

**ΕΛΛΗΝΙΚΟ ΑΝΟΙΚΤΟ ΠΑΝΕΠΙΣΤΗΜΙΟ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ**

**ΜΕΤΑΠΤΥΧΙΑΚΗ ΕΞΕΙΔΙΚΕΥΣΗ
ΣΤΑ ΠΛΗΡΟΦΟΡΙΑΚΑ ΣΥΣΤΗΜΑΤΑ**

Διπλωματική Εργασία

**Μελέτη Επιδημικών Αλγορίθμων σε Περιβάλλοντα
Διάχυτου Υπολογισμού**

**(Τμήμα 1 – Μελέτη χωρικής / χρονικής εγκυρότητας πληροφορίας πλαισίου μέσω επιδημικών
αλγορίθμων)**

Κωνσταντίνος Α. Λυμπέρης

Επιβλέπων: Ευστάθιος Χατζηευθυμιάδης, Επίκουρος Καθηγητής ΕΚΠΑ

Πάτρα
Μάρτιος 2008





Στους αγαπημένους μου



Δηλώσεις

A. Το κείμενο και τα λοιπά αποτελέσματα της παρούσας διατριβής αποτελούν συνιδιοκτησία του ΕΑΠ και του φοιτητή, ο καθένας από τους οποίους έχει το δικαίωμα ανεξάρτητης χρήσης και αναπαραγωγής τους (στο σύνολο ή τμηματικά) για διδακτικούς και ερευνητικούς σκοπούς, σε κάθε περίπτωση αναφέροντας τον τίτλο και το συγγραφέα και το ΕΑΠ, όπου εκπονήθηκε η Διπλωματική Εργασία, καθώς και τον επιβλέποντα και την επιτροπή κρίσης.

© ΕΑΠ, 2008

B. Δηλώνω υπεύθυνα πως το κείμενο της παρούσας διατριβής, η οποία εκπονήθηκε στα πλαίσια της ΘΕ «Διπλωματική Εργασία» του προγράμματος «Μεταπτυχιακή Εξειδίκευση στα Πληροφοριακά Συστήματα» (ΠΛΣ), και τα λοιπά αποτελέσματα της Διπλωματικής Εργασίας (ΔΕ) είναι προϊόν προσωπικής μελέτης και εργασίας και πως όλες οι πηγές που χρησιμοποιήθηκαν αναφέρονται είτε μέσα στο κείμενο είτε στις βιβλιογραφικές και διαδικτυακές παραπομπές.

Κώστας Λυμπέρης



Μελέτη Επιδημικών Αλγορίθμων σε Περιβάλλοντα Διάχυτου Υπολογισμού

(Τμήμα 1–Μελέτη χωρικής / χρονικής εγκυρότητας πληροφορίας πλαισίου μέσω επιδημικών αλγορίθμων)

Κωνσταντίνος Α Λυμπέρης

Στάθης Χατζηευθυμιάδης

Αρχιλέας Καμέας

Βασίλειος Πλαγιανάκος



Περίληψη

Η παρούσα εργασία ασχολείται με τη μελέτη της συμπεριφοράς του επιδημικού αλγορίθμου SIS (Υγιής-Μολυσμένος-Υγιής) εξετάζοντας τις παραμέτρους διάδοσης μιας πληροφορίας (ιός) στην περίπτωση ενός δικτύου 100 κόμβων:

- i. όταν αυτοί είναι ακίνητοι και
- ii. όταν αυτοί κινούνται.

Για την περίπτωση του δικτύου κινούμενων κόμβων επιλέγεται το μοντέλο κινητικότητας Random Waypoint (RWP) και υλοποιείται ένας προσομοιωτής 100 κόμβων στην πλατφόρμα προσομοίωσης J-sim για 1000 χρονικές στιγμές.

Γίνεται κωδικοποίηση του επιδημικού αλγορίθμου σε Java και εκτελείται το πρόγραμμα για διάφορες τιμές των παραμέτρων

- i. κινητικότητα κόμβων (m)
- ii. ακτίνα επικοινωνίας κόμβων (r) και
- iii. το επιδημικό όριο (λ),

κατά τη φάση μιας επιδημικής εξάπλωσης και κατά τη φάση μιας επιδημικής εξασθένησης.

Συνολικά εξετάζονται οχτώ διαφορετικά σενάρια.

Το συμπέρασμα ότι η κινητικότητα των κόμβων είναι ο σημαντικότερος παράγοντας για τη διάδοση μιας επιδημίας σε πολύ υψηλά ποσοστά του πληθυσμού των κόμβων είναι αυτό που «επιδεικτικά» αναδεικνύεται σαν πρώτο. Και η ακτίνα επικοινωνίας (πόσους γείτονες έχει ο κάθε κόμβος) παίζει σημαντικό ρόλο κατά τη φάση της επιδημικής εξάπλωσης. Ενώ το επιδημικό όριο φαίνεται να συμμετέχει με σαφώς μικρότερη βαρύτητα στη διάδοση της επιδημίας.

Κατά τη διαδικασία της επιδημικής εξασθένησης καμία από τις εξεταζόμενες παραμέτρους δε προκύπτει ότι παίζει κάποιο ουσιώδες ρόλο στο χρόνο πλήρους εξυγίανσης του πληθυσμού των κόμβων ή στο να εμποδίσει την πλήρη εξυγίανση. Στη φάση της επιδημικής εξασθένησης τα συμπεράσματα είναι παρόμοια και για την περίπτωση του δικτύου ακίνητων κόμβων και για την περίπτωση του δικτύου των κινούμενων κόμβων.

Η εξυγίανση κάθε κόμβου εξαρτάται μόνο από την παράμετρο του ρυθμού ίασης δ και δεν επηρεάζεται ουσιαστικά από τις υπόλοιπες εξεταζόμενες παραμέτρους.

Θα μπορούσαμε να πούμε ότι η εργασία αυτή επιβεβαιώνει ένα γνωστό από την κοινωνιολογία και τη βιολογία συμπέρασμα: Η κοινωνικοποίηση (κινητικότητα) των ατόμων (κόμβων) μπορεί να εγγυηθεί, με όσο μεγάλη πιθανότητα επιτυχίας επιθυμούμε, τη διάδοση μιας πληροφορίας (ιού), πλήρως, σε ένα τμήμα πληθυσμού.

Λέξεις-κλειδιά: Επιδημικός Αλγόριθμος, Διάδοση πληροφορίας, Προσομοίωση Δικτύων, SIS epidemical model, ad hoc δίκτυα, RWP model, j-sim, Java.



***Epidemical Algorithms for Contextual Information
Dissemination in Pervasive Computing
Environments***

(Part 1—Study of spatial/temporal validity of Contextual information via Epidemical Algorithms)

Kostas A Lyberis

Stathes P. Hadjiefthymiades

Ahhileas Kameas

Vassilis Plagianakos



Abstract

The present work research the behaviour of epidemical algorithm SIS (Susceptible - Infected - Susceptible) examining the parameters of information (virus) dissemination in the case of network of 100 nodes:

- i. when these are motionless and
- ii. when these are moved.

For the case of network of moved nodes is selected the mobility model Random Waypoint (RWP) and is materialized a simulator of 100 nodes in the simulation platform J-sim for 1000 time moments.

Becomes coding of epidemical algorithm in Java and is executed the program for various prices of parameters

- i. mobility of nodes (m)
- ii. communication radius of nodes (r) and
- iii. the epidemical threshold (λ),

at the phase of epidemic Prevalence and at the phase of epidemic Decay.

Globally are examined eight different Scenarios.

The conclusion that the mobility of nodes is the more important factor on the dissemination of epidemic in very high percentages of population of nodes is “flamboyantly” rise to view at first. And the communication radius (how much neighbours has the each node) plays important role at the phase of epidemic Prevalence. While the epidemical threshold appears to participate with, of course, smaller gravity in the epidemic dissemination.

At the process of epidemic Decay no one of the examined parameters results that it plays some essential role in per year complete cleansing of population of nodes or it prevents the complete cleansing. And in the case of network of motionless nodes and in the case of network of moved nodes the conclusions are similar for the phase of epidemic Decay.

The cleansing of each node depends only from the parameter of cure rate (δ) and is not influenced substantially by the remainder-examined parameters.

We can say that this work confirms a acquaintance from the sociology and the biology conclusion: The socialization (mobility) of the individuals (nodes) can guarantee, with any probability of success, we wish, the dissemination of information (virus), completely, in a part of population.

Key-words: Epidemical Algorithm, Information Dissemination, network’s simulation, SIS epidemical model, ad hoc network, RWP model, j-sim , Java.



Ευχαριστίες

Ξεξινώντας τη συγγραφή αυτής της εργασίας θα ήθελα να ευχαριστήσω τον επιβλέποντα, κ. Στάθη Χατζηευθυμιάδη, επίκουρο καθηγητή του τμήματος Πληροφορικής στο ΕΚΠΑ, για τη βοήθεια, την καθοδήγηση και την κριτική του κατά την εκπόνηση της εργασίας.

Επίσης ευχαριστώ τον υποψήφιο διδάκτορα στο τμήμα Πληροφορικής του ΕΚΠΑ κ Χρίστο Αναγνωστόπουλο για τη χρήσιμες υποδείξεις του και τη συνεχή και αγόγγυστη βοήθεια – παρακολούθηση κατά τη διάρκεια εκπόνησης της παρούσας εργασίας.

Στο σημείο αυτό οφείλω να αναφερθώ στο συμφοιτητή μου Παπαδόπουλο Νίκο με τον οποίο συνεργαστήκαμε αρμονικά κατά την εκπόνηση των διπλωματικών μας εργασιών.

Ξεχωριστά θέλω να ευχαριστήσω την αγαπημένη μου Μαρίνα, καθώς και τις κόρες μου Λένα και Νάσια, που μου συμπαραστάθηκαν με αγάπη κι υπομονή, κι έδειξαν ανοχή για το χρόνο που τους «έκλεψα», αυτά τα χρόνια, για να τον διαθέσω στην ολοκλήρωση του Μεταπτυχιακού Διπλώματος.

Τέλος, το μεγαλύτερο ευχαριστώ επιθυμώ βαθειά να απευθύνω στους γονείς μου, που με στήριξαν συναισθηματικά, υλικά και πρακτικά σε όλη μου την πορεία μέχρι τώρα. Ως ελάχιστη ένδειξη σεβασμού και αναγνώρισης για τη δική τους προσφορά και τις θυσίες που έχουν κάνει για τη δική μου μόρφωση, και όχι μόνο, αφιερώνω την εργασία αυτή στους γονείς μου Θανάση και Ελένη.

Κώστας Λυμπέρης



Περιεχόμενα

<i>Εξώφυλλο</i>	<i>i</i>
<i>Δηλώσεις</i>	<i>iv</i>
<i>Τίτλος – Κριτική Επιτροπή</i>	<i>v</i>
<i>Περίληψη</i>	<i>vi</i>
<i>Title - Crisis Committee</i>	<i>vii</i>
<i>Abstract</i>	<i>ix</i>
<i>Ευχαριστίες</i>	<i>ix</i>
<i>Περιεχόμενα</i>	<i>x</i>
1 Πρόλογος.....	1
2 Εισαγωγή στους Αλγορίθμους Μετάδοσης Πληροφορίας βάσει Επιδημικών Μοντέλων.....	3
2.1 Ορισμοί.....	5
2.2 Τύποι διάδοσης πληροφορίας.....	6
2.3 Κατηγορίες επιδημικών αλγορίθμων.....	7
2.4 Μοντέλα Επιδημικών Αλγορίθμων.....	8
2.4.1 Μοντέλο SI.....	8
2.4.2 Μοντέλο SIS.....	8
2.4.3 Μοντέλο SIR.....	9
2.5 Αλγόριθμοι διάδοσης Πληροφορίας βάσει Επιδημικών μοντέλων.....	9
2.5.1 Ο SI Αλγόριθμος.....	9
2.5.2 Ο SIS Αλγόριθμος.....	10
2.5.3 Ο SIR Αλγόριθμος.....	12
2.6 Το εξεταζόμενο SIS μοντέλο.....	13
3 Μοντέλα Κινητικότητας.....	14
3.1 Τύποι μοντέλων κινητικότητας.....	14
3.1.1 Μοντέλα τυχαίας κινητικότητας.....	15
3.1.2 Αιτιοκρατικά Μοντέλα.....	16
3.1.3 Υβριδικά μοντέλα.....	16
3.2 Μοντέλο Random Waypoint (RWP).....	17
3.2.1 Περιγραφή του μοντέλου.....	17
3.2.2 Ορισμοί για το RWP μοντέλο.....	18
3.2.3 Παράμετροι του RWP μοντέλου.....	19



4	Εφαρμογή των Επιδημικών Αλγορίθμων σε περιβάλλοντα κινητού υπολογισμού.	20
4.1	SIS Μοντέλα για ακίνητους κόμβους.	23
4.1.1	ΣΕΝΑΠΙΟ-1: (Επιδημική εξάπλωση συναρτήσει του r).....	23
4.1.2	ΣΕΝΑΠΙΟ -2: (Επιδημική εξάπλωση συναρτήσει του λ).....	25
4.1.3	ΣΕΝΑΠΙΟ -3: (Επιδημική εξασθένιση συναρτήσει του r).....	27
4.1.4	ΣΕΝΑΠΙΟ -4: (Επιδημική εξασθένιση συναρτήσει του λ).....	29
4.2	SIS Μοντέλα για κινούμενους κόμβους.	31
4.2.1	ΣΕΝΑΠΙΟ -5: (Επιδημική εξάπλωση συναρτήσει του r).....	31
4.2.2	ΣΕΝΑΠΙΟ -6: (Επιδημική εξάπλωση συναρτήσει του λ).....	33
4.2.3	ΣΕΝΑΠΙΟ -7: (Επιδημική εξασθένιση συναρτήσει του r).....	35
4.2.4	ΣΕΝΑΠΙΟ -8: (Επιδημική εξασθένιση συναρτήσει του λ).....	37
5	Συμπεράσματα.	39
5.1	Κατά την Επιδημική Εξάπλωση.	39
5.2	Κατά την Επιδημική Εξασθένιση.	39
6	Αναφορές.	40
7	Παράρτημα Α: Προσομοιωτής μοντέλου RWP για κίνηση 100 κόμβων.....	42
7.1	Εισαγωγή.	42
7.1.1	Autonomous Component Architecture (ACA).....	42
7.1.2	Script και J-sim.	43
7.1.3	Συστατικά της J-sim.....	44
7.1.4	J-sim και RWP.....	45
7.1.5	Εντολές στο αρχείο προσομοίωσης.	45
7.2	Προσομοιωτής (Κώδικας για κίνηση 100 κόμβων – Αρχείο prosomiotis_RWP_100_komvon).	49
7.3	Παρουσίαση του αρχείου sintetagmenes.txt	52
8	Παράρτημα Β: Ο αλγόριθμος SIS σε κώδικα Java.....	53
8.1	Ο Αλγόριθμος SIS για ακίνητους κόμβους (αρχείο fix_sis_epidemic.java). 53	
8.2	Ο Αλγόριθμος SIS για κινούμενους κόμβους (αρχείο mob_sis_epidemic.java).....	61
9	Παράρτημα Γ: Παρουσίαση των αρχείων υλοποίησης των μοντέλων της Διατριβής.	69
9.1	Προγραμματιστική φιλοσοφία για τα μοντέλα των ακίνητων κόμβων.	69
9.1.1	Παρουσίαση του αρχείου run_fix_sis_epidemic.bat.	70
9.2	Προγραμματιστική φιλοσοφία για τα μοντέλα των κινούμενων κόμβων ..	71
9.2.1	Παρουσίαση του αρχείου run_mob_sis_epidemic.bat.....	72



1 Πρόλογος.

Η μελέτη της διάδοσης πληροφοριών μέσω επιδημικών αλγορίθμων έχει μπει τα τελευταία χρόνια στο μικροσκόπιο της έρευνας και τροφοδοτείται έντονα λόγω της αποτελεσματικότητας αυτών των αλγορίθμων στα περιβάλλοντα δικτύων υπολογιστών ή κινητών τερματικών κόμβων, χωρίς κεντροποιημένη δομή, όπου το μοντέλο Server-Client σιγά-σιγά εκλείπει, ενώ τα Peer to Peer συστήματα επικρατούν όλο και περισσότερο.

Οι γνώσεις μας από το πεδίο της βιολογίας για τη μετάδοση μολυσματικών ασθενειών σε τμήματα πληθυσμών προσφέρουν την πρώτη ύλη για τη διαμόρφωση αλγορίθμων, που θα είναι ικανοί με αποτελεσματικό τρόπο να διαχέουν πληροφορίες σε μεγάλα σύνολα κόμβων, που λειτουργούν πλέον και σαν server και σαν clients, (κινούμενων ή ακίνητων).

Στην παρούσα εργασία, καταρχήν, μελετώνται, στο Κεφάλαιο-2, γενικά οι επιδημίες. Δίνονται κάποιιοι ορισμοί σχετικοί με τις επιδημίες και ορίζονται οι παράμετροι που υπεισέρχονται στους επιδημικούς αλγορίθμους. Αναφέρονται οι τύποι διάδοσης πληροφορίας (πλαισίου και κατάστασης) και οι κατηγορίες των επιδημικών αλγορίθμων (Push-based, Pull-based, Push&Pull-based).

Στη συνέχεια αναφέρονται κάποια μοντέλα (SI, SIS, SIR) και οι αλγόριθμοι για αυτά τα μοντέλα και εξηγείται ο αλγόριθμος που έχει επιλεγεί (SIS) να εξεταστεί στην παρούσα εργασία.

Στο Κεφάλαιο-3 γίνεται αναφορά στα μοντέλα κινητικότητας, παρουσιάζεται μια γενική κατηγοριοποίηση τους και εξετάζονται πιο συγκεκριμένα τα χαρακτηριστικά του μοντέλου Randon Waypoint (RWP) που έχει επιλεγεί για την παρούσα εργασία.

Το Κεφάλαιο-4 είναι το κεντρικό κεφάλαιο της εργασίας. Παρουσιάζονται τα αποτελέσματα της μελέτης που έγινε και εμφανίζεται σε μορφή γραφημάτων η συμπεριφορά του αλγορίθμου που εξετάστηκε για τις διάφορες τιμές των παραμέτρων (8 σενάρια).

Στο Κεφάλαιο-5 συνοψίζονται τα συμπεράσματα που προκύπτουν από τη μελέτη που παρουσιάστηκε στο Κεφάλαιο-4.

Στο Κεφάλαιο-6 υπάρχει ο πίνακας της βιβλιογραφίας και διαδικτυακών χώρων που χρησιμοποιήθηκαν σαν πηγές για την εργασία.

Τέλος στα παρατήματα παρατίθεται όλη η δουλειά υποδομής που πραγματοποιήθηκε, σε προγραμματιστικό επίπεδο, για να φτάσουμε στα αποτελέσματα του Κεφαλαίου-4.



Στο Παράρτημα Α γίνεται μια παρουσίαση της πλατφόρμας J-sim. Παρατίθεται ο προσομοιωτής που δημιουργήθηκε και το αρχείο συντεταγμένων που αυτός παράγει.

Στο Παράρτημα Β παρουσιάζονται τα αρχεία κωδικοποίησης, του αλγορίθμου που εξετάζουμε, σε γλώσσα προγραμματισμού Java, για ακίνητους και για κινούμενους κόμβους.

Στο παράρτημα Γ εξηγείται ο τρόπος που σχεδιάστηκε και υλοποιήθηκε, προγραμματιστικά, όλη η εργασία.



2 Εισαγωγή στους Αλγορίθμους Μετάδοσης Πληροφορίας βάσει Επιδημικών Μοντέλων

Η μετάδοση πληροφοριών μέσα από συστήματα υπολογιστών ή μέσω του διαδικτύου καθώς και μέσω δικτύων κινητών κόμβων αλλά και συνδυασμένων συστημάτων επικοινωνίας (διάχυτα περιβάλλοντα) αποτελεί ένα σημαντικό πεδίο επενδύσεων από κυβερνητικούς οργανισμούς (π.χ. ασφάλειας, στρατιωτικούς) και εταιρείες, εφαρμοσμένης έρευνας από εκπαιδευτικούς οργανισμούς και ιδιωτικές εταιρείες (π.χ. πολυτεχνικές σχολές, εταιρείες επικοινωνιών) και θεωρητικής μελέτης. Πολλά ερωτήματα παραμένουν ανοιχτά ενώ και νέα προκύπτουν και παραδίδονται στη βάση της μελέτης και της έρευνας. Σ' αυτό το πλαίσιο έχουν αναπτυχθεί και μελετηθεί διάφοροι τύποι αλγορίθμων, όπως οι ντετερμινιστικοί αλγόριθμοι πλημμύρας (flooding algorithms) και οι επιδημικοί αλγόριθμοι (epidemic algorithms).

Οι επιδημικοί αλγόριθμοι έχουν μεγάλη αποδοχή ως ένας σημαντικός και εξελικτικός τρόπος μετάδοσης πληροφοριών σε καταναμημένα συστήματα και ειδικά σε συστήματα μεγάλης κλίμακας. Οι πληροφορίες διαδίδονται σε ένα καταναμημένο σύστημα με τον ίδιο τρόπο που μια επιδημία θα διαδιδόταν σε μια ομάδα ατόμων: Κάθε διαδικασία που επιθυμεί να διαδώσει ένα νέο κομμάτι πληροφορίας στο σύστημα δεν το στέλνει σε έναν κεντρικό υπολογιστή ή σε μια συστάδα κεντρικών υπολογιστών, με σκοπό τη διάδοσή του. Προτιμά την αποστολή του σε ένα σύνολο παρόμοιων διαδικασιών, που επιλέγονται τυχαία. Στη συνέχεια, κάθε μία από τις διαδικασίες κάνει το ίδιο πράγμα, και ούτω καθ' εξής.

Οι επιδημικοί αλγόριθμοι έχουν μελετηθεί θεωρητικά και η ανάλυσή τους στηρίζεται στις καλές μαθηματικές υποδομές [10].

Η αρχή που κρύβεται πίσω από αυτού του τύπου τη διάδοση πληροφοριών μιμείται την τεχνική διάδοσης μιας επιδημίας [14], έτσι αναφερόμαστε σε επιδημική διάδοση πληροφορίας. Κατ' αναλογία βρίσκουμε τη διάδοση μιας φήμης μεταξύ ανθρώπων που κουτσομπολεύουν: τότε αναφερόμαστε σε διάδοση κουτσομπολιού.

Μόλις αρχίσουν, οι επιδημίες είναι δύσκολο να εκλείψουν. Μερικοί άνθρωποι που μολύνονται από μια μεταδοτική ασθένεια (επιδημία) είναι σε θέση να τη διαδώσουν, άμεσα ή έμμεσα, σε ένα μεγάλο πληθυσμό. Οι επιδημίες είναι ελαστικές στις αποτυχίες κατά τη διαδικασία της μόλυνσης. Δηλαδή, ακόμα κι αν πολλοί μολυσμένοι άνθρωποι έχουν πεθάνει πριν μεταδώσουν την ασθένεια, ή είναι ανοσοποιημένοι, η επιδημία είναι ακόμα ικανή να μεταδίδεται στους πληθυσμούς.

Οι επιδημικοί αλγόριθμοι σχετίζονται με την πιθανο-θεωρητική διάχυση της πληροφορίας σ' ένα ανάλογο δίκτυο. Η έννοια της πιθανο-θεωρητικής διάχυσης



πληροφορίας δηλώνει ότι ένας κόμβος i επιλέγει κάποιο γειτονικό του j με κάποια κατανομή πιθανότητας (μπορεί και τυχαία) και μεταδίδει (ή ανακτά) την πληροφορία σ' αυτόν (ή από αυτόν) (δηλ. ο κόμβος i μολύνει τον κόμβο j ή το αντίθετο). Η διαδικασία αυτή επαναλαμβάνεται μέχρις ότου όλοι οι κόμβοι (ή ένα ικανοποιητικό ποσοστό κόμβων) του δικτύου να έχουν μολυνθεί (δηλ. να έχουν λάβει την πληροφορία).

