6. Graphs and Networks visualizing relations

Lecture „Informationsvisualisierung”

Prof. Dr. Andreas Butz, WS 2012/13

Concept and slides: Thorsten Büring,

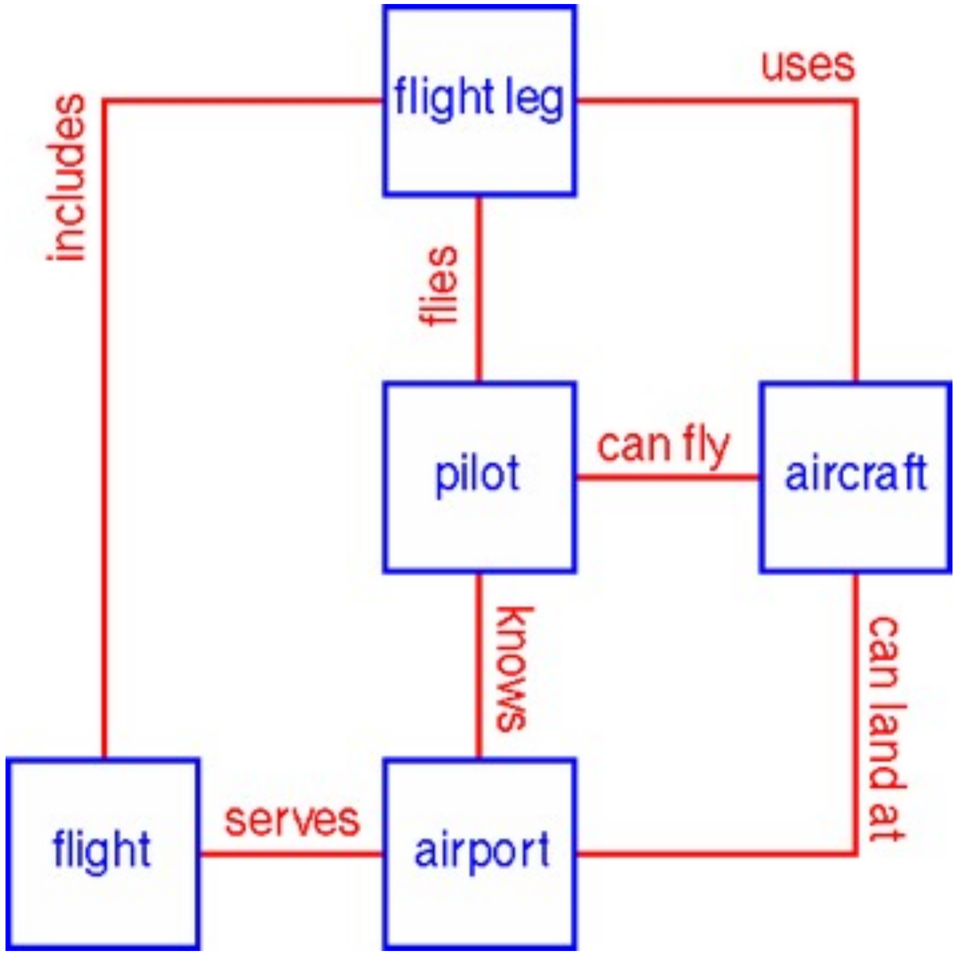
3rd, revised edition

Outline

- Graph overview
  - Terminology
  - Networks and trees
  - Data structures
  - Graph drawing
- Comparison of graph layouts
- Graph visualization examples
  - Social networks
  - Copurchase network
  - Music network
  - Transportation network
- Case study: Telephone network visualizations
- Comparing node-link and matrix representations
- Interaction and animation

Graph Overview

- Graph definition: an abstract structure that is used to model information
- Can represent any information that can be modeled as objects and connections between those objects
- Objects represented by vertices
- Relations between objects represented by edges
- Commonly visualized as node-link diagrams
- Example domains
  - World Wide Web
  - Telephone networks
  - Financial transactions
  - Call graph in software engineering
  - CVS repositories
  - Social networks
  - Transportation networks
  - Co-citations…



Automatically generated airline database schema, Tamassia et al. 1988

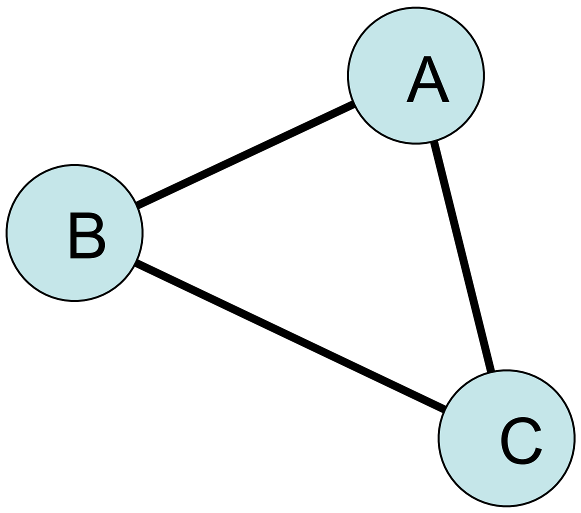
- Graphs in InfoVis shall facilitate the understanding of complex patterns

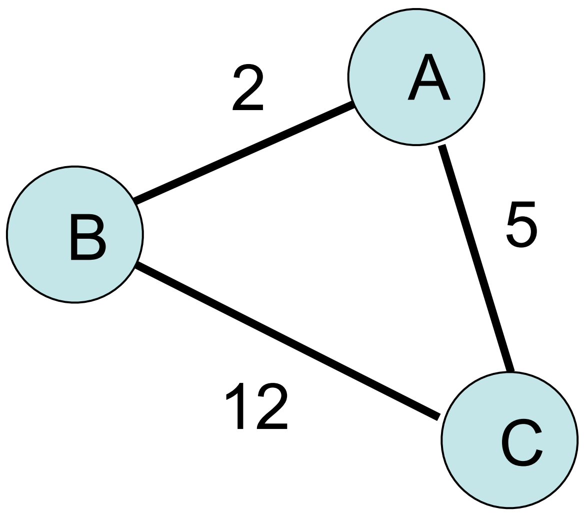
Challenges in Graph Drawing

- Graph Visualization (layout and positioning)
  - How to present a graph to convey the most information and to make it easy to read and interpret it
- Scale
  - Performance of layout algorithms
  - Limited real estate of display area
- Navigation and Interaction
  - How to enable the user to move around the graph and inspect portions of the graph in detail

Graphs Terminology

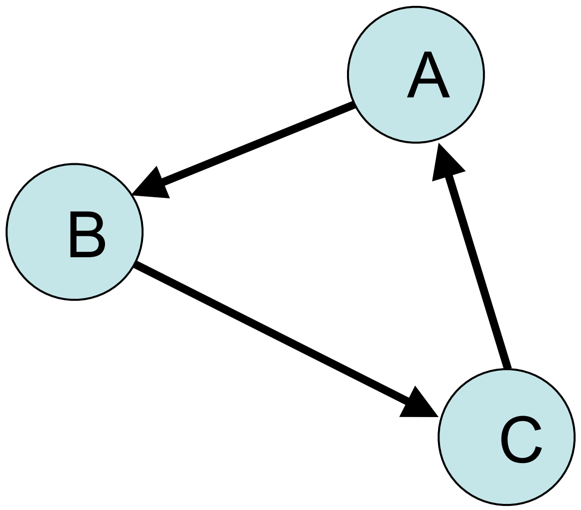
- Graph consists of
  - Nonempty set of vertices (points)
  - Set of edges that link together the vertices
- Undirected graph
- Directed graph (usually indicated by arrows)
- Mixed graph – contains both directed and undirected graphs
- Unweighted vs. weighted (nominal, ordinal quantitative) edges
- Degree of a vertex: the number of edges connected to it
- In-degree and out-degree for directed graphs
- Adjacency
  - Two edges sharing a common vertex
  - Two vertices sharing a common edge





Graphs Terminology

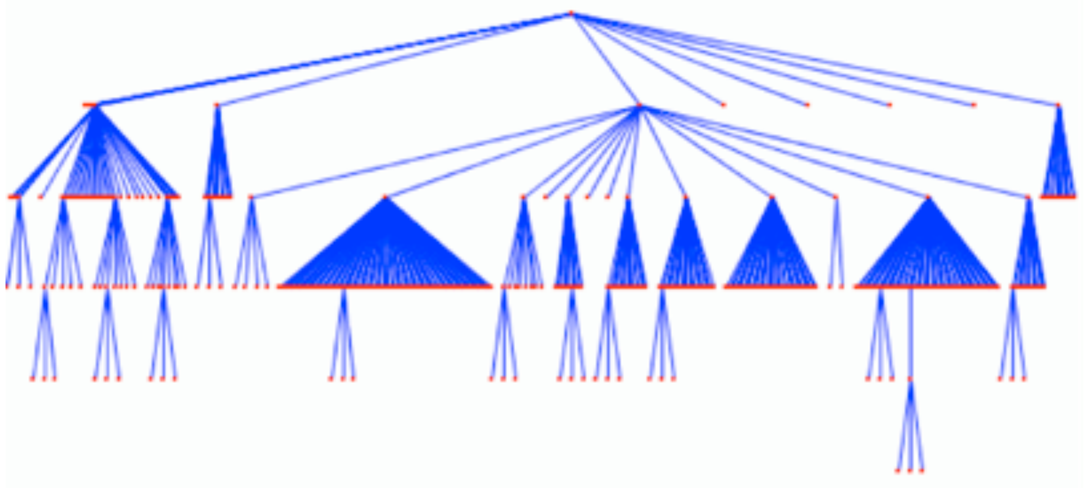
- Path: a traversal of consecutive vertices along a sequence of edges
- Length of the path: number of edges that are traversed along the path
- Simple path: no repeated vertices within the path
- Cycle: a path in which the initial vertex of the path is also the terminal vertex of the path
- Acyclic: a simple directed graph not containing any cycles



Directed Graph Cycle

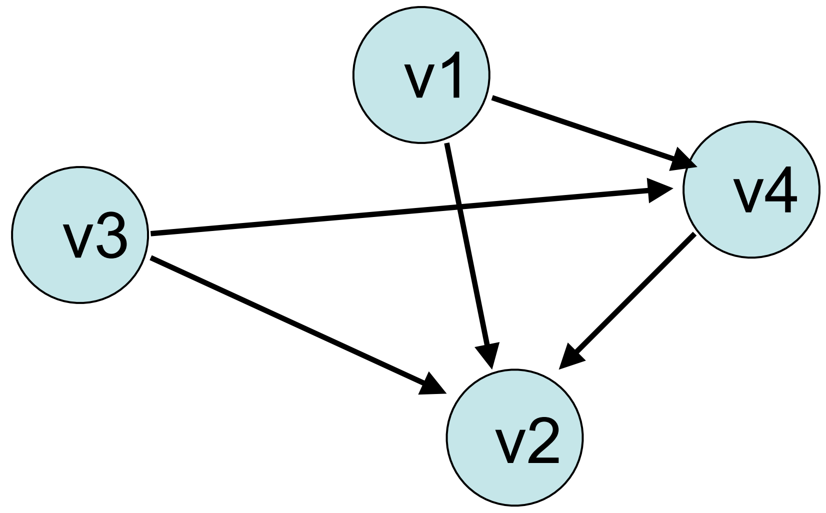
Special Types of Graphs

- Network
  - Directed Graph
  - Usually weighted edges
  - No topological restrictions
  - Examples: social, economic, transportation networks
- Tree
  - No cycles
  - Usually directed edges
  - Usually special designated root vertex
  - Example: organizational chart
  - Will be topic of next lecture!



