Sensitivity Analysis of Graph Distance Measures

Julian Sorensen, Peter Dickinson and Martin Schubert
Defence Science and Technology Organization – Intelligence, Surveillance and Reconnaissance Division.

Abstract

The performance management of computer networks is becoming increasingly important given the dynamic nature of traffic on these networks. There exists a number of graph similarity measures for network monitoring and abnormal change detection. It is necessary to quantify and compare the performance of these measures, against known types of abnormal network behaviour, to assess their suitability for use in a variety of network monitoring activities. We present a preliminary study of the performance of graph distance measures using simulated network traffic.

1. Introduction

Many networks (for example computer, social) are dynamic in nature and hence change in some way over time. An important issue when considering such a network is the identification of points in time at which the change in the network is anomalous given the apparent behaviour of the network up to those points respectively. This raises the obvious question: how does one determine that a change at a particular point is anomalous?

A network can be considered as a (discrete) time series of graphs. For example, a computer network can be considered as a time series of directed, weighted graphs, where each graph represents the network over a particular interval of time. More specifically, the directed edges for a graph correspond to the existence of communication from one node to another node in the time interval represented by the graph. Similarly, the weight assigned to an edge in a graph could correspond to the amount of traffic that has travelled on that edge, or correspond to the duration of communication on that edge over the interval of time represented by the graph.

The problem of determining points of anomalous change in a network can be viewed as the identification of graphs that are anomalous in the corresponding time series of graphs representing it. The graph distance measures found in Shoubridge, Kraetzl, Wallis and Bunke (2002) provide an approach to this view of the problem, which is as follows. The graph distance measures quantify the difference between two graphs. So by applying a graph distance measure sequentially to each pair of consecutive graphs in the time series of graphs, one can track the change occurring in the network over time, and ideally identify points of anomalous change.

This report investigates graph distance measures found in Shoubridge, Kraetzl, Wallis and Bunke (2002) , looking at the following issues:

1. Given a particular type of anomalous change, which measure will best detect it?

2. How severe does an anomalous change need to be in order to be detected by a graph measure?
3. The influence of individual nodes on a distance measure, in particular nodes of a high degree.

The report is an investigation of these issues using simulated data. We designed a simulator that generates a time series of graphs whereby the dynamics controlling the graphs is homogeneous over time. The simulator was also designed so that the user can inject one of a variety of anomalous changes at a point in the time series of graphs. For example, one anomalous change is that of increasing the degree of a specified node at a particular point of time. The magnitude of change can also be chosen. For example, the degree that a specified node is increased by can be chosen.

2. The Simulator

The network simulator generates a discrete time series of directed, weighted graphs, which at a certain point in time, injects an anomalous change that may or may not be permanent. Hence the network simulator can be thought of consisting of two parts, with the first part simulating a network, in the form of a time series of graphs, that undergoes no anomalous changes. The second part can be thought of as overlaying an anomalous change to the time series of graphs generated by the first part. This breakdown of the network simulator into two seemingly disjoint parts is not entirely accurate, but it is convenient for the following more detailed description of the network simulator. The types of network behaviour being simulated are those that would be observed at the logical level.

2.1 First Part of Simulator

As mentioned, the first part of the simulator can be thought of simulating a network that undergoes no anomalous changes. Now, the mechanism generating the network is not highly sophisticated, nor necessarily realistic. However, it does incorporate some characteristics of real networks, which we now explain.

While many real networks, for example computer and social, are dynamic in nature, there normally exists a number of nodes that communicate with each other frequently. Throughout this document we will refer to links arising from such communications as *core* edges of the network. Conversely, links between pairs of nodes that communicate infrequently will be called *non-core* edges. Our simulator incorporates this idea of a network consisting of core and non-core parts. This is best explained by describing how the first part of the simulator works, which is as follows.

The initial graph is generated in the following way. Firstly, the number of possible nodes is set at N. Throughout the course of this report, N = 150. Note that these will be the only nodes over the course of the whole simulation. Also, no self loops are allowed, that is, a node A say, is not allowed to be connected to itself, only to different nodes. Now, out of the N^2 possible edges, a proportion α are randomly chosen as edges for the initial graph. Throughout the course of the simulations α was set to 8%. The edges chosen are designated to be our *core edges*. Conversely, the edges not chosen are designated as our *non-core edges*. Now, before describing how weights are assigned to the edges, we explain how the edges are chosen for the remaining 99 graphs that are generated.

