A Better Model for Generating Test Networks

Matthew B. Doar

Ascom Nexion†

Abstract

Much of the work on routing algorithms, particularly for multicast, which has been done in the past has used fairly simple models to generate the topological graph which represents the nodes in the network. Some such random graphs bear little resemblance to data communication networks which are actually deployed. This paper proposes a more realistic model for such random networks and describes various scenarios which can be more accurately represented. The approach described here can be developed to provide more refined models in the future, and the source code of an implementation is freely available.

1.0 INTRODUCTION

One of the major areas of interest in recent routing research has been how to route multicast packets in a connection-oriented network such as an IP network, and similarly, how to set up multicast connections in a connection-oriented network, such as an ATM network. Many algorithms have been described in the literature. Some produce an optimal solution [Karp72] but many concentrate on heuristic approaches [KMB81, Wall90], or the simplest approach [Doar93b].

Most of these algorithms have validated their approach though the use of extensive simulation. Typically, the graph of a network is generated by hand or algorithmically, with a number of nodes and some edges between them. Costs or weights are associated with each edge. A set of routing or connection requests are then created, again either manually or algorithmically, and the algorithm under test chooses a route or set of connections which meet the requirements of the request. The cost of the chosen route or connection is evaluated as the combination of the costs of the edges used and the constraints imposed upon the routing algorithm. This process is repeated many times for different graphs to avoid unforeseen interactions between the topology of a specif-

---

† Contact information: Matthew B. Doar, Ascom Nexion Inc., 289 Great Road, Acton, MA 01720-4739, USA. Email: mdoar@nexen.com, Tel: (508) 266 3468, Fax: 508 266 2300.

ic network and the algorithm under test. Creating each graph manually is tedious and prone to bias. The purpose of this paper is to examine the algorithmic generation of the graphs which represent the networks.

In Section 2, this paper describes various algorithms which have been used in the past. Section 3 makes some observations about real data communication networks, and indicates areas in which the algorithms of Section 2 fail to capture important aspects of some networks. Section 4 proposes an alternative approach for generating network models and Section 5 shows some preliminary results of the ideas in Section 4.

2.0 NETWORK TOPOLOGY MODELS

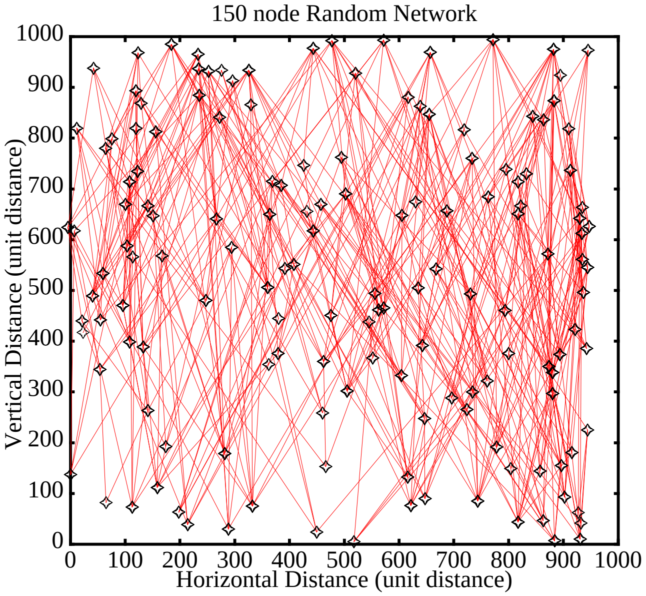
Early work such as that by Wall [Wall90] concentrated upon the theoretical aspects of multicast routing, specifically the problem of producing Minimum Steiner Trees for general graphs. Little simulation work was done and most of the graphs were small (less than 20 nodes), because they were used only for the purposes of proving the concept.

One of the most commonly used models for algorithmically generating graphs is that due to Waxman [Waxman88]. The nodes in the network are distributed at random across a Cartesian coordinate grid. Edges are added to the graph by considering all possible pairs (u,v) of nodes and using the probability function P_e, where

$$P_e(u,v) = \beta \exp(-d(u,v)/\alpha L) \qquad \text{(EQ 1)}$$

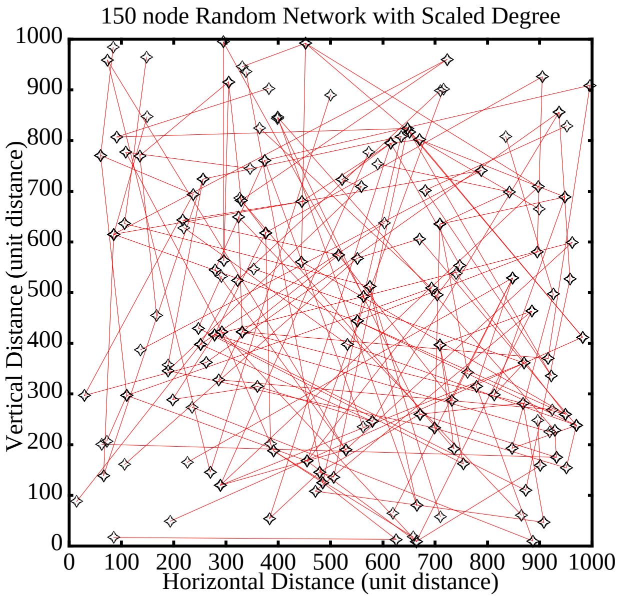
to create an edge, where $d(u,v)$ is the Euclidean distance between the nodes' locations, L is the maximum possible distance between the two nodes and α and β are parameters in the range $0 < \alpha, \beta \leq 1$. A large value of α increases the number of connections to nodes further away, whilst a large value of β increases the number of edges from each node. Besides other problems addressed in Section 3.1, this approach does not guarantee a singly-connected graph, and may have to be run multiple times to obtain one, testing the graph each time with a Minimum Spanning Tree algorithm [Sedgewick83]. Figure 1 shows a typical network generated using this approach. In defense of this approach, its original intention was for generating graphs in order to compare Minimum Steiner Tree algorithms with each other, for. which it is fully adequate