Data Structures for Graphs

- Storing and processing a graph on a computer
- Adjacency List - usually used for graphs with small numbers of edges
- Adjacency Matrix - allows powerful matrix operations but is often more memory demanding
  - Row: edges leaving the vertex
  - Column: edges entering the vertex
- Example for directed graph



v1 -> v2 -> v4
v2 ->
v3 -> v2 -> v4
v4 -> v2

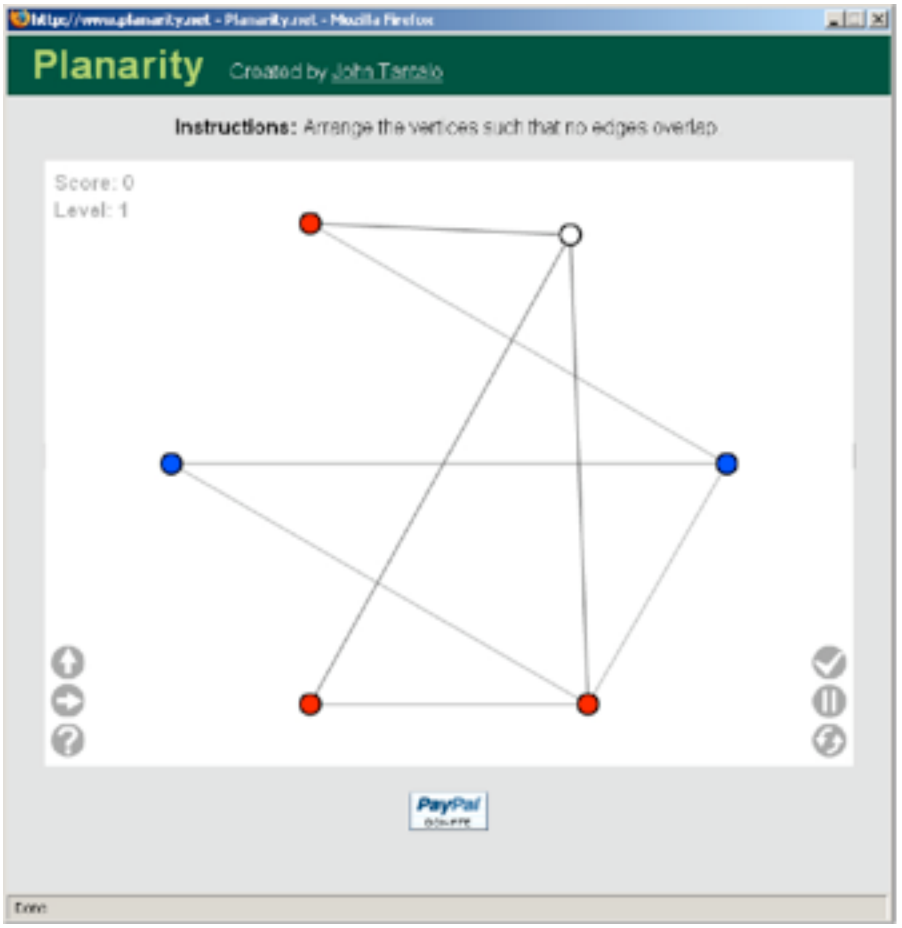
	v1	v2	v3	v4
v1	0	1	0	1
v2	0	0	0	0
v3	0	1	0	1
v4	0	1	0	0

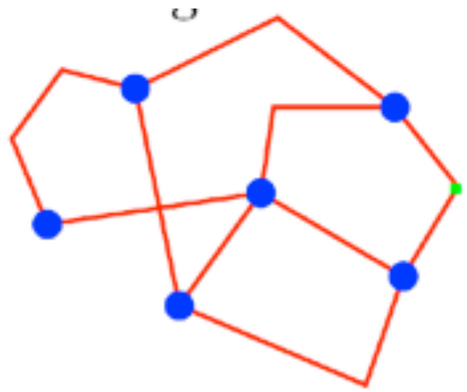
Graph Drawing

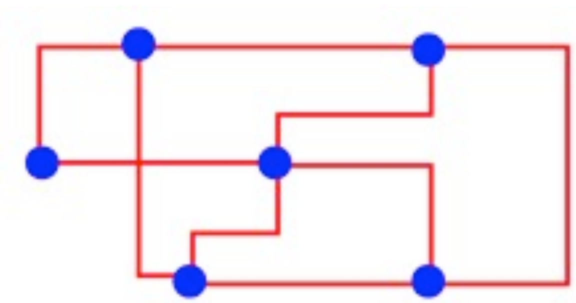
- Many ways to draw a graph
- Vertices are usually represented by circles
- Edges are usually represented by open curves between vertices
- Node-link diagram
- Potential encoding attributes
  - Color
  - Size
  - Form / Shape
- Labeling is often difficult due to clutter

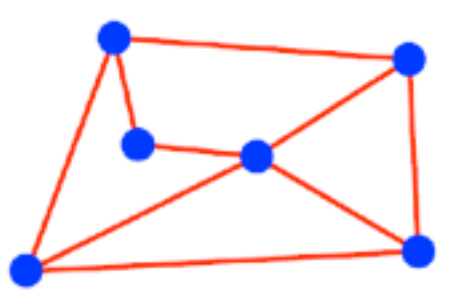
Graph Drawing

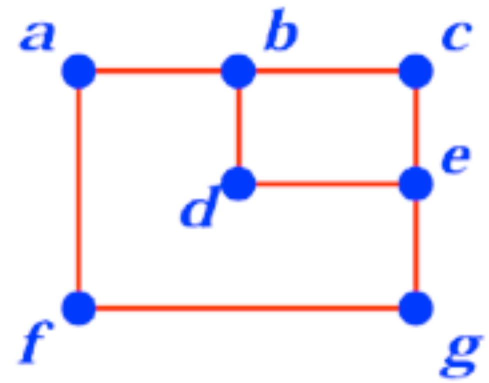
- Layout algorithms can be categorized by the type of layout they generate
- Planar: edges do not intersect
- Straight, polyline (edge with bends) or curved lines
- Orthogonal: polyline drawing that maps each edge into a chain of horizontal and vertical segments







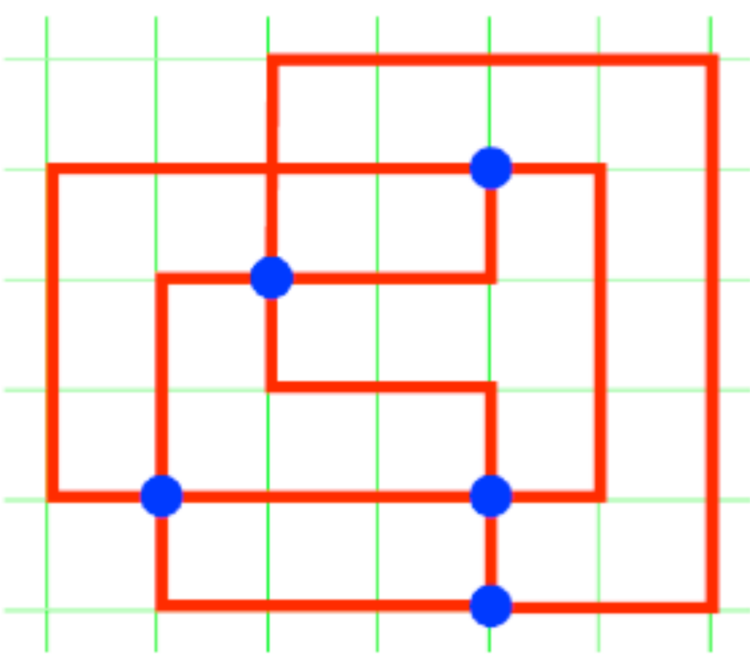


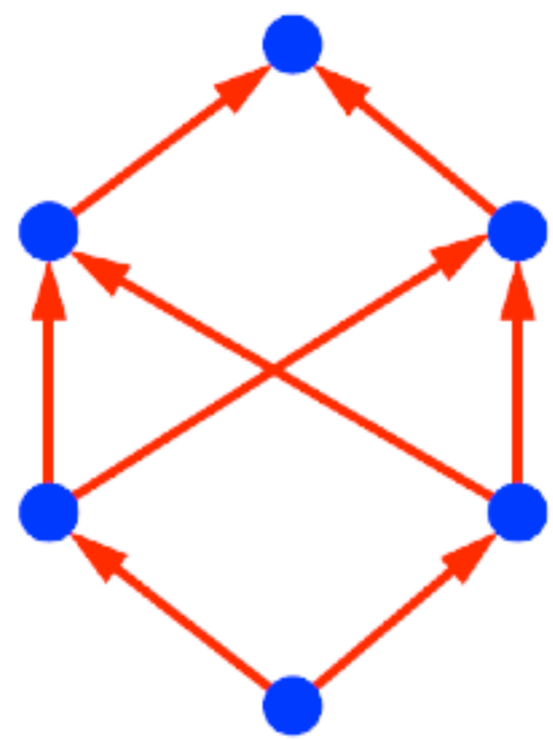


Images taken from Cruz & Tamassia

Graph Drawing

- Grid-based: vertices (and bends of the edges) have integer coordinates – implies minimum distance between vertices and nonincident edges
- Upward / downward drawing for directed acyclic graphs: make edges flow in the same direction, e.g. for visualizing hierarchies





Images taken from Cruz & Tamassia

Layout Aesthetics

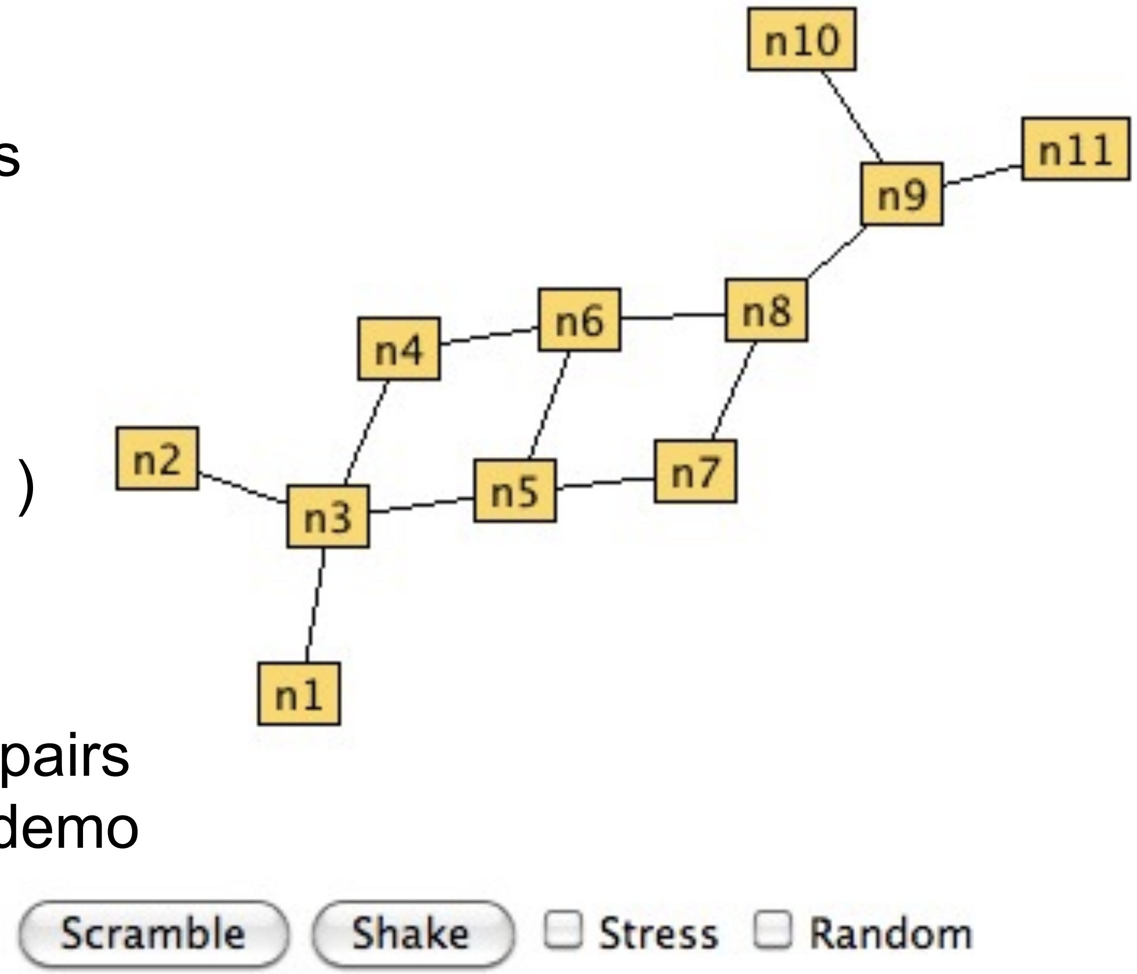
- Minimize crossing – keep the number of times that lines cross to a minimum (hardly applicable in interactive systems)
- Minimize area – keep the area that the graph takes up to a minimum by producing a compact graph
- Minimize the sum of the edge lengths
- Obtain a uniform edge length – try to keep each of the edges at the same lengths
- Minimize bends – keep the number of bends to a minimum
- Display symmetry of graph structure
- Maximize minimum angles between edges
- …

Empirical Results

- Purchase 1997
  - Compare task performance on five pairs of graphs
  - Graph pairs differed according to numbers of edge bends, edge crosses, maximizing the minimum angle, orthogonality and symmetry
  - Result: Reducing crossings is by far most important
- Ware et al. 2002
  - Experimental task: finding the shortest path in spring layout graphs
  - Results indicate the following prioritization of metrics
    - Geometric length of the path (implicit property of a graph)
    - Continuity (keeping multi-edge paths as straight as possible)
    - Number of edge-crossings

Spring Embedder

- Force-directed model for graph layout
- Eades 1984
- Intuitive approach: apply physical model of forces
  - Every vertex is considered a steel ring
  - Every edge a spring
- Resulting layout represents a configuration of minimum energy (force exerted on each ring is 0)
- Can produce well-balanced, symmetrical graphs
- Problem: time consuming – quality of the graph depends on the number of full iterations (visit all pairs of vertices to calculate the effect of the forces) - demo

