Approved E. Upfal
2/15/2001

Monte Carlo Simulation of United States Power Network with Faulty Nodes and Fail Propagation

master project report

Ying Hu

Instructor: Eli Upfal

Apr 10, 2000

Abstract

Stochastic network is a very broad and challenging subject. There are many different kinds of stochastic networks. This master project is aimed at one specific kind of stochastic network: a network where each node can fail and the failure of each node can propagate to neighborhood nodes. As an extension of this kind of network, a network where each edge can fail and the failure of each edge can propagate to neighborhood edges will also be studied. Monte Carlo simulation technique will be the technique to study this kind of network. A software package has been written for further study.

1. Introduction

We have entered and have always been in a network era. Realizing it or not, there are all sorts of network around us such as Internet, power station network, water flow network and so on. Among all those networks, a huge amount of them share a common feature of possible node failure and failure propagation to neighbors. Routers could get jammed by hackers or simply by too many unintentional data packages and there is a high chance that your neighbor routers be jammed too. The goal of this project is to use Monte Carlo simulation technique to simulate such phenomenon in order to obtain some useful information so that it can be used to compare with results of other more advanced study. Since there are so many networks around us, try to simulate all of them will be impossible and unnecessary. A typical network representation of such networks, United States power station network will be used to fulfil our goal based on several reasons. First of all U.S. power network is a relatively well-defined network compared with other networks such as Internet. Secondly U.S. power station network is more static, we won't build a power station every day, while to Internet, many new nodes appear each day. Thirdly it is relatively simpler than some other networks, there aren't so many nodes in the network, small number of nodes makes visualization possible. The final reason is that there is some financial application with the simulation of U.S. power station networks. There are many contractors who want to buy power from power stations and sell power from consumers. If each power station can generate the desired power and each consumer restricted himself to assigned power and the number of consumers stay fixed, then the problem degenerates to a static network problem. However, things are not always so simple and perfect. There is always a chance that each power station can fail or

consumers can ask for more power or there is simply higher population such as current power crisis in C.A. or even more unpredictably, nature disasters can shut down power plants or connection wires. Contractors need to worry about these problems in order to make it more profitable. One feature of such problem is that when a power plant fails, for example, caused by higher demands or nature disasters, there is a high chance that some neighborhood power plants will fail too. When one power plant fails due to high demand, consumers will turn to neighbor power plants and make them fail too. Nature disasters can happen in a broad area, not just a specific area. Meanwhile, if a linkage between power plants or power plants and consumers fail, neighbor linkages can fail too. This report and the software package is used for this specific kind of networks with emphasis on U.S. power plant network. A detailed topology of U.S. power plant network is also available, which makes the study easier and more direct. This report uses Monte Carlo simulation technique to do the study. The result of the simulation will be used to compare with some common sense in order to make sure there is bug in the software package. The choice of Monte Carlo simulation technique is mainly because of the stochastic feature of the problem itself and the simulation technique.

2. Problem Definition

Even if we have restrained our simulation object as Unites States power station network, there are still so many things to study and so many variables to control. In order to study this problem, we need to have a clear definition of the problem. Certain assumption and simplification will be made. Some assumption and simplification are not

to reduce the complexity of the problem, they are introduced so that we can have a clear and simple sample to compare.

Problem:

Given a network, for each node, there is a probability that it will fail. After the node fails, it will have certain propagation probability that each of its neighbors will fail, the failed neighbors will again propagate the failure to their neighbors, the process will continue until there are no more newly failed nodes. During this process, if one node fails, and one of its neighbors has already failed, it will not propagate the failure to that specific neighbor since there is no sense to do so. This is as same as saying a failed node will only propagate its failure once, otherwise, the process can go forever. After the propagation process stops, we will study the property of the network, such as how many nodes failed, how many edges failed, how many paths failed (A path is a route from one node to another node. The difference between path and edge is that an edge connects two adjacent nodes and a path can connect two nodes that are not adjacent. A path is defined as failed if any node on the path failed), what is the probability that one node failed and so on. In particular we will pay extensive amount of attention to the following problem. We choose a set of nodes as producer set P, another set of nodes as consumer set C (In this problem, consumer set C is also chosen from nodes represented by power plants in the network, the reason is that it is very difficult to define consumers, thus mathematically, both producer set P and consumer set C are composed of nodes from power plants). We also require $|P| > |C|$ to form a bipartite graph B, where each edge in the B represents a path from the link's source node (a node in P) to the link's destination

node (a node in C), the edge is failed if the path it represents is failed after the propagation process, we will consider the bipartite graph B' which is derived from B after removing all failed edges, and we want to know what is the probability that there is a matching in B' to satisfy all consumers.

