυπογράφως ότι είμαι αποκλειστικός συγγραφέας της παρούσας Πτυχιακής Εργασίας, για την ολοκλήρωση της οποίας κάθε βοήθεια είναι πλήρως αναγνωρισμένη και αναφέρεται λεπτομερώς στην εργασία αυτή. Έχω αναφέρει πλήρως και με σαφείς αναφορές, όλες τις πηγές χρήσης δεδομένων, απόψεων, θέσεων και προτάσεων, ιδεών και λεκτικών αναφορών, είτε κατά κυριολεξία είτε βάση επιστημονικής παράφρασης. Αναλαμβάνω την προσωπική και ατομική ευθύνη ότι σε περίπτωση αποτυχίας στην υλοποίηση των ανωτέρω δηλωθέντων στοιχείων, είμαι υπόλογος έναντι λογοκλοπής, γεγονός που σημαίνει αποτυχία στην Πτυχιακή μου Εργασία και κατά συνέπεια αποτυχία απόκτησης του Τίτλου Σπουδών, πέραν των λοιπών συνεπειών του νόμου περί πνευματικών δικαιωμάτων. Δηλώνω, συνεπώς, ότι αυτή η Πτυχιακή Εργασία προετοιμάστηκε και ολοκληρώθηκε από εμένα προσωπικά και αποκλειστικά και ότι, αναλαμβάνω πλήρως όλες τις συνέπειες του νόμου στην περίπτωση κατά την οποία αποδειχθεί, διαχρονικά, ότι η εργασία αυτή ή τμήμα της δε μου ανήκει διότι είναι προϊόν λογοκλοπής άλλης πνευματικής ιδιοκτησίας."

Όνομα και Επώνυμο Συγγραφέα (Με Κεφαλαία): ΚΑΡΚΟΥΛΗΣ ΑΝΤΩΝΗΣ

Υπογραφή (Ολογράφως, χωρίς μονογραφή):

Ημερομηνία (Ημέρα – Μήνας – Έτος): 22 ΝΟΕΜΒΡΙΟΥ 2017



ΜΟΝΤΕΛΟΠΟΙΗΣΗ ΚΑΙ ΠΡΟΣΟΜΟΙΩΣΗ ΔΙΚΤΥΩΝ

Ανάλυση από την άποψη του σχεδιαστή και του
προγραμματιστή – Περσεφώνη Δέσποινα Τώρου &
Αντώνης Καρκούλης

Περίληψη

Στην παρούσα εργασία έγινε μία βιβλιογραφική μελέτη πάνω στην τεχνολογία της προσομοίωσης δικτύων. Έγινε ανάλυση πάνω σε μεθόδους προσομοίωσης (Discrete-Event-Simulation) καθώς και σε εργαλεία (προσομοιωτές), αναφορικά με την ευχέρεια και την αξιοπιστία τους. Αρχικά έγινε αναφορά σε βασικές γνώσεις για τα δίκτυα και την προσομοίωση δικτύων. Επίσης, έγινε αναφορά σε βασικές γνώσεις στατιστικής (πιθανοθεωρητικές κατανομές) και δειγματοληψίας τυχαίων αριθμών (μέθοδος Monte-Carlo). Έπειτα, παρουσιάστηκε η μέθοδος Διακριτών Γεγονότων (Discrete Event Simulation) πάνω στην οποία βασίζονται τα περισσότερα προγράμματα προσομοίωσης. Γίνεται, ύστερα, αναφορά σε φημισμένα προγράμματα προσομοίωσης, παρουσιάζονται τα τεχνικά χαρακτηριστικά τους και σχολιάζεται η αποδοτικότητα τους και ευχέρεια τους σε διάφορες περιπτώσεις προσομοίωσης δικτύων. Παρουσιάζονται και σχολιάζονται, επιπλέον, διάφορες έρευνες (άρθρα), στις οποίες γίνεται χρήση φημισμένων προσομοιωτών ή ακόμα και ανάπτυξη νέων μοντέλων για την προσομοίωση και εξομοίωση δικτύων.

Περιεχόμενα

Λίστα με εικόνες	10
1. Δίκτυα και προσομοίωση δικτύων	12
1.1 Εισαγωγή.....	12
1.2 Προσομοίωση δικτύων	21
1.2.1 Τύποι προσομοίωσης.....	22
1.3 Η ανάγκη για μοντελοποίηση και προσομοίωση δικτύων	24
1.3.1 Προσομοίωση έναντι εξομίωσης.....	24
1.3.2 Τύποι δικτύων επικοινωνιών, δομές μοντελοποίησης.....	28
1.3.3 Η απόδοση ενός δικτύου και τα χαρακτηριστικά του	32
2. Βασικές έννοιες στατιστικής	37
3. Discrete event simulation και η συμβολή του χρόνου στην προσομοίωση	54
3.1 Μέθοδος επόμενου γεγονότος.....	60
3.2 Μέθοδος σταθερού χρονικού διαστήματος	62
4. Λογισμικά προσομοίωσης	66
4.1 Προσομοιωτής ns-2	66
4.2 Ο Προσομοιωτής ns-3	70
4.3 Ο προσομοιωτής OMNET++	79

4.4 Ο προσομοιωτής OPNET	96
4.5 Προσομοιωτής Network Workbench	101
4.6 Ο προσομοιωτής Netsim	103
4.7 Ο προσομοιωτής MARS.....	107
4.8 Ο προσομοιωτής COMNET III.....	108
4.9 Ο προσομοιωτής REAL	112
4.10 Ο προσομοιωτής Dummynet	114
5. Παραδείγματα από εργασίες στην προσομοίωση δικτύων.....	119
6. Συμπεράσματα – Μελλοντικές εργασίες	147
Αναφορές	149

Λίστα με εικόνες

Εικόνα 1: Αριθμός συνδέσεων για 4 ή 5 υπολογιστές	15
Εικόνα 2: Κόμβοι σε ένα δίκτυο.....	18
Εικόνα 3: Παραδείγματα προσομοίωσης στη ζωή μας. (α) Προσομοίωση δικτύων, (β) Προσομοίωση ροής σε αρτηρία με ανεύρυσμα, (γ) Προσομοίωση αεροδυναμικής αεροσκάφους.....	37
Εικόνα 4: Δομή ενός συστήματος προσομοίωσης.....	59
Εικόνα 5: Παράγοντες που συναντάμε σε ένα μοντέλο προσομοίωσης διακριτών γεγονότων.....	59
Εικόνα 6: Περιβάλλον προσομοιωτή ns-2.....	69
Εικόνα 7: Παράδειγμα TCPClientServer με γραφικό περιβάλλον.	86
Εικόνα 8: Παράδειγμα TCPClientServer κατά την προσομοίωσή του.	93
Εικόνα 9: Ιστόγραμμα του μέσου χρόνου αναμονής στην ουρά του εξυπηρέτη.	95
Εικόνα 10: Παράδειγμα μοντέλου δικτύου στον Project Editor του OPNET.....	97
Εικόνα 11: Επιλογές οπτικοποίησης αποτελεσμάτων προσομοίωσης στον OPNET.	99
Εικόνα 12: Γραφικό περιβάλλον του προσομοιωτή Network Workbench	102
Εικόνα 13: Γραφικό Περιβάλλον του προσομοιωτή Netsim	106
Εικόνα 14: Γραφικό περιβάλλον του προσομοιωτή COMNET III.....	111
Εικόνα 15: Η αρχή λειτουργίας του Dummynet.	116
Εικόνα 16: Δόμη ενός κόμβου στο Dummynet.....	118
Εικόνα 17: Διάφοροι τρόποι σχηματισμού πειραμάτων στο Netsim.....	118
Εικόνα 18: Δίκτυο Εθνικού Επιστημονικού Ιδρύματος (NSFNET).....	122
Εικόνα 19: Αξιοπιστία συστήματος.....	123
Εικόνα 20: Επιρροή της συσχέτισης για διαφορετικές τιμές χρονικού περιορισμού.	123
Εικόνα 21: Συνιστώσες του προγράμματος προσομοίωσης [32].	126
Εικόνα 22: Βασή δεδομένων του μοντέλου προσομοίωσης [32].	127
Εικόνα 23 Ενεργειακές συνιστώσες (components) ενός τυπικού ασύρματου κόμβου αισθητήρα.	129
Εικόνα 24: Η προτεινόμενη ενεργειακά ενημερούμενη δομή για προσομοίωση ασύρματου δικτύου αισθητήρων.	129

Εικόνα 25: Ένα υπο-προσομοίωση δίκτυο με χρήση του προγράμματος WSNsim	130
Εικόνα 26: Ο προσομοιωτής MSPsim.....	132
Εικόνα 27: Ο προσομοιωτής Prowler.....	133
Εικόνα 28: Προσομοιωτής Wireless Sensor Network Localization Simulator	134
Εικόνα 29: Ο προσομοιωτής AlgoSenSim.....	135
Εικόνα 30: Ο προσομοιωτής SIDnet-SWANS.....	136
Εικόνα 31: Η αρχιτεκτονική του προσομοιωτή TOSSIM	137
Εικόνα 32: Αρχιτεκτονική του NS.....	138
Εικόνα 33: Ισχύς μεταξύ μέσης ισχύος σήματος και απόστασης.....	139
Εικόνα 34: Σχέση μεταξύ αριθμού ληφθέντων πακέτων και ισχύος σήματος.....	140
Εικόνα 35: Σχέση μεταξύ ληφθέντων πακέτων και απόστασης.....	141
Εικόνα 36: Παράμετροι σεναρίου για την προσομοίωση USN δικτύου.....	145
Εικόνα 37: Εκτίμηση αποδοτικότητας του μοντέλου προσομοίωσης μελετώντας τα χαρακτηριστικά κατά ζεύγη αναφορικά με τον αριθμό των κόμβων.....	146

1. Δίκτυα και προσομοίωση δικτύων

1.1 Εισαγωγή

Ο τρόπος εργασίας μας καθώς και της επικοινωνίας έχει αλλάξει σημαντικά λόγω των νέων τεχνολογιών που αναπτύσσονται στον χώρο της πληροφορικής και των επικοινωνιών. Εδώ και χρόνια οι τεχνολογίες αυτές μας έχουν οδηγήσει στην Κοινωνία της πληροφορίας, της οποίας σημαντικό κομμάτι αποτελούν τα δίκτυα επικοινωνίας.

Από πολύ παλιά οι άνθρωποι αναγκάστηκαν να επινοήσουν τρόπους προκειμένου να επιτύχουν επικοινωνία μεταξύ τους. Η χρήση σημάτων καπνού αποτελεί την πιο παλιά μέθοδο επικοινωνίας. Τον 19^ο αιώνα οι άνθρωποι χρησιμοποίησαν την τηλεγραφία. Τον 20^ο αιώνα χρησιμοποίησαν την τηλεφωνία καθώς την 3^η χιλιετία έγινε χρήση των δικτύων των υπολογιστών.

Ας ορίσουμε λοιπόν τι εστί δίκτυο, στον χώρο της πληροφορικής και των τηλεπικοινωνιών. Δίκτυο είναι ένα σύστημα το οποίο στοχεύει στην σύνδεση τερματικών συσκευών κάθε είδους με δομή τέτοια ώστε να οδηγεί στην επίτευξη

της μεταξύ τους επικοινωνίας. Εκτός από τις συνδεδεμένες τερματικές συσκευές, σε ένα δίκτυο περιλαμβάνονται κόμβοι, μέσα για την διακίνηση πληροφοριών καθώς και συσκευές τηλεπικοινωνιών. Ένα δίκτυο σκοπεύει στην πρόσβαση των χρηστών σε υπάρχοντες πόρους. Τέτοιοι πόροι είναι οι συσκευές υλικού, το λογισμικό και τα δεδομένα. Η δικτύωση των υπολογιστών παρέχει την δυνατότητα για συνδυασμό πληροφορίας, επικοινωνιών και διασκέδασης [1].

Παλιά, ο Graham Bell έθεσε σε λειτουργία για πρώτη φορά το τηλέφωνο. Κατάφερε να συνομιλήσει με έναν φίλο του από απόσταση, με χρήση δυο τηλεφωνικών συσκευών και μιας τηλεφωνικής γραμμής. Ύστερα και οι άλλοι φίλοι του του ζήτησαν τη δυνατότητα αυτή. Ο Bell τότε, για κάθε τέτοια σύνδεση είχε στη διάθεση του από δυο τηλεφωνικές συσκευές και από μια γραμμή. Συνεπώς, ο Bell είχε στο σπίτι του αριθμό τηλεφωνικών συσκευών ίδιο με αυτόν των συνδέσεων. Παράλληλα, ο ίδιος αριθμός γραμμών ξεκινούσε από το σπίτι του μέχρι την τοποθεσία των φίλων του [2].

Ως αποτέλεσμα όμως, η αύξηση του αριθμού των χρηστών της τηλεφωνίας θα οδηγούσε σε αύξηση του αριθμού των συσκευών και των γραμμών. Σε μικρό χρονικό διάστημα έγινε υπερβολική αύξηση των χρηστών τηλεφωνίας. Συνεπώς, από ένα σημείο και μετά η κατάσταση δεν ήταν δυνατό να συνεχιστεί διότι έτσι θα έφτανε σε αδιέξοδο. Για να κατανοηθεί το παραπάνω γεγονός ας δούμε το εξής παράδειγμα. Έστω, ότι έχουμε K σημεία. Τα σημεία αυτά για να επικοινωνήσουν μεταξύ τους, θα πρέπει να έχουμε αριθμό συνδέσεων ίσο με:

$$N = \frac{K(K-1)}{2}$$

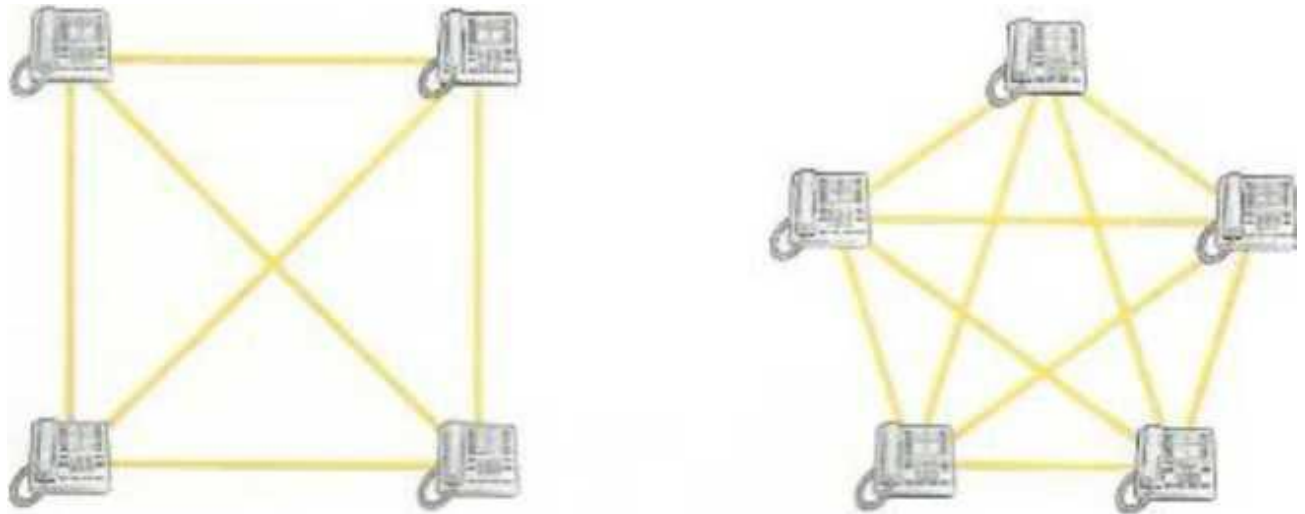
Ο κάθε συνδρομητής θα πρέπει να έχει

$$\Sigma = K - 1$$

συσκευές. Έστω ότι έχουμε 200 σημεία.

Ο αριθμός των γραμμών θα είναι 19900. Κάθε χρήστης θα έχει στη διάθεσή του 199 συσκευές, πράγμα το οποίο δεν μπορεί να υφίσταται.

Δύο συσκευές για να επικοινωνούν θα πρέπει να είναι συνδεδεμένες από σημείο σε σημείο. Η σύνδεση αυτή μπορεί να επιτευχθεί ενσύρματα ή ασύρματα. Όταν έχουμε αύξηση του αριθμού των συσκευών με αποτέλεσμα την επικοινωνία περισσότερων από δυο συσκευών μεταξύ τους καθιστά αδύνατη τη σύνδεση από σημείο σε σημείο [1].



Εικόνα 1: Αριθμός συνδέσεων για 4 ή 5 υπολογιστές

Η ανάγκη για την λύση του προβλήματος της επικοινωνίας οδήγησε στην δημιουργία δικτύου, του οποίου οι κόμβοι να είναι δυνατό να χρησιμοποιηθούν από οποιονδήποτε θέλει να επικοινωνήσει.

Συνεπώς δημιουργήθηκε το Δίκτυο. Πολλά στάδια έλαβαν τόπο προκειμένου να επιτευχθεί η βέλτιστη λύση. Στην αρχή δημιουργήθηκαν τα πρώτα τηλεφωνικά κέντρα. Στα τηλεφωνικά κέντρα ο συνδρομητής συνδεόταν ακτινικά με μια συγκεκριμένη γραμμή και συσκευή. Τότε, η σύνδεση των γραμμών τότε γινόταν με την βοήθεια βυσμάτων από τις τηλεφωνήτριες, οι οποίες χειρίζονταν τα τηλεφωνικά κέντρα. Επιπλέον, υπήρξαν ειδικές ενσωματωμένες γεννήτριες

ρεύματος στις τηλεφωνικές συσκευές. Τότε δεν υπήρχε η επιλογή αριθμού οπότε οι γεννήτριες αυτές επέτρεπαν τις κλήσεις προς το κέντρο.

Έτσι, είχαμε την πρώτη μορφή δικτύου επικοινωνιών φωνής. Τα ηλεκτρομηχανικά τηλεφωνικά κέντρα εξελίχθηκαν με τη χρήση της αυτόματης επιλογής. Έπειτα εξελίχθηκαν σε ηλεκτρονικά κέντρα, τα οποία μας έφεραν σήμερα στη χρήση υπολογιστικών συστημάτων και ψηφιακών τεχνικών επιλογής και μετάδοσης [2].

Με παρόμοιο τρόπο εξελίχθηκαν και τα δίκτυα δεδομένων. Στο πρώτο στάδιο της εξέλιξης ένας τερματικός σταθμός (Data Terminal Equipment) ήταν συνδεδεμένος με έναν άλλον τερματικό σταθμό. Οι σταθμοί χρησιμοποιούσαν το κοινό τηλεφωνικό δίκτυο ή τις μόνιμες αφιερωμένες γραμμές. Σε επόμενο στάδιο της εξέλιξης λόγω της ανάγκης για πολλαπλές συνδέσεις, δημιουργήθηκαν ποικίλα δίκτυα δεδομένων. Πλέον, τα σύγχρονα δίκτυα έχουν φτάσει στο σημείο ώστε να μην είναι αναγκαίες πολλαπλές αφιερωμένες συνδέσεις μεταξύ των συνδρομητών και έτσι ο κάθε συνδρομητής μπορεί να είναι συνδεδεμένος με το πλησιέστερο τηλεφωνικό κέντρο με χρήση μόνο μίας γραμμής.

Συνεπώς ας ορίσουμε το επικοινωνιακό δίκτυο. Είναι ένα σύνολο κόμβων συνδεδεμένων μεταξύ τους με χρήση γραμμών προκειμένου να υπάρχει ανταλλαγή πληροφορίας μεταξύ τους.

Έχουμε δυο είδη κόμβων: οι τερματικοί και οι επικοινωνιακοί κόμβοι. Οι τερματικοί κόμβοι χρησιμοποιούνται για την παραγωγή ή την κατανάλωση πληροφοριών, μεταφερόμενες στο δίκτυο. Η μεταφορά της πληροφορίας όλων επιτυγχάνεται με τη χρήση των επικοινωνιακών κόμβων.

Ένας τερματικός κόμβος είναι μια τερματική συσκευή όπως για παράδειγμα ένας υπολογιστής, μια τηλεφωνική συσκευή ή ένας εκτυπωτής. Ένας επικοινωνιακός

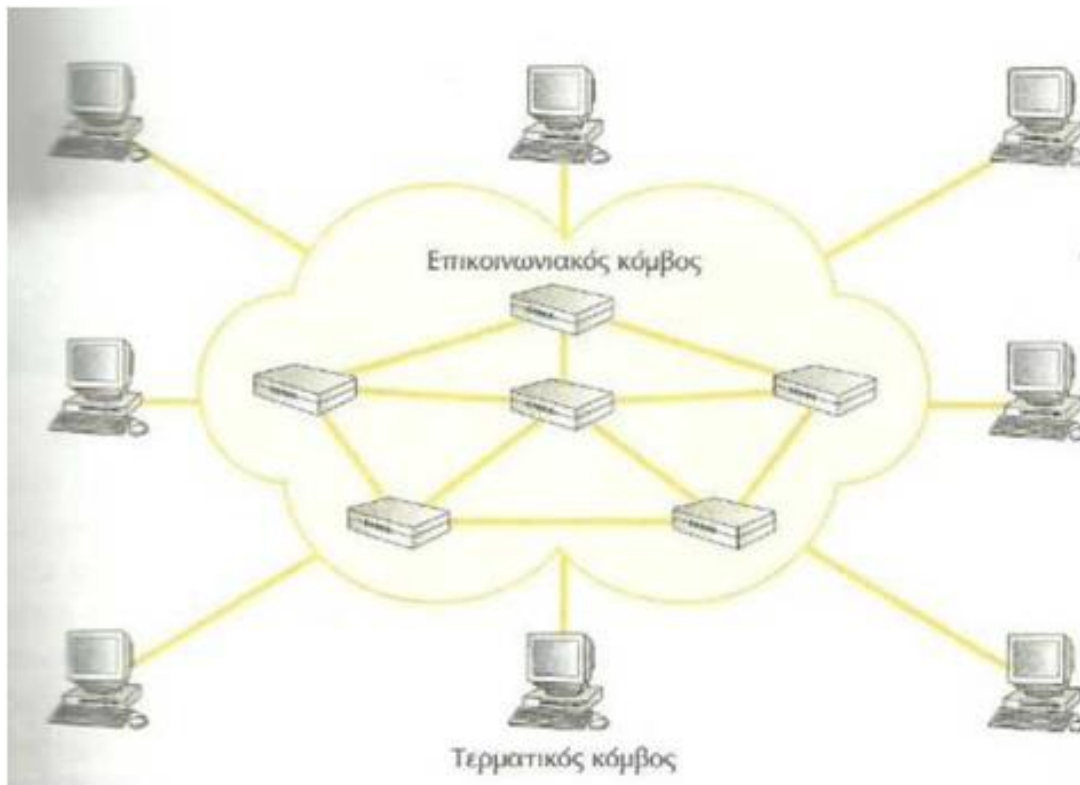
κόμβος είναι μια συσκευή επικοινωνίας όπως ένα τηλεφωνικό κέντρο, ένας δρομολογητής ή μια πύλη [3].

Υπάρχουν πολλά είδη δικτύων επικοινωνίας. Το πιο γνωστό δίκτυο επικοινωνίας είναι το τηλεφωνικό δίκτυο. Άλλο δίκτυο επικοινωνίας είναι το δίκτυο υπολογιστών. Χρήσεις υπολογιστικού δικτύου συναντάμε στα γραφεία προκειμένου να έχουν τη δυνατότητα οι εργαζόμενοι να χρησιμοποιούν κοινά δεδομένα και συσκευές. Επίσης, γίνεται χρήση δικτύων υπολογιστών στα εργοστάσια για την επίτευξη των αυτοματοποιημένων διαδικασιών παραγωγής, αλλά και σε ολόκληρο τον κόσμο ώστε να έχουν οι χρήστες πρόσβαση στο Διαδίκτυο.

Με την βοήθεια ψηφιακών δικτύων υψηλών ταχυτήτων επιτυγχάνονται πλέον οι μετάδοσης φωνής, δεδομένων και εικόνας. Τα ψηφιακά δίκτυα χρησιμοποιούν μεθόδους bandwidth on demand, για την επίτευξη βέλτιστης εκμετάλλευσης της χωρητικότητας των καναλιών και των επικοινωνιακών κόμβων. Μια τεχνική η οποία χρησιμοποιείται είναι η Asynchronous Transfer Mode (ATM) [2].

Η χρήση δικτύων παρέχει σημαντικά οφέλη μερικά από τα οποία είναι ο διαμερισμός πόρων, η υψηλή αξιοπιστία, η επικοινωνία μεταξύ των χρηστών του δικτύου και η εξοικονόμηση χρημάτων [3].

Πλέον οι οπτικές ίνες αποτελούν ισχυρό μέσο μετάδοσης. Με την χρήση οπτικών ινών, οι ρυθμοί μετάδοσης δεδομένων είναι πολύ υψηλοί.



Εικόνα 2: Κόμβοι σε ένα δίκτυο

Τα εργαλεία προσομοίωσης αποτελούν μια πολύ σημαντική παράμετρο στη σύγχρονη τεχνολογία. Είναι δυνατό να εφαρμοστούν σε διαφορετικές επιστήμες, όπως η μηχανική, η φυσική, η βιολογία, η οικονομία και οι κοινωνικές επιστήμες. Άλλες σημαντικές εφαρμογές των προσομοιωτών εντοπίζονται επίσης στους τομείς της μηχανικής, όπως είναι τα έργα του πολιτικού μηχανικού, στις δομικές κατασκευές, στη μηχανολογία, και στη μηχανική ηλεκτρονικών υπολογιστών. Με τη χρήση των υπολογιστικών προσομοιώσεων είναι εφικτό να δοκιμαστεί ένα υποθετικό μοντέλο αντικειμένων ή δραστηριοτήτων σε έναν υπολογιστή, με τέτοιο τρόπο ώστε να μπορούν να μελετηθούν και να παρατηρηθεί η λειτουργία τους τόσο σαν μονάδες, όσο και σαν μέρος συστήματος. Διαφορετικές μεταβλητές

μπορούν να χρησιμοποιηθούν για την πρόβλεψη της συμπεριφοράς του συστήματος. Η προσομοίωση σε υπολογιστή μπορεί να χρησιμοποιηθεί για να βοηθήσει την μοντελοποίηση και την ανάλυση πολλών συστημάτων. Η εφαρμογή της τεχνολογίας προσομοίωσης σε εφαρμογές δικτύωσης, όπως προσομοίωση κίνησης του δικτύου είναι ένα σχετικά νέο πεδίο, με πολλές εφαρμογές, τόσο ανοικτού όσο και κλειστού κώδικα.

Στο πεδίο των εφαρμογών δικτύου, η χρήση συστημάτων πληροφορικής και τεχνολογιών προσομοίωσης αναφέρονται στην προσομοίωση των αλγορίθμων δικτύων ή συστημάτων με τη χρήση τεχνολογίας λογισμικού. Το πεδίο εφαρμογής είναι εξαιρετικά περιορισμένο σε σχέση με το γενικότερο αντικείμενο της προσομοίωσης και είναι αναμενόμενο ότι οι ειδικότερες προδιαγραφές και περιορισμοί θα τεθούν στις εφαρμογές προσομοίωσης δικτύου. Για παράδειγμα, οι προσομοιώσεις του δικτύου επιβάλλεται να δώσουν μεγαλύτερη έμφαση στην απόδοση ή την εγκυρότητα ενός κατανεμημένου πρωτοκόλλου ή αλγόριθμου, ενώ δεν είναι σημαντικό να επικεντρωθούν σε χαρακτηριστικά όπως η οπτική αναπαράσταση των προσομοιώσεων ή η διεξαγωγή τους σε πραγματικό χρόνο. Επιπλέον περιορισμοί προστίθενται καθώς η τεχνολογία των δικτύων εξακολουθεί να αναπτύσσεται διαρκώς και με εξαιρετικά γοργούς ρυθμούς, με την σύμπραξη πολλών και διαφορετικών οργανώσεων. Οι περισσότερες από αυτές διαθέτουν διαφορετικές τεχνολογίες υλικού και λογισμικού και είναι εφοδιασμένες με διαφορετικές εσωτερικές πολιτικές. Αυτός είναι ένας από τους βασικούς λόγους για τον οποίο οι προσομοιώσεις δικτύου απαιτούν πάντοτε ανοικτές πλατφόρμες οι οποίες θα πρέπει να είναι επεκτάσιμες ώστε να συμπεριλάβουν διαφορετικές προσπάθειες και διαφορετικά τμήματα, στις προσομοιώσεις του συνόλου του δικτύου[47].

Το διαδίκτυο έχει επίσης ένα ιδιαίτερο χαρακτηριστικό που απορρέει από το πρότυπο OSI: είναι δομημένο πάνω σε ένα σύνολο πρωτοκόλλων (όπως το πρωτόκολλο IP), γεγονός που έχει σαν συνέπεια, όλες οι διαφορετικές τεχνολογίες να μπορούν να υλοποιηθούν με διαφορετικό τρόπο, αλλά με μια προτυποποιημένη επικοινωνία μεταξύ τους. Έτσι, τα εργαλεία προσομοίωσης δικτύου πρέπει να είναι σε θέση να ενσωματώσουν όλες αυτές τις δυνατότητες και να επιτρέψουν πιθανά μελλοντικά νέα πακέτα να συμπεριληφθούν στο δίκτυο και να προσομοιωθούν με διαφανή τρόπο χωρίς να επηρεάζουν αναίτια υπάρχοντα στοιχεία του δικτύου.

Οι προσομοιωτές δικτύου σήμερα, αξιοποιούνται από μια πληθώρα ατόμων με διαφορετικά επαγγελματικά ενδιαφέροντα, όπως ακαδημαϊκοί ερευνητές, βιομηχανικοί προγραμματιστές, καθώς επίσης και ερευνητές διασφάλισης ποιότητας κατά το σχεδιασμό, την προσομοίωση, την επαλήθευση, και την ανάλυση των επιδόσεων διαφόρων δικτυακών πρωτοκόλλων. Μπορούν επίσης να χρησιμοποιηθούν για να αξιολογηθεί η επίδραση των διαφόρων παραμέτρων στα υπό μελέτη πρωτόκολλα. Γενικά τα λογισμικά προσομοίωσης των δικτύων διαθέτουν ενσωματωμένο ένα ευρύ φάσμα τεχνολογιών δικτύωσης και πρωτοκόλλων ώστε να είναι σε θέση να συνδράμουν τους χρήστες κατά τη δημιουργία πολύπλοκων δικτυακών δομών με χρήση κάποιων βασικών δομικών μονάδων όπως κόμβοι, συστάδες κόμβων, συστάδες συνδέσεων κοκ. Με τη βοήθειά τους, μπορεί κανείς να πραγματοποιήσει το σχεδιασμό διάφορων τοπολογιών δικτύου χρησιμοποιώντας διάφορους τύπους κόμβων, όπως clients, nodes, network bridges, routers[43].

Από την παραπάνω σύντομη ανάλυση διαπιστώνουμε την σημασία που έχουν οι προσομοιωτές δικτύων στην σύγχρονη τεχνολογία, καθώς μπορούν να προσφέρουν σημαντικότερη βοήθεια κατά τον σχεδιασμό και την εφαρμογή νέων

τεχνικών σε όλα τα επίπεδα (πρωτόκολλα, τοπολογίες, πολιτικές κοκ) ενός δικτύου. Τα τελευταία χρόνια, σημαντική ανάπτυξη παρουσιάζουν και τα δίκτυα που αποτελούνται από ασύρματους αισθητήρες. Οι προσομοιωτές που παρουσιάζουμε στη συνέχεια έχουν εφαρμογή σε κάθε δίκτυο, και ως εκ τούτου, αποτελούν εργαλεία στην αιχμή του δόρατος κατά την ανάπτυξη νέων δικτυακών τεχνολογιών.

1.2 Προσομοίωση δικτύων

Σε αυτό το υποκεφάλαιο συζητάμε μεθόδους και τεχνικές για την προσομοίωση των λειτουργιών συστημάτων δικτύων υπολογιστών και εφαρμογών δικτύου σε πραγματικό περιβάλλον. Η προσομοίωση είναι μια από τις πιο ευρέως χρησιμοποιούμενες τεχνικές στο σχεδιασμό και τη διαχείριση δικτύων για την πρόβλεψη της απόδοσης ενός δικτύου ή τον έλεγχο ενός δικτύου πριν το δίκτυο εφαρμοστεί σε πραγματικό περιβάλλον.

1.2.1 Τύποι προσομοίωσης

Συστήματα, μοντέλα και προσομοίωση διακριτών συμβάντων

Ένα σύστημα δικτύου είναι ένα σύνολο στοιχείων, όπως δρομολογητές, διακόπτες, συνδέσεις, χρήστες και εφαρμογές που εργάζονται από κοινού για την επίτευξη ορισμένων εργασιών. Μια μελέτη προσομοίωσης μπορεί να είναι για μόνο ένα σύστημα που είναι απλά ένα μέρος ενός άλλου συστήματος, όπως στην περίπτωση των υποδικτύων. Η κατάσταση ενός δικτύου είναι το σύνολο των σχετικών μεταβλητών και των παραμέτρων που περιγράφουν το σύστημα σε ένα ορισμένο χρονικό διάστημα. Για παράδειγμα, εάν μας ενδιαφέρει η αξιοποίηση ενός συνδέσμου, θέλουμε να γνωρίζουμε μόνο τον αριθμό των bits που μεταδίδονται μέσω αυτού του συνδέσμου και η συνολική χωρητικότητα της σύνδεσης.

Αντί να οικοδομήσουμε ένα φυσικό μοντέλο ενός δικτύου, χτίζουμε ένα μαθηματικό μοντέλο που αντιπροσωπεύει τις λογικές και ποσοτικές σχέσεις μεταξύ των στοιχείων του δικτύου. Με την αλλαγή των σχέσεων μεταξύ των στοιχείων του δικτύου, μπορούμε να αναλύσουμε το μοντέλο χωρίς την κατασκευή του δικτύου φυσικά, υποθέτοντας ότι το μοντέλο συμπεριφέρεται παρόμοια με το πραγματικό σύστημα, στην περίπτωση φυσικά που αναφερόμαστε σε ένα έγκυρο μοντέλο [45].

Για παράδειγμα, μπορούμε να υπολογίσουμε τη χρήση ενός αναλυτικού στοιχείου σύνδεσης χρησιμοποιώντας τον τύπο $U = D / T$, όπου D είναι η ποσότητα των δεδομένων που αποστέλλονται σε έναν ορισμένο χρόνο και το T είναι η χωρητικότητα της σύνδεσης σε bits ανά δευτερόλεπτο. Αυτό είναι ένα πολύ απλό μοντέλο που λειτουργεί πολύ σπάνια σε προβλήματα του πραγματικού κόσμου. Δυστυχώς, η πλειοψηφία των προβλημάτων του πραγματικού κόσμου είναι πολύ περίπλοκη για να απαντηθούν χρησιμοποιώντας απλές μαθηματικές εξισώσεις. Σε πολύ περίπλοκες περιπτώσεις, η προσομοίωση είναι πιο κατάλληλη.

Τα μοντέλα προσομοίωσης μπορούν να ταξινομηθούν με πολλούς τρόπους. Οι πιο συνηθισμένες ταξινομήσεις είναι οι εξής:

Στατικά και δυναμικά μοντέλα προσομοίωσης: Ένα στατικό μοντέλο χαρακτηρίζει ένα σύστημα ανεξάρτητο του χρόνου. Ένα δυναμικό μοντέλο αντιπροσωπεύει ένα σύστημα που αλλάζει με την πάροδο του χρόνου.

Διακριτά και συνεχή μοντέλα: Ένα συνεχές μοντέλο αντιπροσωπεύει ένα σύστημα που αλλάζει συνεχώς με την πάροδο του χρόνου. Παραδείγματα είναι διαφορικές εξισώσεις που καθορίζουν τις σχέσεις για την έκταση της αλλαγής κάποιων μεταβλητών κατάστασης, παράλληλα με την αλλαγή του χρόνου. Ένα διακριτό μοντέλο χαρακτηρίζει ένα σύστημα όπου μεταβάλλονται οι μεταβλητές κατάστασης σε διακριτά σημεία στο χρόνο. Σε αυτά τα διακριτά χρονικά σημεία μπορεί να συμβεί κάποιο γεγονός ή συμβάντα, αλλάζοντας την κατάσταση του

συστήματος. Για παράδειγμα, η άφιξη ενός πακέτου.

Λόγω της σύνθετης φύσης της επικοινωνίας υπολογιστών, τα δικτυακά μοντέλα τείνουν να γίνονται όλο και πιο περίπλοκα. Η ανάπτυξη ειδικών προγραμμάτων ηλεκτρονικών υπολογιστών για ένα συγκεκριμένο πρόβλημα προσομοίωσης είναι μια πιθανότητα, αλλά μπορεί να είναι πολύ χρονοβόρα και αναποτελεσματική. Πρόσφατα, η εφαρμογή των πακέτων προσομοίωσης και μοντελοποίησης έχει γίνει πιο συνηθισμένη, εξοικονομώντας έτσι χρόνο και επιτρέποντας στο μοντέλο να επικεντρωθεί στο πρόβλημα μοντελοποίησης αντί στις λεπτομέρειες του προγραμματισμού. Με την πρώτη ματιά, η χρήση μιας τέτοιας προσομοίωσης για πακέτα, όπως COMNET, OPNET, κλπ, δημιουργεί τον κίνδυνο ότι ο σχεδιαστής πρέπει βασίζεται σε τεχνικές μοντελοποίησης και κρυφές διαδικασίες που μπορεί να είναι αποκλειστικές και μη διαθέσιμες προς βελτίωση από τους χρήστες.

1.3 Η ανάγκη για μοντελοποίηση και προσομοίωση δικτύων

1.3.1 Προσομοίωση έναντι εξομοίωσης

Σε έναν κόσμο με όλο και περισσότερα δεδομένα, υπολογιστές, συστήματα και δίκτυα, ο σχεδιασμός και η διαχείριση των συστημάτων γίνεται ολοένα και πιο

δύσκολο έργο. Καθώς τα δίκτυα γίνονται ταχύτερα, μεγαλύτερα και πιο σύνθετα, οι παραδοσιακοί στατικοί υπολογισμοί δεν είναι πια κατάλληλοι και οι προσεγγίσεις δεν είναι επαρκής. Απαιτήθηκε ένα σχέδιο ενός νέου σχεδιασμού δικτύου. Οι σύνθετοι στατικοί υπολογισμοί και τα υπολογιστικά φύλλα δεν είναι πλέον κατάλληλα εργαλεία λόγω της στοχαστικής φύσης του δικτύου. Οι οργανισμοί και οι επιχειρήσεις εξαρτώνται όλο και περισσότερο από τις νέες τεχνολογίες δικτύων για την υποστήριξη των επιχειρηματικών αναγκών τους. Ως αποτέλεσμα, η κακή απόδοση ενός δικτύου μπορεί να έχει σοβαρές επιπτώσεις στην επιτυχή λειτουργία των επιχειρήσεων [46].

Προκειμένου να αξιολογηθούν οι διάφορες εναλλακτικές λύσεις για το σχεδιασμό δικτύων, οι σχεδιαστές βρίσκουν συνεχώς νέες μεθόδους που τους βοηθούν να αξιολογήσουν τις διάφορες προτάσεις σχεδιασμού πριν από την τελική απόφαση και εφαρμογή. Μια ευρέως αποδεκτή μέθοδος είναι η πρόβλεψη απόδοσης μέσω της προσομοίωσης. Ένα μοντέλο προσομοίωσης μπορεί να χρησιμοποιηθεί από ένα σχεδιαστή δικτύου για την ανάλυση των εναλλακτικών λύσεων και να για να μελετήσει τη συμπεριφορά ενός νέου συστήματος ή τις τροποποιήσεις σε ένα υπάρχον σύστημα χωρίς να το κατασκευάσει φυσικά. Ένα μοντέλο προσομοίωσης μπορεί επίσης να αναλύσει την τοπολογία του δικτύου και των εργασιών που εκτελούνται σε ένα δίκτυο με σκοπό την απόκτηση στατιστικών αποτελεσμάτων σχετικά με την απόδοση του δικτύου.