Η συμπεριφορά των επιδημικών αλγορίθμων παρουσιάζει ιδιαίτερο ενδιαφέρον. Συγκεκριμένα, η τοπολογία του δικτύου, η κινητικότητα των κόμβων μέσα σε καθορισμένο χώρο, καθώς και το μοντέλο κινητικότητας, το πλήθος των κόμβων η ακτίνα μετάδοσης, ο χρόνος ανακάλυψης (ο απαιτούμενος χρόνος για να εγκαθιδρυθεί επικοινωνία με έναν κόμβο που βρίσκεται μέσα στην ακτίνα επικοινωνίας), ο ρυθμός μόλυνσης β (ο ρυθμός μόλυνσης υγιών γειτονικών κόμβων από έναν μολυσμένο κόμβο), ο ρυθμός ίασης δ (ο ρυθμός ίασης ενός μολυσμένου κόμβου), η φύση της επιδημίας (στατικό πλαίσιο πληροφορίας ή δυναμικά και ισχυρά μεταβαλλόμενο πλαίσιο), η φύση της πληροφορίας που πρόκειται να προωθηθεί και ο τύπος του επιδημικού αλγορίθμου επηρεάζουν το συνολικό ποσοστό μόλυνσης των κόμβων του δικτύου, τον ολικό χρόνο μόλυνσης των κόμβων, το χρόνο παραμονής των κόμβων σε κατάσταση μόλυνσης.

Σε έναν επιδημικό αλγόριθμο, κάθε διαδικασία του συστήματος ενδεχομένως να εμπλέκεται στη διάδοση. Βασικά, κάθε διαδικασία αποθηκεύει κάθε μήνυμα (μονάδα πληροφορίας) που λαμβάνει μέχρι μιας ορισμένης χωρητικότητας b (buffer capacity), και προωθεί το μήνυμα κατά ένα περιορισμένο αριθμό φορών t . Η διαδικασία διαβιβάζει το μήνυμα κάθε φορά σε έναν τυχαία επιλεγμένο σύνολο διαδικασιών περιορισμένου μεγέθους f , που καλείται άπλωμα (fanout) της διάδοσης.

Πολλές παραλλαγές των επιδημικών αλγορίθμων διάδοσης υπάρχουν και διακρίνονται από τιμές των παραπάνω παραμέτρων: b , t και f . Αυτές οι παράμετροι διάδοσης μπορούν να είναι είτε ανεξάρτητες από τον αριθμό n των διαδικασιών στο σύστημα είτε μπορούν να μεταβάλλονται με το μέγεθος N του συστήματος. Η αξιοπιστία της μετάδοσης των πληροφοριών εξαρτάται από τις τιμές των παραμέτρων καθώς επίσης και από το μέγεθος n του συστήματος.

Εγγενώς, η αξιοπιστία των επιδημικών αλγορίθμων βασίζεται σ' έναν προ-νοητικό μηχανισμό στον οποίον ο πλεονασμός και η τυχειότητα λειτουργούν έτσι ώστε να παρακάμπτονται πιθανές αποτυχίες διαδικασίας καθώς και αποτυχίες συνδέσεων δικτύου. Αυτό συμβαίνει επειδή, εξ ορισμού, κάθε διαδικασία επιλέγει τυχαία ένα υποσύνολο του μεγέθους f άλλων διαδικασιών στις οποίες οι πληροφορίες διαβιβάζονται. Μια διαδικασία που λαμβάνει τις πληροφορίες επιλέγει ένα άλλο υποσύνολο και ούτω καθ' εξής. Καμία πρόνοια και κανένας μηχανισμός δεν απαιτείται για την ανίχνευση και την αναμόρφωση των αποτυχιών. Αντίθετα από τους αναδραστικούς αλγορίθμους στους οποίους οι διαδικασίες αναδρούν στις αποτυχίες με την αναμετάδοση των ελλειπουσών πληροφοριών.



Κατά κάποιο τρόπο, οι επιδημικοί αλγόριθμοι εμφανίζουν μια διττή συμπεριφορά: υπάρχει ένα κρίσιμο κατώτατο όριο στις τιμές των παραμέτρων διάδοσης για τις οποίες μια αξιόπιστη μετάδοση έχει πολύ υψηλή πιθανότητα. Η πιθανοτική εγγύηση αξιόπιστης μετάδοσης του μηνύματος σχετίζεται άμεσα με τις τιμές των παραμέτρων διάδοσης. Αυτές οι παράμετροι μπορούν να ρυθμιστούν έτσι ώστε με μια αυθαίρετα υψηλή πιθανότητα, ο επιδημικός αλγόριθμος να ικανοποιεί τις εγγυήσεις αξιοπιστίας που παρέχουν και οι ντετερμινιστικοί αλγόριθμοι.

2.1 Ορισμοί.

Παρακάτω αναφέρονται ορισμένοι ορισμοί σχετικοί με τις επιδημίες [10]:

Ορισμός 1: Ένας πληθυσμός είναι ένα σύνολο κόμβων. Ο πληθυσμός μπορεί να οριστεί χωρικά, π.χ. όλοι οι κόμβοι που βρίσκονται σε κάποια συγκεκριμένη γεωγραφική περιοχή. Οι κόμβοι ενός πληθυσμού είναι ικανοί να διαχέουν πληροφορία πλαισίου μεταξύ τους με στόχο να ενημερώνουν και να συμπεραίνουν το πλαίσιο συνεργαζόμενοι μεταξύ τους. Έστω N το μέγεθος του πληθυσμού των κόμβων και ο δείκτης i , ο κάθε κόμβος, i του N .

Ορισμός 2: Οι γείτονες ενός κόμβου i δημιουργούν τη γειτονιά του i , (Γ_i υποσύνολο N). Γ_i είναι το σύνολο των κόμβων (συμπεριλαμβανομένου και του i) που βρίσκονται σε ακτίνα επικοινωνίας με τον i . Ο επιδημικός αλγόριθμος επιλέγει τυχαία ένα υποσύνολο (μεγέθους f) του Γ_i με σκοπό να διαχύσει ο i κόμβος μια ενημερωμένη έκδοση ενός κομματιού πληροφορίας. Τα κριτήρια επιλογής αναφέρονται παρακάτω.

Ορισμός 3: Ένας κόμβος i ονομάζεται υγιής (Susceptible) αν δε μεταφέρει ενημερωμένο πλαίσιο πληροφορίας και είναι ευαίσθητος (εύτρωτος) προς μόλυνση (μπορεί να δεχτεί ενημερωμένο πλαίσιο πληροφορίας) από ένα μολυσματικό κόμβο j που ανήκει στη γειτονιά Γ_i ή αν έχει ιαθεί (π.χ. μεταφέρει ένα πολυδιατηρημένο κομμάτι πλαισίου).

Ορισμός 4: Ένας κόμβος i ονομάζεται μολυσμένος όταν για ένα καθορισμένο χρονικό διάστημα μεταφέρει ενημερωμένο πλαίσιο πληροφορίας και αφού μολυνθεί παραμένει μολυσματικός μέχρι αυτή η πληροφορία να γίνει άχρηστη.

Ορισμός 5: Ένας κόμβος i ονομάζεται ανοσοποιημένος (Recovered) όταν για ένα καθορισμένο χρονικό διάστημα μολύνθηκε, στη συνέχεια ιάθηκε αλλά δε μπορεί να ξαναμολυνθεί (έχει πλέον αντισώματα).



Ορισμός 6: Μια επιδημική μεταστοιχείωση σημαίνει πως η επιδημία αλλάζει αλλά συνεχίζει να παραμένει μολυσματική (αναφερόμενη σε δυνατότερο ιό). Στο πλαίσιο μας η μεταστοιχειωμένη επιδημία αναφέρεται σε μια πιο εξειδικευμένη συνεπαγόμενη κατάσταση (μόλυνση ανώτερου επιπέδου) ώστε ο πληθυσμός να μπορεί να ξαναμολυνθεί.

Ορισμός 7: Ορισμοί των παραμέτρων του επιδημικού αλγορίθμου [10] & [14].

N : Το πλήθος των κόμβων του δικτύου.

b : Η χωρητικότητα της μνήμης των κόμβων (buffer capacity)

t : Ο αριθμός των φορών που ο κάθε κόμβος προωθεί το ενημερωμένο πλαίσιο.

f : Άπλωμα (fanout) της διάδοσης είναι το μέγεθος του υποσυνόλου της γειτονιάς στο οποίο κάθε φορά διαβιβάζεται το μήνυμα.

m : Συμπεριφορά κινητικότητας (η τιμή στο σύνολο $[0, 1]$ καθορίζει την κίνηση ή όχι των κόμβων)

r : Ακτίνα επικοινωνίας (καθορίζει το χώρο εντός του οποίου θεωρούμε ότι οι κόμβοι είναι σε επαφή – δυνατότητα ανταλλαγής πληροφοριών)

β : Ο ρυθμός μόλυνσης (εκφράζει την πιθανότητα ένας κόμβος να μολυνθεί)

δ : Ο ρυθμός ίασης (εκφράζει την πιθανότητα ένας μολυσμένος κόμβος να ιαθεί)

$\lambda(=\beta/\delta)$: Επιδημικό όριο (Η τιμή του καθορίζει αν έχουμε ενδημία, ή πανδημία).

2.2 Τύποι διάδοσης πληροφορίας.

Στους επιδημικούς αλγορίθμους χρησιμοποιούνται δύο τύποι διάδοσης πληροφορίας.

Ο πρώτος αναφέρεται ως «Διάδοση πληροφορίας πλαισίου» («contextual information dissemination») και ο δεύτερος ως «Διάδοση πληροφορίας κατάστασης» («situational context dissemination»).

Σαν πλαίσιο (context) ορίζεται οποιαδήποτε πληροφορία μπορεί να χρησιμοποιηθεί για να χαρακτηρίσει μια συγκεκριμένη ολότητα (entity). Με τον όρο ολότητα μπορεί να περιγραφεί ένας άνθρωπος είτε μία θέση είτε ένα αντικείμενο που θεωρείται σχετικό για την ολοκλήρωση μεταξύ του χρήστη και της εφαρμογής συμπεριλαμβανομένων και των ιδίων [13].

Στον πρώτο τύπο μετάδοσης πληροφορίας η επιδημία αναπαρίσταται από τη μεταφορά δεδομένων πλαισίου (contextual data), όπως μετρήσεις από αισθητήρες (φως,



ταχύτητα, θερμοκρασία), οι οποίες είναι έγκυρες για περιορισμένο χρονικό διάστημα και γεωγραφική περιοχή. Μια ομάδα κόμβων μπορεί να μοιραστεί αυτές τις πληροφορίες με συνεργατικό τρόπο. Αυτό σημαίνει πως κόμβοι που δεν έχουν αισθητήρες, άρα δεν είναι ικανοί να πάρουν μόνοι τους μετρήσεις, μπορούν να αποκτήσουν τα δεδομένα αυτά μετά την εφαρμογή της επιδημικής διάχυσης της πληροφορίας.

Στον δεύτερο τρόπο διάδοσης η συνεπαγόμενη θέση ή κατάσταση ενός κόμβου μπορεί να μεταδοθεί μέσα από ένα σύνολο κόμβων. Πρέπει να τονιστεί πως, η μετάδοση πληροφορίας κατάστασης εξαρτάται άμεσα και ισχυρά από τη μετάδοση πληροφορίας πλαισίου, επειδή η κατάσταση ορίζεται (ή συνεπάγεται) σαν συνδυασμός επιμέρους συστατικών πληροφορίας πλαισίου.

Κατ' αναλογία με το βιολογικό μοντέλο, όπως τα βακτήρια μπορούν να δημιουργούν διαφορετικούς τύπους ασθενειών έτσι από διάφορα κομμάτια πληροφορίας πλαισίου μπορούν να συνεπάγονται διαφορετικές καταστάσεις. Αντιλαμβανόμαστε πως η διάδοση πλαισίου αντιστοιχεί στη διάδοση των βακτηρίων και η διάδοση κατάστασης αντιστοιχεί στην επιδημία.

2.3 Κατηγορίες επιδημικών αλγορίθμων.

Είναι εύκολο να αντιληφθούμε την αναλογία της μετάδοσης πληροφορίας μεταξύ κινητών συσκευών που χρησιμοποιούνται από χρήστες και της διάδοσης μιας μολυσματικής επιδημίας μεταξύ μιας ομάδας ανθρώπων. Και οι δύο μπορούν να χαρακτηριστούν σαν διαδικασίες κατά τις οποίες πρέπει να συμβεί κάποιου είδους «επαφή» για την εξάπλωσή τους, για παράδειγμα ένα «μολυσμένο» αντικείμενο να πλησιάσει σε κάποια απόσταση άλλα μη μολυσμένα αντικείμενα. Έτσι, το είδος των αλγορίθμων που περιγράφουν με αυτόν τον τρόπο τη διάχυση της πληροφορίας ονομάζονται επιδημικοί αλγόριθμοι.

Οι επιδημικοί αλγόριθμοι μπορούν να κατηγοριοποιηθούν σε τρεις μεγάλες κατηγορίες ανάλογα με τον τρόπο που επικοινωνούν οι γειτονικοί κόμβοι μεταξύ τους [10]:

- Αλγόριθμοι Push-based: ο μολυσμένος κόμβος διαχέει τις πληροφορίες του σε επιλεγμένους γειτονικούς κόμβους.
- Αλγόριθμοι Pull-based: ένας υγιής κόμβος ζητά από έναν γειτονικό του να του μεταδώσει την πληροφορία που έχει. Έτσι, σ' αυτή την κατηγορία μόνο αν ο γειτονικός είναι μολυσμένος θα μολυνθεί και ο υγιής.
- Αλγόριθμοι Push&Pull-based: συνδυάζει τους δύο προηγούμενους. Σ' αυτή την κατηγορία είτε ο εξεταζόμενος κόμβος είναι μολυσμένος είτε οι γείτονές του είναι μολυσμένοι, θα μολυνθούν όλοι.



2.4 Μοντέλα Επιδημικών Αλγορίθμων.

Συνδυάζοντας μόλυνση και ίαση (και γενικά τις διάφορες καταστάσεις στις οποίες μπορεί να βρεθεί ένας κόμβος) προκύπτουν διαφορετικά μοντέλα επιδημιών.

2.4.1 Μοντέλο SI.

Το επιδημικό μοντέλο ονομάζεται Susceptible-Infected (SI) δηλαδή Υγιής-Μολυσμένος. Στο μοντέλο SI οι κόμβοι θεωρούνται σε μια από τις δύο διακριτές καταστάσεις: υγιής ή μολυσμένος. Σ' αυτό το μοντέλο αν ένας κόμβος μολυνθεί δε μπορεί να ιαθεί (παραμένει για πάντα μολυσμένος – διατηρεί την πληροφορία πλαισίου που έχει λάβει και η οποία θεωρείται έγκυρη για πάντα). Στο μοντέλο αυτό $\delta=0$.

2.4.2 Μοντέλο SIS.

Το επιδημικό μοντέλο ονομάζεται Susceptible-Infected-Susceptible (SIS) δηλαδή Υγιής-Μολυσμένος-Υγιής. Αυτό είναι ένα πολύ διαδεδομένο μοντέλο που χρησιμοποιείται στην επιδημιολογική έρευνα.

Το μοντέλο SIS:

- i. αγνοεί τις λεπτομέρειες μόλυνσης μέσα στον κόμβο,
- ii. απλοποιεί την επιδημική μετάδοση ως πιθανότητες ανά χρονική μονάδα των ρυθμών β και δ και
- iii. θεωρεί έναν κόμβο σε μία από τις δύο διακριτές καταστάσεις: μολυσμένος ή υγιής.

Το μοντέλο SIS θεωρεί ότι ένας μολυσμένος κόμβος δεν μπορεί να ξαναμολυνθεί. Στην περίπτωση που αναφερόμαστε στη διάχυση πληροφορίας πλαισίου ο περιορισμός της μη επαναμόλυνσης ενός ήδη μολυσμένου κόμβου είναι αποδεκτός. Από τη άλλη όταν αναφερόμαστε σε διάχυση κατάστασης, στην οποία η επιδημία μπορεί να μεταλλαχθεί (περιστασιακό πλαίσιο *situational context*) το μοντέλο SIS επεκτείνεται ώστε ένας μολυσμένος κόμβος να μπορεί να ξαναμολυνθεί από ισχυρότερη μόλυνση (μόλυνση ανώτερου επιπέδου). Επιπλέον το μοντέλο υποθέτει ότι ένας κόμβος δεν μπορεί να γίνει λιγότερο επιρρεπής μετά από οποιαδήποτε μόλυνση. Έτσι υποθέτουμε ότι τα β και δ δεν αλλάζουν με τον χρόνο.



2.4.3 Μοντέλο SIR.

Το επιδημικό μοντέλο ονομάζεται Susceptible-Infected- Recovered (SIR) δηλαδή Υγιής-Μολυσμένος-Ανοσοποιημένος

Στο μοντέλο SIR ένας κόμβος που είναι Υγιής μπορεί με κάποια πιθανότητα να μολυνθεί και να γίνει Μολυσμένος. Από την κατάσταση της μόλυνσης μπορεί να ιαθεί και είτε να ξανακαταστεί ευαίσθητος (Susceptible) για μόλυνση είτε να αποκτήσει αντισώματα και να μη μπορεί πλέον να ξαναμολυνθεί (Recovered)

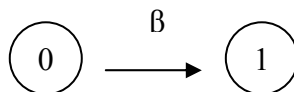
Στο μοντέλο SIR εκτός από τους ρυθμούς β και δ ορίζεται και ο ρυθμός

α :=Ο ρυθμός ανοσοποίησης (εκφράζει την πιθανότητα ένας μολυσμένος κόμβος να μεταβεί από την κατάσταση Infected στην κατάσταση Recovered)

2.5 Αλγόριθμοι διάδοσης Πληροφορίας βάσει Επιδημικών μοντέλων

2.5.1 Ο SI Αλγόριθμος.

Στον αλγόριθμο SI οι καταστάσεις ενός κόμβου μπορούν να είναι δύο: υγιής και μολυσμένος. Οι μεταβάσεις μεταξύ αυτών είναι μόνο από 0 σε 1, που περιγράφεται σαν μόλυνση κόμβου. Σχηματικά η κατάσταση αυτή περιγράφεται στο Σχήμα 1 που ακολουθεί.



Ελέγχεται αν ένας κόμβος j που βρίσκεται σε υγιή κατάσταση μπορεί με κάποια τυχαία πιθανότητα να μολυνθεί από κάποιον από όλους τους μολυσμένους γειτονικούς του κόμβους. Ο αλγόριθμος παρουσιάζεται παρακάτω:



Ο Επιδημικός Αλγόριθμος SI για έναν κόμβο j σε μια χρονική στιγμή t

Input (Infection Probability β)

If S[j] <- SUSCEPTIBLE

For each node $i=1,\dots,L$, (i not j) Do

If(S[i] == INFECTED) // βρέθηκε κόμβος σε κατάσταση INFECTED και
μπορεί να μολύνει κόμβο με πιθανότητα β και
εφόσον είναι μέσα

στην ακτίνα μόλυνσης R (Euclidean Distance

If (SQRT((D[j,t,0] - D[i,t,0])² + (D[j,t,1] - D[i,t,1])²) <= R) Then

If (Random() < β) Then // έλεγχος αν υπάρχει πιθανότητα $P(x) < \beta$ ο κόμβος
να μολυνθεί.

S[j] <- INFECTED

End If

End If

End For

End For

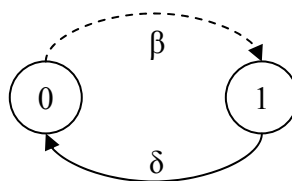
End If

End Algorithm

Αλγόριθμος 1

2.5.2 Ο SIS Αλγόριθμος.

Στον αλγόριθμο SIS οι καταστάσεις ενός κόμβου μπορούν να είναι δύο: υγιής και μολυσμένος. Έτσι, οι μεταβάσεις μεταξύ αυτών είναι από 1 σε 0, που περιγράφεται σαν ίαση κόμβου και από 0 σε 1, που αντιστοιχεί στη μόλυνση κόμβου. Σχηματικά η κατάσταση αυτή περιγράφεται στο Σχήμα 2 που ακολουθεί.





Αρχικά ελέγχεται αν ένας κόμβος j που βρίσκεται σε μολυσμένη κατάσταση με κάποια τυχαία πιθανότητα μπορεί να μεταβεί σε υγιή (ίαση), διαφορετικά για όλους τους μολυσμένους γειτονικούς του κόμβους ελέγχεται αν μπορεί κάποιος από αυτούς να τον μολώνει.

Ο αλγόριθμος παρουσιάζεται παρακάτω:

Ο Επιδημικός Αλγόριθμος SIS για έναν κόμβο j σε μια χρονική στιγμή t

Input(Infection Probability β , Recovery Probability δ)

If ($S[j] == \text{INFECTED}$) Then

If ($\text{Random}() < \delta$) // έλεγχος αν υπάρχει πιθανότητα $P(x) < \delta$ να γίνει ο κόμβος «υγιής»

$S[j] \leftarrow \text{SUSCEPTIBLE}$

End If

Else

For each node $i=1, \dots, L$, ($i \neq j$) Do

If ($S[i] == \text{INFECTED}$) // βρέθηκε κόμβος σε κατάσταση INFECTED και μπορεί να μολώνει κόμβο με πιθανότητα β και εφόσον είναι μέσα στην ακτίνα μόλυνσης R (Euclidean Distance

If ($\text{SQRT}((D[j,t,0] - D[i,t,0])^2 + (D[j,t,10] - D[i,t,1])^2) \leq R$) Then

If ($\text{Random}() < \beta$) Then // έλεγχος αν υπάρχει πιθανότητα $P(x) < \beta$ ο κόμβος να μολυνθεί.

$S[j] \leftarrow \text{INFECTED}$

End If

End If

End If

End For

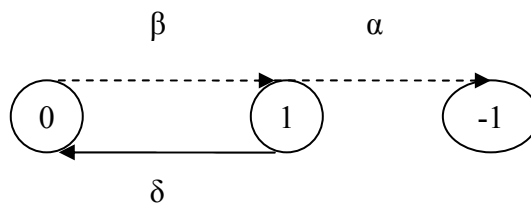
End If

End Algorithm

Αλγόριθμος 2

2.5.3 Ο SIR Αλγόριθμος.

Στον αλγόριθμο SIR οι καταστάσεις ενός κόμβου μπορούν να είναι τρεις: υγιής, μολυσμένος και ανοσοποιημένος. Έτσι, οι μεταβάσεις μεταξύ αυτών είναι από 0 σε 1, που περιγράφεται σαν μόλυνση κόμβου, από 1 σε 0, που αντιστοιχεί σε ίαση του κόμβου και από 1 σε -1 που αντιστοιχεί σε ανοσοποίηση του κόμβου (μη δυνατότητα επαναμόλυνσης). Σχηματικά η κατάσταση αυτή περιγράφεται στο σχήμα Σχήμα 3 που ακολουθεί.



Σχήμα 4

Αρχικά ελέγχεται αν ένας κόμβος j που βρίσκεται σε μολυσμένη κατάσταση με κάποια τυχαία πιθανότητα μπορεί να μεταβεί σε υγιή (ίαση) ή σε ανοσοποιημένη κατάσταση, διαφορετικά, κι εφόσον δεν είναι Recovered, για όλους τους μολυσμένους γειτονικούς του κόμβους ελέγχεται αν μπορεί κάποιος από αυτούς να τον μολύνει.

Ο αλγόριθμος παρουσιάζεται παρακάτω:

Ο Επιδημικός Αλγόριθμος SI για έναν κόμβο j σε μια χρονική στιγμή t

Input(Infection Probability β , Recovery Probability α , Cure Probability δ)

If ($S[j] == \text{INFECTED}$) Then

(If (Random() < α) // έλεγχος αν υπάρχει πιθανότητα $P(x) < \alpha$ να γίνει ο
κόμβος « Recovered »

$S[j] \leftarrow \text{RECOVERED}$

Else If($\alpha < \text{Random()} < \alpha + \delta$) // έλεγχος αν υπάρχει πιθανότητα $P(x) < \delta$ να γίνει
ο κόμβος « Susceptible »

$S[j] \leftarrow \text{SUSCEPTIBLE}$

End If

End If)

End If



```
Else
For each node  $i=1, \dots, L$ , ( $i \text{ not } j$ ) Do
  If( $S[i] == \text{INFECTED}$ ) // βρέθηκε κόμβος σε κατάσταση INFECTED και
                        // μπορεί να μολύνει κόμβο με πιθανότητα  $\beta$  και
                        // εφόσον είναι μέσα στην ακτίνα μόλυνσης  $R$ 
                        // (Euclidean Distance
  If ( $\text{SQRT}((D[j,t,0] - D[i,t,0])^2 + (D[j,t,10] - D[i,t,1])^2) \leq R$ ) Then
    If ( $\text{Random}() < \beta$ ) Then // έλεγχος αν υπάρχει πιθανότητα  $P(x) < \beta$  ο κόμβος
                                // να μολυνθεί.
       $S[j] \leftarrow \text{INFECTED}$ 
    End If
  End If
End If
End For
End If

End Algorithm
```

Αλγόριθμος 3

2.6 Το εξεταζόμενο SIS μοντέλο.