FIGURE 1. Typical graph due to Waxman, for $\alpha = 0.25$ and $\beta = 0.3$



A series of modifications to the model used by Waxman were made by the author [Doar93a] to examine the performance of a naive multicast routing algorithm compared to other more complex heuristics. One modification was to avoid the fact that as the number of nodes increases, the mean number of edges from a node (the degree) also increases. This was achieved by adding a scale factor to P_e related to the number of nodes in the network. A further modification was to scale the exponent in P_e by the number of edges already associated with a node, to decrease the probability of adding still more edges. Using the same values of α and β as used in Figure 1, the results of these modifications are shown in Figure 2.

FIGURE 2. Graph using a modified version of Waxman's algorithm with scaled degree



The work by Waxman, together with the modifications described above, has been used by a series of researchers [Bauer96, Biersack96, Deering96, Salama95, Voigt95] and others to investigate their own routing algorithms.

3.0 NETWORKS IN PRACTICE

The time is ripe to examine more closely exactly what it is that a graph of a network should model, when testing routing algorithms. A network has topological and metric aspects, that is, how the network is connected and what makes up the connections.

Considering the topological aspect first, a network can be represented by a graph at any layer, from the physical layer to the application layer. The graph for each layer may be different. For instance, two nodes which are physically connected may be unconnected at the network layer if they reside on different network layer subnets. Generally, graphs used for testing routing algorithms are a combination of the graphs for the network and all lower layers. The graph is supposed to represent all possible sources and destinations for data, usually assuming an endpoint is a source and/or destination at each node in the graph.

For example, a multi-homed IP connected host with two IP addresses, would be represented as two nodes in the network layer graph of a network. When considering ATM connections, an endnode would represent an AAL Service Access Point (SAP), since it cannot be assumed that replication can be done during reassembly in the AAL. An intermediate node in an ATM network may represent only the lower sublayer of the AAL, where a cell is replicated and switched to its destinations. If the switch contains one of the destinations, then that node also represents an AAL SAP.

The edges of a network layer graph represent the possible paths which may be taken from a source to a destination. In a simple network, there is only one path from source to destination. Extra paths come from redundancy being introduced at the network layer or lower layers. Such redundancy could be in the form of a second physical connection between hosts in a LAN, a second connection to a bridge to other LANs, or a second router attached to a LAN. In the wide area, nodes are often physically separated by wide distance but interconnected in a relatively dense manner to avoid ‘back-hoe loss’[†] and have multiple paths at the other layers too. Since such redundancy incurs extra equipment and configuration costs, it is more likely to occur in the critical areas of a network, such as those closer to a backbone, and also in networks used in failure sensitive situations or for real-time control.

[†] A back-hoe is a large mechanical excavator with a penchant for fibre optic cables.

FIGURE 3. An Actual Corporate Network (LAN and WAN)

