



ΕΛΛΗΝΙΚΟ ΑΝΟΙΚΤΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

Διπλωματική Εργασία

«Μελέτη Επιδημικών Αλγορίθμων σε Περιβάλλοντα Διάχυτου
Υπολογισμού»

Τμήμα 2 – Μελέτη εκμετάλλευση της σημασιολογίας της
πληροφορίας πλαισίου μέσω επιδημικών αλγορίθμων και η
σύγκρισή τους με άλλα μοντέλα διάχυσης πληροφορίας

Παπαδόπουλος Νίκος

Επιβλέπων Καθηγητής: Χατζηευθυμιάδης Στάθης

Ημερομηνία 20/3/2008



© ΕΑΠ, 2008

Η παρούσα διατριβή, η οποία εκπονήθηκε στα πλαίσια της ΘΕ «Διπλωματική Εργασία» του προγράμματος «Μεταπτυχιακή Εξειδίκευση στα Πληροφοριακά Συστήματα» (ΠΛΗΣ), και τα λοιπά αποτελέσματα της αντίστοιχης Διπλωματικής Εργασίας (ΠΕ) 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.



Μελέτη εκμετάλλευση της σημασιολογίας της πληροφορίας πλαισίου μέσω επιδημικών αλγορίθμων και η σύγκρισή τους με άλλα μοντέλα διάχυσης πληροφορίας

Παπαδόπουλος Νικόλαος

**Χατζηευθυμιάδης
Στάθης**

**Ξένος
Μιχάλης**

**Παρασκευάς
Μιχάλης**

Τμήμα Πληροφορικής &
Επικοινωνιών,
Πανεπιστήμιο Αθηνών,
15784, Αθήνα, Ελλάδα

Σχολή Θετικών
Επιστημών & Τεχνολογίας
ΕΑΠ, Πάτρα, Ελλάδα

Πανεπιστήμιο Πάτρας
Πάτρα, Ελλάδα

Περίληψη: Στα σύγχρονα δικτυακά υπολογιστικά συστήματα αποτελεί σημαντική παράμετρο στην ανάπτυξη και στη διαχείρισή τους η μετάδοση της πληροφορίας. Οι αλγόριθμοι που εμπλέκονται σε αυτή τη διαδικασία πρέπει να είναι ευέλικτοι και αποτελεσματικοί ως προς τη διάχυση της πληροφορίας. Συγκεκριμένα, μια κατηγορία που ονομάζεται «επιδημικοί αλγόριθμοι», έχει αποδειχθεί πως είναι ένα ισχυρό εργαλείο προς αυτή την κατεύθυνση για τα σύγχρονα καταναμημένα συστήματα. Αυτό συμβαίνει είτε γιατί το σύστημα είναι πολύπλοκο, είτε επειδή είναι τεράστιο ή είτε απλά ο επιδημικός αλγόριθμος είναι η καλύτερη λύση. Οι επιδημικοί αλγόριθμοι αναπαράγουν τη συμπεριφορά διάδοσης μιας βιολογικής επιδημίας, όταν αυτή εξαπλώνεται μέσα σε έναν πληθυσμό ανθρώπων και σχετίζονται με την πιθανό-θεωρητική διάχυση της πληροφορίας σε ένα ανάλογο δίκτυο. Ένας κόμβος επιλέγει κάποιον γειτονικό του με μια κατανομή πιθανότητας και μεταδίδει την πληροφορία σε αυτόν. Η διαδικασία αυτή συνεχίζεται μέχρις ένα ικανοποιητικό ποσοστό κόμβων του συστήματος ή όλοι να έχουν μολυνθεί. Η εκτενέστερη μελέτη και συμπεριφορά τους πάνω σε θέματα τοπολογίας δικτύων, κινητικότητας κόμβων σε συγκεκριμένη περιοχή, φύσης και μεγέθους της διαχεόμενης πληροφορίας αλλά και αξιοπιστίας αποτελεί πεδίο της σύγχρονης επιδημικής, η οποία ωστόσο έχει ακόμα να ασχοληθεί με πολλά άλυτα, επίκαιρα και σημαντικά θέματα. Από τα πολλά είδη επιδημικών αλγορίθμων εστιάζουμε στο πολύ-επιδημικό μοντέλο δύο επιπέδων μόλυνσης SIS το οποίο περιγράφεται και αναλύεται με τη βοήθεια προσομοιωμένης κίνησης κόμβων.



Λέξεις-κλειδιά: επιδημία, SIS αλγόριθμος, επιδημική, πολύ-επιδημική, μόλυνση επιπέδου 1, μόλυνση επιπέδου 2

Περιεχόμενο: Κείμενο, σχήματα, συγκεντρωτικοί πίνακες και γραφικές παραστάσεις, πρόγραμμα προσομοίωσης μέσω j-sim, πρόγραμμα επιδημικού αλγορίθμου με κινητικότητα κόμβων σε γλώσσα προγραμματισμού java, πρόγραμμα επιδημικού αλγορίθμου χωρίς κινητικότητα κόμβων σε γλώσσα προγραμματισμού java.



**Study for the implementation of framework information
through epidemic algorithms and comparison
with other models for information dissemination**

Papadopoulos Nikolaos

**Hadjiefthymiades
Stathes**

Dept. Informatics &
Telecommunications
University of Athens
15784, Athens, Greece

**Xenos
Michalis**

School of Sciences and
Technology
EAP, Patras, Greece

**Paraskevas
Michalis**

University of Patras,
Patras, Greece

Abstract: The dissemination of information in contemporary computer network systems is a very important factor for their management and development. The algorithms involved in such processes must be flexible and effective as far as data dissemination is concerned. In particular, the so called "epidemic algorithms" have proved to be a powerful tool in that direction for modern systems. This happens either because the system is complicated, or huge, or simply because the epidemic algorithm is the best solution for the system. They reproduce the behavior of a biological epidemic when it spreads in a human population and they are related with the possibilistic dissemination of information in a network. A node chooses one of its neighbours with a possibility to infect it, which means to spread the information it carries. This procedure ends when all the nodes of the system or a significant percentage will be infected. The comprehensive study of their behavior in network topology and node motion in particular areas concerning the nature and size of information as well as its reliability is the object of contemporary epidemics which however has to find solution as yet to many issues. Out of many types of epidemic algorithms we focus on the multi-epidemic model of two levels of infection SIS which is described and explained with the aid of simulated node motion.



Ευχαριστίες

Μέσα από τις λίγες αυτές γραμμές θα ήθελα να εκφράσω ένα μεγάλο ευχαριστώ από τα βάθη της καρδιάς μου στον καθηγητή μου Χατζηευθυμιάδη Στάθη, στον υποψήφιο διδάκτορα Αναγνωστόπουλο Χρήστο, στον Κοντό Φάνη, στον Λυμπέρη Κώστα, στον Ευθυμιάτο Γιώργο και στον Βύρα Παναγιώτη για την ποικιλότητα βοήθεια που μου έδωσαν από την αρχή της χρονιάς για να φέρω εις πέρας τη διπλωματική αυτή εργασία.



Περιεχόμενα

1. ΕΙΣΑΓΩΓΗ	8
2. ΕΙΣΑΓΩΓΗ ΣΤΑ ΜΟΝΤΕΛΑ ΔΙΑΧΥΣΗΣ ΠΛΗΡΟΦΟΡΙΑΣ ΒΑΣΕΙ ΕΠΙΔΗΜΙΚΩΝ ΑΛΓΟΡΙΘΜΩΝ	9
2.1 ΜΟΝΤΕΛΑ ΕΠΙΔΗΜΙΑΣ	10
2.2 ΑΛΓΟΡΙΘΜΟΙ ΣΤΗΝ ΕΠΙΔΗΜΙΚΗ	18
2.2.2 Αλγόριθμος SIS	18
2.3 ΑΛΓΟΡΙΘΜΟΙ ΓΙΑ ΤΗΝ ΠΟΛΥ-ΕΠΙΔΗΜΙΚΗ ΔΙΑΔΟΣΗ ΠΛΗΡΟΦΟΡΙΑΣ	19
2.3.1 Το προτεινόμενο μοντέλο	20
3. ΕΦΑΡΜΟΓΗ ΕΠΙΔΗΜΙΚΗΣ ΚΑΙ ΠΟΛΥ-ΕΠΙΔΗΜΙΚΗΣ ΣΕ ΠΕΡΙΒΑΛΛΟΝΤΑ ΔΙΑΧΥΤΟΥ ΥΠΟΛΟΓΙΣΜΟΥ	25
3.1 Το προτεινόμενο μοντέλο με/χωρίς κινητικότητα κομβών (RWP)	27
3.1.1 Επιδημική επικράτηση	27
3.1.2 Επιδημική αποσύνθεση	42
3.2 ΣΥΓΚΡΙΤΙΚΟΙ ΣΥΝΟΠΤΙΚΟΙ ΠΙΝΑΚΕΣ ΑΠΟΤΕΛΕΣΜΑΤΩΝ	48
4. ΣΥΜΠΕΡΑΣΜΑΤΑ	50
ΑΝΑΦΟΡΕΣ	52
ΠΑΡΑΡΤΗΜΑ Α: J-SIM	53
ΠΑΡΑΡΤΗΜΑ Β: JAVA	65



1. Εισαγωγή

Στην παρούσα διπλωματική εργασία που εκπονήθηκε στα πλαίσια του μεταπτυχιακού προγράμματος «**Μεταπτυχιακή Εξειδίκευση στα Πληροφοριακά Συστήματα**» του **Ελληνικού Ανοικτού Πανεπιστημίου** μελετάται η σημασιολογία της πληροφορίας πλαισίου για τη διάχυσή του μέσα σε δίκτυα υπολογιστών καθώς και σημαντικά μοντέλα επιδημικών αλγορίθμων. Το κείμενο της εργασίας συνοδεύεται από κώδικα που παρατίθεται στα παραρτήματα 1 και 2 καθώς και στον συνοδευτικό ψηφιακό οπτικό δίσκο.

Η οργάνωση της διπλωματικής εργασίας έχει γίνει με τον παρακάτω τρόπο:

Στο Κεφάλαιο 2 «**Εισαγωγή στα Μοντέλα Διάχυσης Πληροφορίας βάσει Επιδημικών Αλγορίθμων**» παρουσιάζεται η εισαγωγή στην επιδημική και στην πολύ-επιδημική στα υπολογιστικά συστήματα.. Περιγράφονται οι σχετικές έννοιες και παρουσιάζονται-αναλύονται οι αντίστοιχοι αλγόριθμοι.

Στο Κεφάλαιο 3 «**Εφαρμογή επιδημικής και πολύ-επιδημικής σε Περιβάλλοντα Διάχυτου Υπολογισμού**» εξετάζονται και παρατίθενται τα αποτελέσματα της εφαρμογής διάφορων αλγορίθμων σε συγκεκριμένο προσομοιωμένο παράδειγμα κινητών και ακίνητων κόμβων.

Στο Κεφάλαιο 4 «**Συμπεράσματα**» συνοψίζονται τα συμπεράσματα της επιδημικής και πολύ-επιδημικής εφαρμογής καθώς και διατυπώνονται προτάσεις για μελλοντική εργασία και εκμετάλλευση των συμπερασμάτων.

Το **Παράρτημα 1** αφορά το περιβάλλον προσομοίωσης J-SIM που χρησιμοποιήθηκε για τη μελέτη της τυχαίας κίνησης των κόμβων του συστήματος. Εξηγούνται βασικά στοιχεία του και περιγράφονται τα εργαλεία που χρησιμοποιήθηκαν για τη δημιουργία του αρχείου που παρήγαγε τις τυχαίες θέσεις των κόμβων. Επιπλέον δίνεται και ο κώδικας του αντίστοιχου αρχείου προσομοίωσης.

Τέλος, στο **Παράρτημα 2** αναφέρεται στο προγραμματιστικό περιβάλλον JAVA που χρησιμοποιήθηκε για τη δημιουργία των επιδημικών αλγορίθμων με κινητικότητα και χωρίς κινητικότητα των κόμβων. Επίσης, παρουσιάζεται και η υλοποίηση του κώδικα των δύο αρχείων.



2. Εισαγωγή στα Μοντέλα Διάχυσης Πληροφορίας βάσει Επιδημικών Αλγορίθμων

Η μετάδοση πληροφοριών μέσα από συστήματα υπολογιστών ή μέσα από το διαδίκτυο τα τελευταία χρόνια έχει απασχολήσει και έχει αποτελέσει έναν πολύ σημαντικό και κρίσιμο παράγοντα έρευνας και μελέτης. Καθώς η ανάπτυξη όλο και πολυπλοκότερων συστημάτων διάχυτου υπολογισμού ή η δημιουργία μεγάλων αδόμητων (ad hoc) ασύρματων κινητών δικτύων συνεχίζεται, το ενδιαφέρον έχει στραφεί αναγκαστικά σε περιοχές έρευνας που αφορούν στον ευκολότερο και απλούστερο τρόπο μετάδοσης της πληροφορίας μέσω των πληροφορικών συστημάτων.

Σημαντικά προβλήματα συνεχίζουν να υφίστανται ως προς την επικοινωνία μεταξύ των κινητών τερματικών σε ασύρματα δίκτυα. Ο περιορισμός που υφίσταται η μετάδοση πληροφοριών, για παράδειγμα μέσω των σταθμών βάσης στα κυψελοειδή δίκτυα, από το ένα τερματικό στο άλλο αποτελεί ανασταλτικό παράγοντα στην ευελιξία και στην ικανότητα χειρισμού ανεπιθύμητων καταστάσεων σε αυτά τα συστήματα. Αρκεί να αναφέρουμε, πως αν οι σταθμοί βάσης πάψουν να λειτουργούν τα τερματικά μεταξύ τους δεν μπορούν να επικοινωνήσουν. Από την άλλη, παρόλο που στα αδόμητα δίκτυα η επικοινωνία μεταξύ των τερματικών είναι εφικτή προκύπτουν ζητήματα που απαιτούν επιτακτικότερα λύση. Τέτοια είναι ο τρόπος δρομολόγησης της πληροφορίας, η μέγιστη χωρητικότητα των δικτύων, η ασφάλειά τους, ο σωστός σχεδιασμός τους κλπ.

Αναδεικνύεται με αυτόν τον τρόπο πως η μελέτη διάχυσης της πληροφορίας στα υπολογιστικά συστήματα, είτε βρίσκονται σε κατάσταση λειτουργίας είτε σε στάδιο σχεδιασμού, αποτελεί ίσως τον κρισιμότερο παράγοντα για τη σωστή διαχείριση και την αξιοποίησή τους. Το γεγονός αυτό ενισχύεται και από τα τεράστια ποσά που επενδύονται προς αυτή την κατεύθυνση, αλλά από την σημαντικότητα που τα δίκτυα αυτά έχουν στην καθημερινή ανθρώπινη δραστηριότητα. Έτσι, στον τομέα της μελέτης διάχυσης της πληροφορίας έχουν προταθεί, αναπτυχθεί και μελετηθεί διάφοροι τύποι αλγορίθμων, όπως οι ντετερμινιστικοί α λ γ ο ρ ι θ μ ο ι π λ η μ μ υ ρ α ς (Flooding Algorithms) ή οι περισσότερο υποσχόμενοι ε π ι δ η μ ι κ ο ί α λ γ ο ρ ι θ μ ο ι (Epidemic Algorithms).

Οι επιδημικοί αλγόριθμοι έχουν αναδειχθεί σ' ένα πολύ ισχυρό εργαλείο μελέτης για τη μετάδοση της πληροφορίας σε καταναμημένα συστήματα υπολογιστών [7]. Η φιλοσοφία που τους χαρακτηρίζει είναι πως αν μια διαδικασία επιθυμεί να διαδώσει ένα νέο κομμάτι πληροφορίας στο σύστημα, δεν το στέλνει σ' έναν κεντρικό υπολογιστή ή σε μια συστάδα κεντρικών υπολογιστών. Προτιμά την αποστολή του σε ένα σύνολο παρόμοιων διαδικασιών,



που επιλέγονται τυχαία. Στη συνέχεια, κάθε μία από τις διαδικασίες αναπαράγει την ίδια συμπεριφορά, και ούτω καθ' εξής.

Η αρχή που κρύβεται κάτω από τη διάδοση πληροφοριών μιμείται την τεχνική διάδοσης μιας *Επιδημίας* (Epidemic), όπως ακριβώς τη γνωρίζουμε από το βιολογικό πεδίο εφαρμογής. Έτσι αναφερόμαστε, κατ' αναλογία, σε *Επιδημική Διάδοση Πληροφορίας* (Epidemic Information Dissemination). Αντίστοιχο και ανάλογο παράδειγμα μπορούμε να επικαλεστούμε για τη διάδοση μιας φήμης μεταξύ ανθρώπων που κουτσομπολεύουν: στην περίπτωση αυτή αναφερόμαστε σε διάδοση κουτσομπολιού.

Μόλις αρχίσουν οι επιδημίες είναι δύσκολο να εξασθενήσουν: μερικοί άνθρωποι όταν μολύνονται από μια μεταδοτική ασθένεια είναι σε θέση να τη διαδώσουν, άμεσα ή έμμεσα, σ' έναν αρκετά μεγάλο πληθυσμό μολυσμένων ή όχι ατόμων. Οι επιδημίες, εξάλλου, 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 πιθανό-θεωρητική διάχυση της πληροφορίας σ' ένα δίκτυο υπολογιστών. Η έννοια της πιθανό-θεωρητικής διάχυσης πληροφορίας δηλώνει ότι ένας κόμβος i επιλέγει κάποιον γειτονικό του j με κάποια κατανομή πιθανότητας για να μεταδώσει την πληροφορία σε αυτόν (στην περίπτωση αυτή λέμε ότι ο κόμβος i μολύνει τον κόμβο j). Η διαδικασία αυτή επαναλαμβάνεται μέχρις ότου όλοι οι κόμβοι (ή ένα ικανοποιητικό ποσοστό κόμβων) του δικτύου να έχουν μολυνθεί (δηλαδή όλοι οι κόμβοι να έχουν λάβει την πληροφορία).

Η συμπεριφορά των επιδημικών αλγορίθμων παρουσιάζει ιδιαίτερο ενδιαφέρον. Συγκεκριμένα, η τοπολογία του δικτύου, η κινητικότητα των κόμβων μέσα στον καθορισμένο χώρο, το πλήθος των κόμβων και η φύση της πληροφορίας που πρόκειται να διανεμηθεί επηρεάζουν τον *ολικό χρόνο μόλυνσης* όλων των κόμβων του δικτύου, το *συνολικό ποσοστό μόλυνσης* των κόμβων, τον *χρόνο παραμονής* των κόμβων σε κατάσταση μόλυνσης, κ.α.

2.1 Μοντέλα Επιδημίας

Είναι εύκολο να αντιληφθούμε την αναλογία μεταξύ 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 κάποιο είδος «επαφής» 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 (Epidemic Algorithms).

Οι επιδημικοί αλγόριθμοι μπορούν να κατηγοριοποιηθούν σε τρεις μεγάλες κατηγορίες ανάλογα με τον τρόπο που επικοινωνούν οι γειτονικοί κόμβοι μεταξύ τους [2]:

- ο Αλγόριθμοι P u s h - b a s e d : ο μολυσμένος κόμβος διαχέει τις πληροφορίες του σε επιλεγμένους γειτονικούς κόμβους.
- ο Αλγόριθμοι P u l l - b a s e d : 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.
- ο Αλγόριθμοι P u s h & P u l l - b a s e d : συνδυάζει τους δύο προηγούμενους τρόπους. Είτε ένας κόμβος είναι μολυσμένος είτε οι γείτονές του είναι μολυσμένοι τότε όλοι (κόμβος και γείτονες) θα μολυνθούν.

Ένας επιδημικός αλγόριθμος χρησιμοποιεί δύο τύπους μετάδοσης πληροφορίας. Ο πρώτος τύπος αναφέρεται στη μετάδοση Π λ η ρ ο φ ο ρ ί α ς Π λ α ι σ ί ο υ (Contextual Information Dissemination) και ο δεύτερος στη μετάδοση Π λ η ρ ο φ ο ρ ί α ς Θ έ σ η ς (Situational Information Dissemination). Σαν Π λ α ί σ ι ο (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 [1].

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 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 ενός κόμβου μπορεί να μεταδοθεί μέσα από ένα σύνολο κόμβων.

Πρέπει να τονιστεί πως, η μετάδοση πληροφορίας θέσης εξαρτάται άμεσα και ισχυρά από τη μετάδοση πληροφορίας πλαισίου, επειδή η θέση ορίζεται σαν ένα συστατικό της πληροφορίας πλαισίου. Κατ' αναλογία με το βιολογικό μοντέλο αν συσχετίσουμε, πως τα βακτήρια μπορούν να δημιουργούν διαφορετικούς τύπους ασθενειών ενώ διάφορα

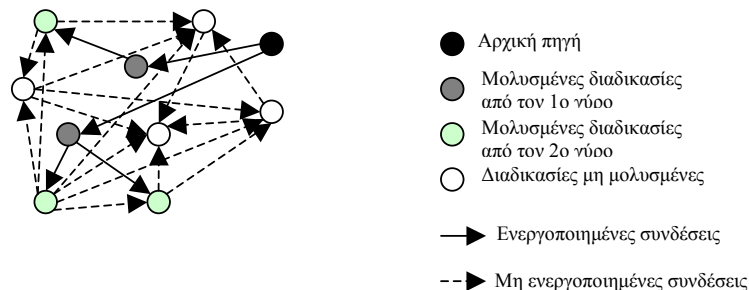


κομμάτια πληροφορίας πλαισίου μπορούν να συνεπάγονται διαφορετικές καταστάσεις, τότε η διάδοση πλαισίου αντιστοιχεί στη διάδοση των βακτηρίων και η διάδοση θέσης αντιστοιχεί στην επιδημία.

