SIMULATING AND MEASURING BURNOUT ROBUSTNESS OF DAMAGED MESH SPATIAL NETWORKS

M.G.B. Johnson and K.A. Hawick

Email: { m.johnson, k.a.hawick}@massey.ac.nz

Complex Systems and Simulations Group, Computer Science, Institute of Information and Mathematical Sciences, Massey University, Albany, North Shore 102-904, New Zealand

http://complexity.massey.ac.nz

Introduction

Mesh networks arise in several application areas including social nets; power systems networks, communications systems, and especially in peer-peer networks of highly mobile devices such as sensors or future generation personal devices such as phones or tablet computers. Studying the resilience and failure modes of simulated mesh networks as highly central nodes "burnout" and are removed can be highly valuble in creating robust networks. We use centrality metrics to quantitatively assess network degradation as critical nodes are progressively removed. Network damage is then parameterised in terms of the number of most critical nodes removed.

Mesh networks are becoming widely used in: urban scenarios; mining scenarios with health and safety implications; security and sensor applications including intruder detection networks; as well as in communication networks where availability is also the key concern. They typically arise in the context of wireless devices deployed at spatial locations that must each receive and transmit their own data as well as act as a relay for other devices in the structure. A degree of interactive collaboration between all nodes is therefore necessary, and the usual enforced hierarchical relationships between hubs, router, server nodes and so forth that exist in a fixed conventional network do not necessarily apply. A mesh network can be further complicated if the nodes are mobile. A mobile *ad-hoc* networks (MANET) can be implemented as a mesh network where the traffic propagation routes are recomputed as individual nodes move around.

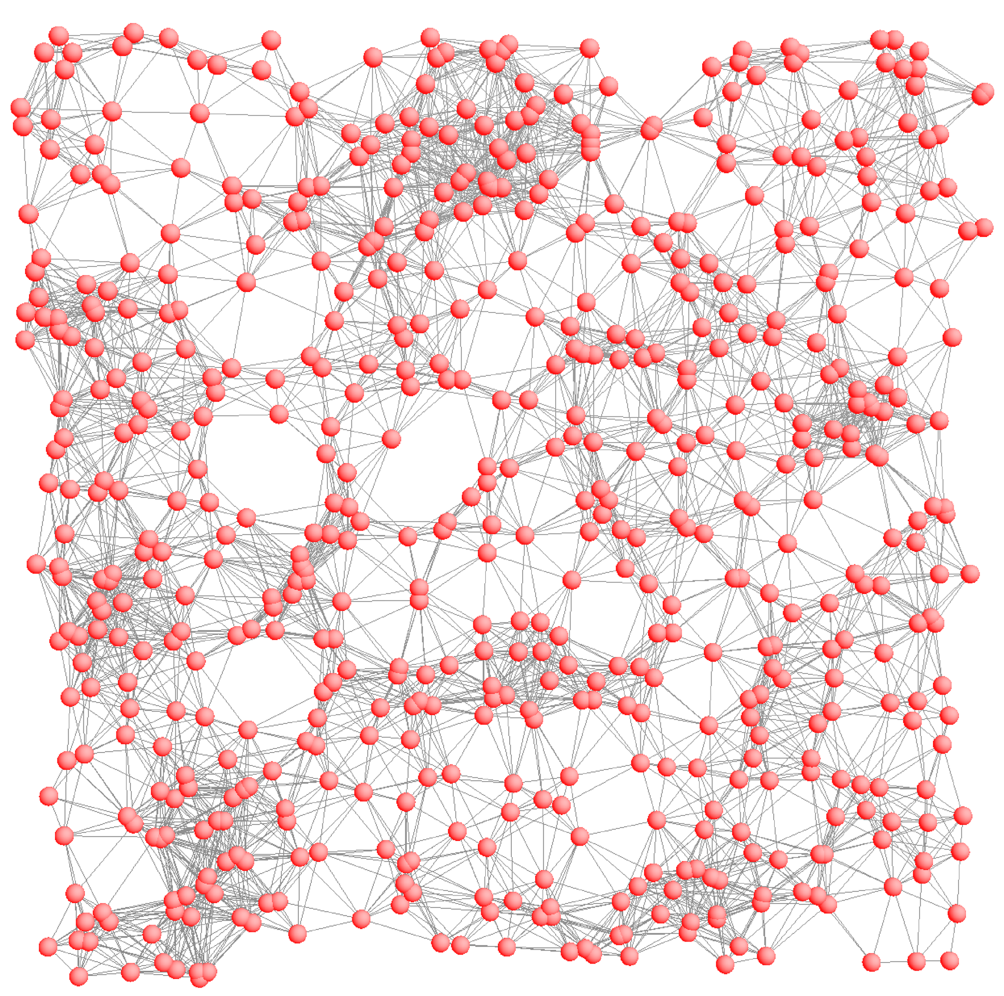

FIGURE 1: Visualisation of a two-dimensional, N = 600, non-periodic system where $R = 0.1$. Showing Connections between linked nodes.