Notation:

- N: number of nodes in the network.
- P: set of producer nodes.
- C: set of consumer nodes.
- B: a bipartite graph with one side as P and the other as C and each edge from P to C represents a path from the node in P to the node in C.
- Trig: probability of initially triggered failure for each node.
- Prop: probability that the failure of one node will be propagated to each of its neighbors.
- Neigh(t): set of neighbors of node t.
- Fre: probability of finding a matching in B that can satisfy all consumers.
- Run: the entire process from initially triggering some failed nodes until the propagation process stops because of no newly generated failed nodes.

Assumptions:

- Each node has same probability to be triggered as failed node initially.
- Each node has same propagation probability to propagate its failure to neighbor nodes.

3. Computer Simulation

Monte Carlo simulation technique is used in the simulation. We found that Monte Carlo simulation is easy and fast enough for this simulation and it can provide fairly good result. The whole simulation was written in Matlab. As in any Monte Carlo simulation, when to stop the simulation is an issue. In this simulation, we will watch the change of the values of properties we want to study, the program stops when the change is very small, in which case we believe the value of properties should be very close to the real value of those properties.

Algorithm to generate failed nodes:

1. First let T = empty, T' = empty. There are two different mechanisms in this step. The first one is to generate one integer random number i from 1 to N, trigger node i to fail. The other is to generate a random number r(i) from 0 to 1 for each of the N nodes, if r(i) < Trig, trigger node i to fail. We will stay to the latter unless explicitly mentioned. T = TU{i}
2. While T ~= T'
   - S = T – T'
   - T = T'
   - For each node t in S
     - For each node p in Neigh(t)

If p has not failed

Generate a random number r from 0 to 1.

If r < Prop

Mark node p fail, T = T U {p}

End if

Endif

End for

End for

End while

3. record all the failed nodes.
4. Repeat step 1 to step 3 to the number of steps specified.

Algorithm to find the probability of finding a matching to satisfy all consumers:

1. N = 0, Load the file generated by first algorithm.
2. For timestep from 1 to MAXTIMESTEP, do step 3 – 5.
3. Let B' = B. For each edge in the bipartite graph B', if there is one node on the link represented by the edge failed, remove the edge.
4. For the bipartite graph B' resulted from step 3, generate a matrix where number of rows = |C| and number of columns = |P|, for each entry (i,j) in that matrix, it is 0 if there is no

edge B', otherwise, generate a random floating point number from 0 to 10 and put it into that entry.

5. For the matrix we get from step 4, if its rank equal to the number of rows, then all consumers can be satisfied. N=n+1.
6. Fre = n / MAXTIMESTEP.

In the simulation, the MAXTIMESTEP has been set to 10,000, which can provide fairly good results from experience. For example, I compared the result of 10,000 steps and the result of 138,000 steps, they are in a very close range, which means the simulation has already converged to a fairly good point after 10,000 steps.

4. Simulation results and discussion

Several kinds of simulation have been done to get some statistics about this network. The following is a list of them.

- Relation between node failure probability and degree of the node. Intuitively, the higher the degree is , the higher the probability that it will fail is. As can be seen from figure 1, the degree of the node can be used as a first degree approach of node fail probability in this network. Obviously, it also depends on many other things, such as the degree of its neighbors and so on.

- In order to discuss figure 2, we need to first define two different triggering mechanisms. The first one is that for each Run, there is a specified number of nodes to be triggered, the second one is that for each Run, there is a specified probability for each node to be triggered. For example, if we say for each round, there is one node to be triggered, it should have same expected value of triggered nodes as each node has same probability of 1/n to be triggered. However, as can be seen from figure 2 that these two different mechanisms can produce slightly different results. The reason is that in first mechanism, there are always one node to be triggered, while in the second mechanisms, the number of nodes to be triggered fluctuate around 1, if the number is 0, there is definitely no failed matching, if the number is 2 or more, it will not influence the result that much since if one triggered has stopped a good matching, there is no need to the other triggered node, which means the effect is smaller than sum of two rounds with each round has one triggered node.
- Figure 3 and 4 are for the latter mechanisms since this is where we are mostly interested. They are mainly used to study the effect of propagation probability on the network.