Στην προσομοίωση, ο στόχος είναι να παραχθεί στατιστικά μοντέλο που να αντιπροσωπεύει στατιστικά τη συμπεριφορά ορισμένων στοιχείων του δικτύου και

των λειτουργιών τους. Στην προσομοίωση διακριτών συμβάντων, θέλουμε να παρατηρήσουμε τα γεγονότα που συμβαίνουν με την πάροδο του χρόνου, και να συλλέξουμε τα μέτρα απόδοσης ώστε να βγει κάποιο συμπέρασμα σχετικά με την απόδοση του δικτύου.

Η προσομοίωση μεγάλων δικτύων με πολλά στοιχεία μπορεί να οδηγήσει σε ένα μεγάλο μοντέλο το οποίο είναι δύσκολο να αναλυθεί λόγω του μεγάλου αριθμού στατιστικών στοιχείων που παράγονται κατά τη διάρκεια της προσομοίωσης. Ως εκ τούτου, είναι θεμιτό ο προγραμματιστής να διαμορφώσει μόνο εκείνα τα τμήματα του δικτύου που είναι σημαντικά για τα στατιστικά στοιχεία που πρόκειται να λάβει από την προσομοίωση.

Είναι σημαντικό να ενσωματωθούν μόνο εκείνες οι λεπτομέρειες που είναι σημαντικές για τους στόχους της προσομοίωσης. Κατά τη διαδικασία ο προγραμματιστής συχνά θέτει τους ακόλουθους στόχους:

- Να παράγει εναλλακτικές προτάσεις σχεδιασμού
 - Η προσθήκη νέων χρηστών, νέων εφαρμογών ή νέων στοιχείων δικτύου.
 - Να βρει τα προβλήματα του προηγούμενου μοντέλου και να τα απαλείψει
- [47]

Ένα άλλο σημαντικό ζήτημα είναι ο βαθμός λεπτομέρειας στον οποίο ένα στοιχείο του δικτύου έχει ως πρότυπο. Για παράδειγμα, πρέπει να αποφασίσουμε αν θέλουμε να μοντελοποιήσουμε την εσωτερική αρχιτεκτονική ενός δρομολογητή ή θέλουμε να μοντελοποιήσουμε ένα ολόκληρο δίκτυο πακετομεταγωγής. Στην πρώτη περίπτωση, θα πρέπει να καθορίσουμε τα εσωτερικά εξαρτήματα του δρομολογητή, και τις αλληλεπιδράσεις μεταξύ των στοιχείων του δρομολογητή.

Αν όμως ο στόχος είναι να αναλύσουμε το χρόνο απόκρισης σε ολόκληρο το δίκτυο μεταγωγής πακέτων, πρέπει να καθορίσουμε τους τύπους των εφαρμογών και πρωτοκόλλων, την τοπολογία των δυνατοτήτων του δικτύου και τη σύνδεση, παρά μάλλον τα εσωτερικά στοιχεία των δρομολογητών. Γενικά, η ανάλυση στις λειτουργίες χαμηλού επιπέδου δεν συμβάλλει σημαντικά στα αποτελέσματα της προσομοίωσης, όταν πρόκειται για έναν ολόκληρο δίκτυο. Η μοντελοποίηση των λεπτομερειών των εσωτερικών λειτουργιών των δρομολογητών μικρής τάξης μεγέθους χρόνου (nanosecond) δεν συμβάλλει σημαντικά στην ανάλυση καθυστέρησης από το ένα άκρο στο άλλο άκρο στην υψηλότερη τάξη μεγέθους των μικροδευτερολέπτων ή δευτερολέπτων.

Η απλοποίηση μπορεί επίσης να γίνει με την εφαρμογή στατιστικής. Η εκτενής ανάλυση μπορεί να δημιουργήσει περισσότερα προβλήματα παρά να τα λύσει σε ένα ικανοποιητικό χρόνο. Αντίθετα, μια στατιστική λειτουργία μπορεί να χρησιμοποιηθεί για να αποφασιστεί πότε ένα στοιχείο έχει υποστεί βλάβη ή έχει χαθεί. Οι λεπτομέρειες ενός στοιχείου δεν χρειάζεται να προσδιοριστούν προκειμένου να υποβληθούν σε εξέταση σφάλματος [43].

Αυτά τα παραδείγματα αποδεικνύουν ότι ο στόχος της προσομοίωσης του δικτύου δεν είναι να αναπαράγει τη λειτουργικότητα του δικτύου, αλλά περισσότερο να μιμηθεί τις κυριότερες λειτουργίες του και να τις αναλύσει με στατιστική.

1.3.2 Τύποι δικτύων επικοινωνιών, δομές μοντελοποίησης

Ένα δίκτυο επικοινωνιών αποτελείται από στοιχεία δικτύου που ονομάζουμε κόμβους (αποστολείς και δέκτες) και μέσων σύνδεσης επικοινωνίας. Μεταξύ πολλών κριτηρίων για την ταξινόμηση των δικτύων χρησιμοποιούμε δύο: την τεχνολογία μετάδοσης και την κλίμακα. Η κλίμακα ή η απόσταση καθορίζει επίσης την τεχνική που χρησιμοποιείται σε ένα δίκτυο: ενσύρματη ή ασύρματη. Η σύνδεση μεταξύ δύο ή περισσότερων δικτύων ονομάζεται internetwork. Η πιο ευρέως διαδεδομένη σύνδεση μεταξύ δικτύων είναι το Διαδίκτυο.

Σύμφωνα με την τεχνολογία μετάδοσης μπορούμε να ορίσουμε δύο βασικές κατηγορίες:

Broadcast networks: Τα δίκτυα που επικοινωνούν στέλνοντας πακέτα που έχουν ληφθεί από όλους τους άλλους κόμβους. Η διεύθυνση του πακέτου καθορίζει τον παραλήπτη ή τους παραλήπτες του πακέτου. Μόνο ο παραλήπτης (οι παραλήπτες) θα επεξεργαστούν το πακέτο.

Point to point networks: Σε αυτή την περίπτωση το πακέτο που αποστέλλεται από μια πηγή σε έναν προορισμό ενδέχεται να πρέπει πρώτα να διασχίσει τους ενδιάμεσους κόμβους όπου αποθηκεύεται και προωθείται έως ότου φτάσει στον τελικό προορισμό [44].

Με έναν άλλο τρόπο ταξινόμησης, κατηγοριοποιούμε με βάση την έκταση κάλυψης, όπου διαχωρίζονται σε:

PANs: Όπου είναι δίκτυα που οι δράσεις τους είναι προορισμένες για το σπίτι, έναν όροφο ή ένα κτίριο, και γενικά καλύπτουν μια περιορισμένη γεωγραφική περιοχή. Αποτελούνται από βασικά στοιχεία, όπως το ποντίκι, το πληκτρολόγιο κτλ.

LANs: Τα δίκτυα LAN συνδέουν προσωπικά δίκτυα σε χώρους εργασίας, διακομιστές και κοινόχρηστους πόρους. Τα δίκτυα LAN μπορούν να ταξινομηθούν περαιτέρω με βάση την τεχνολογία μετάδοσης, την ταχύτητα που μετράται σε δυαδικά ψηφία ανά δευτερόλεπτο και την τοπολογία. Οι ταχύτητες μετάδοσης κυμαίνονται από τις παραδοσιακές 10Mbps στα τοπικά δίκτυα μέχρι τα σημερινά 10Gbps.

MANs: Τα δίκτυα αυτά αναφέρονται σε εκτάσεις πόλεων ή μεγάλων περιοχών και καλύπτουν τηλεοπτικά προγράμματα αλλά και υπηρεσίες Internet, αλλά έχουν ένα

μεγάλο τμήμα του φάσματος εκπομπής αχρησιμοποίητο. Άλλες τεχνολογίες MAN είναι οι Fiber Distributed Data Interface (FDDI) και IEEE ασύρματες τεχνολογίες που εκτενέστερα θα αναλύσουμε παρακάτω[45].

WANs: Ένα WAN αποτελείται από κεντρικούς υπολογιστές (πελάτες και διακομιστές) που συνδέονται με δευτερεύοντα δίκτυα που ανήκουν σε φορείς παροχής υπηρεσιών επικοινωνιών. Τα υποδίκτυα παραδίδουν μηνύματα από τον κεντρικό υπολογιστή προέλευσης στον κεντρικό υπολογιστή προορισμού. Ένα υποδίκτυο μπορεί να περιέχει πολλές γραμμές μεταφοράς, καθεμιά από τις οποίες συνδέει ένα ζευγάρι εξειδικευμένων συσκευών που ονομάζονται δρομολογητές. Οι γραμμές μεταφοράς είναι κατασκευασμένες από διάφορα μέσα όπως χάλκινο σύρμα, οπτικές ίνες, ασύρματες συνδέσεις, κλπ. Όταν ένα μήνυμα αποστέλλεται σε έναν κεντρικό υπολογιστή προορισμού, το μήνυμα χωρίζεται σε μικρότερα κομμάτια, που ονομάζονται πακέτα. Όταν ένα πακέτο φθάνει σε μια εισερχόμενη γραμμή μετάδοσης, το router αποθηκεύει το πακέτο πριν να επιλέγει μια εξερχόμενη γραμμή και προωθεί το πακέτο μέσω της εν λόγω γραμμής. Η επιλογή της εξερχόμενης γραμμής βασίζεται σε έναν αλγόριθμο δρομολόγησης. Τα πακέτα παραδίδονται στον δέκτη και παρέχεται ένας και μοναδικός προορισμός όπου μεταδίδεται το αρχικό μήνυμα.

Τα ασύρματα δίκτυα μπορούν να κατηγοριοποιηθούν ως ασύρματα δίκτυα μικρών αποστάσεων, ασύρματα LAN, και ασύρματα δίκτυα WAN.

Τα ασύρματα δίκτυα μικρών αποστάσεων περιέχουν Global Positioning System (GPS), ακουστικά, ηλεκτρονικούς υπολογιστές, σαρωτές, οθόνες, και πληκτρολόγια που συνδέονται μέσω συνδέσεων μικρής εμβέλειας, δηλαδή με ακτίνα 20- 30 πόδια. Η κύρια μονάδα συστήματος ελέγχει τις λειτουργίες των δευτερευόντων στοιχείων. Η κύρια μονάδα επίσης καθορίζει τις διευθύνσεις που θα χρησιμοποιούν οι δευτερεύουσες συσκευές, καθώς και το πότε και με ποια συχνότητα θα μεταδώσει αυτές τις συχνότητες.

Μια ασύρματη LAN σύνδεση απαιτεί τουλάχιστον ένα ασύρματο modem και μία κεραία για αποστολή και λήψη στοιχείων. Οι υπολογιστές επικοινωνούν μεταξύ τους άμεσα σε διαμόρφωση peer-to-peer ή μέσω του σημείου πρόσβασης που συνδέει τους υπολογιστές σε άλλα δίκτυα. Η τυπική περιοχή κάλυψης είναι περίπου 300 πόδια. Το πρωτόκολλο ασύρματου LAN καθορίζεται στην οικογένεια των προτύπων IEEE 802.11 για ταχύτητες από 11 Mbps έως 108 Mbps.

Τα ασύρματα WANs έχουν εύρος ικανό για ραδιοφωνικά δίκτυα ή για τα κινητά τηλέφωνα και έχουν εξελιχθεί μέσα από τρεις γενιές. Πλέον, θεωρείται ότι έχουμε περάσει στην τέταρτη γενιά. Η πρώτη γενιά σχεδιάστηκε μόνο για φωνητικές επικοινωνίες που χρησιμοποιούν αναλογική σήμανση. Η δεύτερη γενιά διαβίβαζε επίσης μόνο φωνή, αλλά με βάση ψηφιακή τεχνολογία μετάδοσης. Η τρίτη γενιά είναι ψηφιακή και μεταδίδει τόσο φωνή όσο και δεδομένα της τάξης των 2Mbps. Τα δίκτυα WAN υψηλής ταχύτητας παρέχουν πρόσβαση υψηλής ταχύτητας από σπίτια και επιχειρήσεις παρακάμπτοντας τα τηλεφωνικά συστήματα. Το IEEE 802.16 λειτουργεί σε πολύ υψηλότερο εύρος συχνοτήτων, από 10 έως 66 GHz. Τα

οικιακά δίκτυα πλέον μπορούν να ενσωματώσουν υπολογιστές, φορητούς υπολογιστές, PDAs, τηλεοράσεις, DVD, βιντεοκάμερες, συσκευές αναπαραγωγής MP3, φούρνους μικροκυμάτων, ψυγεία, A / C, φώτα, συναγερμούς, κλπ. Πολλά σπίτια είναι ήδη εξοπλισμένα με υψηλής ταχύτητας πρόσβαση στο ίντερνετ μέσω του οποίου μπορούν να δημιουργήσουν ικανοποιητικά δίκτυα.

1.3.3 Η απόδοση ενός δικτύου και τα χαρακτηριστικά του

Σε αυτή την ενότητα θα συζητήσουμε μια μικρή λίστα των χαρακτηριστικών ενός δικτύου στα οποία είναι προσανατολισμένοι οι στόχοι της στατιστικής ανάλυσης, του σχεδιασμού και βελτιστοποίησης των δικτύων των υπολογιστών. Ουσιαστικά, τα μοντέλα δικτύων κατασκευάστηκαν από τον ορισμό της στατιστικής κατανομής του ρυθμού άφιξης και εξυπηρέτησης σε ένα σύστημα αναμονής που στη συνέχεια εξελίσσει αυτά τα χαρακτηριστικά.

Στη συνέχεια θα αναλύσουμε κάποια από τα κυριότερα αυτά χαρακτηριστικά, τα οποία θα χρησιμοποιηθούν και στα επόμενα κεφάλαια της εργασίας:

Ικανότητα ενός καναλιού ή συνδέσμου

Ικανότητα καναλιού ή συνδέσμου είναι ο αριθμός των μηνυμάτων που χειρίζεται ένας σύνδεσμος ανά μονάδα χρόνου. Συνήθως μετριέται σε δυαδικά ψηφία ανά

δευτερόλεπτο. Ένα από τα πιο διάσημα αποτελέσματα της θεωρίας πληροφοριών είναι το θεώρημα του Shannon για ένα κανάλι κωδικοποίησης: «Για ένα δεδομένο κανάλι υπάρχει ένας κωδικός που θα επιτρέψει τη μετάδοση χωρίς σφάλματα σε όλο το κανάλι με ρυθμό R , με την προϋπόθεση $R < C$, όπου C είναι η χωρητικότητα του καναλιού». Η ισότητα επιτυγχάνεται μόνο όταν ο λόγος σήματος προς θόρυβο (SNR) είναι άπειρος.

Εύρος ζώνης

Το εύρος ζώνης είναι η διαφορά μεταξύ υψηλότερων και χαμηλότερων συχνοτήτων που χρησιμοποιούνται για τα σήματα που μεταδίδει ένα δίκτυο. Το εύρος ζώνης είναι επίσης ένας όρος που χρησιμοποιείται για να περιγράψει τη χωρητικότητα διακίνησης ενός συγκεκριμένου συνδέσμου ή πρωτοκόλλου που μετράται σε Kilobits, Megabits, Gigabits, Terabits κ.λπ., σε ένα δευτερόλεπτο.

Χρόνος απόκρισης

Ο χρόνος απόκρισης είναι ο χρόνος που χρειάζεται ένα σύστημα δικτύου για να αντιδράσει σε ένα σήμα μιας συγκεκριμένης πηγής. Ο χρόνος απόκρισης περιλαμβάνει το χρόνο μετάδοσης στον προορισμό, τη διαδικασία στο τόσο την πηγή και όσο τον προορισμό και τα ενδιάμεσα στοιχεία του δικτύου κατά μήκος της διαδρομής, όπως επίσης και το χρόνο μετάδοσης πίσω στην πηγή. Ο μέσος χρόνος απόκρισης είναι ένα σημαντικό μέτρο απόδοσης ενός δικτύου. Για τους

χρήστες, όσο μικρότερος είναι ο χρόνος απόκρισης τόσο το καλύτερο. Οι στατιστικές τιμές του χρόνου απόκρισης (μέσος όρος και διακύμανση) πρέπει να είναι σταθερές. Εξετάζουμε και μέσο χρόνο απόκρισης αλλά και διακύμανση γιατί ένας χαμηλός μέσος χρόνος απόκρισης δεν εγγυάται ότι δεν υπάρχουν και εξαιρετικά μεγάλοι χρόνοι απόκρισης, συνήθως εξαιτίας συμφόρησης ενός δικτύου.

Καθυστέρηση

Καθυστέρηση είναι η ποσότητα του χρόνου που χρειάζεται για ένα πακέτο δεδομένων να μεταδοθεί σε ένα σύνδεσμο του δικτύου. Η καθυστέρηση και το εύρος ζώνης είναι οι δύο παράγοντες που καθορίζουν την ταχύτητα ενός συνδέσμου[45].

Η καθυστέρηση περιλαμβάνει την καθυστέρηση διάδοσης (ο χρόνος που απαιτείται για τα ηλεκτρικά ή οπτικά σήματα να ταξιδέψουν την απόσταση μεταξύ δύο σημείων) και το χρόνο επεξεργασίας. Για παράδειγμα, η καθυστέρηση μεταξύ ενός σταθμού που βρίσκεται σε δορυφόρο με έναν άλλο επίγειο σταθμό (πάνω από 34.000 km την διαδρομή) είναι περίπου 270 χιλιοστά του δευτερολέπτου. Η καθυστέρηση μεταξύ της ανατολικής και δυτικής ακτής των ΗΠΑ είναι περίπου 100ms και η παγκόσμια, δηλαδή του γύρου της Γης είναι περίπου 125ms. Η καθυστέρηση επηρεάζεται από την ταχύτητα του σήματος, αλλά επίσης και από τις συσκευές του δικτύου, δρομολογητές, διακόπτες, την απώλεια σήματος, τυχαίες

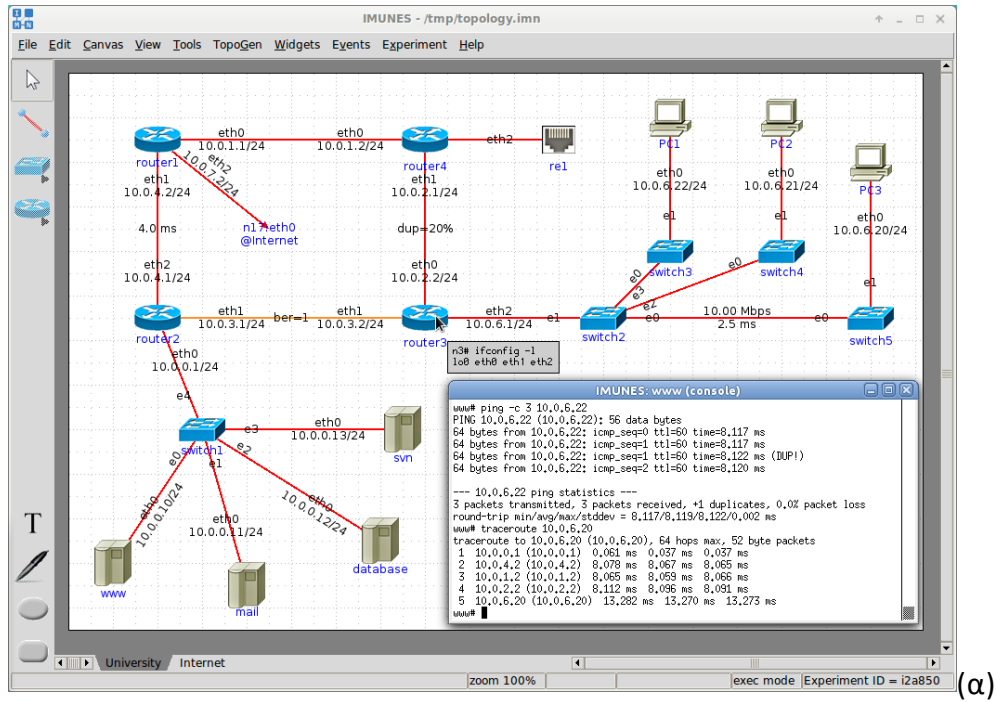
βλάβες συσκευών και σύνδεσης και την υπερφόρτωση.

Η διαδρομή σε ένα τοπικό δίκτυο δεν αποτελεί κρίσιμο ζήτημα, διότι υπάρχει μόνο μία διαδρομή από οποιαδήποτε πηγή σε οποιονδήποτε προορισμό. Όταν το δίκτυο συνδέει διάφορα άλλα δίκτυα και αποτελείται από πολλά μονοπάτια, routers, και συνδέσμους βρίσκοντας την καλύτερη διαδρομή, τότε η διαδρομή γίνεται σημαντικός παράγοντας.

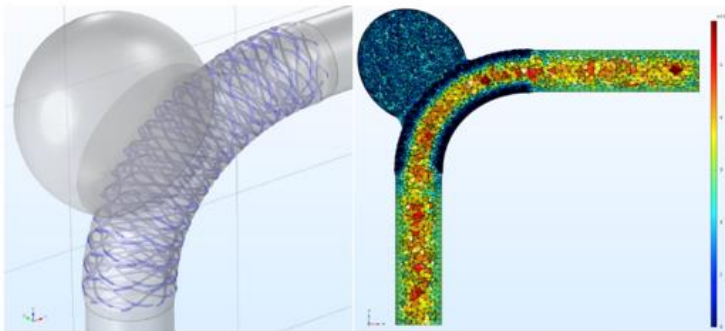
Πρωτόκολλα δρομολόγησης

Η διαδρομές καθορίζονται από τα πρωτόκολλα δρομολόγησης. Ο στόχος των πρωτοκόλλων δρομολόγησης είναι να βρεθεί μια βέλτιστη ή σχεδόν βέλτιστη διαδρομή μεταξύ πηγή και προορισμού, αποφεύγοντας τις συμφόρηση.

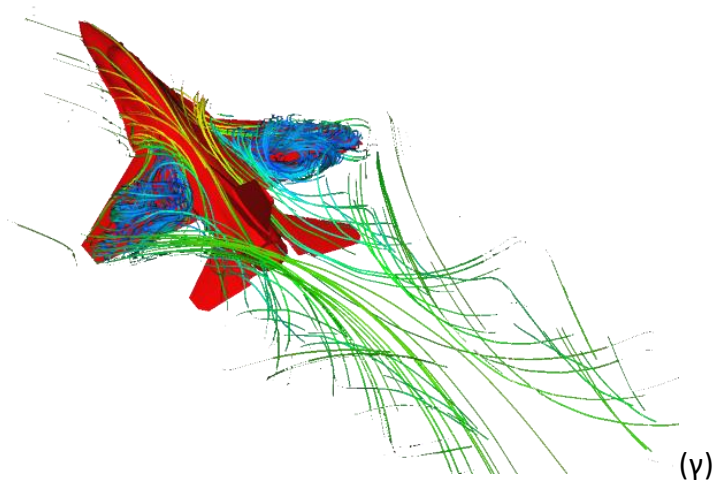
Τα μηνύματα για κάθε πρωτόκολλο και τα δεδομένα της εφαρμογής είναι ενσωματωμένα στο εσωτερικό των μονάδων δεδομένων πρωτοκόλλου, όπως τα πακέτα. Ένα κύριο ενδιαφέρον των σχεδιαστών δικτύων είναι η επιβάρυνση των πρωτοκόλλων. Πόσο γρήγορα μπορούμε πραγματικά να μεταδώσουμε δεδομένα από μια διαδρομή επικοινωνίας και ένα πρωτόκολλο, δηλαδή, πόσο εύρος ζώνης είναι ελεύθερο για εφαρμογές. Τα περισσότερα πρωτόκολλα εισάγουν επιπλέον πρόσθετα έξοδα που σχετίζονται με λειτουργίες διαχείρισης πρωτοκόλλου[47].



(α)



(β)



Εικόνα 3: Παραδείγματα προσομοίωσης στη ζωή μας. (α) Προσομοίωση δικτύων, (β) Προσομοίωση ροής σε αρτηρία με ανεύρυσμα, (γ) Προσομοίωση αεροδυναμικής αεροσκάφους

2. Βασικές έννοιες στατιστικής

Εδώ θα ασχοληθούμε με ένα από τα πιο σημαντικά κομμάτια στη διαδικασία της προσομοίωσης με τη μέθοδο διακριτών γεγονότων, το μηχανισμό δειγματοληψίας

τιμών. Η τιμές αυτές προκύπτουν από φαινόμενα σε κατανομές της πιθανοθεωρίας.

Αρχικά, θα δούμε πώς πραγματοποιείται η παραγωγή τυχαίων αριθμών (random numbers) με φυσικά μέσα και κατόπιν τον τρόπο με τον οποίο αυτή πραγματοποιείται με τη βοήθεια ενός υπολογιστικού συστήματος, δηλαδή, αλγοριθμικά. Στη συνέχεια, θα διερευνήσουμε τον τρόπο με τον οποίο η παραγωγή τυχαίων αριθμών χρησιμοποιείται για τη δειγματοληψία παρατηρήσεων από οποιαδήποτε πιθανοθεωρητική κατανομή.

Η χρήση μιας πιθανοθεωρητικής κατανομής για την παραγωγή ενός τυχαίου γεγονότος προσφέρει δύο πλεονεκτήματα

- ο χρόνος αυτός (και γενικά η παραγόμενη τιμή της τυχαίας μεταβλητής), αν και δεν είναι εκ των προτέρων γνωστός (προσδιοριστικός), κινείται μέσα στα όρια των τιμών που υπαγορεύονται από την κατανομή (διακριτή ή συνεχή) και
- η πιθανότητα να πάρει οποιαδήποτε από τις τιμές αυτές είναι γνωστή από την κατανομή. Έτσι, το σύστημα διατηρεί τη στοχαστική του φύση, ωστόσο είμαστε σε θέση να γνωρίζουμε τις αντίστοιχες πιθανότητες για κάθε τιμή που μπορεί να δώσει κάποια απ' αυτές τις μεταβλητές.

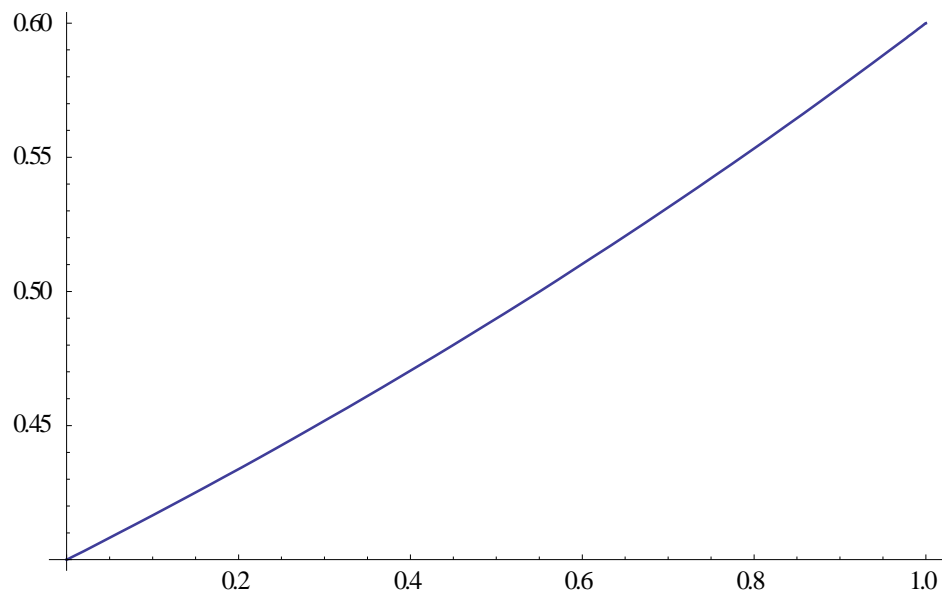
Οι πιθανοθεωρητικές κατανομές χωρίζονται σε διακριτές και συνεχείς.

Παρακάτω δίνονται οι βασικές κατανομές:

Συνεχής κατανομές

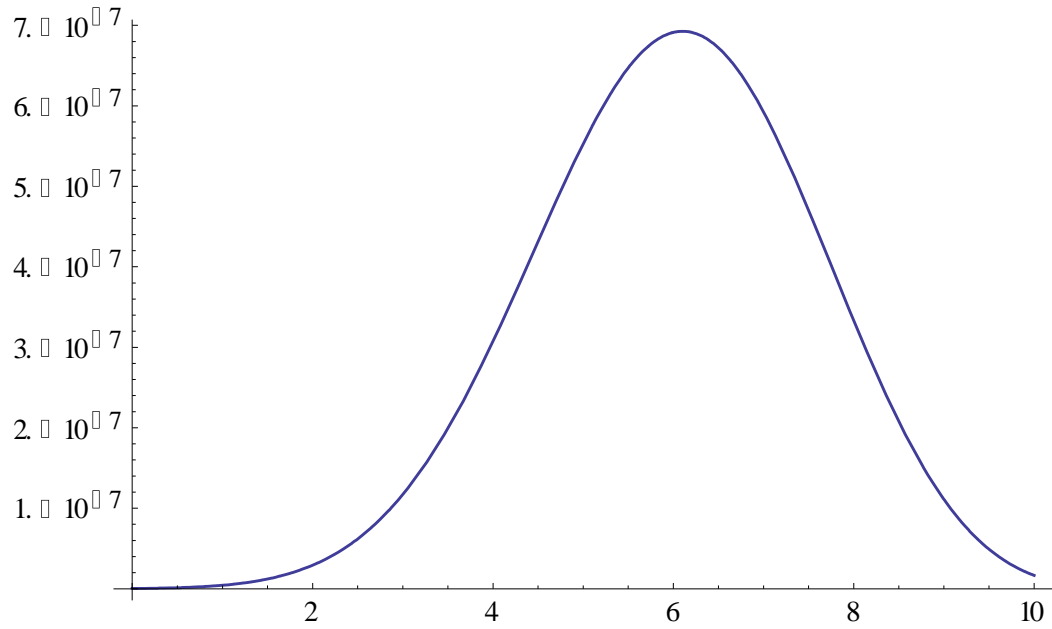
- Κατανομή Bernoulli

$$f(x) = p^x (1-p)^{1-x}, x \in \{0,1\}$$



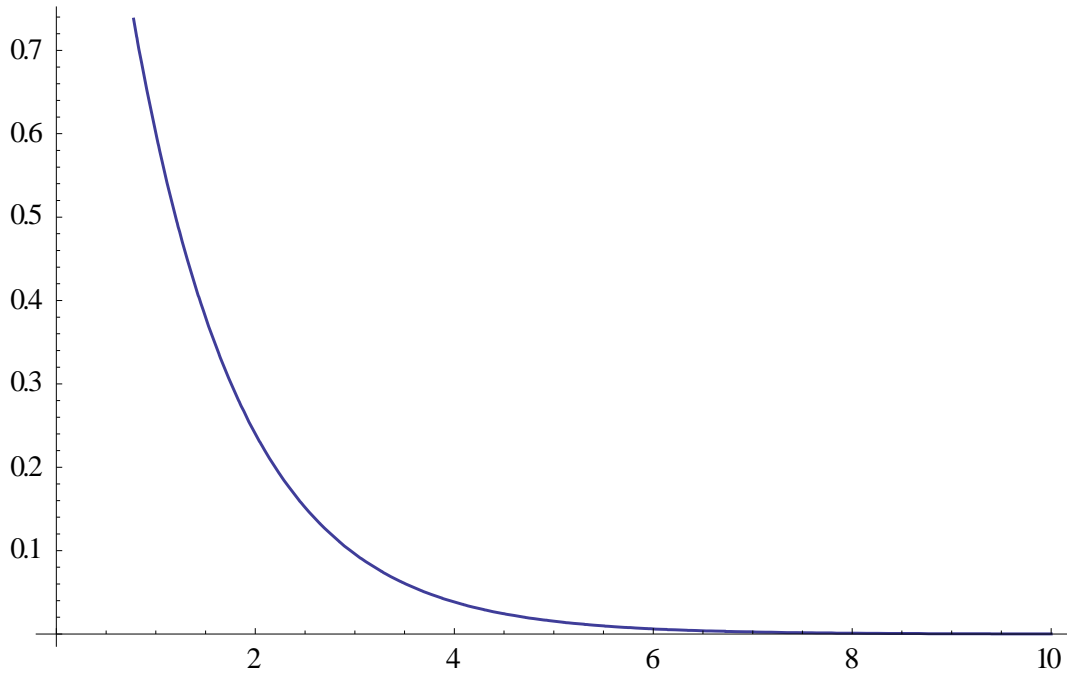
- Διωνυμική κατανομή $B(n,p)$

$$f(x) = \binom{n}{x} p^x (1-p)^{n-x}, x \in \{0,1,2,\dots,n\}$$



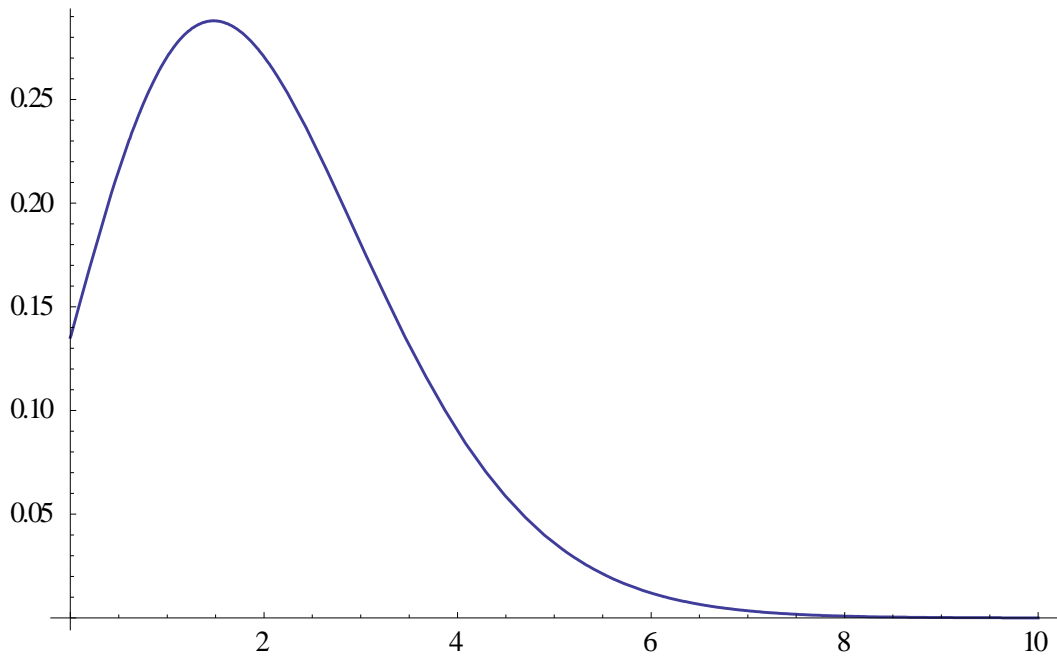
- Γεωμετρική κατανομή $G(p)$

$$f(x) = p(1-p)^{x-1}, x \in \{0, 1, 2, \dots\}$$



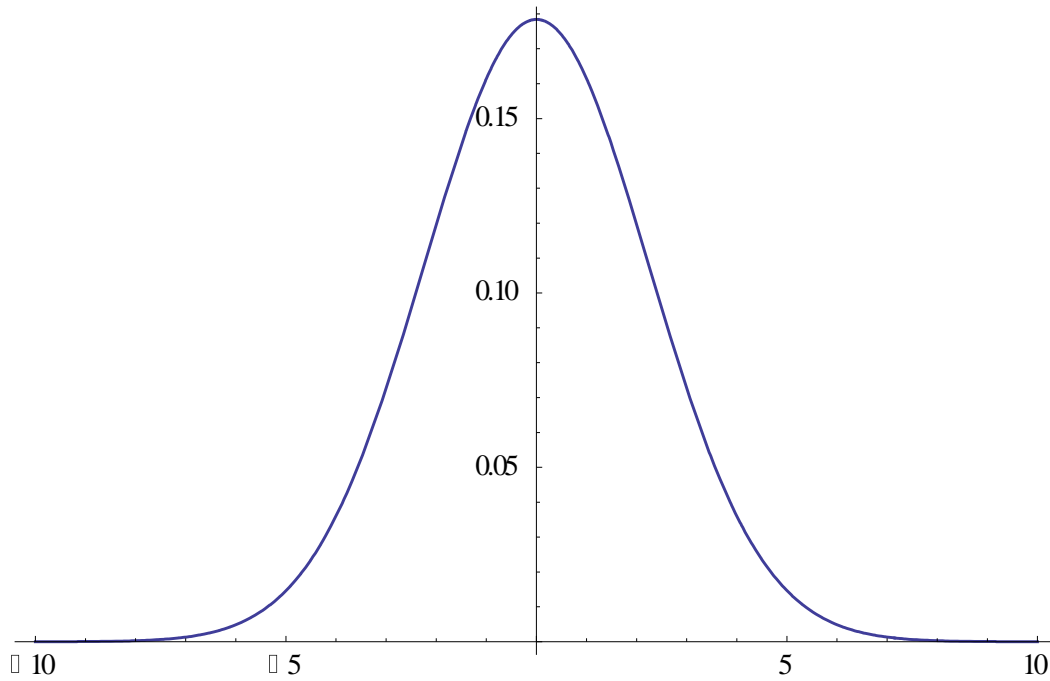
- Κατανομή Poisson $P(\lambda)$

$$f(x) = e^{-\lambda} \frac{\lambda^x}{x!}, x \in \{0, 1, 2, \dots\}$$



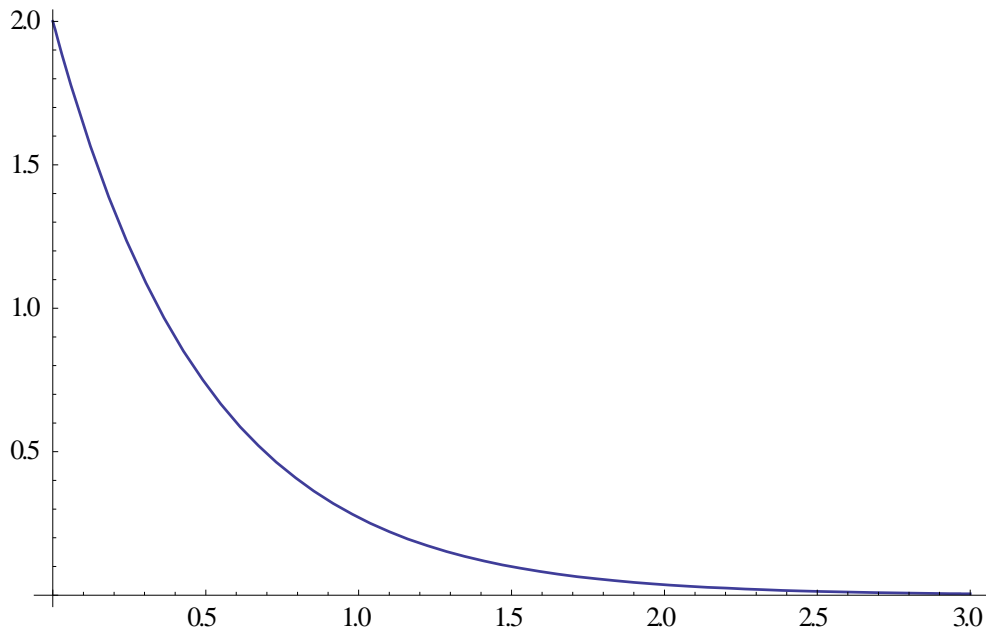
- Κανονική κατανομή $N(\mu, \sigma)$

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$



- Εκθετική κατανομή $E(\lambda)$

$$f(x) = \lambda e^{-\lambda x}$$



Η βασική μέθοδος με την οποία γίνεται παραγωγή τυχαίων αριθμών ονομάζεται μέθοδος *Monte Carlo* [4][5]. Ας δούμε πρώτα τι σημαίνει η έννοια «τυχαίος» αριθμός:

Πρέπει να παρατηρήσουμε ότι ένας μεμονωμένος αριθμός μεταξύ του 0 και του 1 δεν μπορεί να χαρακτηρίζεται (από μόνος του) ως τυχαίος. Οι τυχαίοι αριθμοί προκύπτουν ως ακολουθία (stream) από κάποιον μηχανισμό έτσι ώστε να είναι ανεξάρτητοι (και ισοπίθανοι). Τυχαίος αριθμός οπότε είναι ο αριθμός ο οποίος έχει παραχθεί από μία τυχαία ή στοχαστική διεργασία. Επομένως, μια γεννήτρια παράγει τυχαίους αριθμούς, αν δεν μπορούμε να καθορίσουμε με προσδιοριστικό

τρόπο την παρατήρηση που θα δώσει σε κάθε επανάληψη. Η παραγωγή αριθμών μεταξύ του 0 και του 1 και η αντιστοίχιση καθενός απ' αυτούς με ένα αποτέλεσμα μας οδηγεί σε όμοια αποτελέσματα.