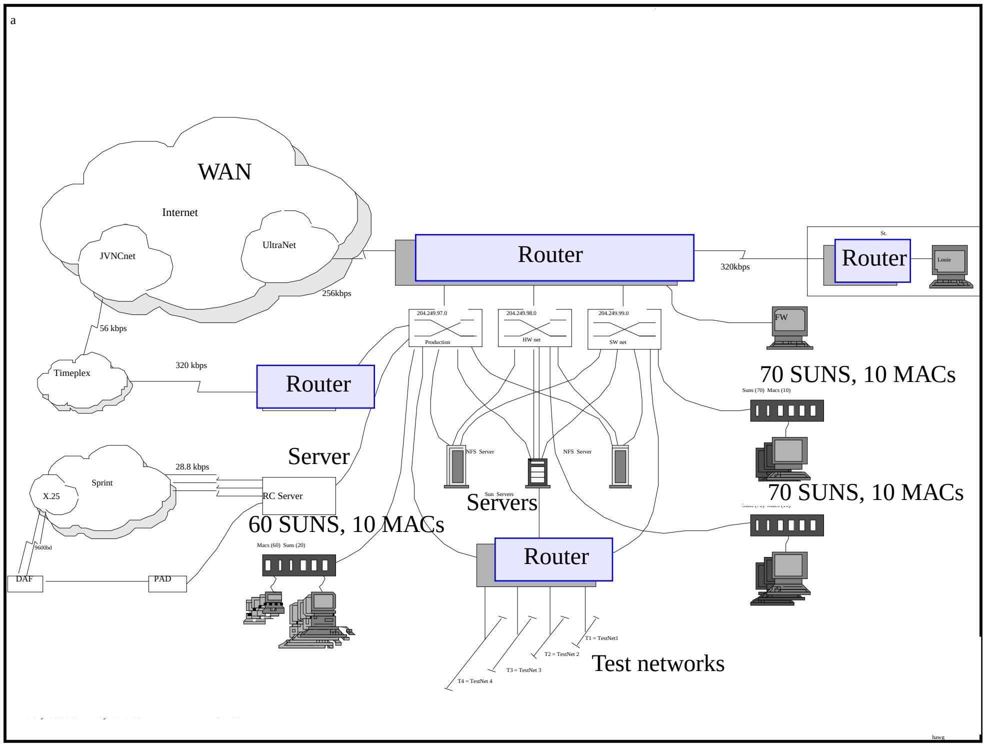
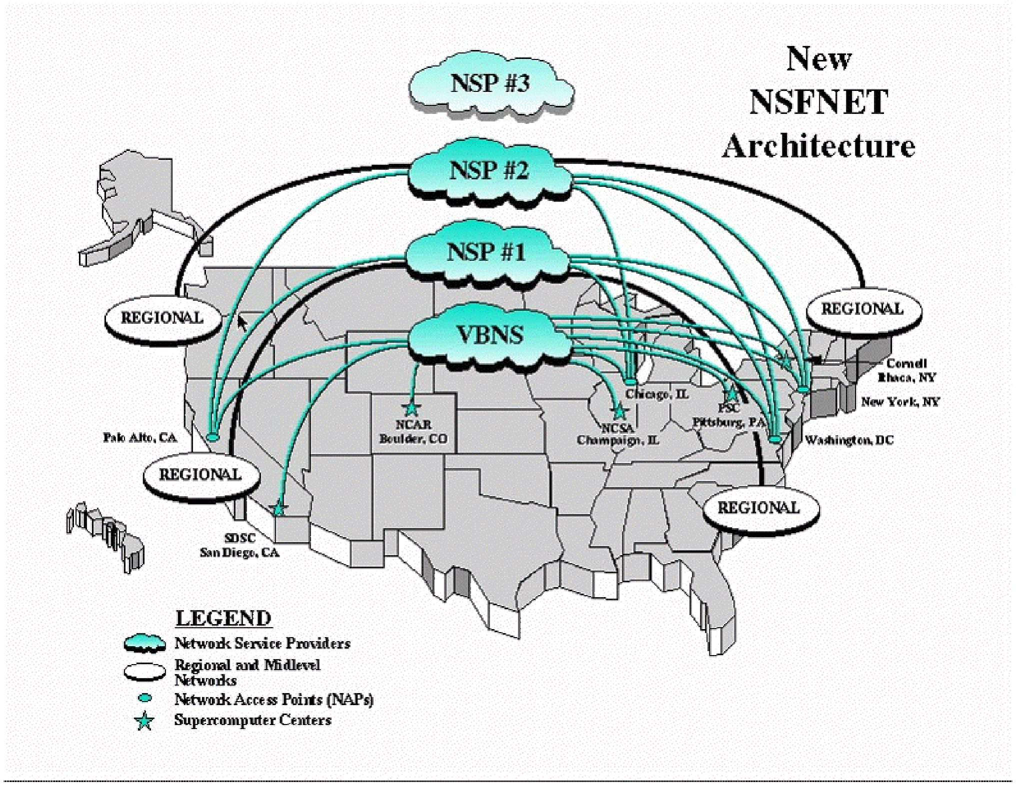


FIGURE 4. An Example WAN - the NSFNET backbone



Now consider the metric aspect of modeling networks. The two types of metrics which occur in networks are constraints and metrics which are to be minimized (or maximized). An example of a constraint is the bandwidth on a link. An example of a metric to be minimized is the transmission delay between a source and a destination. For the purposes of modeling, the transmission delay is often taken to be proportional to the physical distance between two points, with the addition of fixed processing overhead at certain nodes (e.g. routing from one subnet to another). Bandwidth is generally taken to be fixed and assumed to be shared in a linear fashion, i.e. one connection of 10Mb/s is equivalent to ten connections of 1Mb/s each. As a passing observation, this last assumption probably only holds true for large numbers of connections or routes on a link.

3.1 Differences Between Real Networks and Models

A series of observations can be drawn from considering the theoretical action of the network layer, Figure 3 and Figure 4 [NSFNET].

- As expected, redundancy, in terms of multiple paths between nodes, increases the more ‘central’ a node is in the network.
- Most nodes have very few paths to other nodes.
- Almost no nodes at the ‘edge’ of the network have connections directly to other edge nodes outside their own geographic area.
- Shared media topologies are common in the LAN, either as a bus, ring or star.

All of the approaches for generating network graphs described in Section 2.0 fail to agree with one or more of these observations and hence fail to accurately model the topology and metrics of existing networks.

4.0 A BETTER MODEL

Rather than use the idea of a network as a non-hierarchical series of nodes, we consider an internetwork as a series of smaller networks and concentrate upon the relationships between these smaller networks, and the relationships between the yet-smaller networks which make up each level of a network. This follows the concept of layers in a network more closely, where each layer is built upon a number of instances of the previous layer.

4.1 Basis of the Network Model

The model proposed has the following characteristics:

- Shared media LANs such as Ethernet and Token Rings are modeled as star topologies. This significantly reduces the number of edges in the graph and reflects the lack of physical redundancy in most LANs. There may be hundreds of hosts in a LAN. NBMA (Non-Broadcast Multiple Access) LANs such as switched Ethernet are also modeled as stars, with the switch at the centre of the star.
- LANs should be modeled as being interconnected in small numbers, with some small degree of redundancy in their connections.
- Each major institution has a small number of connections to a WAN service provider, who is in turn connected to other Internet Service Providers (ISPs) in a highly redundant fashion. Each ISP can be modeled as a single node and the WAN modeled as a single layer for the purposes of simplification.

- To simplify the graph redundancy in the network is limited to redundancy seen at the topmost (network) layer under consideration. Thus only routers in an IP network or switches in an ATM network appear as non-edge nodes in the graph. Edge nodes are the AAL SAPs in an ATM network and network layer SAPs in an IP network.
- The model shall be able to be run using few floating point operations in order to decrease simulation time when constructing thousands of network models with large numbers of nodes.