Στην παρούσα εργασία θα μελετηθεί η περίπτωση του SIS μοντέλου για ένα μόνο επίπεδο μόλυνσης. Αφού επιλεγεί το μοντέλο κινητικότητας (Random Waypoint - RWP) [11] & [12], των κόμβων θα εξεταστεί η συμπεριφορά του αλγορίθμου για την περίπτωση των κινούμενων, με αυτό το μοντέλο, κόμβων και για την περίπτωση των ακίνητων κόμβων και για διάφορες τιμές των παραμέτρων λ και τ του αλγορίθμου, δηλαδή θα εξετάσουμε τις περιπτώσεις:

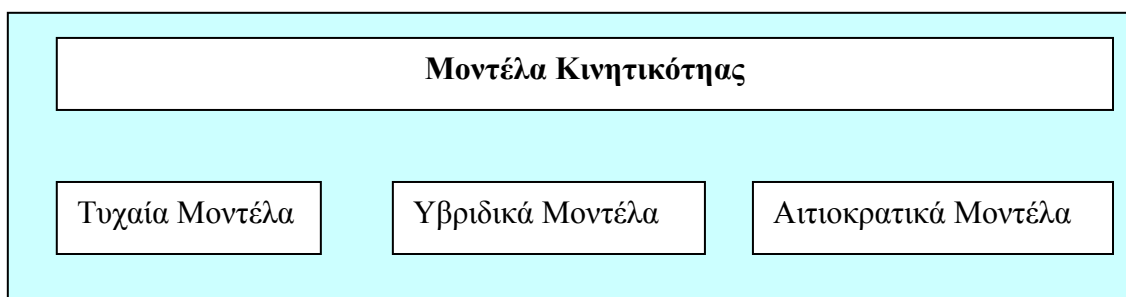
- για $m=0$ (ακίνητοι κόμβοι) και $m=1$ (συνεχώς κινούμενοι κόμβοι),
- για $\lambda > 1$ (εξάπλωση της επιδημίας) και $\lambda < 1$ (εξασθένιση της επιδημίας) και
- για $\tau=2$ (μικρή γειτονιά), $\tau=5$ (ενδιάμεση γειτονιά) και $\tau=10$ (μεγάλη γειτονιά).

3 Μοντέλα Κινητικότητας.

Στο κεφάλαιο αυτό επιχειρείται μια σύντομη παρουσίαση των βασικών τύπων μοντέλων κινητικότητας και ειδικά του μοντέλου που επιλέχθηκε να χρησιμοποιήσει ο προσομοιωτής κίνησης κόμβων στην παρούσα εργασία, του μοντέλου RWP.

3.1 Τύποι μοντέλων κινητικότητας.

Η προσομοίωση μοντέλων δικτύων κινητών κόμβων είναι ένα ιδιαίτερα κρίσιμο θέμα στη μελέτη της κινητικότητας και ανάγεται στην προσομοίωση της κινητικής συμπεριφοράς του χρήστη. Στην Εικόνα 1 εμφανίζεται μια κατηγοριοποίηση των μοντέλων της κινητικότητας χρηστών σε δίκτυα.



Εικόνα 1

Τα Μοντέλα Τυχαίας Κινητικότητας προσομοιώνουν τυχαίες κινήσεις σε καθορισμένη περιοχή και η υλοποίησή τους είναι σχετικά εύκολη. Οι παράμετροι της κίνησης αλλάζουν κατά την προσομοίωση και παράγονται έτσι τυχαίες κατευθύνσεις. Οι περισσότερες προσομοιώσεις δικτύων κινητών κόμβων ακολουθούν αυτό το μοντέλο αφού αυτοσκοπός, δεν είναι συνήθως, η κατασκευή μοντέλων κίνησης, αλλά η ανάπτυξη νέων επικοινωνιακών πρωτοκόλλων ή η έρευνα πάνω σε αλγόριθμους επικοινωνίας (όπως και στην παρούσα εργασία). Τα μοντέλα αυτού του τύπου χρησιμοποιούνται για στρατιωτικές εφαρμογές ή για εφαρμογές έκτακτης ανάγκης. Όμως η τυχαία κίνηση δεν αντιστοιχίζεται με την πραγματική κίνηση, επειδή για παράδειγμα υπάρχουν και φυσικοί περιορισμοί στην πραγματική κίνηση (π.χ. Βουνά, Τοίχοι κλπ), ή ατομικές ή/κ' κοινωνικές συνήθειες των χρηστών (κόμβων) που εμφανίζονται σαν εμπόδια σε μια εντελώς τυχαία κίνηση.



Σε αντίθεση με τα Μοντέλα Τυχαίας Κινητικότητας, υπάρχουν τα Αιτιοκρατικά Μοντέλα Κίνησης που αναπαριστούν την πραγματική πορεία κίνησης των κόμβων (χρηστών) σε μια περιοχή προσομοίωσης. Τέτοια μοντέλα μπορούν κάλλιστα να αναπαραστήσουν την πραγματική κίνηση αλλά περιορίζονται σε προκαθορισμένη περιοχή. Η απόκτηση τέτοιων σημείων κίνησης απαιτεί υπερβολική δουλειά για να τεθούν οι πραγματικοί παράμετροι στον προσομοιωτή. Πρέπει να προετοιμαστούν ερωτηματολόγια θέσης, χιλιάδες αισθητήρες αναφοράς θέσης, μηχανές ιχνηλάτησης ή άλλα όργανα, και γενικά μια διαδικασία πολύπλοκη κι ένας μηχανισμός κοστούβουρος, που θα προωθούν την πληροφορία θέσης της περιοχής προσομοίωσης στον προσομοιωτή.

Οι χρήστες όμως, στην πραγματικότητα, δεν κινούνται τελικά πάντα τυχαία αλλά με κάποιο βαθμό γενικότητας και προβλεψιμότητας στην κίνησή τους. Έτσι πρέπει να βρεθεί μια χρυσή τομή μεταξύ των μη πραγματικών, μη προκαθορισμένων, τυχαίων κινήσεων, με προκαθορισμένες και πραγματικές τροχιές κίνησης των χρηστών.

Τα Υβριδικά Μοντέλα Κίνησης συνδιάζουν τις παραμέτρους της τυχαίας κίνησης με πληροφορίες περιορισμού της ή «καθοδήγησης» της χωρικά ή/κ' χρονικά, με σκοπό οι αναπαραστάσεις της κίνησης των κόμβων του μοντέλου προσομοίωσης να προσεγγίζουν σε πραγματικές κινήσεις με κάποια αποδεκτή πιθανότητα. Τέτοια μοντέλα κυμαίνονται μεταξύ της απλότητας της τυχαίας κίνησης και της αυξημένης πολυπλοκότητας της αιτιοκρατικής κίνησης.

3.1.1 Μοντέλα τυχαίας κινητικότητας.

Οι πρώτες προσπάθειες προσομοίωσης σε δίκτυα κινητών κόμβων χρησιμοποιούν σενάρια που βασίζονται στο σχεδιασμό των κυψελών με την χρήση των σταθμών βάσεων (κεντροποιημένη δομή) με χαρακτηριστικά τέτοια παραδείγματα τα δίκτυα εταιρειών κινητής τηλεφωνίας. Στόχος του προσομοιωτή είναι η δημιουργία μιας λίστας κυψελών από τις οποίες διέρχεται ο κινητός κόμβος. Στο μονοδιάστατο μοντέλο κίνησης η κάθε κυψέλη έχει δύο μόνο γειτονικές κυψέλες με δύο μόνο περιορισμούς κίνησης, ενώ στο δυσδιάστατο μοντέλο μπορεί να υπάρξει ένα πλήθος γειτονικών κυψελών, κυρίως από 6 και πάνω, που προσομοιώνεται μια περιοχή κάλυψης όπως μια πόλη στην οποία οι κόμβοι κινούνται από γειτονική κυψέλη προς γειτονική κυψέλη έχοντας κάποια πιθανότητα ύπαρξης σε μία από τις 6 κυψέλες. Αυτό το είδος κίνησης των κόμβων καλείται Μοντέλο Τυχαίας Πορείας Κίνησης.

Το μοντέλο ορίζεται ως εξής: Σε κάθε διακριτή χρονική στιγμή t ο χρήστης κινείται σε μία από τις γειτονικές κυψέλες με πιθανότητα p ή μένει στην τρέχουσα κυψέλη με πιθανότητα $1-p$, ανεξάρτητη από την προηγούμενη κίνησή του. Εάν ο χρήστης αποφασίσει να κινηθεί σε άλλη κυψέλη τότε μια από τις γειτονικές υποψήφιες κυψέλες επιλέγεται με



ίση πιθανότητα. Επισημαίνεται ότι οι περισσότεροι χρήστες σε τέτοιου είδους δίκτυα είναι πεζοί. Η κίνησή τους λαμβάνει χώρα μέσα σε μια βασική γεωγραφική έκταση με συχνές ενέργειες τύπου «κινούμαι- σταματώ» και συχνές αλλαγές κατευθύνσεων. Οι αλυσίδες Markov και οι διεργασίες Markov μπορούν να χρησιμοποιηθούν για τον μαθηματικό formalισμό του μοντέλου. Η διεργασία Markov είναι μια στοχαστική διεργασία που η μελλοντική της συμπεριφορά προκαθορίζεται μόνο από την τρέχουσα κατάσταση. Αυτή η ιδιότητα επιτρέπει στη διεργασία να μοντελοποιεί την Τυχαία Πορεία Κίνησης της οποίας η συμπεριφορά αποφασίζεται μόνο από την τρέχουσα κατάσταση. Εάν οι καταστάσεις της διεργασίας Markov λαμβάνουν μόνο διακριτές τιμές, τότε μπορεί να περιγραφεί μια μετάβαση με τη χρήση μιας μήτρας πιθανοτήτων P . Ένα στοιχείο της μήτρας $P[i,j]$ περιγράφει την πιθανότητα μετάβασης από την i θέση στην j (π.χ. από μια κυψέλη σε γειτονική της).

3.1.2 Αιτιοκρατικά Μοντέλα.

Μια άλλη κατηγορία πρότυπων μοντέλων είναι η κατηγορία των αιτιοκρατικών μοντέλων κινητικότητας, η οποία καθορίζεται από τα δεδομένα (ίχνη) της κίνησης, που λήφθηκαν από την πραγματική ζωή χρησιμοποιώντας διάφορες κοινωνιολογικές μεθόδους, όπως προσωπικά ερωτηματολόγια, ταξίδια κλπ. ή άλλες μεθόδους, όπως εντοπισμός θέσης των χρηστών κλπ.

Το πλεονέκτημα του μοντέλου είναι η κοντινή του συσχέτιση με την πραγματικότητα, αλλά απ' την άλλη, απαιτεί υψηλή προετοιμασία και μεγάλο κόστος για να διεξαχθεί μια έρευνα ή για να εγκατασταθεί ένας μηχανισμός λήψης δεδομένων, προκειμένου να ληφθούν τα στοιχεία πηγής.

3.1.3 Υβριδικά μοντέλα.

Τα υβριδικά μοντέλα κινητικότητας συνδυάζουν την τυχαία μετακίνηση με την τακτικότητα της αιτιοκρατικής μετακίνησης.

Ένα τέτοιο μοντέλο χτίζεται μεταξύ δύο επιπέδων:

- το καθολικό μοντέλο κινητικότητας (GMM)
- το τοπικό μοντέλο κινητικότητας (LMM)

Το GMM χρησιμοποιείται για να δημιουργήσει intercell (μεταξύ των κυψελών) μετακινήσεις, ενώ το LMM χρησιμοποιείται για να διαμορφώσει intracell (μέσα στην κυψέλη) μετακινήσεις. Η παρουσία του τοπικού μοντέλου προκύπτει από το γεγονός ότι οι περισσότεροι από τους κινητούς χρήστες παρουσιάζουν μερική τακτικότητα στην καθημερινή τους μετακίνηση. Αυτή η τακτικότητα μπορεί να περιγραφεί καλύτερα από



διάφορα πρότυπα κινητικότητας χρηστών. Η ακρίβεια τέτοιων προτύπων είναι πολύ καλή και εμφανίζουν μικρές αποκλίσεις από την πραγματική τροχιά κίνησης του χρήστη.

Η LMM αναπαριστά τη μετακίνηση μέσα σε ένα στοιχείο και περιγράφεται από τρεις μεταβλητές: θέση, ταχύτητα και κατεύθυνση.

Χαρακτηριστικό μοντέλο τέτοιου τύπου είναι το Μοντέλο τυχαίας κίνησης σημείου με εμπόδια RWP (RWP με εμπόδια).

3.2 Μοντέλο Random Waypoint (RWP).

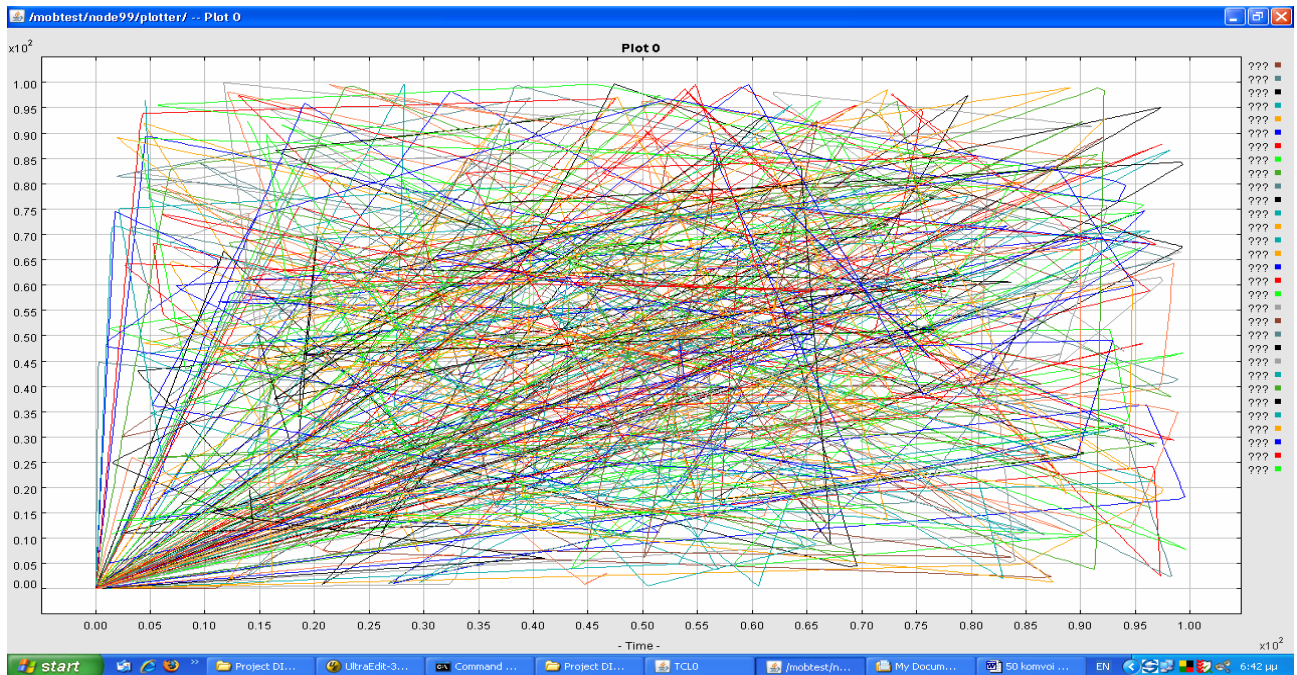
Η ανάλυση απόδοσης κατά την παρουσία κινητικότητας είναι σημαντικής σπουδαιότητας στο σχεδιασμό των ασύρματων δικτύων επικοινωνίας και υπολογιστών. Όπως ήδη αναφέρθηκε η επιλογή του κατάλληλου προτύπου που να προσεγγίζει ικανοποιητικά τις πραγματικές κινήσεις των χρηστών δεν είναι ένα εύκολο ζήτημα, καθώς από την επιλογή αυτή εξαρτώνται εν πολλοίς και τα συμπεράσματα που προκύπτουν για τις παραμέτρους των δικτύων που μελετώνται κάθε φορά.

Το συνηθέστερα χρησιμοποιημένο πρότυπο κινητικότητας στην έρευνα δικτύων, για προσομοιώσεις δικτύων κινητών κόμβων, είναι το πρότυπο RWP [11] και [12].

3.2.1 Περιγραφή του μοντέλου.

Στο μοντέλο κίνησης τυχαίας μετοτόπισης σημείου (RWP), το σημείο προορισμού και η ταχύτητα της κίνησης είναι τα βασικά χαρακτηριστικά του μοντέλου. Το μοντέλο ορίζεται ως εξής:

Στην αρχή όλοι οι κόμβοι τοποθετούνται σε αρχικές θέσεις και επιλέγουν το σημείο προορισμού τους μέσα από την περιοχή προσομοίωσης. Επιλέγουν επίσης την ταχύτητα κίνησης. Η ταχύτητα παραμένει σταθερή κατά τη διάρκεια της κίνησης. Αφού φτάσει στον προορισμό ο κόμβος στέκεται εκεί για συγκεκριμένο χρονικό διάστημα κι έπειτα η κίνησή του συνεχίζεται με τον ίδιο τρόπο. Παραδείγματα των ιχνών θέσεων που προέκυψαν από τον προσομοιωτή που υλοποιήθηκε στα πλαίσια της παρούσας εργασίας απεικονίζονται στην Εικόνα 3. Πρόκειται για απεικόνιση ιχνών 100 κόμβων που κινούνται για 1000 χρονικές στιγμές.



Εικόνα 2

Σε μερικά μοντέλα προσομοίωσης η κίνηση ενός χρήστη ορίζει γενικά την κίνηση μιας ομάδας χρηστών. Αυτό αποτελεί βασικό χαρακτηριστικό στην προσομοίωση κινητών κόμβων. Για παράδειγμα η κίνηση ενός ξεναγού σ' ένα μουσείο καθορίζει και την κίνηση μιας ομάδας επισκεπτών.

3.2.2 Ορισμοί για το RWP μοντέλο.

Περιοχή προσομοίωσης: μία ορθογώνια περιοχή συγκεκριμένων διαστάσεων μέσα στην οποία επιτρέπεται η κίνηση των κόμβων.

Χρόνος παύσης: ο προκαθορισμένος χρόνος μικρής διακοπής που ένας κόμβος παραμένει στατικός όταν φθάνει στον προορισμό του, και ο οποίος μπορεί και να διαφοροποιείται μετά από κάθε βήμα. Έπειτα αρχίζει πάλι κίνηση προς το επόμενο επιλεγμένο σημείο.

Αρχική χωρική κατανομή των κόμβων: ο αρχικός προσδιορισμός της θέσης (αφετηρία) των κόμβων.

Ταχύτητα των κόμβων: η οποία είναι ευθύγραμμη ομαλή μεταξύ δύο επιλεγμένων σημείων και η οποία παίρνει τιμές από ένα διάστημα τιμών.



3.2.3 Παράμετροι του RWP μοντέλου.

Οι παράμετροι που ορίζονται αμέσως παρακάτω περιγράφουν μια γενική υλοποίηση προσομοίωσης του μοντέλου κινητικότητας RWP κατά πλήρη τρόπο:

- ✓ Μέγεθος (διαστάσεις) και μορφή της περιοχής προσομοίωσης
- ✓ Αρχική χωρική κατανομή κόμβων $f_{αρχ}(χ)$
- ✓ Στατική παράμετρος $ρ_σ$, με $0 ≤ ρ_σ ≤ 1$
- ✓ Συνάρτηση πυκνότητας πιθανότητας του χρόνου μικρής παύσης $f_{ΓP}(t_p)$
- ✓ Ελάχιστη ταχύτητα και μέγιστη ταχύτητα κίνησης των κόμβων $0 < v_{min} ≤ v_{max}$

Η παράμετρος $ρ_σ$ αντιπροσωπεύει την πιθανότητα ένας κόμβος να παραμένει στατικός για ολόκληρο το χρόνο προσομοίωσης. Η παράμετρος $ρ_σ$ επιτρέπει τη διαμόρφωση μιας κατάστασης στην οποία δύο τύποι κόμβων χρησιμοποιούνται: ένας τύπος είναι στατικός, και ένας άλλος τύπος είναι κινητός. Η ύπαρξη αυτής της παραμέτρου επιτρέπει στην παράμετρο της ταχύτητας να μην παίρνει την τιμή μηδέν για να δηλωθεί η ακινησία κάποιων κόμβων.

4 Εφαρμογή των Επιδημικών Αλγορίθμων σε περιβάλλοντα κινητού υπολογισμού.

Παρατίθενται παρακάτω τα αποτελέσματα του «τρεξίματος» του εξεταζόμενου αλγόριθμου για διάφορες τιμές των παραμέτρων του. Συγκεκριμένα μελετήθηκε η συμπεριφορά του όταν αυτός εφαρμόζεται σε ένα σύνολο 100 κόμβων όταν οι κόμβοι είναι ακίνητοι ($m=0$) και όταν οι κόμβοι κινούνται ($m=1$) σύμφωνα με το μοντέλο RWP.

Για κάθε μία από αυτές τις περιπτώσεις εξετάστηκε η επιδημική εξάπλωση και η επιδημική εξασθένιση σαν συνάρτηση της ακτίνας επικοινωνίας των κόμβων (μικρή γειτονιά ($r=2$), ενδιάμεση γειτονιά ($r=5$), και μεγάλη γειτονιά ($r=10$), για κάθε κόμβο), για σταθερό επιδημικό όριο λ και σαν συνάρτηση του επιδημικού ορίου (λ) (μεγάλο ($\lambda=100$), πολύ μεγάλο ($\lambda=200$), και πάρα πολύ μεγάλο ($\lambda=500$), για την επιδημική εξάπλωση και μικρό ($\lambda=0.5$), πολύ μικρό ($\lambda=0.2$), και πάρα πολύ μικρό ($\lambda=0.1$), για την επιδημική εξασθένιση), για σταθερή ακτίνα επικοινωνίας r .

Σε όλες τις περιπτώσεις τα μοντέλα ξεκίνησαν με πλήθος υγιών κόμβων 90 και πλήθος μολυσμένων 10. Στις περιπτώσεις των μοντέλων με ακίνητους κόμβους εξετάστηκε η συμπεριφορά του επιδημικού αλγόριθμου για 1000 χρονικές στιγμές (επαναλήψεις) και για σχετική θέση των κόμβων που προέκυψε από ένα στιγμιότυπο του προσομοιωτή, ενώ στις περιπτώσεις των κινούμενων για 899 χρονικές στιγμές (και διαφορετικές άρα θέσεις των κόμβων μεταξύ τους κάθε στιγμή θεωρώντας χωρικό και χρονικό σημείο αφετηρίας εφαρμογής του επιδημικού αλγορίθμου την εκατοστή πρώτη σχετική θέση των κόμβων μεταξύ τους όπως αυτή προέκυψε από την εκτέλεση του προσομοιωτή του δικτύου των 100 κόμβων [Παράρτημα Α]). Σε όλες τις περιπτώσεις οι επαναλήψεις ήταν αρκετές ώστε να βγουν ασφαλή συμπεράσματα για την οριακή συμπεριφορά των αλγορίθμων. Βάση όλων αυτών, και με τις προϋποθέσεις που αναφέρθηκαν, μελετήθηκαν τελικά οχτώ (8) σενάρια:

ΣΕΝΑΡΙΟ-1: Επιδημική εξάπλωση συναρτήσει του r ($m=0$, $\beta=0.1$, $\delta=0.01$, $[\lambda=10]$ και $r=2$ ή 5 ή 10).

Εξετάστηκε το σενάριο των ακίνητων κόμβων (συνθήκη κινητικότητας $m=0$) για σταθερό επιδημικό όριο $\lambda > 1$ (εξάπλωση της επιδημίας) και τρεις διαφορετικές τιμές της ακτίνας επικοινωνίας των κόμβων για τις οποίες προκύπτουν διαφορετικού μεγέθους γειτονίες για τους κόμβους σε κάθε περίπτωση.



