



Πανεπιστήμιο Μακεδονίας  
ΠΜΣ Πληροφορικά Συστήματα  
Τεχνολογίες Τηλεπικοινωνιών & Δικτύων  
Καθηγητές: Α.Α. Οικονομίδης & Α. Πομπόρτσης

University of Macedonia  
Master Information Systems  
Networking Technologies  
Professors: A.A. Economides &  
A. Pomportsis

## ΤΕΧΝΟΛΟΓΙΕΣ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ ΚΑΙ ΔΙΚΤΥΩΝ

Β' ΕΞΑΜΗΝΟ

Αντικείμενο Εργασίας: Δωρεάν λογισμικό για προγραμματισμό και σχεδίαση δικτύων  
Ονοματεπώνυμο Φοιτητή: Λαμπριανίδης Πέτρος  
Αριθμός Μητρώου: 25/06





Πανεπιστήμιο Μακεδονίας  
ΠΜΣ Πληροφορικά Συστήματα  
Τεχνολογίες Τηλεπικοινωνιών & Δικτύων  
Καθηγητές: Α.Α. Οικονομίδης & Α. Πομπόρτσης

University of Macedonia  
Master Information Systems  
Networking Technologies  
Professors: A.A. Economides &  
A. Pomportsis

## NETWORK AND COMMUNICATION TECHNOLOGIES

SEMESTER B

Subject Of Project: Free Tools For Network Planning And Design

Student Name: Lambrianidis Petros

Student Code: 25/06



THESSALONIKI -2007-

## ΠΕΡΙΕΧΟΜΕΝΑ

<b>Iα.</b> Περίληψη.....	σελ 4
<b>Iβ.</b> Overview.....	σελ 4
<b>II.</b> Παρουσίαση του θέματος.....	σελ 5
<b>III.</b> Εισαγωγή.....	σελ 6
<b>IV.</b> Παρουσίαση των εργαλείων.....	σελ 8
<u>1<sup>ο</sup></u> εργαλείο – PlotPath.....	σελ 8
<u>2<sup>ο</sup></u> εργαλείο – JUNG.....	σελ16
<b>V.</b> Συμπεράσματα.....	σελ25
<b>VI.</b> Βιβλιογραφία και Δικτυακές Πηγές.....	σελ27

## TABLE OF CONTENTS

<b>Iα.</b> Summary (in Greek).....	pg 4
<b>Iβ.</b> Overview (in English).....	pg 4
<b>II.</b> Presentation of subject .....	pg 5
<b>III.</b> Introduction.....	pg 6
<b>IV.</b> Presentation of tools .....	pg 8
<u>1<sup>o</sup> tool</u> – PlotPath.....	pg 8
<u>2<sup>o</sup> tool</u> – JUNG.....	pg 16
<b>V.</b> Conclusions .....	pg 25
<b>VI.</b> Bibliography and Network Sources.....	pg 27

## Ια. ΠΕΡΙΛΗΨΗ

Είναι πλέον γεγονός στις μέρες μας η ραγδαία ανάπτυξη της επιστήμης και τεχνολογίας των ηλεκτρονικών υπολογιστών αλλά και του διαδικτύου. Οι βασισμένες στο Internet τεχνολογίες και εφαρμογές έχουν φέρει επανάσταση στον τρόπο με τον οποίο επικοινωνούμε, εργαζόμαστε, συναλλασσόμαστε και συλλέγουμε ή ανταλλάσσουμε πληροφορίες και δεδομένα. Τα δίκτυα αποτελούσαν ανέκαθεν ένα τρόπο διασύνδεσης μεταξύ δύο (ή και περισσότερων) υπολογιστών και κατ' επέκταση χρηστών. Γίνεται επομένως κατανοητό πως είναι περισσότερο από ποτέ επιτακτική η ανάγκη δημιουργίας εφαρμογών και εργαλείων για τον προγραμματισμό και τη σχεδίαση δικτύων προκειμένου να εξετάζεται η βιωσιμότητα και λειτουργικότητα του δικτυακού τύπου που θα υλοποιηθεί και να διαπιστώνεται εάν τα τεχνολογικά χαρακτηριστικά και η δομή της δικτύωσης, ανταποκρίνονται στις εκάστοτε ανάγκες των χρηστών, ενσωματώνονται πλήρως με το υπόλοιπο δίκτυο, ελαχιστοποιούν τον κίνδυνο κατάρρευσης του δικτύου και τέλος ικανοποιούν κάποια κριτήρια ελάχιστης απόδοσης.

Η παρούσα εργασία ασχολείται με εργαλεία προγραμματισμού και σχεδίασης δικτύων που μπορούν να παρέχουν πολύτιμες πληροφορίες για τον τρόπο λειτουργίας τους αλλά και για την απεικόνισή τους ικανοποιώντας κατά αυτόν τον τρόπο όχι μόνο οποιονδήποτε θελήσει να συλλέξει στοιχεία για κάποιο δίκτυο αλλά και τους υπεύθυνους για την υλοποίηση παρέχοντας τους ένα τρόπο εκ των προτέρων μελέτης, παρατήρησης και συλλογής πληροφοριών.

Τα εργαλεία που θα αναπτυχθούν διατίθενται δωρεάν στο παγκόσμιο διαδίκτυο και δεν αποτελούν εμπορικό υλικό κάποιας εταιρίας λογισμικού. Πρόκειται συνεπώς για freeware υλικό.

## Ιβ. OVERVIEW

The rapid growth of the science and technology of computers, as well as the internet is an incontrovertible reality nowadays. Technologies and applications based on the internet have led to a revolution in the way we communicate, work and collect or exchange information and data. Networks always constituted a way of interconnection between two (or even more) computers and at an extension of users. As a result, the need of the creation of tools and applications in order to plan and design networks becomes currently more and more vital. On one hand those tools will aim to examine the viability and operation of the network places that will be created. On the other hand they will aim to establish of whether the technological characteristics and the structure of the networks is responding to the needs of all users, whether they are completely incorporated with the rest of the network, as well as to minimise the danger of network to collapse and finally to satisfy certain criteria of minimal output.

This project deals with tools of planning and designing networks that can provide valuable information about their way of operation, but also on their visualization. This way not only people that want to collect data about a specific network will be satisfied, but also persons in charge for the concretisation, as they will be provided by a preliminary study, observation and collection of information.

The tools that will be thoroughly analyzed below, are distributed free of charge in the global internet and do not constitute commercial material of some software company. It is consequently a freeware material.

## II. ΠΑΡΟΥΣΙΑΣΗ ΤΟΥ ΘΕΜΑΤΟΣ

Το κύριο θέμα το οποίο πραγματεύεται η παρούσα εργασία είναι τα «δωρεάν» εργαλεία για σχεδίαση και προγραμματισμό δικτύων, τα οποία κυκλοφορούν ελεύθερα στο διαδίκτυο. Με μια προσπάθεια αναζήτησης για τέτοια tools μπορεί κανείς να εντοπίσει μερικά από αυτά, ωστόσο γίνεται εύκολα αντιληπτό πως η συντριπτική πλειοψηφία αποτελεί έτοιμα εμπορικά προϊόντα διάφορων επιχειρήσεων λογισμικού και δικτυακών συμβουλών και λύσεων κυρίως. Εδώ θα παρουσιάσουμε εκτενώς 2 free tools, το PlotPath και το JUNG. Τα εργαλεία αυτά είναι παράλληλα και ανοικτού κώδικα (open source) γεγονός που σημαίνει ότι μπορεί ο ενδιαφερόμενος χρήστης να προμηθευτεί και τον πηγαίο κώδικα προγράμματος και να τον τροποποιήσει εάν επιθυμεί ανάλογα με τις δικές του ανάγκες και σε συνδυασμό με τις παραμέτρους που προσπαθεί να ενσωματώσει σε κάποιο δίκτυο που μελετά ή υλοποιεί. Τα δύο αυτά εργαλεία είναι εργαλεία που χρησιμοποιούνται στη Java και αφορούν γραφήματα, κόμβους, ζεύξεις μεταξύ των κόμβων, επικοινωνία των clients μεταξύ τους, δημιουργούν τις προϋποθέσεις για καλύτερη επικοινωνία μέσω ενός δικτύου, επεξηγούν τις ανάγκες και τις δυνατότητες για το σχεδιασμό και την ανάπτυξη ενός όσο το δυνατόν περισσότερο σταθερού και εύχρηστου δικτύου. Αναλύεται λεπτομερώς η δομή τους, ο τρόπος λειτουργίας τους, οι αλγόριθμοι που χρησιμοποιούν και η υλοποίησή τους μέσω βιβλιοθηκών της Java και των Java Applets. Όλα τα ανωτέρω αποτελούν τμήμα του τέταρτου μέρους (IV) της εργασίας.

Παράλληλα, στο μέρος III, στην εισαγωγή που ακολουθεί, παρατίθενται ορισμένες πληροφορίες σχετικά με τη σχεδίαση και ανάπτυξη δικτύων, γίνεται μια συνοπτική αναφορά στη μεθοδολογία σχεδιασμού και στα στρώματα προγραμματισμού των δικτύων και τέλος, τονίζεται η σημασία που έχει η πρόβλεψη στο θέμα σχεδιασμού δικτύων. Όπως έχει προαναφερθεί μία εκ των χρησιμότητων των εργαλείων αυτών είναι να η απλοποίηση και η εύκολη απεικόνιση των δικτύων που επιθυμούμε να υλοποιήσουμε. Άλλωστε προτού τεθεί σε εφαρμογή η κατασκευή κάποιου δικτύου προηγείται η προσομοιωμένη ανάλυση και σχεδίασή του και η απεικόνισή του. Γίνεται λοιπόν αντιληπτή η σημασία της πρόβλεψης.

### III. ΕΙΣΑΓΩΓΗ

#### Λίγα λόγια σχετικά με τον προγραμματισμό και την σχεδίαση δικτύων (network planning and design)

Ο προγραμματισμός και σχεδιασμός δικτύων είναι μια επαναληπτική διαδικασία, κατά την οποία περιγράφονται το τοπολογικό σχέδιο (topological design), η σύνθεση του δικτύου (network-synthesis) και η πραγματοποίηση του δικτύου (network-realization) και η οποία στοχεύει στην εξασφάλιση ότι ένα νέο δίκτυο ή μια υπηρεσία ικανοποιεί τις ανάγκες των συνδρομητών, χρηστών και των χειριστών. Η διαδικασία μπορεί να προσαρμοστεί σύμφωνα με τις εκάστοτε ανάγκες και σε νέα δίκτυα ή υπηρεσίες. Αυτό είναι μια εξαιρετικά σημαντική διαδικασία που πρέπει να εκτελεσθεί πριν από την καθιέρωση ενός νέου δικτύου ή υπηρεσίας τηλεπικοινωνιών.

Μια παραδοσιακή μεθοδολογία σχεδιασμού δικτύων περιλαμβάνει τέσσερα στρώματα προγραμματισμού:

- επιχειρησιακός προγραμματισμός
- μακροπρόθεσμος και μεσοπρόθεσμος προγραμματισμός δικτύων
- βραχυπρόθεσμος προγραμματισμός δικτύων
- διαδικασίες και συντήρηση.