Nodes are categorized into three broad categories: edge nodes (LAN nodes), bridge, router or switch nodes (Metropolitan Area Network - MAN nodes) and gateway (WAN) nodes. When a host is part of a LAN and a MAN (for instance) the host is represented as two interconnected nodes, a LAN node and a MAN node. The metrics of the edge between the two types of nodes reflect the processing delay and bandwidth constraints within the host. This approach enables better estimates of routing algorithms' costs than the strictly edge-based approach taken by earlier work.

4.2 Network Model Parameters

A simple set of parameters is necessary to be able to repeatedly generate networks which bear some resemblance to real networks.The major parameters chosen for this model are:

- N_W, the number of WANs and S_W, the number of nodes in a WAN. N_W is taken as 1 for simplicity.
- N_M, the number of corporate/institutional networks (MANs) and S_M, the number of nodes per MAN. Since every MAN is connected to a WAN node, $N_M \leq N_W$.
- N_L, the number of LANs per MAN and S_L, the number of nodes per LAN. Again, $N_L \leq N_M$, since there is a MAN node for every LAN.

The total number of nodes in the graph, N, is given by

$$N = S_W + N_M S_M + N_M N_L S_L \qquad \text{(Eqn. 2)}$$

Typical values might be $S_L = 50$, $N_L = 5$ and $S_M = 10$, $N_M = 10$ and $N_W = 5$ for a corporate internet. This would make N equal to 2605 nodes. For a smaller graph, modeling just the MAN and WAN environment, typical values might be $N_L = 0$, $N_M = 10$, $S_M = 5$, and $N_W = 5$, giving N = 55 nodes in the network (a more tractable size for short simulations).

The other parameters of the model are:

- The degree of intranetwork redundancy in the WAN (R_W), MAN (R_M) and LAN (R_L). This is expressed simply as the degree (number of directed edges) from a node to another node of the same type. So R_L is usually 1, R_M might be 2 and R_W could be 3.
- The degree of internetwork redundancy between networks. This is the number of connections between a MAN and a WAN (R_{MW}) or a LAN and a MAN (R_{LM}).

The parameter values for the model are taken as the basis for distributions used to obtain the actual value for the algorithm. Extra information can be associated with each parameter to describe the distribution of the parameter. For instance, an upper and lower bound on S_L and the function for distributing the value between the bounds could be described. In this paper, the distributions used always return the exact value of the parameter, except for the redundancy parameters which are treated as upper limits in a uniform random distribution from 1 to the limit. This means that N, the total number of nodes, is constant from run to run. For more realistic scaling, the redundancy parameters R_{MW} and R_{LM} could use a distribution which keeps the mean value low as S, the number of nodes in a network, increases, to model the fact that doubling a network’s size does not necessarily mean adding more connections to the next level of network.

A small table of costs and constraints associated with each type of edge is also defined as part of the model. This table contains an entries for the 3x3=9 kinds of edge: for a WAN to MAN edge, a MAN to WAN edge, a LAN to MAN edge and so on. Each entry contains constraints on the bandwidth and other characteristics such as the speed of transmission in the media (often assumed constant in all parts of the network). For instance, to define an ATM backbone in a company, the bandwidth constraint for MAN to MAN edges might be set at 155Mb/s. This table can also be used to describe characteristics such as the processing delay in a router interconnecting two LANs, since this is where the LAN to MAN edge characteristics are stored.

4.3 Network Model Algorithm

Once the parameters for the network have been chosen (five basic ones, five others and up to six bandwidths, if bidirectional symmetry is assumed) the graph for the network can be constructed by defining the individual edges between the nodes. Any method used to do this must create a singly-connected graph, with the requisite degree of redundancy in each group of node types and between groups of different node types. The model proposed uses a Minimum Spanning Tree [Sedgewick83], being a simple, well understood algorithm with the requisite singly-connected property, which can also handle asymmetrical costs of an edge. It is also often used as the basis for laying out large networks.

The approach used in the earlier work to generate the cost of an edge between two nodes by placing the nodes at random on a grid is useful for its simplicity. However, one of the problems with this approach is that the differences in physical scale between a LAN and WAN are not accurately represented. The algorithm outlined below uses the same idea of a grid of points, but changes the scale of the grid for each type of network.

STEP 1. The WAN is created, with the nodes placed at random in the grid, but possibly rejecting the placement of a node if it is too close to another node (see Appendix A).

STEP 2. When the requisite number of nodes have been placed, a minimum spanning tree is created between them.

STEP 3. Each node is examined, in some random order, to see whether it meets the intranetwork redundancy parameter, R_W. A node may already have more than R_W edges to a peer node due to the action of the minimum spanning tree algorithm. In this case, no further edges are added. If the number of edges to peer nodes is less than R_W, then edges are added to the closest nodes in the network, in increasing order of Euclidean distance.

STEP 4. The metrics for each edge can be calculated using the Euclidean distance between the two nodes in the grid and the table of costs associated with each type of edge.

Steps 1 to 4 are then repeated using a smaller scale of the grid for MAN networks with the MAN parameters, without the proximity rejection test in STEP 1, since MAN (and LAN) nodes are often closer together than WAN nodes. Nodes are permitted to occupy the same points in the grid, since the purpose of the grid is to generate costs using an approximation of nodes’ physical locations.

STEP 5. The LAN networks are created by choosing one node in each LAN as the centre of the star and connecting every other node to it with a single edge. In the unlikely event that R_L is greater than 1, then STEP 3 can be used. STEP 4 follows in any case.

