Exploring Telephone

Michael Ernst

May 9, 2008

Background

Telephone is a classic children's party game. The primary goal of the game is to start with a rather innocuous phrase and eventually turn it into something completely ridiculous. What is particularly intriguing is how this process occurs. All of the participants sit in a circle. One of them starts the game off by whispering a phrase into the ear of one of the people next to them. This second person then continues the process by whispering the phrase to the next person in the circle. This continues until the phrase travels all of the way around to the original individual. Whispering is typically only done once; there is no chance for clarification. At this point it is highly unlikely for the phrase that person hears to be the phrase that they used to begin the game. On the surface, this may seem unusual. Each person is passing on the phrase and it does not necessarily make sense for it to change. However, there are at least two factors that cause these changes. There are legitimate mistakes caused by the difficulty of somebody whispering in your ear and so this can cause problems. Individuals can be forced to guess at what they think they heard. Also, people often intentionally change the phrase to make it more ridiculous or amusing. The important thing to remember is that for whatever reason the phrase can often be changed at many points around the circle. As a result, the hilarity of the game is completely dependent upon the unreliability of communicating the phrase.

Motivation

On the surface the game of telephone is a rather frivolous game (as it should be); however, the basic idea behind it is very intriguing. The central theme of telephone is the corruption of information as it passes through an unreliable network. As mentioned above this is one of the most fundamental components of the game. Also, aside from this fact the structure of the game is very straightforward; there are no major facets that do not serve to facilitate this one purpose. After realizing these important features of the game I became interested in how telephone could be used to look at how information flows through an unreliable network. However, the game itself appears rather straightforward and initially appears uninteresting to model. On the surface, it would appear that the network structure in telephone is very basic and ultimately uninteresting. Given my past experience with graph theory it is this feature that I found the most intriguing.

My primary idea was to take the game of telephone and generalize it to a variety of different network structures. I became curious as to how a game of telephone would unfold if it were played on a more complex network structure, as opposed to simply using a cycle. However, before being able to understand more complex networks it was important for me to determine the effects of all of the basic factors on the original simple network structure. As it turns out, the different effects that one can explore simply by considering only the basic variables already available in the original game of telephone are varied and interesting in themselves. As a result, a significant amount of my time and effort was spent analyzing these different effects. This means I ultimately did all of my research without exploring different network structures. As mentioned above, this is not particularly problematic because the game of telephone already has interesting things to say about communication and reliability without any structural generalizations.

Model

Set-Up

It begins with a directed path from a start node to a destination node. A random bit-string of a chosen length is generated and given to the start node. Also, all of the nodes in the graph are assigned some reliability value between 0 and 1.

Iterate

Each node takes any strings it possesses and passes them along to the next node in the path. There is a chance that the node will mutate the string in some way as it passes it. Periodically, nodes are forced to resolve all of the strings they possess into a single proposed correct string. This resolution is guaranteed to occur after the last step.

Output

The Hamming distance between every nodes final string and the original string is computed to determine how much information is preserved and where in the path it is preserved. Hamming distance measures the number of bits of disagreement between two bit-strings.

Based on the above structure these simulations have a variety of important parameters:

pathLength	The number of intervening nodes between the first and last node on every path between them. (Note that all paths will have the same length)
length	The length of the initial string being passed around.
reliabilities	The list of node reliabilities.
resolveSteps	The number of iterations after which each node is forced to resolve the strings that it contains.
numLoops	The number of resolveSteps that occur in a full run of the simulation on a graph. (This is 1)

(Note: The total number of iterations is resolveSteps*numLoops)

Method of Mutation

The method of mutation I ultimately settled on allows for a great degree of variability. For this type of mutation the reliability actually represents the possibility of mutation for every given bit in the string. This means that bits have a chance of mutating independent of one otherwise, a chance that is the complement of the transmitting node’s reliability.