Οι τυχαίοι αριθμοί παράγονται, λοιπόν, από φυσικές διαδικασίες, οι οποίες θεωρούνται ότι αντιστοιχούν στην ομοιόμορφη κατανομή. Άλλες διαδικασίες μπορεί να πραγματοποιούνται με ηλεκτρονικές, φυσικές ή άλλες συσκευές. Συνήθως, τέτοιες μέθοδοι δεν χρησιμοποιούνται για την απλή και τυχαία δειγματοληψία. Μερικοί λόγοι είναι ότι είναι επίπονες, δαπανηρές και δύσκολο να επαναληφθούν δίνοντας ακριβώς την ίδια ακολουθία αριθμών, όταν αυτό είναι απαραίτητο. Το τελευταίο σχόλιο αναδεικνύει ένα εξαιρετικά σημαντικό εργαλείο για την προσομοίωση, διότι μας δίνει την ευκαιρία σύγκρισης διαφορετικών πολιτικών με την ίδια ακριβώς ροή (ακολουθία) «τυχαίων» γεγονότων.

Μια άλλη ιδέα είναι να παραχθεί ακολουθία τυχαίων αριθμών με κάποια φυσική μέθοδο μία φορά, ακολούθως να σωθεί σε πίνακες/αρχεία (π.χ. σε ηλεκτρονική μορφή τοπικά ή απομακρυσμένα) και να χρησιμοποιείται όποτε υπάρχει ανάγκη. Αν, για παράδειγμα, οι αριθμοί παράγονται και αποθηκεύονται online σε ένα αρχείο (cloud storage), υπάρχει η δυνατότητα πρόσβασης ανά πάσα στιγμή στο αρχείο κάθε φορά που χρειάζεται μια τυχαία παρατήρηση η οποία θα «παράγεται» από τον έτοιμο πίνακα. Αυτή η μέθοδος μπορεί να αναπαράγει την ίδια ακολουθία «τυχαίων» αριθμών όταν αυτό είναι απαραίτητο, αλλά μειονεκτεί στο γεγονός ότι χρειάζεται κάθε φορά πρόσβαση σε απομακρυσμένο μέσο αποθήκευσης για τον εντοπισμό του κάθε επόμενου αριθμού, κάτι που μπορεί να αποτελεί πρόβλημα σε συστήματα με χαμηλή ταχύτητα πρόσβασης. Από την άλλη πλευρά, η φύλαξη του πίνακα τυχαίων αριθμών σε αρχείο που βρίσκεται κάπου στο «σύννεφο» δίνει τη δυνατότητα σε οποιονδήποτε να έχει πρόσβαση σ' αυτούς

τους αριθμούς από έναν οποιοδήποτε υπολογιστή, tablet, smartphone ή γενικά από κάποια συσκευή με πρόσβαση στο διαδίκτυο. Στο σημείο αυτό, τίθεται το θέμα ότι η ύπαρξη του προκαθορισμένου πίνακα τυχαίων αριθμών σε αρχείο καθιστά ουσιαστικά τη διαδικασία μη στοχαστική υπό την έννοια ότι γνωρίζουμε ποιοι είναι οι αριθμοί άρα και τον τρόπο - αλληλουχία εμφάνισής τους, αφού τους έχουμε καταχωρημένους σε οποιοδήποτε αρχείο. Γι' αυτό και νωρίτερα η λέξη τυχαίων χρησιμοποιήθηκε μέσα σε εισαγωγικά. Πρακτικά, το «μειονέκτημα» αυτό δεν είναι κρίσιμο γιατί όπως θα δούμε αμέσως και οι άλλες μέθοδοι, που προτείνονται στη συνέχεια, παράγουν ουσιαστικά «τυχαίους» αριθμούς- με κάποιους αλγορίθμους- καθιστώντας τους απολύτως προβλέψιμους και αναπαραγωγίσιμους. Άλλωστε, όπως τονίστηκε, η αναπαραγωγή της ίδιας ακριβώς ακολουθίας «τυχαίων αριθμών» είναι πολλές φορές αναγκαία. Η παραγωγή, λοιπόν, των τυχαίων αριθμών στον υπολογιστή μας γίνεται με τη χρήση κάποιων αναδρομικών (ή συγκλινουσών) μαθηματικών σχέσεων που ονομάζονται γεννήτριες τυχαίων αριθμών (random number generators) και είναι αυτές που θα μας απασχολήσουν στη συνέχεια.

Οι ιδιότητες, οι οποίες διέπουν τη λειτουργία των γεννητριών/αλγορίθμων, είναι εκ των προτέρων γνωστές και δεν υπάρχει τίποτα το στοχαστικό στον τρόπο λειτουργίας τους. Το βασικό, όμως, χαρακτηριστικό τους είναι ότι παράγουν τιμές (παρατηρήσεις, random variates) από την ομοιόμορφη κατανομή και στη συνέχεια από οποιαδήποτε άλλη κατανομή, και είναι παρατηρήσεις που πράγματι μοιάζουν σαν να είχαν προκύψει από μια τυχαία διαδικασία όπως είναι αυτή της κλήρωσης. Διαχωρίζουμε τους αριθμούς που παράγει μια γεννήτρια από τους πραγματικά τυχαίους αριθμούς και τους ονομάζουμε ψευδοτυχαίους αριθμούς. Επειδή δεν είναι σίγουρο ότι μια γεννήτρια παράγει ψευδοτυχαίους αριθμούς, οι ακολουθίες που παράγονται υπόκεινται σε στατιστικούς ελέγχους ως προς την τυχαιότητά

τους. Στατιστικά ελέγχεται, με κάποια στάθμη σημαντικότητας, αν μπορεί να γίνει αποδεκτή η υπόθεση ότι μία ακολουθία ψευδοτυχαίων αριθμών έχει τις ιδιότητες των πραγματικά τυχαίων αριθμών. Οι έλεγχοι τυχειότητας δεν θα αποτελέσουν αντικείμενο περαιτέρω αναφοράς στο παρόν.

Υπάρχουν, λοιπόν, πολλές μέθοδοι-γεννήτριες-μηχανισμοί παραγωγής τυχαίων αριθμών (με τη χρήση της λέξης «τυχαίων» θα εννοούμε εφεξής τους ψευδοτυχαίους αριθμούς, εκτός αν υπάρχει σαφής, διαφορετική διευκρίνιση) και μάλιστα αποτελούν ένα ενδιαφέρον αντικείμενο έρευνας του εν λόγω κλάδου. Οι συγκλίνουσες γεννήτριες (congruential generators) είναι μια κατηγορία γεννητριών στις οποίες θα αναφερθούμε στο παρόν κεφάλαιο. Αν συμβολίσουμε με x_{n+1} την $n+1$ παρατήρηση στην ακολουθία τυχαίων παρατηρήσεων από την ομοιόμορφη κατανομή $U(0,1)$, τότε οι συγκλίνουσες γεννήτριες έχουν την ακόλουθη γενική μορφή:

$$x_{n+1} = f(x_n)$$

Με $n = 0, 1, 2, \dots$

Η μεικτή συγκλίνουσα γεννήτρια (mixed congruential generator) παράγει μία ακολουθία τυχαίων αριθμών από την εξής σχέση:

$$x_{n+1} = (px_n + s) \bmod(m)$$

Δηλαδή, λαμβάνοντας υπόψιν τις ιδανικές συνθήκες, παράγει μία ακολουθία τυχαίων αριθμών:

$$\{x_n\}_{n=1}^{m-1}$$

που προκύπτει από τα υπόλοιπα των ακέραιων διαιρέσεων της παραπάνω σχέσης. Ο πρώτος αριθμός, που συμβολίζεται με x_0 , και είναι η αρχική συνθήκη της γεννήτριας ονομάζεται **γεννήτορας** ή αλλιώς *seed number*. Η επιλογή του θα μπορούσε να γίνεται τυχαία από έναν γνωστό πίνακα τυχαίων αριθμών ή από το εσωτερικό ρολόι του υπολογιστή ή όποιας συσκευής χρησιμοποιούμε. Οι αριθμοί p , s και m είναι θετικοί ακέραιοι για τους οποίους ισχύει ότι:

$$1 < p < m$$

$$0 \leq s < m$$

Ο αριθμός p λέγεται πολλαπλασιαστής, ο s λέγεται αθροιστής και ο m λέγεται διαιρέτης.

Ας δούμε κάποιες περιπτώσεις:

Έστω ότι $s = 0$.

Τότε έχουμε την πολλαπλασιαστική συγκλίνουσα γεννήτρια. Στη περίπτωση όπου ο πολλαπλασιαστής είναι 1 έχουμε και ο αθροιστής διάφορος του 0, έχουμε την αθροιστική συγκλίνουσα γεννήτρια.

Στη μεικτή συγκλίνουσα γεννήτρια, ο κάθε επόμενος ψευδοτυχαίος αριθμός στην παραγόμενη ακολουθία είναι το x_{n+1} , το οποίο προκύπτει ως το υπόλοιπο της διαίρεσης του $(px_n + s)$ με το m . Έτσι, οι πιθανές τιμές του x_{n+1} είναι μεταξύ των υπολοίπων $0, 1, 2, \dots, m-1$, οπότε το m είναι το μέγιστο πλήθος των διαφορετικών τυχαίων αριθμών. Για την παραγωγή ομοιόμορφα κατανομημένων αριθμών στο διάστημα $[0, 1]$, αρκεί να διαιρέσουμε κάθε αποτέλεσμα δια του m . Έτσι, έχουμε τον τύπο που δίνει έναν τυχαίο αριθμό Rn μεταξύ του 0 και του 1:

$$R_n = \frac{x_n}{m}$$

$$n = 1, 2, \dots, m-1$$

Το προηγούμενο αποτέλεσμα αποδίδει έναν αριθμό ο οποίος αποκόπτεται σε ένα πλήθος δεκαδικών ψηφίων έστω k , όσα χρειάζονται για το πρόβλημα.

Είναι επιθυμητό—αλλά όχι πάντα εφικτό—μια γεννήτρια να είναι ικανή να παράγει όλους τους ακέραιους αριθμούς μεταξύ του 0 και του $m-1$. Επειδή το πλήθος αυτό είναι πεπερασμένο, είναι αναπόφευκτο ότι σε κάποια χρονική στιγμή η γεννήτρια θα έχει παραγάγει ξανά έναν αριθμό που έχει εμφανιστεί και προηγουμένως με αποτέλεσμα να δημιουργηθεί στη συνέχεια η ίδια ακριβώς ακολουθία ψευδοτυχαίων αριθμών. Ο κύκλος αυτός θα επαναλαμβάνεται συνεχώς. Το μήκος αυτού του κύκλου ονομάζεται περίοδος (period) της γεννήτριας. Όταν μέσα σε έναν κύκλο παράγονται και οι m αριθμοί, η γεννήτρια έχει πλήρη περίοδο (full period). Για παράδειγμα, αυτό μπορεί να συμβεί όταν $x_m = x_0$ και $x_n \neq x_0$ για $n=1, 2, \dots, m-1$. Ιδανικά, λοιπόν, η γεννήτρια πρέπει να έχει τον μέγιστο δυνατό κύκλο και φυσικά μεγάλο διαιρέτη (συχνά δύναμη του 2 ή κάποια παραλλαγή, κάτι που αφήνουμε στον αναγνώστη να εξηγήσει γιατί), ώστε να έχει νόημα η όλη διαδικασία.

Τα πλεονεκτήματα της χρήσης των ψευδοτυχαίων αριθμών που παράγονται από μια γεννήτρια, όπως είναι η μεικτή συγκλίνουσα γεννήτρια, έναντι των πραγματικά τυχαίων αριθμών αναφέρθηκαν σποραδικά πιο πάνω και τα συνοψίζουμε αμέσως. Καταρχάς, μπορούν να παραχθούν από τον υπολογιστή με απλή χρήση των παραπάνω τύπων. Έτσι, λοιπόν, δεν χρειάζεται δέσμευση χώρου φύλαξης (είτε τοπικά σε κύρια ή βοηθητική μνήμη είτε σε δίκτυο), πόσο μάλλον η διαδικασία εντοπισμού τους. Επίσης, είναι δυνατόν να αναπαραχθεί η ίδια ακριβώς ακολουθία ψευδοτυχαίων αριθμών, αν διατηρήσουμε τις ίδιες παραμέτρους p , s και m και ξεκινήσουμε από τον ίδιο γεννήτορα. Αυτή η ιδιότητα είναι πολύ σημαντική σε ένα μοντέλο προσομοίωσης που χρησιμοποιείται για συγκρίσεις.

Είναι επιθυμητό η ακολουθία των αριθμών $\{R_n\}_{n=1}^{m-1}$ που παράγεται με τη μεικτή συγκλίνουσα γεννήτρια να έχει τις ακόλουθες ιδιότητες:

- Ο κύκλος (η περίοδος) να είναι μεγάλος, δηλαδή, το πλήθος των παρατηρήσεων που παράγονται μέχρι την επανεμφάνιση του γεννήτορα ιδανικά να πλησιάζει όσο περισσότερο γίνεται το m (κάτι που φυσικά δεν συμβαίνει συχνά). Επίσης, είναι επιθυμητό όλοι οι αριθμοί να έχουν προκύψει ήδη μία φορά πριν να ολοκληρωθεί ο κύκλος (full period generator). Αυτό δεν ισχύει πάντα, μάλιστα για να παράγονται και οι m διαφορετικοί αριθμοί σε έναν κύκλο πρέπει να ισχύουν εκείνες οι συνθήκες που αφορούν τη θεωρία αριθμών και οι οποίες δεν μας απασχολούν εδώ. Έτσι, ο διαιρέτης είναι συνήθως αρκετά μεγάλος (ώστε να έχουμε μεγάλο κύκλο, κάτι που αυξάνει την πιθανότητα να έχουμε και μεγάλη ακολουθία

τυχαίων αριθμών, πριν αρχίσουν να επαναλαμβάνονται οι ίδιοι αριθμοί) και συνήθως, όπως προαναφέρθηκε, είναι δύναμη του 2. Η δύναμη αυτή εξαρτάται με τη σειρά της από το μήκος λέξης του υπολογιστή (byte). Δηλαδή, αν ο υπολογιστής—μέσω του λειτουργικού του συστήματος—μπορεί να έχει μήκος λέξης μέχρι 32 bit ή 64 bit, τότε το m τίθεται ίσο με 232 ή 264 αντίστοιχα. Το πλεονέκτημα στην περίπτωση αυτή είναι φανερό: Η διαίρεση με το 2 ανάγεται σε απλή αφαίρεση δυνάμεων και εκτελείται γρήγορα στον υπολογιστή (που ως γνωστόν, χρησιμοποιεί το δυαδικό σύστημα αρίθμησης). Άλλες επιλογές είναι να θέσουμε $mm=2kk-1$ ή $mm=2kk+1$ όπου k το μήκος λέξης. Στην περίπτωση αυτή μάλιστα, υπάρχει το ενδεχόμενο το m να καταστεί πρώτος αριθμός (prime number, δηλαδή, να διαιρείται μόνο από τον εαυτό του και τη μονάδα). Το γεγονός αυτό είναι επιθυμητό διότι αυξάνει την πιθανότητα το μήκος του κύκλου να είναι το μέγιστο δυνατό χωρίς να χαθεί το πλεονέκτημα που αναφέρθηκε παραπάνω σε ό,τι αφορά τον υπολογισμό που συμβαίνει εσωτερικά στον υπολογιστή.

- Οι παρατηρήσεις θα πρέπει να είναι ομοιόμορφα κατανεμημένες και ανεξάρτητες. Πρακτικά, οι ψευδοτυχαίοι αριθμοί πρέπει να έχουν την ίδια πιθανότητα εμφάνισης σε οποιαδήποτε θέση της ακολουθίας.
- Η ακολουθία πρέπει να παράγεται γρήγορα (σε χρόνο CPU πάντα), διότι κατά την προσομοίωση υπάρχουν αρκετές διαδικασίες που χρειάζονται υπολογιστική ισχύ, και θα θέλαμε φυσικά να αποφύγουμε επιπλέον επιβάρυνση ισχύος κατά την εκτέλεση της προσομοίωσης λόγω του μηχανισμού παραγωγής των τυχαίων αριθμών.

Το πρόβλημα της επιλογής των παραμέτρων p , s και του διαιρέτη m δεν είναι τετριμμένο και οι κατάλληλες τιμές αυτών πρέπει να ικανοποιούν τις παραπάνω ιδιότητες. Όσον αφορά τον τρόπο επιλογής, υπάρχουν διάφοροι κανόνες, ώστε να ικανοποιούνται οι παραπάνω συνθήκες, αλλά δεν θα επεκταθούμε περισσότερο.

3. Discrete event simulation και η συμβολή του χρόνου στην προσομοίωση

Η προσομοίωση διακριτών γεγονότων (discrete-event simulation) είναι η μοντελοποίηση συστημάτων στα οποία οι μεταβλητές κατάστασης μεταβάλλονται στιγμιαία σε διακεκριμένες χρονικές στιγμές. Η έννοια του διακριτού γεγονότος ορίζεται ως μια στιγμιαία αλλαγή στην κατάσταση του συστήματος που συμβαίνει σε διακριτή χρονική στιγμή. Το σύστημα εξυπηρέτησης των ταμείων ενός τραπεζικού υποκαταστήματος θα μπορούσε πράγματι να θεωρηθεί διακριτό, αφού οι μεταβλητές κατάστασης που το χαρακτηρίζουν (μήκος ουράς, κατάσταση ταμιολογιστή) μεταβάλλονται, όταν ένας πελάτης καταφθάνει ή αποχωρεί από το σύστημα μετά την ολοκλήρωση της εξυπηρέτησής του. Οι μεταβολές αυτές στην κατάσταση του συστήματος **καταγράφονται σε διακριτές χρονικές στιγμές**. Επίσης, ένα σύστημα διαχείρισης αποθεμάτων μπορεί να θεωρηθεί διακριτό, εφόσον οι μεταβολές σε μια κλασική μεταβλητή κατάστασης του συστήματος – όπως το επίπεδο του αποθέματος– προκαλούνται από γεγονότα όπως η παραλαβή παραγγελίας (αύξηση του επιπέδου αποθέματος) ή η εκτέλεση μιας παραγγελίας (μείωση του επιπέδου αποθέματος), γεγονότα που συμβαίνουν σε διακριτές χρονικές στιγμές.

Τα μοντέλα προσομοίωσης αναλύονται περισσότερο με μαθηματικές παρά με αναλυτικές μεθόδους. Οι αναλυτικές μέθοδοι χρησιμοποιούν την αφαιρετική

αιτιολόγηση των μαθηματικών για να επιλύσουν μαθηματικά μοντέλα. Στην περίπτωση των μοντέλων προσομοίωσης που χρησιμοποιούν αριθμητικές μεθόδους, τα μοντέλα είναι προτιμότερο να εκτελεστούν παρά να επιλυθούν. Με τη διαδικασία αυτή δημιουργείται μια τεχνητή ιστορία του συστήματος βασισμένη στις αρχικές παραδοχές και στις παρατηρήσεις και η οποία αναλύεται για να καθοριστούν τα κατάλληλα μέτρα της απόδοσης του συστήματος. Από τη στιγμή που τα μοντέλα προσομοίωσης πραγματικού χρόνου είναι μάλλον μεγάλα και η ποσότητα των δεδομένων που αποθηκεύονται και επεξεργάζονται τεράστια, οι δοκιμές των μοντέλων γίνονται συνήθως με τη βοήθεια υπολογιστή. Παρόλα αυτά, είναι δυνατό να οδηγηθεί ο μελετητής σε χρήσιμα συμπεράσματα παρατηρώντας μικρά μοντέλα χωρίς να πρέπει να χρησιμοποιήσει υπολογιστή. Όλα τα μοντέλα προσομοίωσης διακριτών γεγονότων περιέχουν ορισμένους παράγοντες οι οποίοι είναι οργανωμένοι με κάποια λογική για την εύκολη κωδικοποίηση, για την εύρεση λαθών και για μελλοντικές μεταβολές του προγράμματος του μοντέλου προσομοίωσης. Πιο συγκεκριμένα, οι επόμενοι παράγοντες συναντώνται στα περισσότερα μοντέλα προσομοίωσης διακριτών γεγονότων:

- Σύστημα (system): Ένα σύνολο από οντότητες (π.χ. άνθρωποι, μηχανές κτλ) που αλληλεπιδρούν μεταξύ τους για την επίτευξη ενός ή περισσότερων στόχων.
- Μοντέλο (model): Μία αφηρημένη αναπαράσταση ενός συστήματος, που συνήθως περιλαμβάνει δομικές, λογικές ή μαθηματικές σχέσεις που περιγράφουν ένα σύστημα με όρους της κατάστασης του, των οντοτήτων του και των χαρακτηριστικών τους, των συνόλων, των διαδικασιών, των

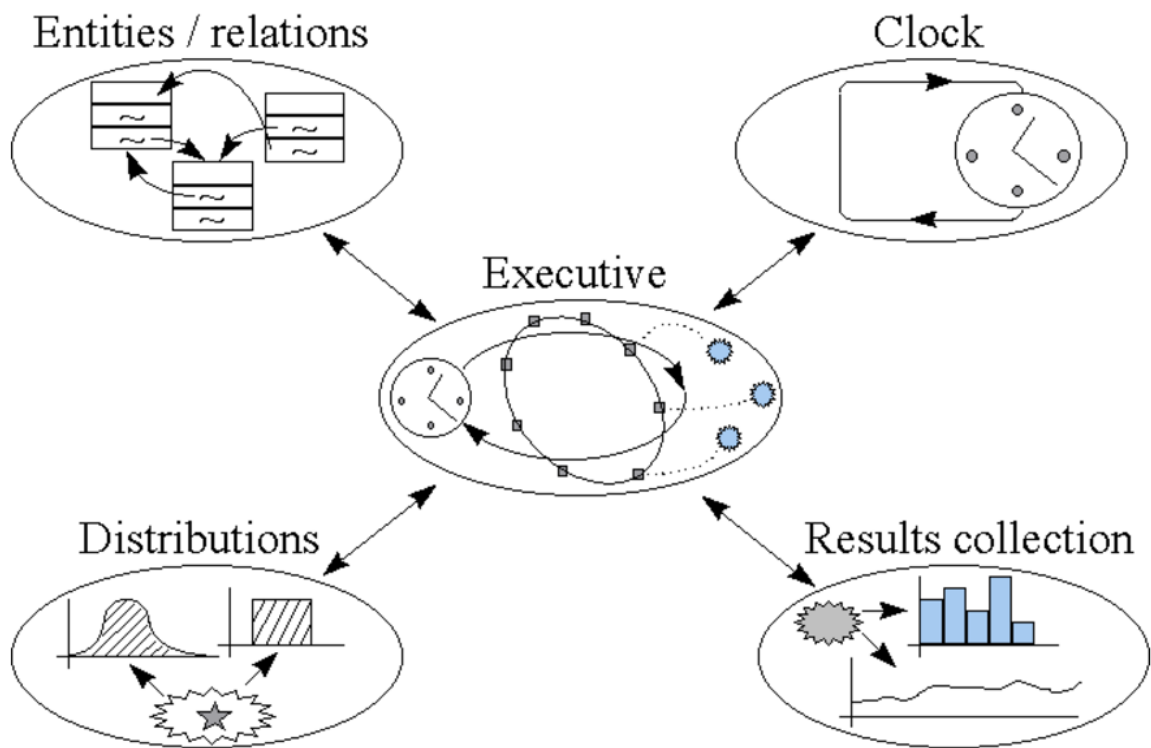
γεγονότων, των δραστηριοτήτων και των καθυστερήσεων.

- Κατάσταση συστήματος (system state): Μια συλλογή μεταβλητών που περιλαμβάνει όλη την πληροφορία που είναι απαραίτητη για την περιγραφή του συστήματος κάποια συγκεκριμένη χρονική στιγμή.
- Οντότητα (entity): Κάθε αντικείμενο ή συστατικό του συστήματος που απαιτεί ξεχωριστή αναπαράσταση σε μοντέλο.
- Χαρακτηριστικά (attributes): Οι ιδιότητες μιας δεδομένης οντότητας.
- Λίστα (list): Ένα σύνολο συσχετιζόμενων οντοτήτων, τοποθετημένων σε μια λογική σειρά.
- Γεγονός (event): Ένα στιγμιαίο συμβάν που μπορεί να αλλάξει την κατάσταση του συστήματος.
- Ειδοποίηση γεγονότος (event notice): Μία εγγραφή ενός γεγονότος που θα συμβεί σε κάποια χρονική στιγμή στο μέλλον. Περιλαμβάνει και τα δεδομένα που χρειάζονται για την εκτέλεση του γεγονότος αυτού.
- Λίστα γεγονότων (event list): Μία λίστα που περιλαμβάνει *ειδοποιήσεις γεγονότων*, διατεταγμένες ως προς τη χρονική στιγμή που θα συμβούν. Είναι επίσης γνωστή και ως FEL (Future Event List).
- Δραστηριότητα (activity): Ένα χρονικό διάστημα συγκεκριμένης διάρκειας

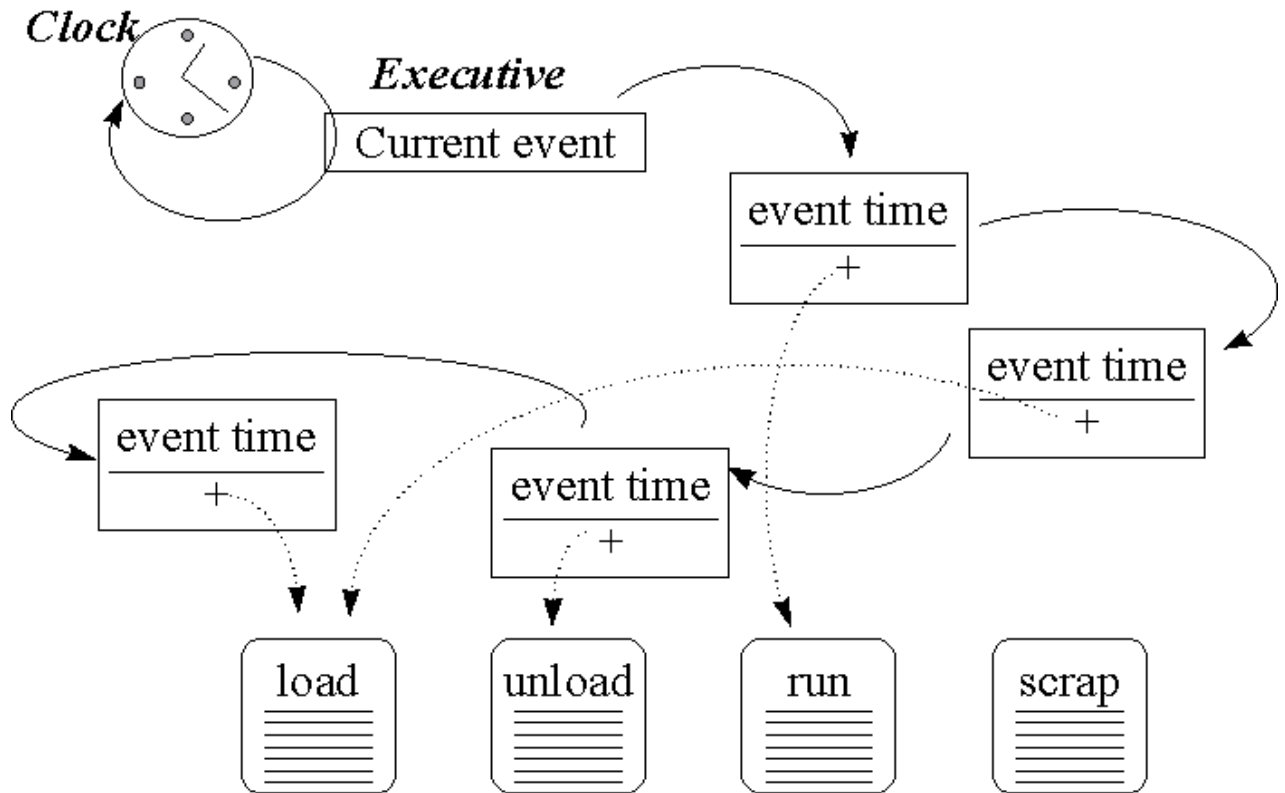
που γίνεται αντιληπτό τη στιγμή που ξεκινά.

- Καθυστέρηση (delay): Ένα χρονικό διάστημα ακαθόριστης διάρκειας, η οποία δεν είναι γνωστή μέχρι να τερματιστεί.
- Ρολόι (clock): Μία μεταβλητή που αναπαριστά το χρόνο προσομοίωσης.
- Στατιστικοί μετρητές (statistical counters): Μεταβλητές που χρησιμοποιούνται για την αποθήκευση στατιστικών μετρήσεων της απόδοσης του συστήματος.
- Ρουτίνα αρχικοποίησης (initialization routine): Ένα υποπρόγραμμα που αρχικοποιεί το μοντέλο προσομοίωσης τη χρονική στιγμή μηδέν.
- Ρουτίνα χρονισμού (timing routine): Είναι ένα υποπρόγραμμα το οποίο καθορίζει ποιο θα είναι το επόμενο γεγονός που θα συμβεί από τη λίστα γεγονότων και το οποίο στη συνέχεια προχωρά το ρολόι προσομοίωσης για να δείξει το χρόνο που θα συμβεί το γεγονός αυτό.
- Ρουτίνα γεγονότος (event routine): Ένα υποπρόγραμμα που ενημερώνει την κατάσταση συστήματος όταν συμβεί ένα γεγονός συγκεκριμένου τύπου (υπάρχει μία τέτοια ρουτίνα για κάθε είδος γεγονότος).
- Ρουτίνες βιβλιοθήκης (library routines): Σύνολο υποπρογραμμάτων τα οποία χρησιμοποιούνται για την παραγωγή τυχαίων παρατηρήσεων από κατανομές πιθανοτήτων που έχουν οριστεί σαν τμήμα του μοντέλου προσομοίωσης.

- Γεννήτρια αναφορών (report generator): Υποπρόγραμμα που υπολογίζει εκτιμήσεις των επιθυμητών μέτρων απόδοσης από τους στατιστικούς μετρητές και παράγει αναφορές όταν τελειώσει η προσομοίωση.
- Κυρίως πρόγραμμα (main program): Το πρόγραμμα που καλεί τη ρουτίνα χρονισμού για να καθοριστεί το επόμενο γεγονός και στη συνέχεια μεταφέρει τον έλεγχο στην αντίστοιχη ρουτίνα γεγονότος, για να ενημερωθεί κατάλληλα η κατάσταση του συστήματος. Το κυρίως πρόγραμμα επίσης ελέγχει τη συνθήκη τέλους και καλεί τη γεννήτρια αναφορών.



Εικόνα 4: Δομή ενός συστήματος προσομοίωσης.



Εικόνα 5: Παράγοντες που συναντάμε σε ένα μοντέλο προσομοίωσης διακριτών γεγονότων

Εκείνο το τμήμα της προσομοίωσης διακριτών γεγονότων που είναι υπεύθυνο για τον έλεγχο της ροής του χρόνου της προσομοίωσης είναι το ρολόι της

προσομοίωσης (simulation clock). Η χρονική στιγμή κατά την οποία λαμβάνουν χώρα τα διάφορα διακριτά γεγονότα καταχωρείται σε μια μεταβλητή, τη μεταβλητή του ωρολογίου (ή ρολογιού) (clock time). Υπάρχουν διάφορες μέθοδοι διαχείρισης του ρολογιού, αλλά εμείς θα επικεντρωθούμε σε δύο απ' αυτές [6] τη μέθοδο του επόμενου γεγονότος (next event method, event paced, critical event method) και τη μέθοδο του σταθερού χρονικού διαστήματος (fixed increment time method, time advance method).

3.1 Μέθοδος επόμενου γεγονότος

Σύμφωνα με την πρώτη μέθοδο, το σύστημα παρατηρείται (ελέγχεται και καταγράφεται το περιεχόμενο των μεταβλητών κατάστασής του) καθώς προχωράει από γεγονός σε γεγονός, όταν δηλαδή συμβαίνει κάποιο γεγονός. Η διαδικασία τερματίζεται όταν ικανοποιείται κάποιο κριτήριο τερματισμού.

Αυτό μπορεί να είναι η ολοκλήρωση μιας προκαθορισμένης ακολουθίας γεγονότων, η πραγματοποίηση κάποιου γεγονότος δείκτη ή η έλευση μιας καθορισμένης χρονικής στιγμής. Κάθε γεγονός που συμβαίνει –κατά κανόνα– σηματοδοτεί την έναρξη ή την ολοκλήρωση δραστηριοτήτων. Το ρολόι της προσομοίωσης ενημερώνεται (προχωρά) για τη χρονική στιγμή που συμβαίνει κάθε γεγονός και επειδή τα γεγονότα δεν συμβαίνουν σε ίσα χρονικά διαστήματα, το ρολόι μπορεί να προχωρά και αυτό κατά **άνισα χρονικά** διαστήματα. Με άλλα

λόγια, στη μέθοδο του επόμενου γεγονότος, το ρολόι της προσομοίωσης αυξάνει (προχωράει) σε μεταβλητά χρονικά διαστήματα και όχι με βάση ένα σταθερό χρονικό διάστημα, Δt , κάθε φορά. Αυτό, λοιπόν, το μεταβλητό διάστημα είναι ο χρόνος που μεσολαβεί ανάμεσα σε δύο διαδοχικά γεγονότα (π.χ. ανάμεσα σε δύο διαδοχικές αφίξεις, ανάμεσα στην ολοκλήρωση μιας εξυπηρέτησης και της εμφάνισης ενός πελάτη, από τη λήψη μιας παραγγελίας μέχρι τη λήψη της επόμενης, από την πώληση μιας ποσότητας προϊόντος μέχρι την επόμενη πώληση). Συνοψίζοντας τη μέθοδο έχουμε:

- Η μεταβλητή του ρολογιού καθορίζει τη χρονική στιγμή που θα συμβεί οποιοδήποτε επόμενο γεγονός και το ρολόι της προσομοίωσης προχωρά σ' αυτήν τη χρονική στιγμή, προσπερνώντας όλον τον ενδιάμεσο χρόνο κατά τον οποίο δεν συμβαίνει τίποτα.
- Το σύστημα ενημερώνεται και αλλάζει κατάσταση σε σχέση με τις αλλαγές που συνεπάγονται λόγω του γεγονότος αυτού (μεταβλητές κατάστασης). Επίσης, για λόγους συνέχισης της διαδικασίας, παράγεται και ο χρόνος που μεσολαβεί μέχρι το επόμενο γεγονός όπως π.χ. η επόμενη άφιξη σε μια ουρά αναμονής (πώς παράγεται;). Οι χρονικές στιγμές κατά τις οποίες θα συμβούν τα επόμενα γεγονότα καταγράφονται στο ημερολόγιο των γεγονότων (θα το σχολιάσουμε σε λίγο). Οι πληροφορίες (π.χ. το τρέχον μήκος της ουράς μετά το συμβάν, ο χρόνος αναμονής ενός πελάτη του οποίου έρχεται τώρα η σειρά μετά το γεγονός της αναχώρησης εκείνου που εξυπηρετούνταν κ.λπ.) καταγράφονται σε κατάλληλες βοηθητικές μεταβλητές, ώστε να πραγματοποιηθεί αργότερα η στατιστική επεξεργασία για την εξαγωγή συμπερασμάτων.

3.2 Μέθοδος σταθερού χρονικού διαστήματος

Σύμφωνα με τη δεύτερη μέθοδο, το σύστημα παρατηρείται και οι τιμές των μεταβλητών κατάστασης καταγράφονται σε συγκεκριμένες, ισαπέχουσες μεταξύ τους χρονικές στιγμές με βάση τη μεταβλητή του ρολογιού. Κάθε φορά, το ρολόι προχωράει σύμφωνα με ένα προκαθορισμένο διάστημα Δt (π.χ. μια στοιχειώδη μονάδα χρόνου). Μόλις το ρολόι προχωρήσει από την τρέχουσα χρονική στιγμή t στην επόμενη χρονική στιγμή ($t+\Delta t$), γίνεται έλεγχος του συστήματος και διεκπεραίωση όλων των δραστηριοτήτων που σηματοδοτούνται από εκείνα τα γεγονότα που συνέβησαν στο διάστημα $t+\Delta t$. Συνοψίζοντας τη μέθοδο αυτή, έχουμε:

- Η μεταβλητή του ρολογιού αυξάνει κάθε φορά σταθερά κατά ένα αρκούντως «μικρό» και σταθερό χρονικό διάστημα, Δt .
- Το σύστημα ενημερώνεται και αλλάζει κατάσταση σε σχέση με τη μετάβαση στη νέα χρονική στιγμή $t+\Delta t$, εντοπίζοντας ποια γεγονότα μεσολάβησαν κατά το χρονικό αυτό διάστημα, διεκπεραιώνοντας όλες τις σχετικές δραστηριότητες και ενημερώνοντας τις μεταβλητές κατάστασης. Και εδώ οι πληροφορίες καταγράφονται σε κατάλληλες βοηθητικές μεταβλητές για να γίνει αργότερα η στατιστική τους επεξεργασία.

Η μέθοδος του επόμενου γεγονότος χρησιμοποιείται κυρίως σε διακριτά συστήματα, ενώ η μέθοδος του σταθερού χρονικού διαστήματος χρησιμοποιείται για την προσομοίωση συνεχών συστημάτων. Ένα βασικό μειονέκτημα της μεθόδου του σταθερού χρονικού διαστήματος είναι ότι το ρολόι προχωρά – μερικές φορές άσκοπα – κατά σταθερό χρονικό διάστημα και κάθε φορά πρέπει να γίνεται έλεγχος για να διαπιστωθεί εάν έχει συμβεί κάποιο γεγονός. Η μέθοδος του επόμενου γεγονότος μειονεκτεί έναντι της μεθόδου του σταθερού χρονικού διαστήματος λόγω της αυξημένης πολυπλοκότητας που απαιτείται στη διαδικασία προγραμματισμού των επόμενων γεγονότων [7]. Όμως, αυτό δεν έχει επηρεάσει τη ευρεία χρήση της στην περίπτωση της προσομοίωσης διακριτών συστημάτων, εξαιτίας του γεγονότος ότι η διαθέσιμη υπολογιστική ισχύς είναι πλέον άφθονη, κάτι που καθιστά την πολυπλοκότητα αυτή μάλλον αμελητέα. Στο επόμενο σχήμα, συνοψίζεται η διαδικασία με τις δύο μεθόδους. Με διακεκομμένες γραμμές συμβολίζεται η μέθοδος του σταθερού χρονικού διαστήματος και με συνεχείς η μέθοδος του επόμενου γεγονότος.