Given a graph G_{n-1}, the edges for the next graph in G_n are chosen according to the following conditional probabilities:

1. P(core edge exists in G_n | core edge exists in G_{n-1})
2. P(core edge exists in G_n | core edge does not exists in G_{n-1})
3. P(non-core edge exists in G_n | non-core edge exists in G_{n-1})
4. P(non-core edge exists in G_n | non-core edge does not exists in G_{n-1})

By setting the first two probabilities high and the last two probabilities low, the network is encouraged to exhibit the property of having the *core* and *non-core* parts we described earlier. This is in fact what we did. Throughout the course of the simulations, these probabilities were set at 0.85, 0.6, 0.3 and 0.0001, respectively.

The weights for the edges are chosen at random according to the Poisson distribution. The simulator allows for the core and non-core edges to have different Poisson rates. Now, there are two reasons why we chose the Poisson distribution for the edge weights. Firstly, the ease by which to do it. Secondly, while there is much debate on the real distribution of traffic for many types of networks, the Poisson is a commonly used distribution for traffic in other contexts, see Albin (1982) and Melamed (1979) for example.

2.2 Second Part of Simulator

As mentioned, the second part of the simulator can be thought of as overlaying an anomalous change to the time series of graphs generated by the first part. The changes are intended to mimic possible events that could occur in real networks, and that would be of interest to a network administrator. For each simulation, a single change was applied to the 60th graph in the time series of graphs. Some of the changes involve varying the properties of a subset of nodes and the links between, which we call the *Logical Community of Interest* (LCI). For programming convenience, the LCI was located in the top left hand corner of the adjacency matrices representing the graphs generated.

The following anomalous changes were implemented:

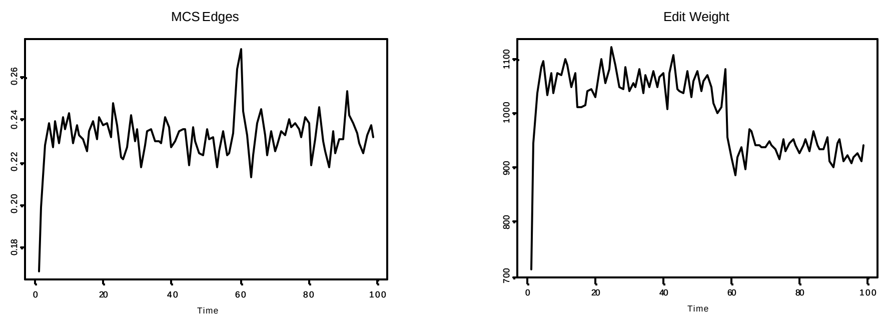
1. **Change in degree of a specified node for one graph.** The degree of node 1 for the 60th graph is increased or decreased. This change is not permanent - for graph 61, the edges involving node 1 are governed again by their conditional probabilities. Setting the degree of the node much higher than it would normally be, is an analogous to a node broadcasting a message, or a new website coming online.
2. **Removal of the edges from the LCI for one graph.** All edges within the LCI are removed for the 60th graph. This is analogous to a group of users who abruptly cease communication with each other for a brief period of time.
3. **Change in traffic rate for an entire graph.** The traffic rate for the edges of the 60th graph is set to a common rate. The traffic rates for edges are reset to their original values after the 60th graph. This is analogous to a special event causing temporary increased/decreased traffic across an entire network.
4. **Temporary change in the communication pattern for the LCI.** A proportion of the edges in the LCI are shuffled in the following way. Consider the LCI edges that do and do not exist for graph 59. For graph 60, a proportion of the edges that did exist in graph 59 are chosen not to exist and a proportion of edges that did not exist in graph 59 are chosen to exist. For graph 61, the edges in the LCI revert back to what they were in graph 59. This is analogous to a change in the structure of communication patterns within some sort of clique.

5. **Change in traffic rate for edges within LCI.** The traffic rate for all edges within the LCI for the 60th graph and all graphs following, is set to a common rate. This is analogous to special event causing increased/decreased traffic among a community of users.
6. **Removal of a path from network temporarily**. A constant path is overlayed on every graph except for the 60th graph.
7. **Removal of a path from network permanently**. A constant path is overlayed graphs 1 to 59.

3. Results

The magnitude for each type of change is controlled by some parameters. For example, the parameter controlling change 7 is the length of the path overlayed on the graphs. Hence for each type of change, the simulator ran over a range of values for the controlling parameters. In addition, the simulator was run at least 4 times for each parameter value chosen to be used. Now, for each time series of graphs generated, the graph distance measures were applied sequentially to each pair of consecutive graphs making up that time series of graphs. This resulted in a time series of numbers for each graph distance measure applied to the time series of graphs.