ΣΕΝΑΠΙΟ -2: Επιδημική εξάπλωση συναρτήσσει του λ ($m=0$, $r=5$, $\delta=0.001$ και $\beta=0.1, [\lambda=100]$ ή $\beta=0.2, [\lambda=200]$ ή $\beta=0.5, [\lambda=500]$).

Εξετάστηκε το σενάριο των ακίνητων κόμβων (συνθήκη κινητικότητας $m=0$) για σταθερή ακτίνα επικοινωνίας των κόμβων ($r=5$, σταθερού μεγέθους γειτονιά σε κάθε περίπτωση για τους κόμβους) και τρεις διαφορετικές τιμές του επιδημικού ορίου (σε κάθε περίπτωση $\lambda > 1$, συνθήκη που εξασφαλίζει την επιδημική εξάπλωση) το οποίο προέκυψε από τη διαφοροποίηση του ρυθμού μόλυνσης β .

ΣΕΝΑΠΙΟ -3: Επιδημική εξασθένιση συναρτήσσει του r ($m=0$, $\beta=0.01$, $\delta=0.1, [\lambda=0.1]$ και $r=2$ ή 5 ή 10).

Εξετάστηκε το σενάριο των ακίνητων κόμβων (συνθήκη κινητικότητας $m=0$) για σταθερό επιδημικό όριο $\lambda < 1$ (εξασθένιση της επιδημίας) και τρεις διαφορετικές τιμές της ακτίνας επικοινωνίας των κόμβων για τις οποίες προκύπτουν διαφορετικού μεγέθους γειτονιές για τους κόμβους σε κάθε περίπτωση.

ΣΕΝΑΠΙΟ -4: Επιδημική εξασθένιση συναρτήσσει του λ ($m=0$, $r=5$, $\delta=0.1$ και $\beta=0.01, [\lambda=0.1]$ ή $\beta=0.02, [\lambda=0.2]$ ή $\beta=0.05, [\lambda=0.5]$).

Εξετάστηκε το σενάριο των ακίνητων κόμβων (συνθήκη κινητικότητας $m=0$) για σταθερή ακτίνα επικοινωνίας των κόμβων ($r=5$, σταθερού μεγέθους γειτονιά σε κάθε περίπτωση για τους κόμβους) και τρεις διαφορετικές τιμές του επιδημικού ορίου (σε κάθε περίπτωση $\lambda < 1$, συνθήκη που εξασφαλίζει την επιδημική εξασθένιση) το οποίο προέκυψε από τη διαφοροποίηση του ρυθμού μόλυνσης β .

ΣΕΝΑΠΙΟ -5: Επιδημική εξάπλωση συναρτήσσει του r ($m=1$, $\beta=0.1$, $\delta=0.01, [\lambda=10]$ και $r=2$ ή 5 ή 10).

Εξετάστηκε το σενάριο των κινούμενων κόμβων (συνθήκη κινητικότητας $m=1$) για σταθερό επιδημικό όριο $\lambda > 1$ (εξάπλωση της επιδημίας) και τρεις διαφορετικές τιμές της ακτίνας επικοινωνίας των κόμβων για τις οποίες προκύπτουν διαφορετικού μεγέθους γειτονιές για τους κόμβους σε κάθε περίπτωση.

ΣΕΝΑΠΙΟ -6: Επιδημική εξάπλωση συναρτήσσει του λ ($m=1$, $r=5$, $\delta=0.001$ και $\beta=0.1, [\lambda=100]$ ή $\beta=0.2, [\lambda=200]$ ή $\beta=0.5, [\lambda=500]$).

Εξετάστηκε το σενάριο των κινούμενων κόμβων (συνθήκη κινητικότητας $m=1$) για σταθερή ακτίνα επικοινωνίας των κόμβων ($r=5$, σταθερού μεγέθους γειτονιά σε κάθε περίπτωση για τους κόμβους) και τρεις διαφορετικές τιμές του επιδημικού ορίου (σε κάθε περίπτωση $\lambda > 1$, συνθήκη που εξασφαλίζει την επιδημική εξάπλωση) το οποίο προέκυψε από τη διαφοροποίηση του ρυθμού μόλυνσης β .



ΣΕΝΑΡΙΟ -7: Επιδημική εξασθένιση συναρτήσει του r ($m=1$, $\beta=0.01$, $\delta=0.1$, $[\lambda=0.1]$ και $r=2$ ή 5 ή 10).

Εξετάστηκε το σενάριο των κινούμενων κόμβων (συνθήκη κινητικότητας $m=1$) για σταθερό επιδημικό όριο $\lambda < 1$ (εξασθένιση της επιδημίας) και τρεις διαφορετικές τιμές της ακτίνας επικοινωνίας των κόμβων για τις οποίες προκύπτουν διαφορετικού μεγέθους γειτονιές για τους κόμβους σε κάθε περίπτωση.

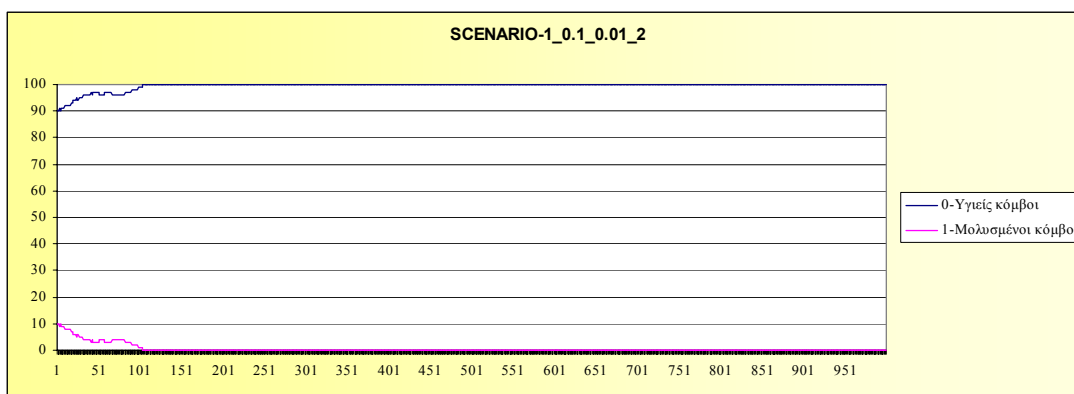
ΣΕΝΑΡΙΟ -8: Επιδημική εξασθένιση συναρτήσει του λ ($m=1$, $r=5$, $\delta=0.1$ και $\beta=0.01$, $[\lambda=0.1]$ ή $\beta=0.02$, $[\lambda=0.2]$ ή $\beta=0.05$, $[\lambda=0.5]$).

Εξετάστηκε το σενάριο των κινούμενων κόμβων (συνθήκη κινητικότητας $m=1$) για σταθερή ακτίνα επικοινωνίας των κόμβων ($r=5$, σταθερού μεγέθους γειτονιά σε κάθε περίπτωση για τους κόμβους) και τρεις διαφορετικές τιμές του επιδημικού ορίου (σε κάθε περίπτωση $\lambda < 1$, συνθήκη που εξασφαλίζει την επιδημική εξασθένιση) το οποίο προέκυψε από τη διαφοροποίηση του ρυθμού μόλυνσης β .

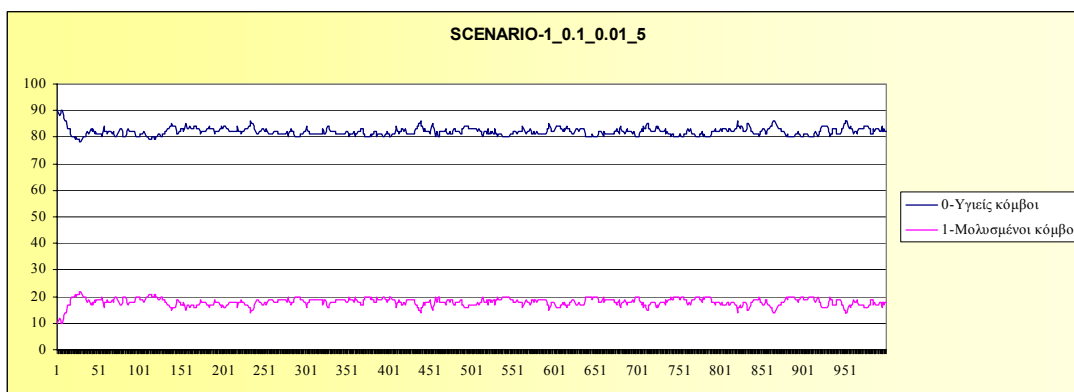
4.1 SIS Μοντέλα για ακίνητους κόμβους.

Εδώ παρατίθενται τα τέσσερα σενάρια για την περίπτωση των ακίνητων κόμβων.

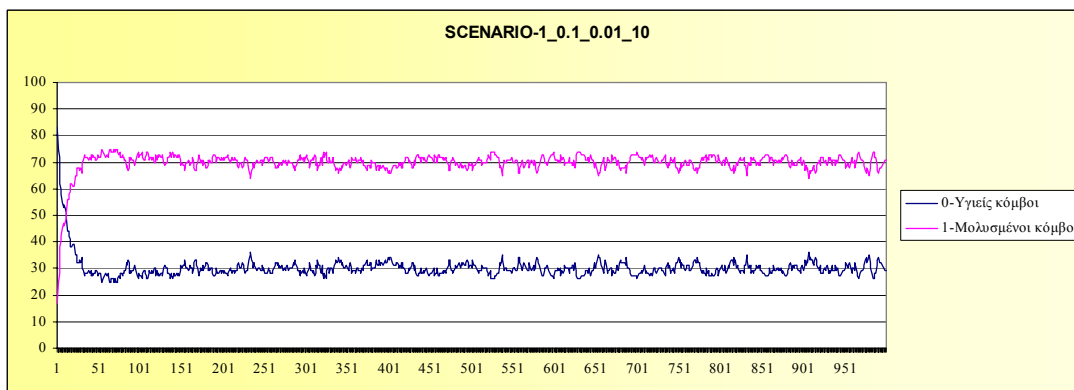
4.1.1 ΣΕΝΑΡΙΟ-1: (Επιδημική εξάπλωση συναρτήσεως του r).



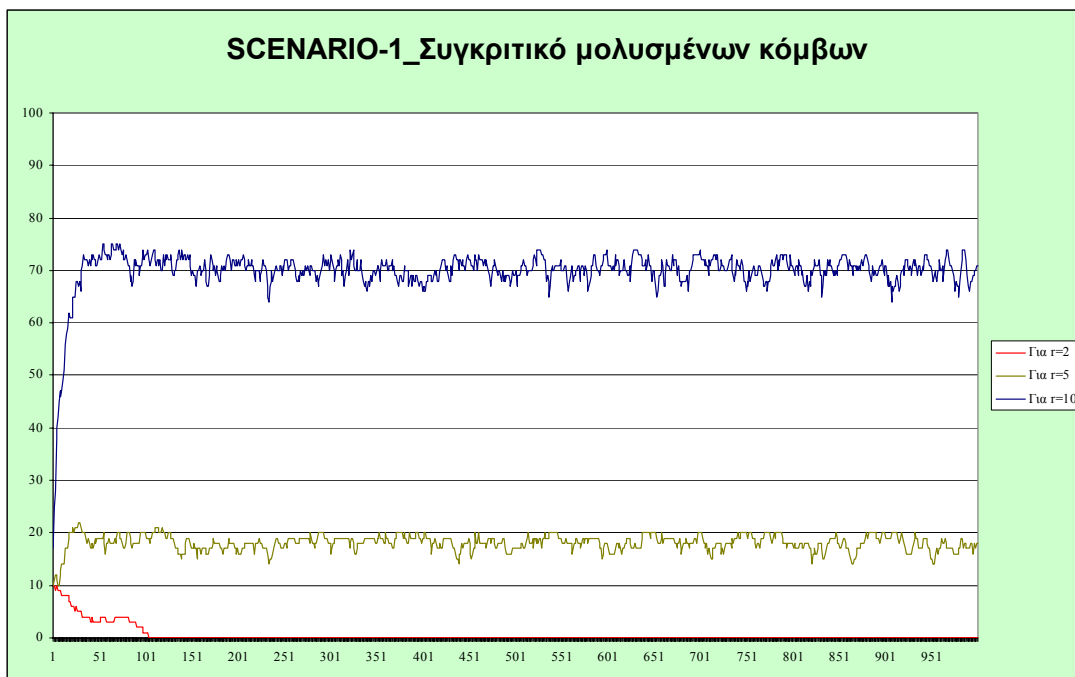
Γράφημα 1



Γράφημα 2



Γράφημα 3

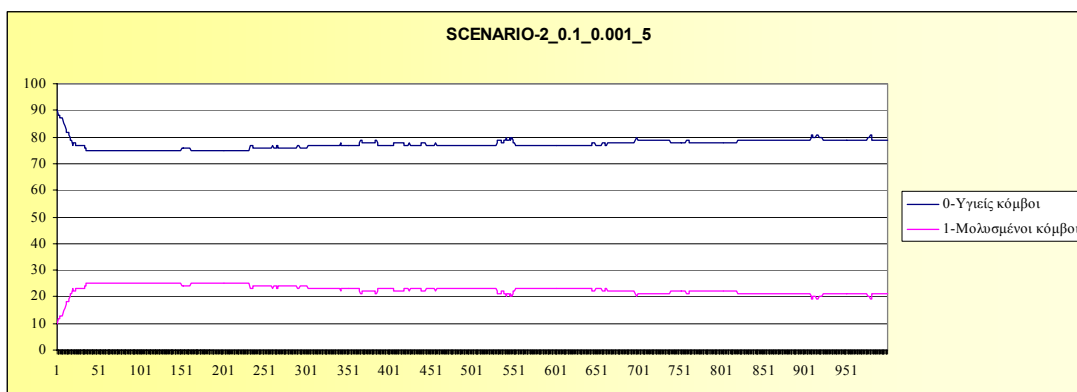


Γράφημα 4

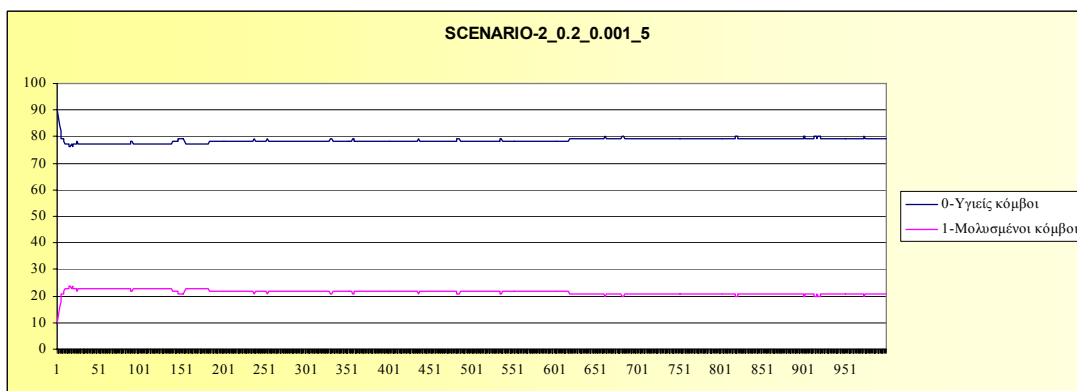
Στο SCENARIO-1 παρατηρείται ότι η ακτίνα επικοινωνίας παίζει πολύ σημαντικό ρόλο για το επίπεδο που θα σταθεροποιηθεί η μόλυνση.

Για μικρή γειτονιά των κόμβων παρουσιάζεται πλήρης ίαση μετά το 100ο βήμα, ενώ απαιτείται να πάρει υψηλή τιμή η ακτίνα για να επιτευχθούν σημαντικά ποσοστά μόλυνσης.

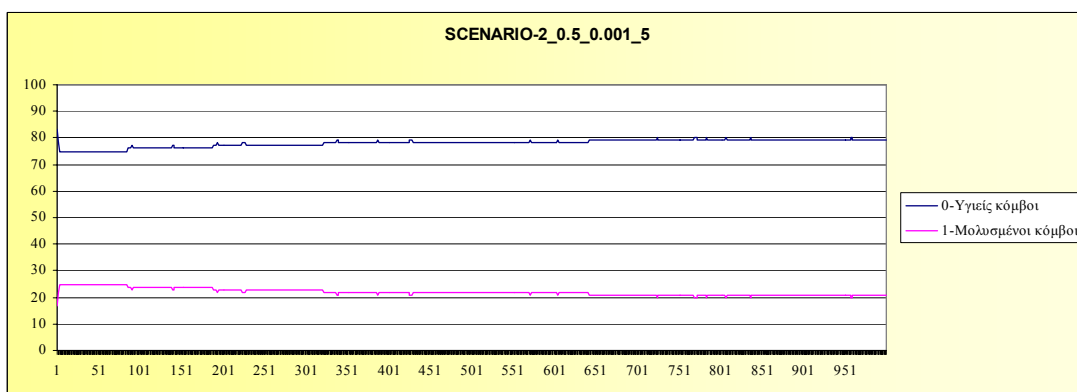
4.1.2 ΣΕΝΑΡΙΟ -2: (Επιδημική εξάπλωση συναρτήσει του λ).



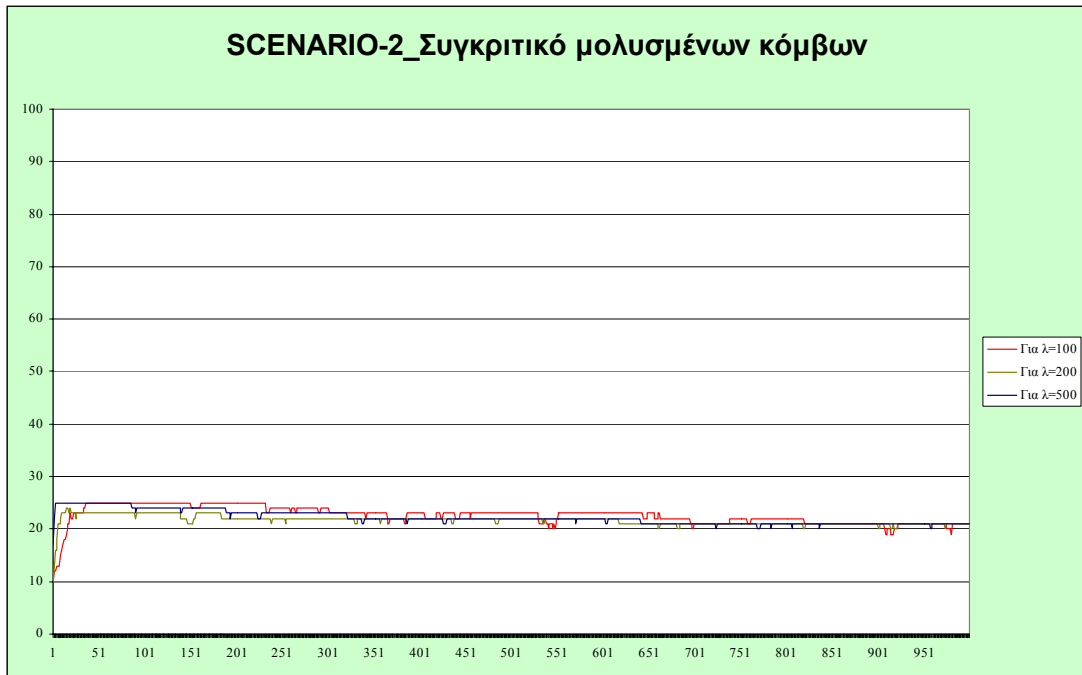
Γράφημα 5



Γράφημα 6



Γράφημα 7



Γράφημα 8

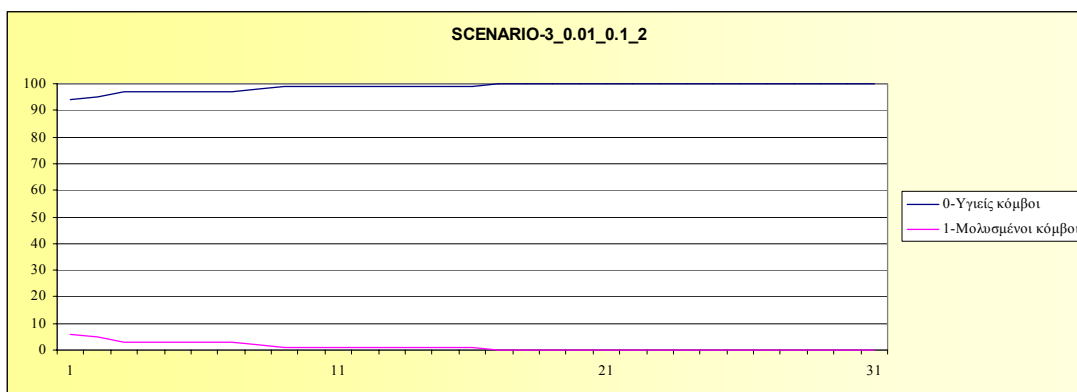
Από το SCENARIO-2 προκύπτει αβίαστα το συμπέρασμα ότι στην περίπτωση των ακίνητων κόμβων το επιδημικό όριο ουσιαστικά δε διαφοροποιεί τα επίπεδα της μόλυνσης που επιτυγχάνονται (και τα οποία στην πραγματικότητα καθορίζονται, όπως ήδη φάνηκε από το προηγούμενο SCENARIO-1 από την παράμετρο r).

Ακόμα και για πολύ μεγάλο λ (δ σχεδόν $=0$ ή/και β σχεδόν $=1$) το ποσοστό μόλυνσης δεν ξεπερνά το 25%.

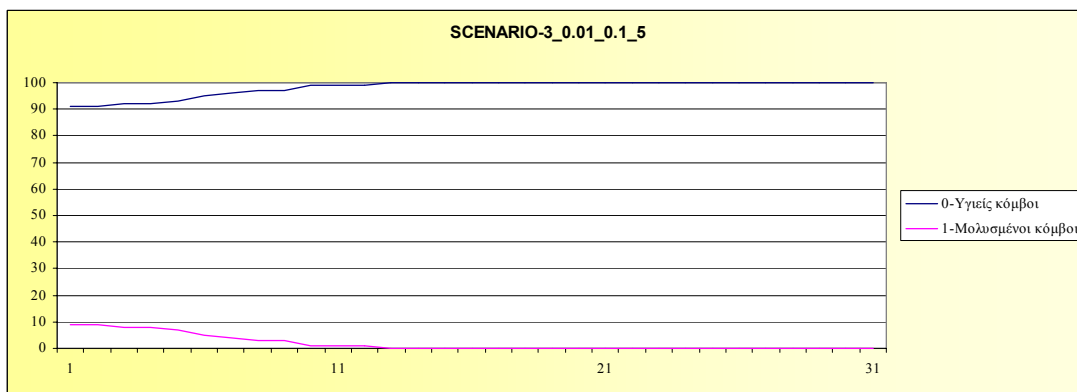
Το επιδημικό όριο φαίνεται να έχει μία μικρή επίδραση στο πόσο γρήγορα επιτυγχάνεται αυτό το ποσοστό (για $\lambda=100$ μετά το 15ο βήμα, για $\lambda=200$ μετά το 6ο βήμα και για $\lambda=500$ μετά το 2ο βήμα).

Σε όλες τις περιπτώσεις το ποσοστό μόλυνσης για $r=5$ σταθεροποιείται μεταξύ 20% και 25%.

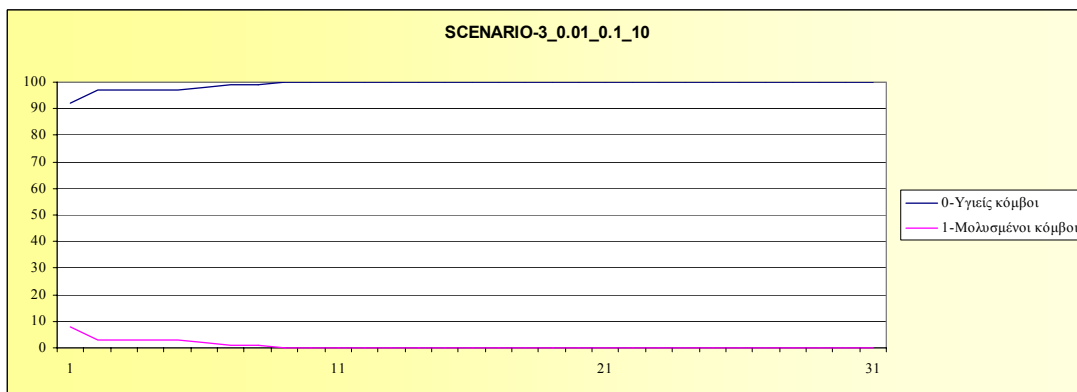
4.1.3 ΣΕΝΑΡΙΟ -3: (Επιδημική εξασθένιση συναρτήσει του r).