The various types of networks are then interconnected in Steps 6, 7 and 8.

STEP 6. The MANs are to be connected to the WAN. This is achieved by randomly choosing a collection, A, of N_M nodes from the N_W WAN nodes, with duplicates permitted. An edge is created between each node in A and one node (X) chosen at random from each MAN. Each MAN is now connected to the WAN by a single edge.

STEP 7. If R_{MW} is greater than 1, then for each MAN, create another edge from another node in the MAN (possibly node X again) to the closest node to the node in A which X was already connected to. This ensures some degree of local connectivity within geographic constraints.

STEP 8. The LANs are connected to the MANs. This is done as in Steps 6 and 7 using the LAN parameters, except that the node X chosen in the LAN is always the centre of the star.

4.4 Complexity

Since testing a routing algorithm requires running it on a large set of test cases, as well as manually generated pathological cases, it is desirable that an algorithm for generating the graphs should have a reasonable performance in terms of the time and memory used. Each step of the algorithm is considered in turn for its time complexity. Space complexity is not assumed to be a major issue.

STEP 1. With no proximity testing, adding nodes is $O(N)$. With proximity testing, it becomes $O(N^2)$, since as each node is placed, it must be compared with every node already placed.

STEP 2. The graph of the potential number of edges to be considered is dense, so a minimum spanning tree algorithm such as Prim's [Sedgewick83], with complexity $O(N^2)$, is appropriate.

STEP 3. The method of adding redundancy implies that the closest R-1 edges to each node must be known. If R is nearly equal to N, then this implies keeping a sorted list for each of the N nodes, giving $O(N^2 \lg N)$. More likely is that R is small and so scanning all the nodes for the R closest ones will be more efficient. The complexity of the latter is $O(N^2)$, since R is a constant much less than N.

STEP 4. Evaluating the cost of each edge is $O(NR)$.

Steps 1 to 4 are repeated to create the MANs.

STEP 5. Connecting the LAN nodes together: if RL = 1, this is $O(N)$, otherwise $O(N^2R)$.

Adding redundancy:

STEP 6. Connecting the MANs to the WAN takes $O(N_M)$ operations.

STEP 7. Adding redundancy to the connections between MANs and the WAN is $O(NR_{MW})$, since the sorting of nodes by proximity has already been done in STEP 3.

STEP 8. Connecting the LANs to the MANs is the same complexity as Steps 6 and 7.

Since all the values of R are likely to be small integers, the expected complexity of the graph generation algorithm is $O(N^2)$. Section 5.2 confirms this. The proposed algorithm is carefully designed to create a sufficient, but not enormous, number of edges for large networks through the use of hierarchies of nodes and the ‘star’ simplification for LAN topologies. Another attractive property of the model is that at most one edge is generated between a pair of nodes, simplifying implementation of the algorithm.

For comparison, the complexity of Waxman’s original algorithm is also $O(N^2)$, with potentially long generation times for the probability function when α was large or β was small. The scaling issue with the original algorithm as noted in Section 2.0 also leads to very large numbers of edges and hence long simulation times.

5.0 RESULTS

Figure 5 shows a single level of network as generated by the algorithm. The basic MST can be seen, together with the edges added for redundancy. Some nodes have a degree larger than R_W due to the action of the MST, but most nodes have only up to R_W edges. The network was originally generated on a 100x100 grid, then scaled by the scale factor for WANs (40), so the units of distance are the smallest unit in the network. No proximity test was applied when placing the nodes.

Figure 6 shows a two node WAN, with a single MAN and its five attached LANs. Each LAN has 25 nodes, which can be seen to be attached as a star. The LANs in the upper right corner of the figure overlap. This is permitted since the grid is just a way to generate costs and to make it easier to draw two dimensional diagrams of networks which have a hierarchy and at least two physical dimensions.

FIGURE 5. A typical generated WAN (or MAN)