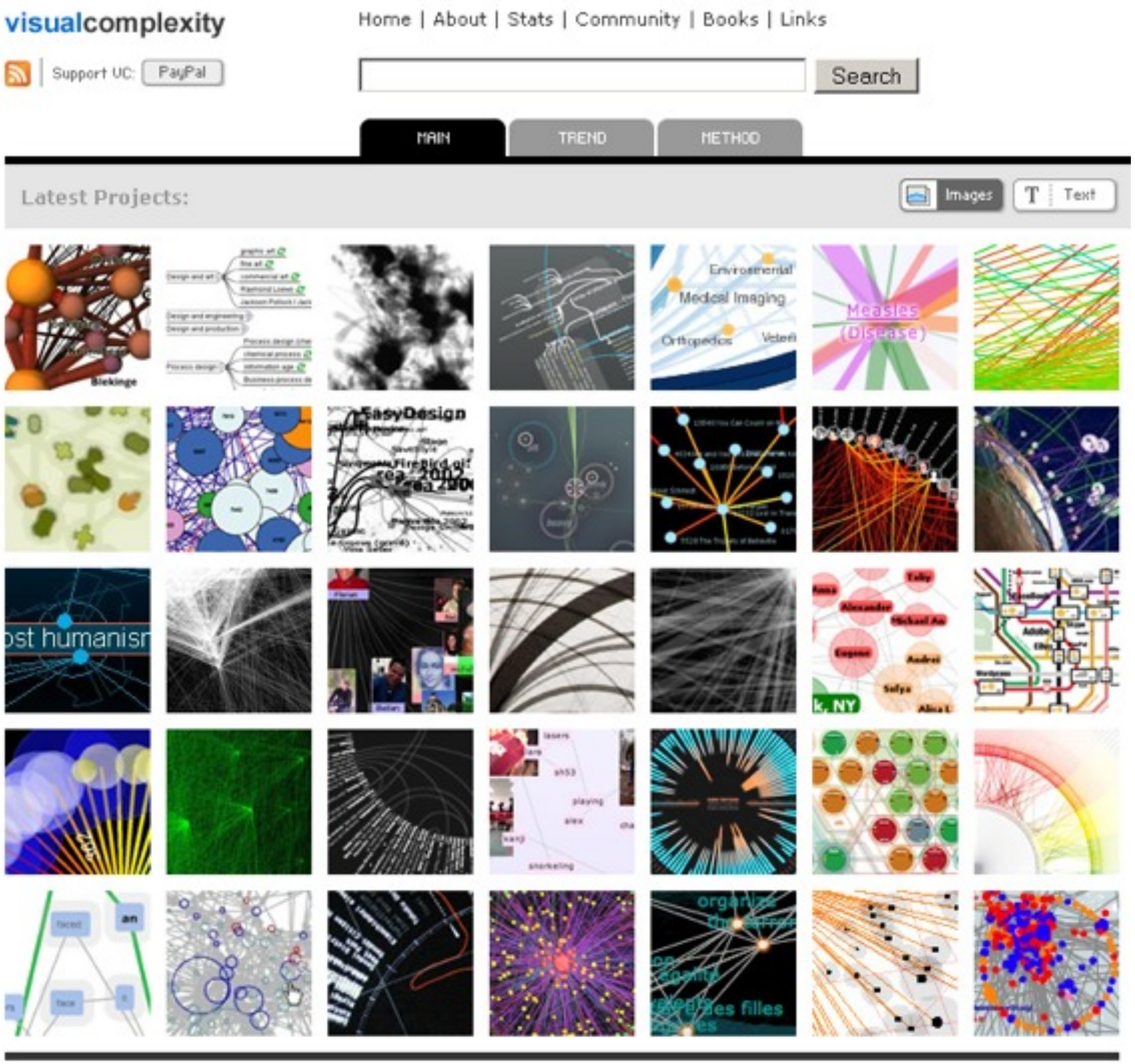


- Overview of graph drawing algorithms: Pajntar 2006 (http://kt.ijs.si/dunja/SiKDD2006/Papers/Pajntar.pdf)
- Open Graph drawing Framework OGDF: http://www.ogdf.net/
- Graph drawing tutorial: http://www.cs.brown.edu/~rt/papers/gd-tutorial/gd-constraints.pdf

Spring embedder Java applet + source code
http://www.inf.uni-konstanz.de/algo/lehre/ss04/gd/demo.html