Figure 1: Some examples of the impact of a change



Removal of path temporarily

Change in traffic rate for LCI

If the changes impacted the time series of numbers, the impact was manifested in either of two ways. The most common way was that the 59th and/or 60th points were abnormally large compared to surrounding numbers. The other less common way was that of producing a change in mean of the time series at the 59th point. Figure 1 illustrates both of these impacts. It is important to note that for a particular combination of measure and change, the change when manifested, always impacted the corresponding time series of numbers in the same way. Now, for time series where change was pronounced, the impact of the change was obvious by visual inspection. However, for time series with less conspicuous changes, a visual inspection was inadequate to judge if change had impacted the time series. Hence we used some statistical tests, which we explain in the following paragraphs.

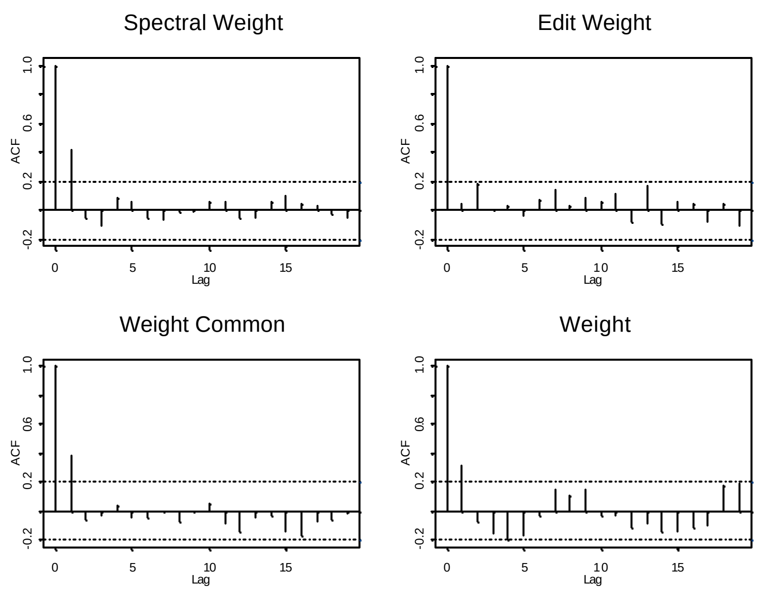
Underlying the statistical tests we used, is the assumption that the time series of numbers are stationary, moving average order processes up to order 2. The assumption of stationarity is based on the fact that the evolution, as determined by the simulator, from one graph to the next, is homogeneous over time (apart from the point at which the anomalous change is injected). It seems reasonable to expect that a measure quantifying this evolution will inherit the homogeneity. We base the assumption that the time series are moving average order processes up to order 2, on the observation that for almost all the time series generated, the sample auto-correlation is zero after lag 1 or 2. See for example Figure 2.

The problem of determining whether the 59th and/or 60th points are abnormally large was phrased in terms of the following two hypotheses:

- H0: neither the 59th or 60th points are additive outliers for a moving average process of order 2.
- H1: at least one of the 59th and 60th points is an additive outlier.

We used a test from Bruce and Martin (1989) to retain or reject the null hypothesis.

Figure 2: Some sample auto correlation functions



The problem of determining if a change in mean has occurred at the 59th point was phrased in terms of the following two hypotheses:

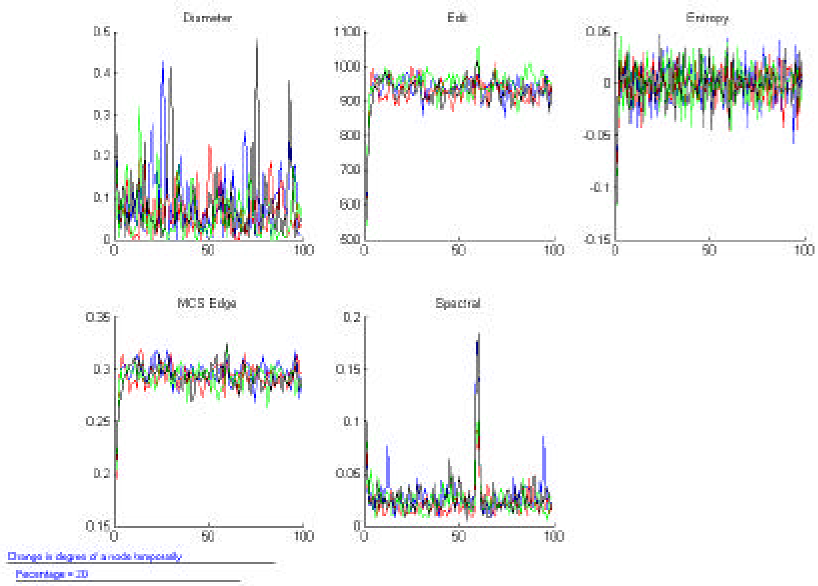
- H0: time series is a moving average order process of order 2.
- H1: at the 59th point, a change in mean occurs, where the mean is allowed to be a polynomial of order 0, 1, 2 or 3.