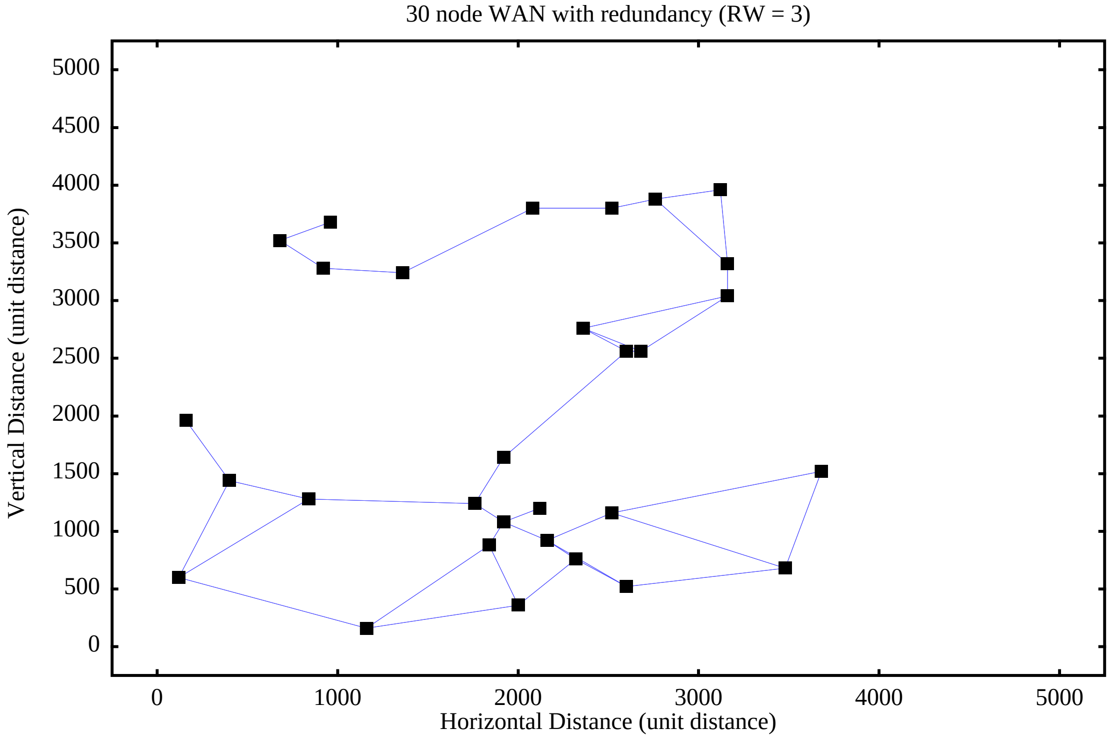


FIGURE 6. Generated MAN with satellite LANs.