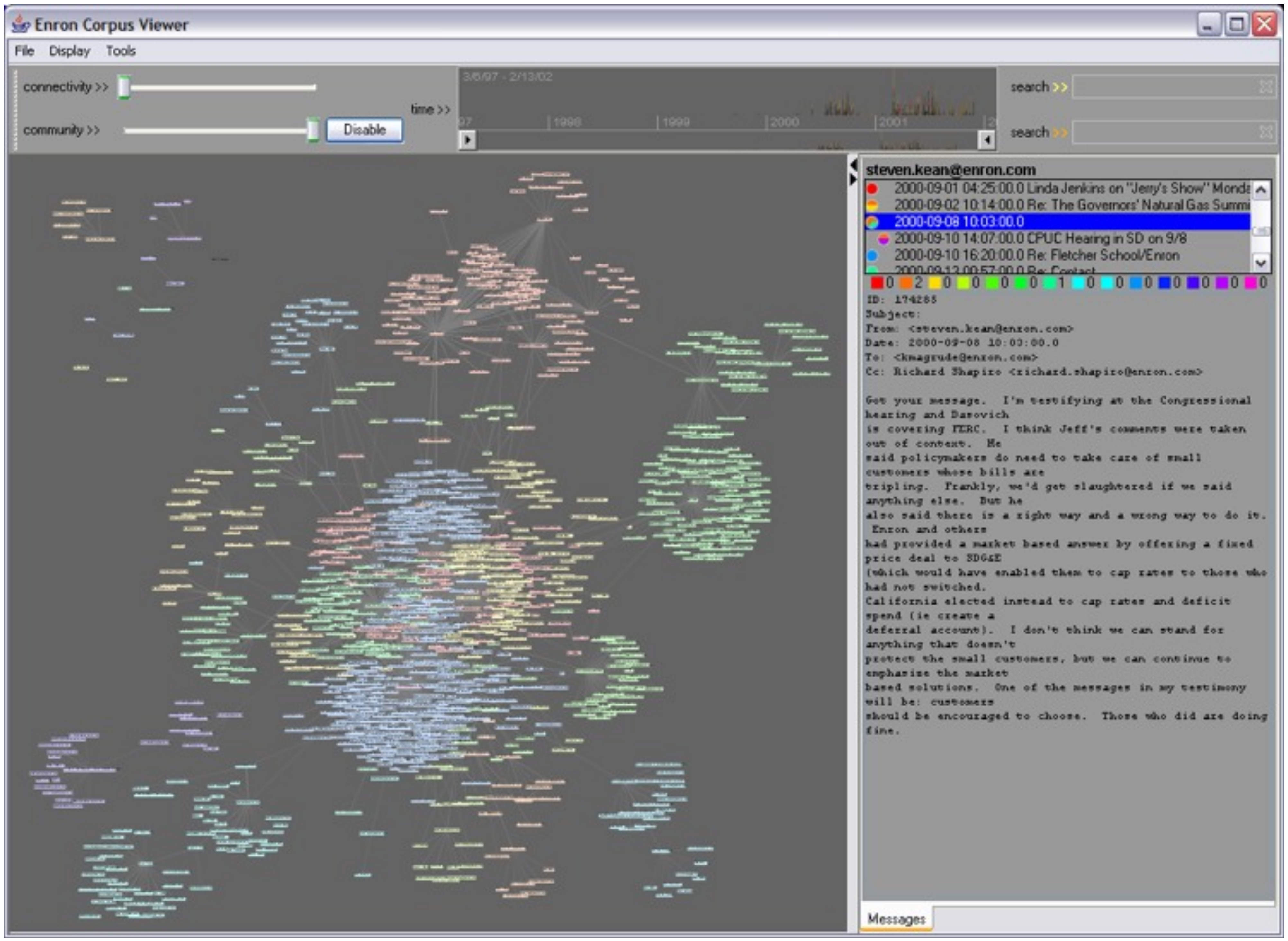
Various Examples of Graph Drawings

- http://www.visualcomplexity.com/



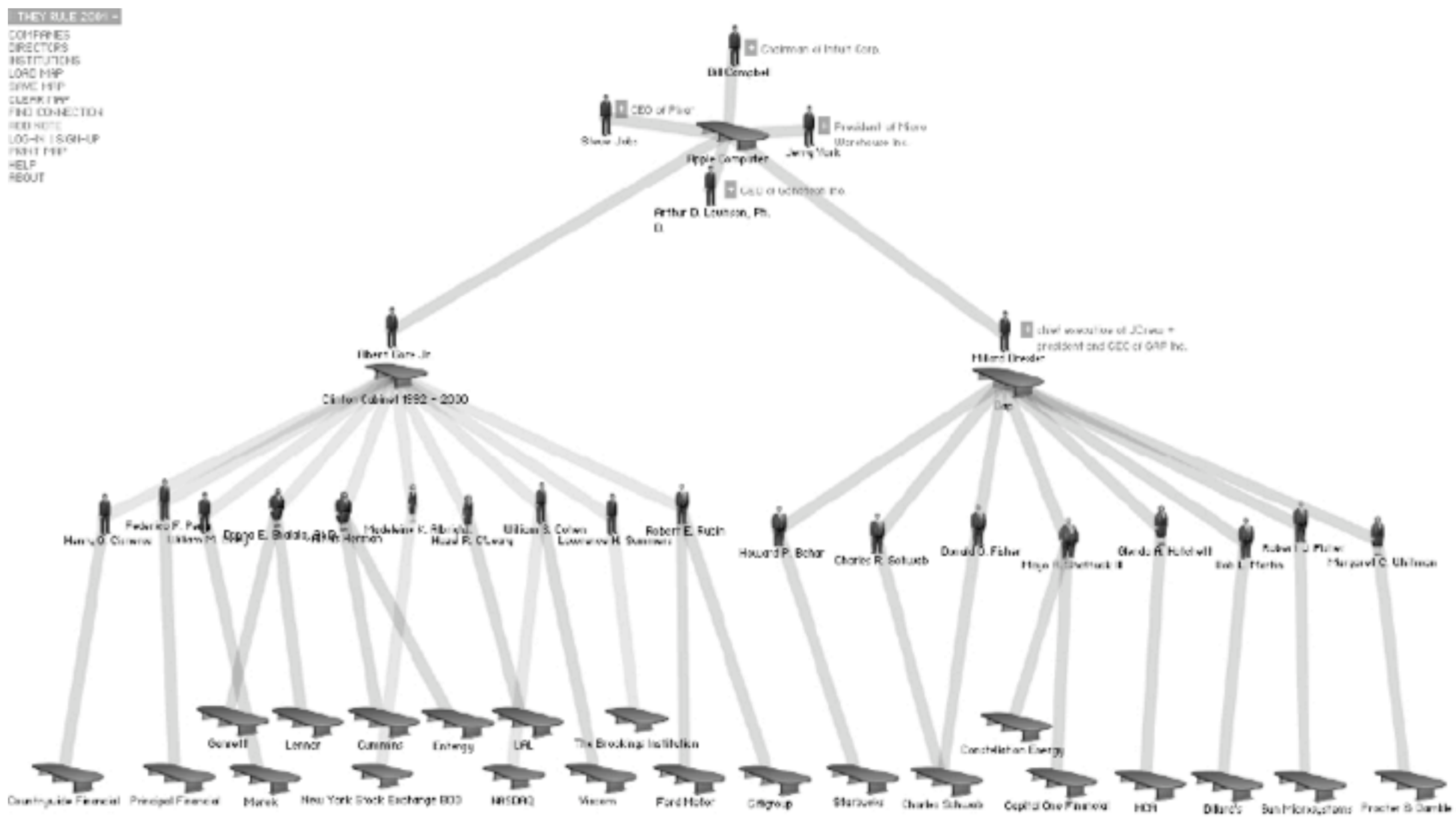
Social Network

- Exploring Enron: http://jheer.org/enron/



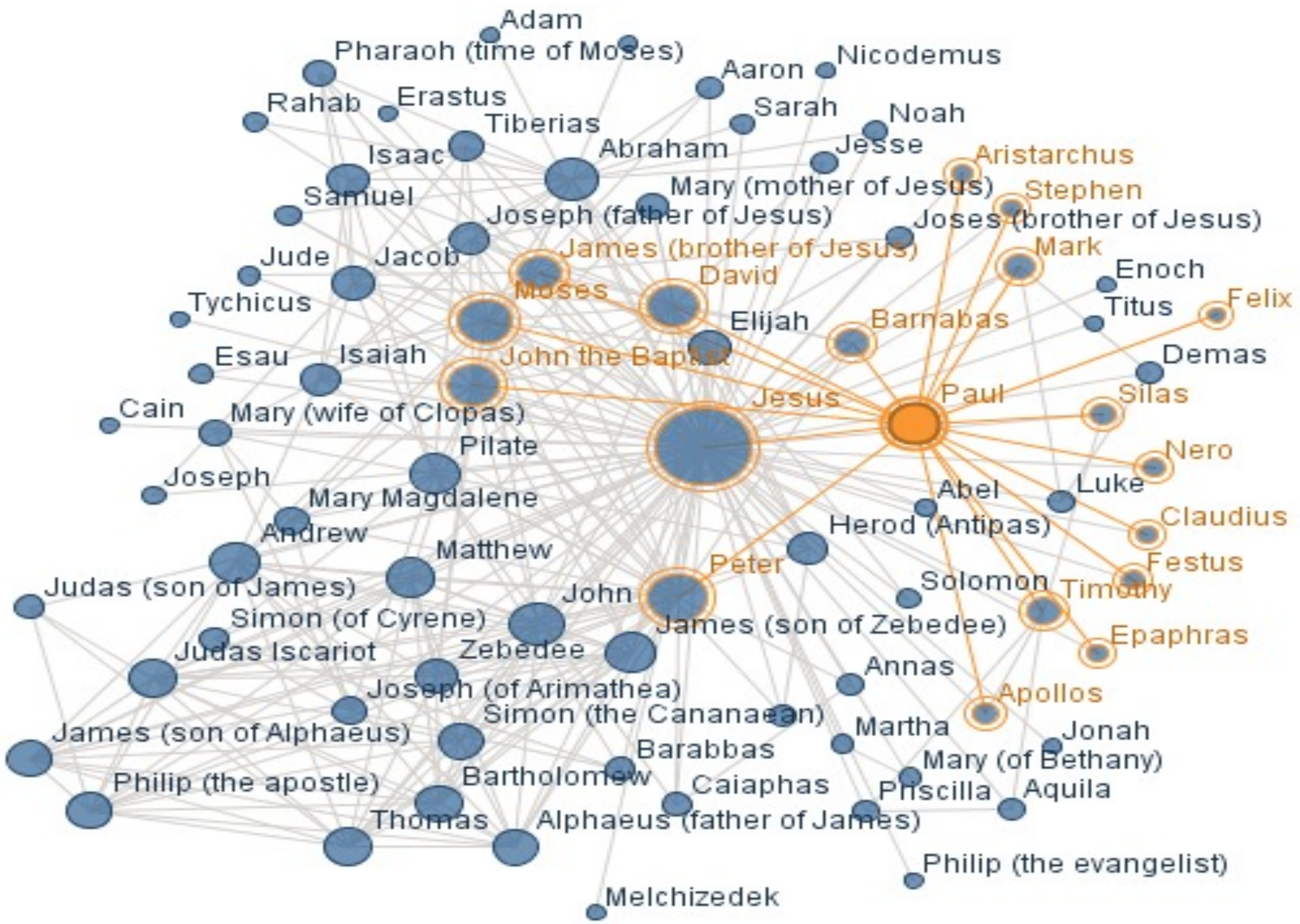
Social Network

- They rule: http://www.theyrule.net/2004/tr2.php



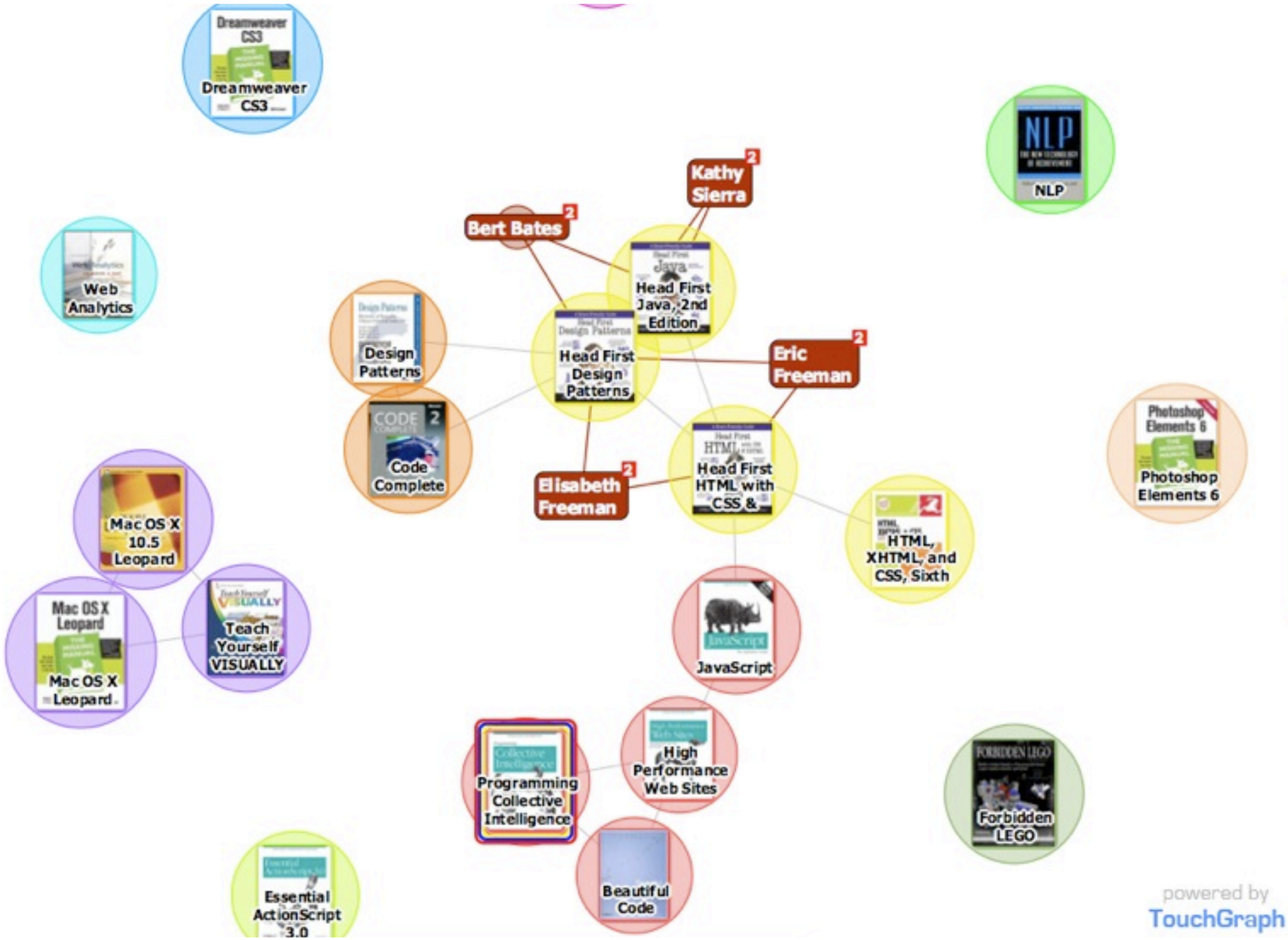
Social Network?

- Co-occurrences of names in the new testament: http://www-958.ibm.com/software/data/cognos/manyeyes/visualizations/89ade5ae1055f49801105a9fb0ac03fd