Τα γεγονότα και οι προγραμματισμένοι χρόνοι εκτέλεσής τους φυλάσσονται στο **ημερολόγιο των γεγονότων** (κατάλογος γεγονότων, event list). Το ημερολόγιο ενδέχεται να είναι αρκετά μακροσκελές και να έχει προετοιμαστεί, εν μέρει, πριν ξεκινήσει η διαδικασία με την παραγωγή μιας σειράς γεγονότων. Μπορεί, εναλλακτικά, να περιέχει μόνο τα επόμενα γεγονότα που πρόκειται να συμβούν στο επόμενο άλμα του ρολογιού και να ενημερώνεται με τα καινούργια γεγονότα και τους αντίστοιχους χρόνους εκτέλεσής τους, κάθε φορά που κάποιο απ' αυτά προγραμματίζεται να συμβεί με τους τρόπους που περιγράψαμε παραπάνω.

Για παράδειγμα, σε ένα μοντέλο ενός συστήματος εξυπηρέτησης με μία ουρά αναμονής, γνωρίζουμε ότι τα δύο σημαντικά γεγονότα που καθορίζουν την

κατάσταση του συστήματος είναι η άφιξη και η αναχώρηση ενός πελάτη. Όταν συμβεί μια άφιξη, τότε σηματοδοτείται η έναρξη κάποιων δραστηριοτήτων τις οποίες θα δούμε στο αμέσως επόμενο παράδειγμα. Όταν ολοκληρωθούν όλες οι ενέργειες που ενεργοποιούνται από την άφιξη αυτή, τότε σύμφωνα με τη μέθοδο του επόμενου γεγονότος το πρόγραμμα ανατρέχει στο (χρονικά ταξινομημένο) ημερολόγιο των γεγονότων και εντοπίζει το επόμενο γεγονός που πρέπει να συμβεί. Το τι είδους γεγονός θα είναι (άφιξη ή αναχώρηση) εξαρτάται από τους τυχαίους χρόνους (χρονικές στιγμές για να είμαστε ακριβείς) της επόμενης άφιξης και της επόμενης αναχώρησης που παράχθηκαν νωρίτερα. Οι χρόνοι αυτοί μπορεί να έχουν καθοριστεί εκ των προτέρων και να τροφοδοτούνται στο πρόγραμμα όταν χρειάζεται ή να παράγονται όταν αυτό είναι απαραίτητο. Για παράδειγμα, η χρονική στιγμή αναχώρησης ενός πελάτη (δηλαδή, η ολοκλήρωση της εξυπηρέτησής του) μπορεί να παράγεται μόλις ξεκινήσει η εξυπηρέτησή του και δεν χρειάζεται να έχει καταγραφεί στο ημερολόγιο νωρίτερα. Το ρολόι της προσομοίωσης μετακινείται στη χρονική στιγμή που συμβαίνει το γεγονός αυτό και η διαδικασία συνεχίζεται. Εναλλακτικά, όλες οι χρονικές στιγμές αφίξεων και αναχωρήσεων παράγονται εκ των προτέρων, ταξινομούνται στο ημερολόγιο μέχρι τον χρόνο λήξης της προσομοίωσης και ξεκινά το τρέξιμο του προγράμματος. Σε κάθε περίπτωση, το τελικό αποτέλεσμα είναι το ίδιο. Δηλαδή, θα έχουμε αποκτήσει το ίδιο δείγμα από τις τιμές των μεταβλητών που θέλουμε να μετρήσουμε, ώστε να προχωρήσουμε στη στατιστική επεξεργασία.

4. Λογισμικά προσομοίωσης

Η διαδικασία της προσομοίωσης δικτύων επιτυγχάνεται με χρήση προσομοιωτών (simulators) τα οποία είναι λογισμικά προσομοίωσης.

Υπάρχουν πακέτα τα οποία έχουν φτιαχτεί για εκπαιδευτικούς σκοπούς και εμπορικά πακέτα προσομοίωσης. Κάποια από τα δημοφιλή εκπαιδευτικά πακέτα προσομοίωσης είναι το ns-2, το Network Workbench και το Netsim. Αντίστοιχα, κάποια από τα εμπορικά πακέτα προσομοίωσης είναι το OPNET και το COMNET III.

Παρακάτω θα παρουσιαστούν και θα σχολιαστούν, αναλυτικά, εκπαιδευτικά και εμπορικά πακέτα προσομοίωσης δικτύων.

4.1 Προσομοιωτής ns-2

Το **ns-2** (Network Simulator) [8-11] το οποίο είναι ένα VINT (Virtual Inter-Δίκτυο Testbed) project από το UC Berkeley / LBL / Xerox PARC, είναι ένας διακριτός

προσομοιωτής που απευθύνεται σε ερευνητικό δίκτυο, το οποίο παρέχει ουσιαστική στήριξη για την προσομοίωση του πρωτοκόλλου TCP, δρομολόγησης και πρωτοκόλλων «πολυεκπομπής» (multicasting protocols). Ο προσομοιωτής χρησιμοποιεί ένα Tcl / Otcl (Tool Command Language / Object oriented Tcl) ως διεπαφή εντολών και ρυθμίσεων.

Υπάρχουν τέσσερις τύποι αρχείων που σχετίζονται με τον προσομοιωτή ns-2:

- Η προέκταση .tcl
- Η προέκταση .ns
- Η προέκταση .tr
- Η προέκταση .nam

Μοντέλο περιγράφεται στα αρχεία .tcl ή .ns που έχουν κοινό υποσύνολο των εντολών, αλλά όχι ακριβώς συμβατές μεταξύ ο ένας του άλλου. Ενώ, ο προσομοιωτής τρέχει ένα μοντέλο που ορίζονται στο αρχείο .tcl / .ns, αρχείο ίχνος προσομοίωσης (.tr) και το αρχείο κινούμενα σχέδια (.nam) δημιουργούνται κατά τη διάρκεια της συνόδου. Αρχεία τύπου NetworkAnimator (.nam) είναι αρχεία που χρησιμοποιούνται για να απεικονίσουν τη συμπεριφορά των πρωτόκολλων δικτύου και την κυκλοφορία (traffic) του μοντέλου. Μόλις δημιουργηθούν, οι χρήστες μπορούν να παίξουν με το αρχείο .nam ακριβώς όπως ένα media player και να ελέγξουν τη συμπεριφορά επανειλημμένα.

Το ns-2 διευκολύνει τα τρία ευρεία θέματα των προσομοιώσεων έρευνας δικτύου:

- επιλογή ενός μηχανισμού από διάφορες επιλογές.

- με τη διερεύνηση σύνθετη συμπεριφορά
- να διερευνήσουμε το αντίθετο παρατηρείται αλληλεπίδραση πολλαπλών πρωτοκόλλων.

Το ns-2 υποστηρίζει την οικογένεια TCP, UDP, CBR, FTP, HTTP, Pareto, Εκθετικά πρωτόκολλα, καλώδια, ασύρματα, unicast, multicast. Οι γλώσσες προγραμματισμού που υποστηρίζει είναι οι C ++ και OTcl. Χαρακτηρίζεται από δυναμική εξόδου.

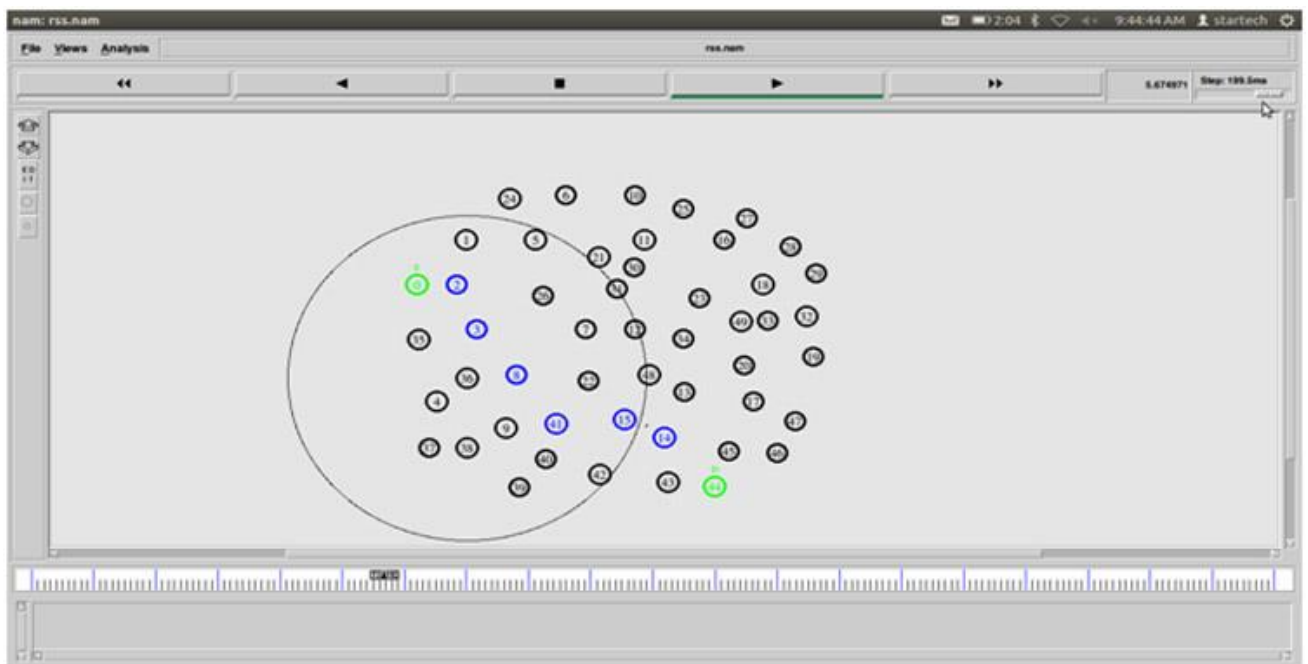
Τα πλεονεκτήματα του προσομοιωτή είναι:

- ο χρήστης μπορεί να σχεδιάσει το μοντέλο τόσο με το χέρι ή τη σύνταξη κώδικα
- εύκολη ρύθμιση και γρήγορη προσομοίωση χρησιμοποιώντας δύο διαφορετικές γλώσσες (OTcl και C ++)
- πολλά πρωτόκολλα που έχουν ήδη εφαρμοστεί
- δυναμική συμπεριφορά μπορεί να απεικονιστεί χρησιμοποιώντας τον επεξεργαστή *nam*
- Ανοιχτό τόσους κώδικα και μπορεί να παραταθεί για το μέλλον.

Οι περιορισμοί του προσομοιωτή είναι:

- Μεγάλος χρόνος για να συνηθίσει τη χρήση του
- δεν υπάρχει μεγάλης κλίμακας sim (π.χ. διαδίκτυο)
- ατελές API
- δεν προσομοίωση πραγματικού χρόνου
- δεν είναι δυνατή η ανάλυση απόδοσης

- άσχημα δοσ υμεντε κώδικας d πηγή
- καμία άμεση υποστήριξη της κινητικότητας και των κοινόχρηστων ασύρματων ραδιοφωνικών σταθμών
- μπορεί να κλιμακώνεται σε δίκτυα μεγέθους Διαδικτύου λόγω των υπολογιστικές απαιτήσεις του λεπτού κόκκου προσομοίωση επίπεδο πακέτου και της μνήμης απαιτείται για να διατηρηθεί ουρές σε κάθε σύνδεσμο ηΔικτύου.



Εικόνα 6: Περιβάλλον προσομοιωτή ns-2

4.2 Ο Προσομοιωτής ns-3

Ο ns-3 είναι ένας προσομοιωτής δικτύων υπολογιστών, όπου οι σχεδιαστές, για λόγους ταχύτητας, επιχείρησαν να συνδυάσουν την ταχύτητα της C++ με την ευκολία που προσφέρει η γλώσσα Python. Οι πόροι που χρειάζεται κανείς για να εγκαταστήσει τον ns-3 είναι:

- Το δωρεάν εργαλείο για διαχείριση μεγάλων πακέτων λογισμικού Mercurial
- Το δωρεάν εργαλείο Waf για όλη την διαδικασία από την μεταγλώττιση του πηγαίου κώδικα έως την δημιουργία εκτελέσιμης μορφής. Αυτό περιλαμβάνεται στον ns-3 και προτιμάται έναντι του make.

Το μεγαλύτερο μέρος του API (Application Program Interface) του ns-3 είναι διαθέσιμο σε Python, αλλά όλα τα μοντέλα έχουν γραφεί σε C++. Η δημιουργία των script για τον ns-3 μπορεί να γίνει σε οποιαδήποτε από τις δύο γλώσσες στις περισσότερες των περιπτώσεων, αλλά μία ικανοποιητική γνώση της C++ είναι

αναγκαία. Ως compiler χρησιμοποιείται ο γνωστός gcc της GNU μαζί με άλλα της εργαλεία. Αν και ο ns-3 διανέμεται σε πηγαία μορφή μαζί με τα περισσότερα αναγκαία εργαλεία, ο χρήστης πρέπει να δημιουργήσει με αυτά ένα σύστημα από βιβλιοθήκες λογισμικού. Ο πηγαίος κώδικας είναι επίσης αναγκαίος, επειδή ο χρήστης δημιουργεί τα προγράμματα που αντιστοιχούν σε προσομοιώσεις του ns-3 με το να ξαναφτιάχνει κάποιες από τις βιβλιοθήκες. Ως λειτουργικό σύστημα θεωρείται ότι χρησιμοποιούμε κάποια έκδοση του Unix.

Μια και χρησιμοποιείται η C++ και έχουμε να κάνουμε με δίκτυα υπολογιστών, η βασική κλάση είναι ο δικτυακός κόμβος που ονομάζεται Node. Αυτή περιέχει διάφορες κατάλληλες μεθόδους για την αναπαράσταση των διαφόρων τύπων κόμβων στις προσομοιώσεις. Σε αυτήν την βασική κλάση προσθέτουμε και άλλα στοιχεία κατά περίπτωση (επεκτείνοντάς την) για να πάρουμε τον τύπο δικτυακής συσκευής και υπηρεσιών που χρειαζόμαστε. Η λογική είναι ότι σε αυτούς τους κόμβους προσθέτουμε διάφορους τύπους εφαρμογών (γενική κλάση Application στον ns-3), οι οποίες γεννούν δικτυακή κίνηση, η οποία προωθείται σε γειτονικούς κόμβους, κ.ο.κ., έως τον τελικό προορισμό-κόμβο, όπου μία αντίστοιχη εφαρμογή φροντίζει να απορροφά την εισερχόμενη κίνηση.

Για να μπορούμε να διασυνδέσουμε τους κόμβους μεταξύ τους και να σχηματίσουμε μία δικτυακή τοπολογία, χρειάζονται κατάλληλες ζεύξεις. Εδώ, τον ρόλο των ζεύξεων τον παίζει η βασική κλάση Channel, από την οποίαν προέρχονται όλοι οι τύποι, από ένα απλό σύρμα, έως ένα πολύπλοκο ασύρματο κανάλι (ή ακόμα και μεταγωγέας).

Όπως στον πραγματικό κόσμο, έτσι και εδώ, κάθε αντικείμενο Node, μπορεί να περιέχει πολλές δικτυακές κάρτες (κλάσης NetDevice), μέσω των οποίων να

συνδέεται με πολλαπλά κανάλια (μία κάρτα ανά κανάλι). Επειδή οι τρεις αυτές κλάσεις και οι επεκτάσεις τους χρησιμοποιούνται πολύ συχνά στον ns-3, παρέχονται έτοιμα κάποια βοηθητικά εργαλεία που συλλογικά ονομάζονται *Topology Helpers*. Με αυτά καθίσταται ευκολότερο το να δημιουργήσουμε τις επιθυμητές συνδέσεις των παραπάνω τύπων αντικειμένων σε προσομοιώσεις μεγάλων μοντέλων, να αναθέσουμε διευθύνσεις IP, κλπ.

Κάθε script (πρόγραμμα) σε ns-3, όπως στο αρχείο 'first. cc' που χρησιμοποιούμε εδώ και βρίσκεται στον κατάλογο 'examples/tutorial', ξεκινά με μία γραμμή οδηγιών για τον κειμενογράφο (editor) emacs, με την παρακάτω τυπική μορφή:

```

|/* -
|*- Mode:C++; c-file-style:"gnu"; indent-tabs-mode:nil; - *
|*/

```

Στην συνέχεια ακολουθούν γραμμές σχολίων (όπως ορίζονται στην γλώσσα C), π.χ.:

```

|/*
 *   This program is free software; you can redistribute it and/or modify
 *   it under the terms of the GNU General Public License version 2 as
 *   published by the Free Software Foundation;
 *
 *   This program is distributed in the hope that it will be useful,
 *   but WITHOUT ANY WARRANTY; without even the implied warranty of
 *   MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 *   GNU General Public License for more details.
 *
 *   You should have received a copy of the GNU General Public License
 *   along with this program; if not, write to the Free Software
 *   Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA */

```

Αυτά ακολουθούνται από διάφορες οδηγίες `#include`, ώστε να συμπεριληφθούν έτοιμα `module` από τις βιβλιοθήκες του `ns-3`, π.χ.:

```

| #include "ns3/core-module.h"
| #include "ns3/network-module.h"
| #include "ns3/internet-module.h"
| #include "ns3/point-to-point-module.h"
| #include "ns3/applications-module.h"

```

Αυτές οι οδηγίες χρησιμοποιούνται από το εργαλείο `Waf` κατά την δημιουργία του εκτελέσιμου κώδικα, για την συμπερίληψη του κώδικα βιβλιοθήκης που αντιστοιχεί σε αυτά. Κάθε όνομα πριν το τμήμα `'module.h'`, αντιστοιχεί στο `module` του `ns-3` που βρίσκεται στον αντίστοιχο κατάλογο (π.χ., `"ns3/core"` στην διανομή λογισμικού του `ns-3`, που έχουμε εγκαταστήσει).

Κατόπιν ακολουθεί η παρακάτω δήλωση, ώστε όλες οι κλάσεις (και το συνολικό project του ns-3) να μη υπάρχει περίπτωση να συγκρουσθούν με άλλα

```
using namespace ns3;
```

Μετά έχουμε μία μακρο-εντολή για καταγραφή, ώστε να οδηγούμαστε στην αντίστοιχη ιστοσελίδα της τεκμηρίωσης, π.χ.:

```
NS_LOG_COMPONENT_DEFINE ("FirstScriptExample");
```

Ακολουθεί το κυρίως μέρος του script με κάποιους Topology Helpers.

- Οι πρώτος έχει να κάνει με τους δύο κόμβους που θέλουμε να δημιουργήσουμε

```

NodeContainer nodes; nodes.Create (2);

PointToPointHelper pointToPoint;
pointToPoint.SetDeviceAttribute ("DataRate", StringValue ("5Mbps"));
pointToPoint.SetChannelAttribute ("Delay", StringValue ("2ms"));
NetDeviceContainer devices;
devices = pointToPoint.Install (nodes);

Ipv4AddressHelper address;
address.SetBase ("10.1.1.0", "255.255.255.0");
Ipv4InterfaceContainer interfaces = address.Assign (devices);

UdpEchoServerHelper echoServer (9);
ApplicationContainer serverApps = echoServer.Install (nodes.Get (1));
serverApps.Start (Seconds (1.0)); serverApps.Stop (Seconds (10.0));

UdpEchoClientHelper echoClient (interfaces.GetAddress (1), 9);
echoClient.SetAttribute ("MaxPackets", UIntegerValue (1));
echoClient.SetAttribute ("Interval", TimeValue (Seconds (1.0)));
echoClient.SetAttribute ("PacketSize", UIntegerValue (1024));
ApplicationContainer clientApps = echoClient.Install (nodes.Get (0));
clientApps.Start (Seconds (2.0)); clientApps.Stop (Seconds (10.0));

```

- Ο δεύτερος έχει να κάνει με την δημιουργία κατάλληλης ζεύξεως μεταξύ τους (των 5 Mbps και καθυστέρηση διάδοσης 2 msec)
- Ο τρίτος εγκαθιστά την κατάλληλη στοίβα πρωτοκόλλων.
- Ο τέταρτος λέει ότι η αυτόματη ανάθεση διευθύνσεων IP θα γίνει από την περιοχή 10.1.1.0 με μάσκα 255.255.255.0, ενώ η (αυτόματη) ανάθεση γίνεται στην τρίτη γραμμή.

- Ο πέμπτος έχει να κάνει με τις κατάλληλες (έτοιμες) εφαρμογές που γενούν την αντίστοιχη κυκλοφορία. Αφού η εφαρμογή μας είναι μία απλή 'echo' σε UDP, θα έχουμε στον κόμβο '1', τον server UDP. Αυτός ξεκινά την χρονική στιγμή 1'' και τερματίζεται την χρονική στιγμή 10'' ('9' είναι ο τυπικός αριθμός θύρας UDP για την υπηρεσία echo)
- Ο έκτος έχει να κάνει με τον πελάτη της υπηρεσίας echo, που τοποθετείται στον κόμβο '0'. Ξεκινά την χρονική στιγμή 2'' και τερματίζεται την χρονική στιγμή 10''. Αποστέλλει το πολύ 1 πακέτο (άρα εδώ ακριβώς ένα πακέτο), με διάστημα τουλάχιστον 1'' από το επόμενο (εδώ δεν υπάρχει), μεγέθους 1.024 byte στο payload του.

Τέλος, χρειάζεται να τρέξουμε την προσομοίωση και να καθαρίσουμε ό,τι έχει απομείνει στο τέλος στην μνήμη:

```

Simulator::Run ();
Simulator::Destroy (); return 0;
}

```

Έχοντας ολοκληρώσει το script στο παραπάνω αρχείο, πρέπει να δημιουργήσουμε το εκτελέσιμο. Ο χώρος εργασίας (workspace) είναι ο κατάλογος 'scratch'. Αφού τοποθετήσουμε το αρχείο script (έστω με όνομα 'myfirst.cc'), προχωράμε στις διαδικασίες δημιουργίας του εκτελέσιμου και λοιπών αρχείων, πληκτρολογώντας:

```
§ ./waf
```

Στην οθόνη θα εμφανισθούν τα σχετικά μηνύματα, τα οποία μας ενημερώνουν για την επιτυχία (ή μη) της όλης διαδικασίας, όπως τα παρακάτω:

```
Waf: Entering directory '/home/craigdo/repos/ns-3-allinone/ns-3-dev/build'  
[614/708]cxx: scratch/myfirst.cc -> build/debug/scratch/myfirst 3.o  
[706/708]cxx link: build/debug/scratch/myfirst 3.o -> build/debug/scratch/myfirst  
Waf: Leaving directory '/home/craigdo/repos/ns-3-allinone/ns-3-dev/build'  
'build' finished successfully (2.357s).
```

Τώρα πλέον μπορούμε να τρέξουμε το πρόγραμμα (πηγαίνοντας όμως προηγουμένως έξω από τον κατάλογο 'scratch'):

```
§ ./waf --run scratch/myfirst
```

Οπότε και θα εμφανισθεί στην οθόνη κάτι σαν τα παρακάτω:

```
Waf: Entering directory '/home/craigdo/repos/ns-3-allinone/ns-3-dev/build'  
Waf: Leaving directory '/home/craigdo/repos/ns-3-allinone/ns-3-dev/build'  
'build' finished successfully (0.418s)  
Sent 1024 bytes to 10.1.1.2 Received 1024 bytes from 10.1.1.1 Received 1024 bytes  
from 10.1.1.2
```

Η καταγραφή των αποτελεσμάτων μπορεί να γίνει και σε κατάλληλα αρχεία εξόδου, τα οποία μπορεί να έχουν την μορφή απλών εκτυπώσεων που ορίζουμε από το πρόγραμμά μας ή και τυποποιημένες καταγραφές διαφορετικού βάθους και λεπτομερειών. Καλύτερο όμως είναι να χρησιμοποιήσουμε το ενσωματωμένο

σύστημα καταγραφής αποτελεσμάτων (traces), σε μορφή ASCII, με την βοήθεια του ASCII Trace Helper. Πιο συγκεκριμένα, εισάγουμε στο τέλος του πηγαίου μας κώδικα, ακριβώς πριν την γραμμή 'Simulator::Run ()', τις παρακάτω γραμμές:

```
AsciiTraceHelper ascii;
pointToPoint.EnableAsciiAll (ascii.CreateFileStream ("myfirst.tr"));
```

Τα αποτελέσματα (traces) τοποθετούνται όπως βλέπουμε στο αρχείο 'myfirst.tr', για κάθε συσκευή με σύνδεση σημείου-προς-σημείο, ενώ έχουν την ίδια κωδικοποίηση με αυτήν στον παλαιότερο ns-2 για κάθε καταγεγραμμένο συμβάν:

- +: Έλαβε χώρα μία λειτουργία τοποθέτησης στην ουρά της συσκευής.
- -: Έλαβε χώρα μία λειτουργία αφαίρεσης από την ουρά της συσκευής.
- d: Απορρίφθηκε ένα πακέτο, τυπικά επειδή η ουρά ήταν πλήρης.
- r: Παρελήφθη ένα πακέτο από την δικτυακή συσκευή.

Υπάρχει η δυνατότητα χρήσης ενός άλλου helper προκειμένου να αποθηκεύονται τα αποτελέσματα σε μορφή '.pcap'. Η σχετική γραμμή κώδικα είναι μόλις μία γραμμή:

```
pointToPoint.EnablePcapAll ("myfirst");
```

Εδώ μπαίνει μόνον το πρόθεμα του ονόματος αρχείου. Ο αντίστοιχος helper θα τα αποθηκεύσει σε ξεχωριστά αρχεία για κάθε συσκευή σημείου-προς-σημείο, δηλαδή, 'myfirst-0-0 .pcap' και 'myfirst- 1-0 .pcap'. Αυτά αντιστοιχούν στον κόμβο 0-συσκευή 0, και στον κόμβο 1-συσκευή 0.

Τέτοια αρχεία μπορούν να διαβασθούν από την εφαρμογή 'tcpdump', αλλά και από τον αναλυτή πακέτων δικτύου 'Wireshark'.

Επίσης, υπάρχει η δυνατότητα να τροφοδοτήσουμε με όλα τα δεδομένα στο αρχείο (αφαιρώντας τα μηνύματα του Waf), ένα πρόγραμμα απεικόνισης όπως το gnuplot και να πάρουμε ένα διάγραμμα, αντί για ίχνη σε μορφή ASCII.

4.3 Ο προσομοιωτής OMNET++

Το OMNeT++ είναι ένα πρόγραμμα προσομοίωσης το οποίο χρησιμοποιεί Discrete Event Simulator. Με βασικό σκοπό την προσομοίωση δικτύων υπολογιστών, το OMNeT++ είναι επεκτάσιμο όπως και το ns-3. Επιπλέον, μπορεί να κάνει προσομοίωση σε δίκτυα ολοκληρωμένων κυκλωμάτων, δίκτυα ουρών, δίκτυα αισθητήρων κλπ. Για την επίτευξη αυτών, γίνεται ανάπτυξη διαφόρων πλαισίων (frameworks), ως ξεχωριστά έργα τα οποία αποτελούν βιβλιοθήκες με όλα τα απαραίτητα δικτυακά στοιχεία, μηνύματα, διαδικτυακά και άλλα πρωτόκολλα.

Επίσης, γίνεται διάθεση και κάποιων εργαλείων. Το κυριότερο αυτών είναι το IDE που βασίζεται στο Eclipse. Το Eclipse επιτρέπει προσπομοιώσεις σε γραφικό περιβάλλον.

Η αρχιτεκτονική στοιχείων-συνιστωσών για δημιουργία μοντέλων αποτελεί τη φιλοσοφία σχεδιασμού του.

Τα στοιχεία, ή αλλιώς *modules*, αυτά προγραμματίζονται σε C++. Κατόπιν συναρμολογούνται σε μεγαλύτερα συστατικά στοιχεία και μοντέλα χρησιμοποιώντας μία σχετικά απλή γλώσσα υψηλού επιπέδου, η οποία ονομάζεται NED. Έτσι, γίνεται εφικτή η επαναχρησιμοποίηση των μοντέλων. Τα βασικά στοιχεία του είναι:

- Η βιβλιοθήκη μαζί με τον πυρήνα για την προσομοίωση.
- Η γλώσσα περιγραφής τοπολογίας NED.
- Το IDE που βασίζεται στην πλατφόρμα Eclipse.
- Μία διεπαφή με τον χρήστη γραμμών εντολών για εκτέλεση προσομοιώσεων ('Cmdenv').
- Βοηθητικά εργαλεία.
- Τεκμηρίωση, ενδεικτικά παραδείγματα προσομοιώσεων, κλπ.

Μπορεί να τρέξει σε Linux, αλλά και σε Windows και Mac OS X. Τα μοντέλα προσομοίωσης που υποστηρίζει προέρχονται από διάφορα πλαίσια, τα οποία αναπτύσσονται ως διαφορετικά έργα. Τα κυριότερα είναι:

- INET. Η τυπική βιβλιοθήκη μοντέλων πρωτοκόλλων για το OMNeT++. Περιέχει πρωτόκολλα και για ενσύρματα και για ασύρματα δίκτυα, ξεκινώντας από το επίπεδο ζεύξης δεδομένων. Στο INET βασίζονται τα δύο επόμενα.
 - OverSim. Παρέχει την δυνατότητα προσομοιώσεων για δίκτυα P2P, καθώς και δίκτυα επικάλυψης, είτε δομημένα (π.χ., Chord, Pastry), είτε αδόμητα (π.χ., GIA).
 - ο Veins. Παρέχει την δυνατότητα προσομοιώσεων για επικοινωνία μεταξύ οχημάτων (IVC - Inter Vehicular Communication), μαζί με ένα μοντέλο για κυκλοφορία αυτοκινήτων σε δρόμο.
- INETMANET. Προέρχεται από το INET, αλλά προσθέτει περισσότερα χαρακτηριστικά και πρωτόκολλα, με έμφαση σε κινητά δίκτυα ad-hoc.
- MiXiM. Προορίζεται για προσομοίωση σταθερών και κινητών δικτύων, με έμφαση στα χαμηλότερα επίπεδα, περιλαμβανομένου και του φυσικού. Κυρίως έχει να κάνει με ασύρματα δίκτυα αισθητήρων, σώματος, κλπ.
- Castalia. Και αυτό προορίζεται για προσομοίωση ασυρμάτων δικτύων αισθητήρων, σώματος, και γενικά δικτύων ενσωματωμένων συσκευών, μικρής ισχύος.

Η προσομοίωση με το OMNeT++ περιλαμβάνει τα παρακάτω στάδια:

- Αρχικά κατασκευάζουμε ένα μοντέλο του δικτύου που θέλουμε να προσομοιώσουμε από modules
- Τα στοιχειώδη (τα πιο απλά) στοιχεία λέγονται *simple modules*. Αυτά

πρέπει να προγραμματισθούν σε C++, χρησιμοποιώντας τον πυρήνα του προσομοιωτή και την βιβλιοθήκη με κλάσεις που διατίθεται (εκτός εάν είναι ήδη διαθέσιμα).

- Δημιουργία ενός κατάλληλου αρχείου ρυθμίσεων και παραμέτρων του μοντέλου για μία ή περισσότερες προσομοιώσεις (π.χ., με διαφορετικές παραμέτρους). Αυτό το αρχείο ονομάζεται τυπικά `omnetpp.ini`.
- Δημιουργία (build) του προγράμματος προσομοίωσης και εκτέλεσή του, μετά από σύνδεσή του με τον πυρήνα του προσομοιωτή και ορισμό της διεπαφής με τον χρήστη (γραφική, διαδραστική ή γραμμής εντολών, μη διαδραστική).
- Αφού τρέξουμε την προσομοίωση, τα αποτελέσματα γράφονται, είτε σε μορφή απλών αριθμών, είτε διανυσμάτων, σε αντίστοιχα αρχεία. Μπορούμε να χρησιμοποιήσουμε το εργαλείο ανάλυσης που παρέχεται ή και άλλα δημοφιλή, όπως Matlab, R, κλπ., επειδή τα αποτελέσματα αυτά είναι γραμμένα σε μορφή ASCII.

Όπως είπαμε παραπάνω, τα διάφορα βασικά στοιχεία είναι τα λεγόμενα *modules*, τα οποία μπορούν να συνδυάζονται σε *σύνθετα (compound) modules*. Μεταξύ τους όμως πρέπει τα επί μέρους στοιχεία του ίδιου επιπέδου από *modules* να συνδέονται για να μπορούν να ανταλλάσσουν μηνύματα. Αυτό γίνεται μέσω συνδέσεων, που ονομάζονται *gates*. Για παράδειγμα, ένα απλό *module* είναι το παρακάτω (EtherMAC):

```
// Ethernet CSMA/CD MAC //
simple EtherMAC {
parameters:
string address;      // gates:
input phyIn;        //
output phyOut;      // input llcIn; //
output llcOut;      //
}
```

Αυτό μπορεί χρησιμοποιηθεί ως θεμέλιος λίθος για κατασκευή μοντέλου υπολογιστή με κάρτα Ethernet (compound module), όπως στο παρακάτω παράδειγμα:

```
//
// Host with an Ethernet interface
//
module EtherStation {
parameters: ...
gates: ...
input in;
app: EtherTrafficGen;
llc: EtherLLC; m
ac: EtherMAC;
connections:
app.out --> llc.hlIn;
app.in <-- llc.hlOut;
llc.macIn <-- mac.llcOut;
llc.macOut --> mac.llcIn;
mac.phyIn <-- in;
mac.phyOut --> out;
```

Κάθε compound module μπορεί να έχει κάποια λογικά μέρη, τα οποία μπορεί να είναι *parameters*, *gates*, *submodules* και *connections*. Στο παραπάνω παράδειγμα, βλέπουμε ότι το compound module EtherStation, περιλαμβάνει: *parameters*, *gates*, *submodules* και *connections*.

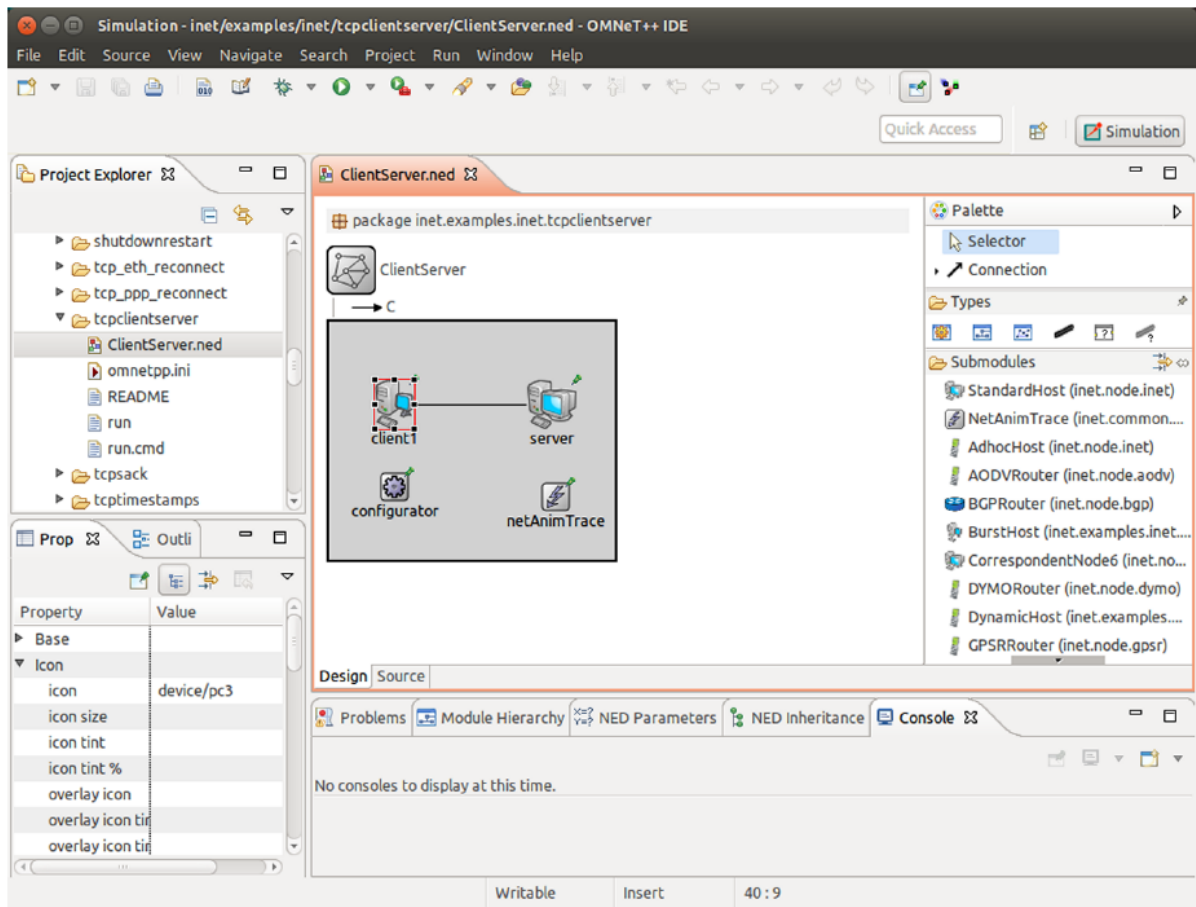
Γενικά, μπορεί να υπάρχει και το μέρος ονόματι *types*. Αυτό έχει να κάνει με τοπικά ονομαζόμενους τύπους από *submodules* και *connections*, με την αντίστοιχη λογική των τοπικών μεταβλητών σε μία συνάρτηση.

Τα *parameters*, ουσιαστικά είναι μεταβλητές που ανήκουν στο συγκεκριμένο module, όπως μία για το χρησιμοποιούμενο πρωτόκολλο, μέγεθος πακέτου που

αποστέλλει, κλπ.

Τα *gates* πάλι είναι τα σημεία μέσω των οποίων συνδέεται ένα module με τον 'έξω κόσμο'.

Τα *connections* χρησιμοποιούνται στο εσωτερικό από compound modules, ώστε να διασυνδέουν τα submodules από τα οποία αποτελείται αυτό το compound module. Δεν μπορούν να χρησιμοποιηθούν για συνδέσεις σε περισσότερα επίπεδα (για το ίδιο module) ή για συνδέσεις με τον 'έξω κόσμο'.



Εικόνα 7: Παράδειγμα TCPClientServer με γραφικό περιβάλλον.

Στην Εικόνα 4.1 φαίνεται ένα παράδειγμα του γραφικού περιβάλλοντος για την δημιουργία και διασύνδεση από modules για ένα απλό δίκτυο, αποτελούμενο από έναν TCP server και έναν TCP client, με το όνομα project “tcpclientserver”.

Όπως βλέπουμε στο παραπάνω στιγμιότυπο, στα αριστερά βρίσκεται ένα μικρό υπο-παράθυρο με τον τίτλο ‘Project Explorer’. Εκεί βλέπουμε ότι έχει ανοιχθεί το

project με τίτλο 'tcpclientserver', το οποίο περιλαμβάνει ως βασικά αρχεία το 'ClientServer.ned' που περιέχει την τοπολογία και άλλα στοιχεία του δικτύου, τα αρχεία 'run' και 'run.cmd' τα οποία είναι απλά script για να τρέξει το μοντέλο ο προσομοιωτής, ένα απλό 'README', και το αρχείο 'omnetpp.ini' στο οποίο και θα αναφερθούμε στην συνέχεια.

Επίσης, βλέπουμε στα δεξιά να υπάρχει μία παλέτα, τύποι και τα υπάρχοντα submodules, τα οποία μπορούν να χρησιμοποιηθούν με λειτουργία 'Drag & Drop' στο μοντέλο προσομοίωσης.

