Lower Bound Reassessment of a Quasi Randomized Rumor Spreading Protocol

Charles Drake Poole
Undergraduate Student; Dept. of Mathematics
UMass Dartmouth, Dartmouth, MA 02747
Email: drake@ise.gen.in

Abstract

Quasi randomized rumor spreading protocols are used to spread information across a network. Doerr-Huber-Levavi showed that a quasi random rumor spreading protocol is as robust as the randomized protocol. They estimate the time steps required for robustness on a complete graph and use this estimation for every spanning subgraph. It is easily demonstrable that the spread of information across a graph is not dependent only on the node count, but also the edge distribution. Through simulation we have shown there exists a subnetwork of a network with an average degree of connectivity exceeding 13% with other nodes that represents a the minimum spread in complexity, with only an increase in average rounds of 14%. Our results show there exists a lower bound than the one proposed for complexity with the given probability of transmission failure when connectivity is considered.

Problem Formulation

Robustness by Protocol

Rumor spreading can be done with a variety of protocols. A basic protocol would go something like,

- If I know look at a list of neighbors
- Tell the first neighbor on the list
- Tell the next neighbor on the list
- Repeat until list is exhausted.

This method will eventually inform every node on a graph, but requires every node to go through it's respective list before the protocol could be considered finished.

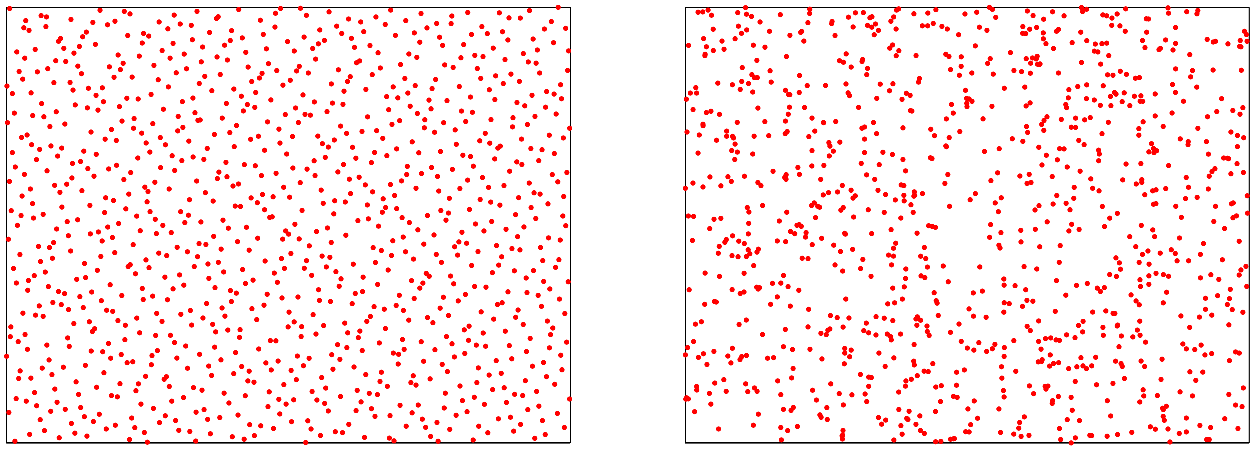
A competing method would be a randomized protocol. A randomized protocol will take the same list and randomly choose neighbors to inform,

- If I know look at a list of neighbors
- Randomly inform a neighbor on the list
- Randomly inform another neighbor on the list
- Repeat until list is exhausted.

The advantage (or maybe disadvantage) of this method is that the path of information is not fixed. There may exist paths that lead to more neighbors being informed in given timesteps, allowing the information to spread across the graph more quickly, and there might be times where the protocol gets stuck due to it's random nature.

The solution is a quasi random rumor spreading protocol. A quasi random protocol works by combining both defining factors from a strict list based spread and a completely random spread.

- If I know look at a list of neighbors
- (1) Randomly inform a neighbor on the list
- (2) Inform next neighbor on list.
- Repeat (2) until list is exhausted.



Quasi Random vs Pseudo Random

Alternative Quasi Random Spread :

- If I know look at a list of neighbors
- Quasi Randomly inform a neighbor on the list
- Quasi Randomly inform another neighbor on the list
- Repeat until list is exhausted.