Method of Resolution

I used a straightforward method of resolution. The majority value at any given bit-location is taken to be the correct value. If there is a tie then it is broken in favor of '1' over '0.'

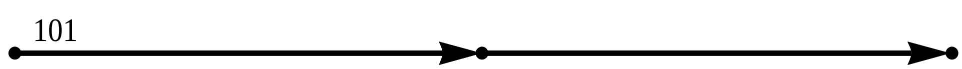
Assumptions

This model carries a couple very important assumptions. First, the length of the message should not change, only its inner constitution. This is a very simplifying assumption because it rules out the possibility of insertion or deletion in the message; there is only the chance of corruption. Also, nodes accept messages as correct irregardless of the node that is passing it to them. This is to say that nodes are blind to each others reliabilities and so trust the most reliable as much as the least reliable. Finally, the method of mutation also has an inherent bias in it. It assumes that all components of the message are equally likely to be corrupted and also that their chances of corruption are entirely independent of one another. These assumptions do simplify the situation, but most of the important form of the game is maintained.

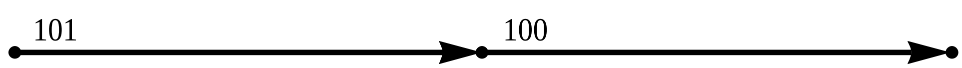
Example:

This example follows the transmission of the bit-string '101,' through a basic length three path. resolveSteps is 3 and so nodes will resolve their messages after every 3 iterations. numLoops is 1 so this will occur exactly once. That is, there will be three iterations, followed by a single resolving step.

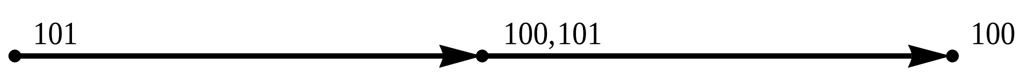
Initially the start node is seeded with the beginning message:



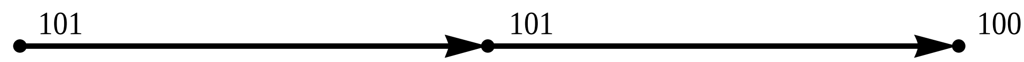
In the first iteration the message is transmitted to the second node. However, the third bit has mutated and so '101' is transmitted as '100':



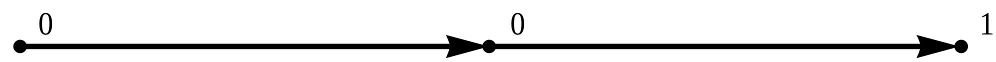
In the second iteration the first node passes its message to the second, this time without mutation, and the second node passes it message to the third, again without mutation:



Since resolveSteps iterations have completed the nodes resolve their contained messages. Only the second node contains multiple messages and these are resolved to '101.' This is because both messages agreed on the first two bits, and the tie-breaker in the third bit is broken in favor of ‘1’:



Finally, since numLoops is 1 the run is completed and the Hamming distance from the original message is computed at all nodes:

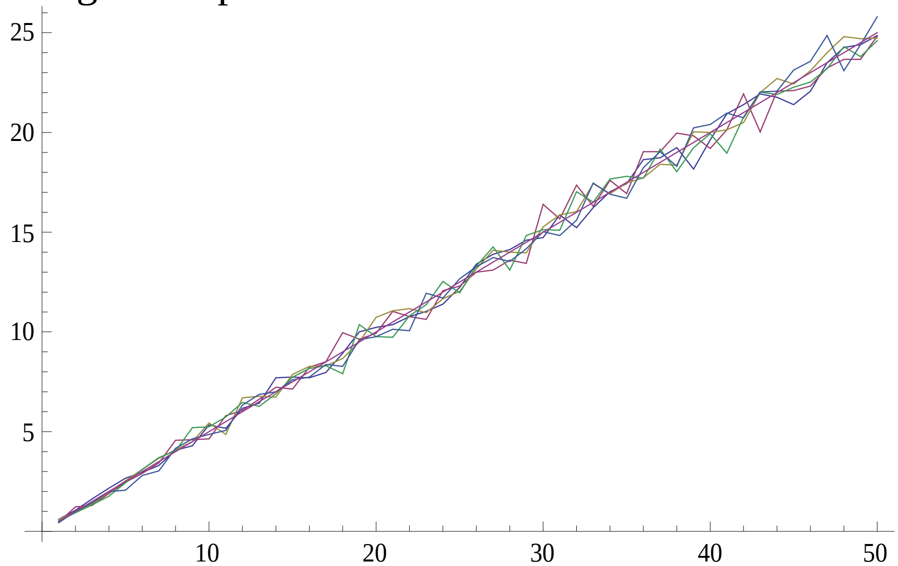


Results

In order to get a solid understanding of the game of telephone I ran a significant number of simulations in order to generate a significant amount of data. I attempted to explore much of the space related to three of the primary variables: pathLength varied from 7 to 11, the length of the bit-string varied from 1 to 50, and for all of those combinations I did 30 runs for each of the reliabilities: .1, .25, .5, .75, and .9. Note that every node is the graph had the same reliability for every simulation. In all of these cases I did just enough iterations for the message to reach the destination node at the end of the path. The only piece of information ultimately gathered was the Hamming distance of that final message received from the original message, which, as you recall, is randomly generated in each case. Also, there was exactly one resolution step in each simulation, though this is unimportant to the data because the final node only ever received a single message. Finally, for each of the 30 runs with identical variables I averaged these final Hamming values to get more consistent data.

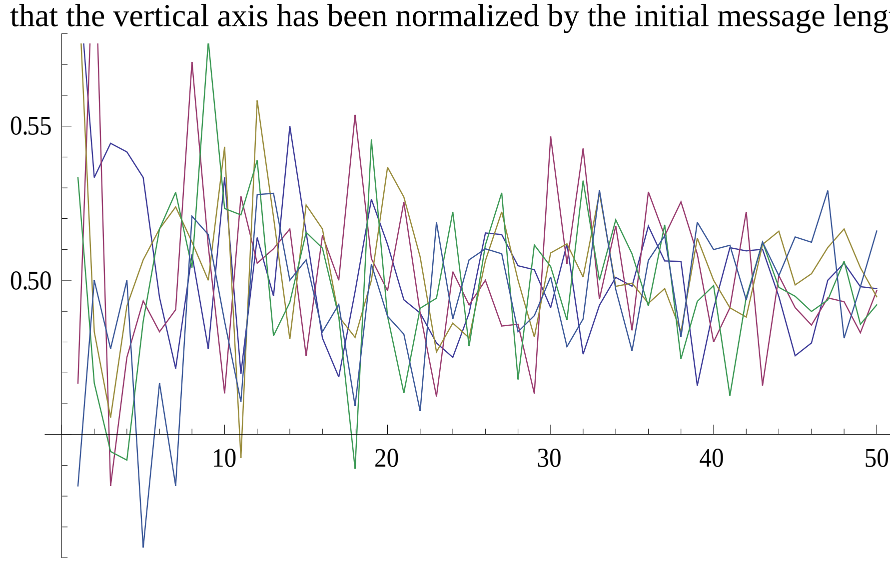
Given all of these simulations there are a variety of different results, which each expresses an aspect of the game of telephone.