Κάθε ένα από αυτά τα επίπεδα ενσωματώνει τα σχέδια για τους διαφορετικούς χρονικούς ορίζοντες, δηλ. το στρώμα επιχειρησιακού προγραμματισμού καθορίζει τον προγραμματισμό που ο χειριστής πρέπει να εκτελέσει για να εξασφαλίσει ότι το δίκτυο θα αποδώσει όπως απαιτείται για την προοριζόμενη διάρκεια ζωής του. Οι διαδικασίες και το στρώμα συντήρησης, εντούτοις, εξετάζουν πώς το δίκτυο θα τρέξει σε καθημερινή βάση.

Η διαδικασία προγραμματισμού-σχεδιασμού δικτύων αρχίζει με την απόκτηση των εξωτερικών πληροφοριών. Αυτό περιλαμβάνει:

- τις προβλέψεις για το πώς το νέο δίκτυο/η υπηρεσία θα λειτουργήσει
- τις οικονομικές πληροφορίες σχετικά με τις δαπάνες και το κόστος υλοποίησης, και τέλος
- τις τεχνικές λεπτομέρειες των δυνατοτήτων του δικτύου.

Πρέπει να ληφθεί υπόψη ότι ο προγραμματισμός ενός νέου δικτύου περιλαμβάνει την εφαρμογή του νέου συστήματος στα πρώτα τέσσερα επίπεδα του μοντέλου-προτύπου OSI. Αυτό σημαίνει ότι προτού να αρχίσει ακόμη η διαδικασία προγραμματισμού δικτύων, κάποιες επιλογές πρέπει να γίνουν όπως το ποια πρωτόκολλα και ποιες τεχνολογίες μετάδοσης θα χρησιμοποιηθούν. Μόλις ληφθούν οι αρχικές αποφάσεις, η διαδικασία προγραμματισμού δικτύων περιλαμβάνει τρία κύρια βήματα:

- **Τοπολογικό σχέδιο:** Αυτό το στάδιο περιλαμβάνει τις αποφάσεις σχετικά με το πού να τοποθετηθούν τα συστατικά του δικτύου (κόμβοι, servers, routers,

πύλες διαδικτύου κ.α.) και πώς θα πραγματοποιηθεί η μεταξύ τους συνδεσμολογία. Οι (τοπολογικές) μέθοδοι βελτιστοποίησης που μπορούν να χρησιμοποιηθούν σε αυτό το στάδιο προέρχονται από έναν τομέα των μαθηματικών αποκαλούμενο Θεωρία Γραφικών Παραστάσεων. Αυτές οι μέθοδοι περιλαμβάνουν τον καθορισμό των δαπανών της μετάδοσης και του κόστους της μετατροπής, και με αυτόν τον τρόπο τον καθορισμό της βέλτιστης μήτρας σύνδεσης και της θέσης των διακλαδώσεων και των διαχωριστών.

- **Σύνθεση Δικτύου:** Αυτό το στάδιο περιλαμβάνει τον καθορισμό του μεγέθους και του βαθμού χρήσης των συστατικών, υπό την έννοια των κριτηρίων απόδοσης όπως είναι π.χ. ο Βαθμός Υπηρεσίας (Grade Of Service GoS). Η μέθοδος που χρησιμοποιείται είναι γνωστή ως "μη γραμμική βελτιστοποίηση", και περιλαμβάνει τον προκαθορισμό της τοπολογίας, το απαραίτητο GoS, το κόστος της μετάδοσης, κ.λπ., και τη χρησιμοποίηση αυτών των πληροφοριών για να υπολογίσει ένα σχέδιο δρομολόγησης, και το μέγεθος των συστατικών.
- **Πραγματοποίηση δικτύου:** Αυτό το στάδιο περιγράφει τον τρόπο κάλυψης των απαιτήσεων δυναμικότητας-χωρητικότητας του δικτύου και την εξασφάλιση αξιοπιστίας μέσα στο δίκτυο. Η χρησιμοποιούμενη μέθοδος είναι γνωστή ως "Multicommodity Flow Optimization", και περιλαμβάνει τον καθορισμό όλων των πληροφοριών σχετικά με την ζήτηση του δικτύου, τις δαπάνες και την αξιοπιστία, και έπειτα τη χρησιμοποίηση αυτών των πληροφοριών για να υπολογίσει ένα πραγματικό φυσικό σχέδιο κυκλωμάτων.

Αυτά τα βήματα είναι αλληλένδετα και επομένως εκτελούνται επαναληπτικά και παράλληλα το ένα με το άλλο. Η διαδικασία προγραμματισμού είναι ιδιαίτερα σύνθετη, γεγονός που σημαίνει ότι σε κάθε επανάληψη, ένας αναλυτής πρέπει να αυξήσει τους ορίζοντες προγραμματισμού του, και με αυτές τις ενέργειες, πρέπει να παραγάγει τα σχέδια για τα διάφορα στρώματα που περιγράφονται ανωτέρω.

### Ο ρόλος της πρόβλεψης

Κατά τη διάρκεια της διαδικασίας του προγραμματισμού και του σχεδιασμού δικτύων, είναι απαραίτητο να υπολογιστεί η αναμενόμενη ένταση κυκλοφορίας και ως εκ τούτου και το αναμενόμενο φορτίο κυκλοφορίας που το δίκτυο πρέπει να υποστηρίξει. Εάν ένα δίκτυο παρόμοιας φύσης υπάρχει ήδη, μπορεί να είναι δυνατό να γίνουν μετρήσεις κυκλοφορίας ενός τέτοιου δικτύου και να χρησιμοποιηθούν τα στοιχεία που προκύπτουν για να υπολογιστεί το ακριβές φορτίο κυκλοφορίας. Εντούτοις, όπως είναι πιθανότερο στις περισσότερες περιπτώσεις, εάν δεν υπάρχει κανένα παρόμοιο δίκτυο, τότε οι αρμόδιοι για το σχεδιασμό δικτύων πρέπει να υιοθετήσουν μεθόδους πρόβλεψης για να υπολογιστεί την αναμενόμενη ένταση κυκλοφορίας. Η διαδικασία πρόβλεψης περιλαμβάνει διάφορα βήματα ως εξής:

- Καθορισμός του προβλήματος
- Απόκτηση στοιχείων
- Επιλογή της μεθόδου πρόβλεψης
- Ανάλυση/πρόβλεψη
- Τεκμηρίωση και ανάλυση των αποτελεσμάτων.



## Διαστασιολόγηση (Dimensioning)

Ο σκοπός διαστασιολόγησης ενός δικτύου είναι να καθοριστούν οι ελάχιστες απαιτήσεις δυναμικότητας και χωρητικότητας που θα επιτρέπουν την ικανοποίηση- κάλυψη ενός αναγκαίου ελάχιστου αποδεκτού Βαθμού Υπηρεσιών (GoS) και Ποιότητας Υπηρεσιών (Quality OF Service QoS). Για να γίνει αυτό, η διαστασιολόγηση περιλαμβάνει τον προγραμματισμό για την κυκλοφορία ώρας αιχμής, δηλ. εκείνη την ώρα κατά τη διάρκεια της ημέρας κατά την οποία η ένταση κυκλοφορίας είναι στην αιχμή της.

Η διαδικασία διαστασιολόγησης περιλαμβάνει τον καθορισμό της τοπολογίας του δικτύου, τη δρομολόγηση του σχεδίου, της μήτρας κυκλοφορίας και των απαιτήσεων GoS, και με τη χρησιμοποίηση αυτών των πληροφοριών καθορίζεται η μέγιστη διαχειριζόμενη δυναμικότητα κλήσης των ζεύξεων και ο μέγιστος αριθμός καναλιών που απαιτούνται μεταξύ των διακλαδώσεων.

Ένας κανόνας διαστασιολόγησης είναι ότι ο αρμόδιος για το σχεδιασμό πρέπει να εξασφαλίσει ότι το φορτίο κυκλοφορίας δεν πρέπει ποτέ να πλησιάζει ένα φορτίο 100%. Για να υπολογίσει το σωστό dimensioning προκειμένου να είναι συνεπής με τον ανωτέρω κανόνα, ο αρμόδιος για το σχεδιασμό πρέπει να πάρει τις τρέχουσες μετρήσεις της κυκλοφορίας του δικτύου, και να διατηρεί και να αναβαθμίζει συνεχώς τους πόρους για να καλύψει τις μεταβαλλόμενες απαιτήσεις.

## **IV. ΠΑΡΟΥΣΙΑΣΗ ΕΡΓΑΛΕΙΩΝ ΣΧΕΔΙΑΣΜΟΥ**

### 1ο **□□□□□□**: *PlotPaths*

Το free tool PlotPaths επιδεικνύει προς τα εμπρός ή και αντίστροφα προς τα πίσω ροή δεδομένων μέσω ενός μονοπατιού δικτύου από μια μοναδική πηγή σε έναν ή περισσότερους προορισμούς. Πολλοί παράγοντες, συμπεριλαμβανομένης της εξισορρόπησης φορτίων, της αλλαγής τοπολογίας και της ασύμμετρης διαδρομής, έχουν ως αποτέλεσμα μια περίπλοκη δομή γραφημάτων για τα δεδομένα που μεταφέρονται από μια διαδρομή δικτύου που συγκεντρώνονται κατά τη διάρκεια του χρόνου. Και η απόκλιση και η επαναφορά σε προηγούμενη κατάσταση (reconversion) των διαδρομών εμφανίζονται συχνά κάνοντας την απεικόνιση των διαδρομών (κανάλια σε ακολουθία) έναν προκλητικό στόχο. Το PlotPaths απεικονίζει, όσο το δυνατόν πιο σαφώς, τις διαδρομές που αρχίζουν από έναν ξενιστή-πηγή σε έναν ή περισσότερους προορισμούς. Παρουσιάζει τις συνδέσεις μεταξύ των ενδιάμεσων κόμβων κατά μήκος κάθε πορείας, συντηρώντας τους υψηλότερους σε ιεραρχία σχηματισμούς ομάδας, όπως π.χ. τα αυτόνομα συστήματα. Αυτή η προσέγγιση δεν ισχύει σε περίπτωση όπου σύνολα από πληροφορίες μεταφέρονται μέσω κόμβων εκ των οποίων πολλοί έχουν παρόμοιες φυσικές θέσεις, και επομένως επικαλύπτονται. Με το PlotPaths σχεδιάζουμε μια γραφική παράσταση τοπολογίας σε στήλες, βασιζόμενες στις οργανωτικές οντότητες, (π.χ. χώρες) και σε σειρές, βασιζόμενες στην τοπολογική απόσταση (αριθμός των hops) κάθε κόμβου από την πηγή. Το πρόγραμμα PlotPaths εκτελεί τις ακόλουθες διαδικασίες:

- Διαβάζει το input και τις παραμέτρους από τα αρχεία τμημάτων
- Υπολογίζει την τοπολογική απόσταση (βάθος hop) από την πηγή σε κάθε κόμβο
- Εφαρμόζει τον αλγόριθμο τοποθέτησης κόμβων
- Αποτυπώνει το output

Φάση 1. Το PlotPaths διαβάζει το input από δύο σχετικά αρχεία δεδομένων. Το αρχείο για τις διαδρομές (path file) διευκρινίζει τα χαρακτηριστικά όλων των διαδρομών στο σύνολο δεδομένων, συμπεριλαμβανομένης της πηγής, του προορισμού και των ενδιάμεσων κόμβων για κάθε ένα από αυτά. Το αρχείο κόμβων (nodes file) επιτρέπει διευκρίνιση του ονόματος κάθε κόμβου καθώς επίσης και του αριθμού και του ονόματος των στηλών του. Παραδείγματος χάριν, θα μπορούσε να επιλεγεί το αυτόνομο σύστημα (Autonomous System) ως μεταβλητή στηλών και να χρησιμοποιηθεί ο αριθμός των κόμβων του ως αριθμός στηλών και το όνομά του ως όνομα στηλών. Με το PlotPaths είναι επίσης επιτρεπτό να μείνει η στήλη του κόμβου απροσδιόριστη. Μια ομάδα path input files μπορεί να μοιραστεί ένα ενιαίο αρχείο κόμβων.