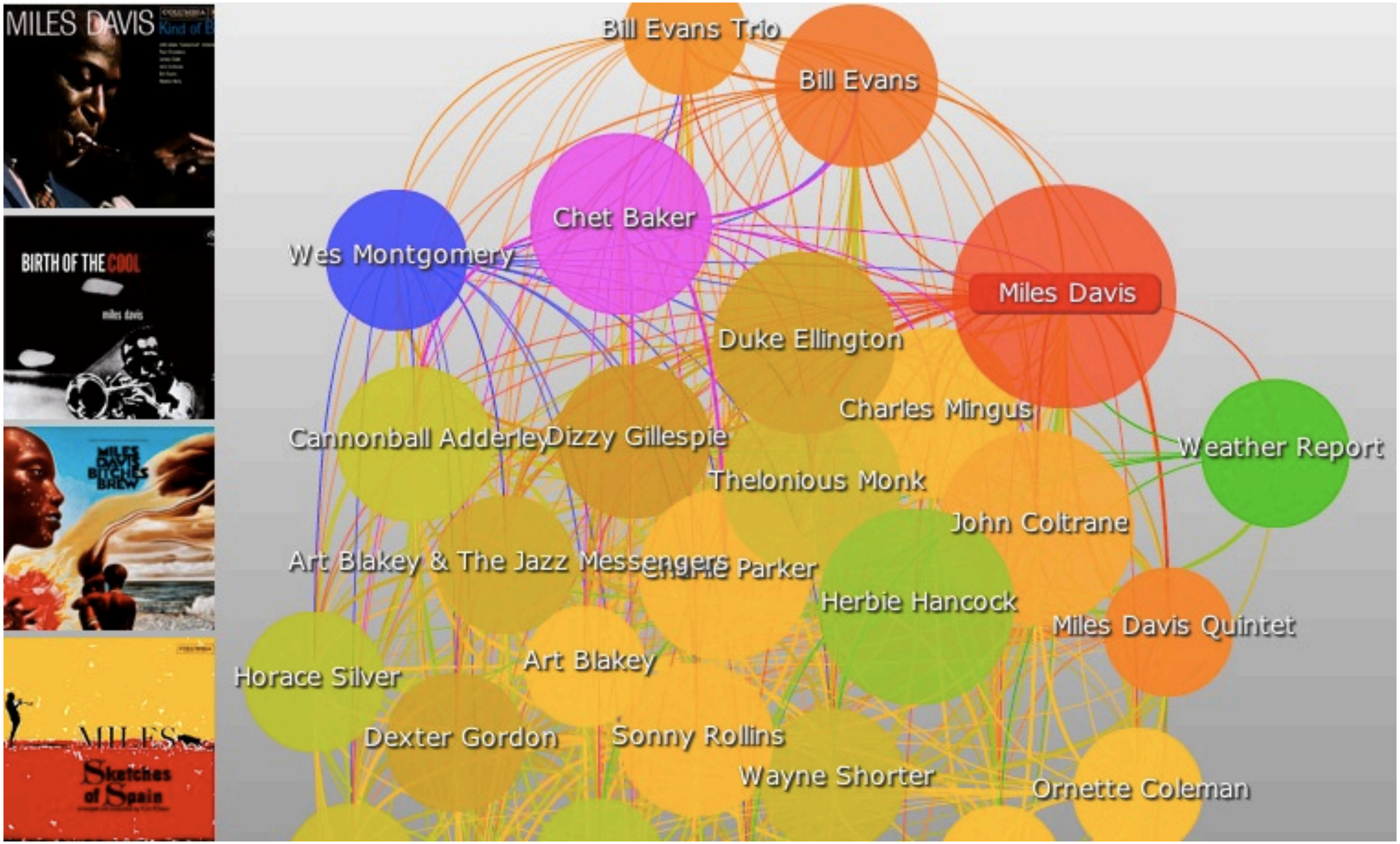
Copurchase Network

- Touch graph: http://www.touchgraph.com/TGAmazonBrowser.html



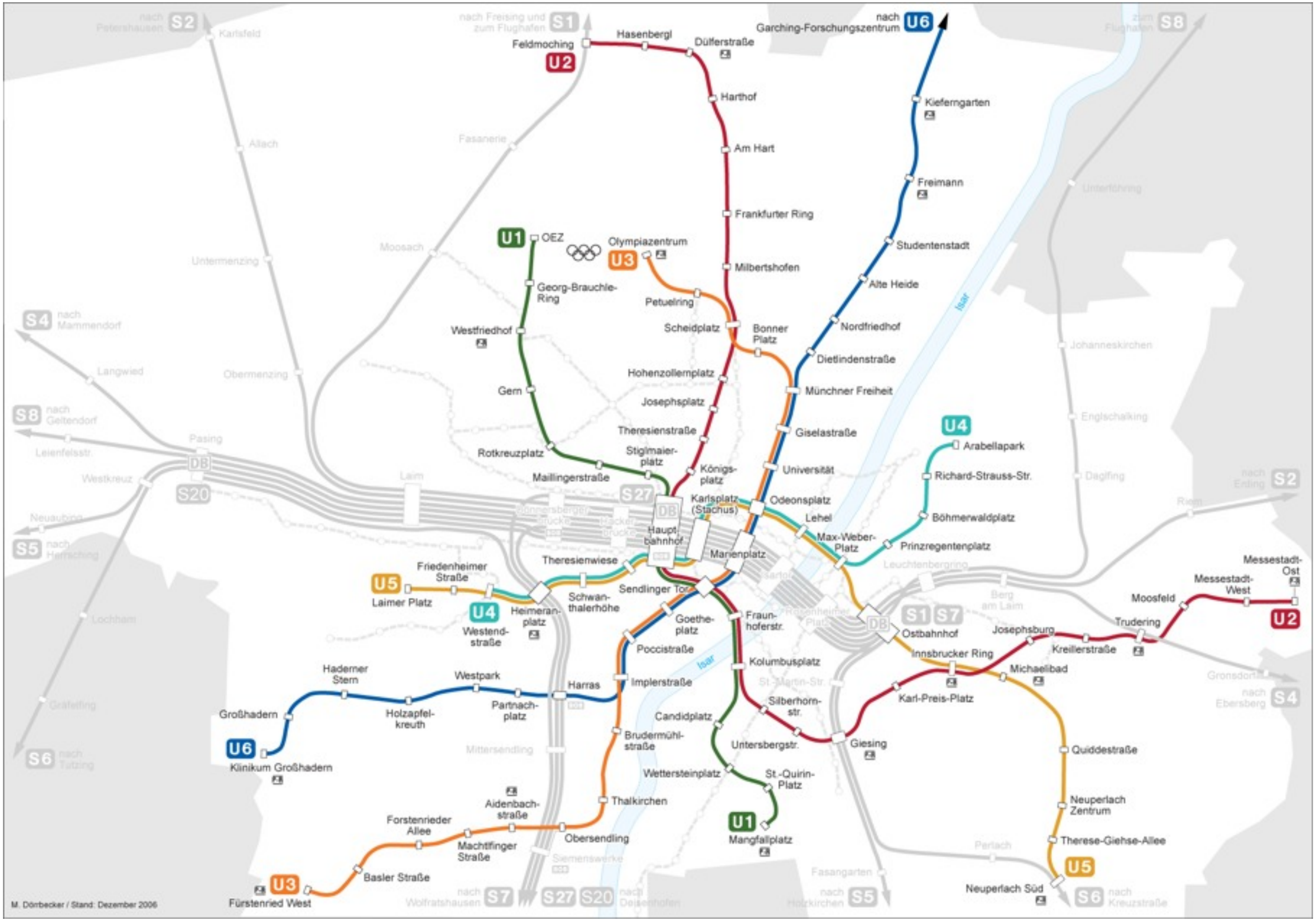
Music + Movie Network

- Liveplasma: http://www.liveplasma.com/
- Mapping and data source unclear

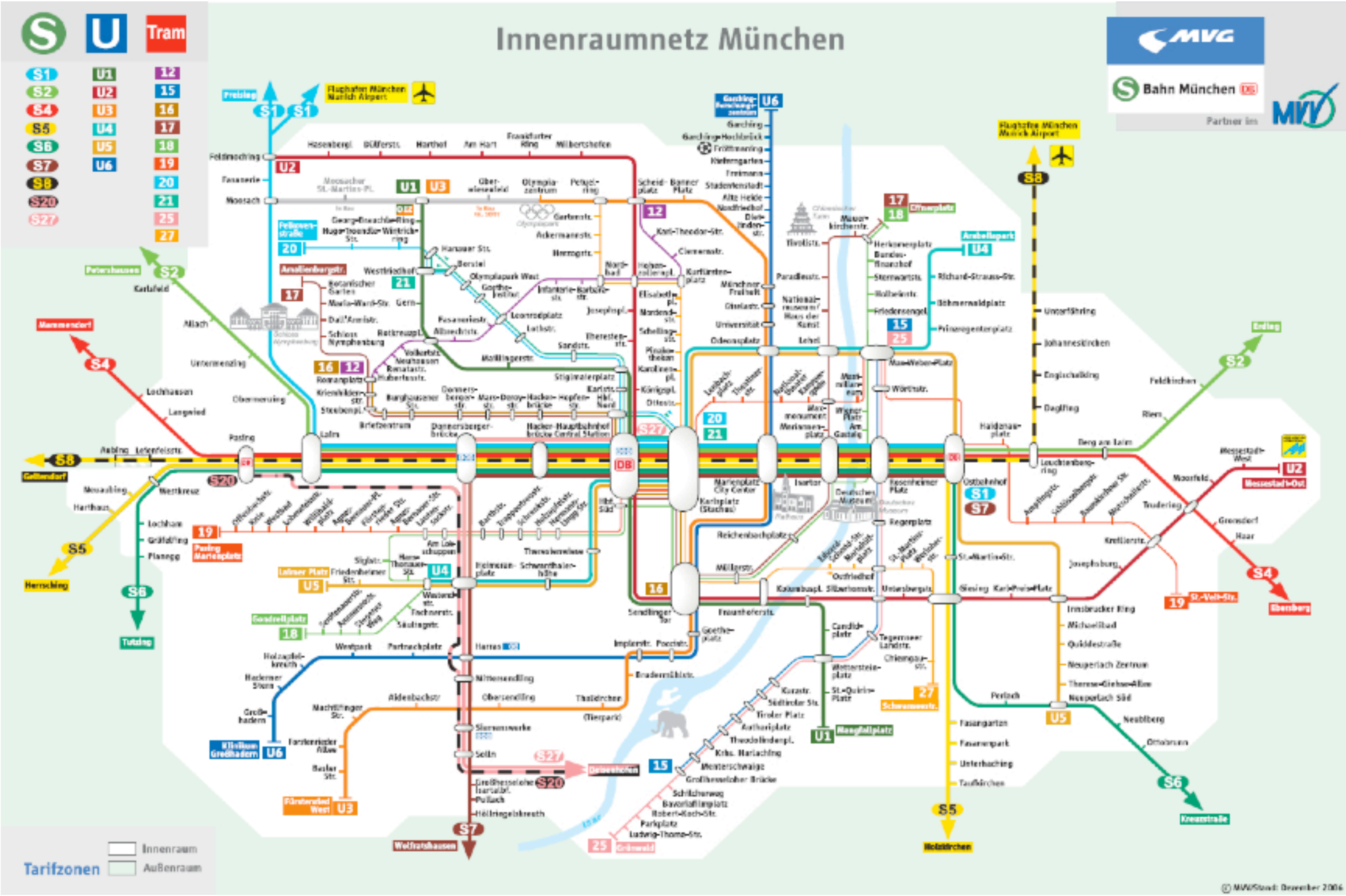


Transportation Network

http://de.wikipedia.org/wiki/U-Bahn_M%C3%BCnchen



Transportation Network



Transportation Network

- Objectives
  - Facilitate understanding of network connections
  - Fit size and aspect ratio constraint (positioned above the doors in the underground)
- Heavily distorted geographic positions, but still good readability for identifying shortest paths between stations

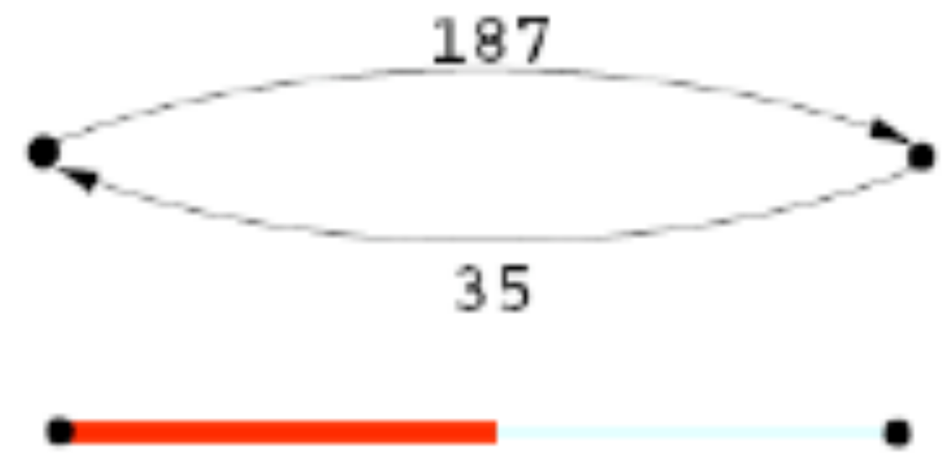
http://de.wikipedia.org/wiki/U-Bahn_M%C3%BCnchen

Telephone Network

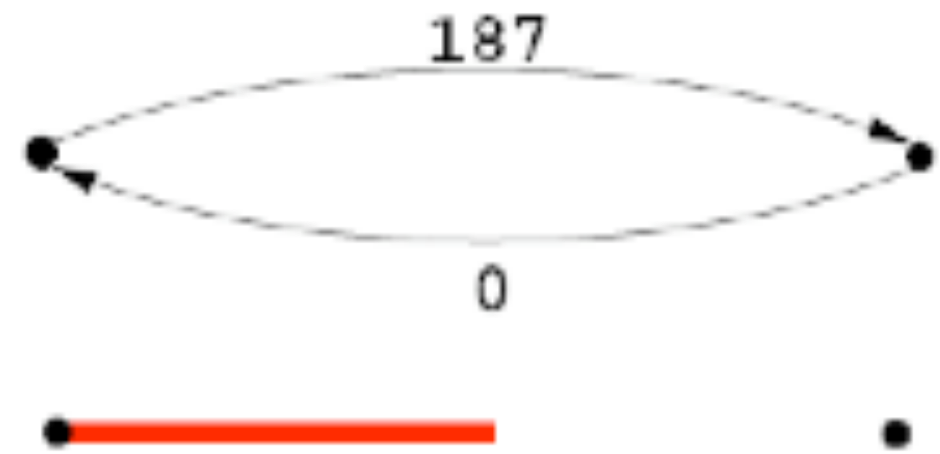
- Becker et al. 1995 - AT&T data
- 110 switches (nearly) completely connected
  - Each vertex has a geographic location
  - Statistics for each vertex, new data every five minutes
- 12,000 links between switches
- October 17, 1989 – earthquake in San Francisco Bay area
- Questions related to network capacity and traffic flows
  - Where are the overloads?
  - Which links are carrying the most traffic?
  - Was there network damage?
  - Are there any pockets for underutilized network capacity?
  - Is the overload increasing or decreasing?
  - Are calls into the affected area completing or are they being blocked elsewhere in the network?
- Different representations: linkmap, nodemap, matrix display

Linkmap Encoding

- Switches (vertices) are arranged according to their geographical position
- Two-tiled edges represent overload of in- and outgoing calls between switches
- Redundant coding to make the important edges more apparent: color and line-thickness both indicate amount of overload
- Reduce clutter by omitting edge segments where the overload value is zero

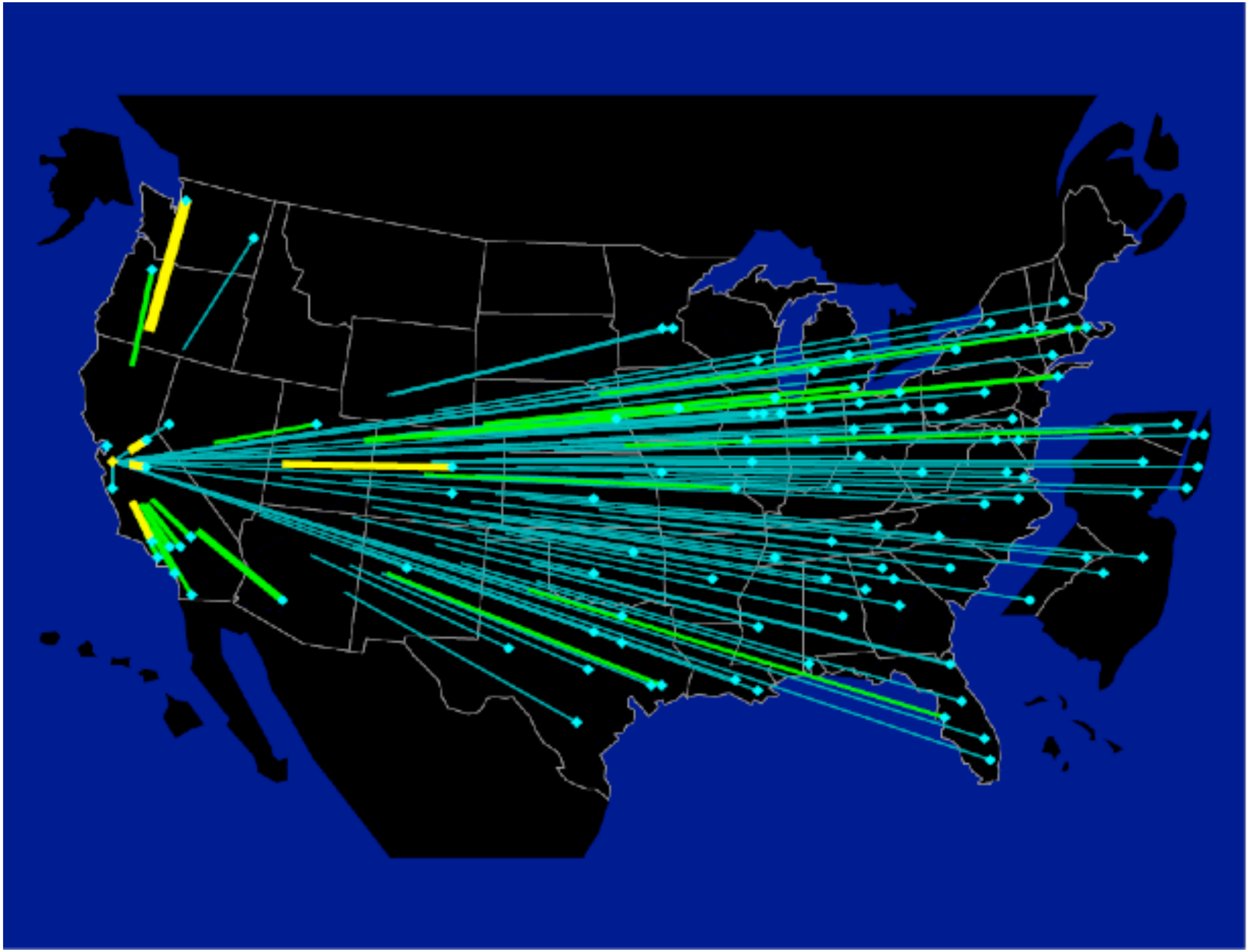


(a)