Length Independence:



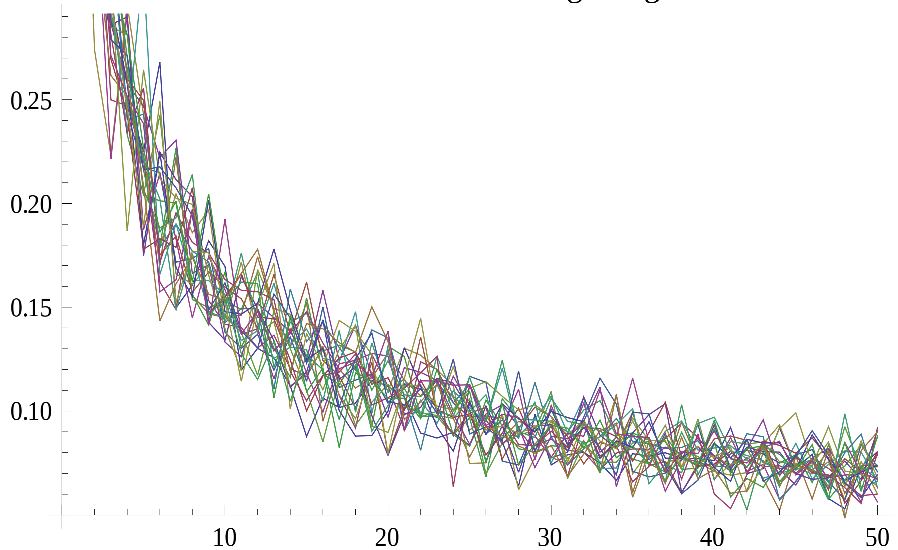
This is a plot of all the .5 reliability simulation means. The horizontal axis is the length of the original message and the vertical axis is the final Hamming distance. Each line represents the same path length as the length of the initial message is increased. Lower lines are better because they represent a smaller final Hamming distance.

The plot illustrates a couple of important features of the game. First, all of the graphs obey a linear trend. This means that there is a fixed proportion between the initial message length and the final Hamming distance and so when you normalize by the initial message length all of the lines are horizontal. Hence, the proportional correctness of the message received at the destination node is independent of the length of the messages being passed. This is clear in the following plot, which is the same data as above, except that the vertical axis has been normalized by the initial message length.



The second important feature is that a reliability of .5 behaves nearly identically at all path lengths. This makes intuitive sense because that level of reliability means that every bit is just as likely to change as it is to stay the same. As a result, every transmission is effectively mutating into a completely random new message that is independent of what was received. This in turn establishes an important baseline for the worst possible reliability. The worst that can be done is generating an entirely random new message, which should agree on half of the bits. As is clear from the normalized plot, this is effectively the behavior of a reliability of .5 in just about all possible conditions. As a result, a message is already effectively meaningless once it drops to 50% correctness because it contains no more information than a randomly generate one.

However, simply because the mean of the results is independent of the initial message length does not mean that the initial message length has no effect. Shown below is a plot of the standard deviations of the normalized means for all of the runs. Again, each line represents a consistent set of graphical structures and reliability values experiencing an increasing initial message length. The vertical axis is the standard deviation and the horizontal is initial message length.