Στο κάτω μέρος του υπο-παραθύρου με την γραφική αναπαράσταση του μοντέλου βλέπουμε ότι βρισκόμαστε στην όψη 'Design', που προφανώς έχει γραφική μορφή. Εάν το επιθυμούμε, μπορούμε να επιλέξουμε την όψη 'Source', η οποία απεικονίζει τον πηγαίο κώδικα για το μοντέλο του δικτύου μας σε γλώσσα NED (τα σχόλια στην κορυφή έχουν απαλειφθεί για εξοικονόμηση χώρου):


```

package inet.examples.inet.tcpclientserver;
import inet.common.misc.NetAnimTrace;
import inet.networklayer.configurator.ipv4.IPv4NetworkConfigurator;
import inet.node.inet.StandardHost;
import ned.DatarateChannel;

network ClientServer {
parameters:
double per = default(0);
@display("bgb=232,193");
types:
channel C extends DatarateChannel {
datarate = 10Mbps; delay = 0.1us; per = per;
}

submodules:
client1: StandardHost { parameters:
@display("p=53,67;i=device/pc3");
}

server: StandardHost {
parameters:
@display("p=181,67;i=device/pc2");
}

configurator: IPv4NetworkConfigurator {
parameters:
@display("p=53,134");
}

netAnimTrace: NetAnimTrace {
@display("p=184,142");
}

connections:
client1.pppg++ <--> C <--> server.pppg++;
}

```

Όπως βλέπουμε, υπάρχει ένα αντικείμενο τύπου 'network', το οποίο έχει έναν νέο τύπο καναλιού, ο οποίος επεκτείνει τον υπάρχοντα βασικό στις βιβλιοθήκες, με χαρακτηριστικά: ρυθμός μετάδοσης 10 Mbps, καθυστέρηση διάδοσης 0,1 msec, και με περιόδους αποστολής 'per' που προκύπτουν από τις ρυθμίσεις της προσομοίωσης (βλ. παρακάτω για το αρχείο 'omnetpp.ini'). Βλέπουμε δε ότι υπάρχουν τέσσερα submodules:

- Το client1, το οποίο είναι τύπου "StandartHost"
- Το server, το οποίο είναι εξίσου "StandartHost"
- Το configurator, το οποίο έχει να κάνει με την αυτόματη αρχικοποίηση διευθύνσεων δικτύου
- Το netAnimTrace, το οποίο έχει να κάνει με την καταγραφή αποτελεσμάτων.

Τέλος, υπάρχει και το 'connections', το οποίο περιλαμβάνει το πώς συνδέονται μεταξύ τους ο client με τον server.

Το μόνο που απομένει είναι ένα αρχείο επιθυμητών ρυθμίσεων για το πώς θέλουμε να τρέξει η προσομοίωση. Αυτά περιέχονται τυπικά στο αρχείο 'omnetpp.ini'. Για την περίπτωση μας το περιεχόμενό του, παρατίθεται παρακάτω. Όποια γραμμή ξεκινά με τον χαρακτήρα '#' θεωρείται σχόλιο και αγνοείται. Επίσης, λόγω του μεγέθους του συγκεκριμένου αρχείου, ένα μέρος του έχει παραληφθεί (όπου υπάρχουν οι συνεχείς παύλες).

Παρατηρούμε ότι το αρχείο αυτό χωρίζεται σε επί μέρους τμήματα τα οποία ξεκινούν με έναν τίτλο μέσα σε αγκύλες που έχει πάντα πρώτη την λέξη-κλειδί 'Config'. Σε ένα τέτοιο αρχείο μπορούμε να έχουμε πολλαπλές παραλλαγές

(σενάρια) προσομοιώσεων για το ίδιο μοντέλο. Ένα είναι το βασικό, ονόματι 'General' και βρίσκεται στο τέλος του παρόντος αρχείου. Σε αντίθεση με τον ns-3, εδώ είναι που τοποθετούνται οι λεπτομέρειες των εφαρμογών σε κάθε κόμβο, όπως και το με ποια άλλη 'συνδέεται' ως προορισμό, αποστέλλοντάς της τα μηνύματα που δημιουργεί.

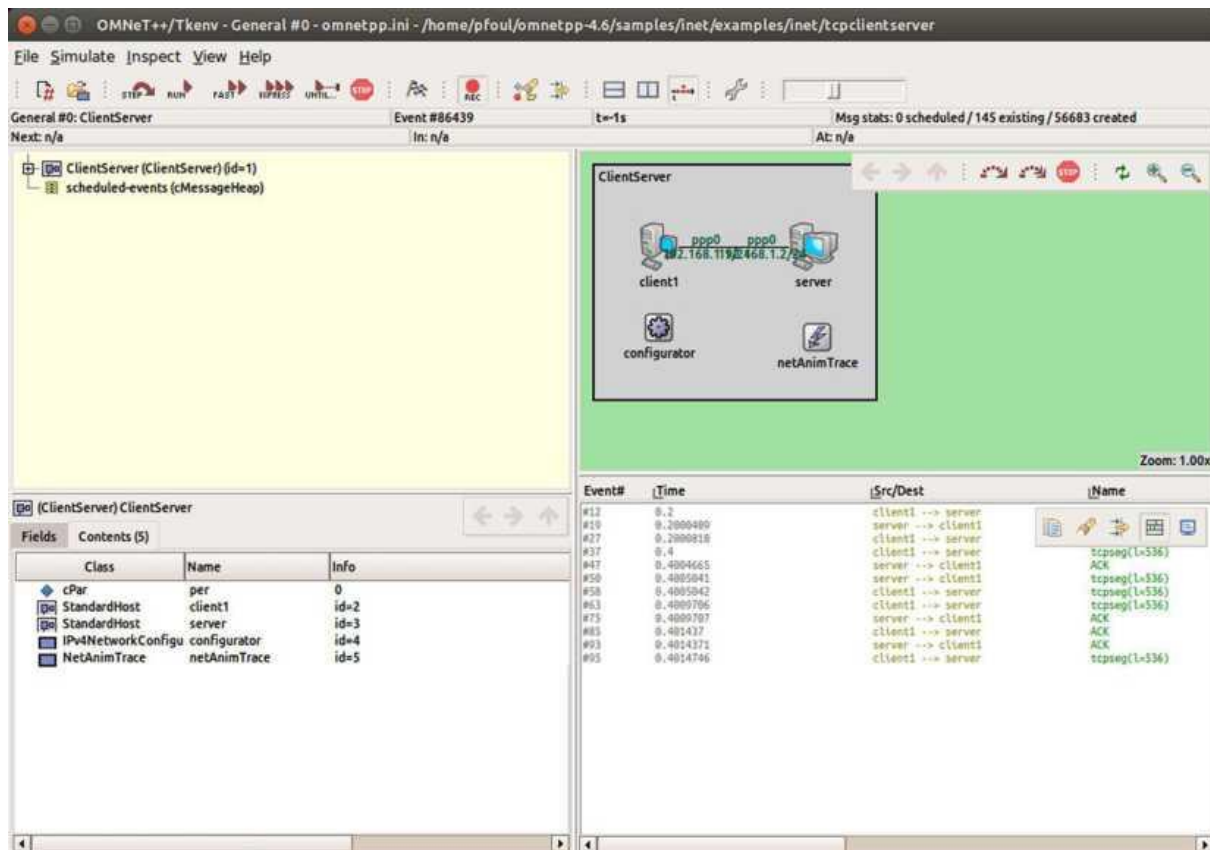
```
[Config default tcp]
description = "default TCP < > default TCP"
[Config inet inet]
description = "inet_TCP < > inet_TCP"
# default TCP implementation
**.tcpType = "TCP"
[Config inet lwip]
description = "inet_TCP < > TCP_lwIP"
# setting TCP stack implementation
**.server*.tcpType = "TCP"
**.client*.tcpType = "TCP_lwIP"
[Config inet-reno]
description = "TCP < > TCP with reno algorithm"
*.server*.tcpType = "TCP"
*.client*.tcpType = "TCP"
**.tcp.tcpAlgorithmClass = "TCPReno"
*.per = 0.01 * ${0, 0.1, 0.2, 0.5, 1, 2, 5}
*.server*.tcpApp[0].echoFactor = 0
-----
```

```

[Config inet-westwood]
description = "TCP < > TCP with Westwood algorithm"
*.server*.tcpType = "TCP"
*.client*.tcpType = "TCP"
**.tcp.tcpAlgorithmClass = "TCPWestwood"
*.per = 0.01 * ${0, 0.1, 0.2, 0.5, 1, 2, 5}
*.server*.tcpApp[0].echoFactor = 0
[Config inet-dumb]
description = "inet TCP < > inet TCP with DumbTCP algorithm"
*.server*.tcpType = "TCP"
*.client*.tcpType = "TCP"
**.tcp.tcpAlgorithmClass = "DumbTCP"
*.per = 0.01 * ${0, 0.1, 0.2, 0.5, 1, 2, 5}
*.server*.tcpApp[0].echoFactor = 0

```

```
#####
[General]
network = ClientServer total-stack = 7MiB
tkenv-plugin-path = ../../../../etc/plugins
#debug-on-errors = true
record-eventlog = true
**.server.numPcapRecorders = 1
**.server.pcapRecorder[0].pcapFile = "
**.client1.numPcapRecorders = 1
**.client1.pcapRecorder[0].pcapFile = "results/client1.pcap"
## tcp apps **.numTcpApps = 1
**.client*.tcpApp[*].typename = "TCPSessionApp"
**.client*.tcpApp[0].active = true
**.client*.tcpApp[0].localPort = -1
**.client*.tcpApp[0].connectAddress = "server"
**.client*.tcpApp[0].connectPort = 1000
**.client*.tcpApp[0].tOpen = 0.2s
**.client*.tcpApp[0].tSend = 0.4s
**.client*.tcpApp[0].sendBytes = 1000000B
**.client*.tcpApp[0].sendScript = ""
**.client*.tcpApp[0].tClose = 25s
**.server*.tcpApp[*].typename = "TCPEchoApp"
**.server*.tcpApp[0].localPort = 1000
**.server*.tcpApp[0].echoFactor = 2.0
**.server*.tcpApp[0].echoDelay = 0
# NIC configuration
**.ppp[*].queueType = "DropTailQueue"
# in routers
**.ppp[*].queue.frameCapacity = 10
# in routers
*.configurator.config=xml("<config><interface hosts='*' "...
"address='192.168.1.x' netmask='255.255.255.0'/></config>")
```



Εικόνα 8: Παράδειγμα TCPClientServer κατά την προσομοίωσή του.

Επανερχόμενοι στην αρχική μας εικόνα, εάν κάνουμε κλικ στο κουμπί 'Run', τότε επιχειρεί να ολοκληρώσει το 'build' και να φτιάξει ένα κατάλληλο run script, συσχετίζοντας τα κατάλληλα από τα περιεχόμενα του 'omnetpp.ini'. Εάν το τελευταίο είναι απλό, τότε προχωρά στην εκτέλεση. Διαφορετικά, μας ρωτά (όπως στο παράδειγμά μας) ποια από όλες τις ρυθμίσεις σεναρίων επιθυμούμε και μετά κάνει τα ίδια. Εάν όλα είναι επιτυχή και δεν έχουμε βάλει κάποια συγκεκριμένη

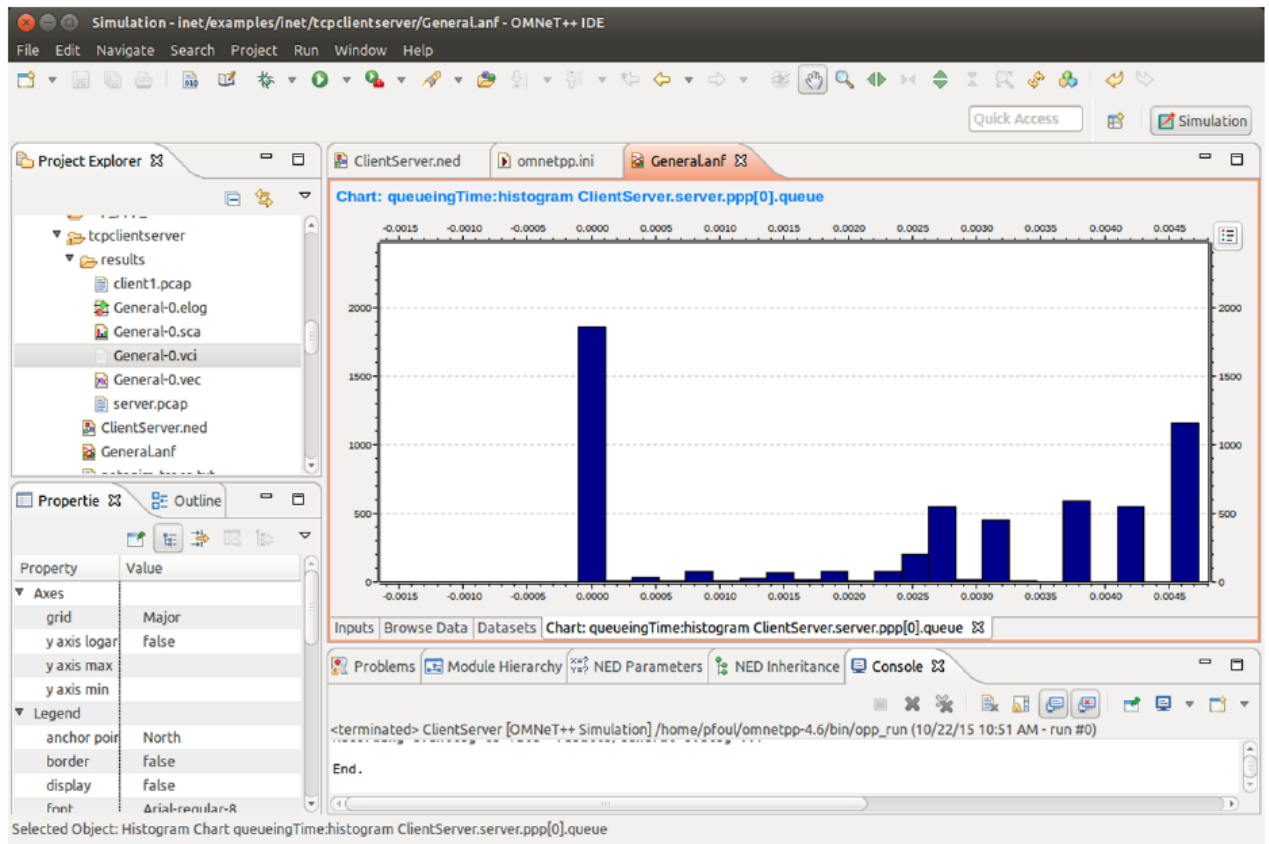
ρύθμιση, εκτελεί την προσομοίωση στο γραφικό και διαδραστικό περιβάλλον, όπως αυτό που βλέπουμε στην Εικόνα 4.2 για το παράδειγμά μας.

Εάν πάλι το επιθυμούμε, μπορούμε να επιλέξουμε μη διαδραστική προσομοίωση, οπότε δεν εμφανίζεται εικόνα ανάλογη της 4.2, αλλά η προσομοίωση τρέχει στην μέγιστη δυνατή ταχύτητα.

Τέλος, υπάρχουν δυνατότητες αποθήκευσης και οπτικοποίησης των αποτελεσμάτων (το τελευταίο με αρκετούς περιορισμούς). Για παράδειγμα, βλέπουμε στο μέρος 'Config General' του παραπάνω αρχείου 'omnetpp.ini' ότι τα αποτελέσματα έχουν ζητηθεί να αποθηκεύονται σε μορφή 'pcap', κατάλληλη για ανάγνωση και ανάλυση από το Wireshark. Εξ ορισμού, όλα τα αποτελέσματα τοποθετούνται σε αρχεία κάτω από τον κατάλογο 'results', ο οποίος δημιουργείται σε κάθε κατάλογο κάποιου project. Ανάλογα με το όνομα που έχει δοθεί σε κάθε μέρος του αρχείου ρυθμίσεων της προσομοίωσης (omnetpp.ini), δημιουργείται και ένα σύνολο από αρχεία αποτελεσμάτων.

Στο παράδειγμά μας, δημιουργούνται τα αρχεία αποτελεσμάτων:

- client1.pcap (επειδή ζητήθηκε στο αρχείο omnetpp.ini).
- server.pcap (επειδή ζητήθηκε στο αρχείο omnetpp.ini).
- General-0.elog.
- General-0.sca (για αποτελέσματα απλών αριθμών - scalar).
- General-0.vec (για αποτελέσματα διανυσμάτων ως προς χρόνο - vector).



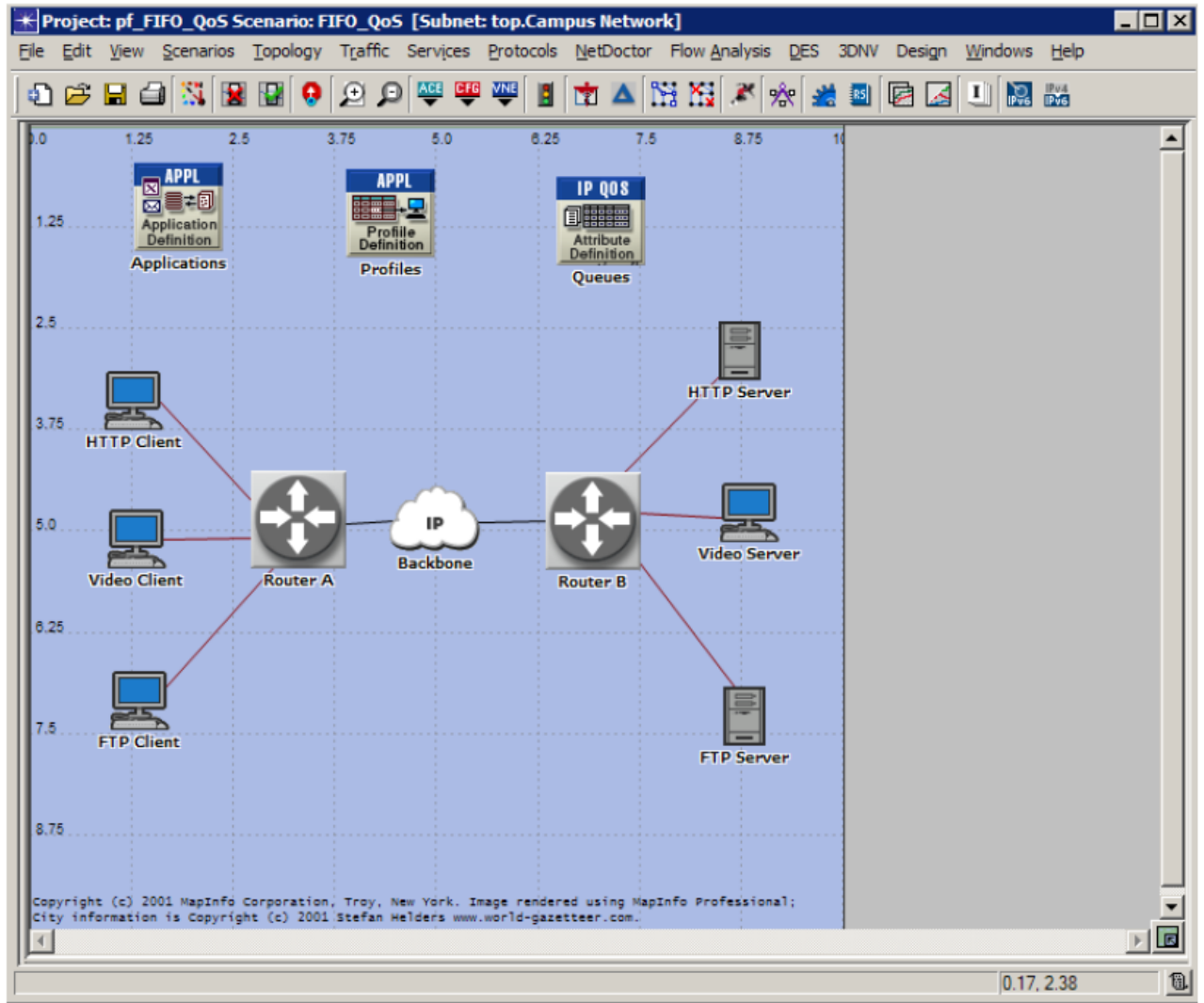
Εικόνα 9: Ιστόγραμμα του μέσου χρόνου αναμονής στην ουρά του εξυπηρέτη.

Κάνοντας διπλό κλικ σε ένα από τα αρχεία '.sca' ή '.vec', μας ζητείται να δημιουργηθεί ένα αρχείο μορφής '.anF', το οποίο εφόσον γίνει θα περιέχει όλα ή όποια αποτελέσματα από τα αρχεία αποτελεσμάτων επιλέξουμε,

ώστε στην συνέχεια να προσπαθήσουμε να κάνουμε οπτικοποίηση με τα ενσωματωμένα εργαλεία. Για παράδειγμα, έστω ότι θέλουμε να δούμε τον χρόνο αναμονής των πακέτων στην ουρά του εξυπηρέτη (κόμβος 'server'). Επειδή αυτό είναι μία ακολουθία από απλούς αριθμούς (χρόνος), αυτομάτως τοποθετείται ως επιλογή οπτικοποίησης (data visualization) σε ιστόγραμμα. Επιλέγοντάς την με διπλό κλικ, εμφανίζεται το ιστόγραμμα που βλέπουμε στην Εικόνα 4.3. Ουσιαστικά βλέπουμε την κατανομή του πλήθους των πακέτων στην ουρά του εξυπηρέτη, ως προς τον χρόνο αναμονής τους εκεί.

4.4 Ο προσομοιωτής OPNET

Το OPNET (Optimized Network Engineering Tool) είναι ένα εμπορικό πρόγραμμα προσομοίωσης δικτύων. Το περιβάλλον του είναι πλήρως γραφικό και τρέχει σε Windows της Microsoft. Εκτός από την ίδια περίπου φιλοσοφία με τους προηγούμενους προσομοιωτές, είναι γραμμένος σε C++ για μέγιστη δυνατή ταχύτητα. Αν και είναι δυνατή η τροποποίηση, προσθήκη ή επέκταση κάθε στοιχείου στην βιβλιοθήκη του σε C++, αυτή η δυνατότητα δεν υπάρχει για την ακαδημαϊκή του έκδοση, η οποία διατίθεται δωρεάν για 6 μήνες με δωρεάν απλή εγγραφή και σχετική ψηφιακή άδεια χρήσης, η οποία είναι ανανεώσιμη. Επίσης, η έκδοση αυτή δεν περιέχει όλες τις δυνατότητες και βιβλιοθήκες της πλήρους εκδόσεως.

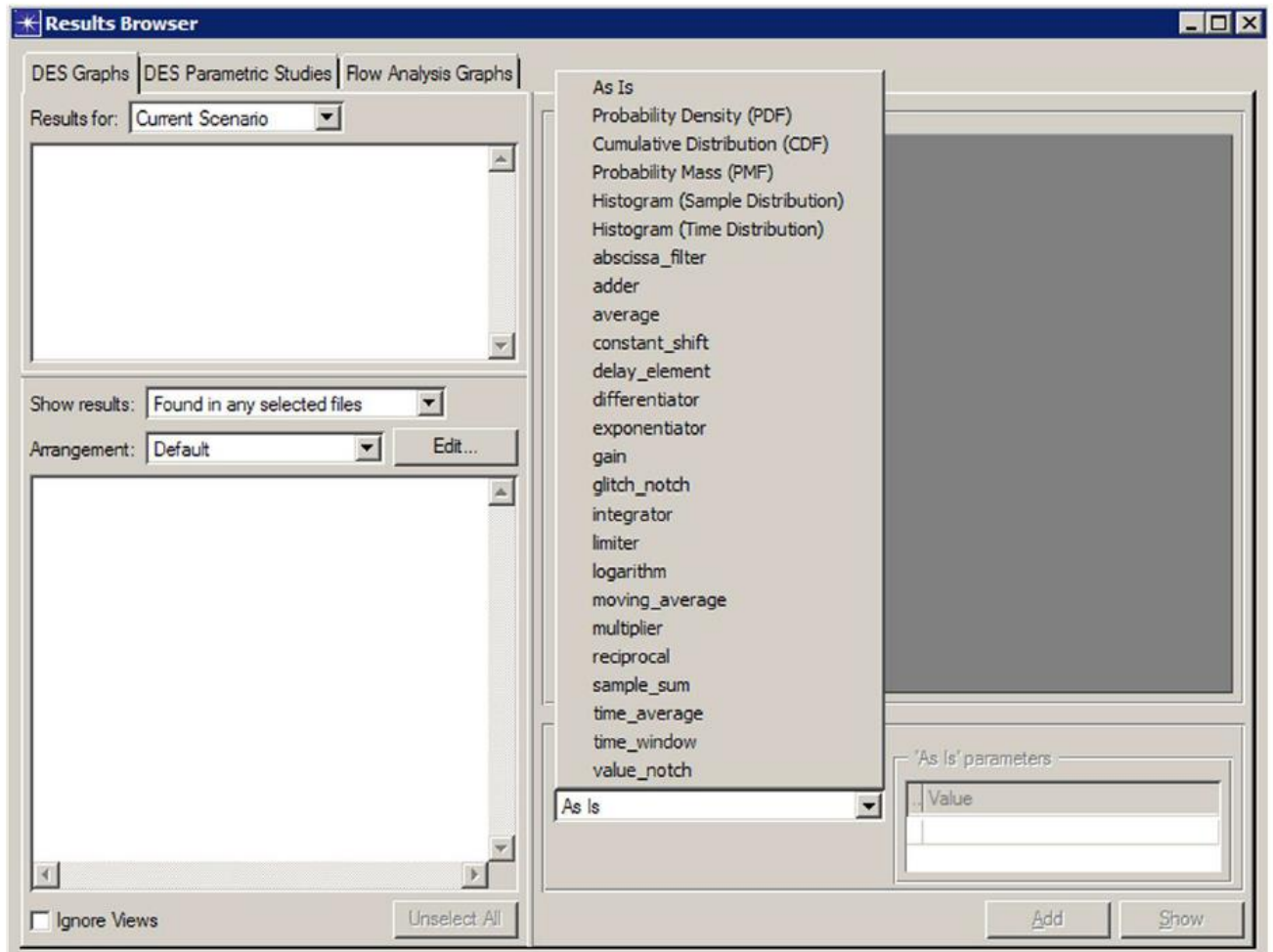


Εικόνα 10: Παράδειγμα μοντέλου δικτύου στον Project Editor του OPNET.

Στην πλήρη του έκδοση έχει το ιδιαίτερο χαρακτηριστικό της κυριολεκτικά τεράστιας βιβλιοθήκης από δικτυακά στοιχεία, και τρεις διαφορετικούς editor για επεξεργασία μοντέλων, ανάλογα με το επίπεδο ιεραρχίας στο οποίο ανήκουν:

- Project (ή Network) Editor. Αυτός είναι ο πλέον συχνά χρησιμοποιούμενος, επειδή με αυτόν κατασκευάζουμε το μοντέλο του δικτύου που θέλουμε να προσομοιώσουμε, χρησιμοποιώντας τύπους κόμβων, ζεύξεων και υποδικτύων από αντίστοιχες παλέτες, καθώς και επιλέγοντας τα είδη στατιστικών στοιχείων για τα οποία θέλουμε να πάρουμε αποτελέσματα. Επίσης εδώ τρέχουμε τις προσομοιώσεις και βλέπουμε τα αποτελέσματά τους μέσω γραφημάτων τα οποία δημιουργούνται αυτόματα με μία απλή επιλογή από το γραφικό περιβάλλον. Το περιβάλλον φαίνεται στην Εικόνα 4.4.
- Node Editor. Χρησιμοποιείται για να κατασκευάσουμε νέα μοντέλα κόμβων. Εδώ ορίζουμε την εσωτερική δομή του κόμβου, ο οποίος μπορεί να είναι από ολόκληρος υπολογιστής, έως ένας απλός αισθητήρας. Η δομή του κάθε κόμβου είναι αρθρωτή (modular), με τα πακέτα και τις δυνατές καταστάσεις να ανταλλάσσονται μεταξύ των modules, τα οποία είναι συνδεδεμένα μεταξύ τους, είτε με ρεύματα πακέτων, είτε με καλώδια στατιστικών χαρακτηριστικών. Κάθε module πρέπει να έχει μία συγκεκριμένη λειτουργία (π.χ., γεννήτρια πακέτων, επεξεργασία πακέτων, μετάδοση, τοποθέτηση σε ουρά, κλπ.).
- Process Editor. Χρησιμοποιείται για να σχεδιάζουμε μοντέλα επεξεργασίας, που ελέγχουν την βασική λειτουργικότητα των μοντέλων των κόμβων. Αυτά τα μοντέλα επεξεργασίας αναπαρίστανται με μηχανές πεπερασμένων

καταστάσεων (FSM - Finite State Machine). Αυτές εκφράζονται σε γλώσσα C++ ή C.



Εικόνα 11: Επιλογές οπτικοποίησης αποτελεσμάτων προσομοίωσης στον OPNET.

Κάθε ένας τύπος αντικειμένου έχει έτοιμο το σύνολο των στατιστικών στοιχείων που μπορεί να συλλέξει. Συνεπώς, το μόνο που χρειάζεται είναι να πάμε και να κάνουμε ένα δεξί κλικ στην επιφάνεια του καμβά και να επιλέξουμε από το σχετικό αναδυόμενο παράθυρο την επιλογή 'Choose Individual DES Statistics'. Εδώ εμφανίζεται ένα άλλο παράθυρο, από όπου μπορούμε να επιλέξουμε το είδος των στατιστικών στοιχείων που επιθυμούμε να συλλεχθούν κατά την προσομοίωση. Υπάρχουν τρεις κατηγορίες:

- Global Statistics. Στατιστικά που αφορούν το συνολικό δίκτυο.
- Node Statistics. Στατιστικά που έχουν νόημα κατά κόμβο.
- Link Statistics. Στατιστικά που έχουν νόημα κατά ζεύξη.

Για παράδειγμα, μπορούμε να ζητήσουμε να συλλεχθούν στατιστικά στοιχεία για το σύνολο των byte που ελήφθησαν από κάποιον κόμβο ή τον αντίστοιχο ρυθμό τους, αλλά και το πλήθος των πακέτων που απεστάλησαν σε όλο το δίκτυο.

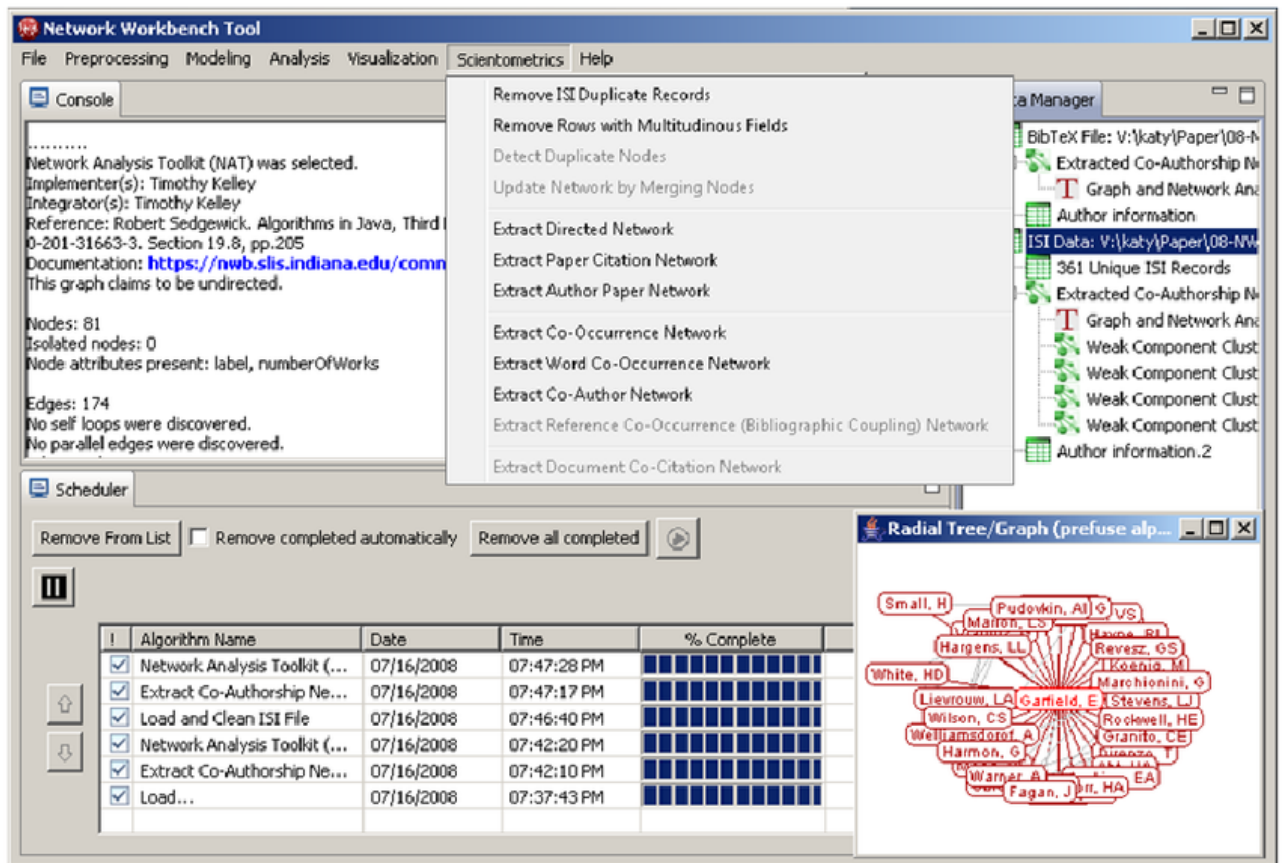
Κάθε στοιχείο εμφανίζει σε μία αναδιπλούμενη λίστα τα στατιστικά που είναι διαθέσιμα προς συλλογή. Με το τέλος της προσομοίωσης, μπορούμε από το κεντρικό μενού να επιλέξουμε 'DES' ^ 'Results' ^ 'View Results'. Εμφανίζεται ένα παράθυρο το οποίο μας επιτρέπει πολλούς τρόπους οπτικοποίησης των αποτελεσμάτων, χωρίς γενικά να χρειάζεται κάποιο εξωτερικό εργαλείο, όπως φαίνεται στην Εικόνα 4.5. Εκτός από την εξ ορισμού απεικόνιση ('As Is') υπάρχει πληθώρα από χρήσιμους τύπους απεικονίσεων που είναι διαθέσιμοι με μία απλή επιλογή από την αναδυόμενη λίστα.

4.5 Προσομοιωτής Network Workbench

Ο Network Workbench [17] είναι ένας προσομοιωτής δικτύου, μεθόδου διακριτών γεγονότων, ο οποίος αναπτύχθηκε για την ακαδημαϊκή έρευνα του πρωτοκόλλου του Internet. Περιέχει μια πλήρη στοίβα πρωτοκόλλων, που αντλείται από το Διαδίκτυο και μια σειρά ασκήσεων που επικεντρώνονται στους κρίσιμους αλγορίθμους πρωτοκόλλων στο Internet. Ο Network Workbench έχει αρκετές εκδόσεις:

- Την έκδοση 0 (1994),
- Την έκδοση 1 (1994),
- Την έκδοση 2 (1997), και
- την έκδοση 3 (1998).

Αυτό το εργαλείο υποστηρίζει αρκετές τοπολογίες δικτύων, DLC έλεγχο σφαλμάτων, CSMA / CD collision υποχώρησης, βέλτιστο υπολογισμό διαδρομής, αξιόπιστη μεταφορά, multicast, και την ένταξη / MAN LAN. Περιέχει πέντε layers (application, transport, network, datalink control και physical). Το εργαλείο Network Workbench μπορεί να χρησιμοποιηθεί για τη μελέτη της τοπολογίας WAN, μορφοποίηση πλαισίου (DLC – Datalink Control Layer), ροή DLC και σφάλμα ελέγχου, CSMA / CD τοπικό δίκτυο δρομολόγησης στρώματος δικτύου, αξιόπιστη μεταφορά. Η γλώσσα που υποστηρίζει είναι η C ++. Το μειονέκτημα του προσομοιωτή αυτού είναι η πολύ περιορισμένη εφαρμογή των πρωτοκόλλων.



Εικόνα 12: Γραφικό περιβάλλον του προσομοιωτή Network Workbench

4.6 Ο προσομοιωτής Netsim

Ο Netsim (Network Simulator MIT) [18,19] είναι ένας προσομοιωτής μεθόδου διακριτών γεγονότων που χρησιμοποιείται για τη διερεύνηση πολλών πτυχών του τοπικού δικτύου (LAN). Ο Netsim έχει τρεις κύριους στόχους:

- την ευελιξία του πειράματος προσδιορισμού του δικτύου και της κυκλοφορίας (traffic)
- προσομοίωση της ακριβής συμπεριφοράς του Ethernet
- έχει χαρακτηριστικά που καθιστούν τις τρεχούμενες ακολουθίες των σχετικών πειραμάτων ευκολότερες.

Τα πειραματικά δεδομένα του Netsim προσομοιωτή αποθηκεύονται σε ένα αρχείο περιγραφής (description file) το οποίο περιέχει:

- τη διάρθρωση του δικτύου που πρόκειται να προσομοιωθεί,
- τη συμπεριφορά παραγωγής κίνησης του σταθμού στο δίκτυο,
- πληροφορίες σχετικά με την επανάληψη της κάθε προσομοίωσης, καθώς και για την ακολουθία παραμέτρων η οποία πρόκειται να χρησιμοποιηθεί στα πειράματα.

Οι είσοδοι προσδιορίζουν φυσικά χαρακτηριστικά: μήκος (length), μέγεθος πακέτου (packet size), ρυθμό δεδομένων (data rate), αριθμό προσκολλημένων σταθμών (number of attached stations) και τη συμπεριφορά δημιουργίας κυκλοφορίας (traffic generation) των προσκολλημένων σταθμών.