Φάση 2. Ο υπολογισμός της τοπολογικής απόστασης (βάθος κόμβων) περιλαμβάνει την κατασκευή ενός κατευθυνόμενου (directed) γραφήματος που θα παράσχει μια συντεταγμένη για τον άξονα Y για κάθε κόμβο. Η τοποθέτηση αρχίζει στην πηγή, με τους κόμβους κατά μήκος της διαδρομής να αυξάνουν σε βάθος. Όταν πολλαπλές διαδρομές οδηγούν σε έναν κόμβο παράγοντας διαφορετικά βάθη κόμβων, το PlotPaths χρησιμοποιεί το μεγαλύτερο, επεκτείνοντας τελικά τις συνδέσεις κατά μήκος των μικρών διαδρομών παρά συσσωρεύοντας τους κόμβους κατά μήκος των μεγάλων διαδρομών. Το τελικό βάθος κάθε κόμβου προέρχεται από όλες τις διαδρομές που φορτώνονται στη γραφική παράσταση.

Φάση 3. Το PlotPaths οργανώνει χωροταξικά τους κόμβους σε σειρές μέσα στις προκαθορισμένες στήλες τους. Οι κόμβοι στο ίδιο βάθος μέσα σε μια στήλη ταξινομούνται και χωρίζονται κατά διαστήματα για να μεγιστοποιήσουν την ορατότητα στο γράφημα. Περιγράφουμε τον αλγόριθμο σχεδιαγράμματος λεπτομερώς παρακάτω.

Φάση 4. Παράγει τη σχηματοποιημένη έξοδο (output) για την εξέταση του εργαλείου.

*Παρουσίαση και ανάλυση του αλγορίθμου που χρησιμοποιείται στο tool PlotPaths(φάσεις 2 και 3). Σημαντικά και λεπτομερή βήματα για την παραγωγή εικόνας και γραφήματος:*

1. υπολογισμός του βάθους κόμβων
2. δημιουργία των σειρών και των στηλών
3. διασκορπισμός επιπρόσθετων στηλών (προαιρετικό)
4. ταξινόμηση των στηλών
5. χωροταξία των στηλών
6. αποφυγή κάθετης επικάλυψης συνδέσεων
7. ορισμός των συντεταγμένων X και Y στους κόμβους
8. τακτοποίηση-τοποθέτηση των κόμβων οριζόντια

### **1)Υπολογισμός του βάθους κόμβων:**

Χρησιμοποιούμε τα αρχεία διαδρομών εισαγωγής (input path file) για να δημιουργήσουμε μια ενιαία κατευθυνόμενη γραφική παράσταση. Προσθέτουμε κάθε διαδρομή χωριστά στη γραφική παράσταση, χρησιμοποιώντας τους υπάρχοντες κόμβους (κόμβοι κοινοί σε άλλες διαδρομές) σε περίπτωση ανάγκης. Επειδή η γραφική παράσταση μπορεί να περιλάβει αμφίδρομα δεδομένα, προσθέτουμε τους κόμβους των αντίστροφων διαδρομών στο γράφημα σε αντίστροφη σειρά. Μόλις προστεθούν όλες οι διαδρομές, υπολογίζουμε το μέγιστο βάθος κάθε κόμβου.

### **2)Δημιουργία των σειρών και των στηλών:**

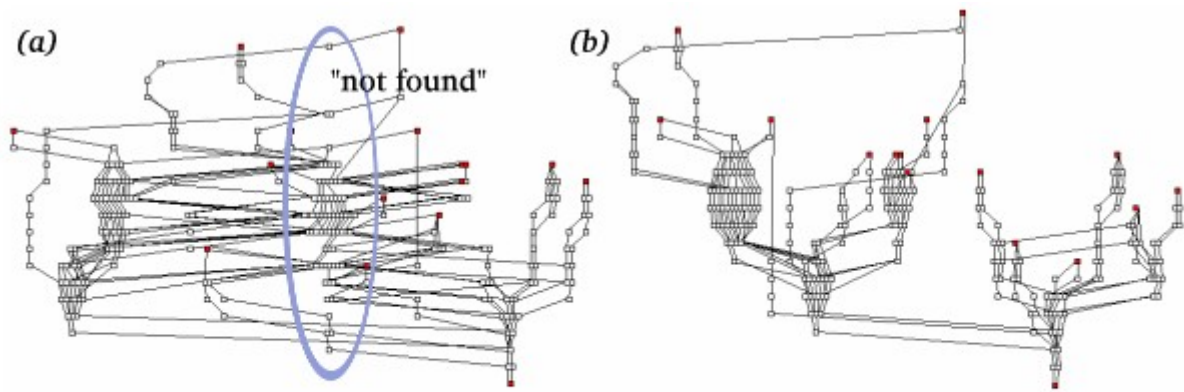
Σε κάθε μεμονωμένο κόμβο ορίζεται μια θέση σειρών που βασίζεται στο μέγιστο βάθος του στη γραφική παράσταση μόλις προστεθούν όλες οι πορείες. Η θέση στηλών για κάθε κόμβο διευκρινίζεται στο αρχείο εισαγωγής κόμβων (node input file). Αυτές οι συντεταγμένες θέσης παρέχουν ένα πλαίσιο για την επικάλυψη συνδέσεων και τους αλγορίθμους ελαχιστοποίησης απόστασης συνδέσεων. Ειδικότερα, το διάστημα των συστάδων κόμβων (σύμπλεγμα κόμβων) σε ένα ενιαίο βάθος κόμβων μέσα σε μια στήλη στηρίζεται στις προκαθορισμένες τιμές σειρών και στηλών. Εφαρμόζουμε τους αλγορίθμους απλοποίησης σχεδιαγράμματος για να επανατοποθετήσουμε το σύμπλεγμα κόμβων προκειμένου να παραχθεί ένα γράφημα που ελαχιστοποιεί την επικάλυψη και μεγιστοποιεί τις ξεκάθαρες διαδρομές.

### **3)Διασκορπισμός των πρόσθετων στηλών:**

Όπως αναφέρθηκε και παραπάνω, το PlotPaths απαιτεί την διευκρίνιση της αξίας στηλών για κάθε κόμβο στο αρχείο εισαγωγής κόμβων. Εντούτοις, πολλά πραγματικά σύνολα δεδομένων μπορούν να είναι διφορούμενα. Κατά συνέπεια το PlotPaths επιτρέπει να διευκρινιστεί στο αρχείο εισαγωγής ότι η αξία στηλών για έναν κόμβο είναι άγνωστη.

Το PlotPaths μπορεί να χειριστεί τους κόμβους με μια άγνωστη αξία στηλών με δύο τρόπους. Ο πρώτος είναι με την παραγωγή μιας χωριστής στήλης για όλους τους κόμβους που στερούνται μιας ιδιότητας στηλών. Εναλλακτικά, μπορεί να προσπαθήσει να τοποθετήσει τους κόμβους στην άγνωστη στήλη στη διευκρινισμένη στήλη με την οποία μοιράζεται το μέγιστο αριθμό συνδέσεων. Σε αυτό το σενάριο, το εργαλείο είναι σε θέση να προσπαθήσει την τοποθέτηση όλων των άγνωστων κόμβων εκτός από τους κόμβους πηγής και προορισμού.

Τα περάσματα αλγορίθμου διασποράς στηλών από το κατώτατο σημείο της άγνωστης στήλης στην κορυφή: το PlotPaths εξετάζει τους γονείς και τα παιδιά κάθε κόμβου και ορίζει τον άγνωστο κόμβο στη στήλη που περιέχει το μέγιστο αριθμό γονέων και παιδιών της. Σε περίπτωση που δύο ή περισσότερες στήλες έχουν τους ίδιους μέγιστους αριθμούς γονέων και παιδιών, το εργαλείο επιλέγει τυχαία μεταξύ των δεμένων στηλών. Εάν ένας άγνωστος κόμβος συνδέεται μόνο με άλλους άγνωστους κόμβους, δεν μπορεί να επανεκχωρηθεί.



(a) all columns are displayed as found in the database file

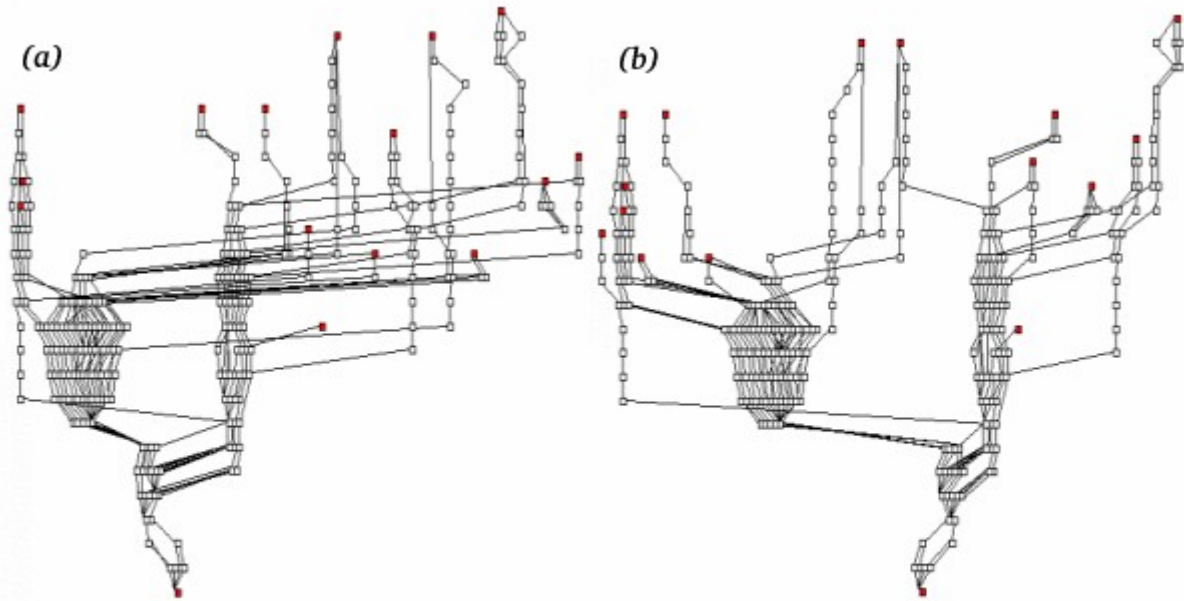
(b) the "not found" column has been dispersed into neighbor columns

#### **4) Ταξινόμηση στηλών:**

Σ' αυτό το στάδιο του αλγορίθμου, κάθε κόμβος έχει οριστεί μόνιμα σε μια στήλη. Πρέπει να κανονίσουμε τις στήλες για να ελαχιστοποιήσουμε και την επικάλυψη των συνδέσεων με άλλες στήλες και το γενικό μέγιστο μήκος συνδέσεων. Εφαρμόζουμε έναν «άπληστο» αλγόριθμο για την τοποθέτηση στηλών που εκτελεί μια πλεονασματική αναζήτηση όλων των πιθανών σχεδιαγραμμάτων.