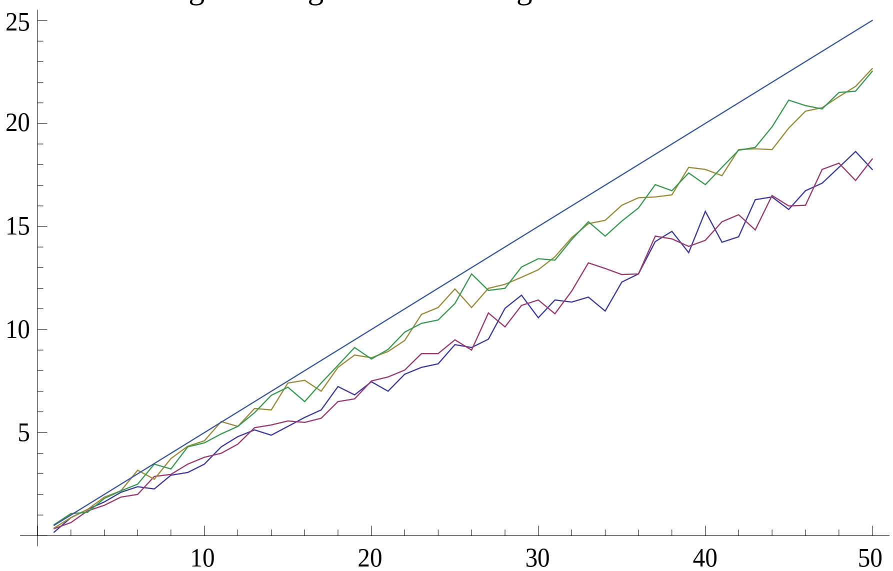


The trend of this plot is very clear. As the length of the initial message increases, the variation between simulations of the same environment decrease. This means that while different arrangements can have different overall reliabilities, they can also have different levels of consistency depending upon the size of that first message. The reason that this is the case is that as the length of the messages increase the mutations are much more likely to follow the statistically suggested path. Longer messages are less subject to random variation because there are some many bits that are subject to possible mutation.

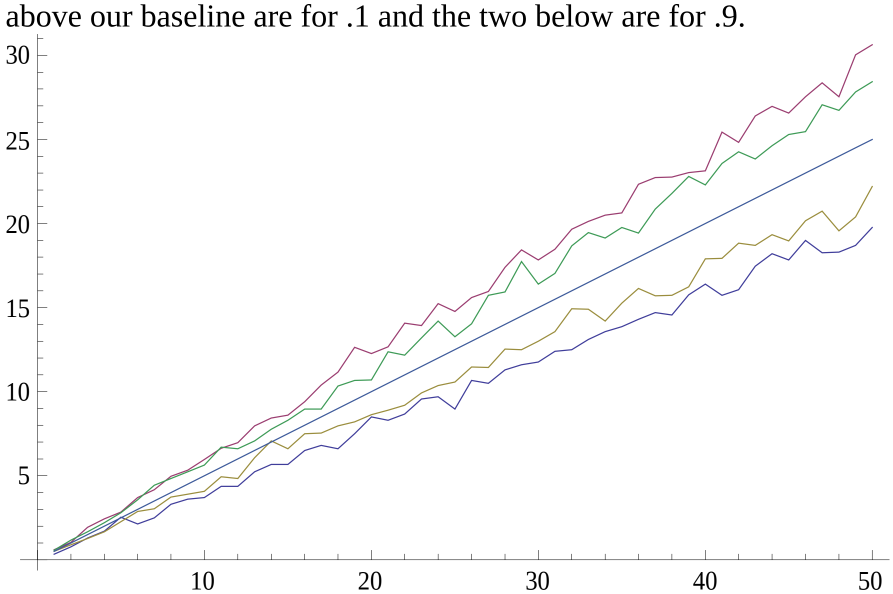
Thus, there are two primary features of message length. The length of messages does not actually affect the mean reliability of a given arrangement. However, it does affect the variation in that mean, so for more consistent behavior you need longer messages.

Symmetry of Complements:

Another important relationship has to do with complementary reliabilities. The plot below illustrates the behavior of reliabilities of .1 and .9 on paths of length 7 and length 11. The straight line is just a line with a slope of .5 passing through the origin; this represents the 50% correct baseline. Any lines below that represent results that are more effective than generating a random string.



The red and green lines are the two data-sets of path length 11, while the blue and red lines are the data for path length 7. What is interesting here is that the reliabilities of .1 and .9 behave nearly identically in both cases. This is generally true for any pair of complementary reliabilities on any paths of odd length. In contrast, the plot below illustrates similar data, except for path lengths of 6 and 8. In this case, the two lines above our baseline are for .1 and the two below are for .9.