We explore the robustness and reliability of mesh networks by simulating the consequences of nodes failing. We consider various centrality measures including the node degree and the between-ness centrality to rank individual nodes according to their importance and criticality to the network as a whole. We investigate how the bulk properties of a mesh network changes as the most highly ranked nodes are progressively "burned out" or removed. This is a fairly realistic scenario as the most heavily critical nodes might well be the most overloaded in a realistic mesh net, and therefore the most likely to fail.

Method

The commonly used notation for a mesh network that van be modelled as a graph is to refer to Graph G with a set V of vertices or nodes. There are then N_V nodes and also a set of E connecting edges of which there are N_E individual peer-peer connections.

Centrality metrics rank the mesh network nodes in an order according to which is the most highly connected or critical to the network as a whole. The simplest centrality metric is the in or out degree of a particular node. This is just the number of other peer nodes that it connects to or from. Simple static degrees like this do not necessarily give insights into the wider implications of a particular node failing.

The "between-ness" centrality metric is defined in terms of the node through which the highest number of pathways connecting any two **other** nodes pass. The between-ness is computed using the shortest path distance between each pair of nodes $(s,t); s \in V, t \in V$. The fraction of the shortest paths that pass through each vertex v is computed and summed over all possible pairs of vertices (s,t). This can then be written as:

$$C_B(v) = \sum_{s \neq v \neq t \in V} \frac{\sigma_{s,t}(v)}{\sigma_{s,t}} \qquad (1)$$

where $\sigma_{s,t}$ is total number of shortest paths from s to t and $\sigma_{s,t}(v)$ is the number that pass through v. This can be normalised by dividing by number of node pairs not including v. This factor is $(n-1)((n-2)$.

Computing shortest-paths for a network is a long-standing problem and there are several algorithms available [1–3]. Often the choice for networks that are not too large is dominated by the ease of integration with the data structures and the other software tools deployed. The computational complexity for the shortest-paths is $\mathcal{O}(N_V^3)$ using the Floyd-Warshall algorithm. There are other and newer algorithms such as Brandes' algorithm, which takes $\mathcal{O}(N_V N_E)$ **[?]**. In this present work the calculations were done repeatedly as networks were progressively allowed to fail and it was sufficient and easiest to use implementations of the Floyd-Warshall algorithm in our own software.

The shortest-path distance sometimes known as the Dijkstra distance is an average over all pairs of nodes in a graph. It characterises in a bulk sense the connectedness of a graph in a single positive distance-like quantity. It is possible to weight the distance by actual spatial distances, but it is also common to simply use number of hops or node-node traversals in its unweighted form. We can write this for our graph or network of vertices $v \in V$ so that for each pair of vertices $s \in V, t \in V$ we find the shortest path $p^*_{s,t}$ that minimises the (weighted or unweighted) path length over all possible paths $P_{s,t}, foralls, t \in V$ between vertices s, t. We can normalise appropriately so we arrive at a Dijkstra distance D so that:

$$D = \frac{1}{N(N-1)} \sum_{s,t} p^*_{s,t} \qquad (2)$$

where $N = |V|$ is the number of vertices. Usually we do not count self connections and we must also take account of the case when the graph fragments and there is no longer a connecting path between a particular pair of nodes. This is normally done by ignoring disconnected pairs, and effectively restricting the sum to pairs within connected components. The transition from a fully connected single component network to a fragmented one thus shows up appropriately as a discontinuity or sudden change in the Dijkstra distance.

Simulation Techniques

The networks we experiment with in this paper are simulated systems, generated according to a model with a single parameter. To simulate the effects of targeted node elimination in the irregular network we randomly generate *N* points on either a two-dimensional unit square or a three-dimensional unit cube. These points are each stored in an eight byte C++ double and passed to a $Node$ class that stores the position of the Node, its betweenness, radius and a vector of links to other nodes within the radius of it.

We create a one-dimensional array of Nodes of size N then each points position is determined randomly in space. Once all points are allocated we iterate over them all and calculate which points are connected to other points and store these links in the C++ STL vector type of Links. The simple $Link$ class contains a pointer to the connected Node and the distance to that node forming an undirected graph.