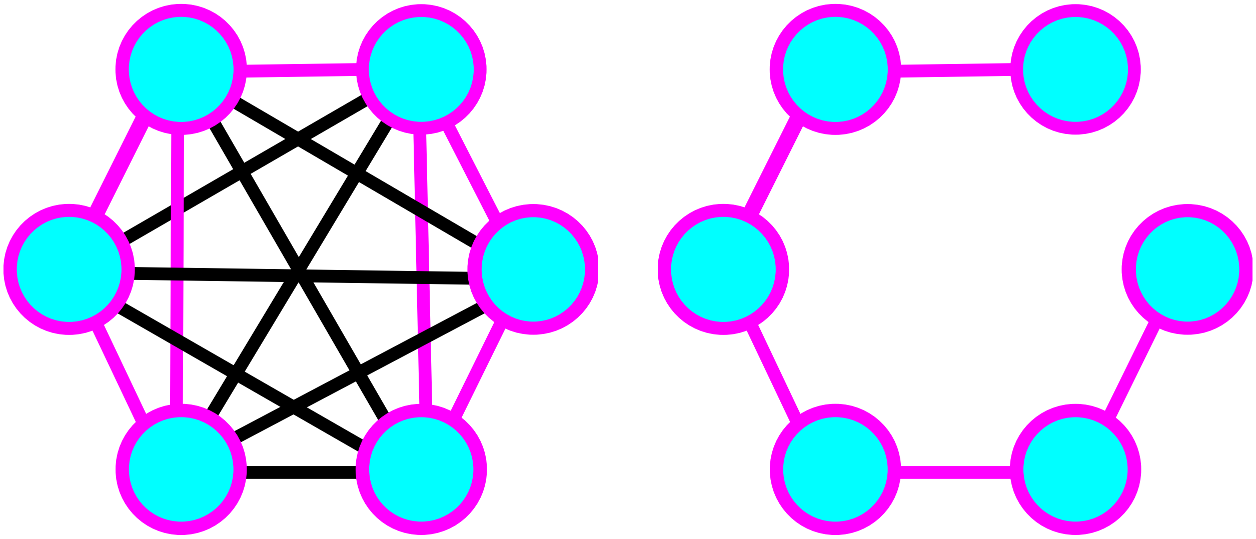
This will result in a many potential routes for information spread on a given network. Some may be more efficient and some less so. This is all before transmission failure is implemented.

When transmission failure is considered varying routes and limitations of the different protocols become more important. With the basic list method a transmission failure of 50% means that if a given node fails to receive the message in must wait for the protocol to iterate through the entire list again before there is another chance for informing. On a randomized protocol it could be reinformed on any timestep. In the quasi random protocol it must again wait for the entire list to iterate.

One method for finding a lower bound for the number of iterations of a quasi random protocol is shown to be with p probability of transmission failure, $(1+\varepsilon)(\frac{1}{\log_2(1+p)}\log_2 n + \frac{1}{p}\ln n)$ iterations with probability $p = 1 - n^{\frac{-p\varepsilon}{40}}$ of being completely informed. [1]

A Tale of Two Graphs

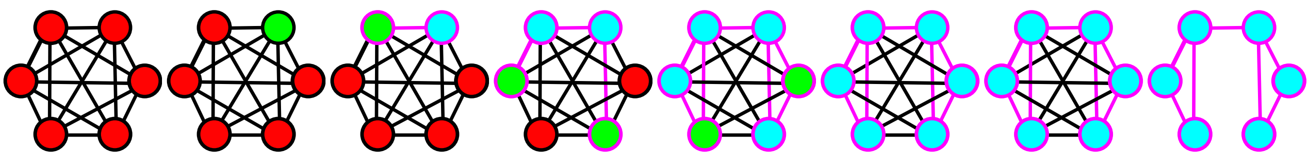
Numerically this can be shown to be an overestimation based on the edge distribution within a graph. The method focuses on vertex count, and the graphs used are complete. A complete graph has many more potential routes for information spread compared to a sparse graph.