(b)

Linkmap - Oakland Switch

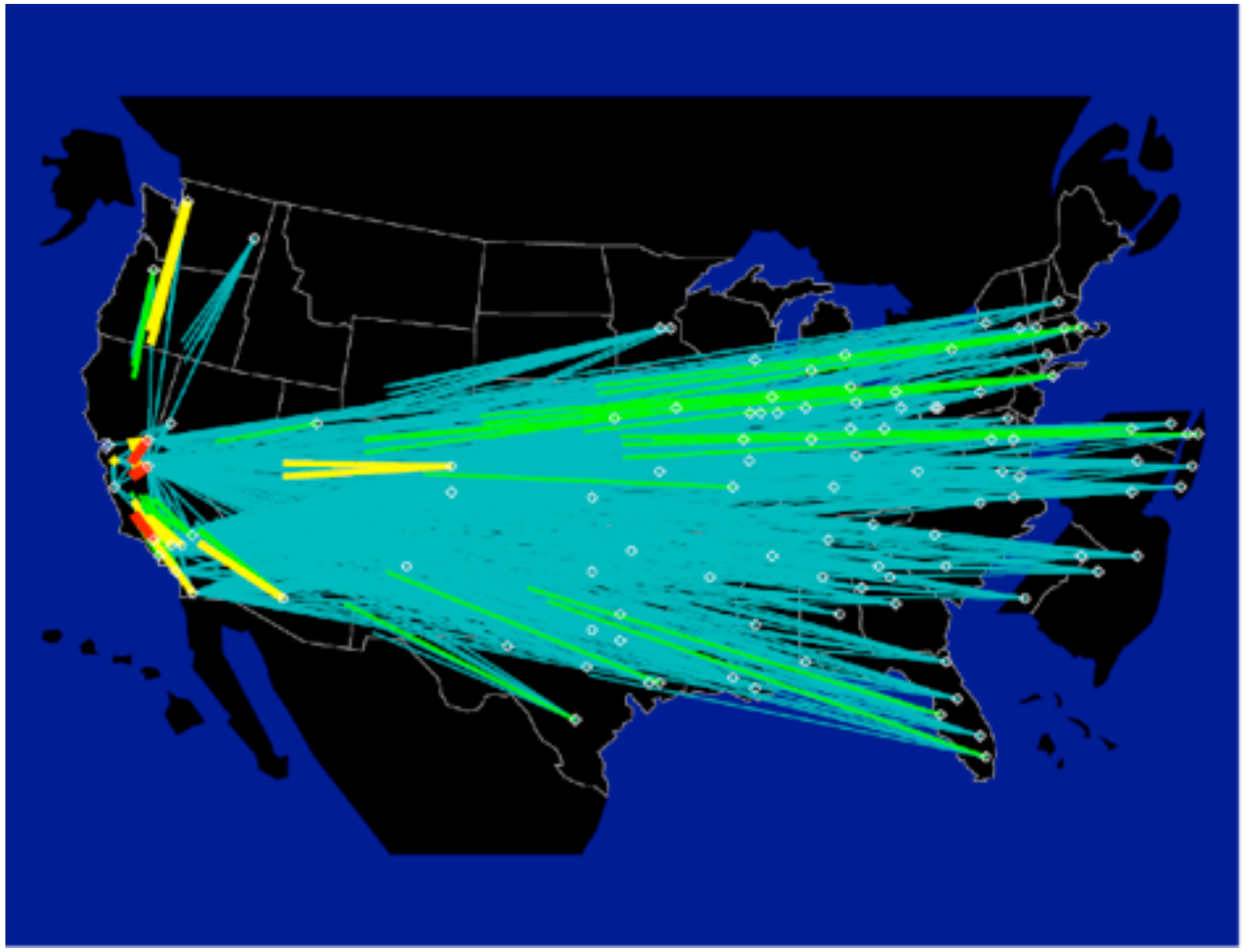


Linkmap - Oakland Switch

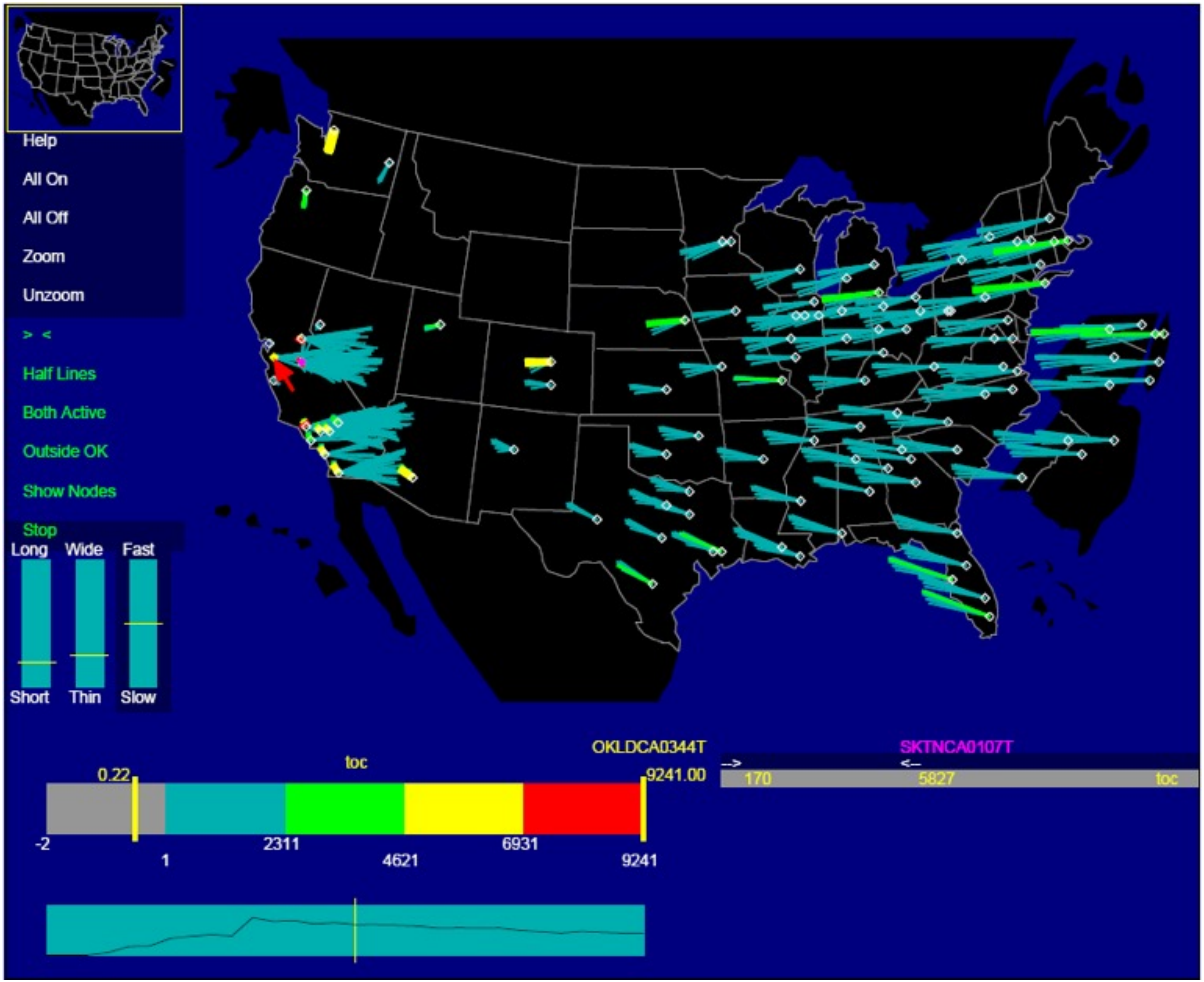
- Overload into one switch
  - Into Oakland switch from every other node (most heavily from Seattle and Denver)
  - Out of Oakland switch to many switches particularly on the east coast
- Island in the Atlantic Ocean is a blow-up of NY / New Jersey area (to reduce density of switches)
- Does work well because the edges hardly overlap
- What about showing total overload?

Linkmap - Total Overload

- Most important links are drawn last
- Still: display is ineffective because long edges from one coast to another obscure much of the country
- To reduce clutter: edge may be drawn only part way between the vertices they connect