Για να βρούμε τη βέλτιστη τοποθέτησή μας, ορίζουμε όλες τις στήλες στο "αποθετήριο" σύνολο. Αυτό είναι ένα άτακτο σύνολο στηλών που περιέχει την ομάδα των υποψηφίων προς τοποθέτηση. Ένα δεύτερο σύνολο περιέχει όλες τις στήλες που έχουν τοποθετηθεί. Ο αλγόριθμός μας επιλέγει μια στήλη από το "αποθετήριο" σύνολο και δοκιμάζει αυτήν σε όλες τις πιθανές θέσεις σχετικά με τις στήλες που τοποθετήθηκαν ήδη. Για κάθε θέση, δύο τιμές καταγράφονται: ο αριθμός συνδέσεων που διασχίζουν τη στήλη που τοποθετείται και τις άμεσα παρακείμενες στήλες αυτής (διατομή συνδέσεων), και το συνολικό μήκος όλων των συνδέσεων στο τοποθετημένο σύνολο.

Η ελαχιστοποίηση της διατομής συνδέσεων προηγείται από την ελαχιστοποίηση του γενικού μήκους συνδέσεων στην επιλογή μιας τοποθέτησης στηλών. Οι δεσμοί στο αποτέλεσμα διατομής στηλών «σπάζουν» με τη χρησιμοποίηση της ελάχιστης αξίας για το συνολικό αποτέλεσμα μήκους συνδέσεων. Μόλις καθοριστεί η βέλτιστη τοποθέτηση, η στήλη θεωρείται ως ταξινομημένη και η σειρά της μέσα στο τοποθετημένο σύνολο καθορίζεται. Όταν όλες οι αποθετήτες στήλες έχουν τοποθετηθεί, η τελική διαβάθμιση στηλών χρησιμοποιείται για να παραγάγει μια εικόνα του συνόλου δεδομένων.



(a) no column sorting

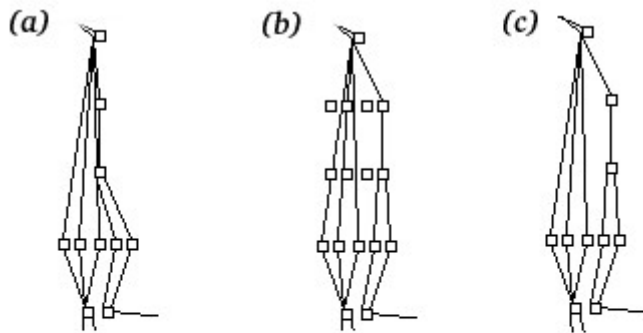
(b) columns sorted to minimize link overlap and link length

### **5)Χωροταξία στηλών:**

Σε αυτό το σημείο, οι σχετικές θέσεις όλων των στηλών καθορίζονται. Οι στήλες θα έχουν ποικίλους αριθμούς κόμβων και, μέσα σε μια στήλη, μερικές σειρές θα έχουν περισσότερους κόμβους από άλλες. Εάν οι στήλες ήταν ένα σταθερό μέγεθος, μερικοί κόμβοι θα συσσωρευόταν μαζί, κρύβοντας τις λεπτομέρειες των συνδέσεών τους, ενώ σε άλλες περιοχές οι κόμβοι θα ήταν αρκετά αραιοί. Για να εξετάσουμε αυτό το πρόβλημα διαφοροποιούμε, μεταβάλλουμε το διάστημα μέσα σε κάθε στήλη. Υπολογίζουμε το πλάτος της ευρύτερης σειράς μέσα σε μια στήλη και ορίζουμε εκείνη την αξία στη στήλη. Δεδομένου ότι ο άξονας των X αντιπροσωπεύει τις ομάδες που διευκρινίζονται στο αρχείο στοιχείων και δεν περιλαμβάνει μια φυσική μετρική απόσταση, το διάστημα μεταξύ των στηλών μπορεί να είναι αυθαίρετο. Επεκτείνουμε τις στήλες πριν από οποιαδήποτε ρύθμιση των κόμβων μέσα στις στήλες έτσι ώστε η θέση των στηλών στα συντεταγμένα διαστήματα να παραμένει σταθερή κατά τη διάρκεια της τελικής τοποθέτησης κόμβων. Αυτό διατηρεί τις σχετικές θέσεις όλων των κόμβων μέσα στην εικόνα.

### **6)Αποφυγή κάθετης επικάλυψης συνδέσεων:**

Περιστασιακά ο αλγόριθμός μας δημιουργεί πλαστές συνδέσεις ή κρύβει τις αληθινές συνδέσεις μεταξύ των διασυνδέσεων μέσα σε μια στήλη (βλ. (α) στο σχήμα που ακολουθεί). Αυτό εμφανίζεται όταν συγκλίνουν διαδρομές από πολλαπλούς κόμβους σε μια σειρά σε έναν ενιαίο προς τα πάνω κόμβο. Για να αποτρέψουμε την κάθετη επικάλυψη, παράγουμε τους λεγόμενους "κόμβους αντικατάστασης" (placeholder nodes) στο διάστημα επιφύλαξης μέσα στις σειρές που εκτείνονται από την κάθετη σύνδεση.



- (a) no special treatment of vertical links  
 (b) placeholder nodes explicitly displayed  
 (c) hidden placeholder nodes

### **7)Ορισμός των συντεταγμένων X και Y στους κόμβους:**

Μόλις καθοριστεί το οριζόντιο διάστημα, μπορούμε να ορίσουμε τις αρχικές συντεταγμένες του X και Y για κάθε κόμβο που βασίζεται στον αριθμό στηλών του, το διάστημα στηλών, και τον αριθμό σειρών μέσα στη στήλη. Αυτές οι τιμές κλιμακώνονται στις συντεταγμένες οθόνης για κάθε κόμβο. Η ανάθεση συντεταγμένων εκτελείται πριν από την τελική τοποθέτηση κόμβων έτσι ώστε η συντεταγμένη X μπορεί να χρησιμοποιηθεί για την ταξινόμηση των κόμβων μέσα σε μια στήλη.

### **8)Τακτοποίηση-τοποθέτηση των κόμβων οριζόντια:**

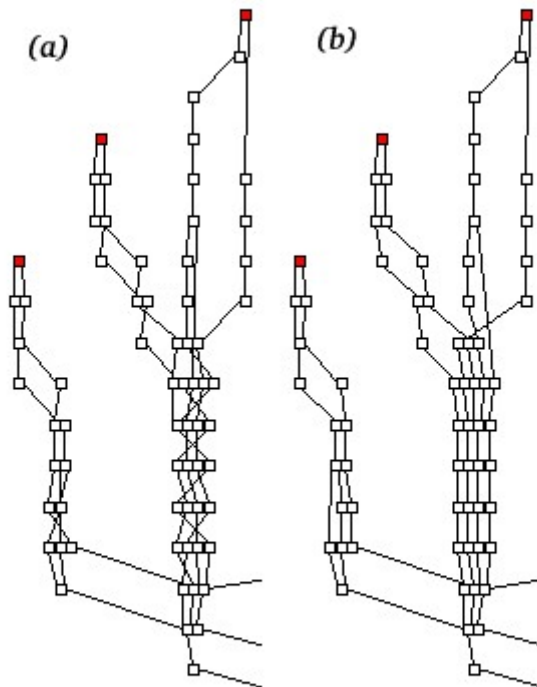
Μειώσαμε προηγουμένως τον αριθμό μπλεγμένων διαδρομών μεταξύ των κόμβων μέσω της ρύθμισης των στηλών και με τον προσδιορισμό του ικανοποιητικού διαστήματος στις σειρές με placeholder κόμβους. Παρά τα μέτρα αυτά, οι συνδέσεις μέσα σε μια στήλη μπορούν να επικαλύπτονται επανειλημμένα, παράγοντας μια μπλεγμένη γραφική παράσταση. Με τον υπολογισμό μιας βέλτιστης οριζόντιας ρύθμισης για τους κόμβους, εξασφαλίζουμε μέγιστη καθαρότητα διαδρομών στην τελική εικόνα.

Ο αλγόριθμος προχωρά από το κατώτατο σημείο στην κορυφή του γραφήματος. Μόλις τοποθετηθεί ένας κόμβος βασισμένος στους κατωτέρω κόμβους, δεν μπορεί να κινηθεί από τις μελλοντικές επαναλήψεις του αλγορίθμου. Για κάθε σειρά μιας στήλης στη γραφική παράσταση, οι κόμβοι μέσα στη σειρά τοποθετούνται με τη μέση τιμή της X-συντεταγμένης των γονέων τους (κόμβοι κάτω από την σύνδεση με αυτούς). Σε περίπτωση που δύο κόμβοι στην ίδια σειρά έχουν έναν γονέα από κοινού, οι κόμβοι τοποθετούνται με τη μέση τιμή των παιδιών τους (ανωτέρω κόμβοι με τους οποίους συνδέονται). Στα παιδιά, ο αλγόριθμος ανιχνεύει μόνο τις συνδέσεις έξω από την τρέχουσα στήλη δεδομένου ότι όλοι οι κόμβοι στη μη τοποθετημένη στήλη επάνω από το τρέχον επίπεδο έχουν την ίδια X-συντεταγμένη. Εάν μια αναζήτηση σε ένα βάθος 1 δεν σπάζει το δεσμό, επαναλαμβάνουμε την επαναληπτική εκτίμηση των μέσων X-αξιών πρώτα κάτω, κατόπιν επάνω από κάθε υψηλότερο κόμβο (παιδιά) έως ότου ικανοποιείται ο ένας από τρεις όρους: ο δεσμός είναι σπασμένος, η γραφική παράσταση δεν έχει άλλους κόμβους στο δεδομένο βάθος, ή έχουμε φθάσει στο προκαθορισμένο μέγιστο βάθος. Εάν ο δεσμός δεν

μπορεί να «σπάσει», οι κόμβοι τοποθετούνται τυχαία και ο αλγόριθμος προχωρεί στο επόμενο πιο υψηλό επίπεδο.