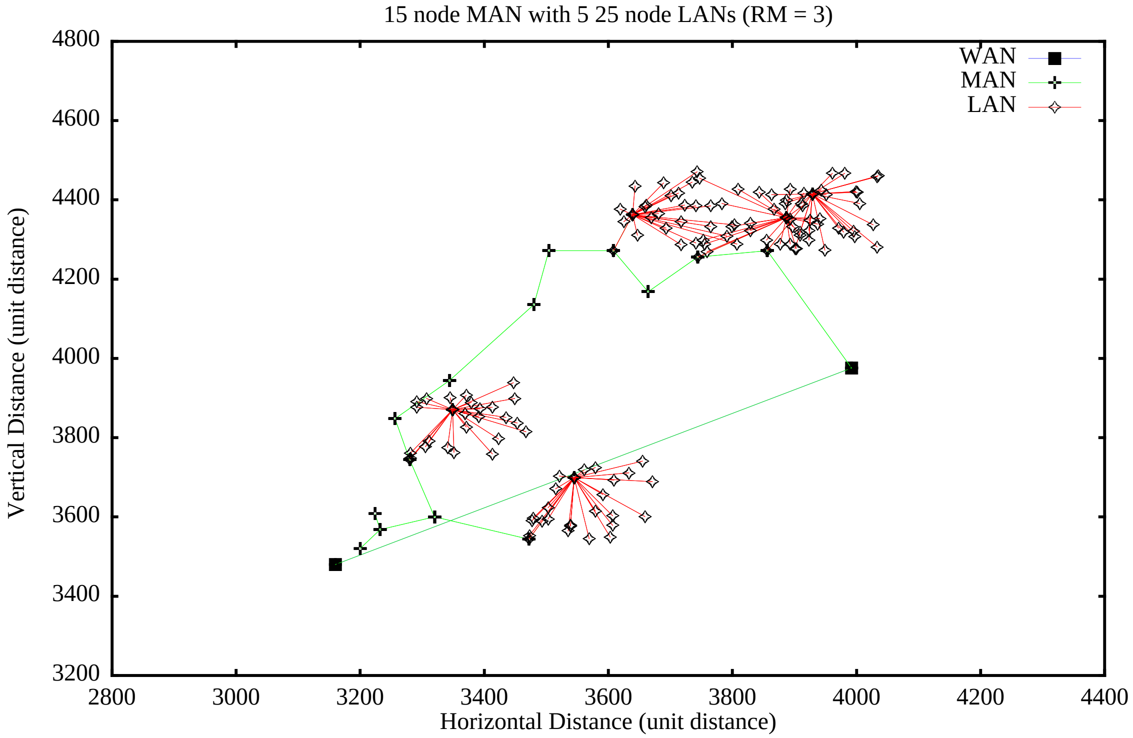


FIGURE 7. A typical generated LAN

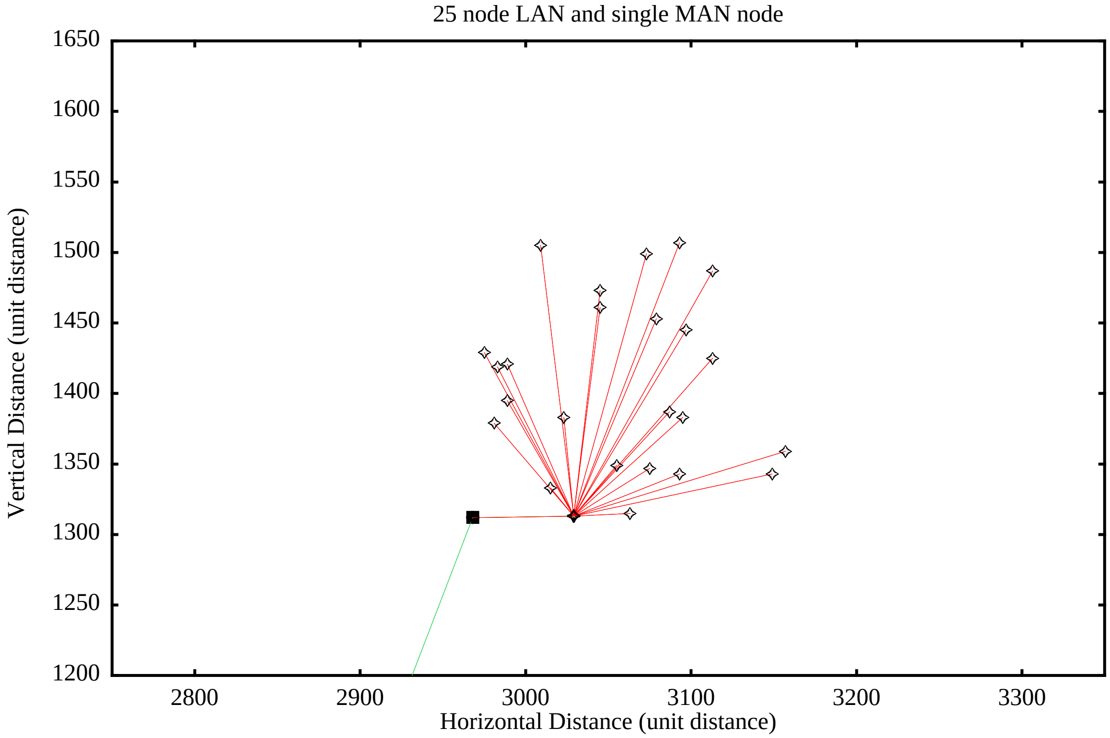


FIGURE 8. Overall picture of a generated network

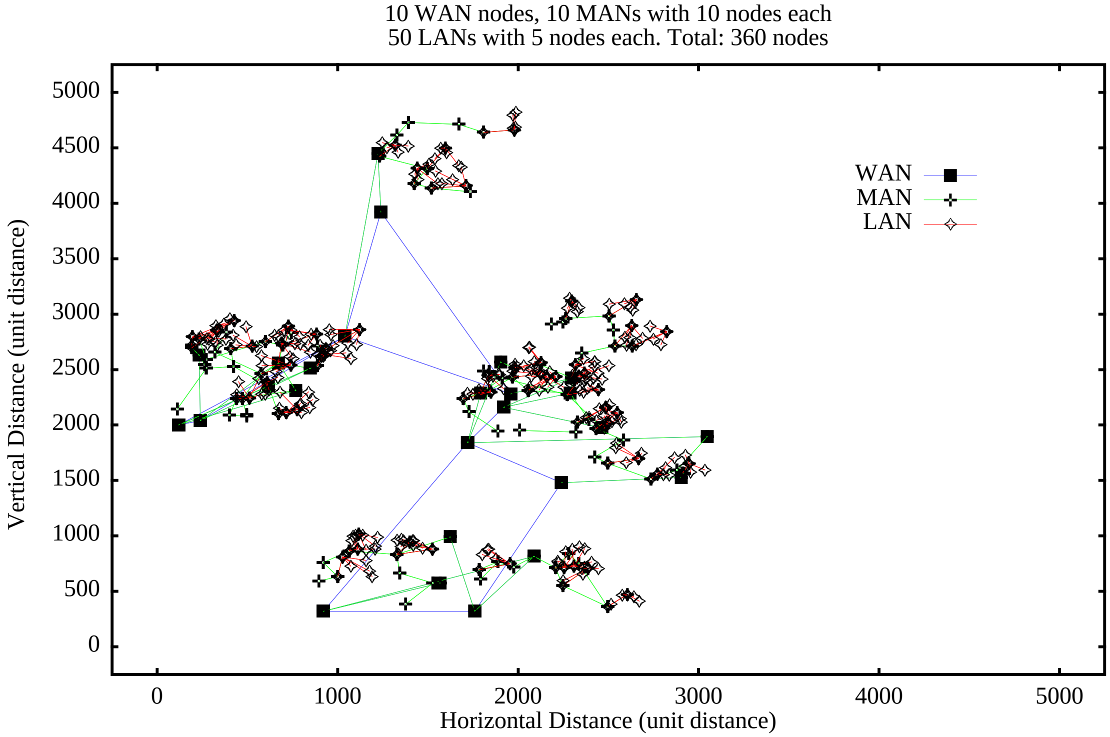


Figure 7 shows a typical LAN and how it is, in this case, attached to just one MAN node.The scale of the figure has changed to a much smaller range. Figure 8 shows an entire network, which contains too much detail for much of the finer points to be seen. However, a WAN backbone is clearly present, and locality of MANs and their LANs and the redundancy of MAN connections to the WAN can be seen ($R_{MW} = 2$). The requirements noted in Section 3.1 are met.

The scale factor used for MANs was 8 and for LANs was 2. These scale factors were chosen to produce illustratory figures, not for realism. More realistic scale factors might be 10^7 for WANs, 10^5 for MANs and 10^3 for LANs, with a unit distance of meters.

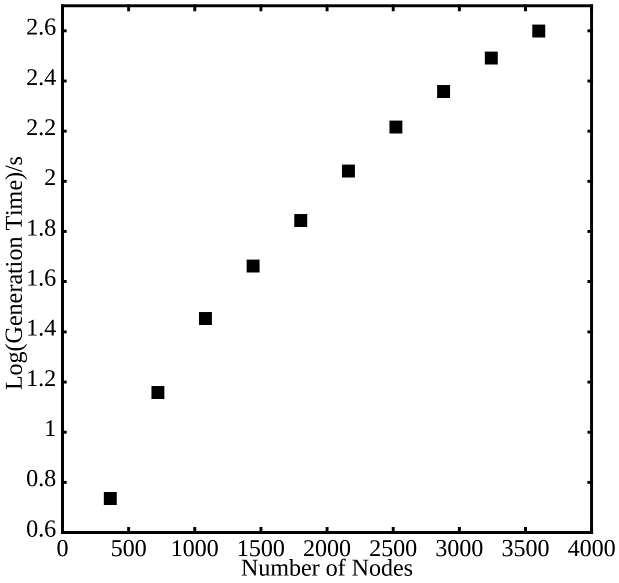
5.1 Validation of the Model

One simple way to compare two networks is to use the parameters which were used to generate the model network as measurements of the real network. Before embarking upon extensive simulation runs, estimates of the five basic parameters and of the five redundancy parameters should be made using real networks. Question such as “how many LANs do we have?” and “how many links to the WAN are there?” are much easier for system administrators to answer than vague questions such as “does this graph resemble your network in any way?”. Future work in this area is to obtain values of the ten parameters from a wide range of institutions. More complex modifications to the basic algorithm for adding redundancy to the model could also be used in future work.