Besides the simulation algorithms, some visualization programs have also been implemented to help users gain better control about

the simulation process. In order to facilitate the study, 20 bipartite graphs have been preset in subdirectory "result". Some simulation results are presented in terms of their sequence no in the bipartite graph series.

Part5: Conclusion

A Monte Carlo simulation package about United States power network with faulty nodes and fail propagation has been implemented by Matlab. The speed of the simulation is acceptably fast. The results of the simulation match our intuition well. In this package, there are some visualization programs besides the numerical simulation programs. Those visualization programs can help us control and get a visual sense of the simulation.

Appendix A: Usage of the package

A small software package has been developed for this project. It includes all the simulation mentioned above plus some extra visualization programs. The package was written by Matlab. In order to use the package, you need to first start matlab by typing "matlab" in the root directory of this package and then run all the programs in the window of Matlab.

- Subdirectories: after you extract the package, you can see a masterproj directory. Cd into this directory, you will see eight subdirectories. They are "edgedowngen", "linkdown", "match", "nodedown", "shortpath", "visualize", "neighnum" and "result"
- General functions of each subdirectory:
  - "shortpath" contains the files to calculate the shortest path between each pair of nodes. It also contains a file to calculate the propagation probability of each edge.
  - "visualize" is used to show the entire network and the propagation result. The files in this subdirectory are used for the case of node failure. The files for the case of edge failure, which is not an emphasis for this project, are in subdirectory "edgedowngen".
  - "nodedown" contains files to run the simulation many times and record the nodes that failed each time.
  - "edgedowngen" contains files that will trigger one edge to down instead of one node down initially, then the fail can propagate too, it is similar to "visualize" with the exception that we are considering edge instead of node this time.

- "linkdown" contains files that calculate the probability that a link(path with several nodes) will fail (if any node on the path fails).
- "match" contains files to calculate the probability of finding a matching to satisfy all consumers and files to visualize the matching.
- "neighnum" contains files to find the probability of finding a match that satisfies all consumers. It also contains files to visualize the results.
- "result" contains files of some simulation results.

- Some common usage of the package
  - To assign a propagation probability for each edge: in directory "shortpath", type "propagation(138, prop)"(138 is the number of nodes in our network, prop is the probability you want to assign to each edge).
  - To find shortest paths between all node pairs: in directory shortpath, type "shortest(138)".
  - To find one shortest path between a specified pair of nodes: in directory "shortpath", type "pairshortest(138, start, end)"
  - To visualize the simulation of node down and failure propagation: in directory "visualize", type "plotnet(138,

trigger, prop)", where trigger is the trigger probability of each node and prop is the propagation probability.

- To visualize the simulation of edge down and fail propagation, in directory "edgedowngen", type: "edgeplotnet(138, trigger, prop)", where trigger is the trigger probability of each edge and prop is the propagation probability.
- To get some statistics about the simulation of node down and propagation, in directory "nodedown", type: "nodedown(138, trigger, prop, MAXRUNTIME)", where trigger is the trigger probability of each node, prop is the propagation probability and MAXRUNTIME is the rounds of simulation you want to do. According to our experience, 10,000 can already provide fairly good result.
- To get some statistics about links(paths from starting node to ending node) failure probability in about the simulation of node down and propagation, you need to do it in 3 steps: in directory "nodedown", type "nodedown(138, trigger, prop, MAXRUNTIME)" and "arraynodedown(138, MAXRUNTIME)", in directory shortpath, type "shortest(138)", then in directory linkdown, type either of "bitpathlinkdown" or "pairlinkdown" or "numpathlinkdown" or "linkdownarray", please use "help"

to see which kinds of parameters you need for these funcitons.

- To find the probability of finding a match to satisfy all consumers in certain bipartite graph, in directory "match/frequency", check for "autorun", it is a script file that will tell you how to do it.
- To visualize the problem of finding a matching to satisfy all consumers, in directory "match/vis2", type either "pathmatchvis" or "ranmatchvis", for "pathmatchvis", you need to do some preparation work, please refer to "match/frequency/autorun.m" to see what preparation work you need to do in order to run the program.