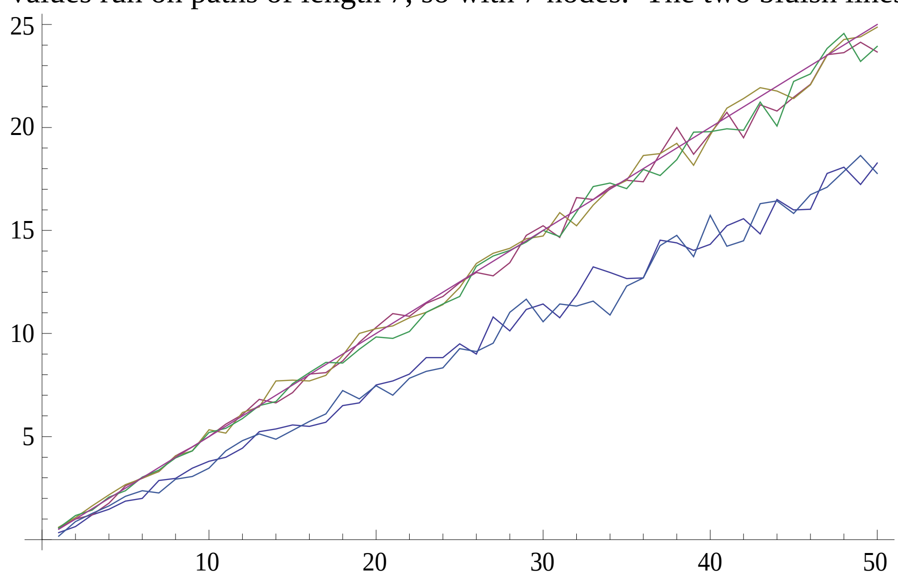


The easiest way to understand why this is the case is to consider a reliability of 0. In that case every single bit in the message flips at every step. That means that after an odd number of steps the message should be wrong at every bit. This would take an even number of nodes because the number of steps is one less than the number of nodes since we are dealing with paths and not cycles. After any even number of steps all of the bits would agree perfectly with the original message. In cases where there are an odd number of nodes there is an even number of steps. This means that in those cases reliability 0 and reliability 1 behave identically. When the number of nodes is odd, reliability 0 looks exactly like reliability 1 reflected over our reference line with slope .5.

Thus, any reliability value of .5 or below behaves identically to its complement when there are an odd number of nodes. When there is an even number of nodes, its behavior is identical to its complement after reflection over our reference line. Hence, if one only considers paths of odd number the space of possible reliabilities is cut in half.

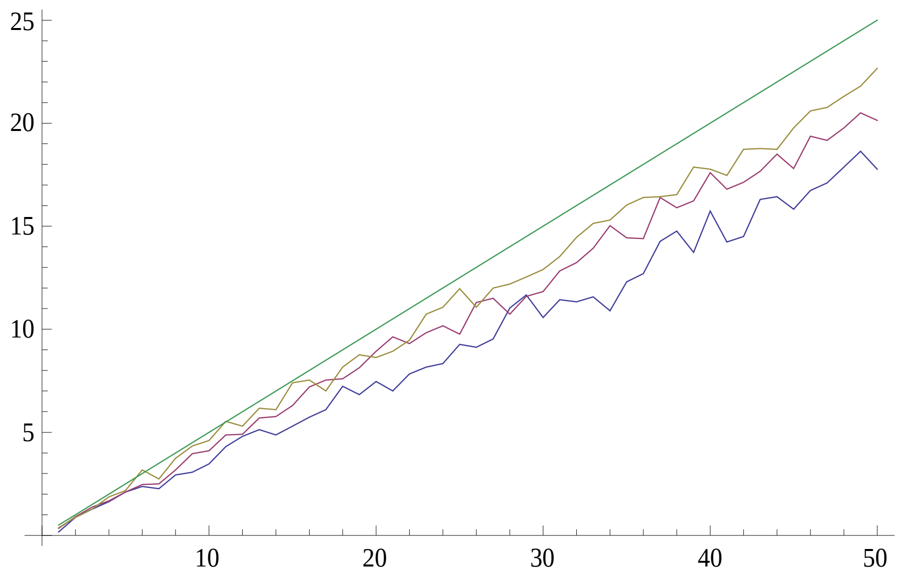
General Effectiveness:

Finally there are a few general intuitive trends to consider as far as the impact of probability and path length. The plot below shows the lines for all of the reliability values run on paths of length 7, so with 7 nodes. The two bluish lines are .1 and .9.