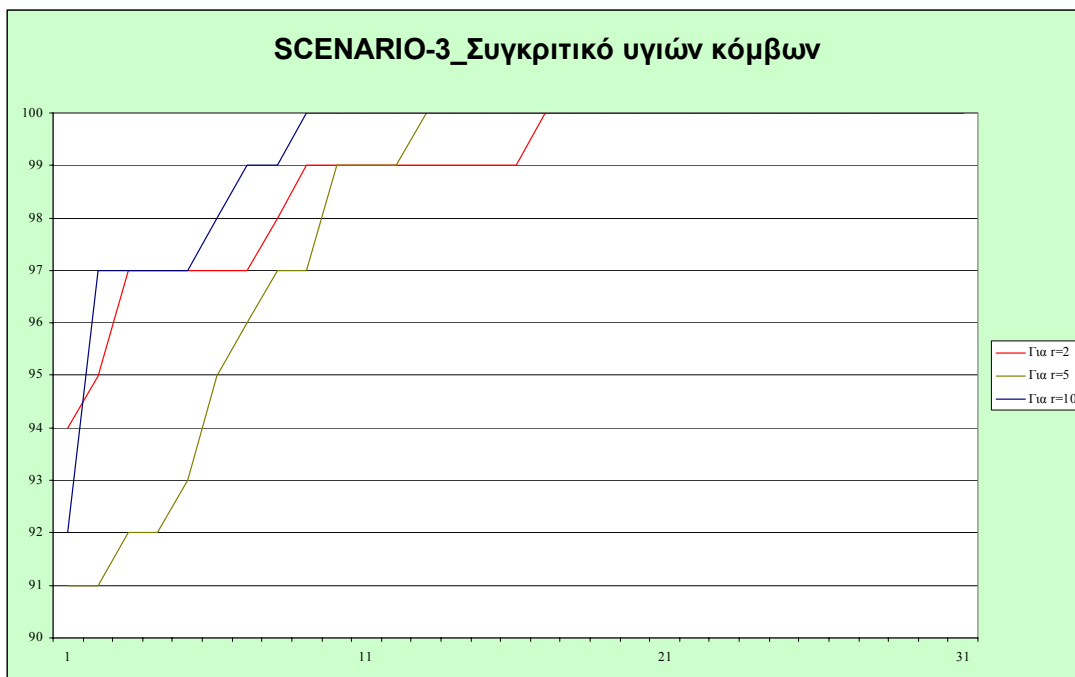
Γράφημα 9



Γράφημα 10



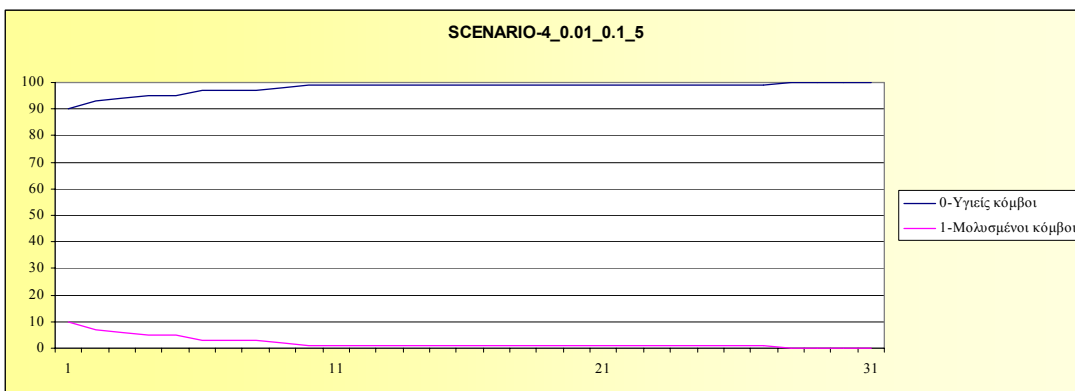
Γράφημα 11



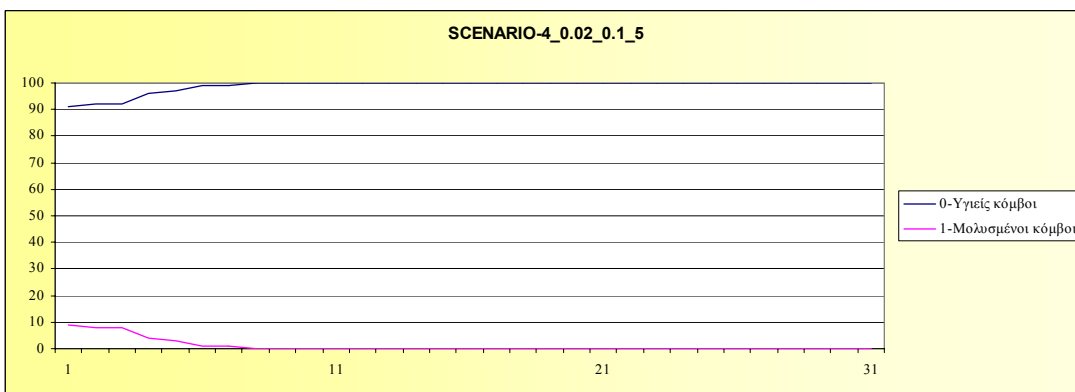
Γράφημα 12

Από το SCENARIO-3 δεν προκύπτει κάποια εξάρτηση της εξυγίανσης των κόμβων από την ακτίνα επικοινωνίας. Γενικά επιτυγχάνεται μια πολύ γρήγορη εξυγίανση όλου του πληθυσμού σε όλες τις περιπτώσεις από το 8ο έως το 17ο βήμα.

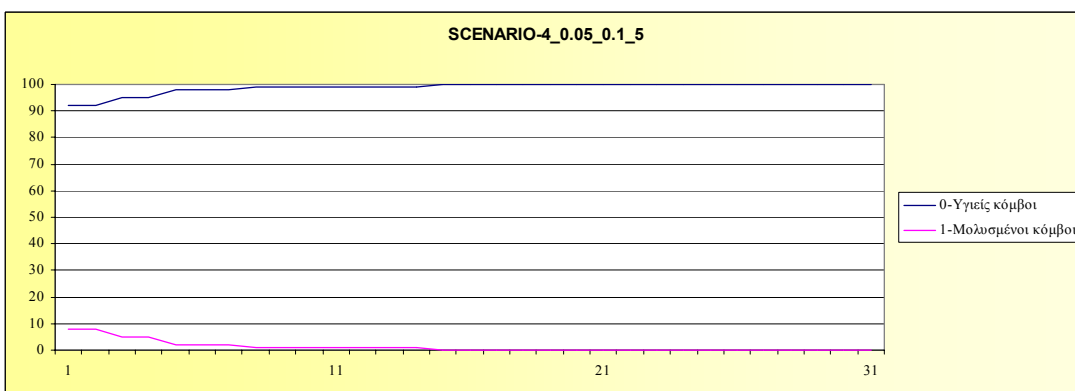
4.1.4 ΣΕΝΑΡΙΟ -4: (Επιδημική εξασθένιση συναρτήσει του λ).



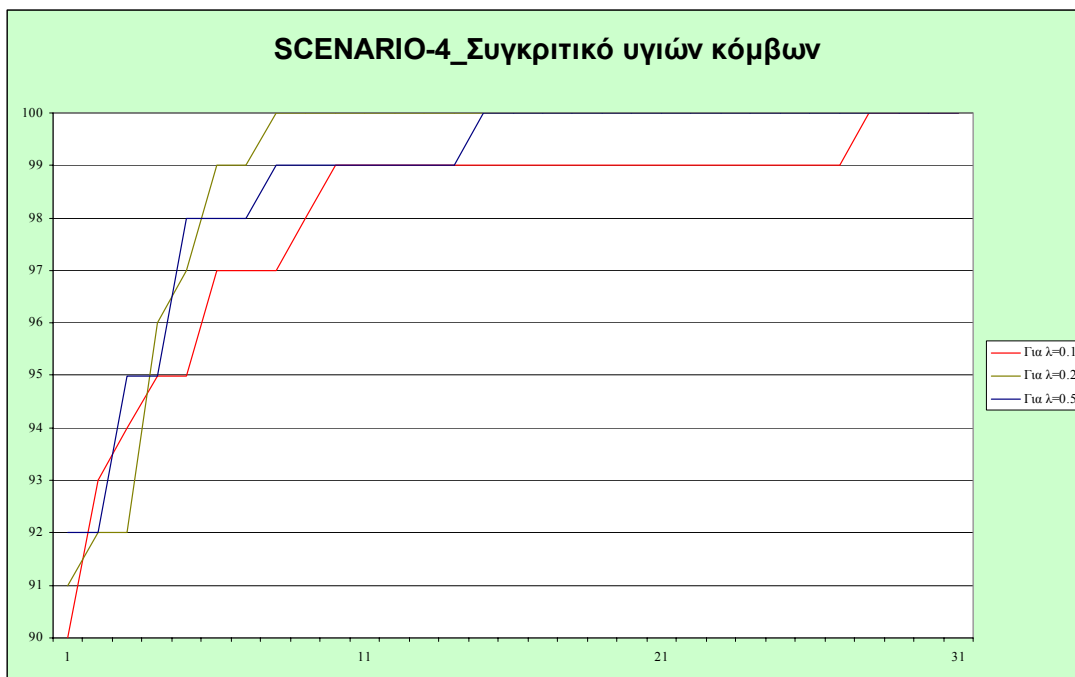
Γράφημα 13



Γράφημα 14



Γράφημα 15



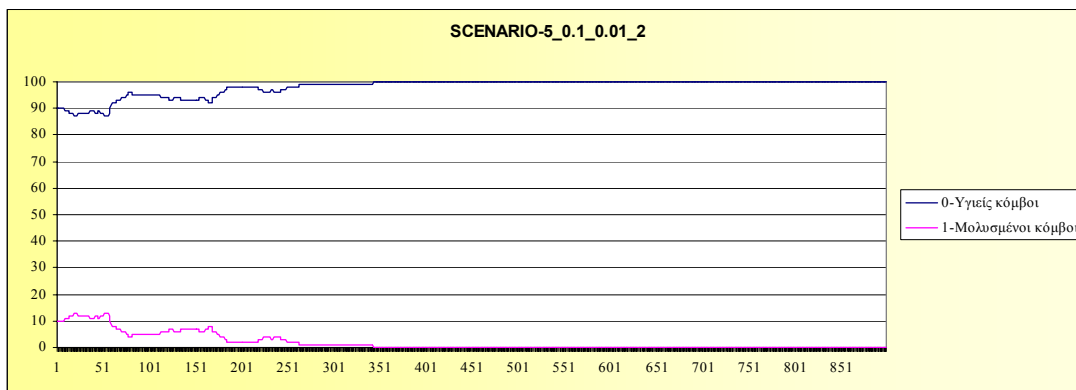
Γράφημα 16

Από το SCENARIO-4 επίσης δε φαίνεται να προκύπτει κάποια εξάρτηση της εξυγίανσης των κόμβων από επιδημικό όριο λ . Γενικά επιτυγχάνεται και σε αυτή την περίπτωση μια πολύ γρήγορη εξυγίανση όλου του πληθυσμού σε όλες τις περιπτώσεις από το 8ο έως το 27ο βήμα.

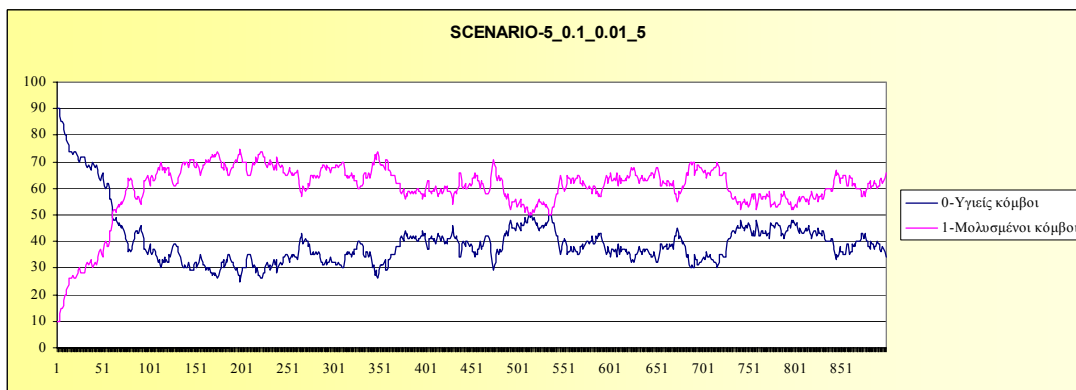
4.2 SIS Μοντέλα για κινούμενους κόμβους.

Εδώ παρατίθενται τα τέσσερα σενάρια για την περίπτωση των κινούμενων κόμβων.

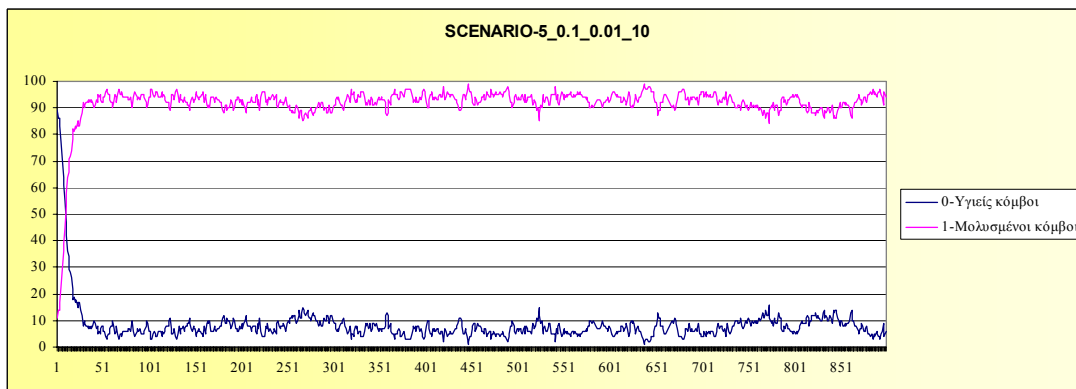
4.2.1 ΣΕΝΑΡΙΟ -5: (Επιδημική εξάπλωση συναρτήσει του r).



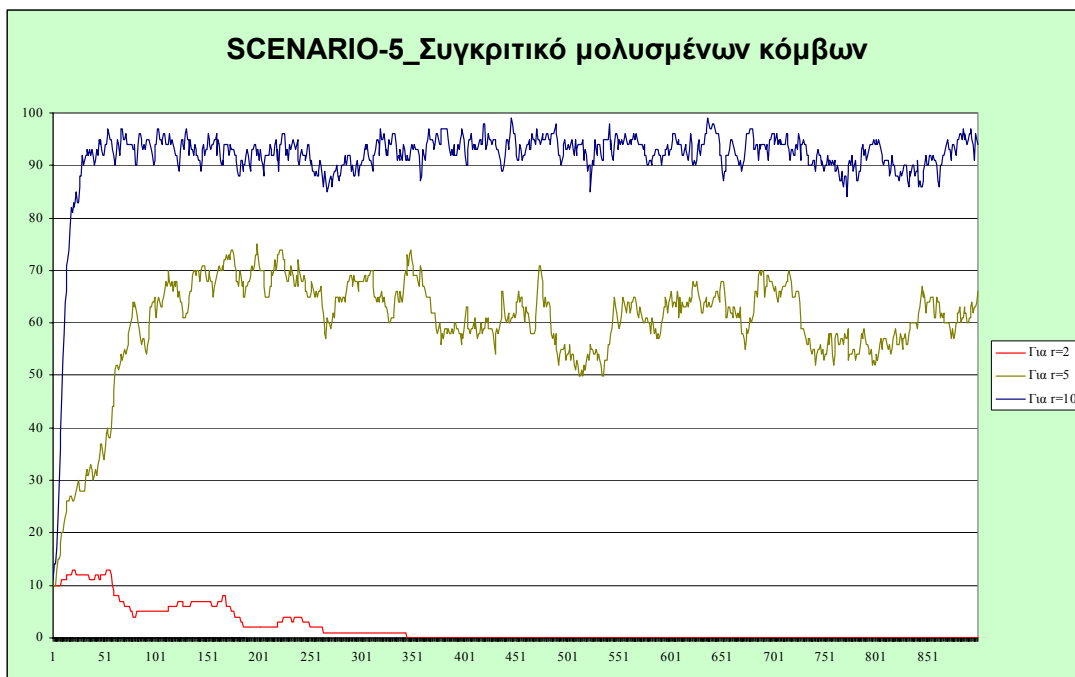
Γράφημα 17



Γράφημα 18



Γράφημα 19



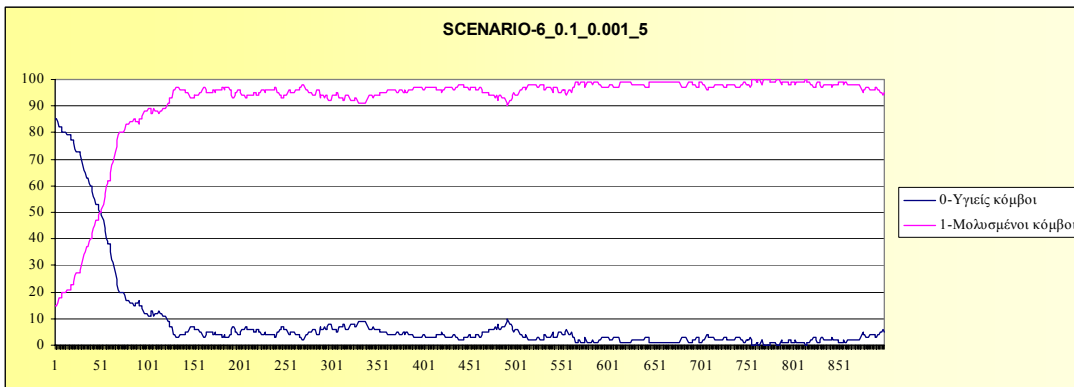
Γράφημα 20

Από το SCENARIO-4 (όπως και από το SCENARIO-1 επίσης) παρατηρείται ότι η ακτίνα επικοινωνίας παίζει σημαντικό ρόλο για το επίπεδο που θα σταθεροποιηθεί η μόλυνση.

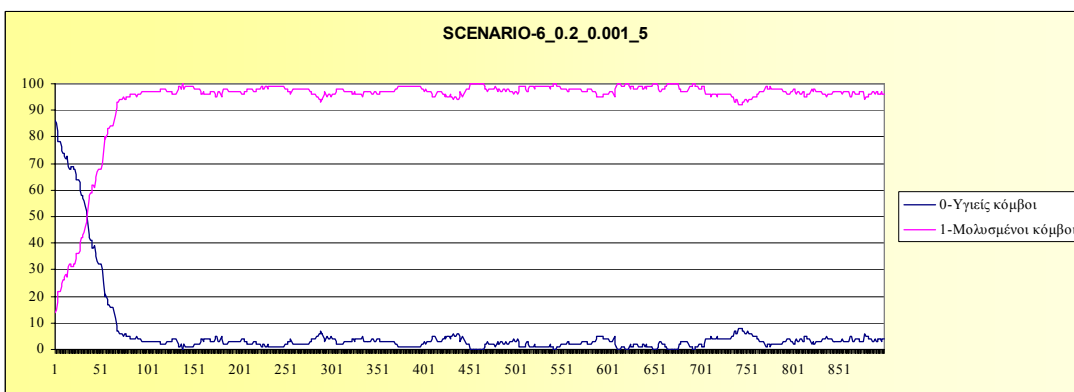
Για μικρή γειτονιά των κόμβων παρουσιάζεται πλήρης ίαση μετά το 400ο βήμα, ενώ αυξανόμενης της τιμής της ακτίνας επιτυγχάνονται υψηλότερα ποσοστά μόλυνσης.

Σε σχέση με το SCENARIO-1 όμως τα ποσοστά μόλυνσης είναι πολύ υψηλότερα για ίδιες ακτίνες επικοινωνίας (για $r=5$ μόλυνση περίπου 25% για $m=0$, ενώ μόλυνση περίπου 60% για $m=1$ και για $r=10$ μόλυνση περίπου 25% για $m=0$, ενώ μόλυνση περίπου 90% για $m=1$).

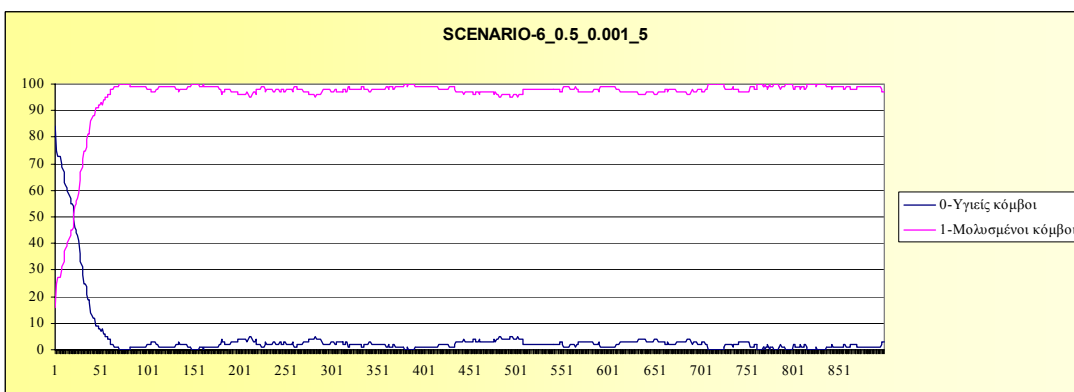
4.2.2 ΣΕΝΑΡΙΟ -6: (Επιδημική εξάπλωση συναρτήσει του λ).



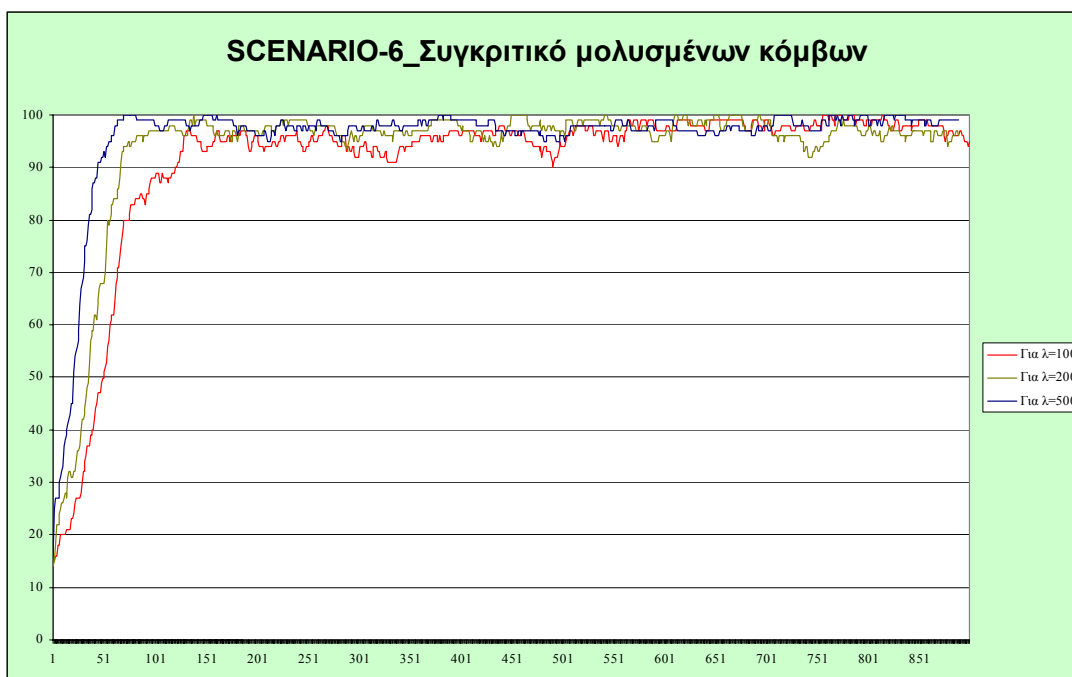
Γράφημα 21



Γράφημα 22



Γράφημα 23



Γράφημα 24

Από το SCENARIO-6 προκύπτει μια μικρή επίδραση του επιδημικού ορίου στο ποσοστό μόλυνση (σε υψηλά πάντως επίπεδα σε κάθε περίπτωση).