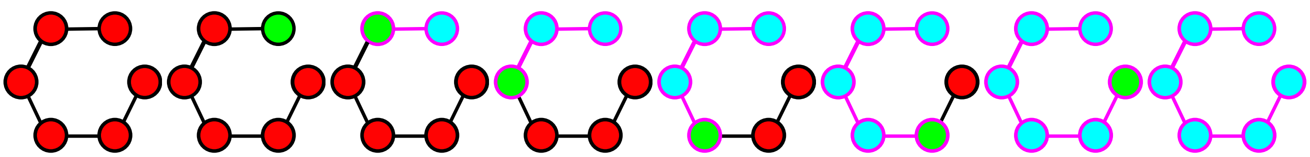
Spread on a Complete Graph vs Sparse Graph

The unused edges in the complete graph are black and stand as an obvious indicator that there is increased pathing available in a complete graph. Here is a visual representation of the "ideal" spread of information,

Complete Graph:



Sparse Graph:



Spread of a rumor across a complete and a sparse graph. Note that the complete graph finishes two rounds before the sparse graph in this example.

Visually, and using a graph small enough to watch all at once, it is clear that edge distribution plays a role in the spread of a information with a given protocol. Numerically this can be shown through simulation.

tl;dr

In short it will be shown (numerically) that the lower bounds set by Doerr-Huber-Lavavi [1] for a rumor to spread through a graph were higher than necessary in most cases. It will also be shown the distribution of edges on a graph is as important as the node count for setting an upper and lower bound for completion. Also considered is the impact of transmission failure and showing that given a certain probability, p, of transmission failure, the lower and upper bounds for completion are still as dependent on edge distribution as on node count.

Methodology

The Model

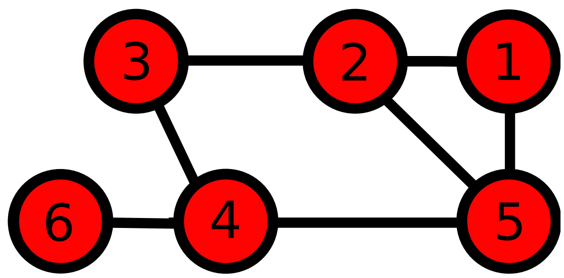
The model would be Agents sit on a graph, when they are informed start going through the protocol we're testing. We skipped right to the randomized protocol. We programmed everything with Matlab.

How it works (in Matlab)

We start with an adjacency matrix that looks something like this,

$$\begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \tag{1}$$

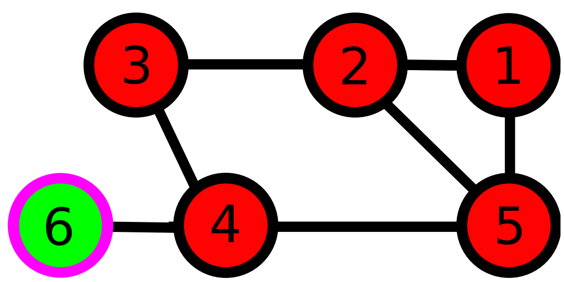
This describes a graph that looks like this,



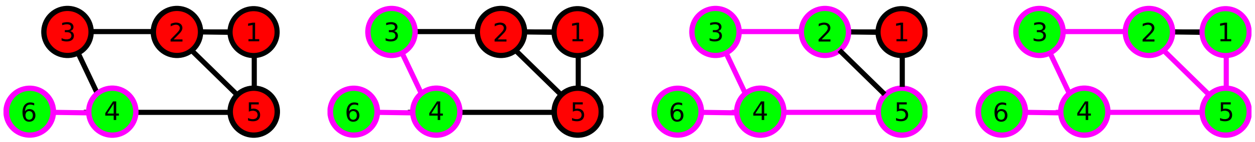
We build a second matrix which describes which node is informed,

$$\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \tag{2}$$

So here we have the 6 node informed.



Node six only has one choice, and lets assume 0% of transmission failure, so we see the rumor spread to 4, with the newly informed matrix representation below,



$$\begin{pmatrix}0\\0\\0\\1\\0\\1\end{pmatrix} \rightarrow \begin{pmatrix}0\\0\\1\\1\\0\\1\end{pmatrix} \rightarrow \begin{pmatrix}0\\1\\1\\1\\1\\1\end{pmatrix} \rightarrow \begin{pmatrix}1\\1\\1\\1\\1\\1\end{pmatrix}.$$