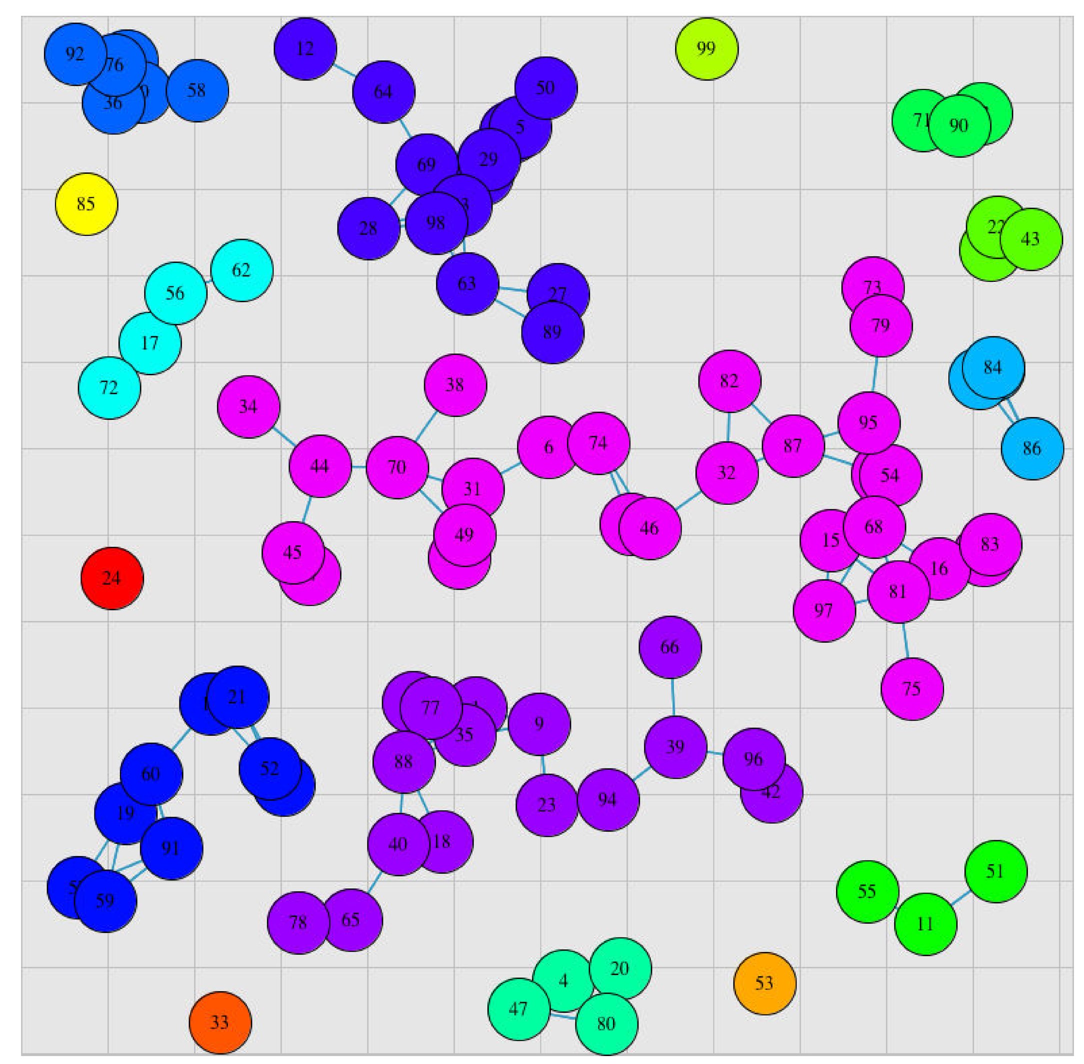

FIGURE 2: A typical generated *ad hoc* network of locally connected nodes, forming clusters in a spatial system.

When assigning positions in a unit square/cube with a static radius (R), higher values of N causes the number of connections per node to increase. This is representative of real world wireless systems where the transmission power of each node is static and the number of connections per node can increase rapidly. We explore the effect of different radii on the between-ness in Figure 2. Once the system is initialised the we run the Floyd-Warshall algorithm to find the paths from each node to any other node and store the paths. We then compute the betweenness using the stored paths and determine the maximum betweenness which we store to be averaged and plotted for results.