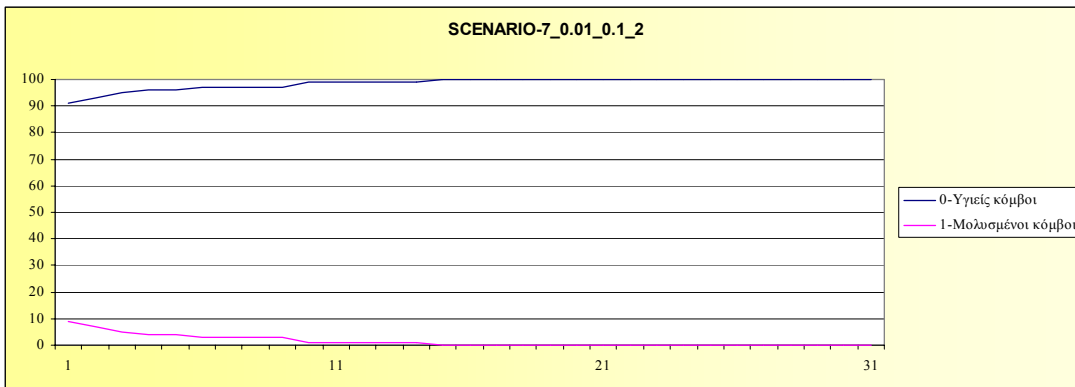
Για $\lambda=100$ η μόλυνση σταθεροποιείται περίπου στο 95%, για $\lambda=200$ η μόλυνση σταθεροποιείται περίπου στο 99% και για $\lambda=500$ η μόλυνση σταθεροποιείται περίπου στο 99%-100%.

Ωστόσο παρατηρείται μια, όχι ασήμαντη, επίδραση στο χρόνο που επιτυγχάνονται αυτά τα ποσοστά μόλυνσης.

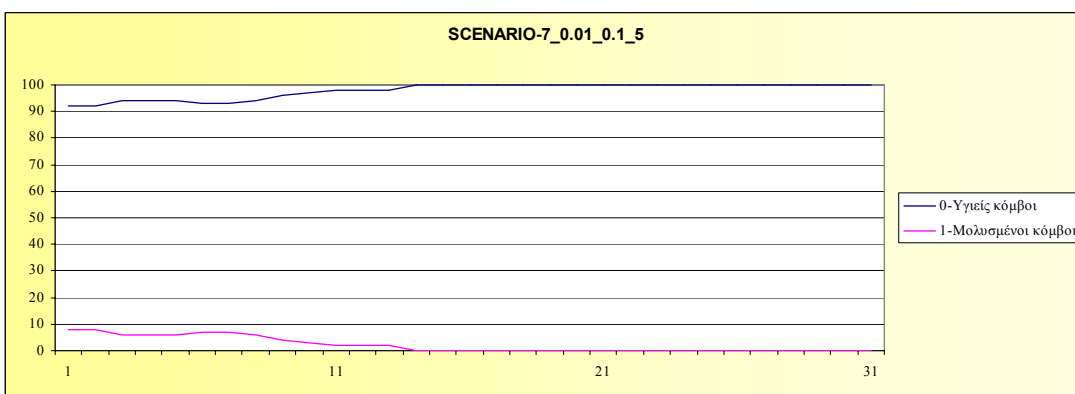
Για $\lambda=100$ η μόλυνση σταθεροποιείται μετά το 100^ο βήμα, για $\lambda=200$ η μόλυνση σταθεροποιείται μετά το 80^ο βήμα, ενώ για $\lambda=500$ η μόλυνση σταθεροποιείται μετά το 60^ο βήμα

Σε σχέση με το SCENARIO-2 όμως τα ποσοστά μόλυνσης είναι πάρα πολύ υψηλότερα για ίδιες ακτίνες επικοινωνίας, για όλες τις περιπτώσεις.

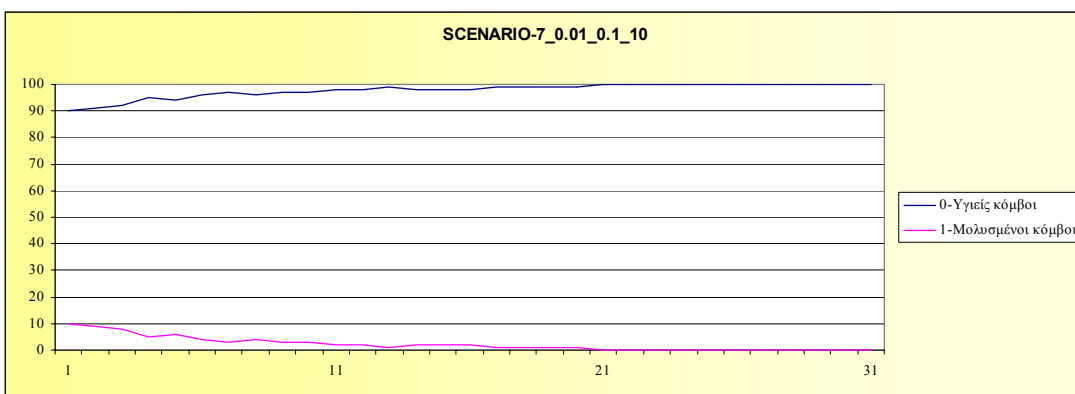
4.2.3 ΣΕΝΑΡΙΟ -7: (Επιδημική εξασθένιση συναρτήσει του r).



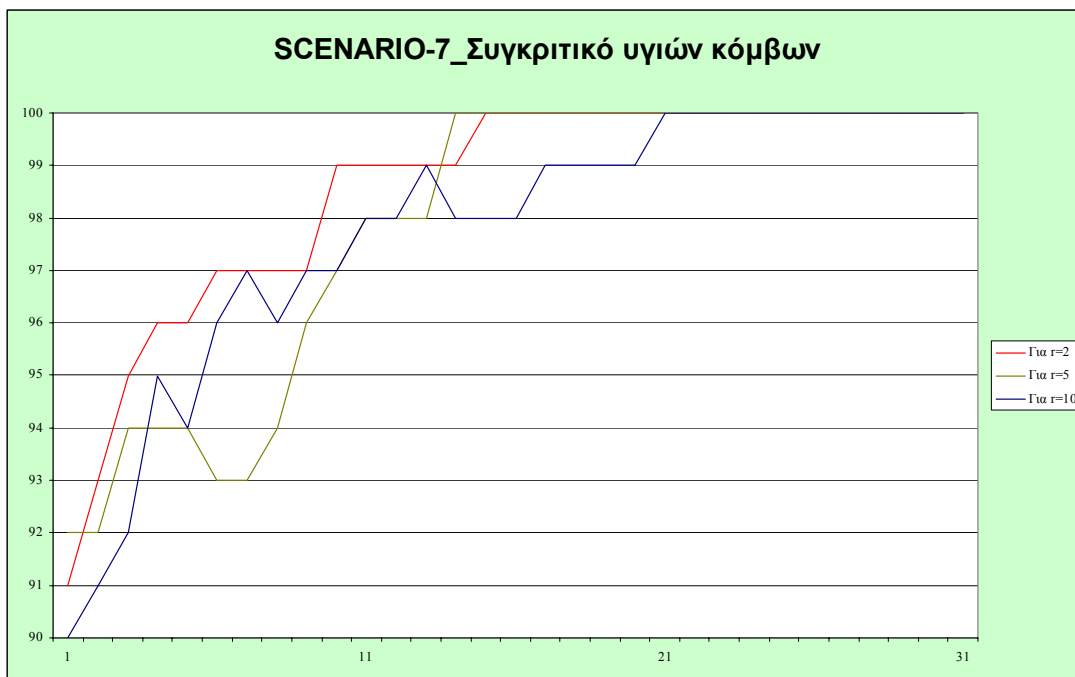
Γράφημα 25



Γράφημα 26



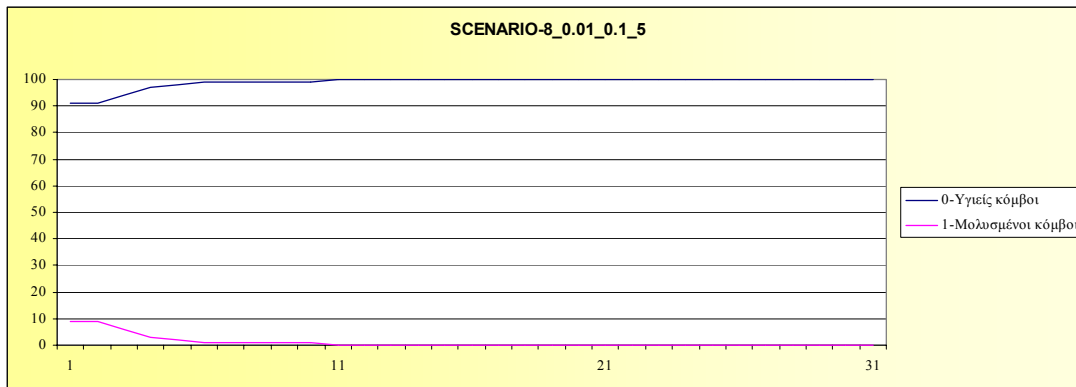
Γράφημα 27



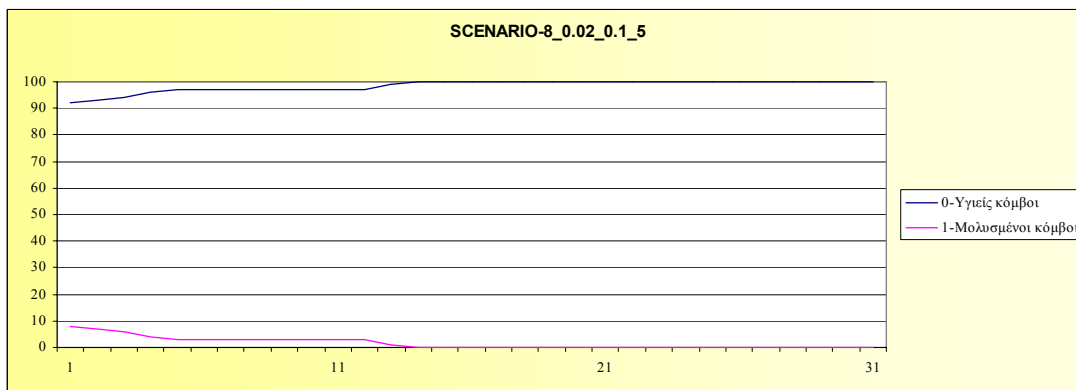
Γράφημα 28

Από το SCENARIO-7 δεν προκύπτει κάποια εξάρτηση της εξυγίανσης των κόμβων από την ακτίνα επικοινωνίας. Γενικά επιτυγχάνεται μια πολύ γρήγορη εξυγίανση όλου του πληθυσμού σε όλες τις περιπτώσεις από το 13ο έως το 20^ο βήμα.

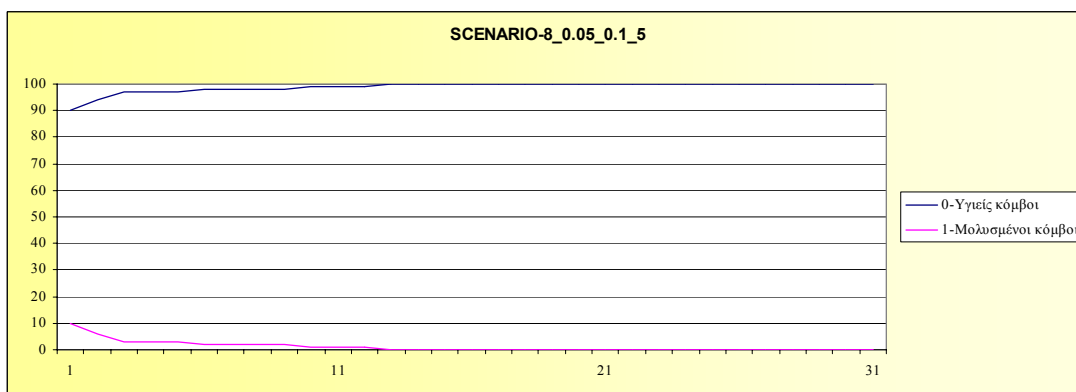
4.2.4 ΣΕΝΑΡΙΟ -8: (Επιδημική εξασθένιση συναρτήσει του λ).



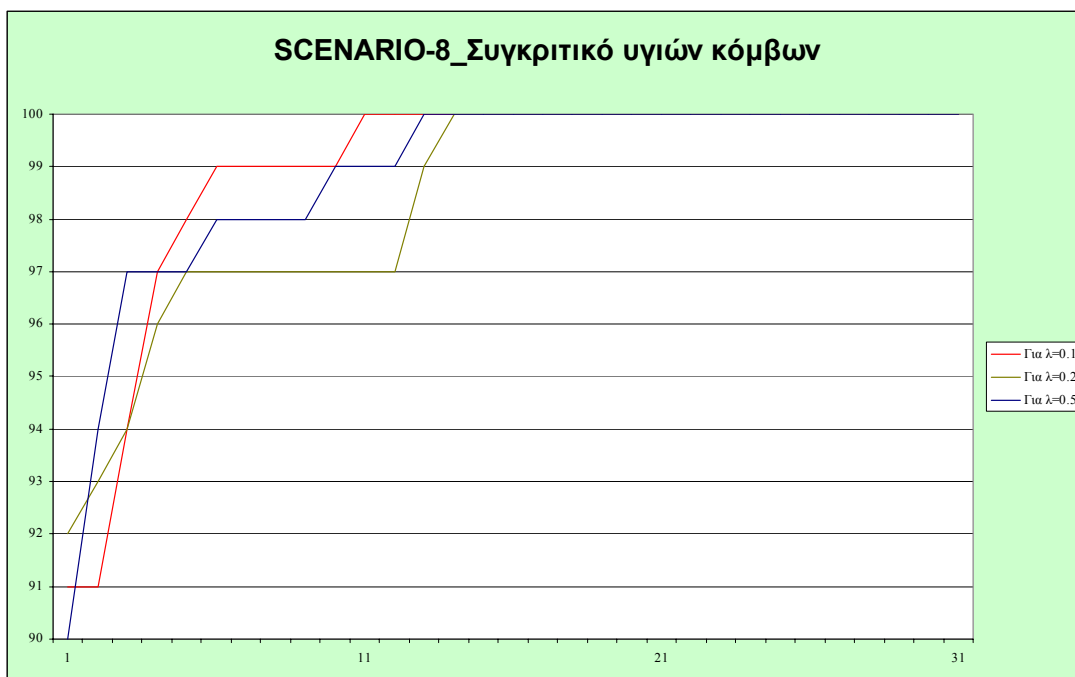
Γράφημα 29



Γράφημα 30



Γράφημα 31



Γράφημα 32

Από το SCENARIO-8 επίσης δεν φαίνεται να προκύπτει κάποια εξάρτηση της εξυγίανσης των κόμβων από επιδημικό όριο λ . Γενικά επιτυγχάνεται και σε αυτή την περίπτωση μια πολύ γρήγορη εξυγίανση όλου του πληθυσμού σε όλες τις περιπτώσεις από το 1ο έως το 14ο βήμα.

Και στο SCENARIO-7 και στο SCENARIO-8 συγκρινόμενα με τα SCENARIO-3 και SCENARIO-4 δε φαίνεται η κινητικότητα να επηρεάζει την επιδημική εξασθένιση.

5 Συμπεράσματα.

5.1 Κατά την Επιδημική Εξάπλωση.

Η ακινησία είναι εμφανώς ανασταλτικός παράγοντας για την εξάπλωση της επιδημίας, ενώ παίζει μεγάλο ρόλο και η ακτίνα επικοινωνίας των κόμβων. Μεγαλώνοντας η ακτίνα επικοινωνίας μεγαλώνει και το ποσοστό μόλυνσης των κόμβων.

Παρατηρούμε ότι για μικρή ακτίνα ($r=2$) στην ακινησία προκύπτει πλήρης εξυγίανση περίπου στο 100° βήμα, ενώ όταν οι κόμβοι είναι κινούμενοι προκύπτει πάλι πλήρης εξυγίανση αλλά μετά το 400° βήμα, δηλαδή πολύ αργότερα.

Το επιδημικό όριο λ δεν επιδρά με την ίδια βαρύτητα. Έχει σαφώς μικρότερη επίδραση στη διάδοση της επιδημίας και στα σενάρια των ακίνητων και στα σενάρια των κινούμενων κόμβων, σε σχέση με την ακτίνα επικοινωνίας.

Το επιδημικό όριο παίζει κυρίως ρόλο στο πόσο γρήγορα (SCENARIO-2 και SCENARIO-6) επιτυγχάνεται το μέγιστο ποσοστό μόλυνσης και το οποίο είναι διαφορετικό για τα μοντέλα των ακίνητων κόμβων (μικρότερα) από ότι για αυτά των κινούμενων κόμβων (μεγαλύτερα). Στην περίπτωση των κινούμενων κόμβων μεγαλύτερο επιδημικό όριο προκαλεί μεγαλύτερο ποσοστό επιδημικής εξάπλωσης, για έτσι κι αλλιώς πολύ υψηλά ποσοστά εξάπλωσης ($> 95\%$).

Η κινητικότητα ανδεικνύεται, με έμφαση σαν η σημαντικότερη παράμετρος για την οποία επιτυγχάνονται γρήγορα πολύ υψηλά ποσοστά μόλυνσης και σταθεροποίησης τους σε αυτά τα επίπεδα, κατά την επιδημική εξάπλωση.

5.2 Κατά την Επιδημική Εξασθένιση.

Εκτός από το ότι η ακινησία των κόμβων προκαλεί πάρα πολύ γρήγορη εξυγίανση όλου του πληθυσμού των κόμβων, καμιά από τις εξεταζόμενες παραμέτρους (κινητικότητα, επιδημικό όριο και ακτίνα επικοινωνίας) δε φαίνεται να παίζει ιδιαίτερο ρόλο στην πλήρη εξυγίανση των κόμβων. Πολύ μικρή σχέση φαίνεται να υπάρχει από την ακτίνα επικοινωνίας r (όσο μικρότερη η ακτίνα τόσο πιο γρήγορη και η εξυγίανση, αλλά, για έτσι κι αλλιώς, μικρά διαστήματα) κι ελάχιστη εξάρτηση από το επιδημικό όριο.

Θα μπορούσαμε να πούμε ότι κάτι τέτοιο είναι αναμενόμενο αφού η εξυγίανση του κάθε κόμβου είναι ατομική υπόθεση κι εξαρτάται μόνο από το ρυθμό ίασης δ και όχι από την κατάσταση κινητικότητας του κόμβου και την επικοινωνία του με άλλους κόμβους (τουλάχιστον στα σενάρια που εξετάζουμε και στα οποία δεν υπάρχει επαναμόλυνση των κόμβων).



6 Αναφορές.

- [1]. ANDREW S. TANENBAUM «ΔΙΚΤΥΑ ΥΠΟΛΟΓΙΣΤΩΝ». Εκδόσεις Παπασωτηρίου, ΤΡΙΤΗ ΕΚΔΟΣΗ, ΑΘΗΝΑ, 2000.
- [2]. DEITEIL & DEITEL «C ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ, Εισαγωγή στην C++ και την Java». Εκδόσεις Μ. Γκιούρδας, Τρίτη Έκδοση, Αθήνα, 2004.
- [3]. Jamew F. Kurose – Keith W. Ross, «Δικτύωση Υπολογιστών», Εκδόσεις Μ. Γκιούρδας, ΔΕΥΤΕΡΗ ΕΚΔΟΣΗ, Αθήνα, 2004.
- [4]. John G. Proakis – Masoyd Salehi «Συστήματα τηλεπικοινωνιών», ΕΚΔΟΣΗ ΕΚΠΑ, ΑΘΗΝΑ, 2002.
- [5]. L. Lemay - R. Cadenhead, «Πλήρες Εγχειρίδιο της Java 2 platform», Εκδόσεις Μ.Γκιούρδας Αθήνα, 1999.
- [6]. P. McBride, «Java ο εύκολος τρόπος», Εκδόσεις Δίαυλος, Αθήνα, 1998.

- [7]. Χρίστος Αναγνωστόπουλος, «Epidemic Models», Pervasive Computing Research Group, University of Athens, epidemics.ppt
- [8]. Χρίστος Αναγνωστόπουλος, Βαγγέλης Ζέρβας Στάθης Χατζηευθυμιάδης, Νταρλαδήμας, «On the use of Epidemical Information Spreading in Mobile Computing Environments», Special Issue on Recent Advances In Heterogeneous Cognitive Wireless Networks, Mediterranean Journal of Computers and Networks.
- [9]. Χρίστος Αναγνωστόπουλος, Στάθης Χατζηευθυμιάδης, «On the Application of Epidemical Spreading in Collaborative Context Aware Computing», ACM SIGMOBILE Mobile Computing and Communications Review (MC2R).
- [10]. Χρίστος Αναγνωστόπουλος, Στάθης Χατζηευθυμιάδης, «A Biological Model for Collaborative Aware System», Transaction on Mobile Computing, TMC-0061-0207.
- [11]. Cristian Bettstetter – Giovanni Resta – Paolo Santi, «The Node Distribution of the Mobility Model for Wirless Ad Hoc Networks».
- [12]. E. Hyytia – H. Koskinen – P. Lassila – A. Penttinen – J. Virtama. «Random Waypoint Model in Wirless Network», HUT, Finland, Ιούνιος 2005
- [13]. James W. Mickenw – Brian D. Noble, «Modeling Epidemic Spreading in Mobile Environments».



- [14]. D. Abowd, «Towards a Better Understanding of Context and Context-Awareness», Proc. Int. Conference of Human Factors in Computing Systems, 2000
- [15]. P.T. Eugster - R. Guerraoui – A.-M. Kermarrec – L. Massoulie, «From Epidemic to Distributed Computing».

- [16]. <http://el.wikipedia.org/wiki/Java>
- [17]. <http://en.wikipedia.org/wiki/Simulation>
- [18]. <http://java.sun.com/developer/onlineTraining/>
- [19]. <http://p-comp.di.uoa.gr/>
- [20]. <http://www.java.com/en/>
- [21]. <http://www.j-sim.org/>
- [22]. http://www.j-sim.org/tutorial/jsim_tutorial.html
- [23]. http://www.j-sim.org/drcl.inet/inet_tutorial.html
- [24]. <http://www.j-sim.org/v1.3/api/index.html>
- [25]. <http://www.j-sim.org/ref.script/index.html#Path%20Expression>
- [26]. <http://www.tcl.tk>



7 Παράρτημα Α: Προσομοιωτής μοντέλου RWP για κίνηση 100 κόμβων.

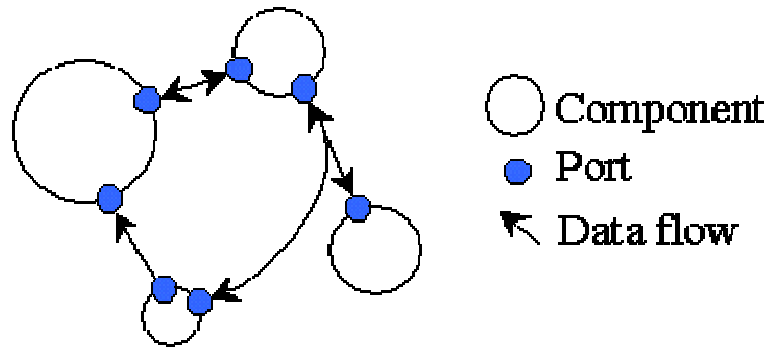
Τα βασικά εργαλεία που χρησιμοποιούνται για την εκπόνηση της διπλωματικής εργασίας είναι η πλατφόρμα προσομοίωσης j-sim και η γλώσσα προγραμματισμού Java. Το περιβάλλον προσομοίωσης j-sim χρησιμοποιείται για τη μοντελοποίηση της κίνησης των κόμβων του ασύρματου δικτύου.

7.1 Εισαγωγή.