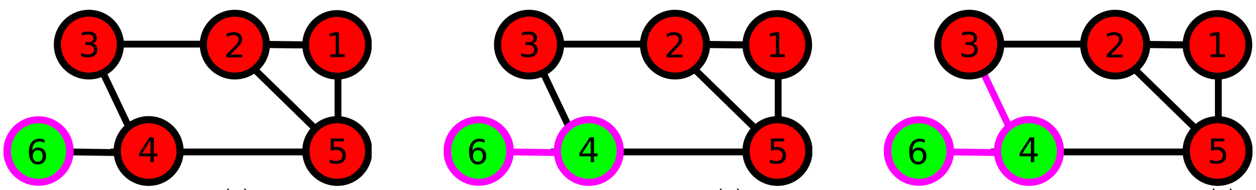
Note the only unused edge is $2 \rightarrow 1$

Until the entire graph is informed. Above is a simple best case where each newly informed doesn't back spread on any time step. Back spread is spreading the information to already informed nodes. This is a big cause of wastefulness. It is observed on the $2 \rightarrow 5$ interaction, but this is the last time step and leads to no waste overall.

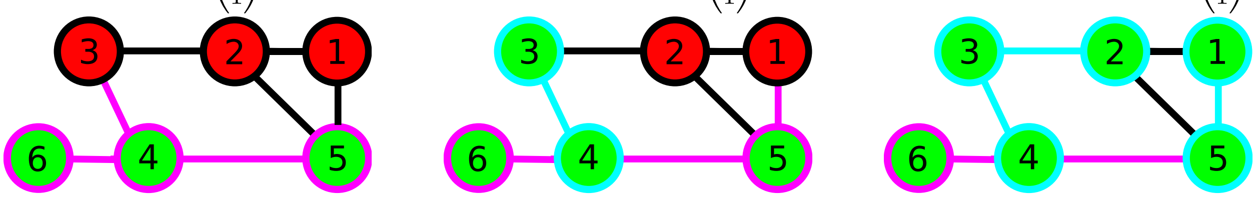
This basic simulation was repeated for larger graphs with varying interconnectivity. We started with N, the node count, at 10 and stepped this up logarithmically to $N = 1000$ testing the interconnectivity by percentage with step 10% and transmission failure up to 90%

The algorithm scales, and the complexity scales with the node count and edge distribution. Per time step computation, with N being the node count, starts at roughly $2N$ for a sparse graph and grows to N^2 for a complete graph. Better computational algorithms weren't explored, but an optimum edge distribution for varying transmission failure is explored.

Same as above with Transmission Failure



$$\begin{pmatrix}0\\0\\0\\1\\0\\1\end{pmatrix} \rightarrow \begin{pmatrix}0\\0\\0\\1\\0\\1\end{pmatrix} \rightarrow \begin{pmatrix}0\\0\\0\\1\\0\\1\end{pmatrix}$$



$$\begin{pmatrix}0\\0\\0\\1\\1\\1\end{pmatrix} \rightarrow \begin{pmatrix}0\\0\\1\\1\\1\\1\end{pmatrix} \rightarrow \begin{pmatrix}1\\1\\1\\1\\1\\1\end{pmatrix}.$$

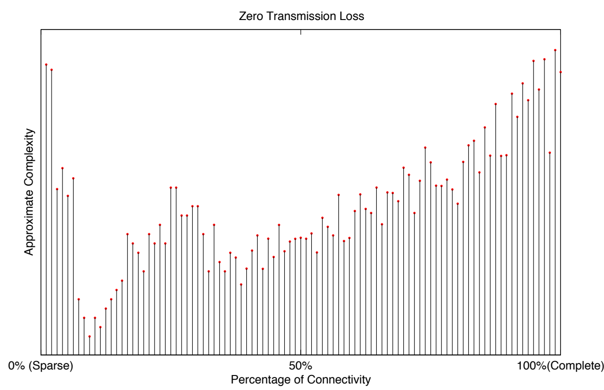
In the third image 4 fails to transmit to 3 and uses the next time step to transmit to 5. We're using blue here to demonstrate the path that involves a failure in the chain at some point. 4 successfully transmits to 3 on the 4th timestep with 5 failing to transmit to 1. It all works out in the end with the same number of over all times steps with 3 informing 2 and 5 informing 1.

Numerical Results

When node count, connectivity, and transmission failure are all accounted for you get a computationally optimum solution for connectivity with relation to the node count and the probability of transmission failure.

For example with a transmission failure of 0% you find the optimum degree of connectivity is approximately 13% but with a transmission failure rate of 90% you get an optimum connectivity of about 30%.