Σε έναν επιδημικό αλγόριθμο, κάθε διαδικασία του συστήματος ενδεχομένως να εμπλέκεται στη διάδοση. Πρακτικά, κάθε διαδικασία αποθηκεύει κάθε μήνυμα (μονάδα πληροφορίας) που λαμβάνει μέχρι μιας ορισμένης *Χωρητικότητας* b (Buffer Capacity), και προωθεί το μήνυμα κατά έναν περιορισμένο *αριθμό φορών* t . Η διαδικασία διαβιβάζει το μήνυμα κάθε φορά σε ένα τυχαία επιλεγμένο σύνολο διαδικασιών περιορισμένου μεγέθους f , που ονομάζεται *Απλωμα της Διάδοσης* (Fanout). [7]

Πολλές παραλλαγές των επιδημικών αλγορίθμων διάδοσης υπάρχουν και διακρίνονται από την αξία των παραπάνω παραμέτρων: b , t και f . Αυτές οι παράμετροι μπορούν να είναι ανεξάρτητες από τον αριθμό n των διαδικασιών στο σύστημα, οπότε σ' αυτή την περίπτωση το φορτίο που επιβάλλεται σε κάθε διαδικασία παραμένει οριακό. Η αξιοπιστία της παράδοσης πληροφοριών θα εξαρτηθεί και από τις τιμές των παραμέτρων καθώς επίσης και από μέγεθος n του συστήματος. Εναλλακτικά, οι παράμετροι διάδοσης μπορούν να εξελιχθούν με το μέγεθος n του συστήματος. Σε αυτήν την περίπτωση, το φορτίο θα μπορούσε να διατηρηθεί εάν οι τιμές των παραμέτρων αυξάνουν αργά με το n , π.χ. λογαριθμικά.

Η αξιοπιστία των επιδημικών αλγορίθμων βασίζεται σ' έναν προ-ενεργό μηχανισμό (pro-active) στον οποίον ο πλεονασμός και η τυχαιοποίηση επεκτείνονται αρχικά στις παραβλεπόμενες πιθανές αποτυχίες διαδικασίας και αποτυχιών συνδέσεων δικτύων. 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 (σχ. 1). Η



Σχήμα 1 Απεικόνιση μιας επιδημίας



αποτυχία κάποιων συνδέσεων δύσκολα επηρεάζει τη διάδοση της επιδημίας.

Κανένας μηχανισμός δεν απαιτείται για την ανίχνευση και την αναδιαμόρφωση των αποτυχιών. Αντίθετα από τους αντιδραστικούς αλγόριθμους στους οποίους οι διαδικασίες αντιδρούν στις αποτυχίες με την αναμετάδοση των ελλειπουσών πληροφοριών. Τελικά, οι μόνιμα αποτυχημένες διαδικασίες θα πρέπει να αφαιρεθούν από το σύστημα.

Υπό κάποια έννοια, οι επιδημικοί αλγόριθμοι εκθέτουν μια αμφιτροπική συμπεριφορά: υπάρχει ένα κατώτατο όριο στις τιμές παραμέτρων διάδοσης για τις οποίες μια αξιόπιστη παράδοση συνοδεύεται από μια υψηλή πιθανότητα. Η πιθανολογική εγγύηση της παράδοσης του μηνύματος συσχετίζεται άμεσα με τις τιμές των παραμέτρων διάδοσης. Αυτές οι παράμετροι μπορούν να συντονιστούν έτσι ώστε με την αυθαίρετα υψηλή πιθανότητα, ο αλγόριθμος να ικανοποιεί τις εγγυήσεις που οι αιτιοκρατικοί αλγόριθμοι παρέχουν.

Όταν εφαρμόζεται στην πράξη ο επιδημικός αλγόριθμος διάδοσης, πρέπει να ληφθούν υπόψη συγκεκριμένοι περιορισμοί σχεδίασης που αφορούν τις απαιτήσεις των πόρων κάθε διαδικασίας. Σε αυτούς τους περιορισμούς περιλαμβάνονται: [7]

(1) Πώς οι διαδικασίες γνωρίζουν η μία την άλλη και πόσες πρέπει να ξέρουν κάθε φορά (Ι δ ό τ η τ α Μ έ λ ο υ ς - Membership);

(2) Πώς μπορούν οι συνδέσεις μεταξύ των διαδικασιών να αντιστοιχίζονται στην πραγματική τοπολογία των δικτύων έτσι ώστε η απόδοση να είναι αποδεκτή (Α ν τ ί λ η ψ η Δ ι κ τ ύ ο υ - Network Awareness);

(3) Ποιες πληροφορίες φεύγουν από μία διαδικασία όταν η μνήμη είναι πλήρης (Δ ι α χ ε ί ρ ι σ η Π ρ ο σ ω ρ ι ν ή ς Μ ν ή μ η ς - Buffer Management);

(4) Πώς θα αποφασιστεί το ενδιαφέρον κάποιας πληροφορίας για μία διαδικασία, ώστε να μειωθεί η πιθανότητα λήψης και αποθήκευσης άχρηστων πληροφοριών (Φ ι λ τ ρ ά ρ ι σ μ α Μ η ν υ μ ά τ ω ν - Message Filtering);

Αν και οι μελέτες των φυσικών επιδημιών μπορούν να παρέχουν χρήσιμες ιδέες για την αντιμετώπιση τέτοιων ζητημάτων, η αναλογία δεν είναι απαραίτητη. Τα ανωτέρω ζητήματα απαιτούν καινοτόμες λύσεις επειδή η γνώση αντιμετώπισής τους έχει συνδεθεί κυρίως με την καταπολέμηση της επιδημίας. Ένα τέτοιο γεγονός στην περίπτωση μας είναι άχρηστο, αφού επιδιώκεται ακριβώς το αντίθετο, η διάδοσή της.

(1) Η ιδιότητα μέλους είναι ένα θεμελιώδες θέμα που διέπει τον τρόπο ανάπτυξης ενός αλγορίθμου για τη διάδοση μιας πληροφορικής επιδημίας. Στην πράξη, για τη διάδοση



μιας επιδημίας, κάθε διεργασία (κόμβος) p η οποία λαμβάνει ένα μήνυμα μπορεί να το αποστέλλει μόνο σε άλλες διεργασίες που γνωρίζει. Είναι σημαντικό να καθοριστεί πώς μια οποιαδήποτε διεργασία p προσλαμβάνει τις δικές της ειδικές πληροφορίες συμμετοχής, αφού αυτό επηρεάζει τον τρόπο που εκτελείται μια διάδοση επιδημίας. Αυτός ο συσχετισμός γνωριμιών (ποιος γνωρίζει ποιον) είναι ίσως ο πιο βασικός παράγοντας που πρέπει να υπολογίσουμε για τον σχεδιασμό κλιμακούμενων εφαρμογών με επιδημικούς αλγορίθμους.

Ένας άμεσος τρόπος να λυθεί το παραπάνω θέμα θα ήταν κάθε διεργασία να έχει κατάλογο όλων των άλλων διεργασιών στο σύστημα και ως εκ τούτου να μπορεί να επικοινωνεί με κάθε μία από αυτές. Η συγκεκριμένη υπόθεση μπορεί να στηριχτεί σε πραγματική βάση μόνο θεωρώντας την εφαρμογή της σε μια ομάδα διεργασιών μετρίου μεγέθους. Καθίσταται μη πρακτική όταν εφαρμόζεται σε μεγάλες ομάδες διεργασιών για διάφορους λόγους, όπως:

- Η αποθήκευση που απαιτείται για την πληροφορία συμμετοχής αυξάνει γραμμικά με το μέγεθος του συστήματος.

- Για μια συνεκτική θεώρηση της συμμετοχής θα απαιτούνταν επιπλέον φόρτος στο δίκτυο, ειδικά σε δυναμικά περιβάλλοντα. Παράδειγμα δυναμικού περιβάλλοντος είναι τα δίκτυα Peer-to-Peer στο διαδίκτυο, όπου οι διεργασίες συχνά επικαλύπτουν εναλλασσόμενες καταστάσεις ή δίκτυα που στήνονται για μια ειδική περίπτωση, όπου τα κανάλια επικοινωνίας ανάμεσα στις διεργασίες μπορεί να αναπτύσσουν νέες ιδιότητες πολύ γρήγορα.

(2) Αν κάθε διαδικασία είναι ελεύθερη να στέλνει σε όποια άλλη θέλει μηνύματα τότε η επίδραση πάνω στο σύστημα μπορεί να είναι ολέθρια, καθώς αυξάνεται δραματικά ο φόρτος του δικτύου και κινδυνεύει η σταθερότητά του. Ο περιορισμός αυτός επιβάλλει την οργανωτική ιεράρχηση των διαδικασιών με στόχο ο αλγόριθμος διάχυσης πληροφορίας να στέλνει πρώτα σε μία ομάδα διαδικασιών την πληροφορία. Αυτή η τεχνική μειώνει τον φόρτο του δικτύου αλλά η ιεράρχηση των διαδικασιών αποτελεί ένα ενεργό πεδίο έρευνας.

(3) Έχοντας εξάρτηση από τον ρυθμό της παραγόμενης πληροφορίας στο σύστημα η χωρητικότητα της μνήμης κάθε διαδικασίας μπορεί να μην είναι αρκετή ώστε να μπορεί κάθε μήνυμα να το αποθηκεύει για αρκετό χρόνο, να το προωθεί ικανοποιητικό αριθμό φορές για να επιτύχει την αναμενόμενη αξιοπιστία. Η λύση θα μπορούσε να δοθεί μέσα από δύο τακτικές: Τα νέα μηνύματα που φτάνουν απορρίπτονται όταν η μνήμη είναι γεμάτη ή παλιά απορρίπτονται όταν φτάνουν νέα. Στην τελευταία περίπτωση όμως ίσως κάποια από τα παλιά να μην είχαν προωθηθεί σε αυτή την περίπτωση αρκετές φορές. Εναλλακτικά τα μηνύματα θα μπορούσαν να έχουν προτεραιότητες και το σύστημα να τα διαχειρίζεται με αυτόν τον τρόπο. Ένας άλλος τρόπος αντιμετώπισης είναι να μειωθεί ο αριθμός των



μηνυμάτων στο σύστημα. Κάθε φορά που μια διαδικασία θέλει να προωθήσει μηνύματα σε άλλες μπορεί να υπολογίζει τη χωρητικότητα των γειτονικών της και ανάλογα να αποφασίζει.

(4) Ενδιαφέρον, αλλά εξαιρετικά πολύπλοκο εγχείρημα, είναι να καθοριστεί ο τρόπος με τον οποίο να αυξάνεται η πιθανότητα κάθε φορά που μία διαδικασία πρόκειται να στείλει ένα μήνυμα να το παραλάβει μια άλλη που να την ενδιαφέρει αυτό ή αντίστοιχα να μειώνεται η πιθανότητα σε περίπτωση που δεν την ενδιαφέρει. Η επιτυχία της διαδικασίας αυτής εξαρτάται από τον τρόπο που θα ενσωματωθεί στον επιδημικό αλγόριθμο το φιλτράρισμα των μεταδιδόμενων μηνυμάτων. Παρόλα αυτά είναι δύσκολο να σχεδιαστεί ένα σύστημα με το φιλτράρισμα μηνυμάτων σαν προτεραιότητα χωρίς να έχει αυτό επίδραση πάνω σε άλλους περιορισμούς, όπως στην αντίληψη του δικτύου.

Το Μοντέλο SIS

Παρακάτω αναφέρουμε ορισμένους ορισμούς σχετικούς με την επιδημία. [2]

Ορισμός 1: Ένας πληθυσμός είναι ένα σύνολο κόμβων. Ο πληθυσμός μπορεί να οριστεί χωρικά, για παράδειγμα, όλοι οι κόμβοι που βρίσκονται σε κάποια συγκεκριμένη γεωγραφική περιοχή. Οι κόμβοι ενός πληθυσμού είναι ικανοί να διαχέουν πληροφορίες πλαισίου μεταξύ τους με στόχο να ενημερώνουν και να συμπεραίνουν το πλαίσιο σε συνεργασία. Έστω M ο αριθμός του πληθυσμού των κόμβων και με δείκτη i ο κάθε κόμβος με $i \in N$.

Ορισμός 2: Οι γείτονες ενός κόμβου i δημιουργούν τη γειτονιά του i , Γ_i που είναι υποσύνολο του αρχικού πληθυσμού M . Γ_i είναι το σύνολο των κόμβων (συμπεριλαμβανομένου και του i) που βρίσκονται σε ακτίνα επικοινωνίας με τον i . Ο επιδημικός αλγόριθμος επιλέγει ένα υποσύνολο του Γ_i με σκοπό να διαχύσει ο i κόμβος μια ενημερωμένη έκδοση ενός κομματιού πληροφορίας.

Ορισμός 3: Ένας κόμβος i ονομάζεται **Μ ο λ υ σ μ έ ν ο ς** (Infected) όταν για ένα καθορισμένο χρονικό διάστημα μεταφέρει ενημερωμένο πλαίσιο πληροφορίας το οποίο μπορεί να μεταφέρει σε γειτονικούς του κόμβους. Αφού μολυνθεί ένας κόμβος παραμένει μολυσματικός μέχρι αυτή η πληροφορία να γίνει άχρηστη. Αντίστοιχα, ένας κόμβος i καλείται **Υ γ ι ή ς** (Susceptible) αν μπορεί να μολυνθεί από έναν μολυσματικό κόμβο j που ανήκει στην γειτονιά του Γ_i ή αν έχει ιαθεί (μεταφέρει ένα απαρχαιωμένο κομμάτι πλαισίου)

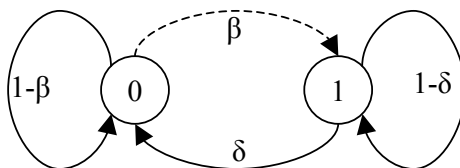


Ορισμός 4: Μια επιδημική Μεταστοιχείωση (Μετάλλαξη) σημαίνει πως η επιδημία αλλάζει αλλά συνεχίζει να παραμένει μολυσματική (αναφερόμενη σε δυνατότερο \mathcal{I}). Στο πλαίσιο μας η μεταστοιχειωμένη επιδημία αναφέρεται σε μια πιο εξειδικευμένη συνεπαγόμενη κατάσταση ώστε ο πληθυσμός να μπορεί να ξαναμολυνθεί.

Διάφοροι παράγοντες επηρεάζουν τους χρόνους μετάδοσης, όπως για παράδειγμα, τους χρόνους διασποράς πλαισίου σε όλους τους κόμβους του πληθυσμού. Αυτοί είναι:

- ο το μέγεθος του πληθυσμού M
- ο η χωρητικότητα μνήμης (buffer) του κόμβου που είναι περιορισμένη και αποθηκεύει τα λαμβανόμενα μηνύματα πληροφοριών
- ο η ακτίνα μετάδοσης R , δηλαδή η ακτίνα επικοινωνίας των κόμβων
- ο ο ρυθμός μόλυνσης β (ο ρυθμός μόλυνσης υγιών γειτονικών κόμβων από έναν μολυσμένο κόμβο)
- ο ο ρυθμός ίασης δ (ο ρυθμός ίασης ενός μολυσμένου κόμβου)
- ο το επιδημικό όριο λ , που ισούται με β/δ και καθορίζει αν το σύστημα θα γίνει υγιές ή θα μολυνθεί
- ο ο χρόνος ανακάλυψης (ο απαιτούμενος χρόνος για να εγκαθιδρυθεί επικοινωνία με έναν κόμβο που βρίσκεται μέσα στην ακτίνα επικοινωνίας)
- ο το μοντέλο κινητικότητας των κόμβων
- ο το μοντέλο του επιδημικού αλγορίθμου που θα χρησιμοποιηθεί
- ο η φύση της επιδημίας (στατικό πλαίσιο πληροφορίας ή δυναμικά και ισχυρά μεταβαλλόμενο πλαίσιο).

Ειδικότερα, οι τιμές των β και δ μιας επιδημίας επηρεάζονται όταν η φύση της επιδημίας αλλάζει. Αυτό σχετίζεται με το εξής: Ένας μολυσμένος κόμβος μπορεί να ξαναμολυνθεί όταν η επιδημία μεταστοιχειώνεται (μεταλλάζεται) σε μια άλλη καινούρια. Συνδυάζοντας μόλυνση και ίαση, το επιδημικό μοντέλο ονομάζεται Susceptible-Infected-Susceptible (SIS), δηλαδή Υγιής-Μολυσμένος-Υγιής. Αυτό είναι ένα πολύ διαδεδομένο μοντέλο που χρησιμοποιείται στην επιδημιολογική έρευνα και απεικονίζεται στο παρακάτω σχήμα (σχ. 2). Όπως φαίνεται, αν είναι υγιής (κατάσταση 0) έχει πιθανότητα β να μολυνθεί και πιθανότητα $1-\beta$ να παραμείνει υγιής. Αντίστοιχα αν είναι μολυσμένος (κατάσταση 1) έχει πιθανότητα δ να ιαθεί και $1-\delta$ να παραμείνει μολυσμένος.



Σχήμα 2 Καταστάσεις Μοντέλου SIS

Το μοντέλο SIS αγνοεί τις πληροφορίες μόλυνσης μέσα στον κόμβο, απλοποιεί την επιδημική μετάδοση και τις πιθανότητες διάδοσης ανά χρονική περίοδο στη μορφή των ρυθμών β και δ , 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 μας, όταν αναφερόμαστε στη διάχυση πλαισίου ο περιορισμός της μη επανα-μόλυνσης ενός μολυσμένου κόμβου είναι αποδεκτός.

Από τη άλλη όταν αναφερόμαστε σε διάχυση κατάστασης, στην οποία η επιδημία μπορεί να μεταστοιχειωθεί (περιστασιακό πλαίσιο - situational context) πρέπει να επεκτείνουμε το μοντέλο SIS ώστε ένας μολυσμένος κόμβος να μπορεί να ξαναμολυνθεί από ισχυρότερη μόλυνση. Τελικώς, το μοντέλο υποθέτει ότι ένας κόμβος δεν μπορεί να γίνει λιγότερο επιρρεπής μετά από μία μόλυνση. Έτσι υποθέτουμε ότι τα β και δ δεν αλλάζουν με τον χρόνο.

Αν είναι $S(t)$ ο αριθμός των υγιών κόμβων και $I(t)$ ο αριθμός των μολυσμένων κόμβων σε μια στιγμή t , τότε η ποσότητα $p(t) = \frac{I(t)}{M}$ δείχνει τους κόμβους που μολύνθηκαν σε πληθυσμό $M = S(t) + I(t)$ μέχρι τη χρονική στιγμή t . Σε ένα ομογενές τύπο δικτύου το επιδημικό μοντέλο χρησιμοποιεί την παρακάτω μη γραμμική διαφορική εξίσωση (1) για να υπολογίσει τον επιδημικό πληθυσμό [3]

$$\frac{dp(t)}{dt} = \beta \cdot p(t)[1 - p(t)] - \delta \cdot p(t) \quad (1)$$

Η λύση της (1) δίνεται από την εξίσωση (2)

$$p(t) = \frac{p_0(1 - \lambda)}{p_0 + (1 - \lambda - p_0) \cdot e^{-(\beta - \delta)t}} \quad (2)$$

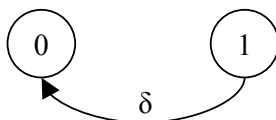
όπου $\lambda = \beta/\delta$ είναι το *Επιδημικό Όριο* ή *Κατώφλι* (Epidemic Threshold). Αν $\lambda > 1$ έχουμε επικράτηση της επιδημίας, αλλιώς εξασθένησή της.



2.2 Αλγόριθμοι στην Επιδημική

Για τη μελέτη της επιδημικής διάδοσης πληροφορίας σε συστήματα υπολογιστών έχουν προταθεί διάφορα μοντέλα. Αυτά περιγράφονται από αντίστοιχους αλγόριθμους.

Το απλούστερο επιδημικό μοντέλο είναι το Susceptible-Infected, που συνοπτικά αναφέρεται σαν μοντέλο SI. Σε αυτό ένας μολυσμένος κόμβος, ο οποίος συμβολίζεται με την κατάσταση 1, μπορεί να μολύνει έναν άλλον κόμβο που είναι υγιής και συμβολίζεται με



0, (σχ. 3). Δεν υπάρχει ίαση στο συγκεκριμένο μοντέλο και ο πληθυσμός του δικτύου που μολύνεται συνεχώς αυξάνει έως ότου μολυνθεί όλος.

2.2.2 Αλγόριθμος SIS

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.

Σχήμα 3 Καταστάσεις Μοντέλου SI

Στον αλγόριθμο SIS, αρχικά ελέγχεται αν ένας κόμβος j που βρίσκεται σε μολυσμένη κατάσταση με κάποια τυχαία πιθανότητα δ μπορεί να μεταβεί σε υγιή (ίαση), διαφορετικά για όλους τους μολυσμένους γειτονικούς του κόμβους (δηλαδή, για όσους βρίσκονται στην ακτίνα δράσης του) ελέγχεται αν μπορεί κάποιος από αυτούς να τον μολύνει.