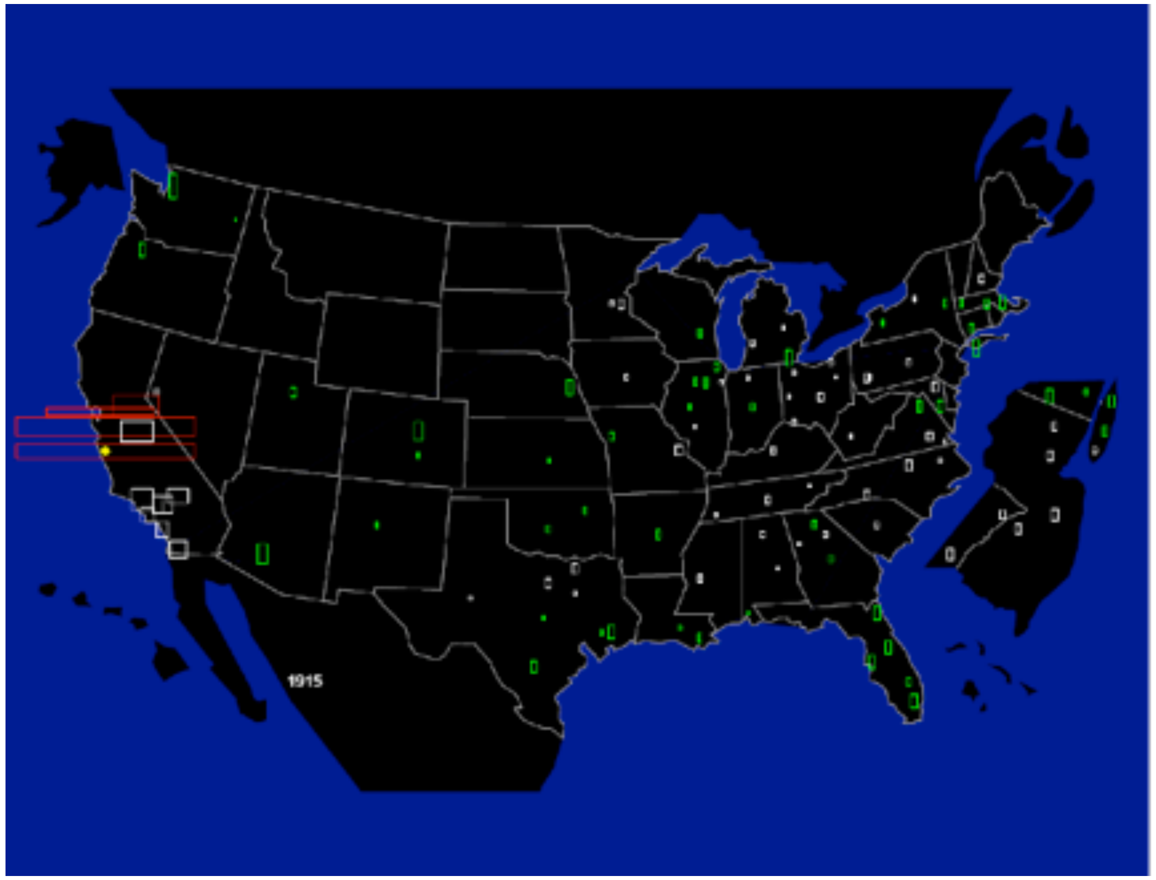


Linkmap - Total Overload



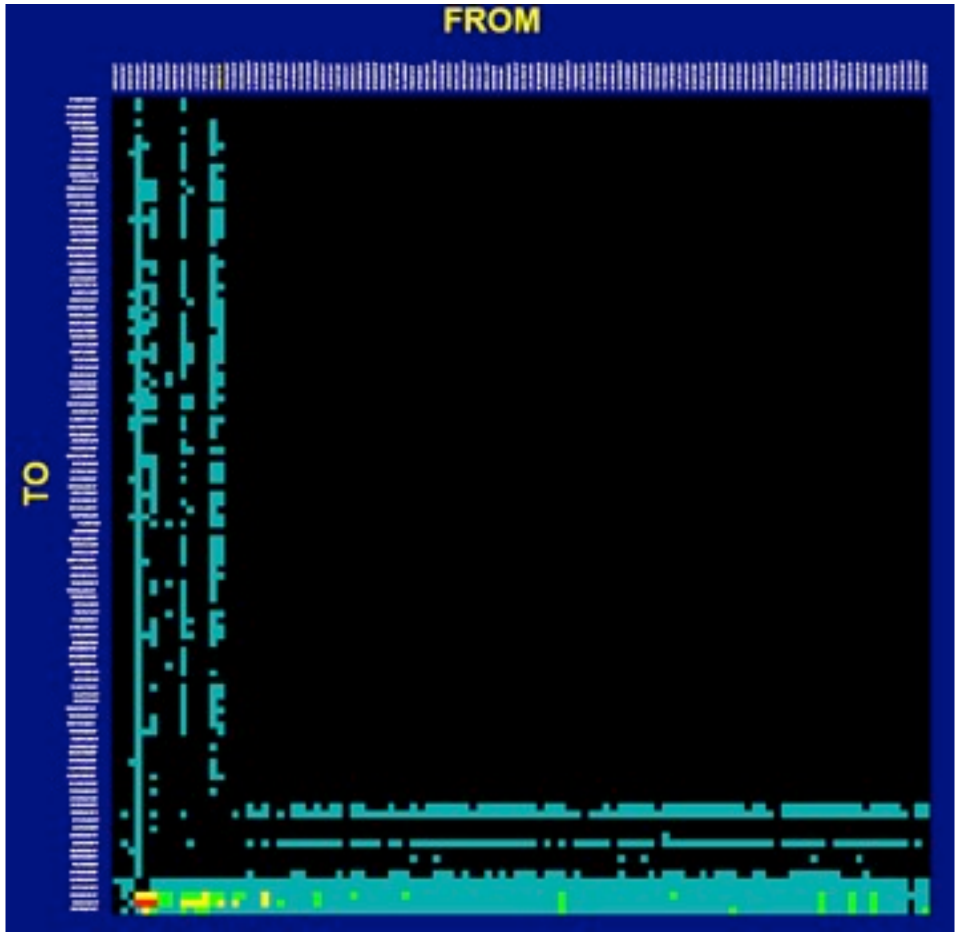
Nodemap

- Glyph encoding
  - Aggregate overload into and out of each switch
  - Rectangle width: proportional to the square root of the number of incoming calls
  - Rectangle height: proportional to the square root of the number of outgoing calls
  - Area of rectangle proportional to total overload
- Interpretation: overload of outgoing calls from nodes to northern and southern California
- Problem with this kind of representation?
- No clutter, but detailed information about particular links between switches is lost

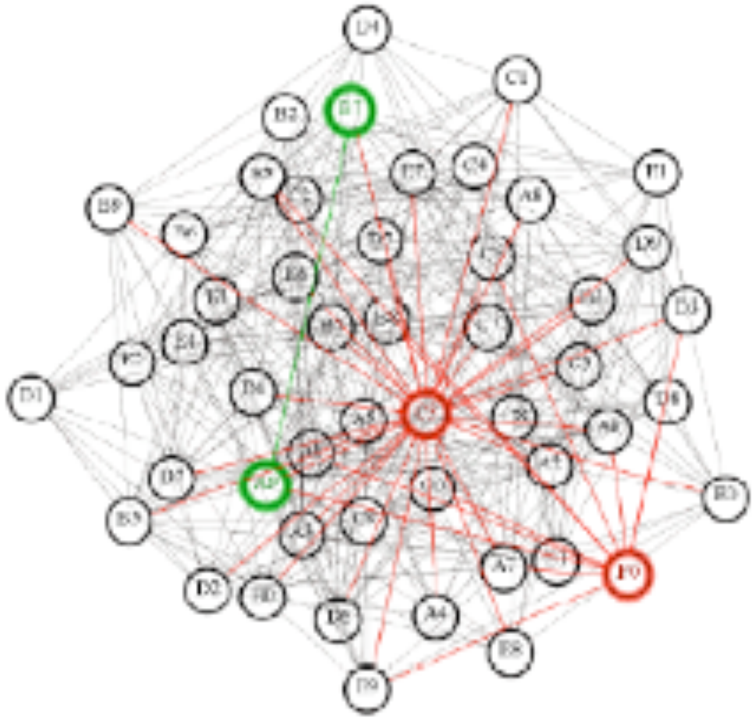


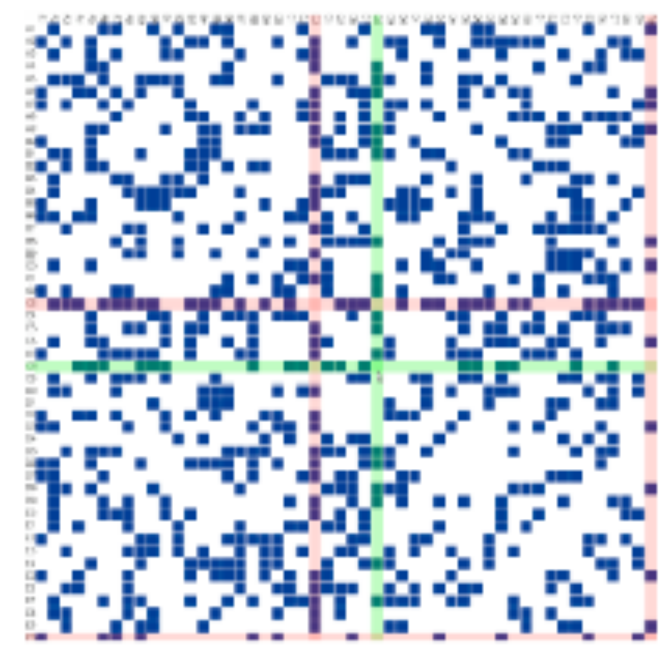
Matrix Display

- Omits information about geography
- Each matrix element is allocated to a directed link (half-line)
- Each switch is assigned to one row (incoming calls) and one column (outgoing calls)
- Switches are arranged west-to-east
- Interpretation
  - Five switches with major incoming overload (rows)
  - One switch with outgoing overload to almost every other node (column)
- Very compact visualization without clutter
- Problems with this kind of representation?
- Inference of the visualization is influenced by the ordering of the rows and columns
- Intuitiveness and readability when compared to a node-link diagram?



Node-link versus Matrix





- Ghoniem et al. 2004
- On-demand highlighting of selected nodes and links
- 36 participants
- Tasks to test readability
  - Estimation of number of vertices in the graph
  - Estimation of number of edges
  - Locating most connected node
  - Locate node by label
  - Find link between two specified nodes
  - Finding a common neighbor between two specified nodes
  - Finding a path between two nodes
- Random undirected graphs of three different sizes (number of vertices) and density (relative number of edges)

Node-link versus Matrix

- Independent variables
  - Graph representation
  - Number of vertices
  - Relative number of edges
- Dependent variables
  - Answer time (results not shown here)
  - Number of correct answers
- All users were familiar with node-link diagrams, but not with matrices
- Node-link diagrams seem to be well suited for small graphs but their readability quickly deteriorates with a growing size of the graph and link density
- Matrix provides a superior readability for large or dense graphs
- Node-link diagram only clearly superior for find-path task

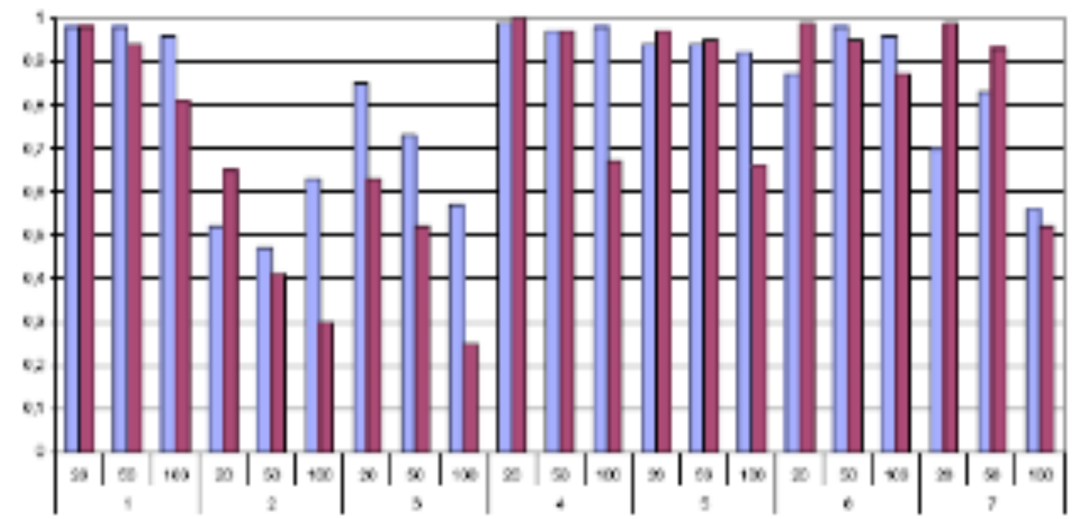


Figure 2 Percentage of correct answers split by task and by size. The matrix representation appears in blue and the node-link in purple.

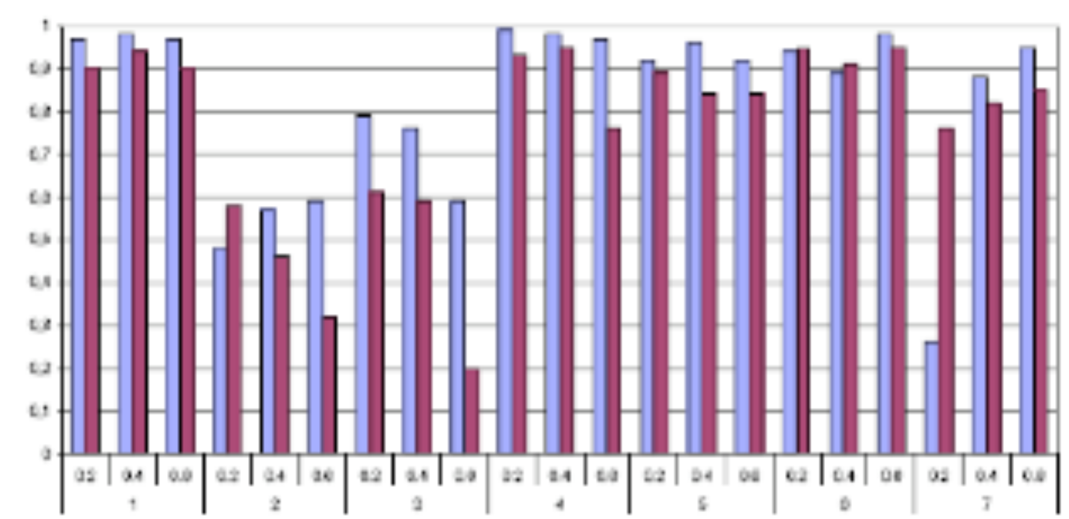


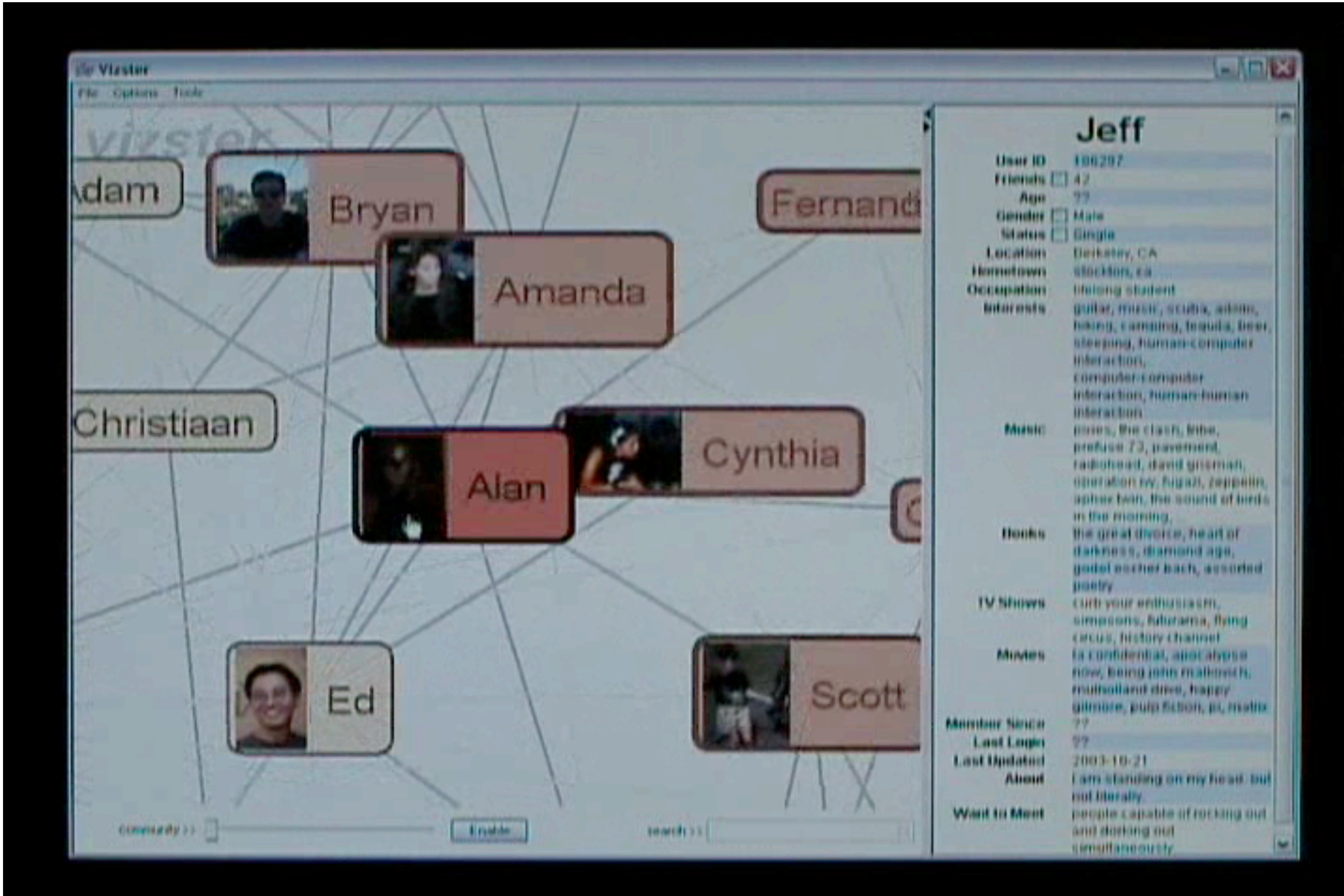
Figure 3 Percentage of correct answers split by task and by density. The matrix representation appears in blue and the node-link in purple.