In english what I believe to be happening is that a sparse graph will have a lot of transmission repeating. A complete graph has less of a chance of repeating. With the operations counts above, and incorporating transmission failure, you're looking at an order of magnitude increase in the complexity with a transmission failure rate of 90% over zero transmission failure.



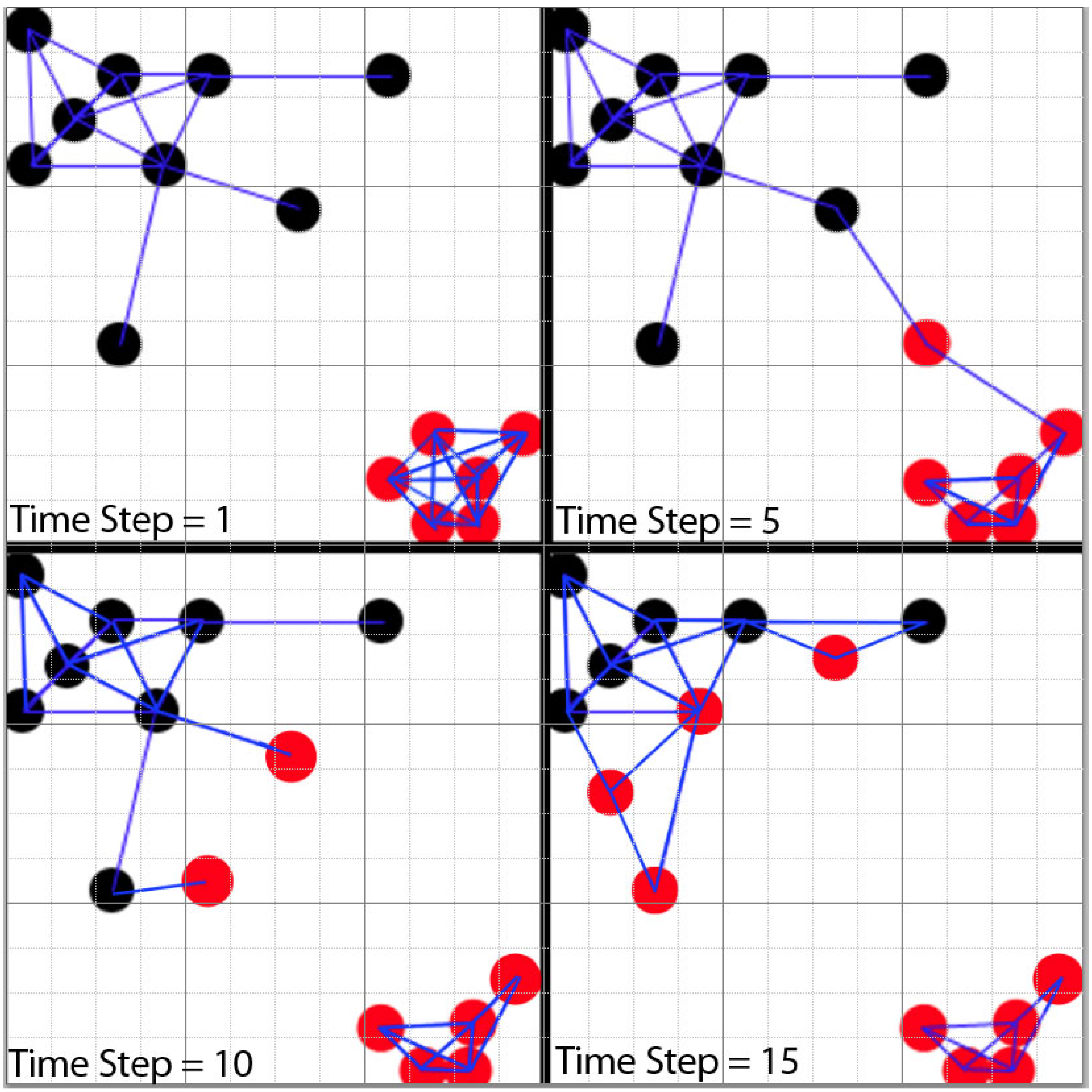
References

[1] Benjamin Doerr, Anna Huber, and Ariel Levavi. Strong robustness of randomized rumor spreading protocols. *CoRR*, abs/1001.3056, 2010.