Στη συνέχεια παρουσιάζεται ο αλγόριθμος:



Algorithm Epidemic For Node j At Time t

```
Input (Infection Probability  $\beta$ , Recovery Probability  $\delta$ )
If (S[j] == INFECTED) Then
  If (Random() <  $\delta$ )
    // έλεγχος αν υπάρχει πιθανότητα  $P(x) < \delta$  να γίνει ο κόμβος «υγιής» ή όχι
    S[j] <- SUSCEPTIBLE
  End If
Else
  For each node  $i=1, \dots, L$ , ( $i \neq j$ ) Do
    If(S[i] == 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 (SQRT((D[j,t,0] - D[i,t,0])2 + (D[j,t,10] - D[i,t,1])2) <= R) Then
        If (Random() <  $\beta$ ) Then
          S[j] <- INFECTED
        End If
      End If
    End If
  End For
End If
End Algorithm
```

2.3 Αλγόριθμοι για την Πολύ-Επιδημική Διάδοση Πληροφορίας

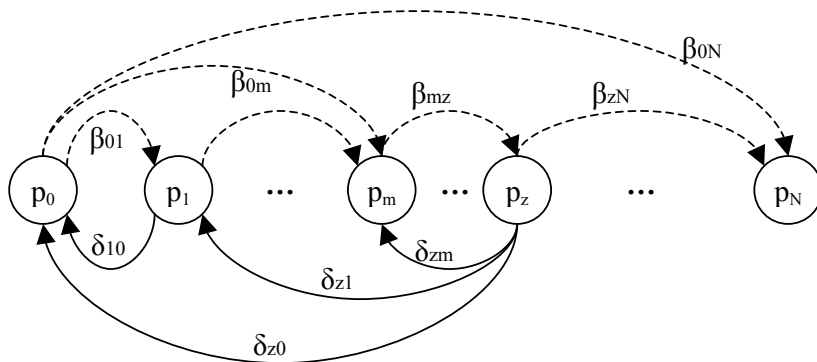
Η πολύ-επιδημική σχετίζεται με πολλές καταστάσεις μόλυνσης για τους κόμβους που αποτελούν το δίκτυο. Αν συμβολίσουμε με N ένα σύνολο κόμβων, τότε, έστω p_k να είναι το γεγονός πως ο κόμβος i βρίσκεται σε μολυσμένη κατάσταση επιπέδου k , με $0 < k < N$. Δηλαδή, αναφέρουμε πως ο κόμβος μεταφέρει την επιδημία p_k . Ένας κόμβος που βρίσκεται σε υγιή κατάσταση θα συμβολίζεται με p_0 , ενώ ο κόμβος στην περισσότερο μολυσμένη με p_N . Τότε, για τις καταστάσεις των κόμβων, προφανώς θα ισχύει:

$$p_N > p_{N-1} > \dots > p_1 > p_0$$



Η παραπάνω ιεραρχία σημαίνει πως η επιδημία p_k είναι ισχυρότερη της p_m , για $k > m$. Ένας κόμβος μολυσμένος με την p_m είναι πιθανόν να ξαναμολυνθεί από την p_k , αφού η τελευταία είναι ισχυρότερη της πρώτης. Η επιδημική μετάλλαξη ή μεταστοιχείωση υποθέτει πως ένας μολυσματικός κόμβος i σε επίπεδο επιδημικής μόλυνσης p_k μπορεί να μολύνει έναν κόμβο j που βρίσκεται σε επίπεδο p_m στη «γειτονιά» του, εάν και μόνο εάν $p_k > p_m$, το οποίο σημαίνει πως ο j είναι είτε υγιής (p_0) είτε βρίσκεται μολυσμένος σε χαμηλότερο επίπεδο μόλυνσης. [4]

Ένας μολυσματικός κόμβος i , επιπέδου μόλυνσης p_k , μπορεί να ιαθεί με δύο τρόπους. Στην περίπτωση της πλήρους ίασης ο κόμβος σε ένα χρονικό βήμα μεταπίπτει στην υγιή κατάσταση p_0 με ρυθμό δ_{k0} . Στην περίπτωση μερικής ίασης ο κόμβος μεταπίπτει σε κατάσταση p_m ασθενέστερη της αρχικής του. Αυτό σημαίνει πως, ο κόμβος μπορεί να είναι ακόμα μολυσματικός, καθώς μετακινήθηκε σε κατάσταση ασθενέστερης επιδημίας. Πρέπει να αναφερθεί πως ο ρυθμός μετάπτωσης β_{km} εξαρτάται από τον αριθμό των γειτονικών του κόμβων, το επίπεδο μόλυνσής τους και τον ρυθμό μόλυνσης. Το διάγραμμα καταστάσεων για τους N κόμβους φαίνεται παρακάτω (σχ. 4).



Σχήμα 4 Διάγραμμα Πολύ-Επιδημικού Μοντέλου N καταστάσεων

2.3.1 Το προτεινόμενο μοντέλο

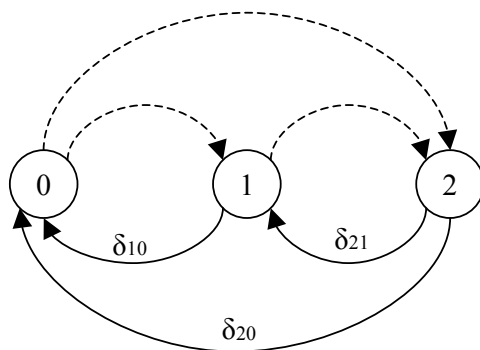
Στη διπλωματική αυτή περιγράφεται μια περίπτωση της πολύ-επιδημικής με 3 καταστάσεις για έναν κόμβο: Η υγιής κατάσταση που συμβολίζεται με 0 (susceptible), η κατάσταση μόλυνσης τύπου 1 (infected 1), ασθενέστερη της κατάστασης μόλυνσης τύπου 2 (infected 2), δηλαδή $N = 2$. Ένας κόμβος με κάποια πιθανότητα μπορεί να παραμείνει στην ίδια κατάσταση ή να περάσει σε κάποια άλλη διαφορετική. Μπορεί να υπάρξουν μεταβάσεις από την κατάσταση μόλυνσης 1 σε κατάσταση υγιή 0, από την κατάσταση μόλυνσης 2 σε



κατάσταση υγιή 0 ή από κατάσταση μόλυνσης 2 σε κατάσταση μόλυνσης 1 (μόλυνση ασθενέστερου επιπέδου ή μερική ίαση). Ένας κόμβος που βρίσκεται σε κατάσταση μόλυνσης 1 δεν μπορεί να μολύνει κόμβο που βρίσκεται σε κατάσταση μόλυνσης 2 και ούτε μπορεί να μολυνθεί κανένας κόμβος από μόλυνση του ίδιου επιπέδου.

Αρχικοποιείται με τις παραμέτρους: ρυθμός μόλυνσης β , ρυθμός ίασης δ_{21} από κατάσταση 2 σε 1, ρυθμός ίασης δ_{10} από κατάσταση 1 σε 0 και ρυθμός ίασης δ_{20} από κατάσταση 2 σε 0.

Στο σχήμα που ακολουθεί (σχ. 5) διακρίνονται αυτές οι καταστάσεις καθώς και οι τρόποι μετάβασης μεταξύ αυτών. Οι συνεχόμενες γραμμές απεικονίζουν ίασεις ενώ οι διακεκομμένες μολύνσεις. Όλες χαρακτηρίζονται από τους αντίστοιχους ρυθμούς ίασης ή μόλυνσης. Στον τύπο του επιδημικού ορίου $\lambda = \beta/\delta$, το δ ισούται με το άθροισμα των τριών ρυθμών ίασης του μοντέλου, δηλαδή $\delta = \delta_{10} + \delta_{20} + \delta_{21}$. Ισχύει πως αν $\lambda > 1$ τότε υπάρχει επιδημική επικράτηση, αλλιώς επιδημική εξασθένηση.



Σχήμα 5 Πολύ-Επιδημικό Μοντέλο με $N = 2$

Ο αλγόριθμος για τις δύο καταστάσεις μόλυνσης 1 και 2, λαμβάνοντας υπόψη την κινητικότητα των κόμβων του συστήματος, παρουσιάζεται παρακάτω:



Algorithm Epidemic for Node j at Time t

```
Input(Infection Probability  $\beta$ , Recovery Probability  $\delta_1$ ,  $\delta_2$ )
If (S[j] == INFECTED1 OR S[j] == INFECTED2) Then
  If (Random() <  $\delta_1$ )
    //έλεγχος αν υπάρχει πιθανότητα  $P(x) < \delta_1$  να γίνει ο κόμβος «υγιής» ή όχι
    S[j] <- SUSCEPTIBLE
  Else If ( $\delta_1 < \text{Random}() < \delta_2$ )
    S[j] <- INFECTED1
  Else
    For each node  $i=1, \dots, L$ , ( $i \neq j$ ) Do
      If( (S[i] == INFECTED2 & S[j] == INFECTED1)
        OR (S[i] != SUSCEPTIBLE & S[j] == SUSCEPTIBLE)
        AND  $\text{SQRT}((D[j,t,0] - D[i,t,0])^2 + (D[j,t,1] - D[i,t,1])^2) \leq R$  )
        // βρέθηκε κόμβος σε κατάσταση INFECTED2 και μπορεί να
        μολύνει κόμβο σε κατάσταση INFECTED1 με πιθανότητα  $\beta$  και εάν είναι μέσα
        στην ακτίνα μόλυνσης R (Euclidean Distance)
        Then
          If (Random() <  $\beta$ ) Then
            S[j] <- S[i]
          End If
        End If
      End For
    End If
  End For
End If
End Algorithm
```



Ο προτεινόμενος αλγόριθμος εκτελεί τα εξής βήματα:

Για έναν δεδομένο κόμβο j αρχικά γίνεται έλεγχος αν με τυχαία διαφορετική κάθε φορά πιθανότητα μπορεί να ιαθεί από κατάσταση 2 σε 1 (μόλυνση χαμηλότερου επιπέδου ή μερική ίαση) ή από 2 σε 0 (ίαση) ή από 1 σε 0 (ίαση).

Στη συνέχεια, για όλους τους γειτονικούς του κόμβους (κόμβοι που βρίσκονται εντός της ακτίνας «δράσης» του) ελέγχεται αν οι γείτονές του σε κατάσταση μόλυνσης 2 μπορούν να τον μολύνουν (ο έλεγχος γίνεται για αυτούς τους κόμβους j που δεν είναι στην κατάσταση μόλυνσης 2). Αν συμβεί μόλυνση επιπέδου 2, τότε ελέγχεται από την αρχή ο επόμενος κόμβος $j + 1$ διαφορετικά ελέγχεται αν κάποιοι από τους υπόλοιπους γείτονες τού j που βρίσκονται σε κατάσταση μόλυνσης 1 μπορούν να τον μολύνουν σε επίπεδο μόλυνσης 1. Αν μολυνθεί, τότε ελέγχεται ξανά αν μπορεί να μολυνθεί από τους γείτονές του που βρίσκονται σε κατάσταση μόλυνσης 2. Δηλαδή, ο δεδομένος κόμβος j μπορεί να αλλάξει διαδοχικά καταστάσεις περισσότερες από μία φορά.

Μια άλλη προσέγγιση του προτεινόμενου πολύ-επιδημικού αλγορίθμου είναι η μελέτη του στην περίπτωση που οι κόμβοι του συστήματος είναι ακίνητοι. Στην περίπτωση αυτή κάθε κόμβος, που παραμένει ακίνητος, «προσπαθεί» να μολύνει τους γειτονικούς του εφαρμόζοντας τον επιδημικό αλγόριθμο για μια σειρά επαναλήψεων (1000 βήματα). Ο αλγόριθμος για τις δύο καταστάσεις μόλυνσης 1 και 2 λαμβάνοντας υπόψη την ακινησία των κόμβων του συστήματος παρουσιάζεται παρακάτω:



Algorithm Epidemic for Node j at Time t

```
Input(Infection Probability  $\beta$ , Recovery Probability  $\delta_1, \delta_2$ )
For each k <= αριθμού επαναλήψεων Do
  If (S[ j ] == INFECTED1 OR S[ j ] == INFECTED2) Then
    If (Random() <  $\delta_1$ )
      //έλεγχος αν υπάρχει πιθανότητα  $P(x) < \delta_1$  να γίνει ο κόμβος «υγιής» ή όχι
      S[ j ] <- SUSCEPTIBLE
    Else If ( $\delta_1 < \text{Random}() < \delta_2$ )
      S[ j ] <- INFECTED1
    Else
      For each node i=1,...,L, (i not j) Do
        If( (S[ i ] == INFECTED2 & S[ j ] == INFECTED1)
          OR (S[ i ] != SUSCEPTIBLE & S[ j ] == SUSCEPTIBLE)
          AND SQRT((D[j,t,0] - D[i,t,0])2 + (D[j,t,1] - D[i,t,1])2) <= R )
          // βρέθηκε κόμβος σε κατάσταση INFECTED2 και μπορεί να
          μολύνει κόμβο σε κατάσταση INFECTED1 με πιθανότητα  $\beta$  και εάν είναι μέσα
          στην ακτίνα μόλυνσης R (Euclidean Distance)
          Then
            If (Random() <  $\beta$ ) Then
              S[ j ] <- S[ i ]
            End If
          End If
        End For
      End For
    End If
  End For
End For
End Algorithm
```

Ο αλγόριθμος χωρίς κινητικότητα εκτελεί τα ίδια βήματα με τον αυτόν που αφορά την κινητικότητα με τη μόνη διαφορά πως επαναλαμβάνει για συγκεκριμένο αριθμό επαναλήψεων τον αλγόριθμο με τις ίδιες πάντα συντεταγμένες κόμβων.



3. Εφαρμογή Επιδημικής και Πολύ-Επιδημικής σε Περιβάλλοντα Διάχυτου Υπολογισμού

Το πρόγραμμα myepidemic.java που υλοποιεί τον αλγόριθμο που περιγράψαμε παραπάνω έχει φτιαχτεί με τις εξής προδιαγραφές: Δέχεται οκτώ ορίσματα, τα οποία κατά σειρά είναι

1. Το πλήθος των κόμβων του δικτύου
2. Το σετ συντεταγμένων που θέλουμε να ξεκινήσουμε την υλοποίηση του αλγορίθμου
3. Το αρχείο με τις συντεταγμένες των κόμβων
4. Ο ρυθμός μόλυνσης β των κόμβων (θεωρείται σταθερός για όλες τις μεταβάσεις καταστάσεων σε μία εξεταζόμενη περίπτωση)
5. Ο ρυθμός ίασης ενός κόμβου δ_{10} από την κατάσταση 1 στην 0 (διαφορετικός ανά εξεταζόμενη περίπτωση)
6. Ο ρυθμός ίασης ενός κόμβου δ_{20} από την κατάσταση 2 στην 0 (διαφορετικός ανά εξεταζόμενη περίπτωση)
7. Ο ρυθμός μερικής ίασης δ_{21} ενός κόμβου από την κατάσταση 2 στην 1 (διαφορετικός ανά εξεταζόμενη περίπτωση)
8. Η ακτίνα ελέγχου R της γειτονιάς του κόμβου

Το πλήθος των κόμβων (όρισμα 1) έχει δηλωθεί σταθερό καθ' όλη τη διάρκεια της συγκεκριμένης μελέτης 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 θέσεων 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. Ο ρυθμός μόλυνσης των κόμβων έχει θεωρηθεί σταθερός για όλες τις μεταβάσεις καταστάσεων σε μια εξεταζόμενη περίπτωση. Πράγμα που δεν συμβαίνει με τους ρυθμούς ίασης των κόμβων δ_{10} , δ_{20} και δ_{21} . Η ακτίνα δράσης ενός κόμβου παίρνει τιμές 2, 5 ή 10 ανά περίπτωση. Η περιοχή προσομοίωσης στην j-sim είχε τεθεί σαν τετράγωνο πλευράς 100 μονάδων.

Για ευκολία έχει δημιουργηθεί το αρχείο runEpidemic.bat για τη γρήγορη εκτέλεση του κώδικα. Το περιεχόμενο του αρχείου παρατίθεται στο Παράρτημα 2 της διπλωματικής εργασίας.



Από τη στιγμή που έχουν καθοριστεί πλήρως τα ορίσματα το πρόγραμμα myepidemic.java εκτελεί τα βήματα:

1. 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
2. Από το βήμα συντεταγμένων της τιμής του ορίσματος 2 ξεκινά να εφαρμόζει την κλάση myEpidemic(), που αποτελεί τον επιδημικό αλγόριθμο του προγράμματος
3. Παράγει αρχεία εξόδου για την καλύτερη αξιοποίηση των αποτελεσμάτων της εφαρμογής του επιδημικού αλγορίθμου. Αυτά είναι:
 - 3.1. Αρχείο out0.csv. Περιέχει μόνο τους κόμβους σε κάθε βήμα (από 101 έως 999) που είναι υγιείς. Βασικό αρχείο καταγραφής της υγιούς κατάστασης των κόμβων.
 - 3.2. Αρχείο out1.csv. Περιέχει μόνο τους κόμβους σε κάθε βήμα (από 101 έως 999) που έχουν μόλυνση τύπου 1. Βασικό αρχείο καταγραφής της κατάστασης 1.
 - 3.3. Αρχείο out2.csv. Περιέχει μόνο τους κόμβους σε κάθε βήμα (από 101 έως 999) που έχουν μόλυνση τύπου 2. Βασικό αρχείο καταγραφής της κατάστασης 2.
 - 3.4. Αρχείο out4.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.
 - 3.5. Αρχείο out5.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 διπλωματικής εργασίας με το πρόγραμμα MSExcel 2002.
 - 3.6. Αρχείο results100.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 τον έλεγχο του αλγορίθμου.

Για το αρχείο myepidemicnomob.java ισχύουν αντίστοιχες παρατηρήσεις.



3.1 Το προτεινόμενο μοντέλο με/χωρίς κινητικότητα κόμβων (RWP)

Τα αποτελέσματα χωρίζονται σε δύο κατηγορίες. Στην κατηγορία *Με Κινητικότητα Κόμβων* (ΜΚΚ) και σε αυτή *Χωρίς Κινητικότητα Κόμβων* (ΧΚΚ).

Τα σενάρια που χρησιμοποιήθηκαν στην προσομοίωση για τους ακίνητους αλλά και για τους κινητούς κόμβους σε σχέση με το πλήθος των μολυσμένων κόμβων είναι δύο. Για συνολικά εκατό κόμβους, στο πρώτο σενάριο 10 κόμβοι (ποσοστό 1/10) δηλώθηκαν σε κατάσταση μολυσμένου τύπου 2, 20 κόμβοι (ποσοστό 1/5) μολυσμένου τύπου 1 ενώ οι υπόλοιποι 70 θεωρήθηκαν υγιείς. Στο δεύτερο σενάριο, αντίστοιχα, έχουμε: 1 κόμβο (ποσοστό 1/100) μολυσμένου τύπου 2, 10 κόμβους (ποσοστό 1/10) μολυσμένου τύπου 1 και οι υπόλοιποι 89 υγιείς. Στη συνέχεια της διπλωματικής θα αναφέρεται το πρώτο σαν *Σενάριο Κόμβων 1* (ΣΚ1) (10, 20, 70) και το δεύτερο σαν *Σενάριο Κόμβων 2* (ΣΚ2) (1, 10, 89), αντίστοιχα.

Σε κάθε γράφημα διακρίνονται στον τίτλο του οι τιμές του ρυθμού μόλυνσης, των ρυθμών ίασεων καθώς και της ακτίνας δράσης του κόμβου.

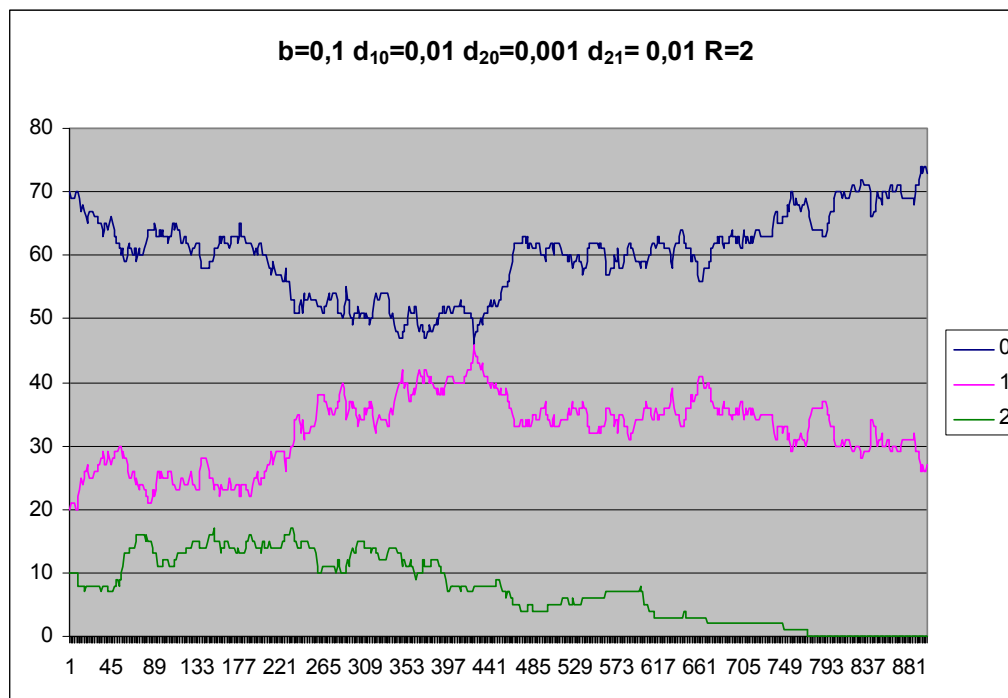
3.1.1 Επιδημική επικράτηση

Με Κινητικότητα Κόμβων και ΣΚ1

Η περίπτωση που μελετάται είναι για τιμές ρυθμού μόλυνσης $\beta = 0,1$, ίασης $\delta_{10} = \delta_{21} = 0,01$ και $\delta_{20} = 0,001$ και ακτίνας $R = 2, 5$ και 10 αντίστοιχα.



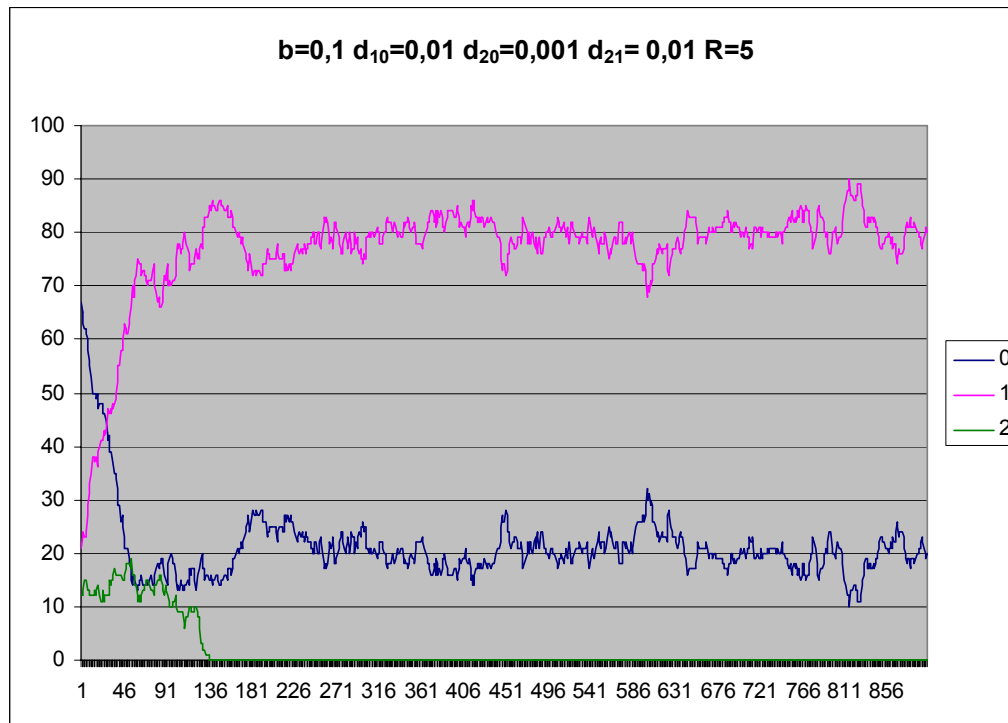
Για τις δοθείσες τιμές ισχύει $\lambda > 1$, οπότε περιμένουμε να εκδηλωθεί επιδημική επικράτηση. Στην περίπτωση του γραφήματος 1 παρατηρούμε για τιμή ακτίνας $R = 2$ πως το σύστημα τελικά ισορροπεί μεταξύ της μόλυνσης 1 και της υγιούς κατάστασης των κόμβων, που είναι τελικά και η επικρατούσα κατάσταση. Η μόλυνση 2 εξαφανίζεται μετά από περίπου 780 βήματα. Παρουσιάζεται μικρή αύξηση του πλήθους των κόμβων στις δύο επικρατούσες καταστάσεις 1 και 0 από το μίαισμα των κόμβων της μόλυνσης επιπέδου 2, η οποία εξαφανίζεται μετά το βήμα 780.