Υπάρχουν πέντε τύποι κατανομών κυκλοφορίας του Netsim:

- εκθετική (exponential)
- ομοιόμορφη (uniform)
- ντετερμινιστική (deterministic)
- συνεχής (continuously queued)
- και διακριτή κατανομή ορισμένη από το χρήστη (user defined)

Το Netsim συλλέγει πληροφορίες για πολλές πτυχές της λειτουργίας της προσομοίωσης δικτύου. Για παράδειγμα, τη μέση καθυστέρηση πακέτων (average packet lagging), το ιστόγραμμα των καθυστερήσεων των πακέτων αναμονής (histogram of packet queuing delays), τα εμπόδια ανά μεταφερόμενο πακέτο (collision per packet transmitted, ποσοστό χρησιμοποίησης (percentage utilization), το πραγματικό ποσοστό δεδομένων (actual data rate), διασπορά καθυστέρησης σειράς (variance of queuing delay), παρατηρούμενος ρυθμός μετάδοσης πακέτων (observed packet transmission rate), μέσο μέγεθος του μηνύματος (average message size), συνολικό παρατηρούμενο ποσοστό εμποδίων (total observed collision rate), συνολικό αριθμό των πακέτων, το μέγιστο μέγεθος του πακέτου, το ελάχιστο μέγεθος του πακέτου, τον μέγιστο ρυθμό δεδομένων κλπ. Το Netsim υποστηρίζει: σύνδεση Ethernet, μια σύνδεση από σημείο σε σημείο, ένα διακόπτη (διακόπτες πακέτων μεταξύ διαφόρων συνδέσμων), μια

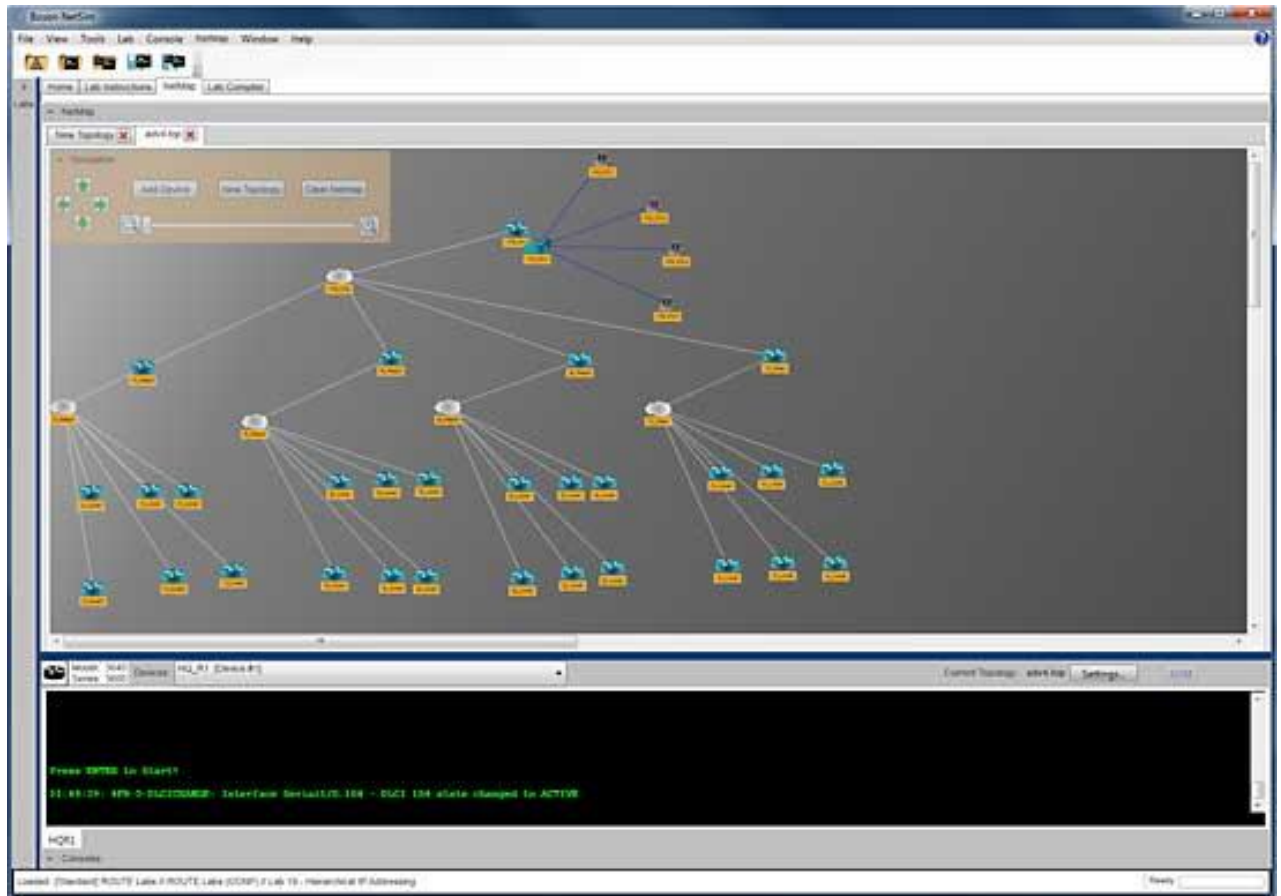
σειρά (περίπου το ίδιο με ένα διακόπτη), υλοποίησης Purdue της ιόντων του πρωτοκόλλου TCP, την παροχή στοιχείων και των καταναλωτών από το TCP, μια απλή Poisson πηγή κυκλοφορίας και ένα packet sink. Η γλώσσα που υποστηρίζει είναι η C.

Τα πλεονεκτήματα του Netsim είναι:

- Το Netsim είναι ένας διαθέσιμος στο κοινό προσομοιωτής ATM που αναπτύχθηκε αρχικά στο MIT
- ο πηγαίος κώδικας είναι ελεύθερα διαθέσιμος και επεξεργάσιμος
- Το Netsim έχει ένα φιλικό προς το χρήστη γραφικό περιβάλλον (Graphical User Interface - GUI).

Τα μειονεκτήματα του Netsim είναι:

- Το GUI, ωστόσο είναι πολύ πρωτόγονο και ανεπαρκές για εκπαιδευτικούς σκοπούς. Δεν είναι ανεκτικό σε σφάλματα σε αρχάριους χρήστες. Μια λανθασμένη ακολουθία των πλήκτρων ή κλικ του ποντικιού θα μπορούσε να προκαλέσει συντριβή της προσομοίωσης
- Περιορισμένη χρήση.



Εικόνα 13: Γραφικό Περιβάλλον του προσομοιωτή Netsim

4.7 Ο προσομοιωτής MARS

Ο MARS (Maryland Routing Simulator) [20,21] είναι προσομοιωτής μεθόδου διακριτών γεγονότων που παρέχει μια ευέλικτη πλατφόρμα για την αξιολόγηση και σύγκριση αλγορίθμων δρομολόγησης δικτύων (network routing algorithms). Το MARS δομείται σε δύο μέρη: το μηχανισμό προσομοίωσης ο οποίος έχει να κάνει με τη λίστα δεδομένων και το γραφικό περιβάλλον, και ένα σύνολο συνιστωσών για την μοντελοποίηση των δικτύων και το χειρισμό διαφόρων λειτουργιών προσομοίωσης.

Το MARS έχει χρησιμοποιηθεί για την αξιολόγηση και την σύγκριση διαφόρων αλγορίθμων δρομολόγησης τύπου next-hop

- Πρωτόκολλο δρομολόγησης Loop-free Bellman-Ford χωρίς επίδραση αναπήδησης (bouncing effect)
- ένας αλγόριθμος τύπου link-state ο οποίος είναι ένας νέος αλγόριθμος δρομολόγησης για το ARPANET.

Το MARS είναι περιορισμένο λόγω της έλλειψης της δυνατότητας του scrolling στο παράθυρο του γραφικού περιβάλλοντος του χρήστη. Έτσι, οι χρήστες περιορίζονται σε αυτά που είναι συγκεντρωμένα απεικονίζονται σε ένα παράθυρο χωρίς τη δυνατότητα του scrolling. Επιπροσθέτως, το MARS περιορίζεται σε μία δομή δικτύου από point-to-point και κατά συνέπεια δεν είναι άμεσα κατάλληλο

για την προσομοίωση δικτύων με την επικοινωνία εκπομπής (broadcast communication)

Το MARS υποστηρίζει πρωτόκολλα δρομολόγησης του δικτύου (SPF, Merlin_Segall, Bellman_Ford), FTP, Telnet, Workload (Poison, ομοιόμορφη).

Οι γλώσσα που υποστηρίζει είναι η C.

Τα μειονεκτήματα του MARS είναι τα εξής:

- Περιορισμένη υποστήριξη του πρωτοκόλλου transport layer και των πηγαίων μοντέλων εφαρμογής (application source models)

4.8 Ο προσομοιωτής COMNET III

Το COMNET III [22,23,24] είναι ένας εμπορικός ολοκληρωμένος αντικειμενοστραφής προσομοιωτής μεθόδου διακεκριμένων γεγονότων για τη μοντελοποίηση και την ανάλυση απόδοσης δικτύων υπολογιστών. Με το COMNET III, οι χρήστες μπορούν να δημιουργήσουν μια ποικιλία αρχιτεκτονικών δικτύου, συμπεριλαμβανομένων LANs, MANs, WANs, μεταγωγή πακέτων, ATM, αναμετάδοση πλαισίου και ούτω καθεξής. Το ComNet χρησιμοποιείται εξ ολοκλήρου από το χρήστη με χρήση του γραφικού περιβάλλοντος. Ο χρήστης έχει τη δυνατότητα να αναπαριστά γραφικά traffic modelling patterns,

χρησιμοποιώντας ρεαλιστικά αντικείμενα του δικτύου ώστε να αντικατοπτρίζει πραγματικά δίκτυα και να εφαρμόσει τις έννοιες του δικτύου.

Το COMNET III επιτρέπει στους χρήστες να διαμορφώσουν μοντελοποιήσουν την απόδοση των διαφόρων τύπων δικτύων. Παρέχει μια εκτενή βιβλιοθήκη των κόμβων, συνδέσμων, των πρωτόκολλων και των αντικειμένων της κυκλοφορίας.

Η βιβλιοθήκη αντικειμένων συνδέσμου περιλαμβάνει δύο κλάσεις:

- συνδέσεις σημείου προς σημείο (point to point link)
- συνδέσεις πολλαπλής πρόσβασης (multi-access link).

Πρωτόκολλα πολλαπλής πρόσβασης, που μπορούν να διαμορφωθούν, περιλαμβάνουν CSMA / CD, CSMA, το ALOHA, Token BUS, Token Ring, FDDI και Polling.

Το COMNET III υποστηρίζει υποδίκτυα και έχει δύο είδη πηγών:

- πηγές εφαρμογής
- πηγές κυκλοφορίας.

Υπάρχουν τέσσερις τύποι γεννητριών κίνησης:

- πηγές μηνύματος (message sources),
- πηγές συνεδρίας (session sources),
- πηγές ανταπόκρισης (response sources)
- πηγές κλήσης (call sources).

Το COMNET III είναι σε θέση να παράγει περίπου 100 αναφορές για διαφορετικό μοντέλο δομικών στοιχείων. Συνηθέστερα παράγει αναφορές οι οποίες περιλαμβάνουν αναφορές για χρήση κόμβων (node utilization), καθυστερήσεις εφαρμογών (application delay), καθυστερήσεις συνδέσμων (link delay), αξιοποίηση του καναλιού (channel utilization), καθυστερήσεις μηνυμάτων (message delay), καθυστερήσεις πακέτων (packet delay), μπλοκαρισμένες κλήσεις (blocked calls), κλπ.

Το COMNET III υποστηρίζει LAN (Ethernet, Token Ring, FDDI), MAN, internet, μεταγωγής πακέτων, διακόπτης κυκλώματος, ATM, εκρηκτική κίνηση, CSMA, CSMA / CD, ALOHA, Polling, πρωτόκολλα (TCP / IP, IPX, ΣΕΛ, DECnet).

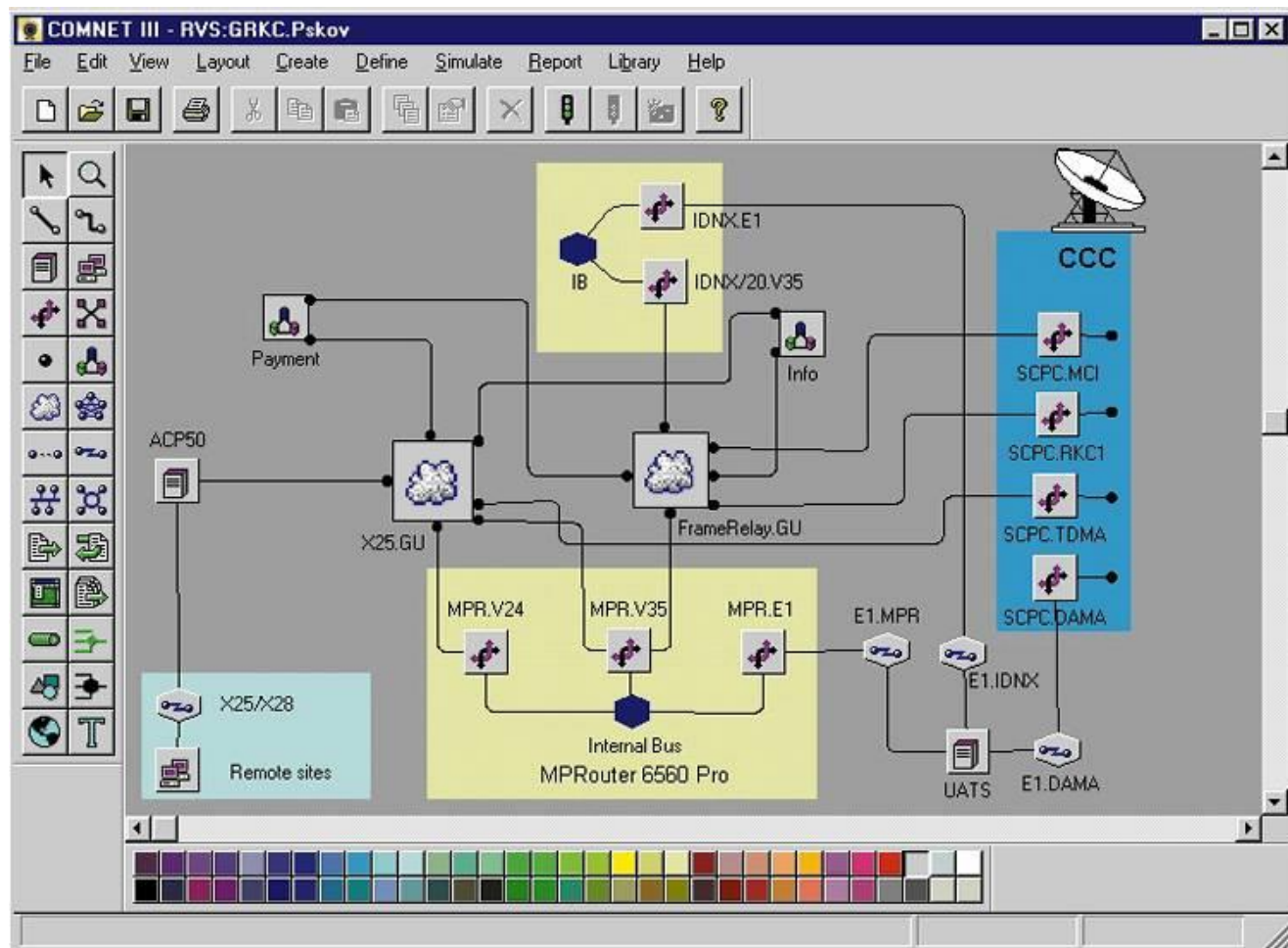
Τα πλεονεκτήματα του COMNET III είναι:

- είναι ένα πακέτο το οποίο παρέχει τη γρήγορη και εύκολη ανάλυση και την πρόβλεψη της απόδοσης του δικτύου υπολογιστών
- επιτρέπει στους χρήστες την ευελιξία να δοκιμάσουν έναν απεριόριστο αριθμό σεναρίων τύπου «what if?», δηλαδή να δοκιμάζουν διαφορετικές παραμέτρους και να βλέπουν την επίδρασή τους στα αποτελέσματα του μοντέλου
- Παρέχει ρεαλιστικά και ακριβή αποτελέσματα.

Τα μειονεκτήματα του COMNET III είναι:

- Ο πηγαίος κώδικας δεν είναι διαθέσιμος

- δύσκολη η προσθήκη νέων modules
- μοντέλα με βάση το περιορισμένο σύνολο πρωτοκόλλων δικτύων το οποίο παρέχει
- δεν μπορεί να προγραμματίζεται από τους χρήστες



Εικόνα 14: Γραφικό περιβάλλον του προσομοιωτή COMNET III

4.9 Ο προσομοιωτής REAL

Το REAL (REalistic And Large) [25] είναι ένας προσομοιωτής δικτύου που αρχικά προορίζονταν για τη μελέτη της δυναμικής συμπεριφοράς των συστημάτων ροής και ελέγχου συμφόρησης (congestion control) σε δίκτυα μεταγωγής πακέτων δεδομένων. Παρέχει περίπου 30 modules τα οποία εξομοιώνουν επακριβώς τις δράσεις αρκετών γνωστών πρωτοκόλλων ελέγχου ροής (όπως το TCP), και 5 αλγορίθμους χρονοπρογραμματισμού, όπως ο Fair Queuing και ο Hierarchical Round Robin). Υπάρχουν σχεδόν 30 τύποι πηγών, που αντιστοιχούν σε περίπου 30 τύπους πρωτόκολλων μεταφοράς και workloads. Οι πηγές μπορούν να ταξινομηθούν σε δύο τύπους:

- σε πηγές μη-ελεγχόμενης ροής δεδομένων
- σε πηγές ελεγχόμενης ροής δεδομένων

Το REAL έχει ένα γραφικό περιβάλλον (GUI), υλοποιημένο σε Java. Ο προσομοιωτής παίρνει ως είσοδο για ένα δίκτυο την περιγραφή της τοπολογίας του δικτύου, τα πρωτόκολλα, φόρτο εργασίας (workload) και τον έλεγχο των παραμέτρων.

Οι χρήστες θα πρέπει καθορίσουν κάποιες παραμέτρους δικτύου, το πρωτόκολλο μεταφοράς (transport protocol) (ειδικότερα, ο έλεγχος ροής) και το workload κάθε πηγής. Τέλος, οι χρήστες πρέπει να προσδιορίζουν τις παραμέτρους ελέγχου, όπως

το λανθάνοντα χρόνο και το εύρος ζώνης της κάθε γραμμής επικοινωνίας, τα μεγέθη πακέτων κλπ.

Το REAL υποστηρίζει TCP / IP, XNS, FTP, Telnet, ill brhaved, FIFO, FCFS, Fair Queuing (FQ),

και τον αλγόριθμο Hierarchical Round Robin.

Οι γλώσσες που υποστηρίζει είναι οι C και JAVA (GUI μόνο).

Τα πλεονεκτήματα του REAL είναι:

- το GUI επιτρέπει στους χρήστες να κάνουν γρήγορα σενάρια για την διεκπεραίωση της προσομοίωσης με μια utility point-and-click
- Ο πηγαίος κώδικας παρέχεται έτσι ώστε οι ενδιαφερόμενοι χρήστες μπορούν να τροποποιήσουν τον προσομοιωτή για δικούς τους σκοπούς
- Είναι επεκτάσιμο (extendible)

Τα μειονεκτήματα του REAL είναι τα εξής:

- Το εργαλείο NEST δεν επιτρέπει τη χρήση χρονομέτρων. Το REAL στέλνει από ένα πακέτο χρονομέτρου από μια πηγή πίσω στον εαυτό του για να επιστρέψει μετά από κάποιο συγκεκριμένο χρονικό διάστημα, αλλά τα χρονόμετρα δεν μπορεί να αρχικοποιηθούν χρησιμοποιώντας αυτή τη μέθοδο
- Ο προσομοιωτής χάνει κάποιες από τις συμπεριφορές στην κοινή εφαρμογή πρωτοκόλλου
- Δεν υποστηρίζουν την συνήθεις εφαρμογές πρωτοκόλλων. Αντί αυτού, χρησιμοποιεί κώδικες με τους οποίους προσομοιώνει την πλειοψηφία των χαρακτηριστικών των πρωτοκόλλων.

4.10 Ο προσομοιωτής Dummynet

Το Dummynet [26,27] είναι ένα απλό, ακριβές και ευέλικτο εργαλείο προσομοίωσης δικτύου με ελάχιστη τροποποίηση σε μια υπάρχουσα στοίβα πρωτοκόλλου, επιτρέποντας πειράματα ώστε να τρέχει σε ένα αυτόνομο σύστημα και μπορεί να χρησιμοποιηθεί για την προσομοίωση δικτύου με αυθαίρετες τοπολογίες.

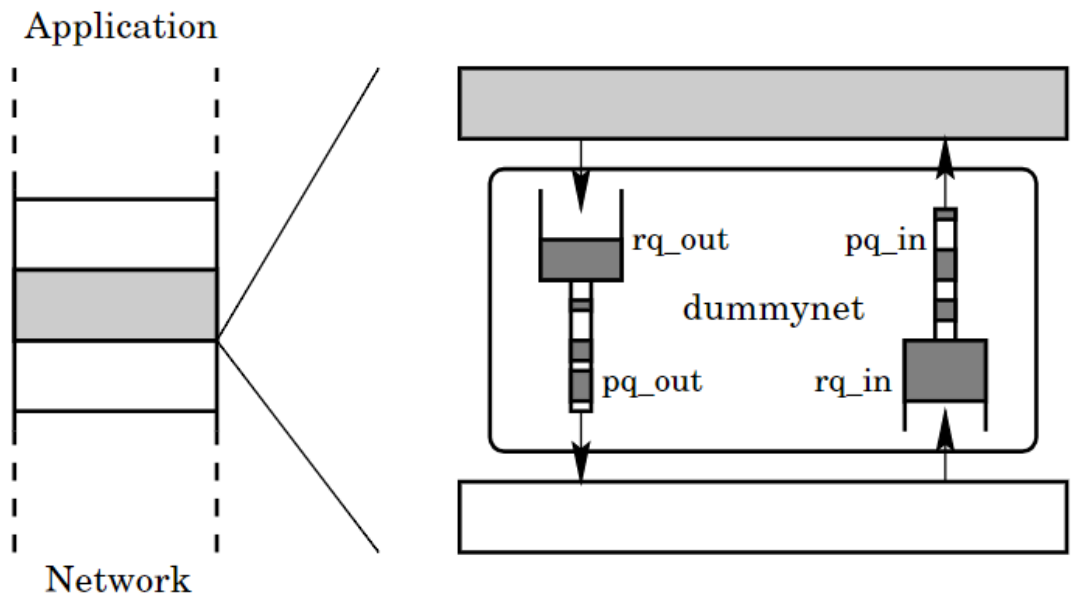
Το Dummynet λειτουργεί με την παρεμπόδιση των επικοινωνιών του layer του υπό δοκιμή πρωτοκόλλου και με την προσομοίωση των επιδράσεων που προκαλούν οι πεπερασμένες ουρές (finite queues), οι περιορισμοί εύρους ζώνης, καθυστερήσεις επικοινωνίας (communication delays) και, ενδεχομένως, απώλειες συνδέσης.

Το εργαλείο επιτρέπει την χρήση των πραγματικών γεννητριών κυκλοφορίας και εφαρμογής πρωτοκόλλων, λύνοντας έτσι το πρόβλημα προσομοίωσης ασυνήθιστου περιβάλλοντος. Η διεξαγωγή ενός πειράματος με αυτό το εργαλείο είναι τόσο εύκολη και γρήγορη όσο τρέχει το επιθυμητό σύνολο των εφαρμογών σε ένα σταθμό εργασίας και ως συνέπεια, δεν θα μπορούν να πραγματοποιηθούν παραπάνω πειράματα από αυτά που μπορούν να πραγματοποιηθούν με τη μέγιστη ταχύτητα λειτουργίας του συστήματος κατά τη χρήση.

Η βασική έκδοση του Dummynet λειτουργεί στη διεπαφή (interface) μεταξύ TCP και IP. Η εφαρμογή εκτείνεται σε λιγότερο από 300 γραμμές κώδικα του πυρήνα στο FreeBSD. Υπό κανονικές συνθήκες, δεν υπάρχει υπερυψωμένο σύστημα. Η

αρχή λειτουργίας της εφαρμογής είναι η εφαρμογή στοίβας τυπικών πρωτοκόλλων, όπου κάθε layer επικοινωνεί με τα γειτονικά. Στο Dummynet, για την προσομοίωση ενός δικτύου μεταξύ δύο επικοινωνούντων, τα ακόλουθα στοιχεία παρεμβάλλονται στη ροή των δεδομένων:

- Ένα router με φραγμένο μέγεθος ουρών
- Επικοινωνιακές συνδέσεις (pipes) με δεδομένο το εύρος ζώνης και τις καθυστερήσεις.



Εικόνα 15: Η αρχή λειτουργίας του Dummynet.

Οι απώλειες λόγω συμφόρησης προσομοιώνονται με χρήση φραγμένου μεγέθους ουρών. Τυχαία αναδιάταξη πακέτων προσομοιώνεται. Αυτά δείχνουν την αναξιопιστία των δικτύων. Σύμφωνα με [22], μια απλούστερη ρύθμιση εισάγεται στο Dummynet, η οποία περιλαμβάνει ένα ή δύο δρομολογητές και ένα σωλήνα. Οι αλγόριθμοι RED, FIFO εφαρμόζονται στο Dummynet. Ορισμένοι κανόνες φιλτραρίσματος που χρησιμοποιούνται για την επιρροή στην κυκλοφορία.

Αρκετά πρωτόκολλα μπορεί να ελεγχθούν με τη χρήση αυτού του εργαλείου και να εντοπίσουν τα σφάλματα και να τα διορθώσουν.

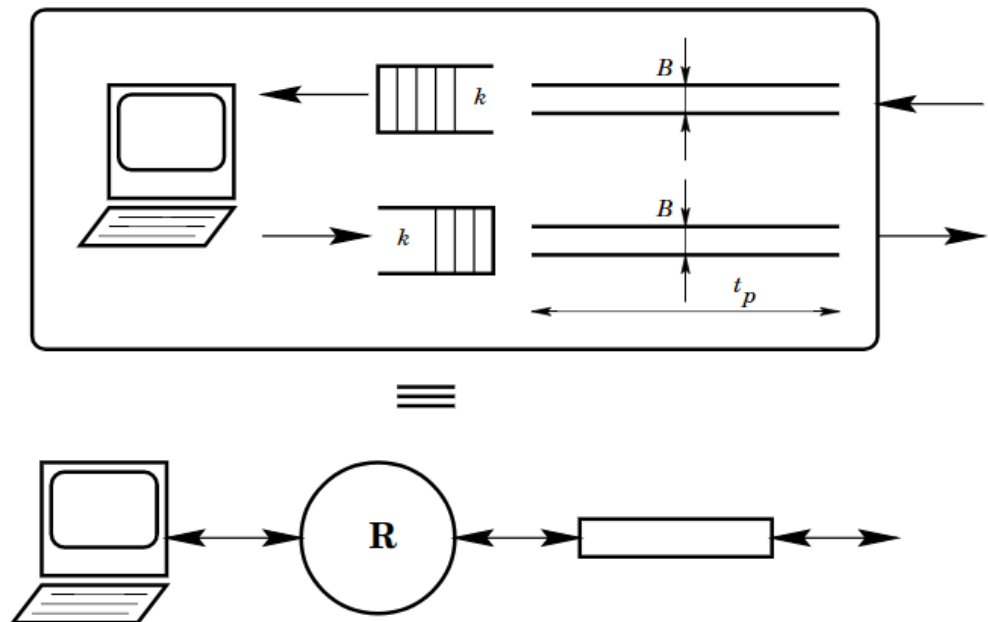
Σύμφωνα με την δημοσίευση [27], το εργαλείο Dummynet μπορεί να επεκταθεί για να προσομοιώνονται σύνθετα μοντέλα δικτύου και να απεικονίζονται γραφικά τα αποτελέσματα. Dummynet τρέχει στο FreeBSD και το αρχείο δεν αποθηκεύεται. Συνεπώς, η επιβάρυνση, η οποία εισάγεται στο εργαλείο είναι σχεδόν αμελητέα. Υποστηρίζει TCP / IP, FTP, Telnet, browsers Web, UDP και είναι επεκτάσιμο. Υποστηρίζει γλώσσα C.

Τα πλεονεκτήματα του Dummynet είναι τα εξής:

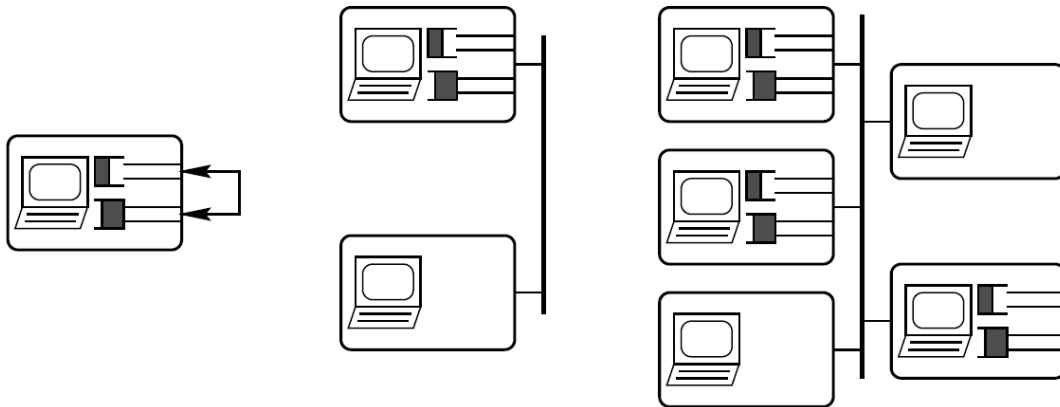
- Το πακέτο είναι δωρεάν.
- Ο πηγαίος κώδικας διαθέσιμος
- Μεγάλος έλεγχος των λειτουργικών παραμέτρων, απλότητα και διαθεσιμότητα να χρησιμοποιήσει την πραγματική γεννήτριες κυκλοφορίας
- υψηλή ακρίβεια

Τα μειονεκτήματα είναι τα εξής:

- Το Dummynet μπορεί να προσεγγίσει μόνο τις συμπεριφορές ενός πραγματικού συστήματος με δεδομένα χαρακτηριστικά
- Η περιοδικό task μπορεί να τρέξει αργά
- Γεγονότα στο Dummynet συμβαίνουν ταυτόχρονα με το χρονόμετρο του συστήματος, το οποίο θα μπορούσε να κρύψει ή να ενισχύσει κάποια πραγματικά φαινόμενα τα οποία συμβαίνουν λόγω συνθήκες της διεξαγωγής της προσομοίωσης.



Εικόνα 16: Δόμη ενός κόμβου στο Dummynet



Εικόνα 17: Διάφοροι τρόποι σχηματισμού πειραμάτων στο Netsim.

5. Παραδείγματα από εργασίες στην προσομοίωση δικτύων

Σε αυτό το κεφάλαιο θα γίνει παρουσίαση από διάφορες εργασίες πάνω στο θέμα της προσομοίωσης δικτύων.

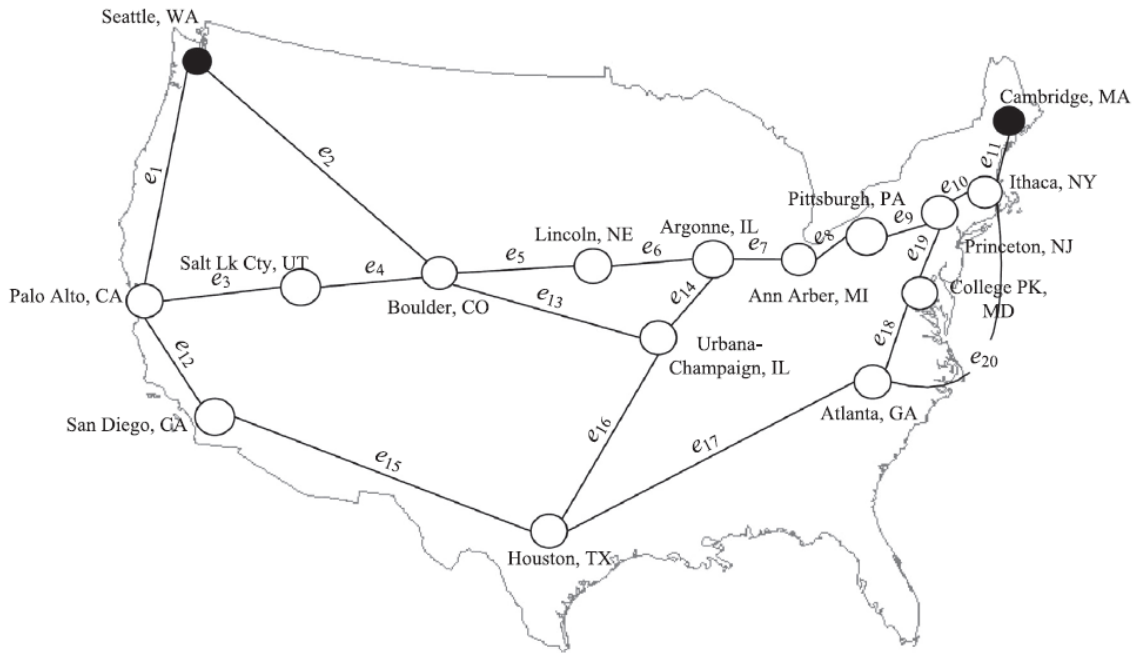
Αρχικά, στην εργασία [28] γίνεται μια προσέγγιση προσομοίωσης για την εκτίμηση της αξιοπιστίας ενός υπολογιστικού δικτύου, βασισμένο σε ένα μοντέλο χωρητικού δικτύου ροής (capacitated flow network - CFN), λαμβάνοντας υπόψιν συγκεκριμένο χρονικό περιθώριο.

Ένα τέτοιο CFN αναφέρεται ως δίκτυο δυναμικής ροής με βάση το χρόνο (time-based - capacitated flow network - TCFN). Η αποτυχία των φυσικών γραμμών που περιλαμβάνουν τις άκρες του TCFN μπορεί να παρουσιάσει συσχετισμένες αποτυχίες. Έτσι, η αποτυχία μιας φυσικής γραμμής μπορεί να συσχετιστεί με την αποτυχία φυσικών γραμμών σε γειτονικές ακμές του TCFN. Αυτή η προσέγγιση καταργεί την περιοριστική υπόθεση ότι οι ικανότητες των διαφορετικών άκρων είναι στατιστικά ανεξάρτητες [29]. Χρησιμοποιείται μια προσέγγιση προσομοίωσης για πολυπαραγοντικές κατανομές Bernoulli [30] για να χαρακτηριστούν συσχετισμένες αποτυχίες των φυσικών γραμμών που περιλαμβάνουν τις άκρες, επιτρέποντας ξεχωριστές αξιόπιστες σχέσεις και συσχετισμούς μεταξύ κάθε

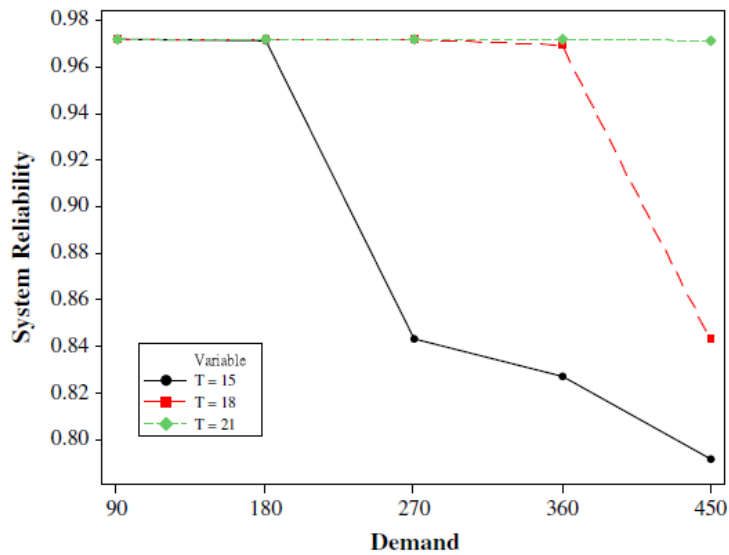
ζεύγους στοιχείων (φυσικές γραμμές) στο TCFN. Επομένως, η προσέγγιση αυτή είναι ικανή να εξετάσει γεγονότα παγκόσμιας κλίμακας που μπορούν να προκαλέσουν ευρεία ζημιά σε ένα δίκτυο και να υποβαθμίσουν σοβαρά την απόδοσή του. Επειδή η προσομοίωση πρέπει να είναι γρήγορη και ακριβής [31], μελετάται επίσης η εμπειρική απόδοση της προσέγγισής στα παραδείγματα δικτύων που μελετήθηκαν. Η συγκεκριμένη εργασία λοιπόν, αναπτύσσει μια προσέγγιση προσομοίωσης για τη μέτρηση του αντίκτυπου των συσχετισμένων βλαβών στην αξιοπιστία του συστήματος των δικτύων υπολογιστών που υπόκεινται σε χρονικούς περιορισμούς μετάδοσης, που αναφέρεται ως ένα δίκτυο χωρητικότητας ροής με βάση το χρόνο (TCFN).

Προτείνεται μια μέθοδος για να εξεταστούν συσχετισμένες αποτυχίες σε μεγάλη κλίμακα μεταξύ των στοιχείων στα άκρα και να παρουσιάζεται η προσέγγιση για τις περιπτώσεις τοπικών και γειτονικών συσχετίσεων. Έτσι, δόθηκε η δυνατότητα ρητά στις φυσικές γραμμές που περιείχαν τις άκρες του δικτύου υπολογιστών και στις φυσικές γραμμές στις γειτονικές ακμές να βιώσουν συσχετισμένες αποτυχίες, που απεικονίζουν την προσέγγισή μας σε μοντέλα παγκόσμιων συσχετισμών. Η προσέγγιση πρώτα βρίσκει τους φορείς ελάχιστης χωρητικότητας που σχηματίζουν τους συνδυασμούς ακραίων επιδόσεων που μπορούν να ικανοποιήσουν ένα δεδομένο επίπεδο ζήτησης d και ένα χρονικό περιορισμό T . Η συσχέτιση στις αποτυχίες των φυσικών γραμμών στη συνέχεια εισήχθη μέσω μίας τεχνικής για την προσομοίωση της πολλαπλής κατανομής Bernoulli. Πραγματοποιήθηκαν δύο πειράματα, συμπεριλαμβανομένης μιας περιπτωσιολογικής μελέτης μεγάλης κλίμακας του Εθνικού Δικτύου Επιστημονικών Ιδρυμάτων (NSFNET) και αναφέρθηκαν οι επιδόσεις αυτών των πειραμάτων προσομοίωσης. Παρατηρήθηκε ότι οι προσομοιώσεις για συγκεκριμένο χρονικό διάστημα και συσχέτιση μπορούν να ολοκληρωθούν μέσα σε λίγα λεπτά. Τα αποτελέσματα αποκάλυψαν ότι οι

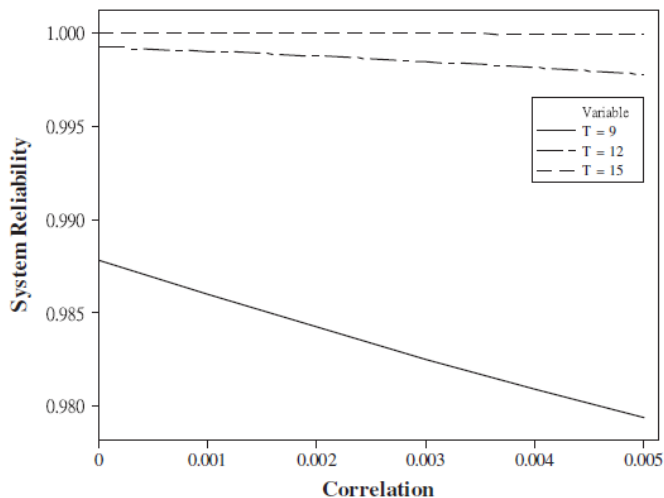
συσχετισμένες αποτυχίες μπορεί να έχουν θετικές ή αρνητικές επιπτώσεις στην αξιοπιστία του TCFN, δικαιολογώντας τον επιπλέον χρόνο που απαιτείται σε σχέση με τις προηγούμενες τεχνικές, οι οποίες υποθέτουν ότι τα ελαττώματα των συστατικών είναι στατιστικά ανεξάρτητα. Η τεχνική αυτή θα επιτρέψει στους σχεδιαστές δικτύων να διερευνήσουν τη δυνητική επίδραση συσχετισμών κατά το σχεδιασμό δικτύων ώστε να εξασφαλιστεί ότι οι απαιτήσεις και οι περιορισμοί χρόνου μπορούν να ικανοποιηθούν με ένα αποδεκτό επίπεδο πιθανότητας. Επιδεικνύεται επίσης η δυνατότητα εφαρμογής της προσέγγισης σε άλλους τύπους μοντέλων δικτύου, όπως η μετάδοση και η μεταφορά ενέργειας. Ένα πρόβλημα βελτιστοποίησης της αξιοπιστίας του συστήματος για τη μεγιστοποίηση της πιθανότητας ότι η ζήτηση d μπορεί να ικανοποιηθεί σε όχι μεγαλύτερες από μονάδες T time, παρά τη δυνατότητα συσχέτισης αποτυχίας, θα διερευνηθούν επίσης.



Εικόνα 18. Δίκτυο Εθνικού Επιστημονικού Ιδρύματος (NSFNET)



Εικόνα 19: Αξιοπιστία συστήματος



Εικόνα 20: Επιρροή της συσχέτισης για διαφορετικές τιμές χρονικού περιορισμού.

Στην εργασία [32] προτείνεται μια γενική προσέγγιση για τη μοντελοποίηση τροφοδοτικού δικτύου (supply network), για την οποία ισχύουν οι εξής παραδοχές:

- Το δίκτυο αποτελείται από ορισμένο αριθμό προμηθευτών, κατασκευαστών, χονδρεμπόρων και εμπόρων λιανικής πώλησης που αντιπροσωπεύουν κόμβους του δικτύου τροφοδοσίας.
- Ορισμένες διαδικασίες εκτελούνται σε κάθε κόμβο.
- Οι διεργασίες είναι διασυνδεδεμένες σε ένα μόνο κόμβο και μεταξύ των κόμβων.