H0 was modelled as a moving average process of order 2. H1 was modelled as a moving average process plus trend, where the trend was allowed to be a polynomial of order 0, 1, 2 or 3. Out of the five models generated by the two hypotheses, the model with the smallest BIC criterion was chosen. See Schwartz (1979) for BIC criterion. If the model chosen didn't correspond to H0, then H0 was rejected. Otherwise it was retained.

3.1 Change in Degree of a Specified Node for one Graph

We compared the performances of the following measures: Diameter, Edit, Entropy, MCS Edges and Spectral. For all of these measures, the change impacted the time series by making the 59^{th} and/or 60^{th} points appear abnormally large.

Figure 3: Change in degree of node



Let β be the percentage of nodes, out of the total number of nodes making up the graphs, that node 1 is forced to be linked to. The values that β took were 10%, 20%, 30%, 40%, 50% and 60%. For each value of β, the simulator was run 4 times. See Figure 3 for β = 20%. Table 1 provides the number of time series for which the statistical tests detected the change.

Table 1: Change in degree of a node

Measures	Percentage						Detection Type
	10%	20%	30%	40%	50%	60%	
Diameter	0	0	0	1	1	0	Outlier
Edit	0	2	3	4	4	4	Outlier
Entropy	0	0	1	3	4	4	Outlier
MCS Edges	0	2	4	4	4	4	Outlier
Spectral	0	4	4	4	4	4	Outlier

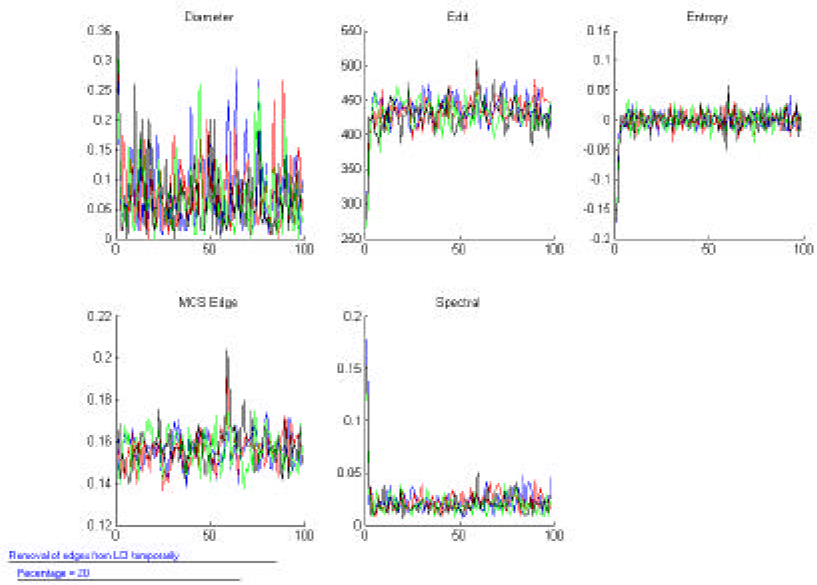
To summarize, we found that:

- For all measures tested, except Diameter, the change is detected for β with values 40% and higher.
- Edit and MCS Edges perform similarly.
- The performance of Spectral was superior to the others – it reliably detected the change for β = 20%, which no other measures did. Also, the peak corresponding to the change was the most pronounced for Spectral.

3.2 Removal of Edges from LCI for one Graph

We compared the performances of the following measures: Diameter, Edit, Entropy, MCS Edges and Spectral. For all of these measures, the change impacted the time series by making the 59th and/or 60th points appear abnormally large.

Figure 4: Removal of edges from LCI



Let β be the percentage of nodes out of the total number of nodes belong to the LCI. The values that β took were 20%, 30%, 40% and 50%. For each value of β, the simulator was run 4 times. See Figure 4 for β = 20%. Table 2 provides the number of time series for which the statistical tests detected the change.