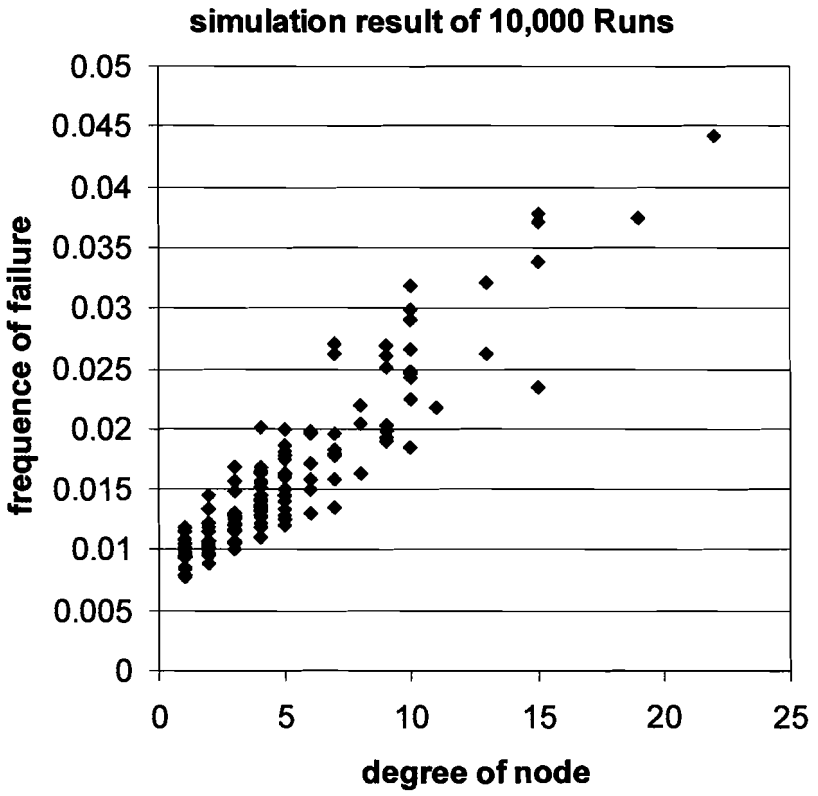

Figure 1. Node failure frequence vs. degree of the node.

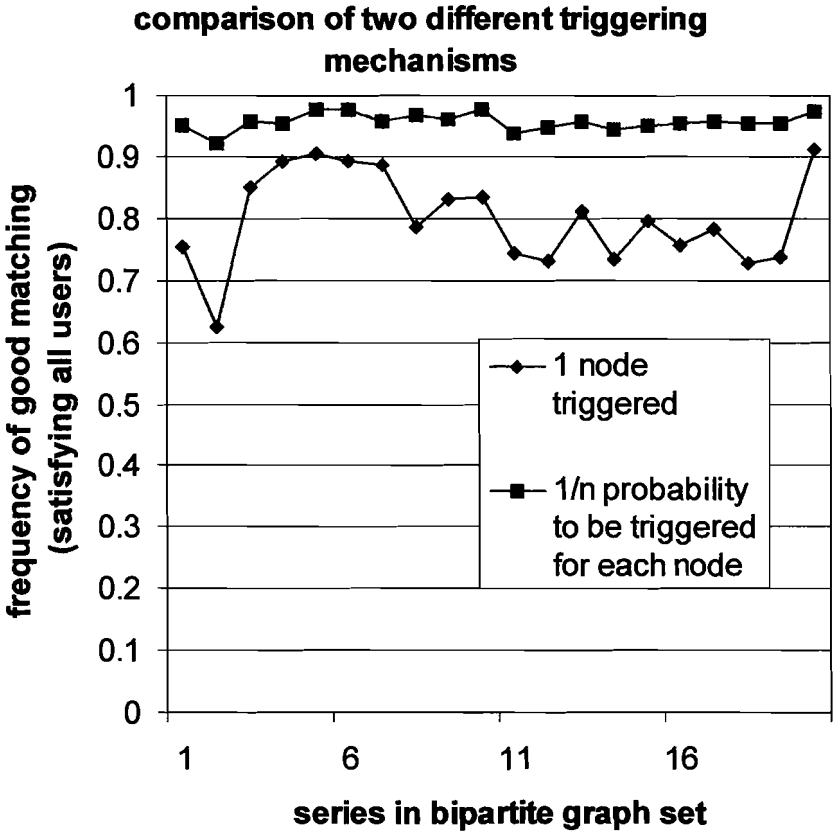

Figure 2. Comparison of two different triggering mechanisms. The first mechanism can generate more failed nodes each time.