Με αυτόν τον τρόπο, αντικατοπτρίζεται η οπτική παρουσίαση της αλυσίδας εφοδιασμού ως συστοιχία στοιχείων που συνδέονται με ευθείες γραμμές το δίκτυο των συνδεδεμένων στοιχείων. Αυτό μας επιτρέπει να μοντελοποιούμε τυχόν δίκτυα παροχής οποιουδήποτε τύπου (προμήθεια, παραγωγή, διανομή, κ.λπ.) και την πολυπλοκότητα (λίγοι ή πολλοί κόμβοι).

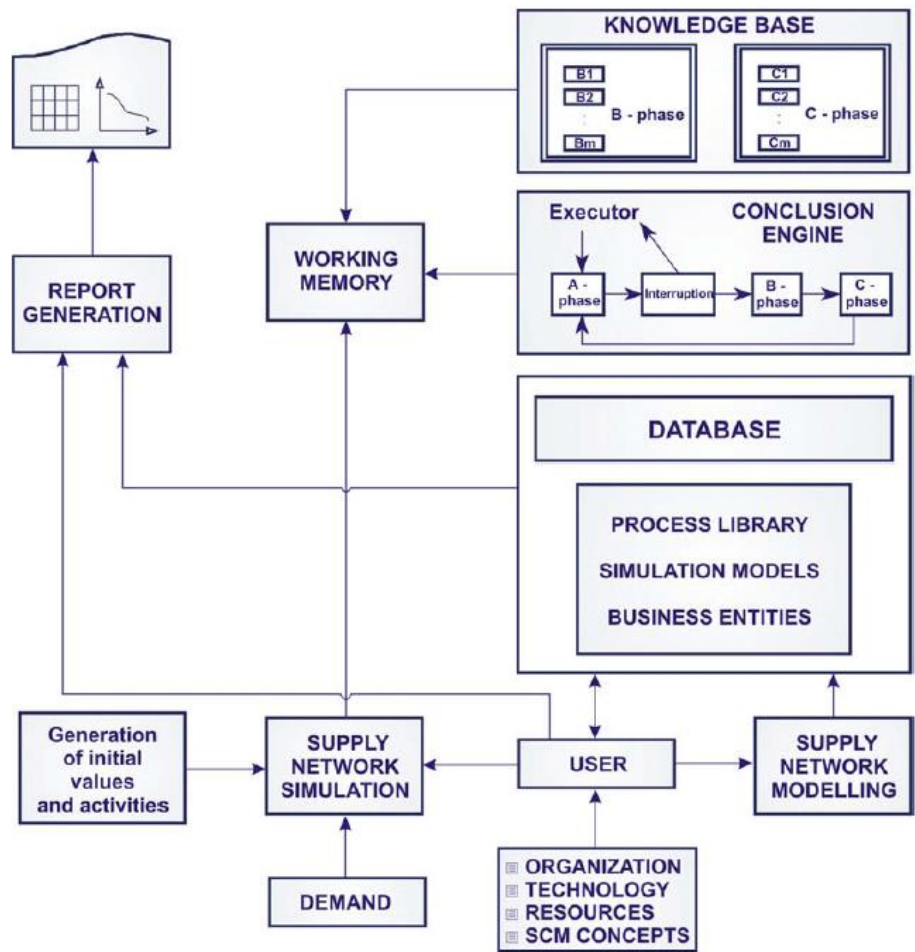
Κατά την κατασκευή του μοντέλου προσομοίωσης του δικτύου τροφοδοσίας ο σχεδιαστής μπορεί να λάβει υπόψη πολλά στοιχεία που βρίσκονται σε πραγματικό σύστημα, όπως το επιχειρηματικό περιβάλλον που αντιπροσωπεύεται από την τεχνολογία, την οργάνωση και τους περιορισμούς των πόρων. Ως αποτέλεσμα αυτής της προσέγγισης, η προσομοίωση εκτελείται σε συνθήκες οι οποίες είναι πολύ κοντά σε ένα πραγματικό δίκτυο παροχής. Η διαδικασία εκτέλεσης και οι αλληλεπιδράσεις τους εκτελούνται από το εξειδικευμένο σύστημα εξειδικευμένης παραγωγής μέσω ενσωματωμένης βάσης γνώσης και μηχανισμού συμπερασμάτων (conclusion engine). Η βάση γνώσεων περιλαμβάνει γνώση σχετικά με τη διαχείριση του δικτύου παροχής, τόσο από την καλύτερη πρακτική όσο και από τη θεωρία. Ένα άλλο αναπόσπαστο μέρος του λογισμικού προσομοίωσης είναι η βάση δεδομένων με το κατάλληλο σύνολο οντότητες που επιτρέπει τη μοντελοποίηση του δικτύου τροφοδοσίας, το σχεδιασμό μοντέλου προσομοίωσης και αποθηκεύει όλα τα αποτελέσματα προσομοίωσης. Όλα αυτά τα στοιχεία κάνουν ευκολότερη τη μοντελοποίηση και την προσομοίωση και επίσης διευκολύνουν τη χρήση του λογισμικού προσομοίωσης για τους τελικούς χρήστες.

Το σημείο εκκίνησης της προσέγγισης προσομοίωσης που παρουσιάζεται είναι η ιδέα ανάπτυξης μοντέλου προσομοίωσης στην οποία βασίζεται σε τέσσερα υπομοντέλα με σκοπό την πληρότητα και τη συνέπεια μοντέλων:

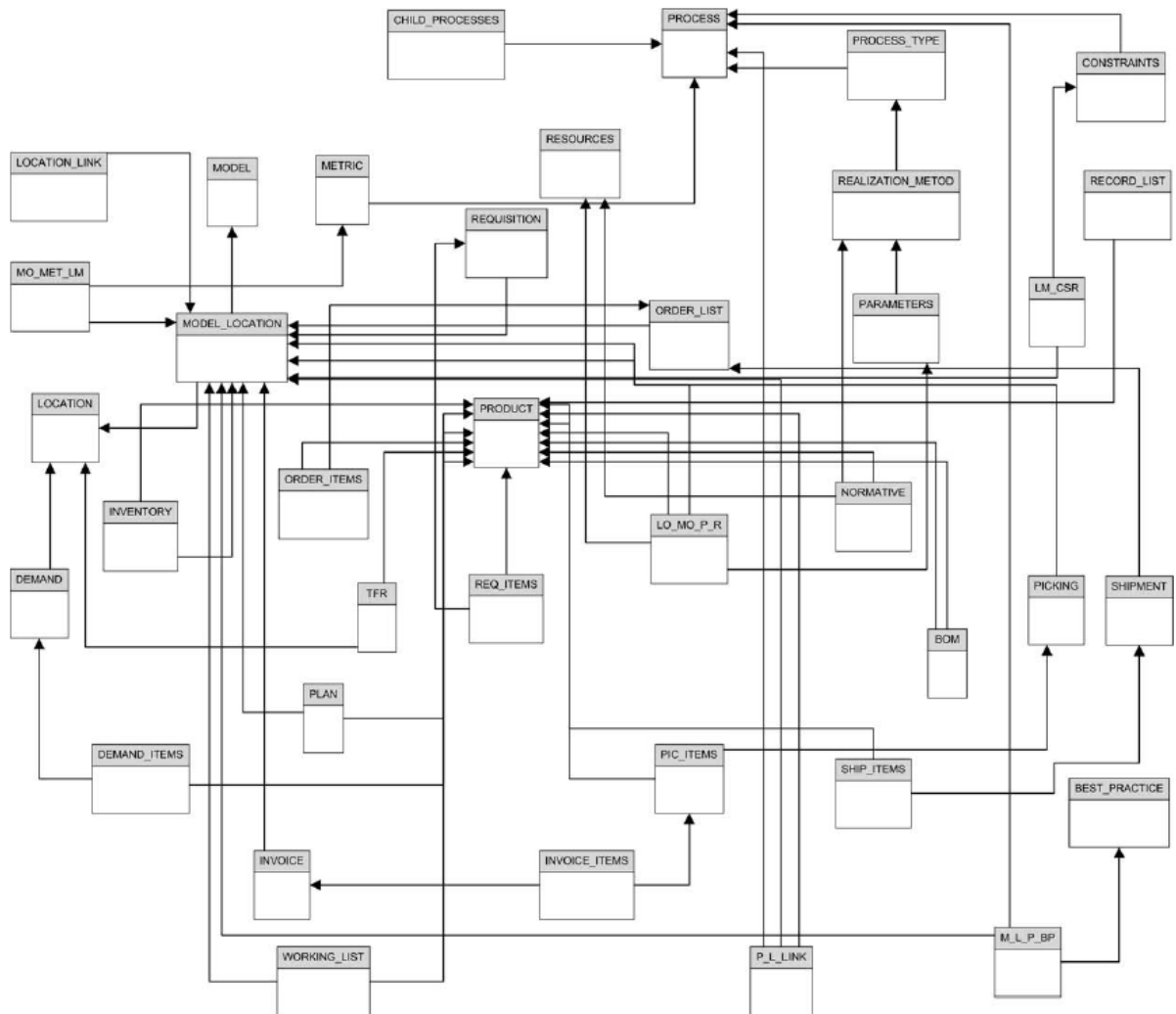
- υπομοντέλα δομής
- υπομοντέλα διαδικασία
- επιχειρηματικό περιβάλλον
- περιορισμοί

Τα δύο πρώτα υπομοντέλα (υπομοντέλα δομής και υπομοντέλα διαδικασία) είναι η βάση του μοντέλου του δικτύου παροχής. Τα άλλα δύο υπομοντέλα (επιχειρηματικό περιβάλλον και περιορισμοί) επιτρέπουν την ανάλυση της βασικής συμπεριφοράς μοντέλου υπό διαφορετικές ρεαλιστικές συνθήκες.

Αυτή η προσέγγιση επιτρέπει όχι μόνο ευρύ πειραματισμό, αλλά και το πλαίσιο για το σχεδιασμό νέων δικτύων παροχής ή την ανάλυση και επανασχεδιασμό του υφιστάμενου δικτύου παροχής διαμορφώσεις.



Εικόνα 21: Συνιστώσες του προγράμματος προσομοίωσης [32].



Εικόνα 22: Βασή δεδομένων του μοντέλου προσομοίωσης [32].

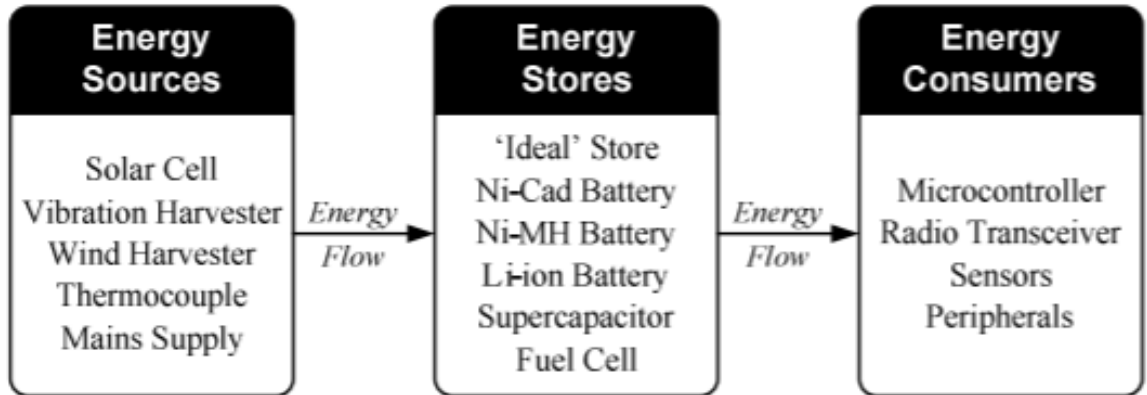
Συμπερασματικά, το λογισμικό προσομοίωσης που παρουσιάζεται στη παρούσα εργασία μπορεί να εφαρμοστεί για τα ακόλουθα πειράματα:

- Ανάλυση ευαισθησίας (sensitivity analysis), δηλαδή προσδιορισμός της επιρροής που έχει μία ή περισσότερες παράμετροι και πως οι διαφορετικές παράμετροι επηρεάζουν το παγκόσμιο δίκτυο παροχής.
- Ανάλυση σεναρίων (scenario analysis), σύγκριση και αξιολόγηση εναλλακτικών δικτύων παροχής.
- Βελτιστοποίηση (Optimization), δηλαδή εύρεση του βέλτιστου συνδυασμού παραμέτρων εισόδου που δίνουν την επιθυμητή έξοδο (π.χ. μεγιστοποίηση εισόδων).

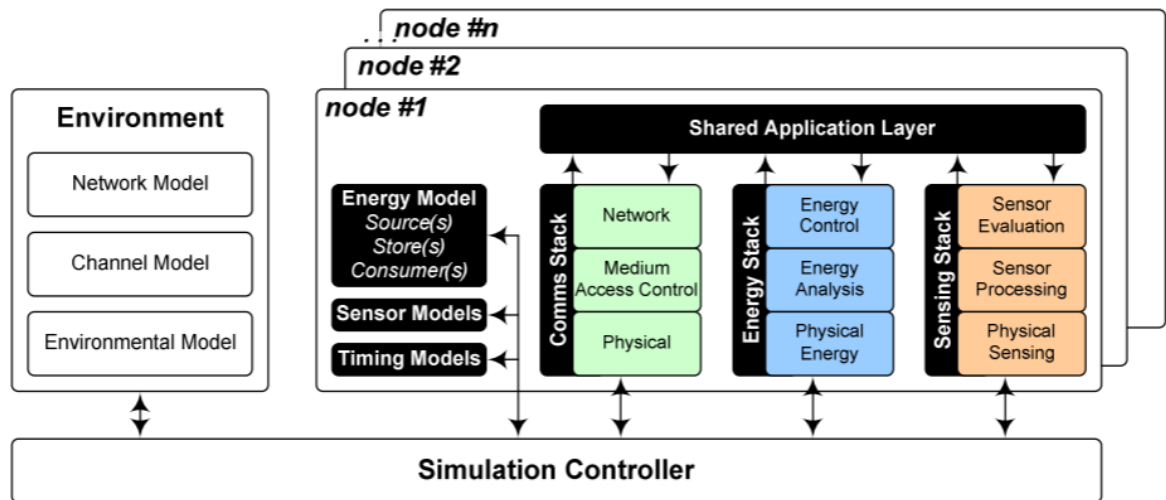
Κατά τη διάρκεια της τελευταίας δεκαετίας περίπου, πολλές έρευνες έχουν γίνει χρησιμοποιώντας προσομοιωτές “πανταχού παρών αισθητήρων” δικτύων (Ubiquitous Sensor Networks - USN), επιδεικνύοντας τη χρησιμότητα και τη σημασία αυτών των εργαλείων στις εφαρμογές USN.

Κατά συνέπεια, ένα σημαντικό μέρος των ερευνών έχει συγκεκριμένα και γενικά επισκοπήσει, συγκρίνει, και αξιολογήσει διάφορες πτυχές της προσομοίωσης USN..

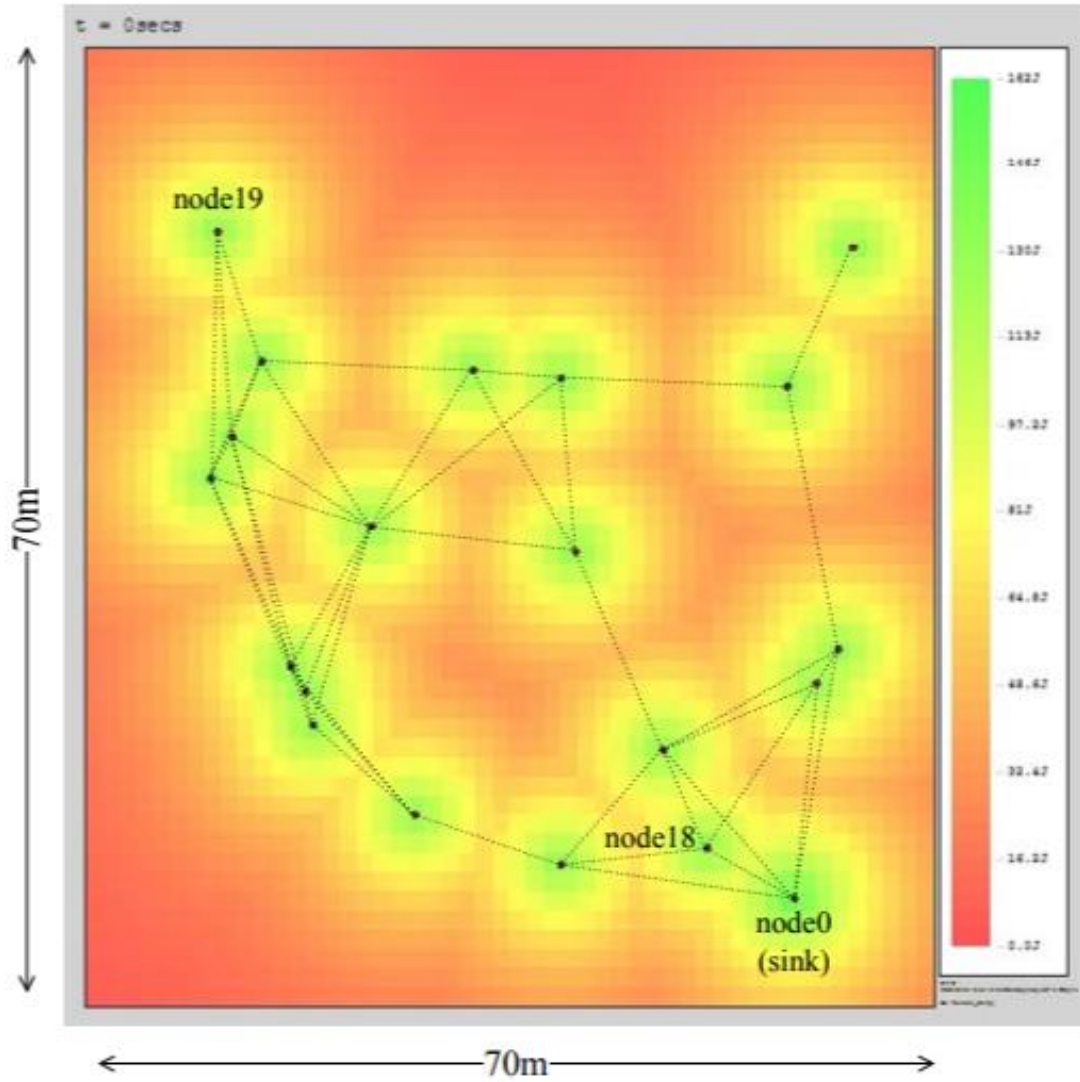
Από τη συγκεκριμένη άποψη, στην εργασία [33] διερευνήθηκε η ενεργειακή ευαισθησία ενός αριθμού προσομοιωτών USN και στη συνέχεια προτάθηκε μια νέα δομή για την προσομοίωση των ενεργειακά ευαίσθητων USN.



Εικόνα 23 Ενεργειακές συνιστώσες (components) ενός τυπικού ασύρματου κόμβου αισθητήρα.



Εικόνα 24: Η προτεινόμενη ενεργειακά ενημερούμενη δομή για προσομοίωση ασύρματου δικτύου αισθητήρων.



Εικόνα 25: Ένα υπο-προσομοίωση δίκτυο με χρήση του προγράμματος WSNsim

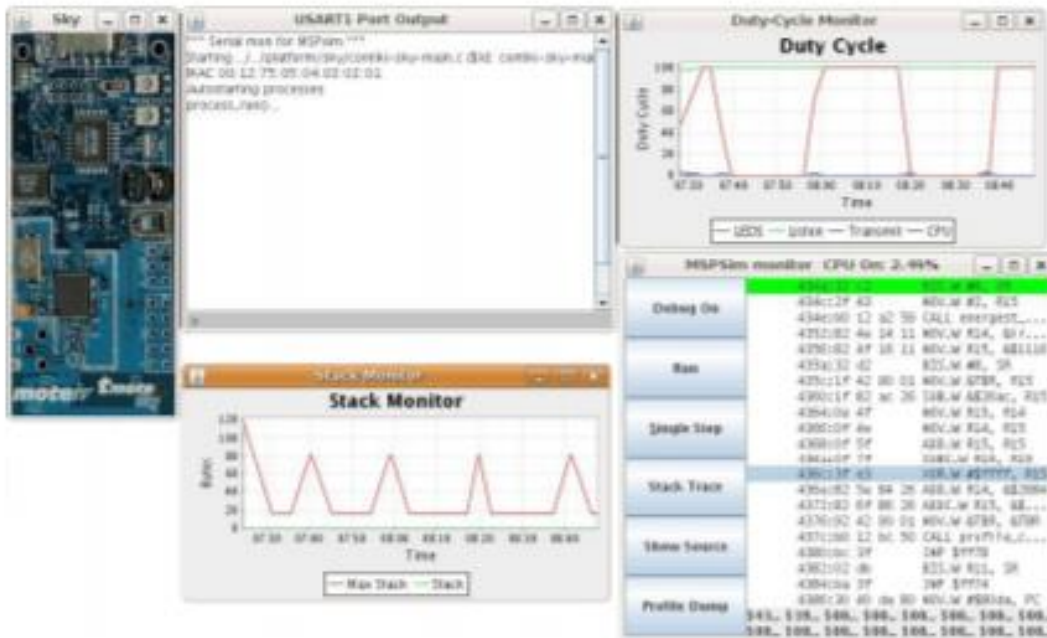
Στην εργασία [34], εισάχθηκαν και αξιολογήθηκαν χαρακτηριστικά κάλυψης και η συνδεσιμότητας δημοφιλών εργαλείων προσομοίωσης USN. Επίσης, αναλύθηκε η συσχέτιση της κάλυψης με την συνδεσιμότητα.

Επίσης, στην εργασία [35] παρείχαν υπόβαθρο σε διάφορα εργαλεία προσομοίωσης μαζί με τα πλεονεκτήματα και τα μειονεκτήματα του καθενός. Ως εκ τούτου, προτάθηκε μια μεθοδολογία αξιολόγησης προκειμένου να αξιολογηθούν οι δυνατότητες κάθε εργαλείου προσομοίωσης. Αν και η σημασία των εξειδικευμένων αυτών ερευνών είναι αδιαμφισβήτητη, τα αποτελέσματα δεν μπορεί να επεκταθεί σε όλα τα χαρακτηριστικά αυτού του διακεκριμένου εργαλείου. Σε άλλες λέξεις, οι εφαρμογές USN συνήθως αποτελούνται από ένα σύνολο από τα στάδια και τις υλοποιήσεις που θα έπρεπε ένας προσομοιωτής να είναι σε θέση να χειριστεί. Η ισχύς ενός προσομοιωτή σε συγκεκριμένο χαρακτηριστικό δεν εγγυάται ότι και άλλα χαρακτηριστικά λειτουργούν καλά. Επομένως, πρέπει να αξιολογηθούν διάφορες γενικές πτυχές προσομοιωτών παράλληλα. Από γενική άποψη, η πλειοψηφία της δημοσιευμένης ερευνών έχει ερευνήσει περιβάλλον προσομοίωσης και εξομοίωσης USN είτε με ποσοτικούς ή ποιοτικούς τρόπους, και σπάνια ένας συνδυασμός αυτών των δύο μπορεί να βρεθεί στη βιβλιογραφία. Με τις ποσοτικές μελέτες, οι περισσότερες έρευνες έχουν αναθεωρήσει ένα μεγάλο αριθμό προσομοιωτών σε επιφανειακά επίπεδα με την παροχή σύντομες περιγραφές και γενικές ανασκοπήσεις των εργαλείων μέχρι στιγμής.

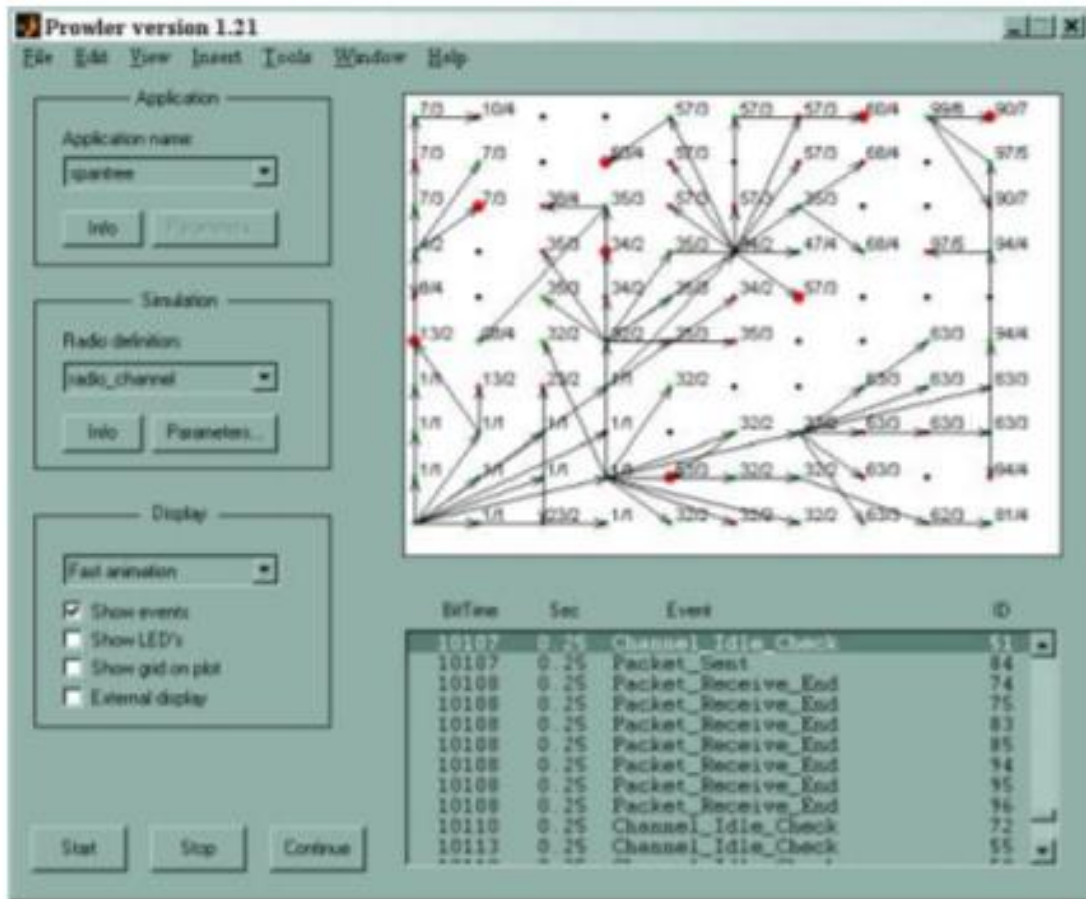
Για την για παράδειγμα, στην εργασία [36] παρουσιάστηκαν 63 προσομοιωτές, 14 εξομοιωτές, 19 εργαλεία οπτικοποίησης δεδομένων, 46 testbeds, 26 εργαλεία

εντοπισμού σφαλμάτων, 10 εργαλεία ενημέρωσης κώδικα και 8 οθόνες δικτύου στο USN.

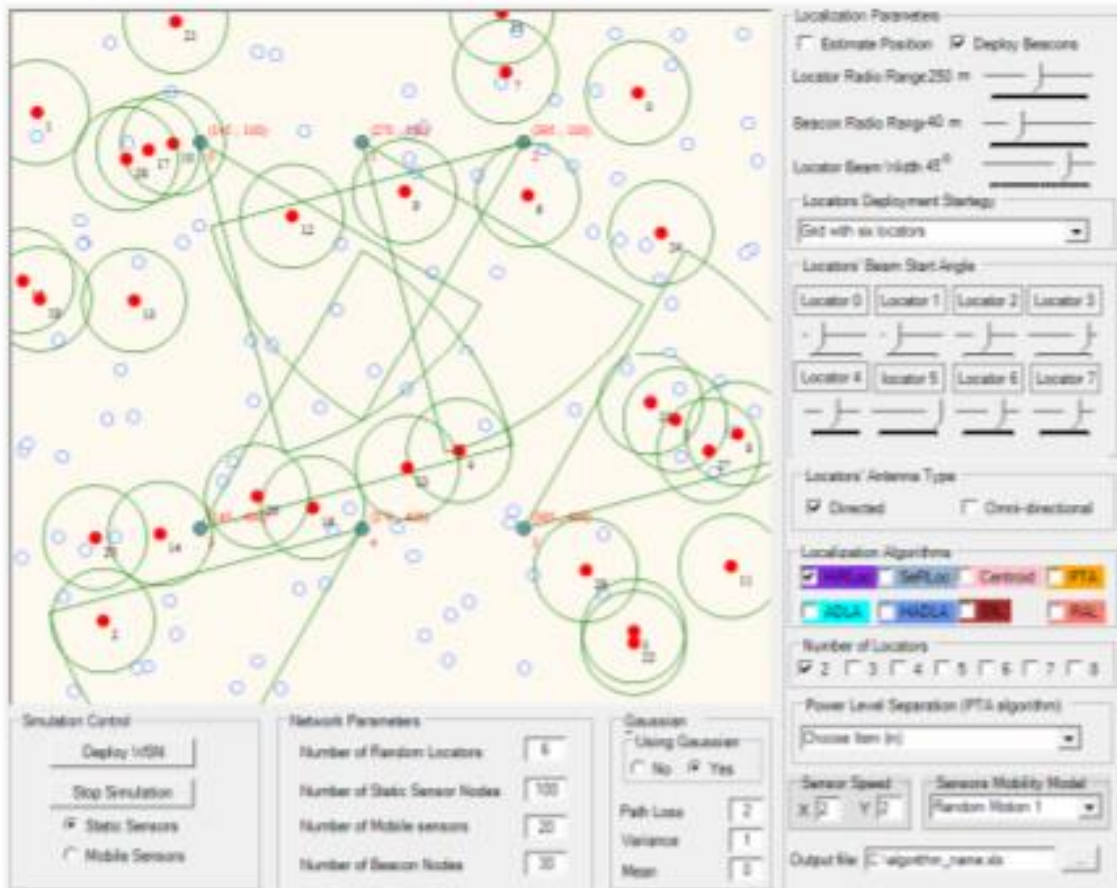
Σύγχρονα και βασικά χαρακτηριστικά παρουσιάστηκαν στην εργασία [37] καθώς και τα στιγμιότυπα GUI από τα 35 περιβάλλοντα προσομοίωσης που χρησιμοποιούνται ευρέως στην έρευνα και ανάπτυξη.



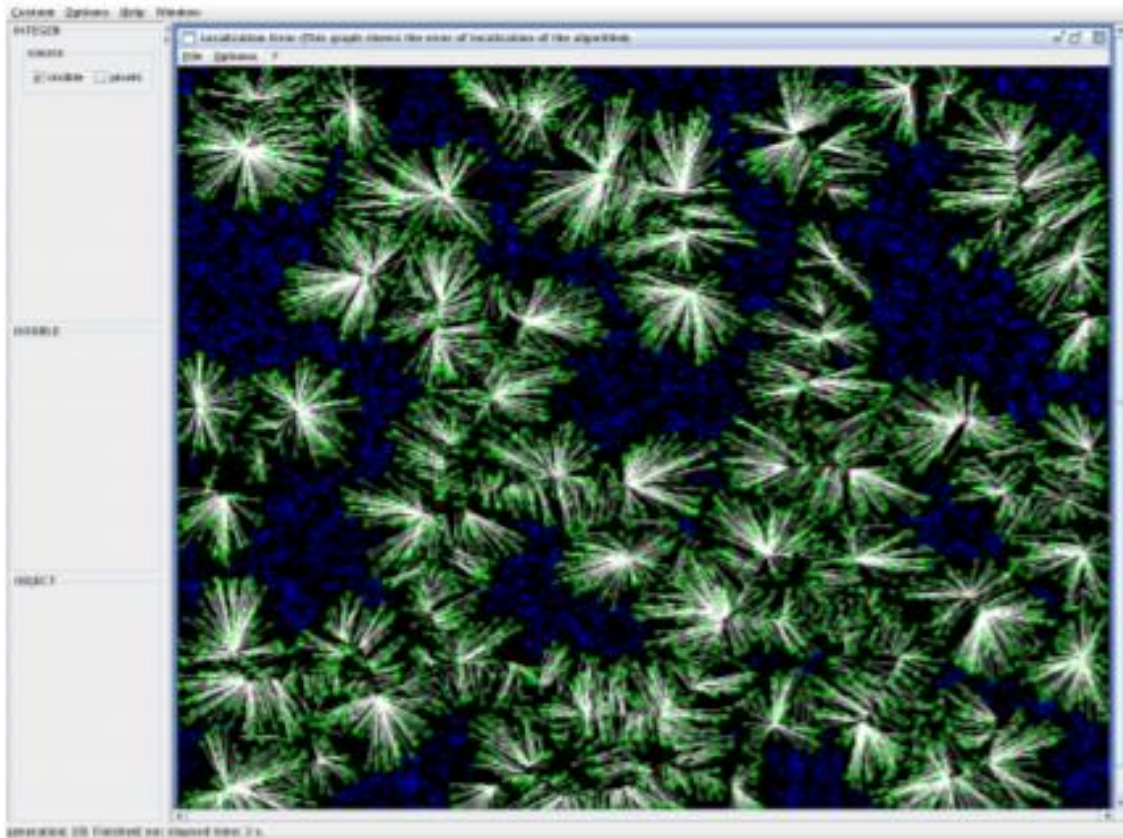
Εικόνα 26: Ο προσομοιωτής MSPsim



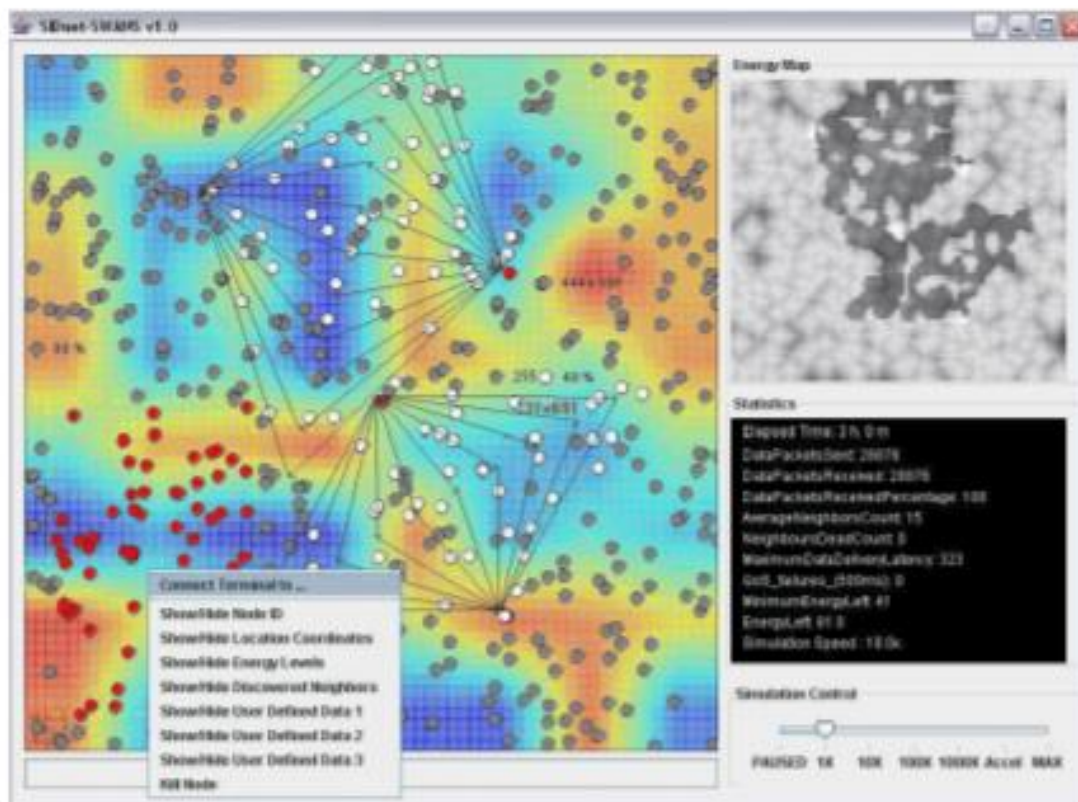
Εικόνα 27: Ο προσομοιωτής Prowler



Εικόνα 28: Προσομοιωτής Wireless Sensor Network Localization Simulator



Εικόνα 29: Ο προσομοιωτής AlgoSenSim



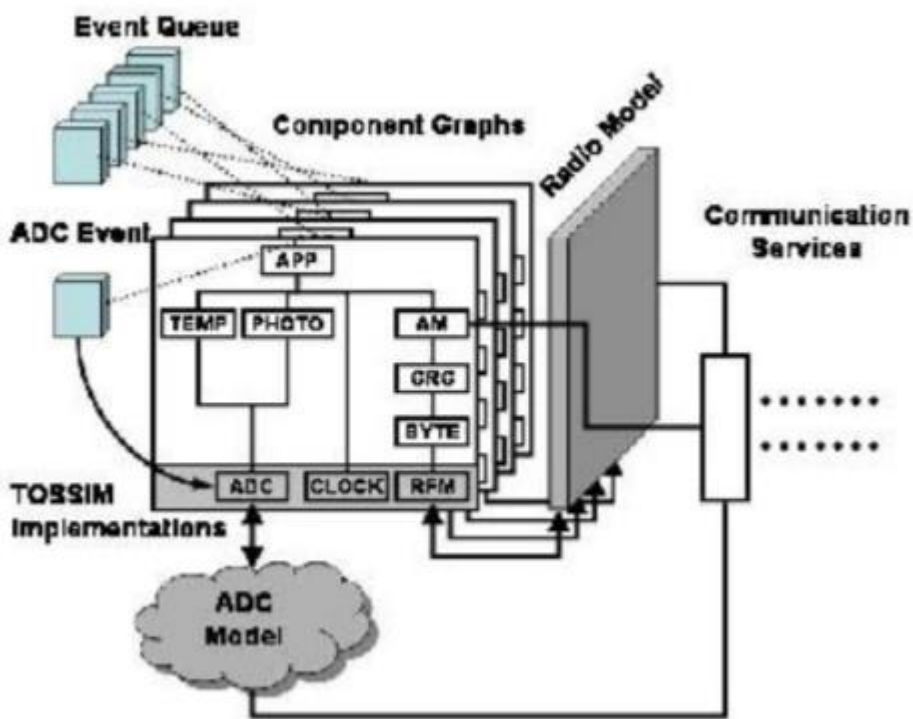
Εικόνα 30: Ο προσομοιωτής SIDnet-SWANS

Απαριθμούνται 74 USN προσομοιωτές και εξομοιωτές που συνοδεύουν τα χαρακτηριστικά και τις ιδιότητές τους στην εργασία [38].