Κατά συνέπεια οι κόμβοι που ωθούνται στα αριστερά μιας στήλης θα τείνουν να έχουν τα παιδιά τους επάνω από ωθημένους στα αριστερά επίσης κόμβους. Αυτό επιτρέπει στο θεατή να κατανοήσει τις σχέσεις μεταξύ των κόμβων ευκολότερα. Το παράδειγμα (α) παρουσιάζει τα αποτελέσματα μετά από την εφαρμογή του αλγορίθμου στην μπλεγμένη γραφική παράσταση του (β)

### Arrange Nodes Horizontally:

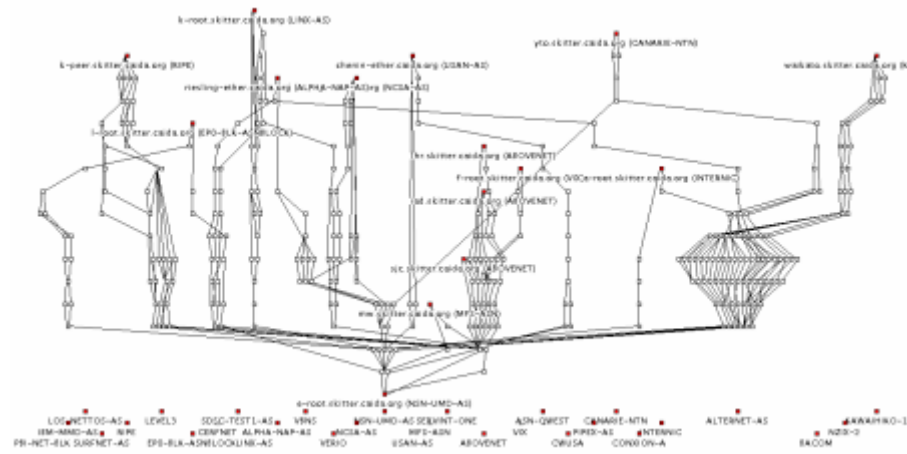


(a) no node sorting

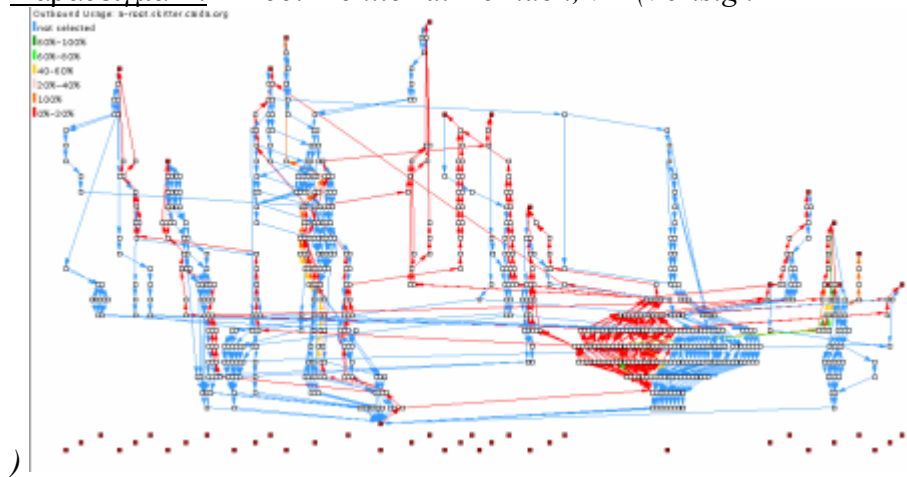
(b) node sorting to avoid link overlap

Παρακάτω παρατίθενται μερικά παραδείγματα σχεδίασης και προγραμματισμού δικτύων που βασίστηκαν στο εργαλείο PlotPath

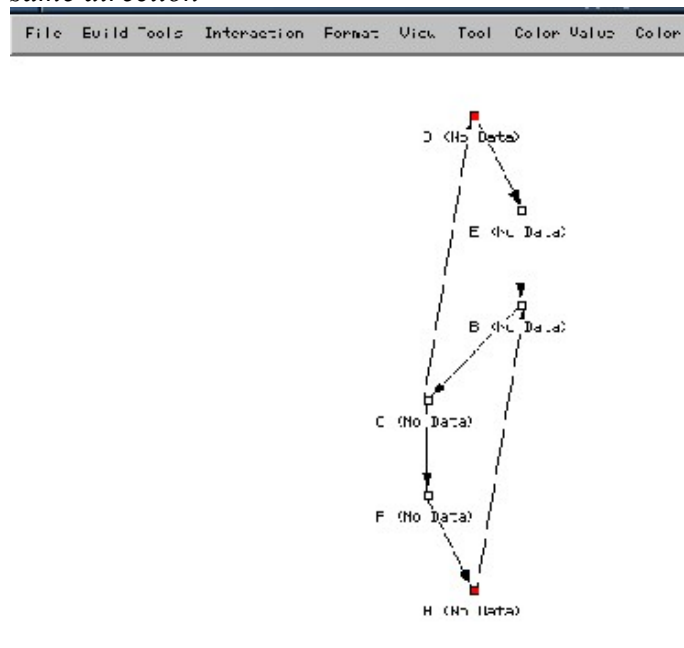
Παράδειγμα A: E-Root Monitor at Moffet Field, CA (NASA)



Παράδειγμα B: A-Root Monitor at Herndon, VA (Verisign)

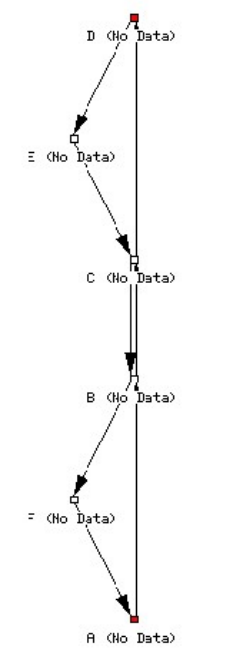


Παράδειγμα Γ1: Sample of paths showing inbound and outbound traversal in the same direction





Παράδειγμα Γ2: *Second sample of paths showing bidirectional links.*



## **2ο ΕΡΓΑΛΕΙΟ: JUNG**

Το JUNG (Java Universal Network/Graph Framework) αποτελεί ουσιαστικά μια “open-source” βιβλιοθήκη λογισμικού, η οποία παρέχει μια διαδομένη και ταυτόχρονα με δυνατότητα επέκτασης γλώσσα για τη διαμόρφωση, την ανάλυση και την οπτικοποίηση (modeling-analysis-visualisation) στοιχείων που μπορούν να απεικονιστούν και να παρουσιαστούν ως γραφική παράσταση ή δίκτυο. Είναι γραμμένο σε Java, γεγονός που επιτρέπει σε εφαρμογές του προγράμματος να εκμεταλλευτούν όλες τις εκτενείς δυνατότητες που παρέχει η Java API, καθώς επίσης και εκείνες που προσφέρονται από υπάρχουσες βιβλιοθήκες Java τρίτων.

Το JUNG είναι έτσι σχεδιασμένο ώστε να υποστηρίζει μια ευρεία γκάμα από συμβολισμούς και απεικονίσεις οντοτήτων και των σχέσεων τους, όπως για παράδειγμα κατευθυνόμενες και μη κατευθυνόμενες γραφικές παραστάσεις (directed and undirected graphs), πολύμορφες γραφικές παραστάσεις (multi-modal graphs), γραφικές παραστάσεις με παράλληλες άκρες (graphs with parallel edges) και hypergraphs. Παρέχει έναν μηχανισμό που επιτρέπει το σχολιασμό/εμπλουτισμό γραφικών παραστάσεων, οντοτήτων, και άλλων σχέσεων με μεταδεδομένα (metadata). Αυτό έχει σαν αποτέλεσμα τη δημιουργία αναλυτικών εργαλείων για σύνθετα σύνολα πληροφοριών, που μπορούν να εξετάσουν και να επεξεργαστούν τις σχέσεις μεταξύ των οντοτήτων, καθώς επίσης και τα μεταδεδομένα, που συνδέονται με κάθε οντότητα και σχέση.

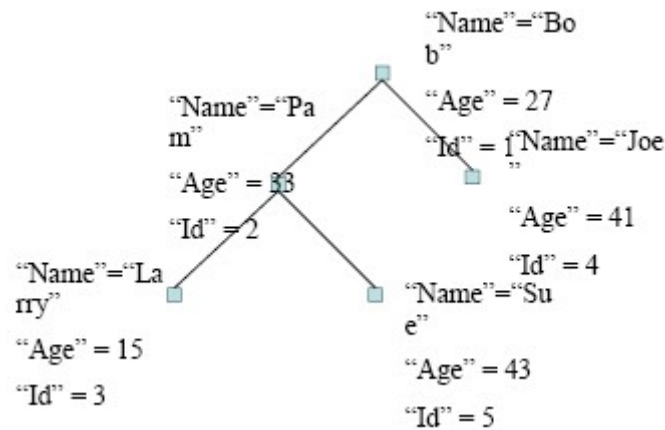
Η τρέχουσα διανομή/έκδοση της JUNG περιλαμβάνει εφαρμογές διάφορων αλγορίθμων από τη θεωρία των γραφικών παραστάσεων, την ανάσυρση δεδομένων, και την κοινωνική ανάλυση δικτύων. Ως παραδείγματα μπορούν να αναφερθούν οι σειρές εντολών για τις συστοιχίες (clustering), την αποσύνθεση (decomposition), τη βελτιστοποίηση (optimization), την τυχαία παραγωγή γραφικών παραστάσεων (random graph generation), τη στατιστική ανάλυση και τον υπολογισμό των

αποστάσεων δικτύων, των ροών (flows) και των μετρήσεων σημασίας (centrality, PageRank, HITS, κ.λ.π.).