Γράφημα 1 ΜΚΚ και ΣΚ1, $b=0,1 d_{10}=0,01 d_{20}=0,001 d_{21}=0,01 R = 2$



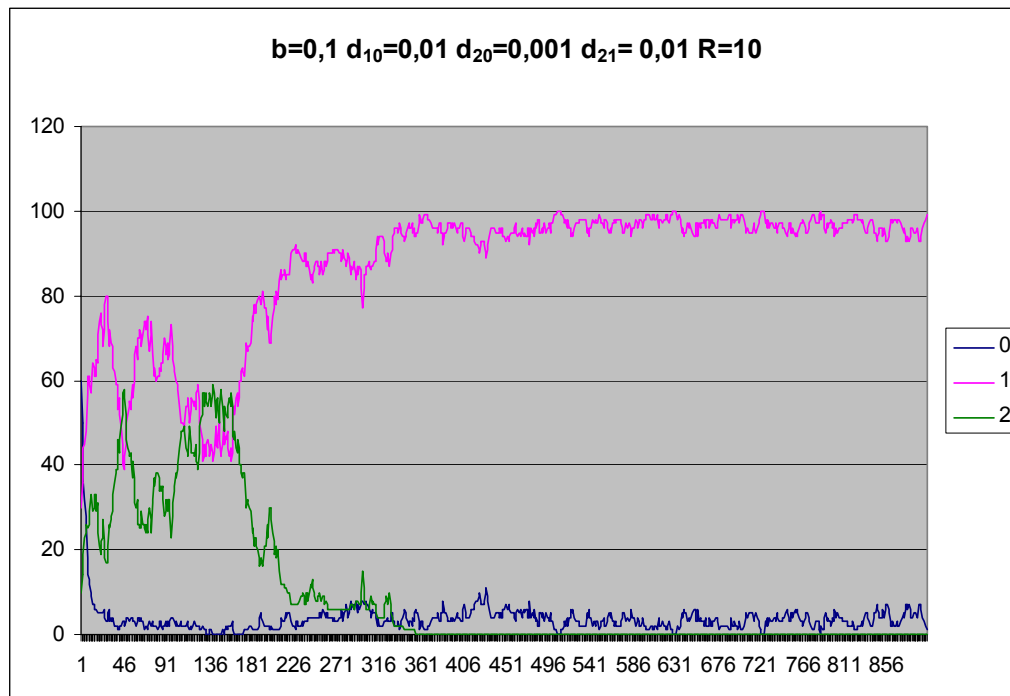
Στην περίπτωση που η ακτίνα είναι $R = 5$ (γράφημα 2) παρατηρούμε σχετικά γρήγορη εξαφάνιση (βήμα 135) της επιδημίας 2 και ισορροπία των καταστάσεων 1 και 0. Όμως, φαίνεται καθαρά πως η επιδημία 1 επικρατεί σε ποσοστό περίπου 80% και το σύστημα παραμένει σε μολυσμένη κατάσταση. Επίσης παρατηρείται η απότομη μείωση του αριθμού των υγιών κόμβων και η αύξηση των μολυσμένων επιπέδου 1 (βήμα 75).



Γράφημα 2 ΜΚΚ και ΣΚ1, $b=0,1$ $d_{10}=0,01$ $d_{20}=0,001$ $d_{21}=0,01$ $R = 5$



Σε αυτή την περίπτωση (ακτίνα $R = 10$, γράφημα 3) παρατηρείται η τάση του συστήματος για σχεδόν καθολική επικράτηση της κατάστασης 1 (ποσοστό 95%). Επίσης, παρατηρείται και πιο απότομη αλλά και πιο γρήγορη (βήμα 30) μείωση των υγιών κόμβων σε σχέση με την περίπτωση της ακτίνας $R = 5$. Η μόλυνση 2 είναι πιο ανθεκτική από την προηγούμενη περίπτωση (ακτίνα $R = 5$), χωρίς όμως να επικρατεί. Παρουσιάζει, ωστόσο, αρχικά μία τάση αυξητική, η οποία περίπου από το βήμα 140 μετατρέπεται σε μείωση έως την ολοκληρωτική εξάλειψή της (βήμα 360).

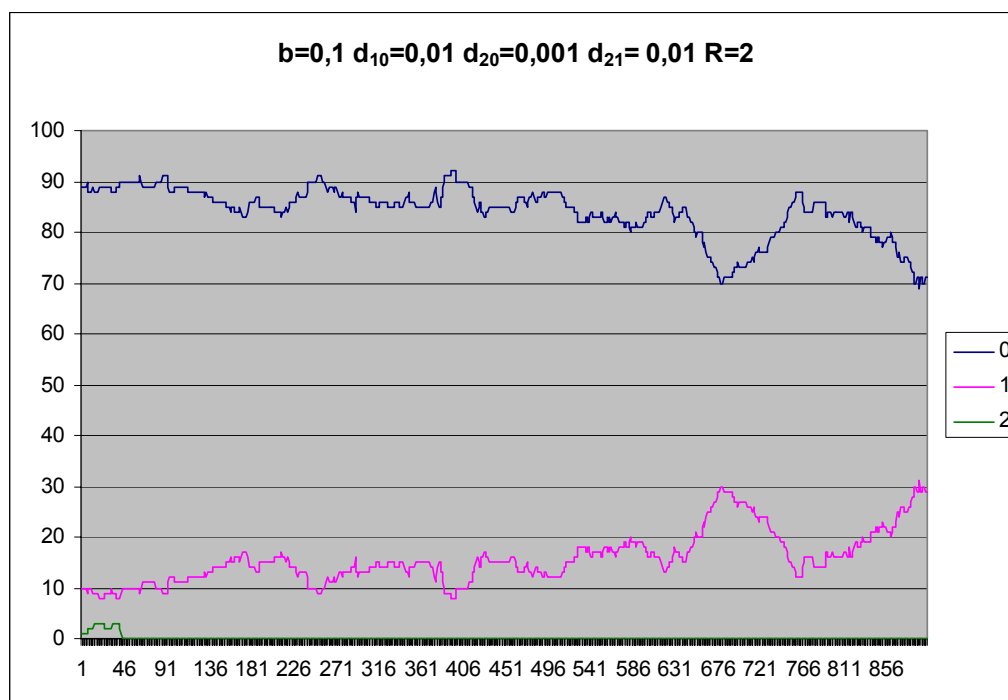


Γράφημα 3 ΜΚΚ και ΣΚ1, $b=0,1$ $d_{10}=0,01$ $d_{20}=0,001$ $d_{21}=0,01$ $R = 10$



Με Κινητικότητα Κόμβων και ΣΚ2

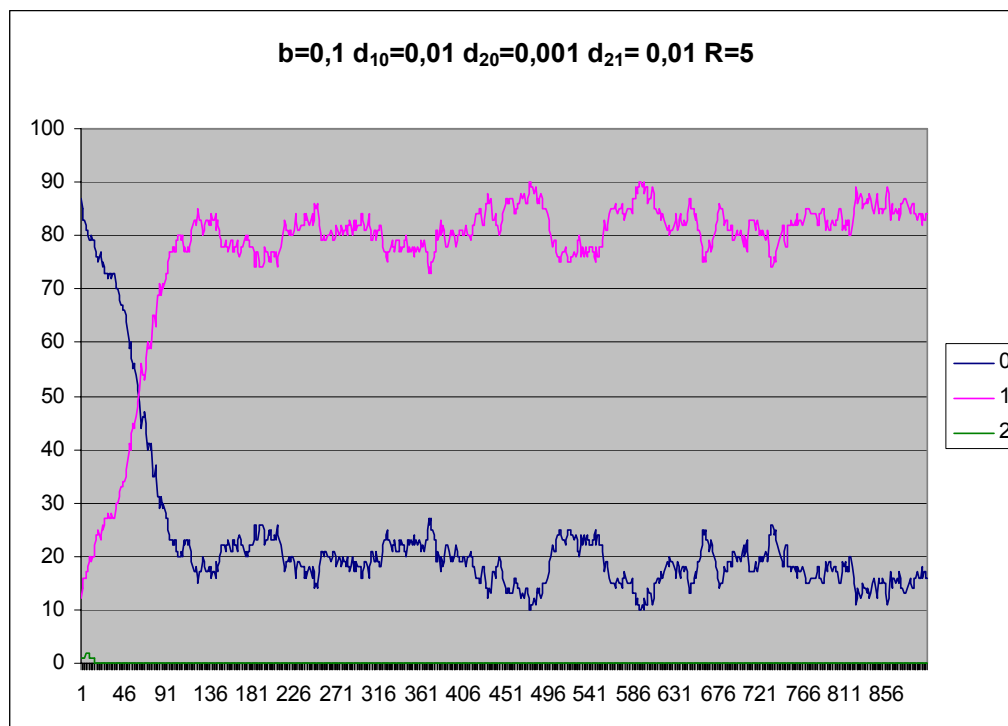
Στο σενάριο αυτό υπάρχει μόνο ένας κόμβος σε μόλυνση επιπέδου 2. Όπως φαίνεται και στο παρακάτω γράφημα 4 για την περίπτωση της ακτίνας $R = 2$, η επιδημία 2 εξαφανίζεται ολοκληρωτικά στο βήμα 50 (που αντιστοιχεί σε γρηγορότερο βήμα από το αντίστοιχο διάγραμμα 1 του ΣΚ1 για την ίδια ακτίνα). Υπάρχει μία τάση επικράτησης της υγιούς κατάστασης κόμβων στην ισορροπία που φτάνει το σύστημα, αλλά με εμφανή μείωση του αριθμού τους (και ταυτόχρονη αύξηση του αριθμού της επιδημίας 1) σε σχέση με το διάγραμμα 1 του ΣΚ1 για την ίδια ακτίνα.



Γράφημα 4 ΜΚΚ και ΣΚ2, b=0,1 d₁₀=0,01 d₂₀=0,001 d₂₁=0,01 R = 2



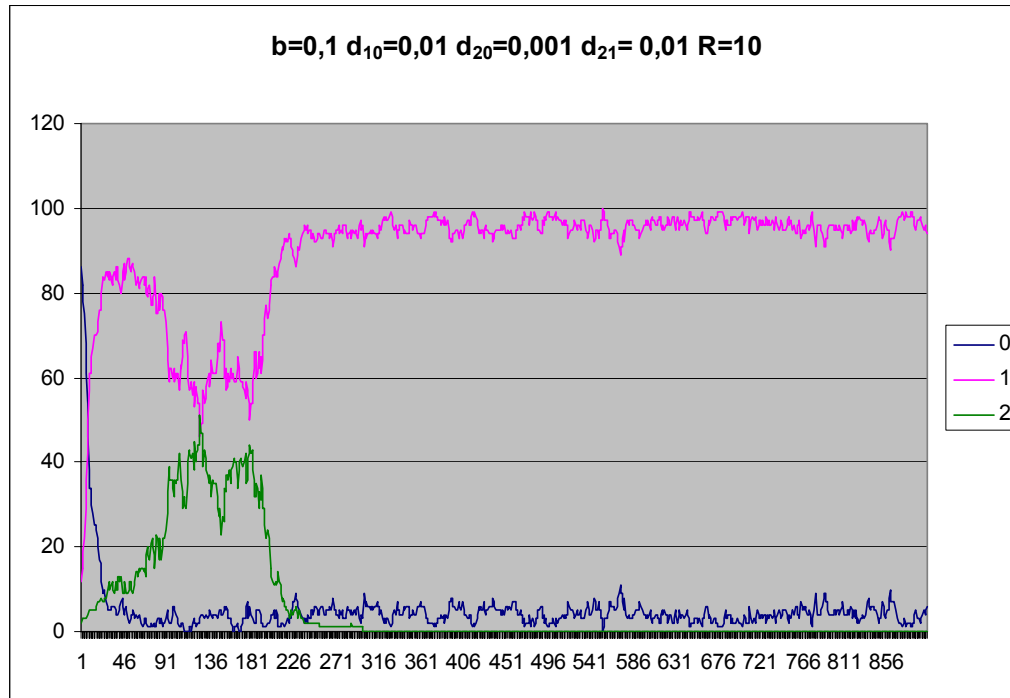
Στην περίπτωση για ακτίνα $R = 5$ του ΣΚ2, η οποία παρουσιάζεται στο διάγραμμα 5, είναι αντιληπτή η σχεδόν άμεση εξάλειψη της επιδημίας 2 και η επικράτηση της επιδημίας 1 σε ποσοστό περίπου ίδιο (80%) με αυτό του ΣΚ1 για την ίδια ακτίνα. Παράλληλα η επικράτηση αυτή συμβαίνει σε γρηγορότερο βήμα (περίπου στο 110) σε σχέση με την αντίστοιχη περίπτωση του ΣΚ1 (διάγραμμα 2).



Γράφημα 5 ΜΚΚ και ΣΚ2, $b=0,1$ $d_{10}=0,01$ $d_{20}=0,001$ $d_{21}=0,01$ $R = 5$



Η τελευταία περίπτωση για ακτίνα $R = 10$ (διάγραμμα 6) έχει εξαιρετική ομοιότητα με την αντίστοιχη περίπτωση του ΣΚ1. Πράγμα που μας κάνει να υποθέσουμε πως ανεξάρτητα από το πλήθος των μολυσμένων κόμβων επιπέδου 2 το σύστημα τελικά θα ισορροπήσει μεταξύ της σχεδόν καθολικά επικρατούσας επιδημίας 1 (ποσοστό 95%) και της υγιούς κατάστασης (ποσοστό 5%). Η επιδημία 2 αυξάνει και μειώνεται περίπου με τον ίδιο ρυθμό όπως στο αντίστοιχο γράφημα 3 του ΣΚ1.



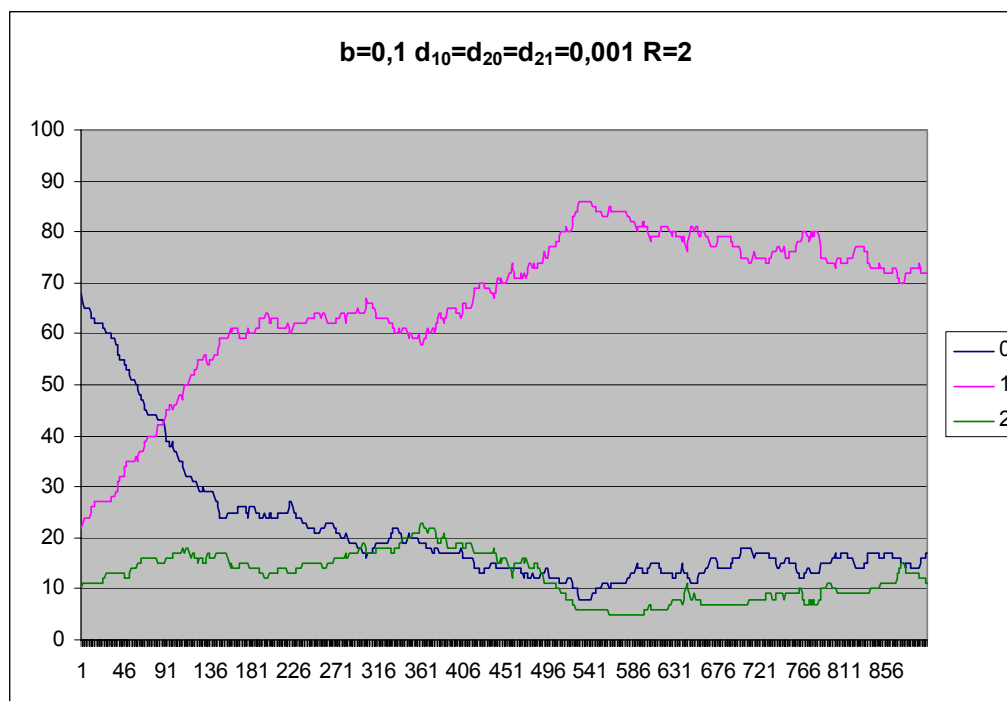
Γράφημα 6 ΜΚΚ και ΣΚ2, b=0,1 d₁₀=0,01 d₂₀=0,001 d₂₁=0,01 R = 10



Με Κινητικότητα Κόμβων και ΣΚ1

Στην επόμενη περίπτωση μελετάται για το ΣΚ1 (πλήθος κόμβων 70 20 10) η συμπεριφορά της επιδημικής μόλυνσης για τιμές ρυθμών ίασης ίσες με 0,001, ρυθμό μόλυνσης $\beta = 0,1$ και για τις ίδιες ακτίνες δράσης των κόμβων 2, 5, 10.

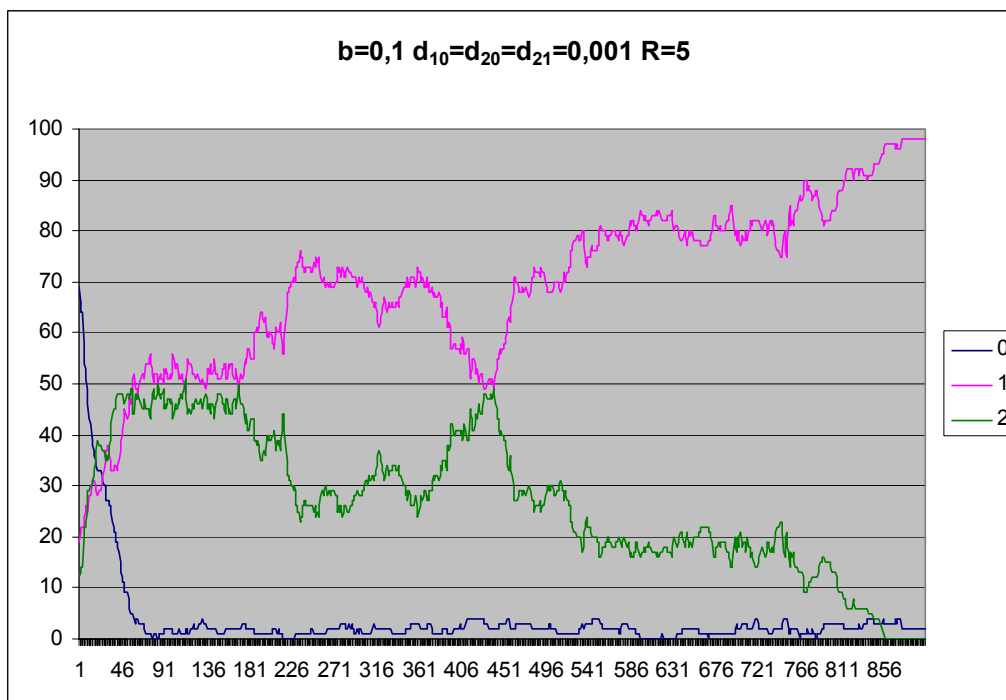
Για τις δοθείσες τιμές ισχύει $\lambda > 1$, οπότε περιμένουμε να εκδηλωθεί επιδημική επικράτηση. Στο επόμενο διάγραμμα (γράφημα 7) παρατηρούμε πως για τιμή ακτίνας $R = 2$ στο σύστημα παραμένουν και οι τρεις καταστάσεις με επικρατούσα την επιδημία 1. Μετά το βήμα 540 επέρχεται ισορροπία για τις τρεις καταστάσεις.. Αξιοσημείωτο γεγονός είναι η διατήρηση της επιδημίας 2 γεγονός που μάλλον πρέπει να οφείλεται στον συνδυασμό μικρής ακτίνας δράσης των κόμβων και πολύ μικρού ρυθμού ίασης.



Γράφημα 7 ΜΚΚ και ΣΚ1, $b=0,1$ $d_{10}=d_{20}=d_{21}=0,001$ $R = 2$



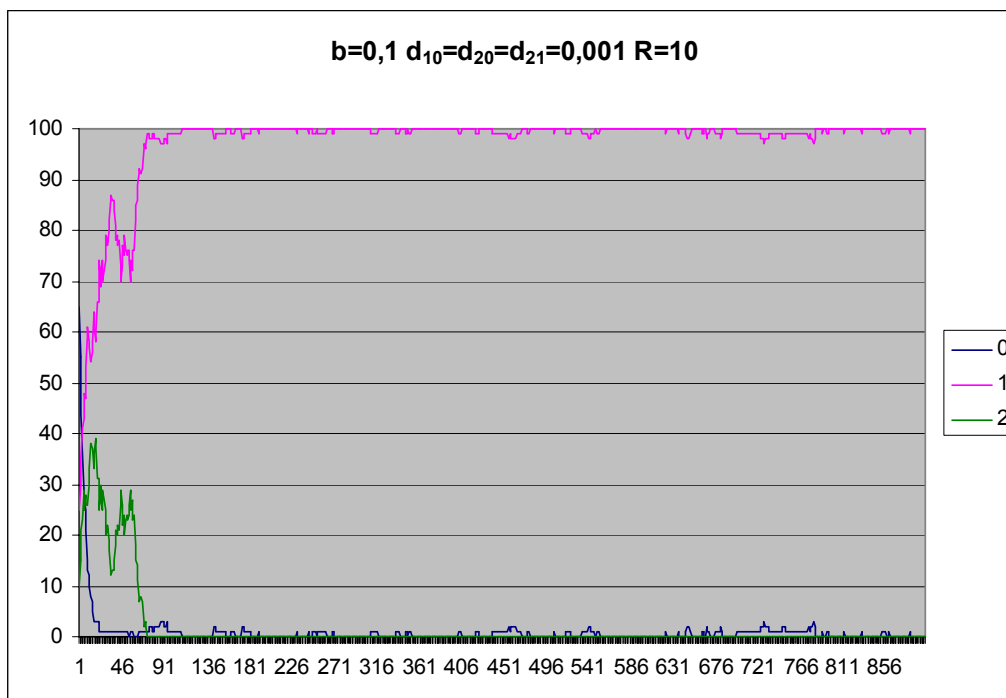
Στο γράφημα 8 (ακτίνα $R = 5$) διακρίνεται καθαρά η απότομη μείωση των υγιών κόμβων και η σταθερή επικράτηση της επιδημίας 1. Επίσης, η κατάσταση μόλυνσης 2 παρουσιάζει βραδεία μείωση μέχρι εξάλειψής της περίπου στο βήμα 850. Στο χρονικό διάστημα που η κατάσταση 0 μειώνεται (περίπου μέχρι το βήμα 80) η αύξηση των επιδημιών 1 και 2 είναι ταυτόχρονη με τον ίδιο ρυθμό. Στη συνέχεια επικρατεί μία σταθερότητα μεταξύ τους (μέχρι το βήμα 180) και ακολουθεί μια παράλληλη αυξομείωση του αριθμού τους (οριζόντιος άξονας συμμετρίας στο 50).