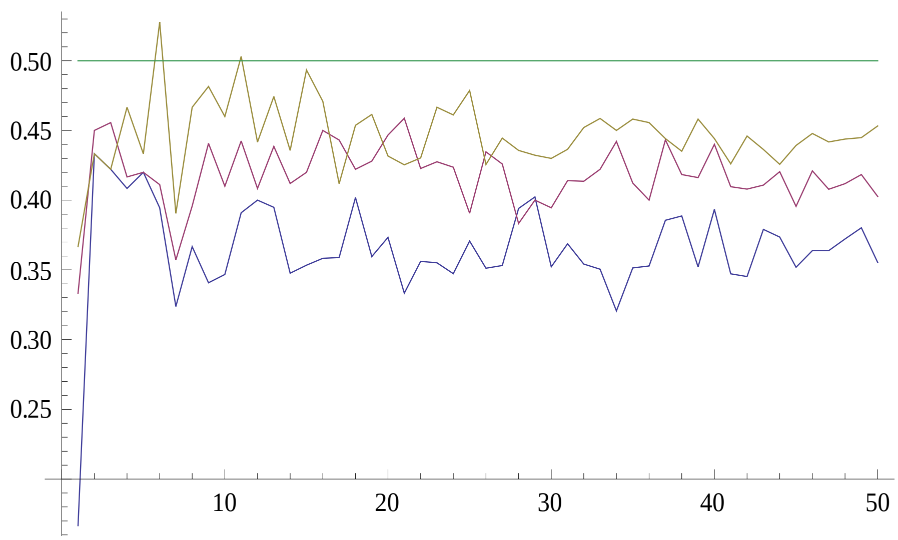


If you recall from above, the lower the line the more effective that arrangement has been at preserving information. Here it is clear that the .1 and .9 reliabilities have been the most effective. What this means is that higher reliabilities are more effective at preserving information than lower probabilities, assuming they are all above .5. Given what we've established about complements we can consider .1 and .9 equivalent in this case. Hence, higher node reliability implies higher arrangement reliability, which is intuitively what we'd expect.

Another important factor is the effect of path length on overall arrangement reliability. Below is a plot of .9 reliability simulations taken at length 7, 9, and 11. The lines representing the longer paths are above those representing shorter paths. Again our reference line is included.



This plot makes clear that as the path length increases the overall reliability decreases because the lines are shifting upwards. This is even clearer in the normalized version of that plot, shown below.



Longer paths create less reliable arrangements. This behavior is pretty much exactly as expected simply because longer paths allow for greater degradation of the message. The longer the path the more iterations required to reach the finish; the more iterations there are in the simulation the more likely the message is to undergo negative mutation until it reaches the same level as randomness.

Conclusion

Though it may be a simple children's game, telephone contains an amazing degree of complexity. On the one hand, there are interesting unexpected characteristics of the game; specifically the behavior of complementary reliabilities. On the other hand, the system often behaves exactly as expected; higher reliabilities are more effective, longer paths are less effective. Also, the game allows for a wide variety of extensions.

Aside from the content given above I also explored more briefly the use of multiple paths, with different reliabilities, between the start and destination. I also explored different mutation methods and how they affected the system. Resolution strategies and timing were incorporated into my code and some of my simulations, but was ultimately removed from my final study because it added too much complexity. Also, for resolution to be meaningful it requires more complex graphical structures, which I did not explore. The true testament to the intricate and interesting nature of the game of telephone is that even simple structures produce interesting results; there is no necessity to move to more complicated structures. I set out to do very intricate things on very involved graphs and ultimately found that the behavior in the simplest cases is complex enough to warrant a significant amount of study.