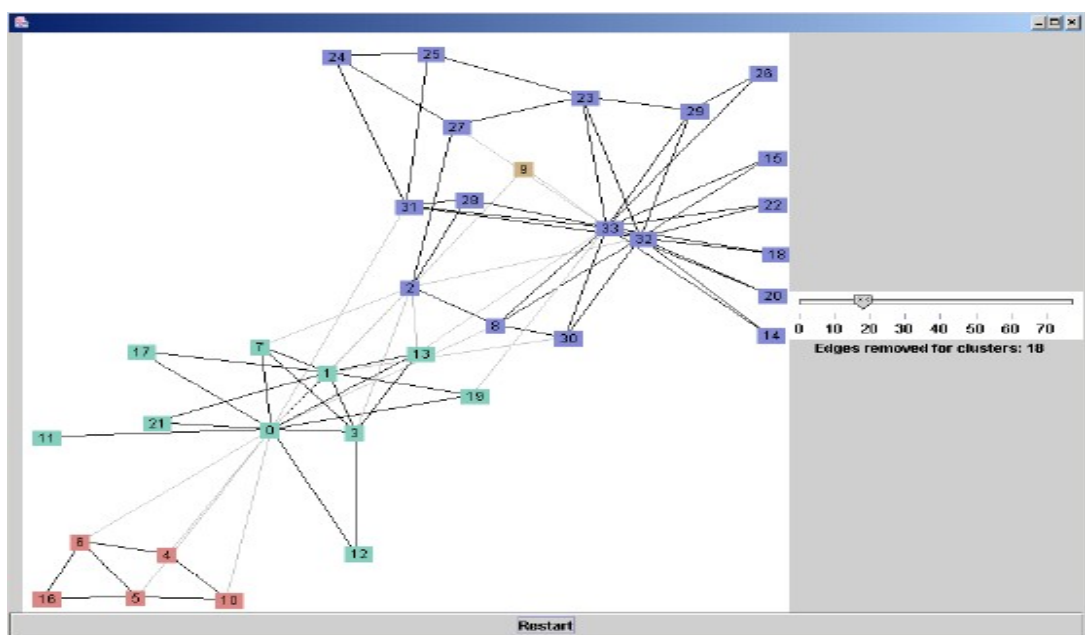
Το JUNG παρέχει επίσης ένα πλαίσιο απεικόνισης που καθιστά εύκολη τη κατασκευή εργαλείων για τη διαδραστική εξερεύνηση στοιχείων δικτύου. Οι χρήστες δύνανται να χρησιμοποιήσουν έναν από τους αλγορίθμους σχεδιαγράμματος (layout algorithms) που παρέχονται ή να χρησιμοποιήσουν το πλαίσιο, για να δημιουργήσουν σχεδιαγράμματα της αρεσκείας τους. Παράλληλα, παρέχονται και μηχανισμοί φιλτραρίσματος (filtering mechanisms) που επιτρέπουν στους χρήστες να επικεντρώνουν την προσοχή τους ή τους αλγορίθμους τους σε συγκεκριμένα σημεία της εκάστοτε γραφικής παράστασης που επεξεργάζονται.

Ως “open-source” βιβλιοθήκη, το JUNG παρέχει ένα συνηθισμένο πλαίσιο για την ανάλυση και σχεδιασμό γραφημάτων των δικτύων. Επιπλέον καθιστά ευκολότερη την εργασία για εκείνους που αξιοποιούν τις προσπάθειες ανάπτυξης των υπόλοιπων χρηστών και εκμεταλλεύονται σχετικά στοιχεία από τις έρευνές τους, αποφεύγοντας ανώφελο διπλό κόπο.

παράδειγμα1: γραφική παράσταση με τη χρήση του JUNG, όπου οι κόμβοι εμφανίζουν το όνομα και το προσδιοριστικό (id) τους



παράδειγμα2: java applet που αναπαριστά την σύνδεση κόμβων σε ένα δίκτυο



## Λόγοι και στόχοι δημιουργίας του προγράμματος JUNG

Το πρόγραμμα JUNG δημιουργήθηκε, με στόχο να καλύψει καταρχάς την ανάγκη για τη δημιουργία ενός γενικού, εύκαμπτου, και ισχυρού API για το χειρισμό, την ανάλυση, το σχεδιασμό και την απεικόνιση δικτύων και γραφικών παραστάσεων στην Java. Ένας δεύτερος λόγος είναι ότι κανένα από τα άλλα εργαλεία/APIs δεν ήταν κατάλληλο για τις συγκεκριμένες απαιτήσεις των μελετών που δούλευαν την δεδομένη περίοδο.

Το JUNG είναι ένα πλαίσιο στο οποίο μπορούν να δημιουργηθούν εφαρμογές και εργαλεία για το σχεδιασμό και την απεικόνιση πληροφοριών που σχετίζονται με δίκτυα και γραφικές παραστάσεις. Δύναται να χρησιμοποιηθεί είτε σε απλά αποκόμματα του κώδικα, είτε στα πλαίσια ενός περίπλοκου εργαλείου με ένα γραφικό user interface, προκειμένου να εξετάσει ιδέες.

Το JUNG δεν είναι και δεν έχει προσδιοριστεί να είναι ένα τελειωμένο εργαλείο. Μπορείτε να χτίσετε ένα εργαλείο που χρησιμοποιεί το πρόγραμμα, αλλά αυτό θα απαιτήσει κάποιο προγραμματισμό Java. Το JUNG παρέχει μερικά δείγματα μικρών εφαρμογών που χρησιμοποιούν το πρόγραμμα.

## Σχολιασμός του JUNG

Το πακέτο “utils” του JUNG παρέχει έναν ενσωματωμένο μηχανισμό, την κλάση `UserData class`, για τον εμπλουτισμό των γραφημάτων με πληροφορίες. Αυτός ο μηχανισμός είναι ο πιο κατάλληλος για το χειρισμό πληροφοριών, που είναι είτε προσωρινές, είτε ιδιοσυγκρασιακές (δηλ. πληροφορίες που δεν διαθέτουν ή έχουν ανάγκη όλα τα μέρη του γραφήματος).

Κάθε μια από τις εφαρμογές των γραφημάτων, σημείων και ακμών του JUNG (`graph`, `vertex`, and `edge implementations`), «καταλήγουν» σε `UserData`, το οποίο παρέχει τις ακόλουθες λειτουργίες:

- `addUserDatum (key, datum, copyaction)`: προσθέτει το καθορισμένο `datum` του αντικειμένου μαζί με το καθορισμένο κλειδί ανάκτησης (`retrieval key`) στην αποθήκη δεδομένων των χρηστών αυτού του αντικειμένου, με το καθορισμένο `copyaction`.
- `getUserDatum (key)`: ανακτά το αντικείμενο που έχει το καθορισμένο κλειδί ανάκτησης από την αποθήκη δεδομένων των χρηστών αυτού του αντικειμένου.
- `removeUserDatum (key)`: αφαιρεί το αντικείμενο που έχει το διευκρινισμένο κλειδί ανάκτησης από την αποθήκευση των στοιχείων χρηστών αυτού του αντικειμένου.
- `setUserDatum (key, datum, copyaction)`: αντικαθιστά το αντικείμενο (εφόσον υπάρχει) που έχει το καθορισμένο κλειδί ανάκτησης με το διευκρινισμένο `datum` και `copyaction` του αντικειμένου. Εάν δεν υπάρχει κάποιο τέτοιο αντικείμενο, τότε αυτή η μέθοδος είναι ισοδύναμη με την `addUserDatum (key, datum, copyaction)`.
- `importUserData (udc)`: παίρνει τα δεδομένα των χρηστών που αποθηκεύονται σε `udc` (την αποθήκη δεδομένων μιας άλλης παραμέτρου του γραφήματος) και τα αντιγράφει στην αποθήκη των δεδομένων των χρηστών του αντικειμένου, σύμφωνα με τους περιορισμούς της εντολής αντιγραφής (`copy action`) του κάθε `datum`.

- `getUserDatumKeyIterator ()`: Παρέχει ένα iterator του key set της αποθήκης δεδομένων των χρηστών του αντικειμένου. Αυτό επιτρέπει σε κάποιον χρήστη να εξετάσει το περιεχόμενο της αποθήκης δεδομένων χρηστών αυτού του αντικειμένου.
- `getUserDatumCopyAction (key)`: ανακτά την εντολή αντιγραφής (copy action) για το datum με το καθορισμένο κλειδί ανάκτησης από την αποθήκη δεδομένων των χρηστών του αντικειμένου.

### Input and Output

Το YUNG αυτήν την περίοδο παρέχει μερική υποστήριξη για δύο διαφορετικούς τύπους αρχείων (formats): Αρχείο τύπου Pajek (Pajek format, reading and writing) και αρχείο τύπου GraphML (GraphML format, reading only). Οι σχετικές κλάσεις και οι μέθοδοι μπορούν να βρεθούν στο `io package`.

Να σημειωθεί ότι το πρόγραμμα JUNG δεν έχει κάποιο εγγενή τύπο αρχείου (native file format) για την απεικόνιση γραφημάτων.

### Αλγόριθμοι

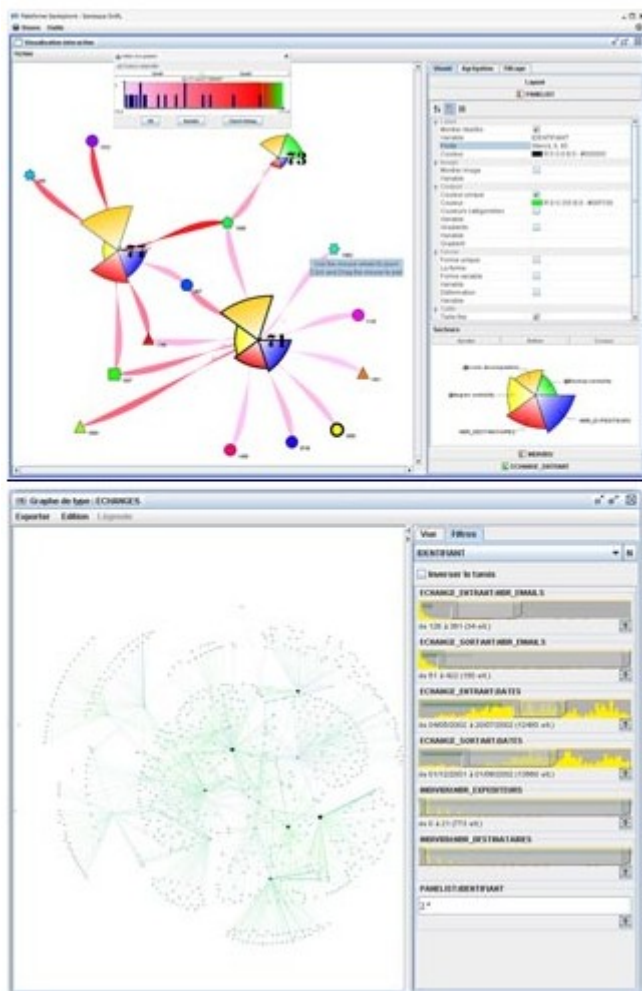
Το JUNG παρέχει μια σειρά από διαφορετικούς αλγόριθμους δικτύων και γραφημάτων. Γενικά, κάθε τύπος αλγορίθμου έχει το δικό του package. Οι αλγόριθμοι που χρησιμοποιούνται σε αυτό το tool μπορούν να κτηγοριοποιηθούν και αναλυτικά είναι οι εξής:

#### **1)Graph/Matrix Operations**

Αυτοί οι αλγόριθμοι «διαμένουν» στο κύριο πακέτο αλγορίθμων (στην κατηγορία `GraphMatrixOperations` και το interface `MatrixElementOperations`). Οι μήτρες είναι μια συνηθισμένη αναπαράσταση για δεδομένα δικτύων. Το `GraphMatrixOperations` αποτελείται από δύο κατηγορίες λειτουργιών:  
 -σε αυτές που χαρακτηρίζονται τυπικά ως λειτουργίες μητρών (matrix operations), αλλά μπορούν να εφαρμοστούν αποτελεσματικά εκμεταλλευόμενες τη δομή του γραφήματος και  
 -σε λειτουργίες που χρησιμοποιούν το πακέτο CERN Colt για τον αριθμητικό χειρισμό μητρών.