Γράφημα 8 ΜΚΚ και ΣΚ1, $b=0,1$ $d_{10}=d_{20}=d_{21}=0,001$ $R = 5$



Η περίπτωση της ακτίνας 10 είναι αρκετά αξιοσημείωτη. Λόγω της μεγάλης δράσης του καθενός κόμβου φαίνεται στο γράφημα 9 η καθολική επικράτηση της μόλυνσης 1 (βήμα 80). Η αστάθεια στο σύστημα διαρκεί περίπου τόσα βήματα όσο και η ταυτόχρονη αυξητική μολυσματική συμπεριφορά για τις μολύνσεις 1 και 2 που παρατηρήθηκε στο γράφημα 8. Στη συνέχεια η μόλυνση τύπου 2 εξαφανίζεται.



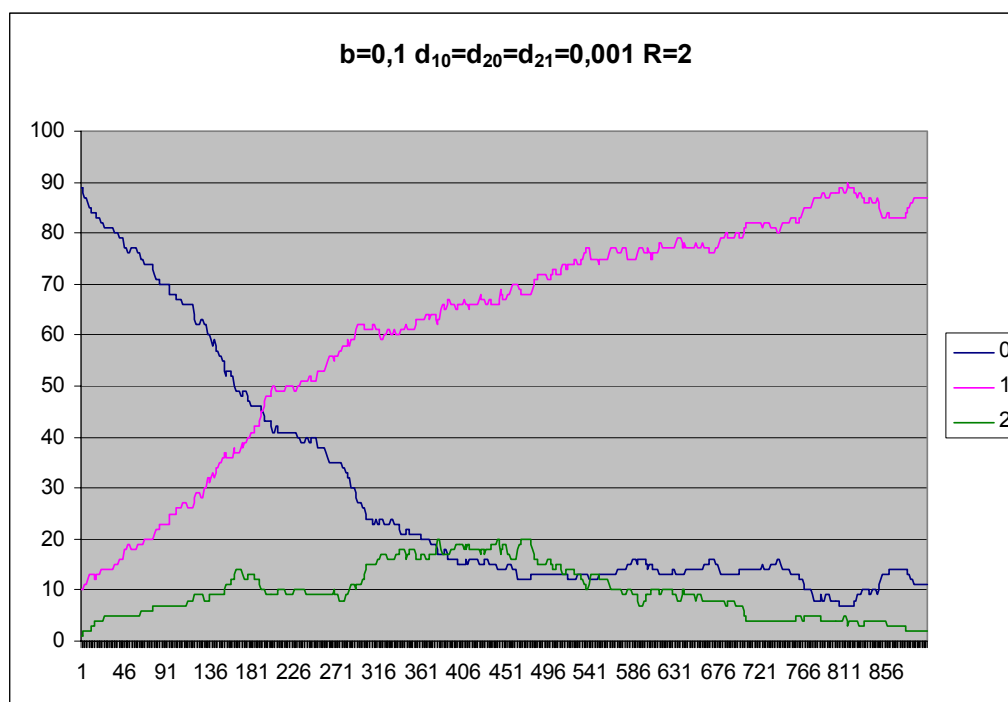
Γράφημα 9 MKK και ΣΚ1, $b=0,1$ $d_{10}=d_{20}=d_{21}=0,001$ $R = 10$



Με Κινητικότητα Κόμβων και ΣΚ2

Στην επόμενη περίπτωση μελετάται για το ΣΚ2 (πλήθος κόμβων 89 10 1) η συμπεριφορά της επιδημικής μόλυνσης για τιμές ρυθμών ίασης ίσες με 0,001, ρυθμό μόλυνσης $\beta = 0,1$ και για τις ίδιες ακτίνες δράσης των κόμβων 2, 5, 10.

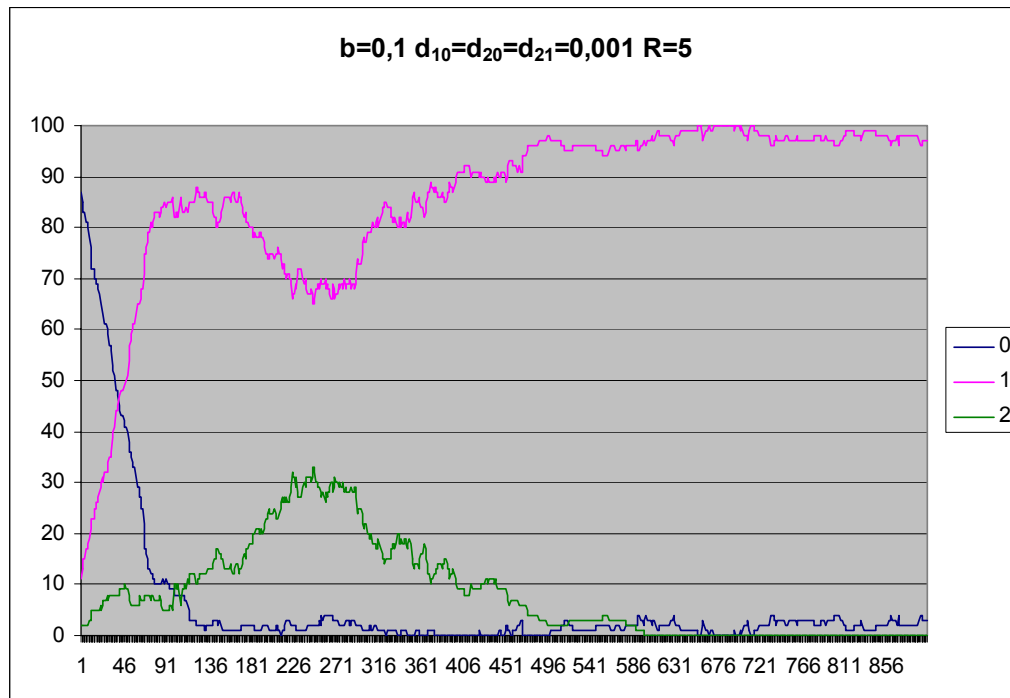
Για τις δοθείσες τιμές ισχύει $\lambda > 1$, οπότε περιμένουμε να εκδηλωθεί επιδημική επικράτηση. Στο επόμενο διάγραμμα (γράφημα 10) φαίνεται πως για την τιμή ακτίνας $R = 2$ στο σύστημα παραμένουν και οι τρεις καταστάσεις με τελικά επικρατούσα την επιδημία 1. Η επιδημία 2 αρχικά αυξάνεται φτάνοντας στο μέγιστό της στο βήμα 430 ενώ στη συνέχεια μειώνεται.



Γράφημα 10 MKK και ΣΚ2, b=0,1 d₁₀=d₂₀=d₂₁=0,001 R = 2



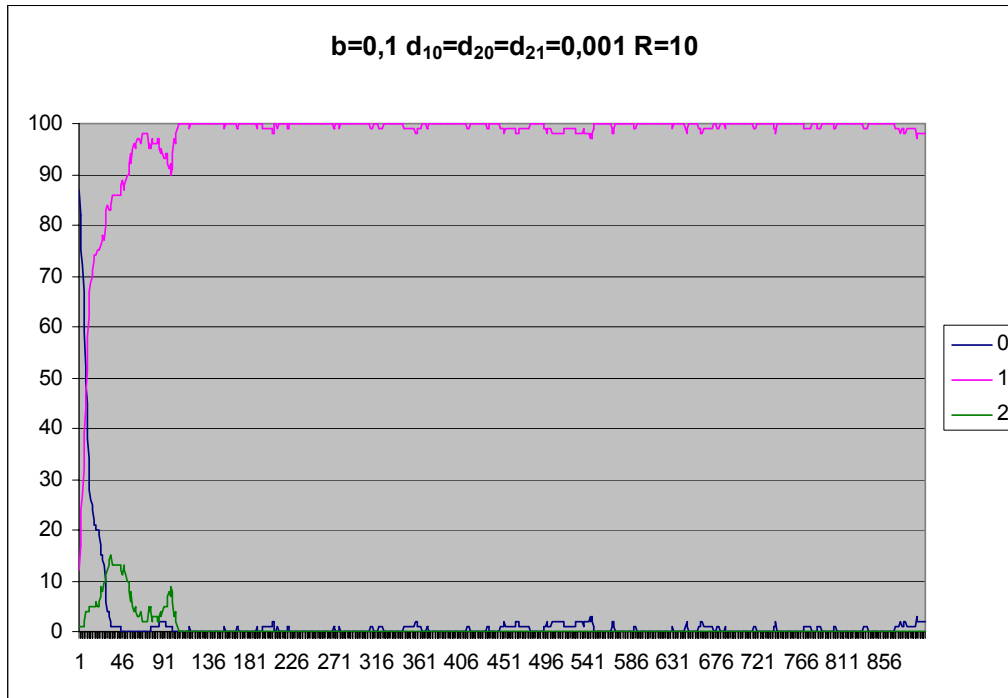
Η περίπτωση του γραφήματος 11 (ακτίνα $R = 5$) παρουσιάζει ομοιότητα με το γράφημα 10. Η μόνη διαφορά τους είναι πως στο γράφημα 11 έχουν μετατοπιστεί προς τα αριστερά του οριζόντιου άξονα, το σημείο τομής της επιδημίας 1 με την κατάσταση 0 και το μέγιστο της επιδημίας 2. Τελικά το σύστημα φτάνει στην ισορροπία μετά το βήμα 630, έχοντας εξαλειφθεί η επιδημία 2.



Γράφημα 11 MKK και ΣΚ2, $b=0,1$ $d_{10}=d_{20}=d_{21}=0,001$ $R = 5$



Στο τρέχον γράφημα 12 (ακτίνα δράσης $R = 10$) η προς τα αριστερά μετατόπιση της ισορροπίας του συστήματος (βήμα 120) είναι ακόμα εμφανέστερη σε σχέση με τις δύο προηγούμενες περιπτώσεις για $R = 2$ και 5. Οπότε μπορούμε να συνάγουμε το συμπέρασμα πως με εξαιρετικά μικρούς ρυθμούς ίασης το σύστημα σίγουρα θα μείνει σε κατάσταση ολοκληρωτικής μόλυνσης, ενώ δεν πρέπει να αγνοηθεί και η συνεισφορά της επίδρασης της ακτίνας δράσης των κόμβων ως προς το χρονικό σημείο στο οποίο θα επέλθει η ισορροπία αυτή.



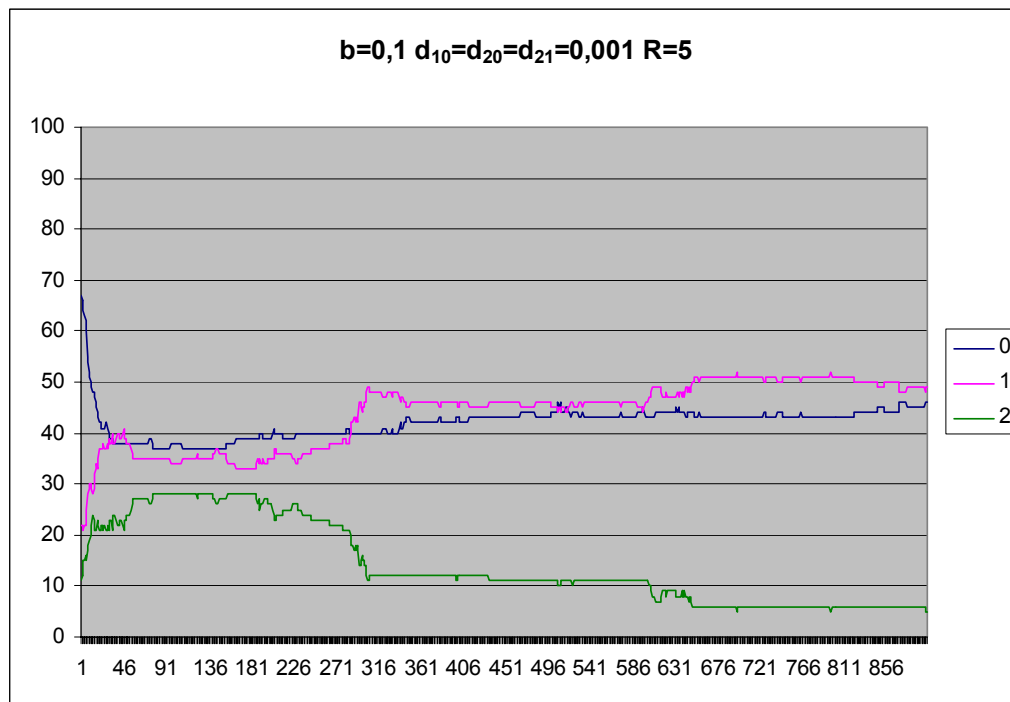
Γράφημα 12 ΜΚΚ και ΣΚ2, $b=0,1$ $d_{10}=d_{20}=d_{21}=0,001$ $R = 10$



Χωρίς Κινητικότητα Κόμβων και ΣΚ1

Η περίπτωση που μελετάται είναι για ρυθμό μόλυνσης $\beta = 0,1$, ίασης $d_{10} = d_{21} = d_{20} = 0,001$ για τιμή ακτίνας $R = 5$ μόνο. Δηλαδή, έχει επιλεγθεί μία ακτίνα «μέσης» τιμής.

Για τις δοθείσες τιμές ισχύει $\lambda > 1$, οπότε περιμένουμε να εκδηλωθεί και σε αυτή την περίπτωση επιδημική επικράτηση. Στο διάγραμμα που ακολουθεί (γράφημα 13) παρατηρούμε για τιμή ακτίνας $R = 5$ πως και οι τρεις καταστάσεις συνεχίζουν να υπάρχουν. Μετά το βήμα 320 η επιδημία 1 επικρατεί με υπερδιπλασιασμό του αρχικού της ποσοστού και όλες οι επιδημίες βρίσκονται σε σχετική ισορροπία. Η επιδημία 2 αρχικά αυξάνει τα ποσοστά μόλυνσης τελικά όμως έρχεται σε ισορροπία περίπου στο μισό ποσοστό του αρχικού της. Η υγιής κατάσταση μετά την απότομη πτώση που πλήθους κόμβων που είχε (βήμα 35) σταθεροποιείται.

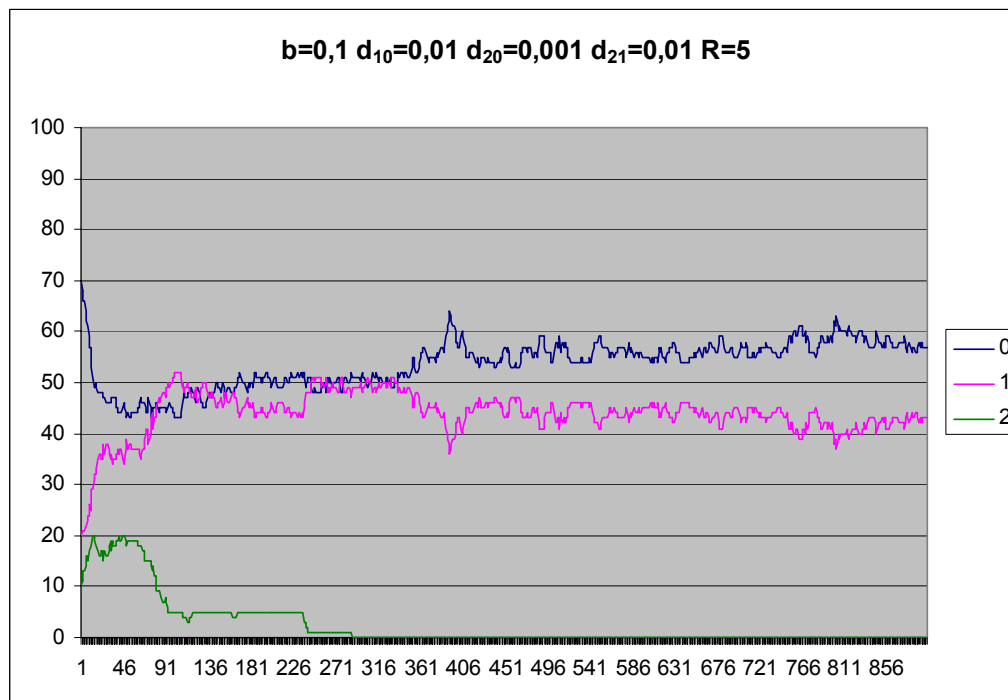


Γράφημα 13 ΧΚΚ και ΣΚ1, $b=0,1$ $d_{10}=d_{20}=d_{21}=0,001$ $R = 5$



Η περίπτωση που μελετάται είναι για ρυθμό μόλυνσης $\beta = 0,1$, ίασης $\delta_{10} = \delta_{21} = 0,01$
και $\delta_{20} = 0,001$ για τιμή ακτίνας $R = 5$.

Για τις δοθείσες τιμές ισχύει $\lambda > 1$, οπότε περιμένουμε να εκδηλωθεί και σε αυτή
την περίπτωση επιδημική επικράτηση. Στο γράφημα 14 παρατηρούμε ότι η επιδημία 1
αυξάνει το αρχικό ποσοστό της (περίπου 45%) το οποίο παραμένει σταθερό μετά το βήμα
400. Η επιδημία 2 έχει παρόμοια συμπεριφορά με αυτή που φάνηκε στο γράφημα 13 αλλά
σε μικρότερο αριθμό βημάτων. Η υγιής κατάσταση επικρατεί με μείωση όμως του αρχικού
της ποσοστού.



Γράφημα 14 ΧΚΚ και ΣΚ1, $b=0,1$ $d_{10}=0,01$ $d_{20}=0,001$ $d_{21}=0,01$ $R = 5$

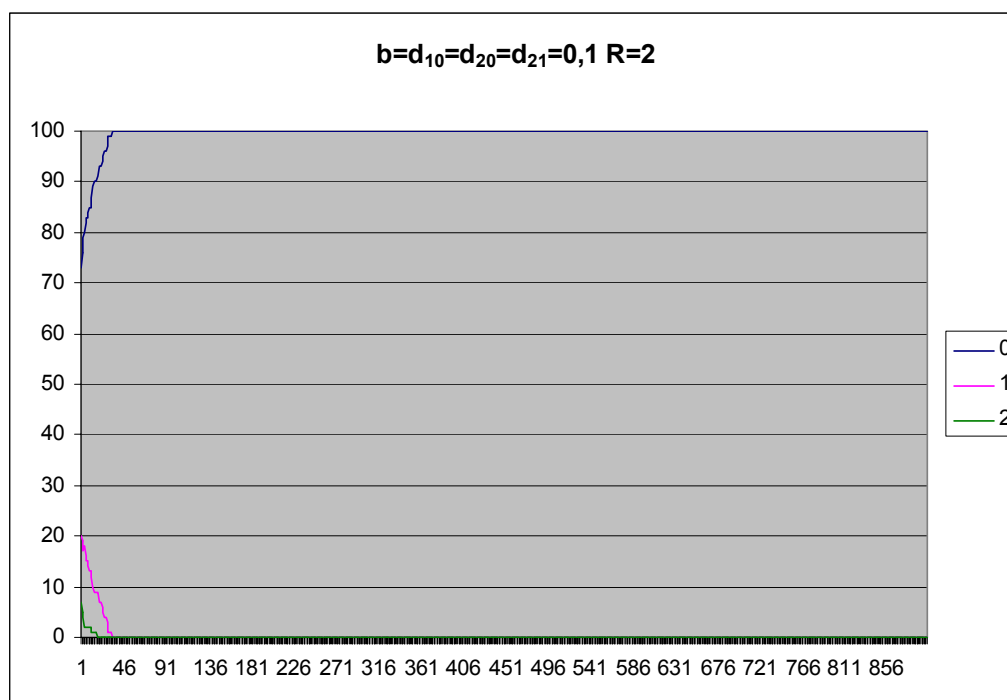


3.1.2 Επιδημική αποσύνθεση

Με κινητικότητα Κόμβων και ΣΚ1

Στην επόμενη περίπτωση μελετάται για το ΣΚ1 η συμπεριφορά της επιδημικής μόλυνσης για όλες τις τιμές ρυθμών ίασης ίσες με τον ρυθμό μόλυνση ίσες με 0,1 και για ακτίνες δράσης των κόμβων 2, 5, 10.

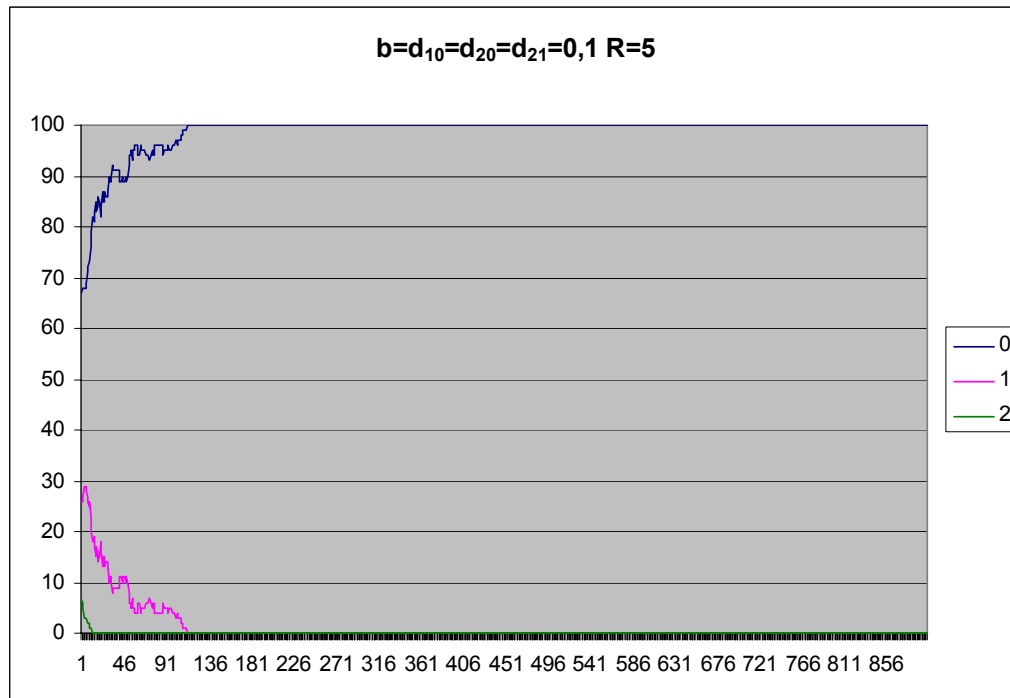
Για τις δοθείσες τιμές ισχύει $\lambda < 1$, οπότε περιμένουμε να εκδηλωθεί επιδημική αποσύνθεση του συστήματος. Στο επόμενο διάγραμμα (γράφημα 15) φαίνεται πως για την τιμή ακτίνας $R = 2$ το σύστημα φτάνει σε κατάσταση ίασης και παραμένει σε αυτή σχεδόν στα πρώτα βήματα εκτέλεσης (βήμα 40). Η μικρή ακτίνα δράσης των κόμβων δεν βοηθά να συντηρηθεί η επιδημία με αποτέλεσμα να εξαφανιστούν και οι δύο επιδημίες κατάστασης 1, 2 παρά τον σχετικά αρχικά μεγάλο αριθμό κόμβων που έχουμε και για τις δύο τους (συνολικά 30 κόμβοι).