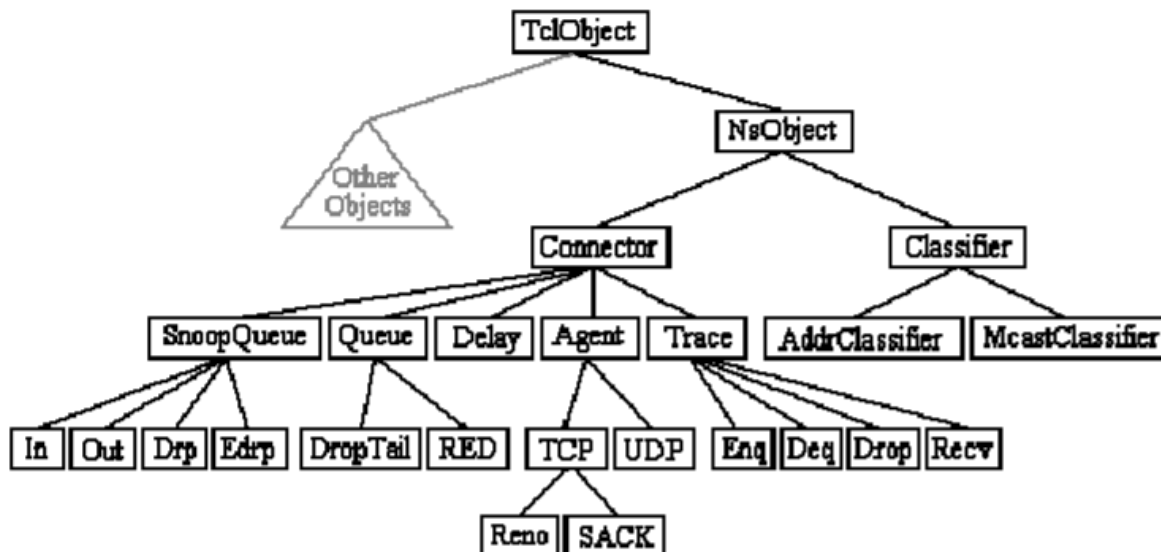
Παρόλο που τέτοιες μελέτες αντιμετωπίζονται ως άρθρα επισκόπησης, κανένα από αυτά δεν συγκρίνει και αξιολογεί σε βάθος το προσομοίωση ή τα εργαλεία εξομοίωσης. Με τις ποιοτικές μελέτες, μελετήθηκαν μερικές έρευνες με έναν

περιορισμένο αριθμό προσομοιωτών παρέχοντας λεπτομερείς λεπτομέρειες και ειδικά χαρακτηριστικά αυτών των εργαλείων.

Για παράδειγμα, ο Karl στην εργασία [39], συνέκρινε εις βάθος τον προσομοιωτή NS-2 και τον TOSSIM εξομοιωτή από άποψη μοντέλων, το εργαλείο απεικόνισης, την αρχιτεκτονική και τα εξαρτήματα.



Εικόνα 31: Η αρχιτεκτονική του προσομοιωτή TOSSIM

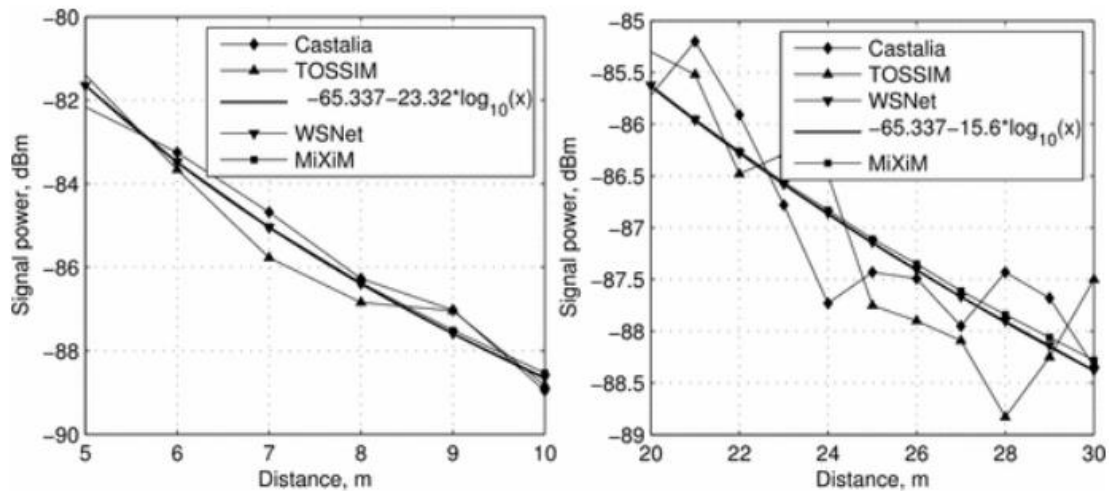


Εικόνα 32: Αρχιτεκτονική του NS.

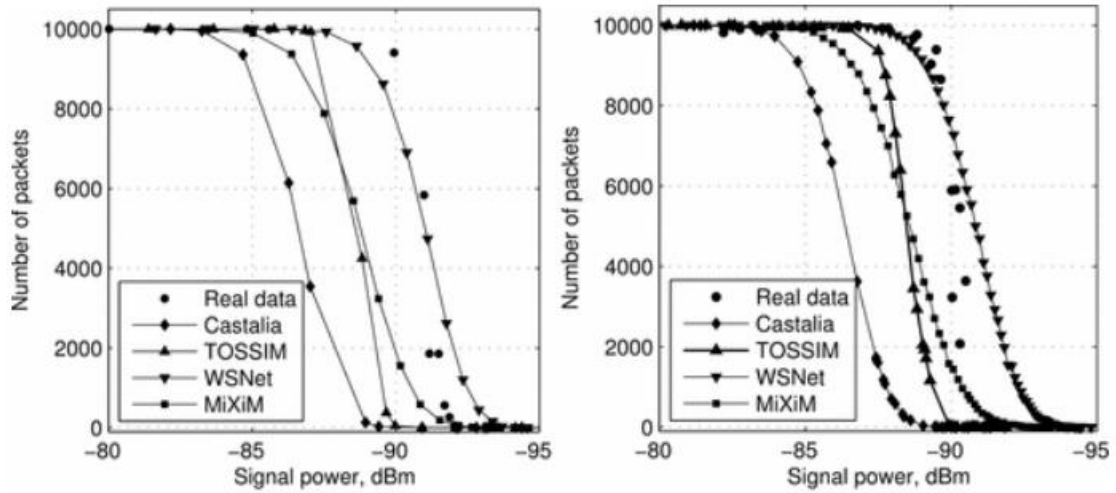
Στην εργασία [40], αξιολογήθηκε η διεπαφή, η προσβασιμότητα και η υποστήριξη των χρηστών, η διαθεσιμότητα δομικών στοιχείων (structural elements) USN, και επεκτασιμότητα επτά (δηλαδή, NS-2, OMNeT ++, GloMoSim, OPNET, SENSE, TOSSIM, GTSNetS) περιβάλλοντα προσομοίωσης και εξομοίωσης.

Ο Stetsko στην εργασία [41], προσπάθησε να εξετάσει την ρύθμιση κεραίας, διάδοση ραδιοφώνου, το θόρυβο, τον έλεγχο του μέσου πρόσβασης, την τοπολογία και το μοντέλο κατανάλωσης ενέργειας σε τέσσερα (δηλ. Castalia, MiXiM, TOSSIM, WSNnet) προσομοιωτές και εξομοιωτές USN. Είναι προφανές από τις μελέτες, όπως αυτές που προαναφέρθηκαν, ότι τα ιδιαίτερα χαρακτηριστικά του περιορισμένου αριθμού εργαλείων έχει εκτιμηθεί κανονικά και δεν υπάρχει ότι δύο μελέτες αξιολογούν τα ίδια χαρακτηριστικά ενός προσομοιωτή. Για

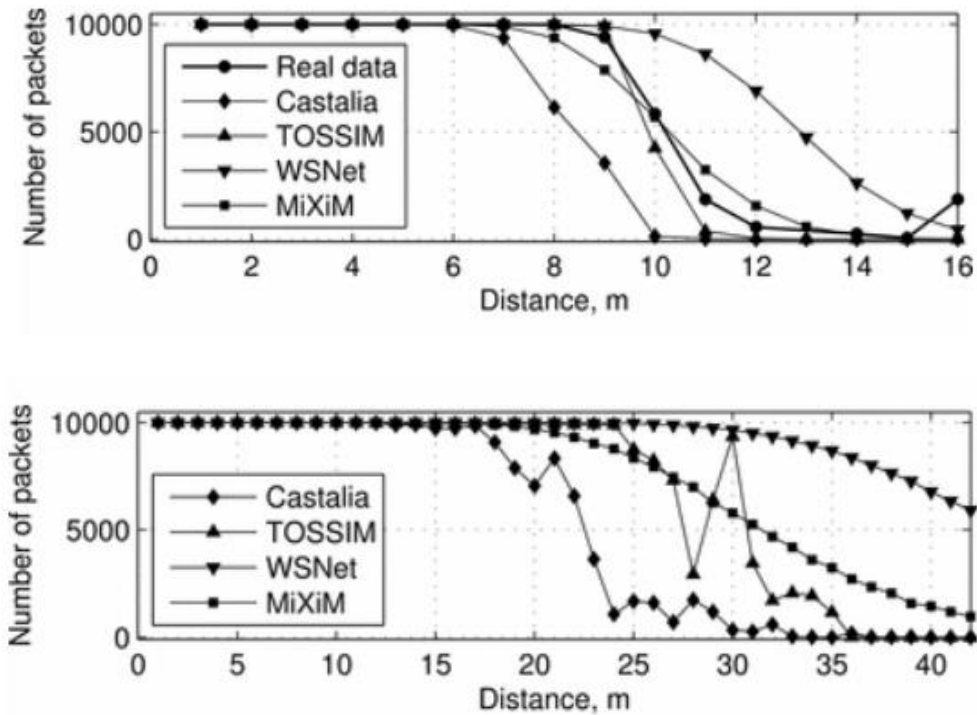
παράδειγμα, τόσο ο [39] όσο και ο Khan [40] μελετήθηκε ο NS-2, αν και με διαφορετικά κριτήρια. Khan [40] και Stetsko [41] μελετήθηκε το TOSSIM με σχεδόν ξεχωριστές ιδιότητες.

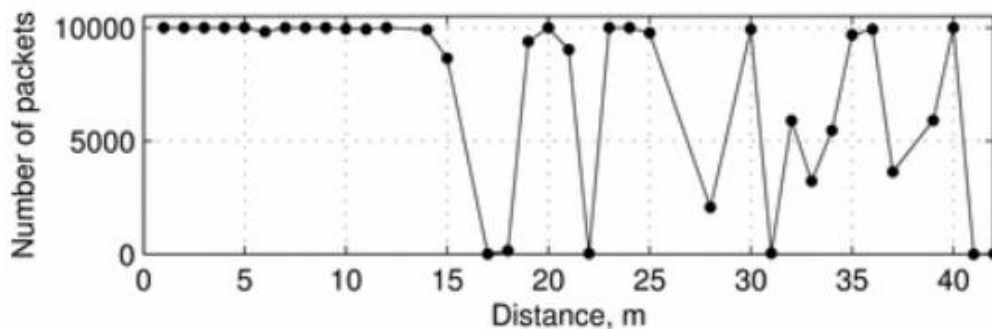


Εικόνα 33: Ισχύς μεταξύ μέσης ισχύος σήματος και απόστασης



Εικόνα 34: Σχέση μεταξύ αριθμού ληφθέντων πακέτων και ισχύος σήματος.





(c) Indoor, real measurements

Εικόνα 35: Σχέση μεταξύ ληφθέντων πακέτων και απόστασης.

Ο Πίνακας 4 παρουσιάζει μια χρονολογική επισκόπηση τόσο των ποσοτικών όσο και των ποσοτικών ποιοτικές μελέτες κατά την τελευταία δεκαετία. Αυτό το καλοκαίρι διεξάγεται από επανεξέταση των διαθέσιμων στο κοινό και των δημοσιευμένων εγγράφων, συμπεριλαμβανομένου άρθρα περιοδικών, κεφάλαια βιβλίων, πρακτικά συνεδρίων, διατριβές και τεχνικές αναφορές. Μεταξύ των προηγούμενων εισηγήσεων σχετικά με το αξιολόγηση προσομοιωτών και εξομοιωτών για το USN, κανένας από αυτούς δεν έχει επικεντρώθηκε βαθιά σε όλα τα υπάρχοντα αναπτυγμένα / εκτεταμένα εργαλεία μέχρι σήμερα, πολύ λίγοι από αυτούς έχουν εκτεταμένα αξιολογήσει τα εξέχοντα εργαλεία με βάση διάφορα κριτήρια και όχι τη διαρθρωμένη ταξινόμηση που έχει προταθεί για αυτά τα

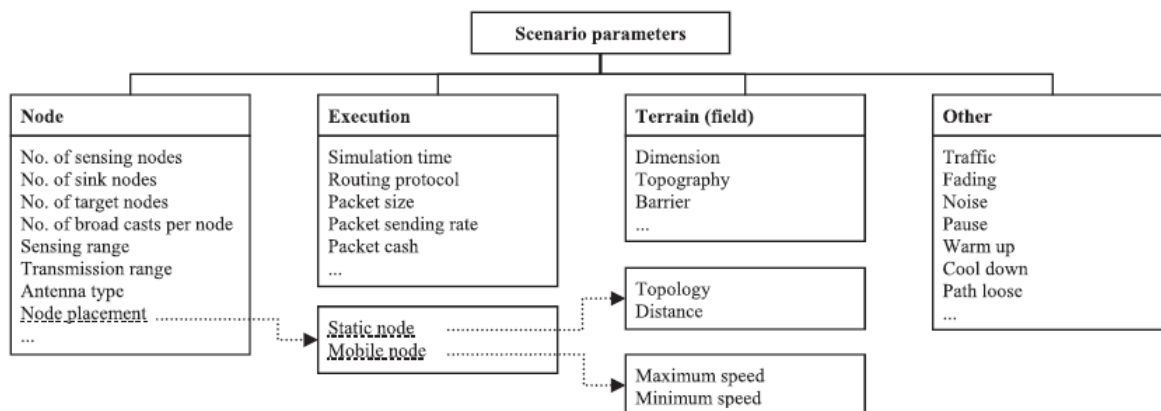
εργαλεία στην βιβλιογραφία. Επιπλέον, εξ όσων γνωρίζουμε, δεν υπάρχει έρευνα που έχει μελετήσει τις παραμέτρους αξιολόγησης της απόδοσης προσομοιωτών και εξομοιωτών σε σενάρια USN. Επιπλέον, μερικά νέα εργαλεία έχουν κυκλοφορήσει πρόσφατα και μερικά από τα γνωστά παραδοσιακά προσομοιωτές έχουν αναπτυχθεί από τα τελευταία χρόνια πρέπει να εξεταστούν. Ως εκ τούτου, υπάρχει μια πρωταρχική ανάγκη να τα γεμίσουμε κενά σε ένα νέο άρθρο έρευνας. Αυτή η έρευνα είναι διαφορετική από τις υπάρχουσες αναθεωρήσεις σε τρία σημαντικά πτυχές. Πρώτον, αυτή η έρευνα διευρύνει την έρευνά της σε όλα τα (θεμελιωμένα) περιβάλλοντα προσομοίωσης και εξομοίωσης USN μαζί με τα παραγόμενα παράγωγά τους και τις επεκτάσεις τους μέχρι σήμερα.

Reference	Year	Simulators and/or Emulators	Description
(Khemapech et al., 2005)	2005	NS-2, SENSE, GloMoSim, SENS, SensorSim, ATEMU, OMNeT++, Prowler, J-Sim, Shawn, TOSSIM, OPNET, TOSSF	Properties
(Egea-Lopez, 2005)	2005	NS-2, OMNeT++, J-Sim, NCTUns2.0, JIST/SWANS, GloMoSim, SSFNet, Ptolemy II, TOSSIM, EmStar/EmSim/EmTOS, SENS, ATEMU, Prowler/JProwler, SNAP	Overview and implementation issues
(Karl, 2005)	2005	NS-2, TOSSIM	Models, visualization, architecture, components
(Egea-Lopez, 2006)	2006	SSF, SWANS, J-Sim, NCTUns2.0, NS-2, OMNeT++, Ptolemy, SNAP, ATEMU, EmStar, TOSSIM	Models, type of visualization
(Neves et al., 2007)	2007	NS-2, J-Sim, SENSE	USN application in medicine, overview, and comparison
(Mekni and Moulin, 2008)	2008	NS-2, OPNET, OMNeT++, J-Sim, NCTUns, JIST/SWANS, GloMoSim, SSFNet, TOSSIM, TOSSF, TYTHON, EmStar/EmSim/EmTOS, ATEMU, SENSE, SENS, Prowler/JProwler, ModelNet/Nisnet, SwarmNet/Shawn, Glonemo, Avrora	Evaluation in terms of reusability and extensibility, performance and scalability, operating system portability, semantics scripting languages, realism level of virtual environment, graphics, and debug and trace
(Singh et al., 2008)	2008	NS-2, GloMoSim, OPNET, SensorSim, J-Sim, SENSE, OMNeT++, Sidh, SENS, TOSSIM, ATEMU	Overview
(Köksal, 2008)	2008	J-Sim, OMNeT++, NS-2, OPNET	Comprehensive overview, features, and comparison
(Lessmann, 2008)	2008	J-Sim, OMNeT++, NS-2, ShoX	Overview, installation, implementation and documentation, and visualization and statistics
(Kuonlehto et al., 2008)	2008	WISENES, SensorSim, sQualNet, NRL simulator, SWAN, SENSIM, EYES, J-Sim, VisualSense, Prowler, H-MAS, SENSE, TOSSIM, ATEMU, SENS, TOSSF, Em* EmSim, SNAP	Comparison table
(Wei)	2009	NS-2, SensorSim, J-Sim, SENS, TOSSIM, ATEMU, Avrora, EmStar, COOJA	Overview and comparison
(Mehta, 2009)	2009	NS-2, GloMoSim, J-Sim, OMNeT++, OPNET, QualNet	Comparison table
(Korkalainen, 2009)	2009	NS-2, OMNeT++, Prowler, TOSSIM, OPNET	Overview and comparison table
(Jevtić et al., 2009)	2009	NS-2, Castalia, TOSSIM, COOJA/MSPSim	Overview and comparison table
(Madani et al., 2010)	2010	NS-2, OMNeT++, NesCT, PAWIS, GloMoSim, OPNET, SENSE, Ptolemy II, J-Sim, CeI-DEVS, GTNetS, SystemC, NCTUns2.0, JIST/SWANS, SSFNet	Overview
(Kellner et al., 2010)	2010	GloMoSim/QualNet, OPNET, TOSSIM, OMNeT++ (Mobility Framework, MIXM, Castalia, INET Framework, NesCT), NS-2 (SensorSim), Avrora, J-Sim, ATEMU, EmStar, SENS, SENSE, Shawn	Overview and comparison table
(Imran et al., 2010)	2010	SensorSim, Nsrksensorsim, Castalia, VisualSense, Viptos, Sidh, Prowler/JProwler, SENS, TOSSIM, ATEMU, Avrora, SENSE, EmStar	Overview
(Khan, 2011)	2011	NS-2, OMNeT++, GloMoSim, OPNET, SENSE, TOSSIM, GTSNetS	Overview, comparison table, and performance analysis (CPU time and network lifetime)
(Moravek et al., 2011)	2011	NS-2, SensorSim, NRL, OMNeT++, SenSim, Castalia, MixIM, PAWIS, J-Sim, SENSE	Overview
(Stehlik, 2011)	2011	NS-2, OMNeT++ (MIXIM), Worldsens (WSim and WSNet), TOSSIM, COOJA, OPNET, J-Sim, TRMSim-WSN, WSNim	Comprehensive overview and comparison table
(Sundani, 2011)	2011	NS-2, TOSSIM, GloMoSim, UWSim, Avrora, SENS, COOJA, Castalia, Shawn, EmStar, SENSE, VisualSense, JProwler	Overview, comparison table, and performance analysis (CPU time and memory usage)
(Stetsko et al., 2011)	2011	Castalia, MIXIM, TOSSIM, WSN	Focusing on topology, antenna, radio propagation, noise, radio, medium access control and energy consumption modeling
(Dwivedi and Vyas, 2011)	2011	<i>Simulators:</i> Network Simulator (NS-2 and NS-3), Mannasim, TOSSIM, TOSSF, PowerTOSSIM Z, ATEMU, COOJA, GloMoSim, QualNet, SENSE, VisualSense, AlgoSenSim, GTNetS, OMNeT++, Castalia, J-Sim, JIST/SWANS, JIST/SWANS++, Avrora, Sidh, Prowler, JProwler, LeccSim, OPNET, SENS, EmStar, EmTOS, SenQ, SIDnet-SWANS, SensorSim, Shawn, SSFNet, Ataraya, NetTopo, WiseNet, SimGate, SimSync, SNetSim, SensorMaker, TRMSim-WSN, PAWIS, OLIMPO, DiSenS, WISDOM, Sinalgo, Sensoria, Capricorn, H-MAS, Starsim, Motesim, SNSim, SNIPER-WSNim, SNAP, SimPy, Mule, CaVi, Ptolemy, Maple, WISENES, WSN-Worldsens, WSim, LSU Sensor Simulator, WSNGE, TikTak. <i>Emulators:</i> VMNET, ATEMU, EmStar, TOSSIM, AvroraZ/Avrora, Freemote, EmPro, NetTopo, OCTAVEX, SENSE, UbiSec & Sens, Emuli, MSPSim, MEADOWS	Short description
(Paul, 2012)	2012	ATEMU, Avrora, Castalia, JProwler, SENSE	Short description
(Kumar, 2012)	2012	GloMoSim/QualNet, OMNeT++, NS-2, OPNET, J-Sim	Overview and comparison
(Mishra, 2012)	2012	NS-2, OMNeT++, J-Sim, GloMoSim, SSFNet, EmStar/EmSim/EmTOS	Overview
(Abuaraqoub, 2012)	2012	SensorSim, TOSSIM, TOSSF, GloMoSim, QualNet, OPNET, EmStar, SENS, J-Sim, Dingo, NS-3, Shawn	Overview and comparison table

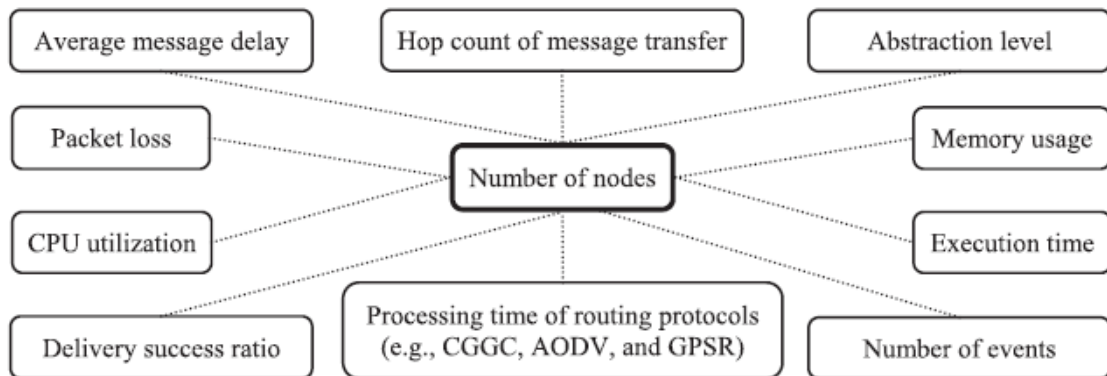
Reference	Year	Simulators and/or Emulators	Description
(Ali, 2012)	2012	Mannasim, NRL SensorSim, RTNS, OMNeT++, Castalia, MiXIM, NesCT, PAWIS, SENSIM (SensorSimulator), Ptolemy II, Viptos, VisualSense	Overview and comparison table, performance analysis of USN in MATLAB
(Cheour, 2013)	2013	NS-2, TOSSIM, OMNeT++, J-Sim, ATEMU, Avrora, SENSE, SensorSim	Overview
(Mieyeville, 2012)	2013	NS-2, NS-3, OMNeT++, TOSSIM and its derivatives, Avrora, Worldsens, WISENES, IDEA1	Overview
(Chhimwal et al., 2013)	2013	NS-2, NS-3, TOSSIM, J-Sim, Castalia, QualNet	Overview, merits and demerits
(Leelavathi, 2013)	2013	NS-2, J-Sim, OPNET, OMNeT++, GloMoSim, Ptolemy II, JIST/SWANS, NCTUns2.0, SSFNet, TrueTime toolbox (MATLAB), TOSSIM, PowerTOSSIM, TOSEF & TYTHON, EmStar/EmSim/EmCee, ATEMU, Avrora, Prowler/JProwler, UWSim, Shawn, COOJA/MSPSim	Comparison table and simulator analysis
(Gupta, 2013)	2013	NS-2, NS-3, OMNeT++, J-Sim	Overview, architecture, advantages and limitations, and comparison table
(Chand et al., 2013)	2013	NS-3, OPNET, GloMoSim, MiXIM, Castalia, J-Sim, Avrora	Overview
(Sethi et al., 2012)	2013	NS-2, TOSSIM, GloMoSim, UWSim, J-Sim, SENSE, COOJA, SENSE, VisualSense, JProwler, Shawn, Castalia	Comparison table
(Chandrasekaran et al., 2013)	2013	NS-2, TOSSIM, GloMoSim, QualNet, OPNET, J-Sim, OMNeT++	Overview, architecture, merits and limitations, and comparison table
(Khalifa, 2013)	2013	NS-2, NS-3, OMNeT++, Wireshark (Ethereal), OPNET, GloMoSim/QualNet, J-Sim, GNS3	Comparison table
(Bhatt and Kathiriya, 2013)	2013	J-Sim, OMNeT++, NS-2, OPNET	Overview, GUI, comparison table of simulation features
(ZVKOVIC, 2014)	2014	NS-2, NS-3, GNS3, Wireshark (Ethereal), OPNET, OMNeT++, GloMoSim/QualNet, J-Sim, JIST/SWANS, VisualSense, Ptolemy II, TOSSIM, Castalia, EmStar, ATEMU, SENSE, SENS, JProwler, Avrora, COOJA, Shawn	Overview, comparison tables, advantages and disadvantages
(Sahin and Ammari, 2014)	2014	NS-2, OMNeT++, J-Sim, OPNET, TOSSIM	Comprehensive overview, features, components, comparison tables, and shortcomings
(Dhviya and Arthi, 2014)	2014	NS-2, EmStar, GloMoSim, Shawn, UWSim, VisualSense, J-Sim, OMNeT++, Aqua-Sim, QualNet	Underwater USN overview, merits and demerits, and comparison table
(Dwivedi and Vyas, 2014)	2014	Simulators: Network Simulator (NS-2 and NS-3), Mannasim, TOSSIM, TOSEF, PowerTOSSIM Z, ATEMU, COOJA, GloMoSim, QualNet, SENSE, VisualSense, AlgoSenSim, GTNetS, OMNeT++, Castalia, J-Sim, JIST/SWANS, JIST/SWANS++, Avrora, Sidh, Prowler, JProwler, LeccSim, OPNET, SENS, EmStar, EmTOS, SenQ, H-MAS, SensorSim, Shawn, NetTopo, Atarraya, SSFNet, WiseNet, SimGate, SimSync, SNetSim, SensorMaker, TRMSim-WSN, PAWIS, OLIMPO, DiSenS, WISDOM, Sinalgo, Sensoria, Capricorn, SIDnet-SWANS, Starsim, SNSim, SNIPER-WSNim, SNAP, SimPy, Mule, CaVi, Ptolemy, Maple, WISENES, WSN-Worldsens and WSim, LSU Sensor Simulator, WSNGE, TIKTak, Motesim, Boris, SmartSim, WSNim, EnergySim, MOB-YOSSIM, AEON, Sensor Security Simulator (S3), Wireless Sensor Network Localization Simulator, Xen WSN Simulator, UWSim, Network in a box (NAB)	Overview, categorization of USN-specific simulators
(Roy and Jain, 2015)	2015	QualNet, NS-2, NS-3, OPNET modeler, Net Sim, OMNeT++, J-Sim	Overview
(Nayyar and Singh, 2015)	2015	NS-2, NS-3, OMNeT++, J-Sim, Mannasim, SensorSim, NRL SensorSim, NCTUns, SSFNet, GloMoSim, QualNet, sQualNet, OPNET, SENSE, DRMSim, NetSim, UWSim, VisualSense, Viptos, Ptolemy II, SENS, Shawn, SIDnet-SWANS, SIDH, NetTopo, WSim/Worldsens/WSNet, WSN Localization Simulator, Prowler, MATLAB, PiccSIM, LabVIEW	Overview, architecture, interface/GUI, and comparison table
(Minakov, 2016)	2016	NS-2, NS-3, Castalia, MiXIM, PAWIS, WSN, DANSE, NetTopo, PASES, Sense, TOSSIM, Avrora, COOJA/MSPSim, VIPTOS	Overview, categorization of simulators, comparative study
(Fahmy, 2016)	2016	NS-2, NS-3, GloMoSim, OPNET, OMNeT++, TOSSIM, ATEMU, Avrora, EmStar, SensorSim, NRL SensorSim, J-Sim, Prowler/JProwler, SENS, SENSE, Shawn, SenSim, PAWIS, MSPsim, Castalia, MiXIM, NesCT, SUNSHINE	Overview, component, structure

Πίνακας 1: Βιβλιογραφία πάνω στο αντικείμενο της προσομοίωσης δικτύων.

Το άρθρο [42] εστιάζεται όχι μόνο στην ποσοτική πλευρά των προσομοιωτικών και εξομοιωτικών περιβαλλόντων, αλλά είναι αξιολογεί επίσης ποιοτικά αυτά που έχουν χρησιμοποιηθεί ευρέως, τακτικά ενημερωθεί και υποστηριχθεί καλά από τους προγραμματιστές τους βασισμένους σε πολυάριθμα κριτήρια. Προτείνει επίσης μια κατηγοριοποίηση για αυτά τα εργαλεία γενικά και ειδικού σκοπού. Δεύτερον, η έρευνα αυτή παρέχει μια γενική εικόνα σχετικά με τα πλέον σύγχρονα κριτήρια αξιολόγησης τόσο για τους προσομοιωτές όσο και για τους εξομοιωτές. Αυτό καθορίζει κατά συνέπεια ποιο εξέχον εργαλείο είναι επαρκές για το είδος του σκοπού (δηλαδή ακαδημαϊκό, ερευνητικό, ή εμπορικός). Τρίτον, αυτή η έρευνα προτείνει μια σειρά επιδόσεων παραμέτρους αξιολόγησης για προσομοιωτές και εξομοιωτές σε πανταχού παρόν προσομοίωσης και εξομοίωσης.



Εικόνα 36: Παράμετροι σεναρίου για την προσομοίωση USN δικτύου.



Εικόνα 37: Εκτίμηση αποδοτικότητας του μοντέλου προσομοίωσης μελετώντας τα χαρακτηριστικά κατά ζεύγη αναφορικά με τον αριθμό των κόμβων

6. Συμπεράσματα – Μελλοντικές εργασίες

Όπως είναι αντιληπτό, η προσομοίωση δικτύων είναι ένα αντικείμενο εξέχουσας σημασίας για επιστήμονες και μηχανικούς. Με χρήση προσομοίωσης γίνεται αναπαράσταση της λειτουργίας του δικτύου. Συνεπώς, η λειτουργία του μπορεί να ελεγχθεί δοκιμάζοντας διαφορετικές τιμές παραμέτρων και έτσι, με βάση τα συμπεράσματα που θα προκύψουν, μπορεί να επιτευχθεί η βελτιστοποίηση του. Το αντικείμενο της παρούσας εργασίας ήταν η βιβλιογραφική ανασκόπηση στο αντικείμενο της προσομοίωσης δικτύων, η ανάλυση μεθόδων και διαφόρων εργαλείων. Έγινε παρουσίαση φημισμένων εργαλείων προσομοίωσης δικτύων και σχολιάστηκαν αναφορικά με την ευχέρεια τους στις διάφορες περιπτώσεις δικτύων. Παρουσιάστηκαν, επίσης, εργασίες σχετικά με την επιστήμη της προσομοίωσης δικτύων. Στις εργασίες αυτές έγινε χρήση και αξιολόγηση εμπορικών προγραμμάτων προσομοίωσης ή ακόμα και ανάπτυξη νέων μοντέλων προσομοίωσης, απαραίτητα για την επίλυση συγκεκριμένων προβλημάτων.

Τέλος, σαν μελλοντική εργασία, θα ήταν καλό να διεξαχθεί έρευνα στην προσομοίωση δικτύων, όπου ο φοιτητής-ερευνητής, να καλεστεί να χρησιμοποιήσει ένα πρόγραμμα ή ακόμα και να αναπτύξει ο ίδιος ένα υπολογιστικό μοντέλο προσομοίωσης. Αρχικά, θα επιλέξει ένα σύγχρονο πρόβλημα δικτύου προς επίλυση, και ύστερα θα αναπτύξει, σύμφωνα με τις γνώσεις που έχει αποκομίσει στο αντικείμενο, το δικό του μοντέλο προσομοίωσης.

Έτσι, ως αποτέλεσμα, εκτός από το αντικείμενο της προσομοίωσης δικτύων, ο μαθητής θα αποκτήσει εξοικείωση και με τα αντίστοιχα μαθηματικά και προγραμματισμό.

Αναφορές

- [1] Μαδεμλής Ι., Σημειώσεις για το μάθημα «Δίκτυα Υπολογιστών», ΤΕΙ Σερρών, ανάκτηση στις 15/04/2014 από <http://www.teicm.gr/ICD/staff/mademlis/networks I 1.pdf>
- [2] Αλεξόπουλος Α., Λαγογιάννης Γ., Τηλεπικοινωνίες και δίκτυα υπολογιστών, Αθήνα 2003, 6^η έκδοση
- [3] Τεχνολογία δικτύων επικοινωνιών, ΥΠΕΠΘ, Αθήνα
- [4] Law A.M., Kelton W.D. (2000). *Simulation Modelling and Analysis, 3rd edition*, New York: McGraw Hill.
- [5] Kleijnen J.P.C., Van Groenendaal W. (1992). *Simulation. A Statistical Perspective*, Chichester, UK: Wiley.
- [6] Law, A.M. (2007). *Simulation Modeling and Analysis (4th ed.)*. New York: McGraw-Hill
- [7] Pidd, M. (2005). *Computer Simulation in Management Science, 5th ed.* Chichester, UK: Wiley
- [8] L. Bresla, et.al. (2000), Advances in network simulation, The VINT Project, IEEE

- [9] K. Fall, K. Varadhan (2003), The ns manual, The VINT Project, http://www.isi.edu/nsnam/ns/doc/ns_doc.pdf
- [10] Network Simulators, by Sally Floyd. Available from: <http://www.icir.org/models/simulators.html>
- [11] Network Simulation (Wikipedia). Available from: http://en.wikipedia.org/wiki/Network_simulation
- [12] ns-3. (2015). *ns-3 (Version 3.24.1)* [Software], <https://www.nsnam.org/releases/latest>
- [13] Python. (2013). *Python (Version 3.5.0)* [Software], <https://www.python.org/downloads/>
- [14] Mercurial. (2015). Mercurial (Version 3.5.2) [Software], <https://www.mercurial-scm.org/>
- [15] Eclipse. (2015). Eclipse IDE [Software], <https://eclipse.org/>
- [16] OMNeT++. (2015). *OMNeT++ (Version 4.6)* [Software], <https://omnetpp.org/omnetpp>
- [17] J.M. Pullen (2000), The network workbench: network simulation software for academic investigation of internet concepts, Computer Networks. Elsevier Science Ltd
- [18] Network Simulation (Wikipedia). Available from: http://en.wikipedia.org/wiki/Network_simulation
- [19] B.L. Barnett III (1993), An ethernet simulator for undergraduate networking, ACM SIGCSE Bulletin, in: Proceedings of the 24th SIGCSE Technical Symposium on

Computer Science Education, vol. 25(1), Indianapolis, IN, USA, , pp. 145–150.

[20] D. Griffin (2000), Traffic engineering for quality of services in the internet, at large-scale, Technical Report, CEC Deliverable Number: 201/UCL/bl, TEQUILA Consortium.

[21] C. Alaettinoglu, I. Dussa-Zieger, A.U. Matta, Shankar, O. Gudmundsson, Introducing MaRS, a Routing Testbed, ACM SIGCOMM, pp. 95–96.

[22] S.P. Ahjua, Comnet III: A Network Simulation Laboratory Environment For A Course In Communications Networks, Frontiers in Education Conference (IEEE/ASEE conference), Tempe, AZ, 1998. Available from: <http://fie.engrng.pitt.edu/fie98/papers/1205.pdf>

[23] CACI Products Company, COMNET III User’s Manual, January 1995.

[24] J.G. Goble, R. Mills (1994), COMNET III: object-oriented network performance prediction, in: Proceedings of the 1994 Winter Simulation Conference 443–445.

[25] J.H. Cowie, D.M. Nicol, A.T. Ogieliski, Modeling the global internet, Computing in Science and Engineering 1 (1) (1999) 42–50.

[26] Luigi Rizza, Dummynet: a simple approach to the evaluation of network protocols, ACM SIGCOMM, Computer Communication Review 27 (1) (1997) 31–41.

[27] Luigi Rizzo, Dummynet Tool (Accessed 12.9.2004).

[28] Y-K Lin, L Fiondella, P-C Chang (2013), Simulation approach to estimate the system reliability of a time-based capacitated flow network susceptible to correlated failures

[29] L.R. Ford, D.R. Fulkerson, Flows in Networks, Princeton University Press, New Jersey, 1962.

[30] L. Fiondella, Reliability and sensitivity analysis of coherent systems with negatively correlated component failures, International Journal of Reliability, Quality and Safety Engineering 17 (2010) 505–529.

[31] M.Y. Ata (2007), A convergence criterion for the Monte Carlo estimates, Simulation Modelling Practice and Theory 15 237–246.

[32] D. Stefanovic, N. Stefanovic, B. Radenkovic (2009), Supply network modelling and simulation methodology

[33] Merrett G.V., et al., 2009. Energy-Aware Simulation for Wireless Sensor Networks, Proceedings of the 6th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks.

[34] Zhu C., et al. , A survey on coverage and connectivity issues in wireless sensor networks.

[35] Mekni M., B. Moulin (2008). A survey on sensor webs simulation tools, Proceedings of the Second International Conference on Sensor Technologies and Applications, SENSORCOMM'08, IEEE.

[36] Dwivedi A., Vyas O. (2007), An exploratory study of experimental tools for wireless sensor networks. Wirel. Sens. Netw. 3 (07), 215–240.

[37] Musznicki, B., Zwierzykowski, P., Survey of simulators for wireless sensor networks. *Int. J. Grid Distrib. Comput.* 5 (3), 23–50.

[38] Dwivedi, A., Vyas, O., Recent developments in simulation tools for WSNs: an analytical study. In: Monowar, M.M., Khan, S., Pathan, A.-S.K. (Eds.), *Simulation Technologies in Networking and Communications*. CRC Press, Boca Raton, FL, 495–518.

[39] Karl M. (2005). A comparison of the architecture of network simulators ns-2 and tossim, *Proceedings of Performance Simulation of Algorithms and Protocols Seminar*.

University of Stuttgart.

[40] Khan M.Z., et al., (2011). Limitations of simulation tools for large-scale wireless sensor networks. In: *Proceedings of the IEEE Workshops of International Conference on Advanced Information Networking and Applications (WAINA)*. IEEE

[41] Stetsko, A., M. Stehlik, V. Matyas, (2011). Calibrating and comparing simulators for wireless sensor networks. In: *Proceedings of the 8th International Conference on Mobile Adhoc and Sensor Systems (MASS)*. IEEE.

[42] M. Sharif, A. Sadeghi-Niarakia, *Ubiquitous sensor network simulation and emulation environments: A survey*

[43] R. G. Addie, M. Zukerman, T. Neame. Broadband traffic modeling: Simple solutions to hard problems. *IEEE Communications Magazine*, (8):88–95, 1998. 219

[44] D. Anick, D. Mitra, M. Sondhi. Stochastic theory of a data handling system with multiple sources. 'The Bell System Technical Journal, 61:1871—1894, 1982. 219

[45] M. Crovella, A. Bestavros. Self-similarity in World Wide Web traffic: Evidence and possible causes. IEEE/ACM Transactions on Networking, 5(6):835-846, 1997. 219

[46] Erramilli, O. Narayan, W. Willinger. Experimental queueing analysis with long-range dependent packet- traffic. IEEE/ACM Transactions. 4(2):209-223,1996.

[47] IEEE Journal on Selected Areas in Communication, 4:856-868, 1986. 219