Ορισμένες από τις λειτουργίες του `GraphMatrixOperations` απαιτούν εφαρμογή του `MatrixElementOperations`, προκειμένου να καθοριστεί η συμπεριφορά της λειτουργίας ενός συγκεκριμένου τύπου στοιχείων (element type). Ουσιαστικά, οι μέθοδοι του `MatrixElementOperations` καθορίζουν το πώς να προχωρήσει το ισοδύναμο του διανυσματικού εσωτερικού προϊόντος (inner dot product). Παραδείγματος χάριν, οι εφαρμογές αυτού του interface θα καθορίσουν εάν τα στοιχεία του γραφήματος πρέπει να αντιμετωπιστούν ως αριθμητικά (numeric) ή τιμές Boolean (Boolean values).

### παράδειγμα



Το παράδειγμα αυτό δείχνει την εξερεύνηση ενός εσωτερικού δικτύου κυκλοφορίας ηλεκτρονικού ταχυδρομείου επιχείρησης. Το πάνω μέρος της εικόνας παρουσιάζει τη χρήση των κόμβων τομέα για να επιδείξει τις αριθμητικές (numeric) μεταβλητές (χαρακτηριστικά μέτρα SNA) στους κόμβους. Το χαμηλότερο μέρος παρουσιάζει πολυμεταβλητό δυναμικό φιλτράρισμα στη γραφική παράσταση.

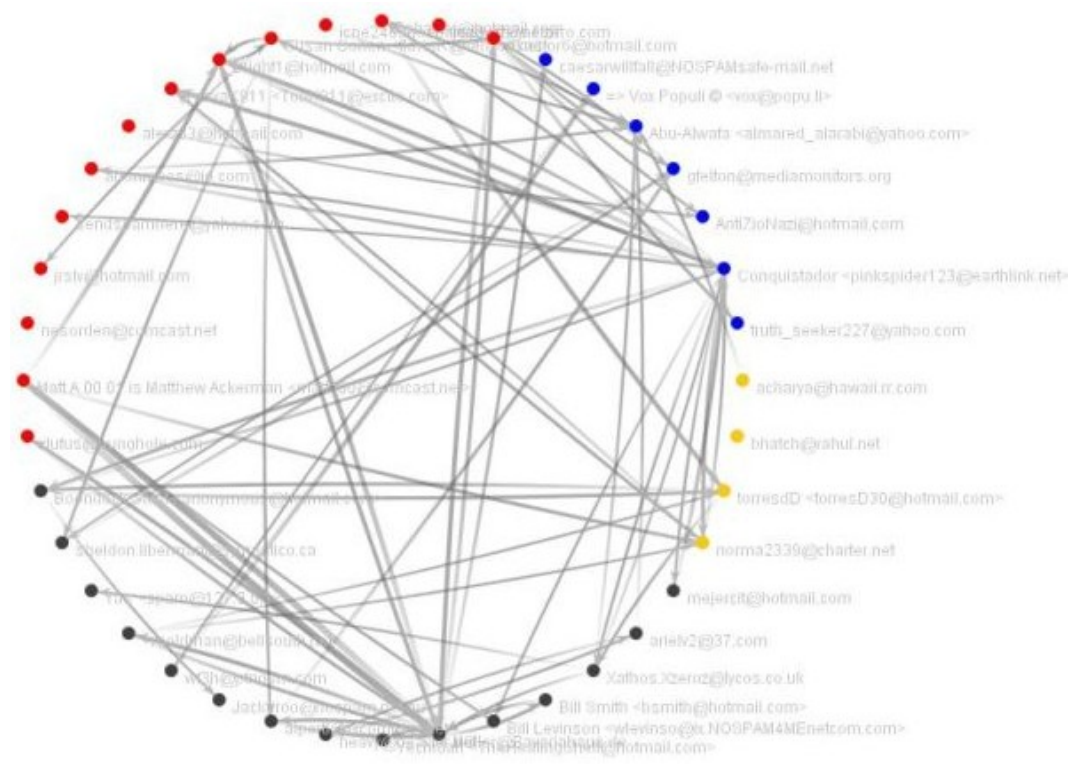
## 2)Ομαδοποίηση (Clustering)

Αυτοί οι αλγόριθμοι «διαμένουν» στο πακέτο `algorithms.cluster`. Ένα cluster είναι μια συλλογή αντικειμένων που είναι όλα παρόμοια το ένα με το άλλο με κάποιον τρόπο. Σε ένα δίκτυο, η ομοιότητα είναι συχνά βασισμένη σε τοπολογικές ιδιότητες όπως η συνδετικότητα (connectivity), αλλά μπορεί επίσης να βασιστεί σε ιδιότητες των vertices ή edges του δικτύου.

## 3)Τοπολογία, Διαδρομές και Ροές (Topology, Paths, and Flows)

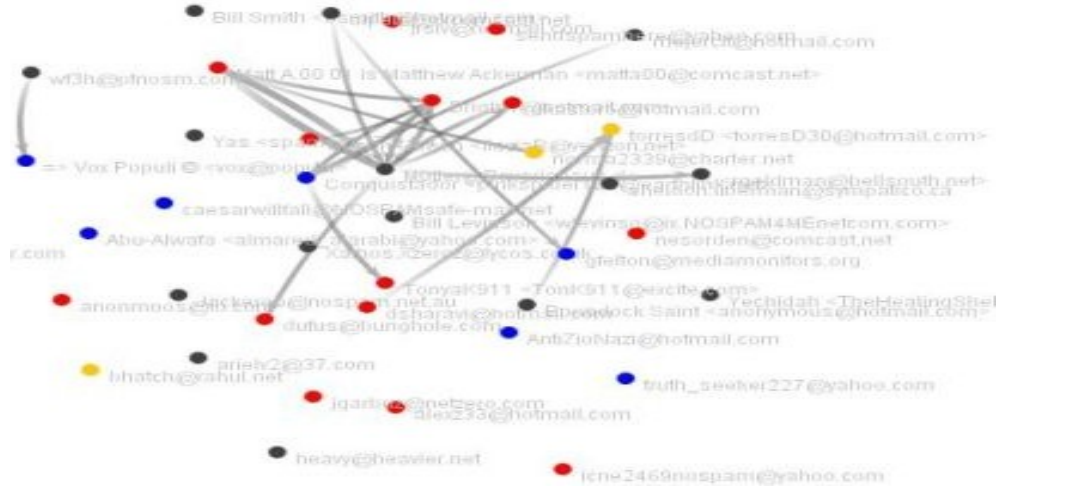
Αυτοί οι αλγόριθμοι εκτελούν λειτουργίες (και υπολογίζουν ιδιότητες) σε γραφήματα που αφορούν στην τοπολογία του γραφήματος, δηλαδή τις δομές και

υποδομές που διαμορφώνονται από τους τρόπους που τα vertices (σημεία) συνδέονται μεταξύ τους με τη βοήθεια των edges (ακμών).  
παράδειγμα κυκλικής τοπολογίας:



Το JUNG 1,60 απεικονίζει τις συνδέσεις μεταξύ των ζευγαριών των ανθρώπων σε μια online ομάδα πληροφόρησης (newsgroup). Οι συμμετέχοντες σε αυτήν κωδικοποιήθηκαν από την πολιτική τους προτίμηση (χρώμα). Οι ακμές κλίσης που παράγονται, χρωματίζουν τους κόμβους ανάλογα με την πολιτική προτίμηση. Παρατηρούμε τους από κοινού ομαδοποιημένους CircleLayout κόμβους της παρόμοιας πολιτικής προτίμησης. Σημειώνουμε παράλληλα ότι οι περισσότερες, εάν όχι όλες, οι συνδέσεις τρέχουν μεταξύ (παρά μέσα) των πολιτικών ευθυγραμμίσεων.

παράδειγμα: το ανωτέρω παράδειγμα με τη χρησιμοποίηση ενός άλλου αλγόριθμου σχεδιασμού-διάταξης (GraphLayout Algorithm). Συγκεκριμένα του αλγορίθμου Kamada-Kawai.



#### 4)Σημασία (Importance)

Αυτοί οι αλγόριθμοι «διαμένουν» στο πακέτο `algorithms.importance`. Οι αλγόριθμοι σημασίας δικτύων (`network importance algorithms`) μετρούν τη σημασία του κάθε vertex (ή edge) σύμφωνα με μια σειρά/σύνολο κριτηρίων που είναι συνήθως βασισμένη στον προσδιορισμό θέσης του vertex/edge, σχετικά με το υπόλοιπο του γραφήματος.

Μερικοί από τους ακόλουθους αλγορίθμους υποθέτουν ότι «ορίζονται» σε ένα δίκτυο Markov: ένα κατευθυνόμενο ζυγισμένο γράφημα στο οποίο τα σημεία (vertices) αντιπροσωπεύουν κράτη, οι άκρες (edges) αντιπροσωπεύει τις πιθανές κρατικές μεταβάσεις, και τα «βάρη» των ακρών (edge weights) αντιπροσωπεύουν τις πιθανότητες μετάβασης. Η στάσιμη πιθανότητα (stationary probability) για ένα vertex  $v$  σε ένα τέτοιο δίκτυο είναι η περιοριστική πιθανότητα ότι, θεωρώντας δεδομένη μια αυθαίρετη αρχική κατάσταση και έναν μεγάλο αριθμό μεταβάσεων, η επικρατούσα κατάσταση θα είναι αυτή του  $v$ .

#### Οπτικοποίηση – Visualization

Το πακέτο οπτικοποίησης του JUNG παρέχει τους μηχανισμούς για το σχεδιασμό και τη φωτοσκίαση (rendering) των γραφημάτων. Οι τρέχουσες εφαρμογές φωτοσκίασης (renderer) χρησιμοποιούν το Java Swing API προκειμένου να απεικονίσουν τα γραφήματα, αλλά αυτές ενδέχεται να εφαρμοστούν χρησιμοποιώντας άλλα κουτιά εργαλείων.