Γράφημα 15 MKK και ΣΚ1, $b=d_{10}=d_{20}=d_{21}=0,1 R = 2$



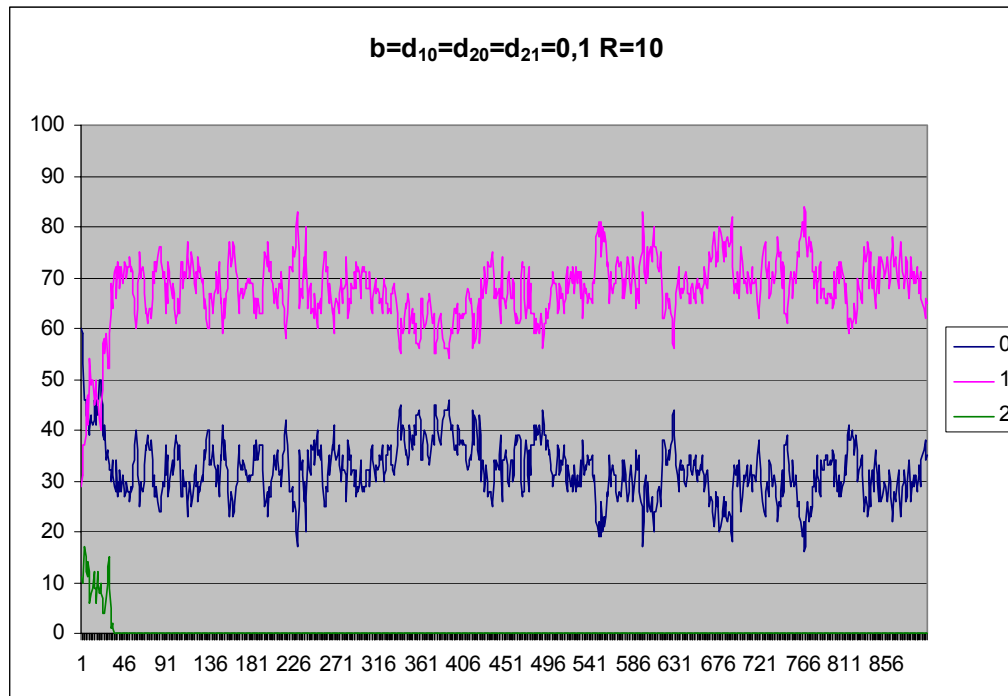
Στην περίπτωση, τώρα, που η ακτίνα δράσης γίνεται $R = 5$ η επιδημική αποσύνθεση εκδηλώνεται με το γράφημα 16. Παρατηρούμε πως η μικρή αύξηση ακτίνας απλά καθυστερεί την εκδήλωση ισορροπίας του συστήματος (βήμα 120) σε σχέση με την προηγούμενη περίπτωση. Το σύστημα έχει ιαθεί και δεν παρατηρείται απολύτως καμία μεταβολή στη συμπεριφορά του.



Γράφημα 16 MKK και ΣΚ1, $b=d_{10}=d_{20}=d_{21}=0,1 R = 5$



Στο γράφημα 17 που ακολουθεί παρατηρούμε για την τιμή της ακτίνας $R = 10$ μια τελείως διαφορετική συμπεριφορά του συστήματος. Η επιδημία 2 εξαφανίζεται πολύ νωρίς (βήμα 40) και τελικά το σύστημα δεν βρίσκεται σε κατάσταση ίασης αλλά σε κατάσταση μόλυνσης με την επιδημία 1. Η ισορροπία μεταξύ των καταστάσεων 0 και 1 επέρχεται ακριβώς στο βήμα που εξαφανίστηκε η επιδημία 2.



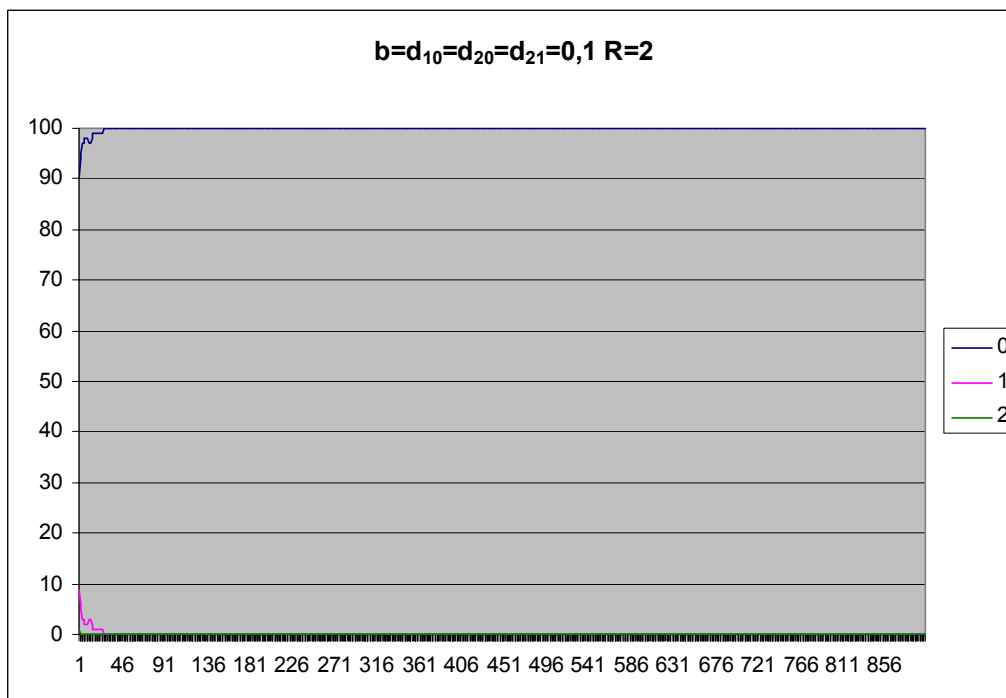
Γράφημα 17 MKK και ΣΚ1, $b=d_{10}=d_{20}=d_{21}=0,1 R = 10$



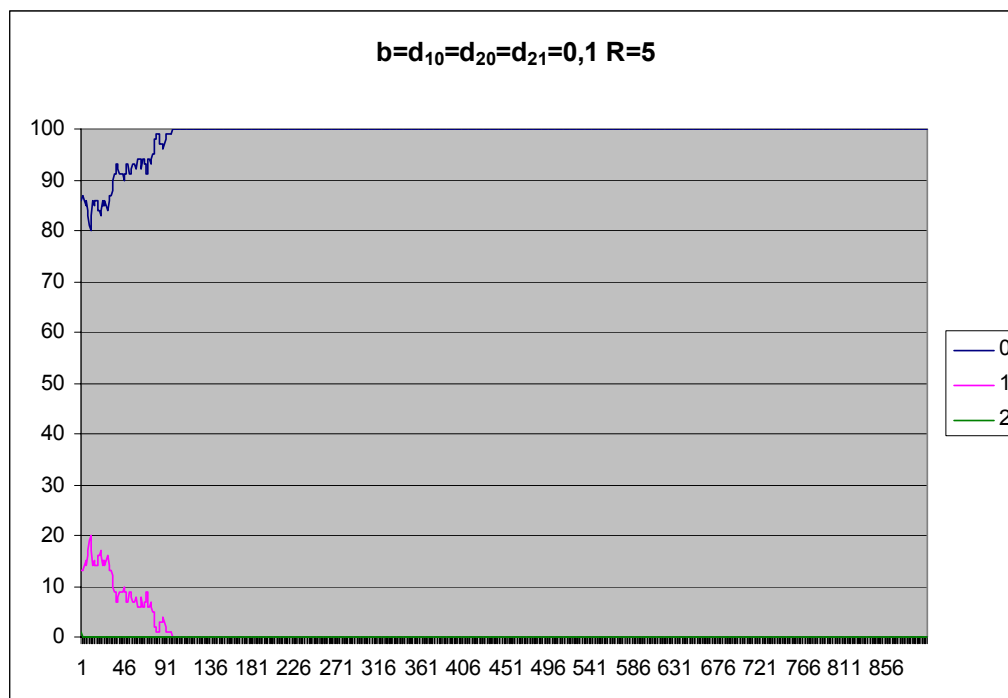
Με Κινητικότητα Κόμβων και ΣΚ2

Στην επόμενη περίπτωση μελετάται για το ΣΚ2 η συμπεριφορά της επιδημικής μόλυνσης για όλες τις τιμές ρυθμών ίασης ίσες με τον ρυθμό μόλυνση ίσες με 0,1 και για ακτίνες δράσης των κόμβων 2, 5, 10.

Για τις δοθείσες τιμές ισχύει $\lambda < 1$, οπότε περιμένουμε να εκδηλωθεί επιδημική αποσύνθεση του συστήματος. Οι περιπτώσεις της ακτίνας $R = 2$ (γράφημα 18) καθώς και αυτή της ακτίνας $R = 5$ (γράφημα 19) μοιάζουν εκπληκτικά κατ' αντιστοιχία με το γράφημα 15 και γράφημα 16. Από τη σύγκριση των δύο σεναρίων ΣΚ1 και ΣΚ2 φαίνεται πως ο αριθμός μολυσμένων κόμβων δεν παίζει κανένα ρόλο και το σύστημα φτάνει σε κατάσταση ίασης σχετικά νωρίς.

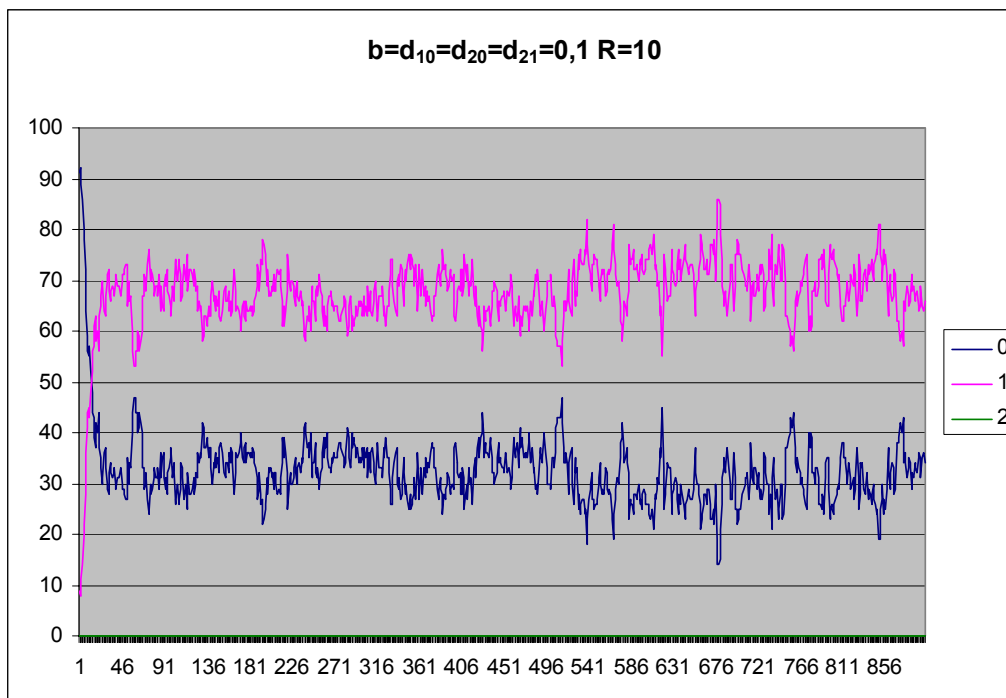


Γράφημα 18 ΜΚΚ και ΣΚ2, $b=d_{10}=d_{20}=d_{21}=0,1 R = 2$



Γράφημα 19 ΜΚΚ και ΣΚ2, $b=d_{10}=d_{20}=d_{21}=0,1 R=5$

Τέλος, για την περίπτωση της ακτίνας $R = 10$, φαίνεται στο γράφημα 20 η αντιστοιχία με το γράφημα 17. Το σύστημα παραμένει μολυσμένο σε κατάσταση 1 και ισορροπεί με ποσοστό 70% μόλυνση 1 και 30% υγιές. Μια σημαντική παρατήρηση είναι η απότομη πτώση του αριθμού υγιών κόμβων και η αντίστοιχη απότομη αύξηση της επιδημίας 1 από τα πρώτα βήματα του αλγορίθμου ενώ η επιδημία 2 έχει εξαφανιστεί.

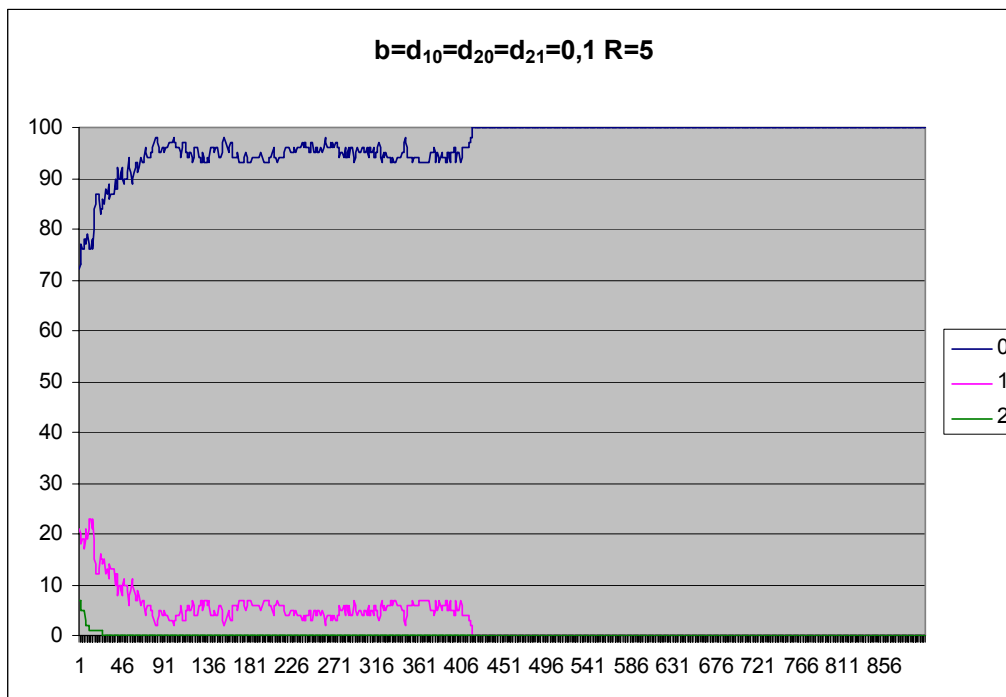


Γράφημα 20 MKK και ΣΚ2, $b=d_{10}=d_{20}=d_{21}=0,1 R = 10$

Χωρίς Κινητικότητα Κόμβων και ΣΚ1

Στην επόμενη περίπτωση μελετάται για το ΣΚ1 η συμπεριφορά της επιδημικής μόλυνσης για όλες τις τιμές ρυθμών ίασης ίσες με τον ρυθμό μόλυνση ίσες με 0,1 και για ακτίνα δράσης των κόμβων $R = 5$ μόνο.

Για τις δοθείσες τιμές ισχύει $\lambda < 1$, οπότε περιμένουμε να εκδηλωθεί επιδημική αποσύνθεση. Από το γράφημα 21 φαίνεται καθαρά η εξαφάνιση της επιδημίας μετά το βήμα 420 στο σύστημα και η πλήρης ίαση του. Παρά τον σχετικά επαρκή αριθμό κόμβων με επιδημία 2 τελικά αυτή εξαφανίζεται σε μικρό αριθμό επανάληψης βημάτων.



Γράφημα 21 ΧΚΚ και ΣΚ1, $b=d_{10}=d_{20}=d_{21}=0,1$ $R = 5$

3.2 Συγκριτικοί Συνοπτικοί Πίνακες Αποτελεσμάτων

Για την καλύτερη κατανόηση και ευκολότερη αξιολόγηση των στοιχείων που προέκυψαν παραθέτουμε τους Πίνακες Αποτελεσμάτων ανά Κατηγορία Μοντέλου και Σενάριο Αριθμών Κόμβων. Σε όλους τους πίνακες δεν απεικονίζεται η τιμή του ρυθμού μόλυνσης β η οποία είναι σταθερή και ίση με 0,1. Επίσης, τα σκιασμένα κελιά αντιστοιχούν στην περίπτωση επιδημικής εξασθένησης.

Πίνακας 1 Τελικά ποσοστά μόλυνσης (ΜΚΚ ΣΚ1)

Μοντέλο Κινητικότητας	ΜΚΚ								
	ΣΚ1 (70 20 10)								
Σενάριο κόμβων	$\delta_{10}=\delta_{20}=\delta_{21}=0,1$			$\delta_{10}=\delta_{20}=\delta_{21}=0,001$			$\delta_{10}=\delta_{21}=0,01 \delta_{20}=0,001$		
Τιμές ρυθμών ίαςσης									
Ακτίνα	2	5	10	2	5	10	2	5	10
Ποσοστό υγιών κόμβων	100%	100%	35%	17%	2%	0%	73%	20%	1%
Ποσοστό μόλυνσης 1	0%	0%	65%	72%	98%	100%	27%	80%	99%
Ποσοστό μόλυνσης 2	0%	0%	0%	11%	0%	0%	0%	0%	0%



Πίνακας 2 Τελικά ποσοστά μόλυνσης (ΜΚΚ ΣΚ2)

Μοντέλο Κινητικότητας	ΜΚΚ								
	ΣΚ2 (89 10 1)								
Τιμές ρυθμών ιάσης	$\delta_{10}=\delta_{20}=\delta_{21}=0,1$			$\delta_{10}=\delta_{20}=\delta_{21}=0,001$			$\delta_{10}=\delta_{21}=0,01 \delta_{20}=0,001$		
Ακτίνα	2	5	10	2	5	10	2	5	10
Ποσοστό υγιών κόμβων	100%	100%	34%	11%	3%	2%	71%	16%	6%
Ποσοστό μόλυνσης 1	0%	0%	66%	87%	97%	98%	29%	84%	94%
Ποσοστό μόλυνσης 2	0%	0%	0%	2%	0%	0%	0%	0%	0%

Πίνακας 3 Τελικά ποσοστά μόλυνσης (ΧΚΚ ΣΚ1)

Μοντέλο Κινητικότητας	ΧΚΚ		
	ΣΚ1 (70 20 10)		
Τιμές ρυθμών ιάσης	$\delta_{10}=\delta_{20}=\delta_{21}=0,1$	$\delta_{10}=\delta_{20}=\delta_{21}=0,001$	$\delta_{10}=\delta_{21}=0,01 \delta_{20}=0,001$
Ακτίνα	5	5	5
Ποσοστό μόλυνσης 0	100%	46%	56%
Ποσοστό μόλυνσης 1	0%	49%	44%
Ποσοστό μόλυνσης 2	0%	5%	0%

Στον Πίνακα 3 τα εμφανιζόμενα ποσοστά είναι αυτά που αναλογούν στις τιμές 900 βημάτων από τα 1000 της προσομοίωσης του αλγορίθμου.



4. Συμπεράσματα

Όπως ήδη αναφέρθηκε στην αρχή οι επιδημικοί αλγόριθμοι συνδέονται με την πιθανο-θεωρητική διάχυση της πληροφορίας μέσα σε ένα δίκτυο κόμβων. Μια πρώτη παρατήρηση των αποτελεσμάτων δείχνει πως στις περισσότερες των περιπτώσεων τα αποτελέσματα της προσομοίωσης συμφωνούν με τα αντίστοιχα θεωρητικά.

Ο αριθμός των κόμβων δεν φαίνεται να έχει ιδιαίτερη επίδραση στα αποτελέσματα της προσομοίωσης για τα δύο σενάρια. Οποιαδήποτε και αν είναι η αναλογία μολυσμένων και υγιών κόμβων σε αντίστοιχες περιπτώσεις, όπως φαίνεται από τους πίνακες 1 και 2, έχουμε παρόμοια τελικά ποσοστά. Εξαίρεση αποτελούν, με αρκετά σημαντικές διαφορές, δύο περιπτώσεις: για ακτίνα 2 και για τιμές ρυθμών ίασης όλες ίσες με 0,001 και για ακτίνα 10 της τελευταίας στήλης του καθενός πίνακα. Το γεγονός αυτό μάλλον μπορεί να εξηγηθεί από τον ακραίο συνδυασμό ακτίνας - τιμής ρυθμών ίασης. Στο επίπεδο της προσομοίωσης αυτό μπορεί να σχετίζεται με τον μικρό αριθμό γειτονικών κόμβων που κάθε φορά έχει ένας δεδομένος κόμβος.

Η ακτίνα δράσης των κόμβων δείχνει να έχει πολύ σημαντική επίδραση στο επιδημικό μοντέλο. Η αύξησή της επιδρά στη διάδοση μιας μόλυνσης και αυτό είναι προφανές από τα εκάστοτε αποτελέσματα.

Στους συνδυασμούς τιμών ρυθμών ίασης που σχετίζονται με την επιδημική εξασθένηση έχουμε ένα παράδοξο για τιμή ακτίνας 10. Ο πληθυσμός των κόμβων βρίσκεται σε μόλυνση που μάλλον οφείλεται στην μεγάλη τιμή ακτίνας, άρα και μεγαλύτερη πιθανότητα μόλυνσης γειτόνων. Επίσης άξιο λόγου είναι πως και στις δύο περιπτώσεις κινητικότητας τα συστήματα βρίσκονται σε ισορροπία από πολύ νωρίς, όπως φαίνεται και στις αντίστοιχες γραφικές τους απεικονίσεις.