Once this process is completed we remove the node with the maximum betweenness, recompute the links between all of the nodes and run the algorithm again up to $N/4$ times. We arrived at the $N/4$ value through trialling various configurations. This complete simulation in then run two hundred times to get an average for the final results.

Results and Discussion

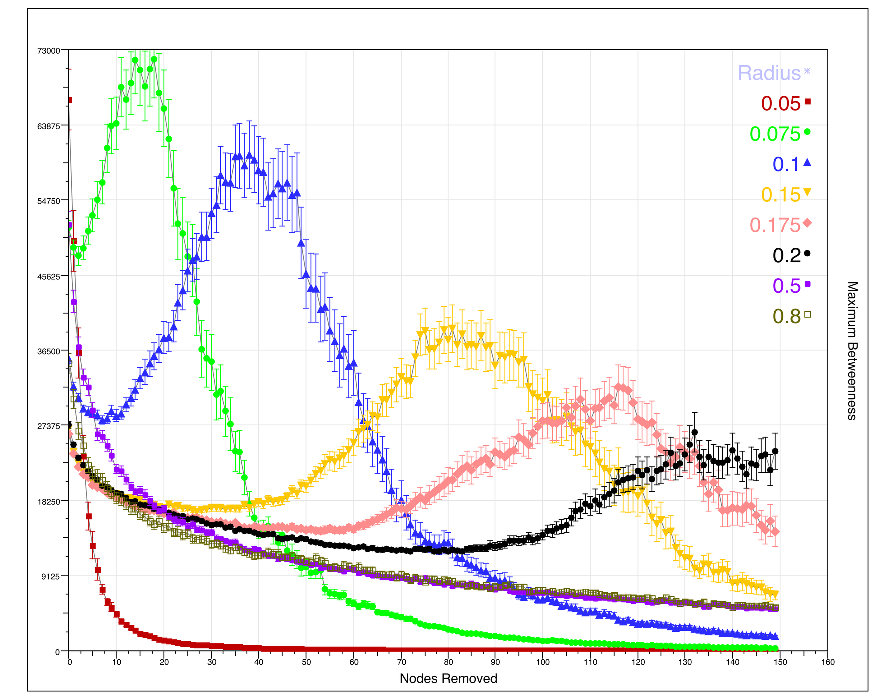

FIGURE 3: Dependance of critical temperature of the damage parameter p for a $Q = 2$ system

The figure above shows us that the effect of different radii on the maximum between-ness vs the number of culled nodes in a non periodic, two dimensional system where $N = 600$. The first radius tested was $r = 0.05$ and this network failed to maintain any stability as the number of connections was too small to maintain a contiguous network. For radius $r = 0.75$ the effect of removing the nodes with the highest between-ness shows a high peak followed by steep degradation of the network. We see the between-ness drop from the initial configuration to a local minimum then rise sharply to a peak. this local minimum represents the most desirable network configuration because the maximum reliance on a single node is minimised. This effect is visible from $r = 0.75$ through $r = 0.2$. For $r = 0.5$ and 0.8 we see a gradual degradation of that network rather than a specific peak.

We see that the between-ness metric can be usefully applied as a metric to study the properties of a damaged mesh network, which might arise from mobile sensors, or from some other *ad hoc* arrangement of devices or individuals deployed in a spatial system. We developed the model of N randomly placed devices or individuals with the single model parameter of spatial proximity radius r. The number of individuals placed in a unit box effectively controls the system density and the radius controls the connectivity. In effect it appears that it is the ratio between these that governs behaviour of the model.

We have seen that system behaviour changes quite dramatically depending upon whether the model system is periodic or not. This seems to be largely due to the more complete connectivity for a given radius, in a periodic system. In a non-periodic system the nodes are classified as interior nodes with a straightforward average connectivity, or boundary nodes which will have a reduced connectivity. We also see measurable differences in the change in the between-ness for two and three dimensional systems and the resulting differing average connectivities. There is scope to explore this effect further with four-dimensional or higher dimensional simulated systems. It might also be usefully investigated in other mesh patterns or geometries.

References

[1] Prim, R.: Shortest connection networks and some generalizations. The Bell System Technical Journal **November** (1957) 1389–1401

[2] Dijkstra, E.: A Note on Two Problems in Connexion with Graphs. Numerische Mathematik **1**(**1**) (1959) 269–271

[3] Goldberg, A.V.: A simple shortest path algorithm with linear average time. In: Proc. 9th Annual European Symposium on Algorithms. (2001) 230–241

For full paper and references please see the full text at http://complexity.massey.ac.nz/