Η j-sim (java sim) είναι ένα περιβάλλον προσομοίωσης που βασίζεται στα συστατικά (components) και παρέχεται από το Distributed Computing Laboratory του Ohio State University. Η j-sim είναι μια εφαρμογή της Autonomous Component Architecture (ACA) της Java, που την καθιστά διαθέσιμη για σχεδόν οποιαδήποτε πλατφόρμα σήμερα. Η αρχική της έκδοση (2001) στόχευε στην έρευνα για ενσύρματα δίκτυα, αλλά το 2004 συμπεριέλαβε την επέκταση για τα ασύρματα/κινητά δίκτυα. Έτσι, γίνεται ένα ιδανικό εργαλείο προγραμματισμού για την ασύρματη δικτυακή μοντελοποίηση και την προσομοίωση δικτύων.

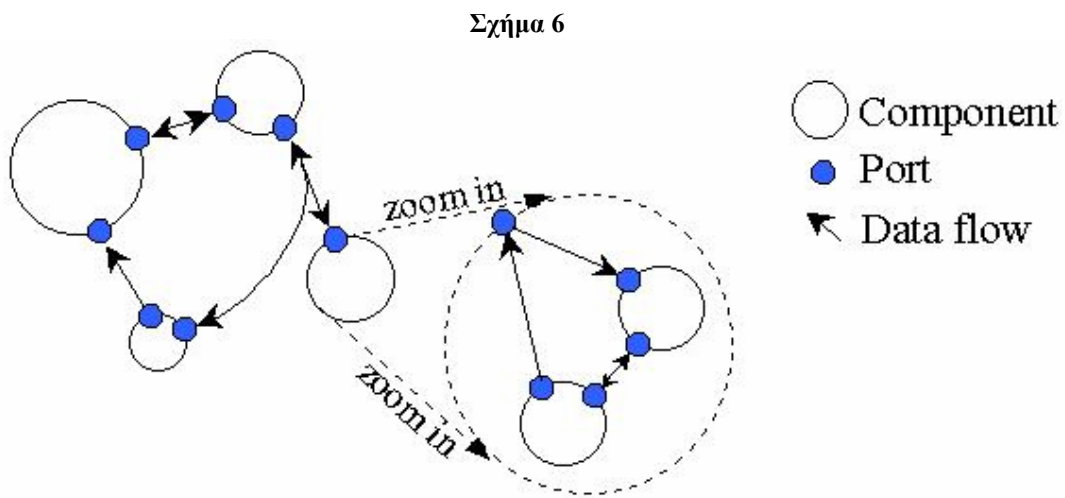
7.1.1 Autonomous Component Architecture (ACA).

Στην ACA βασική οντότητα είναι το συστατικό. Κάθε συστατικό περιλαμβάνει ένα ή περισσότερα σημεία τέλους (end points) που ονομάζονται θύρες (ports). Κάθε εφαρμογή θεωρείται μία σύνθεση συστατικών, όπως φαίνεται στο παρακάτω Σχήμα 5. Τα συστατικά στην j-sim είναι αόριστα συνδεδεμένα, επικοινωνούν το ένα με το άλλο "καλωδιώνοντας" τις θύρες τους και είναι συνδεδεμένα στις συμβάσεις (contracts). Οι συμβάσεις διευκρινίζουν την αιτιότητα των στοιχείων που στέλνονται/λαμβάνονται μεταξύ των συστατικών, αλλά δεν διευκρινίζουν ποια συστατικά συμμετέχουν στην επικοινωνία. Ένα άμεσο πλεονέκτημα των συστατικών είναι πως μπορούν να αναπτυχθούν ανεξάρτητα και να ενσωματωθούν αργότερα στις εφαρμογές.



Σχήμα 5

Ένα συστατικό μπορεί να είναι ένα σύνθετο συστατικό. Δηλαδή μπορεί να αποτελείται από διάφορα εσωτερικά συστατικά. Στα πλαίσια της προσομοίωσης δικτύων, ένα δίκτυο είναι ένα σύνθετο συστατικό και αποτελείται (στον χρόνο εκτέλεσης) από τους κόμβους, τις συνδέσεις, τα πρωτόκολλα, και τις κατάλληλες ενότητες. Το Σχήμα 6 απεικονίζει την έννοια των σύνθετων συστατικών.



Σχήμα 6

7.1.2 Script και J-sim.

Η j-sim είναι ένα διπλό-γλωσσικό περιβάλλον: Η Java χρησιμοποιείται για να υλοποιήσει όλες τις κλάσεις και η Tcl τη χρησιμοποιεί ως συνδετική γλώσσα script για τη δημιουργία, τη διαμόρφωση, ή/και την προσομοίωση δικτύων ελέγχου στο χρόνο εκτέλεσης. Εκτός μερικών από τις τυποποιημένες εντολές της Tcl, η j-sim χρησιμοποιεί επίσης δύο σύνολα εκτεταμένων εντολών Tcl :



(1) Tcl /Java: καθορίζει την επέκταση της Java σε Tcl και είναι η "γέφυρα" μεταξύ τους. Οι εντολές Tcl /Java μπορούν να δημιουργήσουν και να έχουν πρόσβαση σε οποιαδήποτε αντικείμενα της Java μέσα από το περιβάλλον της Tcl.

(2) Εντολές συστημάτων RUV: Η έννοια των σύνθετων συστατικών σε j-sim καθιστά πιθανό να οργανώσει τα συστατικά σε μια συστατική ιεραρχία. Η ιεραρχία είναι πολύ παρόμοια με ένα σύστημα αρχείων σε ένα σύγχρονο λειτουργικό σύστημα. Για να διευκολυνθεί η διαμόρφωση της προσομοίωσης δικτύων, έχει αναπτυχθεί ένα σύνολο εικονικών (RUnTime Virtual - RUV) εντολών συστημάτων χρόνου εκτέλεσης για να χειριστεί ο προγραμματιστής τη συστατική ιεραρχία με τον ίδιο τρόπο όπως διάφορες εντολές του λειτουργικού συστήματος Unix κάνουν στο σύστημα αρχείων.

7.1.3 Συστατικά της J-sim.

Συστατικό MobilityModel

Το συστατικό MobilityModel μιμείται τη κίνηση του ασύρματου κόμβου σ' ένα ad-hoc δίκτυο. Δύο πρότυπα κινητικότητας μπορούν να εφαρμοστούν: Το Random Waypoint πρότυπο κινητικότητας και το πρότυπο που βασίζεται σε καθορισμένη τροχιά. Εάν μια τροχιά έχει εγκατασταθεί, το συστατικό MobilityModel αυτόματα μιμείται τη μετακίνηση κόμβων σύμφωνα μ' αυτήν, σε διαφορετική περίπτωση χρησιμοποιείται το Random Waypoint πρότυπο κινητικότητας.

Συστατικό NodePositionTracker

Αυτό το συστατικό διαιρεί το επίπεδο της προσωμοιούμενης περιοχής σε πολλαπλάσιες υποπεριοχές. Το MobilityModel κάθε κινητού κόμβου επιστρέφει περιοδικά τη θέση του στο συστατικό NodePositionTracker. Όταν το συγκεκριμένο συστατικό NodePositionTracker λάβει αίτημα από έναν γειτονικό κόμβο από το τμήμα καναλιών, το NodePositionTracker ανακαλύπτει αρχικά σε ποια υποπεριοχή ο κόμβος αποστολής βρίσκεται και επιστρέφει το αναγνωριστικό pid όλων των κινητών κόμβων που βρίσκονται αυτήν την χρονική περίοδο σε όλες τις γειτονικές υποπεριοχές.

Συστατικό FileComponent

Αυτό το συστατικό γράφει σε αρχείο τα εισερχόμενα δεδομένα από τους κόμβους. Δηλαδή στην περίπτωση της διπλωματικής τις συντεταγμένες των θέσεων των κόμβων.

Συστατικό Plotter

Με το συστατικό αυτό γίνεται γραφική έξοδος της κίνησης των κόμβων.



7.1.4 J-sim και RWP.

Το Random Waypoint λειτουργεί ως εξής: μετά από την εκκίνηση του συστατικού MobilityModel αρχίζει να επιλέγει τυχαία έναν προορισμό στον οποίο θα μετακινηθεί ο κόμβος (ο προορισμός πρέπει να είναι μέσα στην περιοχή που έχει οριστεί για την κίνηση και έχει προηγουμένως διευκρινιστεί στη λειτουργία setTopologyParameters). Η ταχύτητα κίνησης είναι σταθερή και δεν μπορεί να είναι μεγαλύτερη από την ανώτατη ταχύτητα που είναι ήδη καθορισμένη στη λειτουργία setPosition. Κατά συνέπεια ο χρόνος που χρειάζεται ο κινητός κόμβος για να κινηθεί από την τρέχουσα θέση στον προορισμό του υπολογίζεται από τον τύπο:

$$\text{changeDestinationPeriod} = \text{dist} / \text{speed},$$

όπου dist είναι η απόσταση μεταξύ της τρέχουσας θέσης και του προορισμού. Στη συνέχεια προσδιορίζεται τυχαία ένας νέος προορισμός για την κίνηση του κόμβου και η προηγούμενη διαδικασία επαναλαμβάνεται.

7.1.5 Εντολές στο αρχείο προσομοίωσης.

attach_simulator

Συνδέει έναν χρόνο εκτέλεσης προσομοίωσης με τα συστατικά.

ΣΥΝΟΨΗ

```
attach_simulator ?-aq? ?<#threads allowed> |event? <path1> ?<path2>...?
```

ΕΠΙΛΟΓΗ

- a ταιριάζει όλα τα κρυμμένα συστατικά (και τις θύρες εάν το -p διευκρινίζεται).
- q αγνοεί άχρηστες προειδοποιήσεις όπως η "άκυρη διαδρομή".
- event χρησιμοποιεί εκτελέσεις προσομοίωσης διαδοχικών συμβάντων παρά το αντίστοιχο σε πραγματικό χρόνο.

ΠΕΡΙΓΡΑΦΗ

Η εντολή συνδέει έναν χρόνο εκτέλεσης προσομοίωσης προεπιλογής με τα συστατικά που διευκρινίζονται στη διαδρομή. Εάν ο αριθμός των νημάτων που επιτρέπονται δεν διευκρινίζεται, ο χρόνος εκτέλεσης δημιουργεί όσο το δυνατόν περισσότερα νήματα είναι απαραίτητο.



cd

Αλλάζει τον τρέχοντα κατάλογο εργασίας.

ΣΥΝΟΨΗ

cd? ?-aq? ?<path>?

ΕΠΙΛΟΓΗ

- a ταιριάζει όλα τα κρυμμένα συστατικά (και τις θύρες εάν το -p διευκρινίζεται).
- q αγνοεί άχρηστες προειδοποιήσεις όπως η "άκυρη διαδρομή".

ΠΕΡΙΓΡΑΦΗ

Εάν πολλαπλά συστατικά αντιστοιχούνται στην διαδρομή, το πρώτο συστατικό που αντιστοιχείται γίνεται ο τρέχων κατάλογος εργασίας. Κανένας συγκεκριμένος κανόνας δεν καθορίζεται στον καθορισμό της πρώτης αντιστοιχίας.

connect

Συνδέει συστατικά/θύρες.

ΣΥΝΟΨΗ

connect ?-acpqs? <source_path>... -to|-and <destination_path>...

ΕΠΙΛΟΓΗ

- a ταιριάζει όλα τα κρυμμένα συστατικά (και τις θύρες εάν το -p διευκρινίζεται).
- and οργανώνει μια αμφίδρομη σύνδεση.
- c δημιουργεί τις θύρες εάν δεν υπάρχουν.
- p ταιριάζει τις θύρες.
- q αγνοεί άχρηστες προειδοποιήσεις όπως η "άκυρη διαδρομή".
- s οργανώνει μια κοινή αμφίδρομη σύνδεση (με το -and).
- to οργανώνει μια μονόδρομη σύνδεση.

ΠΕΡΙΓΡΑΦΗ

Η εντολή συνδέει τα συστατικά/θύρες στην αρχική διαδρομή και τα συστατικά/θύρες στη διαδρομή προορισμού. Εάν διευκρινίζονται τα συστατικά, η εντολή βρίσκει αρχικά τις διαθέσιμες θύρες στα συστατικά προκειμένου να οργανωθεί η σύνδεση. Μια αμφίδρομη σύνδεση είτε αποτελείται από δύο μονόδρομες συνδέσεις είτε μια κοινή σύνδεση. Εξ ορισμού, η εντολή δημιουργεί δύο μονόδρομες συνδέσεις σε απάντηση της επιλογής

-and.



Οι αρχικές συνδέσεις (εάν υπάρχουν) των θυρών θα ενωθούν όλες μαζί μετά την εκτέλεση της εντολής connect.

mkdir

Δημιουργεί τα συστατικά/θύρες. Αυτή η εντολή επιστρέφει την αναφορά δημιουργίας του αντίστοιχου αντικειμένου.

ΣΥΝΟΨΗ

mkdir ?-αφ? <Java_class_name> <path1> ?<path2>...?

mkdir ?-αφ? <Java_object_ref> <path1> ?<path2>...?

mkdir ?-αφ? <port_path1> ?<port_path2>...?

mkdir ?-αφ? <component_path>

ΕΠΙΛΟΓΗ

- a ταιριάζει όλα τα κρυμμένα συστατικά (και τις θύρες εάν το -p διευκρινίζεται).
- q αγνοεί άχρηστες προειδοποιήσεις όπως η "άκυρη διαδρομή".

ΠΕΡΙΓΡΑΦΗ

Η εντολή δημιουργεί ένα ή περισσότερα συστατικά/θύρες στη διαδρομή που διευκρινίζεται. Εάν το όνομα κατηγορίας ή το αντικείμενο της Java δεν είναι ένα συστατικό/θύρα, ένα συστατικό περιτυλίγματος (*drcl.comp.WrapperComponent*) δημιουργείται για να ενθυλακώσει το αντικείμενο. Η τέταρτη μορφή χρησιμοποιεί την προεπιλεγμένη κλάση για να δημιουργήσει το συστατικό στη συγκεκριμένη διαδρομή. Είναι ισοδύναμο με το "*mkdir* ?-αφ? <default_class> <component_path>".

rt

Έχει πρόσβαση στο χρόνο εκτέλεσης και τις πληροφορίες του.

ΣΥΝΟΨΗ

rt ?-αφ? <path> ?<method> ?<arg1>...??

ΕΠΙΛΟΓΗ

- a ταιριάζει όλα τα κρυμμένα συστατικά (και τις θύρες εάν το -p διευκρινίζεται).
- p ταιριάζει τις θύρες.
- q αγνοεί άχρηστες προειδοποιήσεις όπως η "άκυρη διαδρομή".

ΠΕΡΙΓΡΑΦΗ

Η εντολή ανακτά το χρόνο εκτέλεσης πίσω από το συστατικό που διευκρινίζεται, και κάνει μια κλήση μεθόδου εάν μια μέθοδος διευκρινίζεται.



run

Εκκινεί τα συστατικά και όλα τα υποσυστατικά που βρίσκονται μέσα σε αυτά.

ΣΥΝΟΨΗ

run ?-arq? <path1> ?<path2>...?

ΕΠΙΛΟΓΗ

- a ταιριάζει όλα τα κρυμμένα συστατικά (και τις θύρες εάν το -p διευκρινίζεται).
- p ταιριάζει τις θύρες.
- q αγνοεί άχρηστες προειδοποιήσεις όπως η "άκυρη διαδρομή".

stop

Σταματά την εκτέλεση των συστατικών και όλων των υποσυστατικών που βρίσκονται μέσα σε αυτά.

ΣΥΝΟΨΗ

Stop? ?-arq? <path1> ?<path2>...?

ΕΠΙΛΟΓΗ

- a ταιριάζει όλα τα κρυμμένα συστατικά (και τις θύρες εάν το -p διευκρινίζεται).
- p ταιριάζει τις θύρες.
- q αγνοεί άχρηστες προειδοποιήσεις όπως η "άκυρη διαδρομή".



7.2 Προσομοιωτής (Κώδικας για κίνηση 100 κόμβων – Αρχείο *prosomiotis_RWP_100_komvon*).

```
//EAP_pls_de_ _Kostas_Lyberis
//Prosomiotis Random Way Point (RWP) se j-sim gia kinisi 100 komvon

//Prosomiotis_RWP_100_komvon

puts mobtest

set minx 0.0
set maxx 100.0
set miny 0.0
set maxy 100.0
set maxz 0.0
set minz 0.0
set dx 1.0
set dy 1.0
set dz 0.0
set maxSpeed 2.0
cd [mkdir mobtest]

for {set i 0} {$i<=99} {incr i} {
    mkdir drcl.inet.Node node$i
    cd node$i
    set mob$i [mkdir drcl.inet.mac.MobilityModel mobility]
    set convert$i [mkdir drcl.inet.mac.PositionReportConvert converter]
    set plot [mkdir drcl.comp.tool.Plotter plotter]

    set file [mkdir drcl.comp.io.FileComponent .file]
    $file open "sintetagmenes.plot";
    connect -c $plot/.output@ -to $file/in@

    cd ..
}
}
```



\$mob0 setTopologyParameters \$maxx \$maxy \$maxz \$minx \$miny \$minz \$dx \$dy \$dz

\$mob0 setPosition \$maxSpeed 0 0 0

\$mob1 setTopologyParameters \$maxx \$maxy \$maxz \$minx \$miny \$minz \$dx \$dy \$dz

\$mob1 setPosition \$maxSpeed 0 0 0

\$mob2 setTopologyParameters \$maxx \$maxy \$maxz \$minx \$miny \$minz \$dx \$dy \$dz

\$mob2 setPosition \$maxSpeed 0 0 0

\$mob3 setTopologyParameters \$maxx \$maxy \$maxz \$minx \$miny \$minz \$dx \$dy \$dz

\$mob3 setPosition \$maxSpeed 0 0 0

....

....

\$mob95 setTopologyParameters \$maxx \$maxy \$maxz \$minx \$miny \$minz \$dx \$dy \$dz

\$mob95 setPosition \$maxSpeed 0 0 0

\$mob96 setTopologyParameters \$maxx \$maxy \$maxz \$minx \$miny \$minz \$dx \$dy \$dz

\$mob96 setPosition \$maxSpeed 0 0 0

\$mob97 setTopologyParameters \$maxx \$maxy \$maxz \$minx \$miny \$minz \$dx \$dy \$dz

\$mob97 setPosition \$maxSpeed 0 0 0

\$mob98 setTopologyParameters \$maxx \$maxy \$maxz \$minx \$miny \$minz \$dx \$dy \$dz

\$mob98 setPosition \$maxSpeed 0 0 0

\$mob99 setTopologyParameters \$maxx \$maxy \$maxz \$minx \$miny \$minz \$dx \$dy \$dz

\$mob99 setPosition \$maxSpeed 0 0 0

\$plot setXRange 0 \$minx \$maxx

\$plot setYRange 0 \$miny \$maxy



```
connect $mob0/.query@ -and $convert0/query@  
connect -c $convert0/out@ -to $plot/0@0
```

```
connect $mob1/.query@ -and $convert1/query@  
connect -c $convert1/out@ -to $plot/1@0
```

```
connect $mob2/.query@ -and $convert2/query@  
connect -c $convert1/out@ -to $plot/2@0
```

```
....  
....
```

```
connect $mob97/.query@ -and $convert97/query@  
connect -c $convert97/out@ -to $plot/97@0
```

```
connect $mob98/.query@ -and $convert98/query@  
connect -c $convert98/out@ -to $plot/98@0
```

```
connect $mob99/.query@ -and $convert99/query@  
connect -c $convert99/out@ -to $plot/99@0
```

```
set sim [attach_simulator .]
```

```
run .
```

```
rt . stopAt 1000
```



7.3 Παρουσίαση του αρχείου *sintetagmenes.txt* .

Παρουσιάζεται εδώ ένα τμήμα του αρχείου *sintetagmenes.txt*, το οποίο έχει συνολικά 99900 εγγραφές. Η μορφή και οι εγγραφές του αρχείου είναι όπως και του τμήματος που φαίνεται εδώ. Κάθε εκατοντάδα εγγραφών αποτελούν ένα στιγμιότυπο της κίνησης των 100 κόμβων που έχουν οριστεί στον προσομοιωτή.

```
### NEW PLOT--0--<No Title>--'- Time -'  
##### NEW SET--0--39--???  
m--0--39--0.2975557890737861--9.766519948288604E-4--???  
##### NEW SET--0--38--???  
m--0--38--1.137685190686975--0.358297666244716--???  
##### NEW SET--0--37--???  
m--0--37--1.4743983169853416--1.0034844351867074--???  
##### NEW SET--0--36--???  
m--0--36--0.5087524426528307--0.7644423208475746--???  
##### NEW SET--0--35--???  
m--0--35--0.8605713389365158--1.0068099028356732--???  
p--0--99--0.3422008502095312--1.461065368589111--???  
##### NEW SET--0--34--???  
m--0--34--0.689112569715524--0.1994883739391733--???  
p--0--98--0.4228998918893514--0.19805995846955504--???  
##### NEW SET--0--33--???  
m--0--33--0.2472763241058414--0.22292226157642697--???  
p--0--97--0.5832491114663666--0.39141236800362217--???  
##### NEW SET--0--32--???  
m--0--32--0.5665200837630159--0.8248334596538551--???  
p--0--96--0.09367843940762419--0.17721958722124093--???  
p--0--31--0.22377464428352403--0.09850954899523016--???  
p--0--95--0.5298445441698746--0.4799564638901354--???  
##### NEW SET--0--30--???  
m--0--30--0.4044194478162925--0.43627383179668155--???  
p--0--94--0.25633413665367216--0.133606542870451--???  
##### NEW SET--0--93--???  
m--0--93--0.0773633100490461--0.18898372158023774--???  
##### NEW SET--0--92--???  
m--0--92--0.9900842494638004--0.563280522354205--???  
##### NEW SET--0--91--???  
m--0--91--0.3501062558705328--0.34601677430934086--???  
##### NEW SET--0--90--???  
m--0--90--0.3226046547896875--0.8059927887329463--???  
p--0--29--0.04704603135101521--0.009801595380282316--???  
p--0--28--0.8148684516347077--1.2446328205906705--???  
p--0--27--0.37458042369898326--0.35328208750304035--???  
p--0--26--0.668276311490136--1.0236097335122283--???
```



8 Παράρτημα Β: Ο αλγόριθμος SIS σε κώδικα Java.