5.2 Running Times

Figure 9 shows how the time to generate a network varies as the total number of nodes in the network increases. The network used was the one shown in Figure 8 and was scaled by incrementing S_W and N_M. The time was measured as the sum of the system and user processing times and the mean of five samples was used as a data point. Disregarding the first two data points, a straight line fitted to the rest of the points gives a complexity O($N^{2.2}$), tending to confirm the complexity discussion of Section 4.4

FIGURE 9. Generation Time *vs.* Number of Nodes



6.0 CONCLUSION

Given that small differences in a network's topology and metrics have a large impact upon the behavior of routing algorithms, it is important to ensure that the network models being used to test the algorithm bear some resemblance to the real networks for which the algorithm is designed.The proposed network generation algorithm is believed to generate such networks. The proposed algorithm also has a reasonably small number of parameters, most of which might be expected to remain constant between runs of generated networks. The parameters all have obvious meanings which relate directly to the generated networks, unlike other approaches described earlier.

A simplification to reduce number of edges, and hence improve simulation times, is to model LAN topologies as stars. Another useful idea is to treat a node which connects two types of networks as two nodes, with one node in each type of network. This permits modeling of the delay in transferring data from one network to another, which is often neglected in other network models. In the future, once more data about the characteristics of the networks to be modeled has been gathered, better distributions for the parameters' values can be defined. Examining how the values of the parameters change with time may even help predict how networks change within organizations.

Further examples of generated networks and the complete C++ code used to produce them can be found at `http://www.TODO.com/papers/tiers` or by contacting the author. Many thanks are due to the system support staff at Ascom Nexion for information concerning their corporate network.

6.1 Appendix A

An approximate value is required for when one point is 'too close' to another after distributing N points in a K x K grid. One very simple way to estimate this is to view the K^2 area of the grid as divided up into $\sqrt{N}$ rectangles, each with a point in its centre. The distance from point to point is then $K/\sqrt{N}$. In a proximity test, one might use a further scale factor and say that if the distance between two points is less than $0.5K\sqrt{N}$, the two points are too close.

As noted in Section 4.4, the complexity of this test is $O(N^2)$. For small values of N, this complexity of the proximity test is reasonable. For larger numbers, schemes such as hashing the coordinates into buckets and measuring bucket occupancy could be more appropriate.

6.2 References

[Ballardie9x] TODO - how did they simulate CBT?

[Bauer96] *Distributed Algorithms for Multicast Path Setup in Data Networks*, F. Bauer and A. Varma, IEEE/ACM Transactions on Networking, April 1996

[Biersack96] *WAVE: A New Multicast Routing Algorithm for Static and Dynamic Multicast Groups*, E. Biersack, J. Nonnenmacher, Institut Eurecom, Sophia-Antipolis, France. `ftp://ftp-i4.informatik.rwth-aachen.de/pub/Public/biersack_new.ps.Z`

[Deering96] *The PIM Architecture for Wide-Area Multicast Routing*, S. Deering, D. Estrin, D. Farinacci, V. Jacobson, C.-G. Liu and L.Wei., IEEE/ACM Transactions on Networking, April 1996

[Doar93a] *Multicast in the Asynchronous Transfer Mode*, J.M.S. Doar, Ph.D. Thesis, University of

Cambridge Computer Laboratory, April 1993. Available as TR 298 `http://www.cl.cam.ac.uk/ftp/papers/reports/TR298-jmsd-thesis.ps.gz`

[Doar93b] *How Bad is Naive Multicast Routing?*, M. Doar and I.M. Leslie, Infocom 1993, San Francisco. `http://www.cl.cam.ac.uk/ftp/papers/jmsd/jmsd-multicast-routing.[ps|tex].gz`

[Karp72] *Reducibility among Combinatorial Problems,* R.M. Karp, in *Complexity of Computer Communications*, R.E. Miller and J.W. Thatcher (Eds), pp. 85-103, Plenum Press, New York, 1972

[KMB81] *A Fast Algorithm for Steiner Trees,* L. Kou, G. Markowsky and L. Berman, Acta Informatica 15:141-145, 1981

[NSFNET] See `http://www.merit.edu/nsf.architecture` for details concerning the NSFNET.

[Salama95] *"Evaluation of Multicast Routing Algorithms for Distributed Real-Time Applications of High-Speed Networks"*, H. Salama, Reeves, Viniotis and Sheu, 6th IFIP Conference on High-Performance Networks (HPN '95), September 1995

[Sedgewick83] *Algorithms,* Robert Sedgewick, Addison-Wesley, 1983

[Voigt95] *A Tool for Configuring Multicast Data Distribution over Global Networks.* R. Voigt, R. Barton, S. Shukla, Proceedings of Hypermedia INET '95 Conference

[Wall90] *Selected Broadcast in Packet-Switched Networks*, D.W. Wall, in *Multicast Communication in Distributed Systems*, M. Ahamad (Ed), IEEE Computer Society Press, 1990

Waxman88] *Routing of Multipoint Connections*, B.M.Waxman, IEEE J. Selected Areas in Communications 6(9), December 1988