Agent Based Rumor Spreading Model with Random Walks on a Grid

Another thing I did over the summer was play with agents moving about on a grid. The basic graph concept has a visual representation like this,

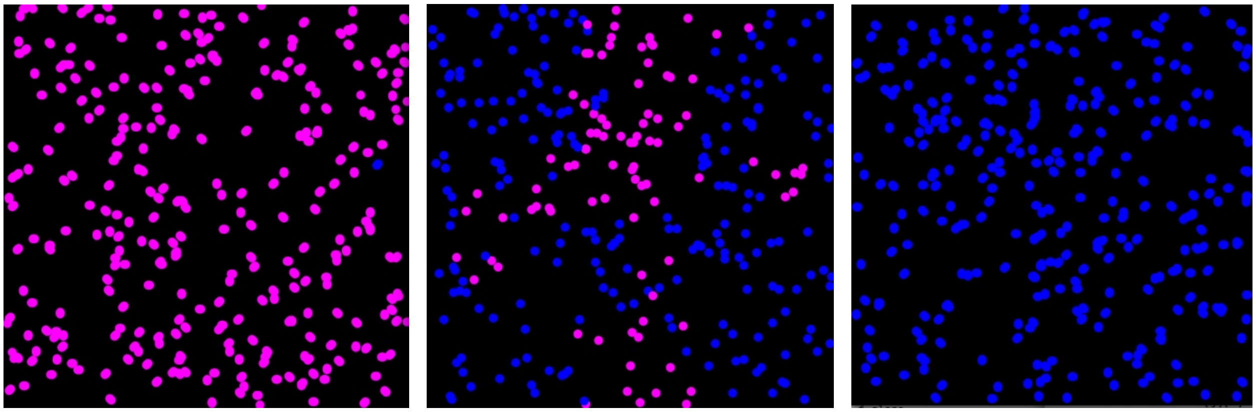
Within this model all agents were allowed to move unconstrained about a grid on a torus. Agent movement, dynamic graphing, and remote sensing were allowed to wrap on the edges. Allowing for dynamically formed graphs in an agent based model, agents have more chances to be informed as total informed increases.



The above is a simplified example of the actual process. All agents are moving randomly and form their connections based on proximity. When coming into proximity with other nodes an agent will use the same selection process as the stationary model; the agent will select one node in a quasi-random manner and pass on the rumor.

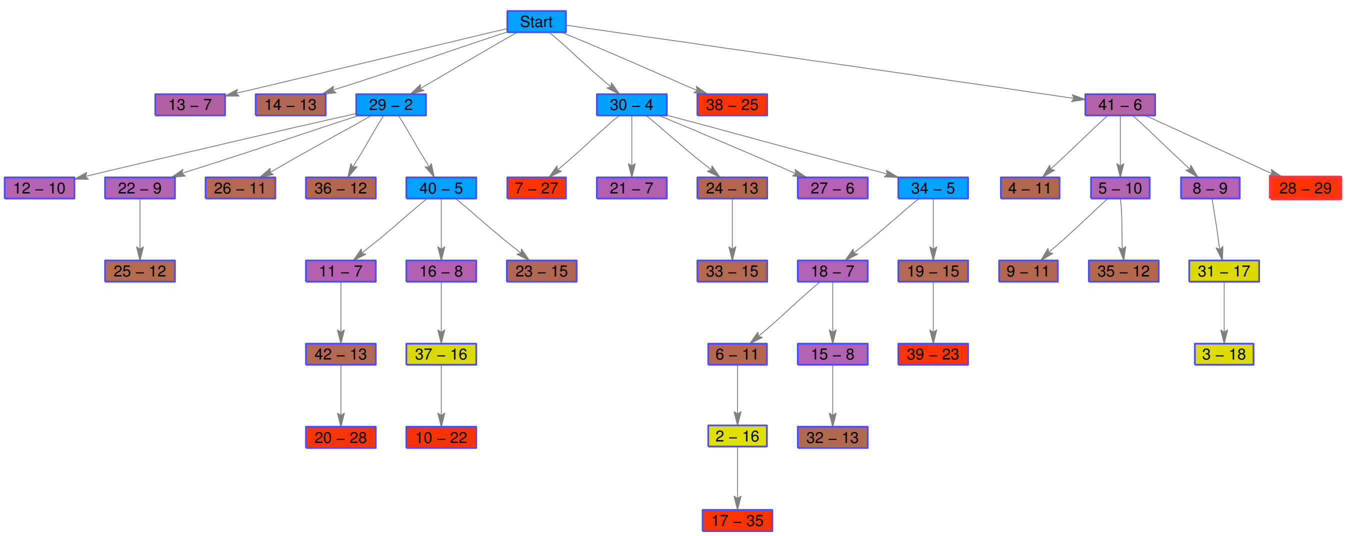
The agents are on a grid, move around, and form transient graphs used to transmit information when they are in range. The transient graphs are formed based on proximity.