8.1 Ο Αλγόριθμος SIS για ακίνητους κόμβους (αρχείο *fix_sis_epidemic.java*).

```
//EAP_pls_de__Kostas_Lyberis  
//Algorithmos SIS gia statherous komvous &' ena epipedo molinsis  
  
//fix_sis-epidemic  
  
import java.util.Random;  
import java.io.*;  
  
public class fix_sis_epidemic {  
  
    static final int SUSCEPTIBLE = 0;  
    static final int INFECTED = 1;  
  
    static final String COMMA = "," ;  
    static final String FILENAME0 = "fix0.csv"; //Oi igiis komvoi  
    static final String FILENAME1 = "fix1.csv"; //Oi molismenoi komvoi  
    static final String FILENAME2 = "fix2.csv"; //Oloi oi komvoi  
    static final String FILENAME3 = "fix3.csv"; //Mono ta sunola igion-  
                                                molismenon  
  
    static double A[][];  
    static int S[];  
    static int M[];  
    static int N=0;  
    static int NI=0;  
    static int startcount=0;  
    static int nN;  
    static String filename="";  
    static String line = "";  
    static double b=0.0;  
    static double d=0.0;  
    static double r=0.0;
```




```
public static void main( String[] args) {
    if (args.length != 6) {
        System.err.println ("Usage: java fix_sis_epidemic Nodes startcount
inputFilename b d radius");
        return;
    }

    N = Integer.parseInt( args[0] );
    startcount = Integer.parseInt( args[1] );
    filename = args[2];
    b = Double.parseDouble( args[3] );
    d = Double.parseDouble( args[4] );
    r = Double.parseDouble( args[5] );
    A = new double[N][2];
    M = new int[3];
    initS();
    readFile();
    results();
    return;
}
```

```
static void initS() { // arxikopoei ton pinaka katastasis ton komvon
    int i;
    NI = (int)(N /10);

    S= new int[N];

    for(i=0 ; i<NI ; i++) S[i]=INFECTED;
    for(i=NI ; i<N ; i++) S[i]=SUSCEPTIBLE;
}
```



```
static void mySplit() { // διαvazei mia grammi tou arxeiou
                        // sintetagmenon
    int node, stop, start = 0;
    double x,y;
    String s;

    stop = line.indexOf("--", start);
    s = line.substring(start, stop); // Vriskei to stoixeio ths protis stilis mias
    start = stop + 2; // Vriskei to stoixeio ths pemptis stilis mias
                        // grammis tou arxeiou sintetagmenes.plot
                        // grammis tou arxeiou

    stop = line.indexOf("--", start);
    s = line.substring(start, stop); // Vriskei to stoixeio ths pemptis stilis mias
    start = stop + 2; // grammis tou arxeiou

    stop = line.indexOf("--", start);
    s = line.substring(start, stop); // Entopizei ton komvo tis grammis sto
    node = Integer.parseInt(s); // arxeio
    start = stop + 2;

    stop = line.indexOf("--",start);
    s = line.substring(start,stop); // Entopizei ti sintetagmeni x gia enan
    x = Double.parseDouble(s); // komvo sti grammi
    start = stop + 2;

    stop = line.indexOf("--",start);
    s = line.substring(start,stop); // Entopizei ti sintetagmeni y gia enan
    y = Double.parseDouble(s); // komvo sti grammi

    A[node][0] = x; // Thetei tis sintetagmenes pou diavazei apo
    A[node][1] = y; // to arxeio ston pinaka sintetagmenon
}
}
```



```
static void readFile() {                                     // διαvazei to arxeio sintetagmenon gia
                                                           kathe xroniki stigma

    try {
        FileReader f = new FileReader(filename);
        BufferedReader b = new BufferedReader(f);

        BufferedWriter outfile[] = new BufferedWriter[4] ;
        outfile[0] = new BufferedWriter(new FileWriter(FILENAME0));
        outfile[1] = new BufferedWriter(new FileWriter(FILENAME1));
        outfile[2] = new BufferedWriter(new FileWriter(FILENAME2));
        outfile[3] = new BufferedWriter(new FileWriter(FILENAME3));

        line = b.readLine();
        int lineCounter = 0;
        int x=0;
        String s="";

        while(line != null) {
            if(line.charAt(0)!='#') {
                mySplit();
                lineCounter++;
                if(lineCounter%N==0) {
                    nN = lineCounter / N ;
                    for(int e=0; e<1000; e++) //loop 1000
                                                           epanalipseon tou
                                                           algorithmou
                    {
                        if(nN==startcount) { //Statheropoiisi
                                                           stigmatipou
                                                           sintetagmenon
                            myCalc();

                            M[0] = 0;
                            M[1] = 0;

                            //Write Step number
```



```
        for(int k=0; k<4;k++) {
            outfile[k].write(""+nN);
        }
        for (int i=0 ; i<S.length ; i++){
            x=S[i];
            M[x]++;
            s = COMMA + x;
            outfile[2].write(s);
            outfile[x].write(COMMA + i);
        }

        //Write Metrites

        outfile[2].write(COMMA+M[0]+COMMA+M[1]);

        outfile[3].write(COMMA+M[0]+COMMA+M[1]);

        //Allagh se ola ta arxia

        for(int k=0 ; k<4 ; k++) {
            outfile[k].newLine();
        }
    }
}

line = b.readLine();
}
b.close();
for(int k=0 ; k<4 ; k++) {
    outfile[k].flush();
    outfile[k].close();
}
}
```



```
catch(IOException e) {
    System.err.println("Exception caught: " + e.getMessage());
}
}

static void results() {                                // apotelesmata molinsis komvon
double p;
int c=0;

    for(int i=0 ; i<N ; i++) {
        if (S[i]==1) c++;
    }

    p = (double)c/N;
    System.out.println ("\n\nnb=" + b + " d = " + d + " r= " +r);
    System.out.println ("N=" + N +
                                                                    "\nc=" + c + " p="+p);
}
}
```



```
static void myCalc() { // ilopoiisi epidimikou algorithmou

    System.err.println("\nNow Calc " +nN);

    double ra;
    double sqrt1=0;
    int i, k;
    int nbs[];

    nbs = new int[N];

    for(int j=0 ; j<N ; j++) {
        System.out.println("j = " + j + " nN = " +nN);
        if(S[j] == 1) { // Elegxos an enas komvos tha ginei igieis 1 se 0
            ra = Math.random();
            if(ra < d){
                S[j] = 0;
                System.out.println(nN + " iasi 1->0: kombou " + j + " ra="
+ra + " d= "+d);
            }
            ra = Math.random();
            if(ra < d){
                S[j] = 0;
                System.out.println(nN + " iasi 1->0: kombou " + j +
" ra=" +ra + " d= "+d);
            }
        }
    }

    k = 0;

    for(i=0 ; i<N ; i++) nbs[i]=0;

    for(i=0 ; i<N ; i++) {
        if (i==j) continue;
        sqrt1 = Math.sqrt( Math.pow((A[j][0] - A[i][0]),2) + Math.pow((A[j][1] - A[i][1]),2)
);
        if(sqrt1 <= r){
```



```
nbs[k] = i;
System.out.println("nbs[" +k+"]=" +i + " molinsi "+S[i]);
k++;
System.out.println("geitonikos komvos einai o " +i);
}
}

System.out.println("synolo geitonon k=" +k);

for(i=0 ; i<k ; i++){
    if(S[nbs[i]] == 1){
        ra = Math.random();
        if(ra < b){
            S[j] = 1;
            System.out.println("mpike S[" +j+"]=1");
        }
    }
}

}

}

}
```



8.2 Ο Αλγόριθμος SIS για κινούμενους κόμβους (αρχείο *mob_sis_epidemic.java*).

```
//EAP_pls_de__Kostas_Lyberis  
//Algorithmos SIS gia kinoumenous komvous & ena epipedo molinsis  
  
//mob_sis-epidemic  
  
import java.util.Random;  
import java.io.*;  
  
public class mob_sis_epidemic {  
  
    static final int SUSCEPTIBLE = 0;  
    static final int INFECTED = 1;  
  
    static final String COMMA = "," ;  
    static final String FILENAME0 = "mob0.csv";           //Oi igiis komvoi  
    static final String FILENAME1 = "mob1.csv";           //Oi molismenoi komvoi  
    static final String FILENAME2 = "mob2.csv";           //Oloi oi komvoi  
    static final String FILENAME3 = "mob3.csv";           //Mono ta sunola igion  
                                                         -molismenon  
  
    static double A[][];  
    static int S[];  
    static int M[];  
    static int N=0;  
    static int NI=0;  
    static int startcount=0;  
    static int nN;  
    static String filename="";  
    static String line = "";  
    static double b=0.0;  
    static double d=0.0;  
    static double r=0.0;
```




```
public static void main( String[] args) {
    if (args.length != 6) {
        System.err.println ("Usage: java mob_sis_epidemic Nodes startcount
inputFilename b d radius");
        return;
    }

    N = Integer.parseInt( args[0] );
    startcount = Integer.parseInt( args[1] );
    filename = args[2];
    b = Double.parseDouble( args[3] );
    d = Double.parseDouble( args[4] );
    r = Double.parseDouble( args[5] );
    A = new double[N][2];
    M = new int[3];
    initS();
    readFile();
    results();
    return;
}

static void initS() { // αρχικοποιεί τον πίνακα καταστάσι των κόμβων
    int i;
    NI = (int)(N /10);

    S= new int[N];

    for(i=0 ; i<NI ; i++) S[i]=INFECTED;
    for(i=NI ; i<N ; i++) S[i]=SUSCEPTIBLE;
}
```



```
static void mySplit() { // διαvazei mia grammi tou arxeiou sintetagmenon
    int node, stop, start = 0;
    double x,y;
    String s;

    stop = line.indexOf("--", start);
    s = line.substring(start, stop); // Vriskei to stoixeio ths protis stilis mias grammis
    // tou arxeiou sintetagmenes.plot
    start = stop + 2;

    stop = line.indexOf("--", start);
    s = line.substring(start, stop); // Vriskei to stoixeio ths pemptis stilis mias
    // grammis tou arxeiou
    start = stop + 2;

    stop = line.indexOf("--", start);
    s = line.substring(start, stop); // Entopizei ton komvo tis grammis sto arxeio
    node = Integer.parseInt(s);
    start = stop + 2;

    stop = line.indexOf("--",start);
    s = line.substring(start,stop); // Entopizei ti sintetagmeni x gia enan komvo sti
    // grammi
    x = Double.parseDouble(s);
    start = stop + 2;

    stop = line.indexOf("--",start);
    s = line.substring(start,stop); // Entopizei ti sintetagmeni y gia enan komvo sti
    // grammi
    y = Double.parseDouble(s);

    A[node][0] = x; // Thetei tis sintetagmenes pou diavazei apo to
    // arxeio ston pinaka sintetagmenon
    A[node][1] = y;
}
```



```
static void readFile() { // διαβάζει το αρχείο sintetagmenon gia kathe xroniki stigmi
    try {
        FileReader f = new FileReader(filename);
        BufferedReader b = new BufferedReader(f);

        BufferedWriter outfile[] = new BufferedWriter[4] ;
        outfile[0] = new BufferedWriter(new FileWriter(FILENAME0));
        outfile[1] = new BufferedWriter(new FileWriter(FILENAME1));
        outfile[2] = new BufferedWriter(new FileWriter(FILENAME2));
        outfile[3] = new BufferedWriter(new FileWriter(FILENAME3));

        line = b.readLine();
        int lineCounter = 0;
        int x=0;
        String s="";

        while(line != null) {
            if(line.charAt(0)!='#') {
                mySplit();
                lineCounter++;
                if(lineCounter%N==0) {
                    nN = lineCounter / N ;
                    if(nN>=startcount) {
                        myCalc();

                        M[0] = 0;
                        M[1] = 0;

                        //Write Step number

                        for(int k=0; k<4;k++) {
                            outfile[k].write(""+nN);
                        }
                        for (int i=0 ; i<S.length ; i++){
                            x=S[i];
                            M[x]++;
                        }
                    }
                }
            }
        }
    }
}
```



```
s = COMMA + x;
outfile[2].write(s);
outfile[x].write(COMMA + i);
}

//Write Metrites

outfile[2].write(COMMA+M[0]+COMMA+M[1]);
outfile[3].write(COMMA+M[0]+COMMA+M[1]);

//Allagi grammis se ola ta arxeia

for(int k=0 ; k<4 ; k++) {
    outfile[k].newLine();
}
}
}
line = b.readLine();
}
b.close();
for(int k=0 ; k<4 ; k++) {
    outfile[k].flush();
    outfile[k].close();
}
}
catch(IOException e) {
    System.err.println("Exception caught: " + e.getMessage());
}
}
```



```
static void results() {                                     // apotelesmata molinsis komvon
double p;
int c=0;

    for(int i=0 ; i<N ; i++) {
        if (S[i]==1) c++;
    }

    p = (double)c/N;
    System.out.println ("\n\nnb=" + b + " d = " + d + " r= " +r);
    System.out.println ("N=" + N +
        "\nc=" + c + " p="+p);
}
```



```
static void myCalc() {                                     // ilopoiisi epidimikou algorithmou

    System.err.println("\nNow Calc " +nN);

    double ra;
    double sqrt1=0;
    int i, k;
    int nbs[];

    nbs = new int[N];

    for(int j=0 ; j<N ; j++) {
        System.out.println("j = " + j + " nN = " +nN);
        if(S[j] == 1) {                                     // Elegxos an enas komvos tha ginei igieis 1 se 0
            ra = Math.random();
            if(ra < d){
                S[j] = 0;
                System.out.println(nN + " iasi 1->0: kombou " + j + " ra="
+ra + " d= "+d);
            }
            ra = Math.random();
            if(ra < d){
                S[j] = 0;
                System.out.println(nN + " iasi 1->0: kombou " + j + " ra=" +ra + "
d= "+d);
            }
        }

        k = 0;

        for(i=0 ; i<N ; i++) nbs[i]=0;

        for(i=0 ; i<N ; i++) {
            if (i==j) continue;
            sqrt1 = Math.sqrt( Math.pow((A[j][0] - A[i][0]),2) +
Math.pow((A[j][1] - A[i][1]),2) );
            if(sqrt1 <= r){
```



```
nbs[k] = i;
System.out.println("nbs[" +k+"]=" +i + "   molinsi "+S[i]);
k++;
System.out.println("geitonikos komvos einai o " +i);
}
}

System.out.println("synolo geitonon k=" +k);

for(i=0 ; i<k ; i++){
    if(S[nbs[i]] == 1){
        ra = Math.random();
        if(ra < b){
            S[j] = 1;
            System.out.println("mpike S[" +j+"]=1");
        }
    }
}

}

}

}
```



9 Παράρτημα Γ: Παρουσίαση των αρχείων υλοποίησης των μοντέλων της Διατριβής.

Παρατίθεται στο παράρτημα αυτό η βασική φιλοσοφία προγραμματιστικής υλοποίησης των διαφόρων μοντέλων για να βοηθηθεί ο αναγνώστης στην κατανόηση του τρόπου λειτουργίας του κώδικα και της χρήσης των αρχείων που παρουσιάζονται στην παρούσα εργασία.

9.1 Προγραμματιστική φιλοσοφία για τα μοντέλα των ακίνητων κόμβων.

Το πρόγραμμα `fix_sis_epidemic.java` που υλοποιεί τον αλγόριθμο SIS για ακίνητους κόμβους έχει φτιαχτεί με τις εξής προδιαγραφές: Δέχεται έξι ορίσματα, που κατά σειρά είναι:

1. Το πλήθος των κόμβων του δικτύου.
2. Το set συντεταγμένων που θέλουμε να ξεκινήσουμε την υλοποίηση του αλγορίθμου.
3. Το αρχείο με τις συντεταγμένες των κόμβων (`sintetagmenes.txt` για τα εξεταζόμενα σενάρια).
4. Ο ρυθμός μόλυνσης β των κόμβων (θεωρείται σταθερός για όλες τις μεταβάσεις καταστάσεων σε κάθε εξεταζόμενη περίπτωση).
5. Ο ρυθμός ίασης δ (θεωρείται σταθερός για όλες τις μεταβάσεις καταστάσεων σε κάθε εξεταζόμενη περίπτωση).
6. Η ακτίνα επικοινωνίας (καθορίζει και το μέγεθος της γειτονιάς) του κόμβου.

Το πλήθος των κόμβων (όρισμα 1) έχει δηλωθεί σταθερό καθ' όλη τη διάρκεια της συγκεκριμένης διατριβής, στους εκατό κόμβους. Αποτέλεσμα της παραπάνω δήλωσης είναι, πως από το αρχείο συντεταγμένων (όρισμα 3) διαβάζονται εκατοντάδες γραμμών κάθε φορά, οι οποίες αντιστοιχούν σε κάθε βήμα της προσομοίωσης της κίνησης των κόμβων, σε διαφορετικό στιγμιότυπο των θέσεών τους. Για να έχουμε, ένα συγκεκριμένο στιγμιότυπο των κόμβων (ακίνησια, συγκεκριμένη μεταξύ των κόμβων σχετική θέση) στην περιοχή προσομοίωσης το επιλέγουμε από τον αντίστοιχο πίνακα συντεταγμένων μέσω του ορίσματος 2. Ο ρυθμός μόλυνσης των κόμβων έχει θεωρηθεί σταθερός για όλες τις μεταβάσεις καταστάσεων σε κάθε εξεταζόμενη περίπτωση, όπως και ο ρυθμός ίασης δ . Η ακτίνα επικοινωνίας ενός κόμβου παίρνει επίσης σταθερή τιμή ανά περίπτωση. Η περιοχή προσομοίωσης στην `j-sim` είχε τεθεί σαν τετράγωνο πλευράς 100 μονάδων.



Για ευκολία έχει δημιουργηθεί το αρχείο `run_fix_sis_epidemic.bat` για τη γρήγορη εκτέλεση του κώδικα. Το περιεχόμενο του αρχείου παρατίθεται στην παράγραφο 9.1.1.

Από τη στιγμή που έχουν καθοριστεί καλώς τα ορίσματα το πρόγραμμα `fix_sis_epidemic.java` εκτελεί τα βήματα:

- Διαβάζει με τη βοήθεια της κλάσης `readFile()` το αρχείο συντεταγμένων κατά μία εκατοντάδα θέσεων κάθε φορά και μέσω της κλάσης `mySplit()` θέτει τις συντεταγμένες των κόμβων για το αντίστοιχο στιγμότυπο στον πίνακα συντεταγμένων.
- Από το την τιμή του ορίσματος 2 ξεκινά να εφαρμόζει την κλάση `myCalc()`, που υλοποιεί τον επιδημικό αλγόριθμο του προγράμματος.
- Παράγει αρχεία εξόδου για την καλύτερη αξιοποίηση των αποτελεσμάτων της εφαρμογής του επιδημικού αλγορίθμου. Αυτά είναι:
 - Αρχείο `fix0.csv`. Περιέχει μόνο τους κόμβους σε κάθε βήμα (από 101 έως 999) που είναι υγιείς. Βασικό αρχείο καταγραφής της υγιούς κατάστασης των κόμβων.
 - Αρχείο `fix1.csv`. Περιέχει μόνο τους κόμβους σε κάθε βήμα (από 101 έως 999) που έχουν μολυνθεί (κατάσταση 1). Βασικό αρχείο καταγραφής της κατάστασης 1.
 - Αρχείο `fixt2.csv`. Περιέχει συνολικά τις καταστάσεις των 100 κόμβων σε κάθε βήμα (από 101 έως 999) και στο τέλος παρουσιάζεται αθροιστικά ο αριθμός ανά κατάσταση. Συγκεντρωτικό αρχείο στοιχείων.
 - Αρχείο `fix3.csv`. Περιέχει μόνο τα αθροιστικά αποτελέσματα καταστάσεων για τους εκατό κόμβους σε κάθε βήμα (από 101 έως 999). Βασικό αρχείο. Χρησιμοποιήθηκε για τη δημιουργία των γραφικών παραστάσεων της διατριβής.
 - Αρχείο `test_fix_sis_epidemic100.txt`. Περιέχει διάφορα μηνύματα και δείχνει με ακρίβεια το βήμα στο οποίο πραγματοποιήθηκε κάποια αλλαγή κατάστασης για τις τιμές όλων των παραμέτρων εκτέλεσης. Το αρχείο αυτό δεν είναι βασικό αρχείο εξόδου. Χρησιμοποιήθηκε κυρίως για τη διαδικασία ελέγχου του κώδικα.

9.1.1 Παρουσίαση του αρχείου `run_fix_sis_epidemic.bat`.

```
java fix_sis_epidemic N NI sintetagmenes.txt β, δ, r > test_fix_sis_epidemic100.txt  
pause
```

Όπου `N`, `NI`, `β`, `δ`, `r`, παίρνουν τις συγκεκριμένες κάθε φορά τιμές για την εκτέλεση του σεναρίου που εξετάζουμε.



9.2 Προγραμματιστική φιλοσοφία για τα μοντέλα των κινούμενων κόμβων

Το πρόγραμμα `mob_sis_epidemic.java` που υλοποιεί τον αλγόριθμο SIS για κινούμενους κόμβους έχει φτιαχτεί με τις εξής προδιαγραφές: Δέχεται έξι ορίσματα που κατά σειρά είναι

7. Το πλήθος N των κόμβων του δικτύου.
8. Το `set` συντεταγμένων N_I που θέλουμε να ξεκινήσουμε την υλοποίηση του αλγορίθμου.
9. Το αρχείο με τις συντεταγμένες των κόμβων (`sintetagmenes.txt` για τα εξεταζόμενα σενάρια).
10. Ο ρυθμός μόλυνσης β των κόμβων (θεωρείται σταθερός για όλες τις μεταβάσεις καταστάσεων σε κάθε εξεταζόμενη περίπτωση).
11. Ο ρυθμός ίασης δ (θεωρείται σταθερός για όλες τις μεταβάσεις καταστάσεων σε κάθε εξεταζόμενη περίπτωση).
12. Η ακτίνα επικοινωνίας r (καθορίζει και το μέγεθος της γειτονιάς) του κόμβου.

Το πλήθος των κόμβων (όρισμα 1) έχει δηλωθεί σταθερό καθ' όλη τη διάρκεια της συγκεκριμένης διατριβής, στους εκατό κόμβους. Αποτέλεσμα της παραπάνω δήλωσης είναι, πως από το αρχείο συντεταγμένων (όρισμα 3) διαβάζονται εκατοντάδες γραμμών κάθε φορά, οι οποίες αντιστοιχούν σε κάθε βήμα της προσομοίωσης της κίνησης των κόμβων, σε διαφορετικό στιγμιότυπο των θέσεών τους. Για να έχουμε, μέσω του μοντέλου κινητικότητας, επαρκή εξάπλωση των κόμβων στον χώρο προσομοίωσης, εξαιρούνται τα εκατό πρώτα στιγμιότυπα θέσεων (όρισμα 2) από τον αντίστοιχο πίνακα συντεταγμένων. Το 101° στιγμιότυπο της κίνησης των κόμβων είναι στην ουσία το σημείο χωρικής και χρονικής αφετηρίας υλοποίησης του επιδημικού αλγορίθμου. Ο ρυθμός μόλυνσης των κόμβων έχει θεωρηθεί σταθερός για όλες τις μεταβάσεις καταστάσεων σε κάθε εξεταζόμενη περίπτωση, όπως και ο ρυθμός ίασης δ . Η ακτίνα επικοινωνίας ενός κόμβου παίρνει επίσης σταθερή τιμή ανά περίπτωση. Η περιοχή προσομοίωσης στην `j-sim` είχε τεθεί σαν τετράγωνο πλευράς 100 μονάδων.

Για ευκολία έχει δημιουργηθεί το αρχείο `run_mob_sis_epidemic.bat` για τη γρήγορη εκτέλεση του κώδικα. Το περιεχόμενο του αρχείου παρατίθεται στην παράγραφο 9.2.1.

Από τη στιγμή που έχουν καθοριστεί καλώς τα ορίσματα το πρόγραμμα `mob_sis_epidemic.java` εκτελεί τα βήματα:



- Διαβάζει με τη βοήθεια της κλάσης `readFile()` το αρχείο συντεταγμένων κατά μία εκατοντάδα θέσεων κάθε φορά και μέσω της κλάσης `mySplit()` θέτει τις συντεταγμένες των κόμβων για το αντίστοιχο στιγμιότυπο στον πίνακα συντεταγμένων.
- Από το την τιμή του ορίσματος 2 ξεκινά να εφαρμόζει την κλάση `myCalc()`, που υλοποιεί τον επιδημικό αλγόριθμο του προγράμματος.
- Παράγει αρχεία εξόδου για την καλύτερη αξιοποίηση των αποτελεσμάτων της εφαρμογής του επιδημικού αλγορίθμου. Αυτά είναι:
 - Αρχείο `mob0.csv`. Περιέχει μόνο τους κόμβους σε κάθε βήμα (από 101 έως 999) που είναι υγιείς. Βασικό αρχείο καταγραφής της υγιούς κατάστασης των κόμβων.
 - Αρχείο `mob1.csv`. Περιέχει μόνο τους κόμβους σε κάθε βήμα (από 101 έως 999) που έχουν μολυνθεί (κατάσταση 1). Βασικό αρχείο καταγραφής της κατάστασης 1.
 - Αρχείο `mob2.csv`. Περιέχει συνολικά τις καταστάσεις των 100 κόμβων σε κάθε βήμα (από 101 έως 999) και στο τέλος παρουσιάζεται αθροιστικά ο αριθμός ανά κατάσταση. Συγκεντρωτικό αρχείο στοιχείων.
 - Αρχείο `mob3.csv`. Περιέχει μόνο τα αθροιστικά αποτελέσματα καταστάσεων για τους εκατό κόμβους σε κάθε βήμα (από 101 έως 999). Βασικό αρχείο. Χρησιμοποιήθηκε για τη δημιουργία των γραφικών παραστάσεων της διατριβής.
 - Αρχείο `test_mob_sis_epidemic100.txt`. Περιέχει διάφορα μηνύματα και δείχνει με ακρίβεια το βήμα στο οποίο πραγματοποιήθηκε κάποια αλλαγή κατάστασης για τις τιμές όλων των παραμέτρων εκτέλεσης. Το αρχείο αυτό δεν είναι βασικό αρχείο εξόδου. Χρησιμοποιήθηκε κυρίως για τη διαδικασία ελέγχου του κώδικα.

9.2.1 Παρουσίαση του αρχείου `run_mob_sis_epidemic.bat`.

```
java mob_sis_epidemic N NI sintetagmenes.txt β δ r > test_mob_sis_epidemic100.txt  
pause
```

Όπου N , NI , β , δ , r , παίρνουν τις συγκεκριμένες κάθε φορά τιμές για την εκτέλεση του σεναρίου που εξετάζουμε.