Table 2: Removal of Edges from LCI

Measures	Percentage				Detection Type
	20%	30%	40%	50%	
Diameter	0	2	4	4	Outlier
Edit	2	3	4	4	Outlier
Entropy	2	4	4	4	Outlier
MCS Edges	2	4	4	4	Outlier
Spectral	1	2	4	4	Outlier

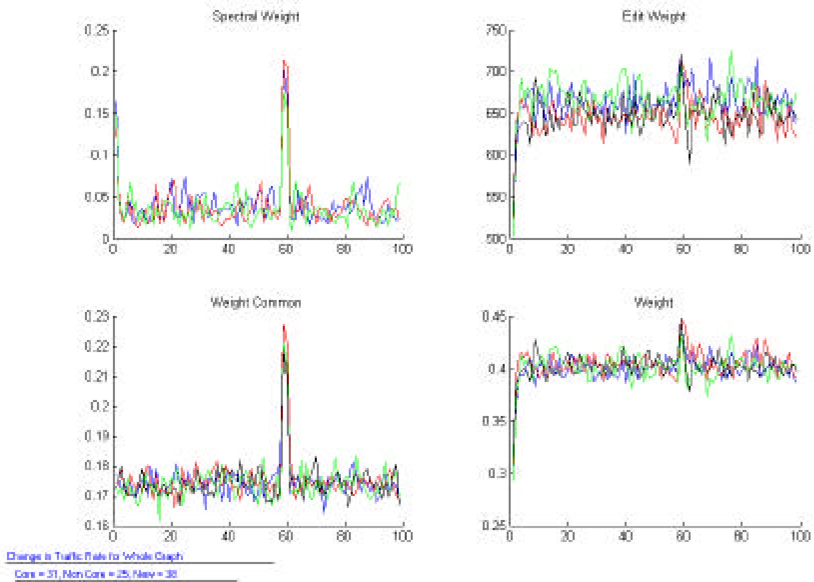
To summarize, we found that:

- For all measures, the change was detected for values of β greater than 30%.
- Entropy and MCS Edges performed the best, with performance of edit distance being almost as good.
- The change was not visually detectable for Spectral and Diameter when β took values 20% and 30%.

3.3 Change in Traffic Rate for an Entire Graph

We compared the performances of the following measures: Edit Weight, Spectral Weight, Weight Common and Weight. For all of these measures, the change impacted the time series by making the 59^{th} and/or 60^{th} points appear abnormally large.

Figure 5: Change in traffic rate for entire graph



Let γ_{core} and $\gamma_{non\text{-}core}$ be the Poisson rates for the core and non-core edges before the change respectively. Let γ_{new} be the special Poisson rate for all edges of the 60^{th} graph. A number of combinations of γ_{core}, $\gamma_{non\text{-}core}$ and γ_{new}, from the numbers 13, 19, 25, 31, 50, 63, 68 and 75 were used. These numbers were chosen as they are proportions of 25 and 50, for example 13/25 $\cong$?/2 and 63/50 $\cong$ 5/4. For each combination, the simulator was run 4 times.

Table 3: Change in Traffic Rate for Whole Graph

Measures	Traffic Rates		# Detected	Detection Type
Edit Weight	Core Rate	25	1	Outlier
Spectral Weight	Non Core Rate	13	4	Outlier
Weight Common	New Rate	31	4	Outlier
Weight			2	Outlier
Edit Weight	Core Rate	25	0	Outlier
Spectral Weight	Non Core Rate	25	4	Outlier
Weight Common	New Rate	31	4	Outlier
Weight			1	Outlier
Edit Weight	Core Rate	31	2	Outlier
Spectral Weight	Non Core Rate	25	4	Outlier
Weight Common	New Rate	25	4	Outlier
Weight			4	Outlier
Edit Weight	Core Rate	31	2	Outlier
Spectral Weight	Non Core Rate	25	4	Outlier
Weight Common	New Rate	38	4	Outlier
Weight			4	Outlier

In general, we found that:

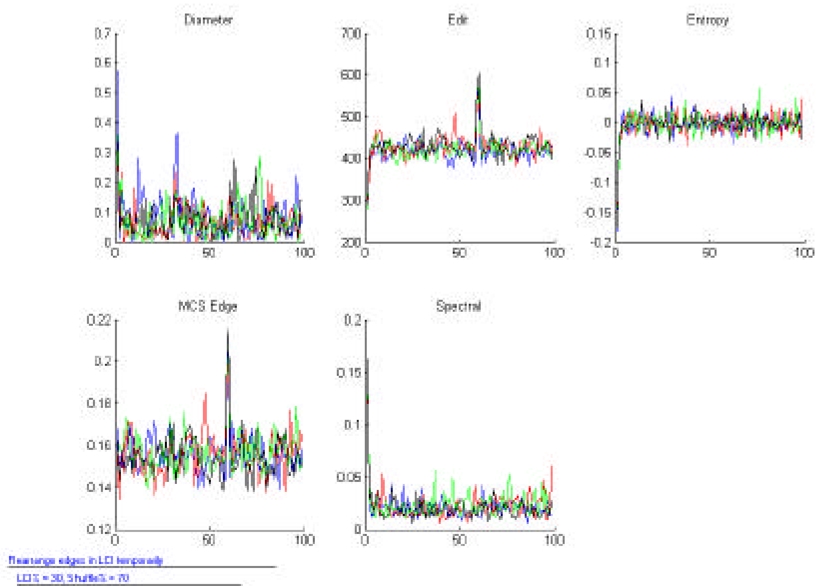
- Most of the changes were detected by all of the measures tested.
- Spectral Weight and Weight Common performed the best, detecting the change in almost all cases.
- Edit Weight performed the worst.

Table 3 highlights the cases where the change in traffic rate for a given graph is not detected well using Edit Weight and Weight. Figure 5 shows results when the traffic rate is set to 38 at the change point.

3.4 Temporary Change in the Communication Pattern for the LCI

We compared the performances of the following measures: Diameter, Edit, Entropy, MCS Edges and Spectral. For all of these measures, the change impacted the time series by making the 59th and/or 60th points appear abnormally large.

Figure 6: Temporary change in communication pattern for LCI



Let β_{LCI} be the percentage of nodes out of the total number of nodes belonging to the LCI. We allowed β_{LCI} to take values 20%, 30% and 40%. Let $\beta_{Shuffle}$ be the percentage of the edges in the LCI that are shuffled. We allowed $\beta_{Shuffle}$ to take values 50%, 70% and 90%. The simulator was run for all possible combinations of β_{LCI} and $\beta_{Shuffle}$. For each combination, the simulator was run 4 times. See Figure 6 for β_{LCI} = 30% and $\beta_{Shuffle}$ = 70%. Table 4 provides the number of time series for which the statistical tests detected the change.

To summarize, we found that:

- Diameter, Entropy and Spectral failed to detect this change in almost all instances.
- Edit and MCS Edges detected this change in almost all instances.

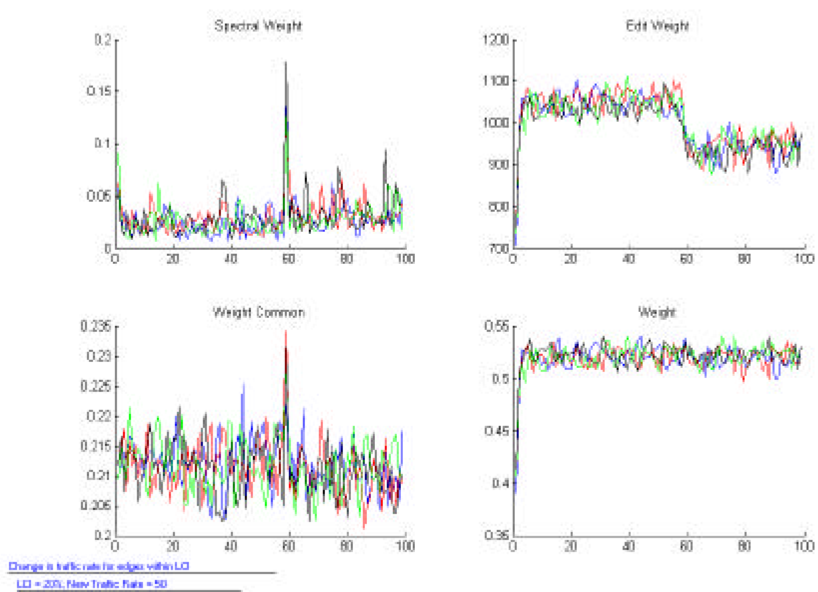
Table 4: Change in Communication Pattern for LCI

Measures	LCI%	Shuffle%			Detection Type
		50%	70%	90%	
Diameter	20	0	0	0	Outlier
Edit		3	4	4	Outlier
Entropy		0	0	0	Outlier
MCS Edges		2	2	3	Outlier
Spectral		0	0	0	Outlier
Diameter	30	0	0	1	Outlier
Edit		4	4	4	Outlier
Entropy		0	0	0	Outlier
MCS Edges		4	4	4	Outlier
Spectral		0	0	0	Outlier
Diameter	40	0	0	0	Outlier
Edit		4	4	4	Outlier
Entropy		0	0	0	Outlier
MCS Edges		4	4	4	Outlier
Spectral		0	0	0	Outlier