Our transient agent model examines grouping and preconditions to make predictions about completion time. In addition we examine methods of releasing information in a manner which minimizes total spread time.



Progression of Transitory Agent Model

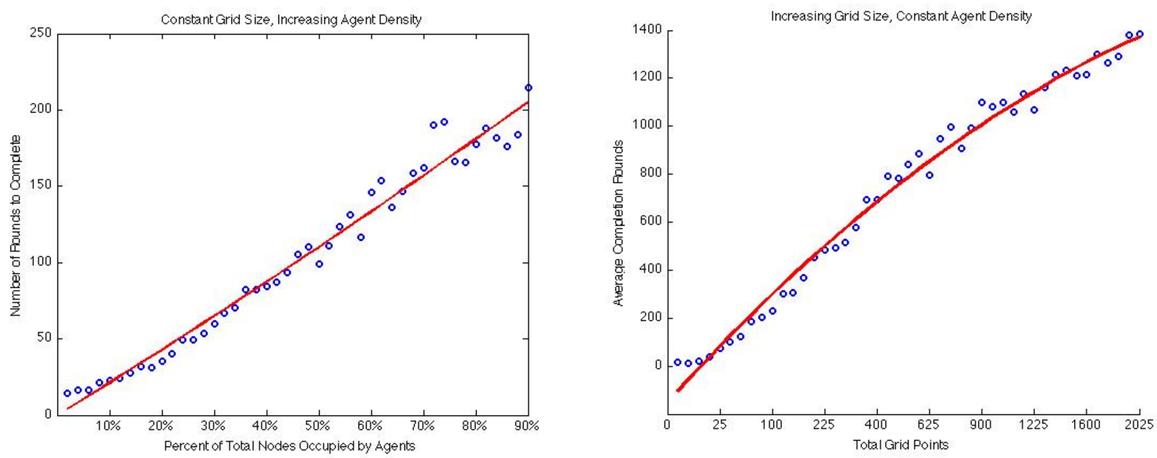
Currently we are tracking the path of a rumor through a transient network. The most important thing tracked at the moment is who informed whom, and at what time step.



Flow of Rumor through Agents by Number

The above graph shows the spread by time step and informer. There is very little apparent structure in the flow of information, so there seems to be little that can be done at that step to increase the completion rate and minimize computation.

We have performed some experiments related to grid size and agent density to get an idea of what happens when these vary.



Increasing Density, and Increasing Grid Size

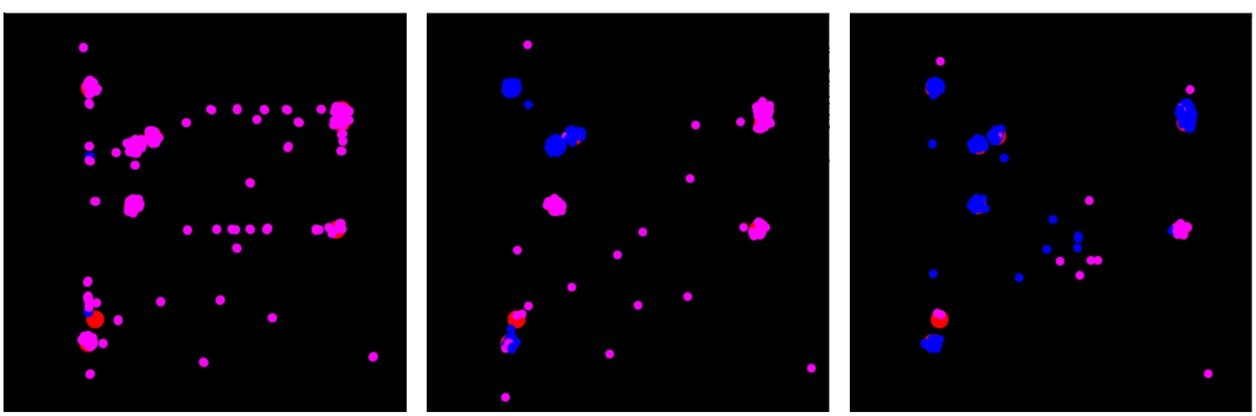
The results so far are pretty basic. When the number of agents for a given grid is increased the increase in completion time is linear. When the grid size is increased and the number of agents or agent density is kept static, the increase in time is logarithmic.