Graph Interaction

- Dynamic visualization & interaction is essential for exploring / navigating graphs
  - Dragging and highlighting of vertices and edges
  - Filtering
  - Zooming & panning
  - Focus+context distortion
- Animation can support exploration

Focus+Context graph - Jankun-Kelly et al. 2003

interactive graph example: Vizster



Transitions in Radial Tree Layout

- Yee et al. 2001
- Radial tree layout: common technique in which the graph is arranged around a focus node
- Users can change the layout by selecting a different focus node
- Animated transitions of node translation
- Objective: keep the transitions easy to follow
- Animation mechanism
  - Linear interpolation of polar coordinates of the nodes
  - Follows ordering and orientation constraints

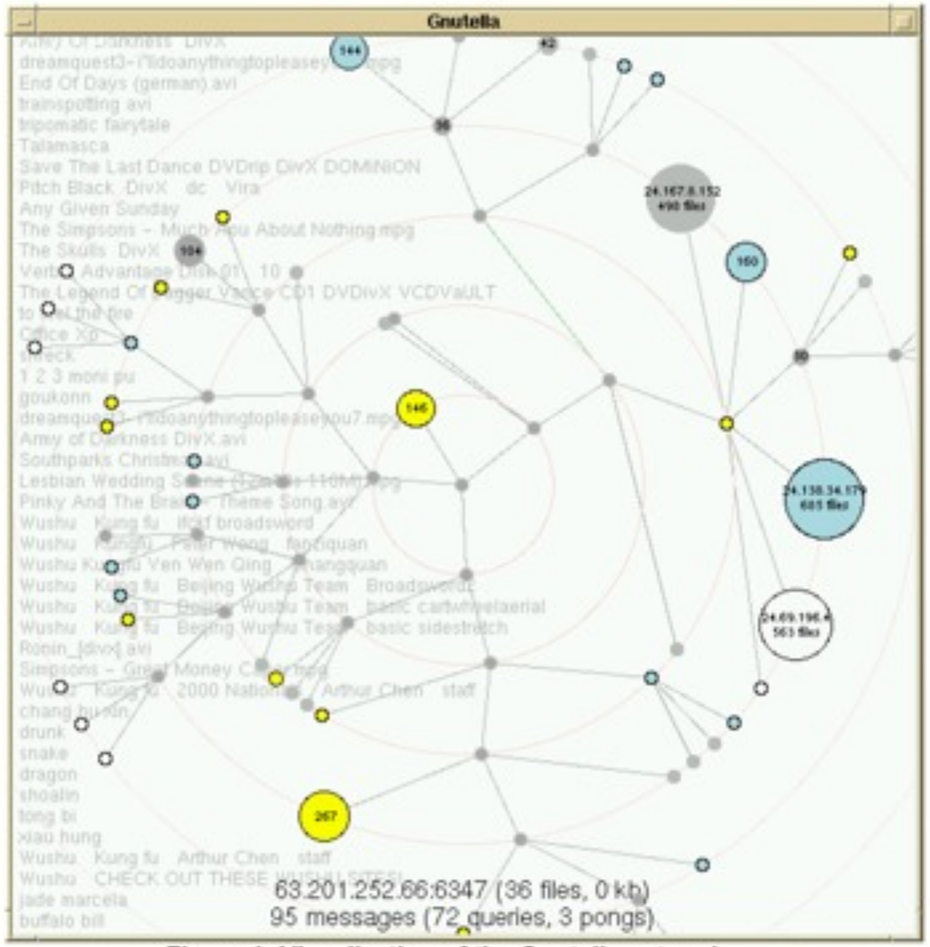
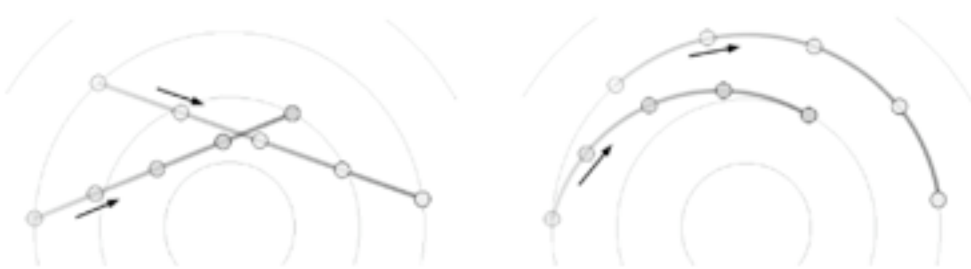
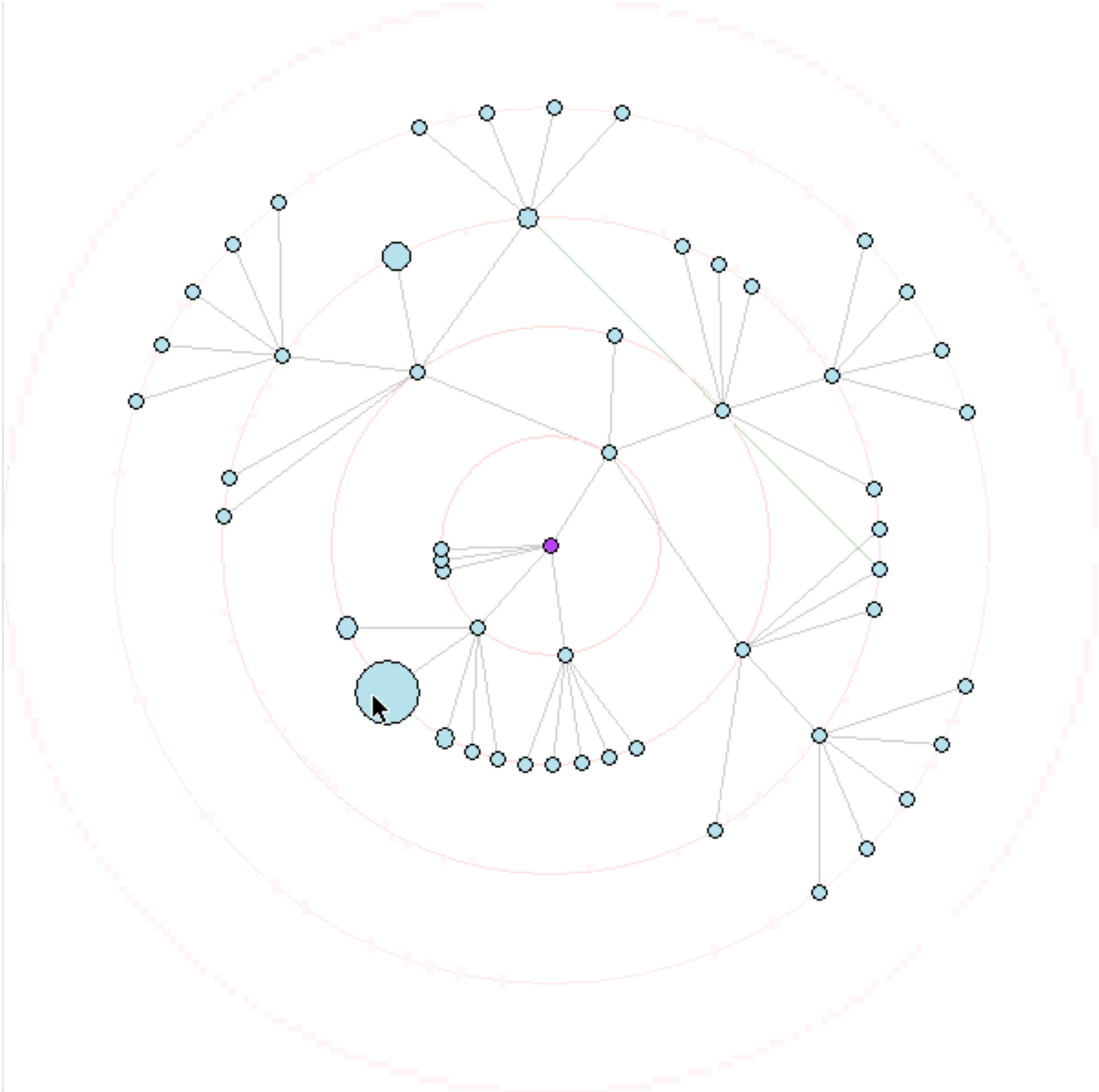


Figure 1: Visualization of the Gnutella network.

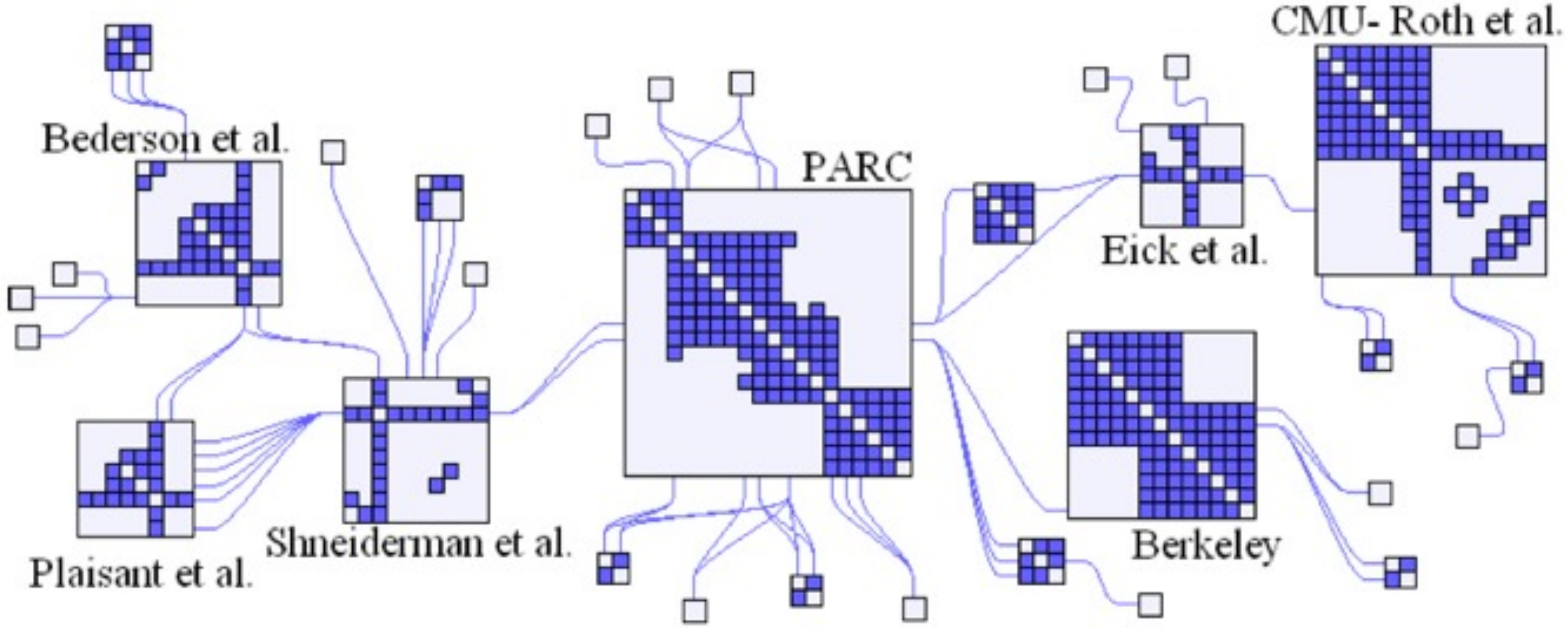


Transitions in Radial Tree Layout



Additional Sources and Literature

- recommended reading
  - Nathalie Henry, Jean-Daniel Fekete, and Michael J. McGuffin: "NodeTrix: A Hybrid Visualization of Social Networks", InfoVis, 2007.
  - http://insitu.lri.fr/~nhenry/docs/Henry-InfoVis2007.pdf



- Tutorials for graph theory and graph drawing
  - http://davis.wpi.edu/~matt/courses/graphs/