3.5 Change in Traffic Rate for Edges within LCI

We compared the performances of the following measures: Edit Weight, Spectral Weight, Weight Common and Weight. For Spectral Weight, Weight Common and Weight, the change impacted the time series by making the 59th and/or 60th points appear abnormally large. For Edit Weight, the change caused a change in mean at the 59th point.

Figure 7: Change in traffic rate for LCI



Let β_{LCI} be the percentage of nodes out of the total number of nodes belonging to the LCI. We allowed β_{LCI} to take values 10%, 15% and 20%. Let γ_{new} be the new Poisson rate for the edges making up the LCI after the 59th graph. We allowed γ_{new} to take values

30, 40, 50 and 60. The simulator was run for all possible combinations of β_{LCI} and γ_{new}. For each combination, the simulator was run 4 times. See Figure 7 for β_{LCI} = 20% and γ_{new} = 50%. Table 5 provides the number of time series for which the statistical tests detected the change.

Table 5: Change in Traffic Rate for LCI

Measures	LCI Size												Detection Type
	10%				15%				20%				
	Traffic Rate				Traffic Rate				Traffic Rate				
	30	40	50	60	30	40	50	60	30	40	50	60	
Edit Weight	4	4	4	4	4	4	4	4	4	4	4	4	Mean
Spectral Weight	0	1	0	2	0	2	3	3	1	4	4	4	Outlier
Weight Common	0	0	0	1	0	1	0	2	0	2	3	4	Outlier
Weight	0	0	0	0	0	0	0	0	0	0	0	1	Outlier

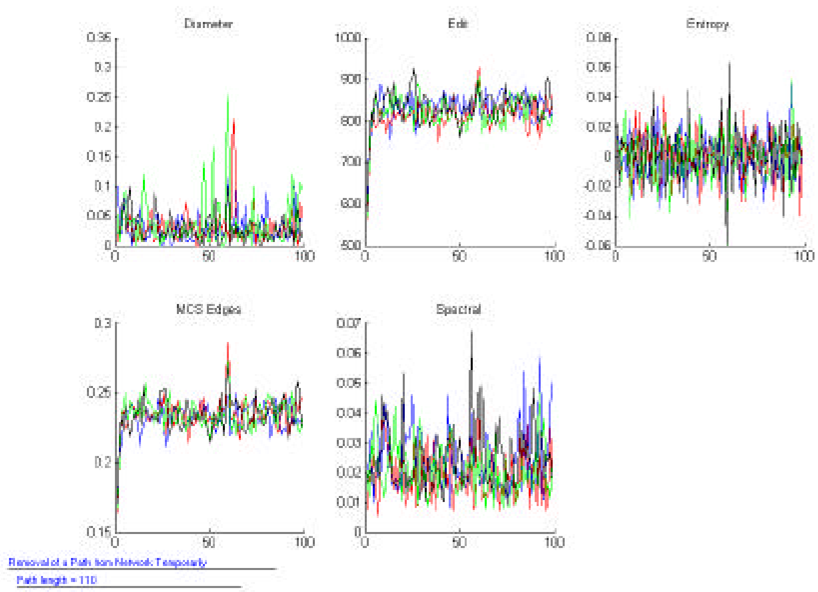
To summarize, we found that:

- Edit Weight by far outperformed the other measures.
- Weight performed by far the worst, failing to detect the change in all cases.
- Spectral Weight performed the best after Edit Weight.

3.6 Removal of a Path from Network Temporarily

We compared the performances of the following measures: Diameter, Edit, Entropy, MCS Edges and Spectral. For all of these measures, the change impacted the time series by making the 59th and/or 60th points appear abnormally large.

Figure 8: Removal of path temporarily



Let β be the number of nodes making up the path. The values β took were 110, 120 and 130. For each value of β, the simulator was run 4 times. See Figure 8 for the path length

110. Table 6 provides the number of time series for which the statistical tests detected the change.

To summarize, we found that:

- MCS Edges performed the best.
- The other measures struggled to detected this change.