Future Work

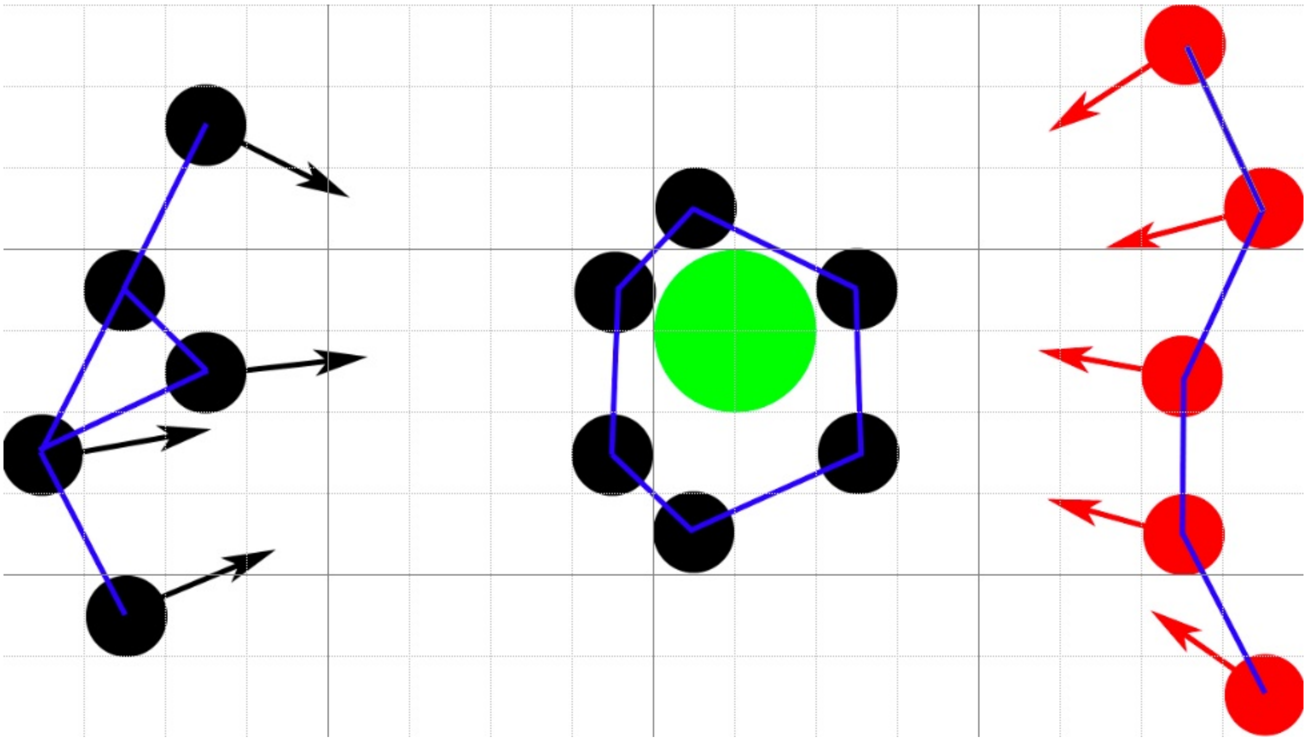
Currently, we are working on agent structure and attractors. We'd also like to change how the agents interact with their environment by implementing impassible regions. Agent behavior will be altered in the following ways:

- Aggressive Spreading - Agents behave as predators forcibly hunting uninformed agents
- Random Walks with Memory - Give agents memory to spread without backtracking
- Genetic Algorithms - Agents optimize for navigation in restricted access grids

Attractors (shown above and on next page) are important in simulation. They simulate public areas where people gather in close proximity and facilitate disease and information spreading.



Demonstration of selective attraction



Details on attractor functionality

This shows an attractor (green) and some infected (or informed) approaching, and some uninfected (uninformed) approaching as well, and some uninfected in a stable arrangement around the attractor. In close proximity to an attractor free movement is allowed, but when an agent is sets an attractor as it's destination they proceed as directly as possible to the attractor.