Τέλος, τεράστιες αποκλίσεις έχουμε μεταξύ μοντέλου κινητικότητας και μοντέλου χωρίς κινητικότητα αντίστοιχα όσον αφορά τις τιμές μόλυνσης για το ΣΚ1. Η ακινησία των κόμβων επηρεάζει την συμπεριφορά των μολύνσεων. Στις περιπτώσεις της επιδημικής επικράτησης το σύστημα διατηρεί μία χρονική ισορροπία.

Συμπερασματικά η ιδέα της εφαρμογής ενός βιολογικού μοντέλου εξάπλωσης στη διάδοση πληροφορίας σε πληροφορικά συστήματα είναι πολύ ενδιαφέρουσα. Οι επιδημικοί αλγόριθμοι είναι ευέλικτοι και μπορεί εύκολα να προσαρμοστούν στις τιμές των διαφόρων παραμέτρων που τους χαρακτηρίζουν ώστε να υπάρξει υψηλή αξιοπιστία στην προσδοκώμενη διάχυση πληροφορίας. Με την παρούσα εργασία επεκτείνεται το παραδοσιακό επιδημικό μοντέλο SIS με τη χρήση δύο επιπέδων πληροφορίας και μελετώνται θέματα κινητικότητας και παραμέτρων διάχυσης πληροφορίας. Ωστόσο, ένα τεράστιο πεδίο μελέτης μένει ανοικτό και αφορά τον τρόπο με τον οποίο συμπεριφέρονται οι πολύ-επιδημικοί αλγόριθμοι περισσότερων επιπέδων μόλυνσης. Θα πρέπει να μελετηθούν



*Παπαδόπουλος Νίκος, 'Μελέτη εκμετάλλευση της σημασιολογίας της πληροφορίας
πλαισίου μέσω επιδημικών αλγορίθμων και η σύγκρισή τους με άλλα μοντέλα
διάχυσης πληροφορίας'*

θέματα εγκυρότητας της πληροφορίας, αποδοτικής μεταφοράς της, προσαρμογής της στην
τοπολογία του εκάστοτε δικτύου κλπ.



Αναφορές

1. D. Abowd, 'Towards a Better Understanding of Context and Context-Awareness', Proc. Int. Conference of Human Factors in Computing Systems, 2000
2. C. Anagnostopoulos, S. Hadjiefthimiades, 'A Biological Model for Collaborative Context Aware Systems', Transactions on Mobile Computing, TMC-0061-0207
3. C. Anagnostopoulos, 'Epidemic Models', Pervasive Computing Research Group, University of Athens, epidemics.ppt
4. C. Anagnostopoulos, S. Hadjiefthimiades, E. Zervas, Y. Ntarladimas, 'Epidemics-based Collaborative Context Awareness', Mobiquitous Dublin Ireland, 2008
5. L. Lemay & R. Cadenhead, 'Πλήρες Εγχειρίδιο της Java 2 platform', Εκδόσεις Γκιούρδας 1999
6. P. McBride, 'Java ο εύκολος τρόπος', Εκδόσεις Δίαυλος, 1998
7. P.T. Eugster, R. Guerraoui, A.M. Kermarrec, L. Massoulie, 'From Epidemics to Distributed Computing', IEEE Computer, vol. 37, num. 5, 2004, p. 60-67, 2004

Διαδικτυακές Αναφορές

- [Δ1] <http://www.j-sim.org/>
- [Δ2] http://www.j-sim.org/tutorial/jsim_tutorial.html
- [Δ3] http://www.j-sim.org/drcl.inet/inet_tutorial.html
- [Δ4] <http://www.j-sim.org/v1.3/api/index.html>
- [Δ5] <http://www.j-sim.org/ref.script/index.html#Path%20Expression>
- [Δ6] <http://www.tcl.tk>
- [Δ7] <http://java.sun.com/developer/onlineTraining/>
- [Δ8] <http://en.wikipedia.org/wiki/Simulation>
- [Δ9] <http://el.wikipedia.org/wiki/Java>



Παράρτημα A: J-SIM

Εισαγωγή

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.

Με τον όρο προσομοίωση στην πληροφορική εννοούμε τη διαδικασία με την οποία προσπαθούμε να αναπαράγουμε ένα πραγματικό ή υποθετικό περιβάλλον με στόχο τη μελέτη του και την ανταπόκρισή του σε αλλαγές διαφόρων παραμέτρων που το χαρακτηρίζουν.

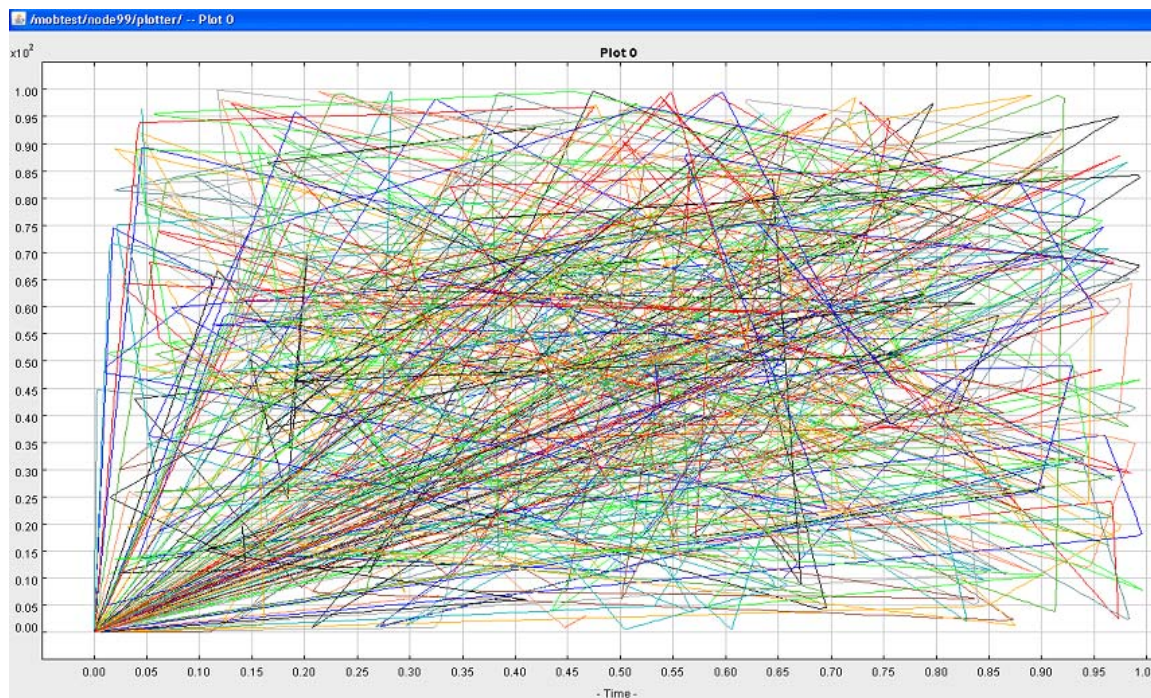
Η προσομοίωση με τη βοήθεια υπολογιστών αποτελεί πλέον ένα χρήσιμο και αναπόσπαστο κομμάτι της μελέτης πολλών συστημάτων για ένα ευρύτατο φάσμα επιστημών. Τέτοιες είναι η φυσική, η χημεία, η βιολογία, τα οικονομικά και οι κοινωνικές επιστήμες (υπολογιστική κοινωνιολογία) καθώς επίσης και η εφαρμοσμένη μηχανική. Ένα καλό παράδειγμα της αξίας της χρησιμοποίησης των υπολογιστών που προσομοιώνουν περιβάλλοντα μελέτης μπορεί να βρεθεί στον τομέα της προσομοίωσης κυκλοφορίας δικτύων. Σε τέτοιες προσομοιώσεις, η πρότυπη συμπεριφορά θα αλλάξει τη μορφή του συστήματος σύμφωνα με το σύνολο των αρχικών παραμέτρων που υποτίθενται για το περιβάλλον. [Δ8]

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. [Δ1]

Στα πλαίσια της διπλωματικής αυτής εργασίας η j-sim χρησιμοποιήθηκε αποκλειστικά για τη δημιουργία των θέσεων των εκατό κόμβων για 1000 βήματα εκτέλεσης. Το αρχείο .tcl που φτιάχτηκε ονομάστηκε *n100.tcl* και εφαρμόστηκε σ' αυτό το μοντέλο κινητικότητας Random Way Point. Με αυτό το μοντέλο εξασφαλίζεται για τον επιδημικό αλγόριθμο η τυχαιότητα στην κίνηση των κόμβων τού προς εξέταση συστήματος. Το αρχείο συντεταγμένων που παράγεται από την προσομοίωση ονομάστηκε *n100.plot* και είναι ένα αρχείο κειμένου.



Ένα στιγμιότυπο από την οθόνη προσομοίωσης για την κίνηση των εκατό κόμβων φαίνεται παρακάτω (σχ. 6).

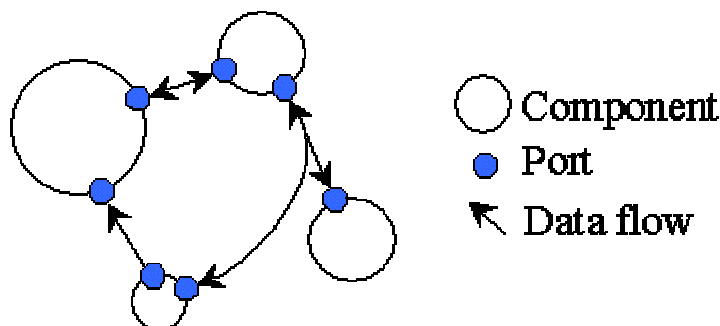


Σχήμα 6 Οθόνη προσομοίωσης κίνησης 100 κόμβων



Autonomous Component Architecture (ACA)

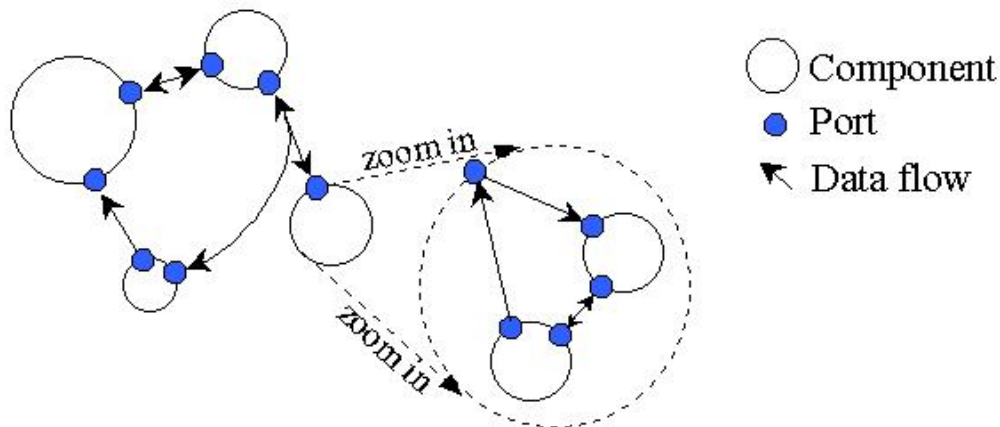
Στην αυτόνομη αρχιτεκτονική συστατικών ACA βασική οντότητα είναι το *Συστατικό* (Component). Κάθε συστατικό περιλαμβάνει ένα ή περισσότερα *σημεία τέλους* (end points) που ονομάζονται *θύρες* (ports). Κάθε εφαρμογή θεωρείται μία σύνθεση διάφορων συστατικών (σχ. 7).



Σχήμα 7 Σύνθεση συστατικών στην j-sim

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.

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 8 απεικονίζει την έννοια των σύνθετων συστατικών. [Δ2], [Δ3]



Σχήμα 8 Απεικόνιση σύνθετου συστατικού στην j-sim

Scripts

Η j-sim είναι ένα διπλό-γλωσσικό περιβάλλον: Η Java χρησιμοποιείται για να υλοποιήσει όλες τις κλάσεις και η Tcl χρησιμοποιείται 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 παρόμοια με ένα σύστημα αρχείων σε ένα σύγχρονο λειτουργικό σύστημα υπολογιστή. Για να διευκολυνθεί η διαμόρφωση της προσομοίωσης δικτύων, έχει αναπτυχθεί το σύνολο εντολών RUV, δηλαδή, ένα σύνολο εικονικών εντολών συστημάτων χρόνου εκτέλεσης (Runtime Virtual system commands) για να χειριστεί ο προγραμματιστής τη συστατική ιεραρχία με τον ίδιο τρόπο όπως διάφορες εντολές του λειτουργικού συστήματος Unix κάνουν στο σύστημα αρχείων.

Συστατικό MobilityModel



Το συστατικό *MobilityModel* μιμείται τη κίνηση του ασύρματου κόμβου σ' ένα ad-hoc δίκτυο. Δύο πρότυπα κινητικότητας μπορούν να εφαρμοστούν: Το *Random Way Point* (RWP) πρότυπο κινητικότητας και το πρότυπο που βασίζεται σε καθορισμένη τροχιά. Εάν μια τροχιά έχει εγκατασταθεί, το συστατικό *MobilityModel* αυτόματα προσομοιώνει την κίνηση των κόμβων σύμφωνα μ' αυτήν, σε διαφορετική περίπτωση χρησιμοποιείται το *Random Way Point* πρότυπο κινητικότητας. [Δ4]

Συστατικό *NodePositionTracker*

Αυτό το συστατικό διαιρεί το επίπεδο της προσομοιούμενης 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 *nid* 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 στην οθόνη προσομοίωσης tcl.

Random Waypoint μοντέλο κινητικότητας

Το *Random Way Point* λειτουργεί με την εξής διαδικασία: μετά από την εκκίνηση του συστατικού *MobilityModel* και εφόσον σ' αυτό δεν έχει καθοριστεί συγκεκριμένη τροχιά, αρχίζει να επιλέγει τυχαία έναν προορισμό στον οποίο θα μετακινηθεί ο κόμβος. Οι συντεταγμένες του επιλεγμένου προορισμού πρέπει να είναι μέσα στην περιοχή που έχει οριστεί για την κίνησή του και έχει προηγουμένως διευκρινιστεί στη λειτουργία *setTopologyParameters*. Η ταχύτητα κίνησης *speed* 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:

$$changeDestinationPeriod = \frac{dist}{speed}$$



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.

Μια παράμετρος στο μοντέλο αυτό είναι ο *χρόνος καθυστέρησης*, δηλαδή ο χρόνος που μένει ο κόμβος στην τρέχουσα θέση μέχρι να επιλεγεί και να ξεκινήσει για την καινούρια του θέση. Στην περίπτωση που εξετάζουμε θεωρήθηκε ότι ο χρόνος αυτός είναι μηδενικός.



Εντολές που χρησιμοποιήθηκαν στο αρχείο προσομοίωσης [Δ5]

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 απαραίτητα.

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 αντιστοίχιας.

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 ?-α? <Java_class_name> <path1> ?<path2>...?`

`mkdir ?-α? <Java_object_ref> <path1> ?<path2>...?`

`mkdir ?-α? <port_path1> ?<port_path2>...?`

`mkdir ?-α? <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 έναν ή περισσότερους 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 ?-α? <default_class> <component_path>`".

rt

Έχει πρόσβαση στο χρόνο εκτέλεσης και τις πληροφορίες του.

ΣΥΝΟΨΗ

`rt ?-α? <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 ?-α? <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 ?-α? <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".



Αρχείο προσομοίωσης για τη δημιουργία συντεταγμένων 100 κόμβων: n100.tcl

```
puts mobtest

set minx 0.0          # H elaxisti timi x ths perioxis prosomiosis
set maxx 100.0       # H megisti timi x ths perioxis prosomiosis
set miny 0.0          # H elaxisti timi y ths perioxis prosomiosis
set maxy 100.0       # H megisti timi y ths perioxis prosomiosis
set maxx 0.0         # H megisti timi z ths perioxis prosomiosis
set minz 0.0         # H elaxisti timi z ths perioxis prosomiosis
set dx 1.0           # To mikos x ths perioxis plegmatos
set dy 1.0           # To platos y ths perioxis plegmatos
set dz 0.0           # To ipsos z ths perioxis plegmatos
set maxSpeed 2.0     # H taxitita ton komvon

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 "n100.plot"; ## to onoma tou arxeiou sintetagmenon
    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

$mob4 setTopologyParameters $maxx $maxy $maxz $minx $miny $minz $dx $dy $dz
$mob4 setPosition $maxSpeed 0 0 0

$mob5 setTopologyParameters $maxx $maxy $maxz $minx $miny $minz $dx $dy $dz
$mob5 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 $convert2/out@ -to $plot/2@0
connect $mob3/.query@ -and $convert3/query@
connect -c $convert3/out@ -to $plot/3@0
connect $mob4/.query@ -and $convert4/query@
connect -c $convert4/out@ -to $plot/4@0
connect $mob5/.query@ -and $convert5/query@
connect -c $convert5/out@ -to $plot/5@0

.
.
.

connect $mob95/.query@ -and $convert95/query@
connect -c $convert95/out@ -to $plot/95@0
connect $mob96/.query@ -and $convert96/query@
connect -c $convert96/out@ -to $plot/96@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
```



Αρχείο συντεταγμένων 100 κόμβων: n100.plot

Παραθέτουμε ενδεικτικά τις πρώτες γραμμές του αρχείου συντεταγμένων των 100 κόμβων που δημιουργήθηκε από το πρόγραμμα προσομοίωσης *n100.tcl*. Το αρχείο αυτό για χίλια βήματα προσομοίωσης περιέχει 99932 γραμμές ή ισοδύναμα 999 εκατοντάδες θέσεων συντεταγμένων των κόμβων.

```
### 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--???  
##### NEW SET--0--25--???  
m--0--25--0.6818000450512274--0.06301497336144578--???  
p--0--89--0.029450646877562166--0.5755566986834881--???  
##### NEW SET--0--24--???  
m--0--24--1.524386665829603--0.11587401366195145--???  
p--0--88--0.42300252958742585--0.47283860865774646--???  
##### NEW SET--0--23--???  
m--0--23--1.108814813147809--0.8401783677477626--???
```



p--0--87--0.062365036496209864--0.04906338751498503--???
NEW SET--0--22--???
m--0--22--0.9340435474632783--0.5247067351565747--???
p--0--86--1.8961084087518378--0.05677889229250563--???
NEW SET--0--21--???
m--0--21--0.4825075441807981--0.9016415094021905--???
p--0--85--1.2035175270536453--0.33461251012683063--???
NEW SET--0--20--???
m--0--20--1.3816681717387629--0.17948008666704232--???
p--0--84--0.3064767398039979--0.3819861074592534--???
NEW SET--0--83--???
m--0--83--0.12105540104040505--0.06244779505035895--???
NEW SET--0--82--???
m--0--82--0.010072445168851692--1.7998206962573144--???
NEW SET--0--81--???
m--0--81--0.7521382810473336--0.5202356882478165--???
NEW SET--0--9--???
m--0--9--0.34632779352236376--0.23763900779151115--???
NEW SET--0--80--???
m--0--80--0.8904154319877683--0.5156747054334072--???
NEW SET--0--8--???
m--0--8--0.43924871234739754--1.9048676326431753--???
p--0--7--0.07944272615932392--0.05043615063999555--???
p--0--6--1.421634425165325--0.9810532492650377--???
p--0--5--0.15156697720113624--0.6728955455188758--???
p--0--4--0.6515562672368325--0.0970930980705182--???
p--0--3--0.014755676943200124--0.7139778104165282--???
p--0--2--0.5448174598267235--0.3581660811345293--???
p--0--1--0.8537818833134297--1.4298498794771952--???
p--0--0--0.7521851717758202--0.5851617880652291--???
p--0--19--0.9540693078884314--1.25680058879331--???
p--0--18--0.6983158497575057--1.2917978872066902--???
p--0--17--1.0605449002807736--1.5042233656404775--???
p--0--16--1.1439309092870067--0.9945122755146623--???
NEW SET--0--15--???
m--0--15--0.6939089253344303--1.4660950281291185--???
p--0--79--0.9800567574710458--0.05597557789597041--???
NEW SET--0--14--???
m--0--14--0.11200019032516868--0.06993154409519634--???
p--0--78--1.0364663714546434--0.88982336092284--???
NEW SET--0--13--???
m--0--13--0.7937662864712717--1.2044842289909958--???
p--0--77--0.46648195153911926--0.5059993563154183--???
NEW SET--0--12--???
m--0--12--0.026082711902087094--1.0968547984607109--???
p--0--76--0.6642863987768225--0.30568012534002326--???
NEW SET--0--11--???
m--0--11--0.16156715435865318--0.17088106223868038--???
p--0--75--1.016282542997647--1.2376718268943343--???
NEW SET--0--10--???
m--0--10--0.5379442760134772--1.352638279159933--???
p--0--60--0.5379442760134772--1.352638279159933--???
p--0--74--0.9555077437555726--0.50264064938478--???
p--0--73--0.14171656384377213--1.141015433506164--???



Παράρτημα Β: JAVA

Η γλώσσα προγραμματισμού JAVA χρησιμοποιήθηκε για την κωδικοποίηση του επιδημικού αλγορίθμου διάχυσης πληροφορίας για τα σενάρια της κινητικότητας και χωρίς κινητικότητα κόμβων. Το αποτέλεσμα είναι η δημιουργία των αρχείων myepidemic.java και myepidemicnomob.java σε περιβάλλον Windows XP.

Η java, της οποίας το λογότυπο φαίνεται στο σχήμα 9, είναι μία Αντικειμενοστρεφής Γλώσσα Προγραμματισμού (Object Oriented Programming Language) με πολλές δυνατότητες και μεγάλο λεξιλόγιο, σχεδιασμένη από τη Sun Microsystems, για χρήση σε δίκτυα και ειδικά στο Internet. Ένα από τα βασικά πλεονεκτήματα της γλώσσας έναντι των περισσότερων άλλων γλωσσών είναι η ανεξαρτησία του λειτουργικού συστήματος και πλατφόρμας. Όλα τα προγράμματα σε java τρέχουν με τον ίδιο τρόπο σε διάφορα λειτουργικά συστήματα, όπως WINDOWS, LINUX, UNIX, MACINTOSH χωρίς να χρειαστεί να ξαναγίνει η μεταγλώττιση ή να αλλάξει ο πηγαίος κώδικας. Αυτό κατέστη δυνατό χάρη στην ανάπτυξη της Εικονικής Μηχανής (Virtual Machine) [Δ9].