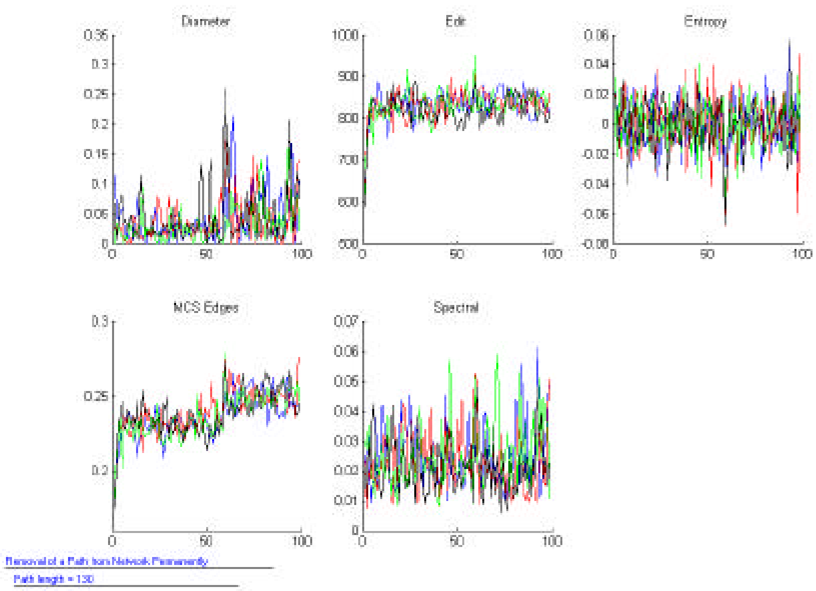
Table 6: Removal of a Path Temporarily

Measures	Path Length			Detection Type
	110	120	130	
Diameter	3	1	1	Outlier
Edit	1	2	2	Outlier
Entropy	1	2	2	Outlier
MCS Edges	3	4	4	Outlier
Spectral	0	1	2	Outlier

3.7 Removal of a path from Network Permanently

We compared the performances of the following measures: Diameter, Edit, Entropy, MCS Edges and Spectral. For Diameter, Edit, Entropy and Spectral, the change impacted the time series by making the 59^{th} and/or 60^{th} points appear abnormally large. For MCS Edges, the change caused a change in mean at the 59^{th} point.

Figure 9: Removal of a path permanently



Let β be the number of nodes making up the path. The values β took were 110, 120, 130 and 140. For each value of β, the simulator was run 4 times. See Figure 9 for the path length 130. Table 7 provides the number of time series for which the statistical test detected the change.

To summarize, we found that:

- MCS Edges performed the best, followed closely by Entropy.
- Diameter and Spectral struggled to detect this change.

Table 7: Removal of a path permanently

Measures	Path Length				Detection T
	110	120	130	140	
Diameter	1	0	1	1	Outlier
Edit	0	3	1	4	Outlier
Entropy	3	4	4	4	Outlier
MCS Edges	4	4	4	4	Mean
Spectral	1	1	1	0	Outlier

4. Summary of Simulation Results

It is clear from the seven cases considered, the performances of different measures varied: Spectral was superior for case 1; Entropy and MCS Edges were superior for case 2; Spectral Weight and Weight Common were superior for case 3; Edit and MCS Edges were superior for case 4; Edit Weight was superior for case 5; MCS Edges was superior for case 6; MCS Edges was superior for case 7.

Another finding is that for most cases, there were measures that failed to detect the change (at least over the range of magnitude the changes occurred): Diameter failed to detect most changes it was given; Diameter, Entropy and Spectral failed to detect change 4; Weight failed to detect change 5.

The simulations also provided an idea of what magnitude a change must be in order to be detected by the various measure. For example, Spectral reliably detected change 1 when the degree of node 1 was 20% of the number of number of nodes making up the graph, whereas MCS Edges required this percentage to be 30%.

References

Albin, S.L. (1982) Poisson Approximations for Superposition Arrival Processes in Queues. *Management Science* 28, no. 2, 126-137.

Bruce, A.G. and Martin, R.D. (1989) Leave-k-out diagnostics for time series. *J. R. Statist. Soc. B* 51, 363-424.

Melamed, B. (1979) On Poisson Traffic Processes in Discrete State Markovian Systems with Applications to Queueing Theory. *Adv. Applied Probability* 11, 218-239.

Schwartz, G. (1979) Estimating the dimension of a model. *Annals of Statistics* 6, 461-464.

Shoubridge, P.J., Kraetzl, M., Wallis, W.D. and Bunke, H. (2002) Detection of Abnormal Change in a Time Series of Graphs. *Journal of Interconnection Networks* 3, 85-101.