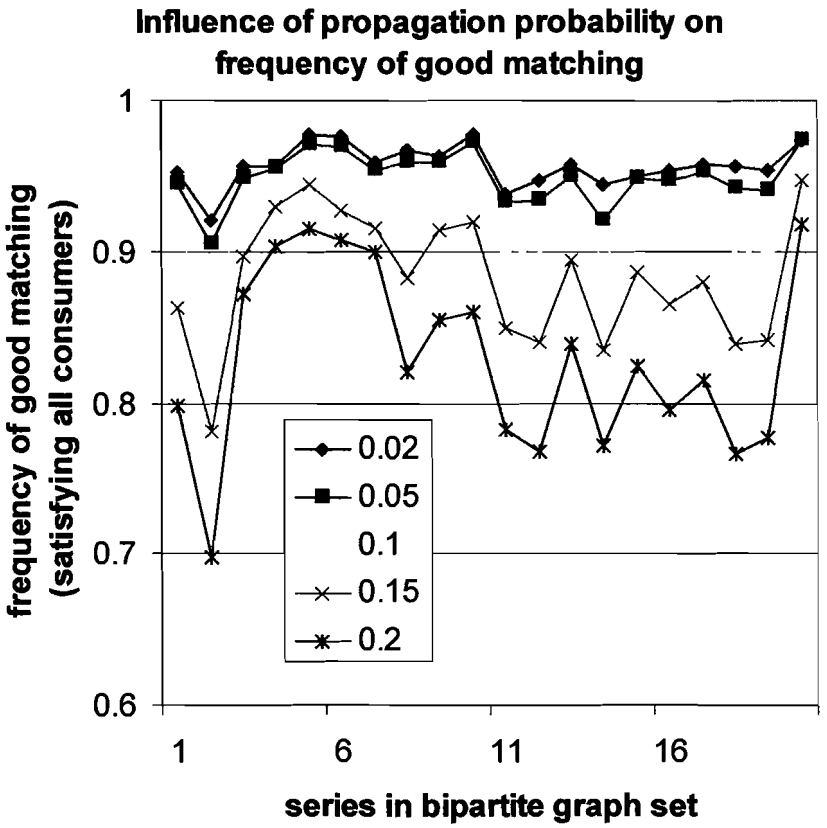

Figure 3. Influence of different propagation prabability on frequence of good matching. 1/n is the initial triggering probability.The resules are listed by different propagation probability.

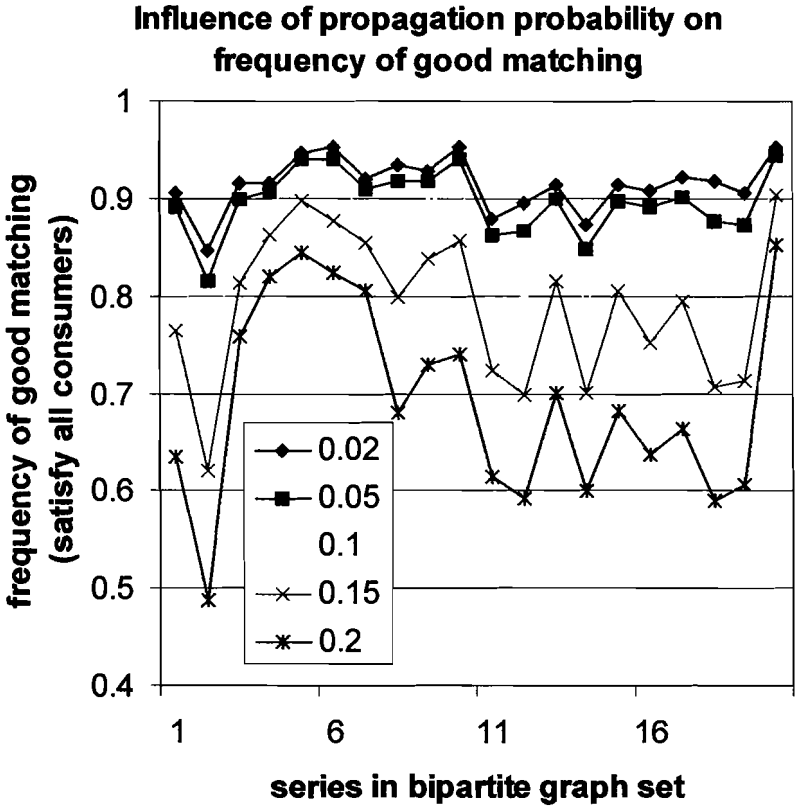

Figure 4. Influence of different propagation prabability on frequence of good matching. 2/n is the initial triggering probability.The resules are listed by different propagation probability.