Σχήμα 9
Λογότυπο
JAVA

Αφού το πρόγραμμα γραφτεί, μεταγλωττίζεται από τον μεταγλωττιστή javac και παράγει έναν αριθμό από αρχεία .class. Τα τελευταία είναι η μορφή που παίρνει ο πηγαίος κώδικας της java όταν μεταγλωττίζεται. Αν προσπαθήσουμε, λοιπόν, να εκτελέσουμε την εφαρμογή μας η εγκατεστημένη στον υπολογιστή εικονική μηχανή της γλώσσας θα διαβάσει τα αρχεία .class και θα τα μεταφράσει σε γλώσσα και εντολές μηχανής που υποστηρίζει το λειτουργικό σύστημα του υπολογιστή. Η εικονική μηχανή είναι λογισμικό εξαρτημένο από το λειτουργικό που έχει ο υπολογιστής μας. Οτιδήποτε θελήσει πλέον ο χρήστης γίνεται μέσω του προγράμματος αυτού. Η εικονική μηχανή είναι υπεύθυνη για την επικοινωνία του χρήστη με τον υπολογιστή. Ο χρήστης-προγραμματιστής δεν μπορεί να γράψει «κακόβουλο» κώδικα γιατί η εικονική μηχανή θα τον ανιχνεύσει και δεν θα επιτρέψει την εκτέλεσή του. Το ίδιο συμβαίνει και στην περίπτωση που ο χρήστης προσπαθήσει να κατεβάσει από το ίντερνετ κακό λογισμικό και να το εκτελέσει. Από τα παραπάνω, είναι φανερό, πως η εικονική μηχανή ενισχύει την ασφάλεια του συστήματος.

Ένα άλλο σημαντικό χαρακτηριστικό της γλώσσας είναι ο Συλλέκτης Απορριμμάτων (Carbage Collector). Σχετίζεται με τη διαχείριση της μνήμης και το άδειασμά της από κομμάτια που δεν είναι άλλο απαραίτητα. Η εικονική μηχανή επεμβαίνει όταν είναι αναγκαίο ενεργοποιώντας τον συλλέκτη απορριμμάτων. Έτσι ο χρήστης δεν ανησυχεί για το πότε πρέπει να ελευθερωθεί χώρος ούτε για το αν κάποιος



δείκτης δείχνει σε κενό χώρο μνήμης, γεγονός που βελτιώνει τη σταθερότητα του συστήματος. [Δ9]

Το τελικό στιγμιότυπο της οθόνης εκτέλεσης του αρχείου *myepidemic.java* μέσα από το παράθυρο του command prompt των Windows XP φαίνεται στο σχήμα 10 που ακολουθεί.

```
C:\WINDOWS\system32\cmd.exe
Now Calc 989
Now Calc 990
Now Calc 991
Now Calc 992
Now Calc 993
Now Calc 994
Now Calc 995
Now Calc 996
Now Calc 997
Now Calc 998
Now Calc 999
C:\Documents and Settings\Nikos\Desktop\Project DIPLOMATIKI\Java epidemic files\
Me kinitikotita komvon\70 20 10>pause
Press any key to continue . . .
```

Σχήμα 10 Οθόνη εκτέλεσης του αρχείου *myepidemic.java*

Ακολουθούν ο κώδικας του αρχείου *myepidemic.java* που αφορά το επιδημικό μοντέλο κινητικότητας των κόμβων και ο κώδικας του αρχείου *myepidemicnomob.java* που αφορά το επιδημικό μοντέλο χωρίς κινητικότητα κόμβων. Το αρχείο συντεταγμένων μετονομάστηκε σε *n100.txt*.



Κώδικας αρχείου: myepidemic.java

```
import java.io.*;

public class myepidemic {

    static final int SUSCEPTIBLE = 0;
    static final int INFECTED1 = 1;
    static final int INFECTED2 = 2;

    static final String COMMA = ",";
    static final String FILENAME0 = "out0.csv"; //INFECTED 0
    static final String FILENAME1 = "out1.csv"; //INFECTED 1
    static final String FILENAME2 = "out2.csv"; //INFECTED 2
    static final String FILENAME3 = "out3.csv"; //ALL
    static final String FILENAME4 = "out4.csv"; //COUNTERS ONLY

    static double A[][];
    static int S[];
    static int M[];
    static int N=0;
    static int N1=0;
    static int N2=0;
    static int startcount=0;
    static int nN;
    static String filename="";
    static String line = "";
    static double b=0.0;
    static double d10=0.0;
    static double d21=0.0;
    static double d20=0.0;
    static double r=0.0;

    public static void main( String[] args) {
        if (args.length != 8) {
            System.err.println ("Usage: java myepidemic: #ofNodes startcount
inputFilename b d10 d20 d21 radius");
            return;
        }

        N = Integer.parseInt( args[0] );
        startcount = Integer.parseInt( args[1] );
        filename = args[2];
        b = Double.parseDouble( args[3] );
        d10 = Double.parseDouble( args[4] );
        d20 = Double.parseDouble( args[5] );
        d21 = Double.parseDouble( args[6] );
        r = Double.parseDouble( args[7] );

        A = new double[N][2];
        M = new int[3];

        initS();
        readFile();
    }
}
```



```
        results();
        return;
    }

    static void initS() {
// αρχικοποιεί τον πίνακα καταστάσεων των κόμβων
        int i;
        N1 = (int)(N/5);
        N2 = (int)(N/10);

        S = new int[N];

        for(i=0 ; i<N1 ; i++) S[i]=INFECTED1;
        for(i=N1 ; i<N1+N2 ; i++) S[i]=INFECTED2;
        for (i=N1+N2 ; i<N ; i++) S[i]= SUSCEPTIBLE;
    }

    static void mySplit() {
// διαβάζει μια γραμμή του αρχείου sintetagmenon n100.txt

        int node, stop, start = 0;
        double x,y;
        String s;

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

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

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

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

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

// Thetei tis sintetagmenes pou diavazei apo to arxeio ston pinaka sintetagmenon ton komvon
        A[node][0] = x;

        A[node][1] = y;
```



```
}  
  
static void readFile() {  
// διαvazei to arxeio sintetagmenon gia kathe xroniki stigmi-vima  
try {  
    FileReader f = new FileReader(filename);  
    BufferedReader b = new BufferedReader(f);  
  
    BufferedWriter outfile[] = new BufferedWriter[5];  
    outfile[0] = new BufferedWriter(new FileWriter(FILENAME0));  
// dimiourgei to arxeio eksodou 0  
  
    outfile[1] = new BufferedWriter(new FileWriter(FILENAME1));  
// dimiourgei to arxeio eksodou 1  
  
    outfile[2] = new BufferedWriter(new FileWriter(FILENAME2));  
// dimiourgei to arxeio eksodou 2  
  
    outfile[3] = new BufferedWriter(new FileWriter(FILENAME3));  
// dimiourgei to arxeio eksodou 3  
  
    outfile[4] = new BufferedWriter(new FileWriter(FILENAME4));  
// dimiourgei to arxeio eksodou 4  
  
    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) {  
                    myEpidemic();  
  
                    M[0] = 0;  
                    M[1] = 0;  
                    M[2] = 0;  
  
                    // Grafei arithmo vimatos  
                    for(int k=0 ; k<5 ; k++) {  
                        outfile[k].write(""+nN);  
                    }  
                    for (int i=0 ; i<S.length ; i++){  
                        x=S[i];  
                        M[x]++;  
                        s = COMMA + x;  
                        outfile[3].write(s);  
                        outfile[x].write(COMMA + i);  
                    }  
                }  
            }  
        }  
    }  
}
```



```
// Grafei Metrites

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

outfile[4].write(COMMA+M[0]+COMMA+M[1]+COMMA+M[2]);

//Αλλαγή γραμμής σε όλα τα αρχεία
for(int k=0 ; k<5 ; k++) {
    outfile[k].newLine();
}
}
}
}
line = b.readLine();
}
b.close();
for(int k=0 ; k<5 ; k++) {
    outfile[k].flush();
    outfile[k].close();
}
}
catch(IOException e) {
    System.err.println("Exception caught: " + e.getMessage());
}
}

static void results() {
// synoptika telika apotelesmata molinsis komvon
double p1, p2;
int c1 = 0, c2 = 0;

for(int i=0 ; i<N ; i++) {
    if (S[i] == 1){
        c1++;
    }else if (S[i] == 2){
        c2++;
    }
}

p1 = (double)c1/N;
p2 = (double)c2/N;
System.out.println ("\n\nnb=" + b + " d10 = " + d10 + " d20 = " + d20 + "
d21 = " + d21 + " r= " +r);
System.out.println ("N=" + N +
"\nc1=" + c1 + " p1= "+p1 +
"\nc2=" + c2 + " p2= "+p2);
}

static void myEpidemic() {
// ilopoiisi epidimikou algorithmou

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

double ra;
```



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

nbs = new int[N];

for(int j=0 ; j<N ; j++) {

//1ο kommati epidimikou algorithmou

        System.out.println("j = " + j + " nN = " +nN);
        if(S[j] == 2) {
// Elegxos an enas komvos tha ginei igiis apo katastasi 2 se 1 h apo katastasi 2 se 0
            ra = Math.random();
            if(ra < d21){
                S[j] = 1;
                System.out.println(nN + " iasi katastasis 2->1:
kombou " + j + " ra=" +ra + " d21=" +d21);
            }
            if(ra > d21 && ra < (d21+d20)) {
                S[j] = 0;
                System.out.println(nN + " iasi katastasis 2->0:
kombou " + j + " ra=" +ra + " d20=" +d20);
            }
        }

        if(S[j] == 1){
            ra = Math.random();
            if(ra < d10){
// Elegxos an enas komvos tha ginei igiis apo katastasi 1 se 0
                S[j] = 0;
                System.out.println(nN + " iasi katastasis 1->0:
kombou " + j + " ra=" +ra + " d10=" +d10);
            }
        }

//2ο kommati epidimikou algorithmou

        infection = 0;
        k = 0;

        for(i=0 ; i<N ; i++) nbs[i]=0;
// arxikopoiisi pinaka geitonikon komvov

        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);
            }
        }
// emfanisi geitonikou komvou
```



```
                }  
            }  
  
            System.out.println("Synolo geitonikon komvon k=" +k);  
            // anafora synolou geitonon  
  
            for(i=0 ; i<k ; i++){  
                if(S[nbs[i]] == 2 && S[j] != 2){  
                    ra = Math.random();  
                    if(ra < b){  
                        S[j] = 2;  
                        System.out.println("Molinsi S[" +j+"]=2");  
                        infection = 1;  
                    }  
                }  
            }  
  
            if(infection == 0){  
                for(i=0 ; i<k ; i++){  
                    if(S[nbs[i]] == 1 && S[j] != 1){  
                        ra = Math.random();  
                        if(ra < b){  
                            S[j] = 1;  
                            System.out.println("Molinsi S[" +j+"]=1");  
                        }  
                    }  
                }  
            }  
        }  
    }  
}  
//next j  
}  
//myEpidemic  
}  
//class
```

Κώδικας αρχείου: runEpidemic.bat

Παρακάτω παρουσιάζεται το αρχείο δέσμης που δημιουργήθηκε για να διευκολύνει την εισαγωγή των ορισμάτων και την εκτέλεση του αρχείου του επιδημικού αλγορίθμου με κινητικότητα κόμβων.

```
java myepidemic 100 101 n100.txt 0.1 0.1 0.1 0.1 2 > results100.txt  
pause
```




Κώδικας αρχείου: myepidemicnomob.java

```
import java.io.*;

public class myepidemicnomob {

    static final int SUSCEPTIBLE = 0;
    static final int INFECTED1 = 1;
    static final int INFECTED2 = 2;

    static final String COMMA = ",";
    static final String FILENAME0 = "out0.csv"; //INFECTED 0
    static final String FILENAME1 = "out1.csv"; //INFECTED 1
    static final String FILENAME2 = "out2.csv"; //INFECTED 2
    static final String FILENAME3 = "out3.csv"; //ALL
    static final String FILENAME4 = "out4.csv"; //COUNTERS ONLY

    static double A[][];
    static int S[];
    static int M[];
    static int N=0;
    static int N1=0;
    static int N2=0;
    static int startcount=0;
    static int nN;
    static String filename="";
    static String line = "";
    static double b=0.0;
    static double d10=0.0;
    static double d21=0.0;
    static double d20=0.0;
    static double r=0.0;

    public static void main( String[] args) {
        if (args.length != 8) {
            System.err.println ("Usage: java myepidemicnomob: #ofNodes startcount
inputFilename b d10 d20 d21 radius");
            return;
        }

        N = Integer.parseInt( args[0] );
        startcount = Integer.parseInt( args[1] );
        filename = args[2];
        b = Double.parseDouble( args[3] );
        d10 = Double.parseDouble( args[4] );
        d20 = Double.parseDouble( args[5] );
        d21 = Double.parseDouble( args[6] );
        r = Double.parseDouble( args[7] );

        A = new double[N][2];
        M = new int[3];

        initS();
        readfile();
    }
}
```



```
        results();
        return;
    }

    static void initS() {
// αρχικοποιεί τον πίνακα καταστάσεων των κόμβων
        int i;
        N1 = (int)(N/5);
        N2 = (int)(N/10);

        S = new int[N];

        for(i=0 ; i<N1 ; i++) S[i]=INFECTED1;
        for(i=N1 ; i<N1+N2 ; i++) S[i]=INFECTED2;
        for (i=N1+N2 ; i<N ; i++) S[i]= SUSCEPTIBLE;
    }

    static void mySplit() {
// διαβάζει μια γραμμή του αρχείου sintetagmenon n100.txt

        int node, stop, start = 0;
        double x,y;
        String s;

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

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

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

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

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

// Thetei tis sintetagmenes pou diavazei apo to arxeio ston pinaka sintetagmenon ton komvon
        A[node][0] = x;

        A[node][1] = y;
```



```
}  
  
static void readFile() {  
// διαvazei to arxeio sintetagmenon gia kathe xroniki stigmi-vima  
try {  
    FileReader f = new FileReader(filename);  
    BufferedReader b = new BufferedReader(f);  
  
    BufferedWriter outfile[] = new BufferedWriter[5];  
    outfile[0] = new BufferedWriter(new FileWriter(FILENAME0));  
// dimiourgei to arxeio eksodou 0  
  
    outfile[1] = new BufferedWriter(new FileWriter(FILENAME1));  
// dimiourgei to arxeio eksodou 1  
  
    outfile[2] = new BufferedWriter(new FileWriter(FILENAME2));  
// dimiourgei to arxeio eksodou 2  
  
    outfile[3] = new BufferedWriter(new FileWriter(FILENAME3));  
// dimiourgei to arxeio eksodou 3  
  
    outfile[4] = new BufferedWriter(new FileWriter(FILENAME4));  
// dimiourgei to arxeio eksodou 4  
  
    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 z=0 ; z<1000 ; z++) {  
// epanalipsi gia 1000 vimata tou epidimikou algorithmou  
  
                    if(nN == startcount) {  
                        myEpidemic();  
  
                        M[0] = 0;  
                        M[1] = 0;  
                        M[2] = 0;  
  
                        // Grafei arithmo vimatos  
                        for(int k=0 ; k<5 ; k++) {  
                            outfile[k].write(""+nN);  
                        }  
                        for (int i=0 ; i<S.length ; i++){  
                            x=S[i];  
                            M[x]++;  
                            s = COMMA + x;
```



```
        outfile[3].write(s);
        outfile[x].write(COMMA + i);
    }

    // Grafei Metrites

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

    outfile[4].write(COMMA+M[0]+COMMA+M[1]+COMMA+M[2]);

        //Αλλαγή γραμμής σε όλα τα αρχεία
        for(int k=0 ; k<5 ; k++) {
            outfile[k].newLine();
        }
    }
}

    line = b.readLine();
}
b.close();
for(int k=0 ; k<5 ; k++) {
    outfile[k].flush();
    outfile[k].close();
}
}
catch(IOException e) {
    System.err.println("Exception caught: " + e.getMessage());
}
}

static void results() {
// synoptika telika apotelesmata molinsis komvon
double p1, p2;
int c1 = 0, c2 = 0;

    for(int i=0 ; i<N ; i++) {
        if (S[i] == 1){
            c1++;
        }else if (S[i] == 2){
            c2++;
        }
    }

    p1 = (double)c1/N;
    p2 = (double)c2/N;
    System.out.println ("\n\n\nnb=" + b + "  d10 = " + d10 + "  d20 = " + d20 + "
d21 = " + d21 + "  r= " +r);
    System.out.println ("N=" + N +

        "\nc1=" + c1 + " p1=" +p1 +
        "\nc2=" + c2 + " p2=" +p2);

}
}
```



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

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

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

    nbs = new int[N];

    for(int j=0 ; j<N ; j++) {

//1o kommati epidimikou algorithmou

        System.out.println("j = " + j + " nN = " +nN);
        if(S[j] == 2) {
// Elegxos an enas komvos tha ginei igiis apo katastasi 2 se 1 h apo katastasi 2 se 0
            ra = Math.random();
            if(ra < d21){
                S[j] = 1;
                System.out.println(nN + " iasi katastasis 2->1:
kombou " + j + " ra=" +ra + " d21= "+d21);
            }
            if(ra > d21 && ra < (d21+d20)) {
                S[j] = 0;
                System.out.println(nN + " iasi katastasis 2->0:
kombou " + j + " ra=" +ra + " d20= "+d20);
            }
        }

        if(S[j] == 1){
            ra = Math.random();
            if(ra < d10){
// Elegxos an enas komvos tha ginei igiis apo katastasi 1 se 0
                S[j] = 0;
                System.out.println(nN + " iasi katastasis 1->0:
kombou " + j + " ra=" +ra + " d10= "+d10);
            }
        }
    }

//2o kommati epidimikou algorithmou

    infection = 0;
    k = 0;

    for(i=0 ; i<N ; i++) nbs[i]=0;
// arxikopoiisi pinaka geitonikon komvov

    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) );
```




Κατάλογος Σχημάτων

Σχήμα 1 Απεικόνιση μιας επιδημίας.....	12
Σχήμα 2 Καταστάσεις Μοντέλου SIS.....	17
Σχήμα 3 Καταστάσεις Μοντέλου SI.....	18
Σχήμα 4 Διάγραμμα Πολύ-Επιδημικού Μοντέλου N καταστάσεων.....	20
Σχήμα 5 Πολύ-Επιδημικό Μοντέλο με $N = 2$	21
Σχήμα 6 Οθόνη προσομοίωσης κίνησης 100 κόμβων.....	54
Σχήμα 7 Σύνθεση συστατικών στην j-sim.....	55
Σχήμα 8 Απεικόνιση σύνθετου συστατικού στην j-sim.....	56
Σχήμα 9 Λογότυπο JAVA.....	65
Σχήμα 10 Οθόνη εκτέλεσης του αρχείου myepidemic.java.....	66

Κατάλογος Γραφημάτων

Γράφημα 1 MKK και ΣΚ1 $b=0,1$ $d_{10}=0,01$ $d_{20}=0,001$ $d_{21}=0,01$ $R=2$	28
Γράφημα 2 MKK και ΣΚ1 $b=0,1$ $d_{10}=0,01$ $d_{20}=0,001$ $d_{21}=0,01$ $R=5$	29
Γράφημα 3 MKK και ΣΚ1 $b=0,1$ $d_{10}=0,01$ $d_{20}=0,001$ $d_{21}=0,01$ $R=10$	30
Γράφημα 4 MKK και ΣΚ2 $b=0,1$ $d_{10}=0,01$ $d_{20}=0,001$ $d_{21}=0,01$ $R=2$	31
Γράφημα 5 MKK και ΣΚ2 $b=0,1$ $d_{10}=0,01$ $d_{20}=0,001$ $d_{21}=0,01$ $R=5$	32
Γράφημα 6 MKK και ΣΚ2 $b=0,1$ $d_{10}=0,01$ $d_{20}=0,001$ $d_{21}=0,01$ $R=10$	33
Γράφημα 7 MKK και ΣΚ1 $b=0,1$ $d_{10}=d_{20}=d_{21}=0,001$ $R=2$	34
Γράφημα 8 MKK και ΣΚ1 $b=0,1$ $d_{10}=d_{20}=d_{21}=0,001$ $R=5$	35
Γράφημα 9 MKK και ΣΚ1 $b=0,1$ $d_{10}=d_{20}=d_{21}=0,001$ $R=10$	36
Γράφημα 10 MKK και ΣΚ2 $b=0,1$ $d_{10}=d_{20}=d_{21}=0,001$ $R=2$	37
Γράφημα 11 MKK και ΣΚ2 $b=0,1$ $d_{10}=d_{20}=d_{21}=0,001$ $R=5$	38
Γράφημα 12 MKK και ΣΚ2 $b=0,1$ $d_{10}=d_{20}=d_{21}=0,001$ $R=10$	39
Γράφημα 13 ΧΚΚ και ΣΚ1 $b=0,1$ $d_{10}=d_{20}=d_{21}=0,001$ $R=5$	40
Γράφημα 14 ΧΚΚ και ΣΚ1 $b=0,1$ $d_{10}=0,01$ $d_{20}=0,001$ $d_{21}=0,01$ $R=5$	41
Γράφημα 15 MKK και ΣΚ1 $b=d_{10}=d_{20}=d_{21}=0,1$ $R=2$	42
Γράφημα 16 MKK και ΣΚ1 $b=d_{10}=d_{20}=d_{21}=0,1$ $R=5$	43
Γράφημα 17 MKK και ΣΚ1 $b=d_{10}=d_{20}=d_{21}=0,1$ $R=10$	44
Γράφημα 18 MKK και ΣΚ2 $b=d_{10}=d_{20}=d_{21}=0,1$ $R=2$	45
Γράφημα 19 MKK και ΣΚ2 $b=d_{10}=d_{20}=d_{21}=0,1$ $R=5$	46
Γράφημα 20 MKK και ΣΚ2 $b=d_{10}=d_{20}=d_{21}=0,1$ $R=10$	47
Γράφημα 21 ΧΚΚ και ΣΚ1 $b=d_{10}=d_{20}=d_{21}=0,1$ $R=5$	48

Κατάλογος Πινάκων

Πίνακας 1 Τελικά ποσοστά μόλυνσης (MKK ΣΚ1).....	48
--	----



Πίνακας 2 Τελικά ποσοστά μόλυνσης (ΜΚΚ ΣΚ2)	49
Πίνακας 3 Τελικά ποσοστά μόλυνσης (ΧΚΚ ΣΚ1).....	49