Γενικά, μια οπτικοποίηση ολοκληρώνεται με:

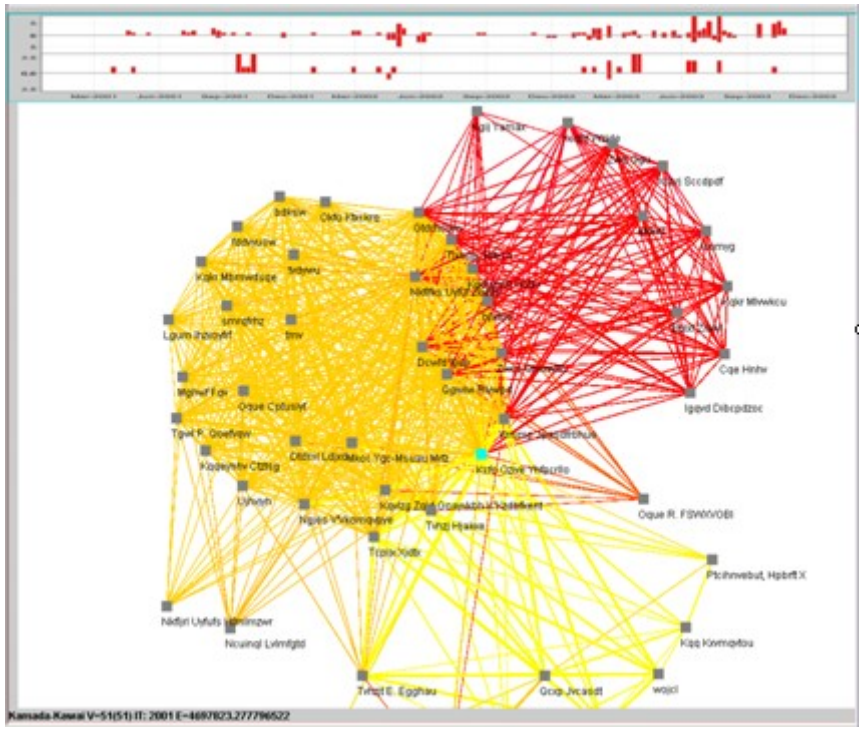
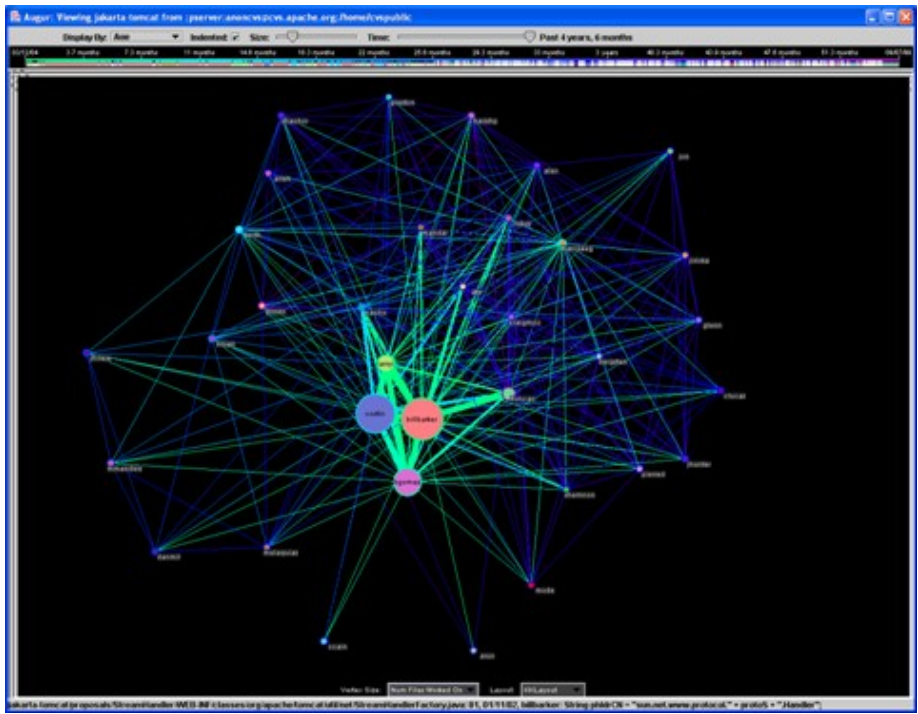
- Ένα σχεδιάγραμμα (layout), το οποίο παίρνει ένα γράφημα και καθορίζει τη θέση στην οποία θα σχεδιαστεί κάθε ένα από τα σημεία του.
- Μια συνιστώσα ταλάντευσης (swing component), στο οποίο φωτοσκιάζονται τα δεδομένα. (Οι τρέχουσες εφαρμογές χρησιμοποιούν ένα `VisualizationViewer`, το οποίο είναι μια επέκταση της κατηγορίας `Swing JPanel`.)
- Ένα `Renderer`, που παίρνει τα στοιχεία που παρέχονται από το σχεδιάγραμμα και χρωματίζει τα σημεία και τις ακμές στην παρεχόμενη συνιστώσα.

Κατά συνέπεια, με την επιλογή ενός εκ των τριών, είναι δυνατό να συντονιστεί το σχέδιο (coordinate drawing). Η εφαρμογή προεπιλογής «διαπερνά» το σχεδιάγραμμα, «ρωτώντας» το για τις θέσεις των vertices και στη συνέχεια τους χρωματίζει χωριστά με το `Renderer` μέσα στη συνιστώσα ταλάντωσης. Επιπλέον, η υποδομή `GraphDraw` απλοποιεί πολλούς από αυτούς τους μετασχηματισμούς με το να πακετάρει το `VisualizationViewer`, το `Renderer`, και το σχεδιάγραμμα/layout από κοινού. Οι χρήστες μπορούν έπειτα να προσαρμόσουν αυτόν τον viewer ανάλογα με την περίπτωση.

Η τρέχουσα εφαρμογή υποστηρίζει μόνο δυσδιάστατους αλγορίθμους σχεδιαγράμματος (δυναδικών) γραφημάτων. Δεν υποστηρίζει την οπτικοποίηση άλλων τύπων γραφημάτων (όπως των hypergraphs).







## V. ΣΥΜΠΕΡΑΣΜΑΤΑ

Σύμφωνα με τα όσα έχουν αναλυθεί παραπάνω μπορούμε να προβούμε στα εξής συμπεράσματα:

### ΜΕΙΟΝΕΚΤΗΜΑΤΑ

- Μετά την έρευνα και αναζήτηση στο διαδίκτυο είναι πλέον σαφές ότι η συντριπτική πλειοψηφία των εργαλείων σχεδιασμού και προγραμματισμού δικτύων δεν είναι «ελεύθερα», αλλά αποτελούν προϊόν εμπορικής εκμετάλλευσης επιχειρήσεων.
- Τα περισσότερα free tools σχετίζονται με network monitoring και measurement, τα οποία έχουν να κάνουν με ποσοτικές μετρήσεις διαφόρων στοιχείων (όπως π.χ. μέτρηση ροής πακέτων, μέτρηση ταχύτητας καναλιών, βαθμός αποδοτικότητας server, utilization rate και πολλά άλλα) τα οποία όμως δεν ενέχουν στοιχεία σχεδιασμού δικτύων με τον οποίο ασχολείται η παρούσα εργασία.
- Τα δύο εργαλεία που αναλύθηκαν δεν μπορούμε να πούμε ότι είναι αποκλειστικά για network planning&design αλλά συνδυάζουν και στοιχεία της απεικόνισης και οπτικοποίησης δικτύων (network visualization). Μάλιστα πολλά από τα tools εμπεριέχουν και στοιχεία από προσομοίωση δικτύων (network simulation). Συνεπώς ένα tool με εξειδίκευση στον τομέα design μάλλον θα αποτύγχανε για χρήστες υψηλών απαιτήσεων (αναλυτές, τεχνικοί και μηχανικοί δικτύων)
- Απαιτείται γνώση προγραμματισμού σε JAVA προκειμένου να χρησιμοποιήσει κανείς το PlotPath και το JUNG. Δεν είναι δηλαδή εύκολα στη χρήση και δεν έχουν φιλικό user interface

### ΠΛΕΟΝΕΚΤΗΜΑΤΑ

- Τα δύο αυτά εργαλεία είναι «ανοικτού κώδικα». Άρα μπορεί ο κώδικάς τους να αναπτυχθεί και να βελτιωθεί. Επίσης ο κάθε ενδιαφερόμενος μπορεί να κάνει τροποποιήσεις ανάλογα με τις ανάγκες ή τα δεδομένα του.

- Η χρήση της Java και των Java Applet ισοδυναμούν με σύνδεση των εργαλείων αυτών με τον παγκόσμιο ιστό (η Java το υποστηρίζει). Επομένως το output των εργαλείων μπορεί να είναι και online μέσω ενός Applet και επομένως και πιο διαδραστικό με τους χρήστες. Επίσης ένα άλλο πλεονέκτημα των δύο αυτών tool που οφείλεται πάλι στην Java είναι το γεγονός ότι τα οπτικά αποτελέσματα είναι ποιοτικότερα και σε μερικές περιπτώσεις χρησιμοποιούν και 3D animation.
- Πέρα από το ότι χρησιμοποιούν τη Java, τα γενικά πλεονεκτήματα των δύο εργαλείων είναι το ότι
  - Παρέχουν μια κοινή γλώσσα για τις γραφικές παραστάσεις
  - Συμπληρώνουν παρά αντικαθιστούν άλλες συσκευασίες γραφικών παραστάσεων/εργαλεία δικτύων
  - Αποτελούν ισχυρά εργαλεία με πλούσια χαρακτηριστικά δομής για σχεδιασμό δικτύων με γραφήματα
  - Είναι κατάλληλα για την οικοδόμηση των εργαλείων/εφαρμογών που αφορούν την εξερεύνηση δικτύων και την ανάσυρση δεδομένων

## VI. ΒΙΒΛΙΟΓΡΑΦΙΑ ΚΑΙ ΔΙΚΤΥΑΚΕΣ ΠΗΓΕΣ

### Βιβλιογραφία

- [1] Penttinen A., *Chapter 10 – Network Planning and Dimensioning, Lecture Notes: S-38.145 - Introduction to Teletraffic Theory*, Helsinki University of Technology, Fall 1999.
- [2] Farr R.E., *Telecommunications Traffic, Tariffs and Costs – An Introduction For Managers*, Peter Peregrinus Ltd, 1988.
- [3] HUMMEL. *Network Planning and Design Guide*, CISCO PRESS, 2002
- [4] Mishra, A.R., *Fundamentals of Network Planning and Optimisation*, John Wiley and Sons Ltd, 2004
- [5] Kenyon, Tony, *High Performance Data Network Design*, Elsevier Science & Technology, 2000
- [6] Gilbert,Held / *Network Design*, Taylor & Francis Ltd, 2000
- [7] Διακονικολάου Γιώργος, Αγιακάτσικα Αθανασία, Μπούρας Ηλίας / *Επιχειρησιακή διαδικτύωση, Κλειδάριθμος*, 2004

### Διαδικτυακές Πηγές

[http://en.wikipedia.org/wiki/Network\\_planning\\_and\\_design](http://en.wikipedia.org/wiki/Network_planning_and_design)  
<http://jung.sourceforge.net/>  
<http://jung.sourceforge.net/pmwiki/index.php/Main/ImageGallery>  
<http://jung.sourceforge.net/doc/manual.html>  
<http://www.caida.org/tools/visualization/plotpaths/>  
[http://www.caida.org/tools/visualization/plotpaths/plotpaths\\_shots.xml](http://www.caida.org/tools/visualization/plotpaths/plotpaths_shots.xml)  
<http://www.opnet.com/products/itguru/home.html>  
<http://webopedia.internet.com/Networks/>  
<http://sun.com>  
<http://www.manageability.org/blog/stuff/open-source-graph-network-visualization-in-java/view>  
<http://www.meriton.com/products/9500.php>  
<http://java.sun.com/products/java-media/3D/index.jsp>  
<http://manageengine.adventnet.com/>  
<http://www.netreference.com/>  
<http://connect.shift-think.com/>  
<http://netscan.research.microsoft.com/>

