International Journal of Science and Research (IJSR)
ISSN: 2319-7064
SJIF (2020): 7.803

Analyzing Network Reliability Based on Individual Link Reliabilities Using Exhaustive Enumeration

Krishna Mohan Pitchikala

Graduate Student, University of Texas at Dallas, Texas, USA

Abstract: ***This project aims to explore the extent to which the reliability of the various links in a network contributes to the overall reliability of the network. The studied structure represents a complete undirected graph of five nodes (n = 5) and ten links (m = 10). Each link has a reliability, which depends on a parameter p and a unique identification number. The reliability of the network, defined as the probability of maintaining full connectivity, is computed using exhaustive enumeration. By employing the method of exhaustive enumeration, we systematically evaluate all possible combinations of link states (working or failed) to determine their impact on the network's ability to maintain connectivity. In this paper, we present an algorithm, its implementation, and experimental results that demonstrate the relationship between the reliability of the network and individual link reliabilities. Additionally, we study the impact of random computation errors on the network's reliability by flipping system conditions and observing changes.***

Keywords: network reliability, undirected graph, exhaustive enumeration, link states, computation errors

1. Introduction

Problem Statement:

The aim of the project is to study how network reliability depends on the individual link reliabilities using method of exhaustive enumeration. We have a Complete Undirected Graph with 5 nodes. Complete graphs mean that every node in the network is connected to every other node. (self - loops and parallel edges are excluded) As we have 5 nodes and with all the above - mentioned constraints, we have 10 edges/links in the graph, and this is what the graph looks like.

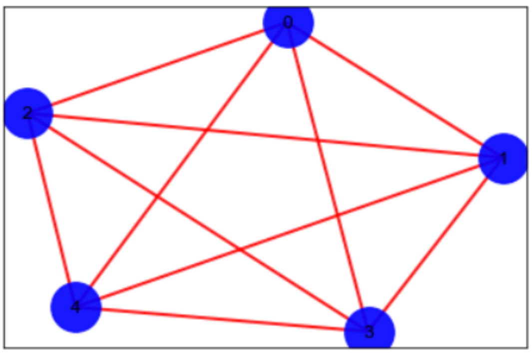

Figure 1: Network Topology of Complete Undirected Graph

As we can see that there are 10 possible edges so there will be 2^{10} = 1024 combinations of graph depending on which link is up and which one is down. The nodes are always up, only the links may fail. Reliability of the link is determined by the parameter 'p' which is same for every link and lies within range of [0, 1]. For link i, where $1 \le i \le 10$, let its reliability be p_i, which is given by the following formula

$$p_i = p^{\lceil di/3 \rceil}$$

where di is the i^{th} digit in a 10 - digit random number, the number consists of 10 digits d1, d2, …., d10. ⌈⌉ denotes the upper integer part, which rounds up any non - integer number to the nearest larger integer. As we can see that the reliability of the link 'i' is dependent on the parameter 'p', we change it over an interval in constant steps to check how the network reliability changes with changing in the p - value. And next keeping this p - value constant we also address how the system reliability changes with random errors by keeping the system at constant p - value which is 0.9 and by flipping the system for k different combinations. Where k is the number of component states that are chosen from 1024 different combinations.

10 - digit random number is 2021455519, hence link reliability is as follows.

Table: link reliabilities

Link (i)	Reliability of Link 'i' (P_i)
1	p
2	0
3	p
4	p
5	p^2
6	p^2
7	p^2
8	p^2
9	p
10	p^2

Note: The system is operational only if the network is connected.

Network reliability:

Reliability of a network is defined as the capacity of the network to offer the same services even during an event of failure. The reliability R of the system is some function of the component reliabilities:

$R = f(p_1, p_2, \ldots\ldots, p_n)$

Where, n is the number of components.

The function f above depends on the configuration, which defines when the system is considered operational, given the states of the components.

Each component has two possible states: operational or failed. The failure of each component is an independent event. Component i is functioning (operational) with probability pi and is in failed with probability 1 - p_i. (These probabilities are usually known)

International Journal of Science and Research (IJSR)
ISSN: 2319-7064
SJIF (2020): 7.803

Method of Exhaustive Numeration:
For the configurations that are neither series nor parallel we use exhaustive method of numeration for calculating the network reliability. In this method, we list all the possible states of the system with accounting the up/down state of the links. The reliability is then calculated by summing the probabilities of all the Up states. In this method as we list all possible states of a path from the source node '1' to the destination node 'n'. Therefore, if there are 'k' edges in the path there would be 2^k possible states of the system. This method is practically applicable only for small network systems because the calculation grows exponentially with increase in the number of links. In our case we have 5 nodes, with 10 links (as the graph is complete) and 1024 (2^{10}) states, hence it is okay to apply here. The graph is said to be connected if each node can be visited from the starting point.

Consider the following network with 3 edges as shown below:

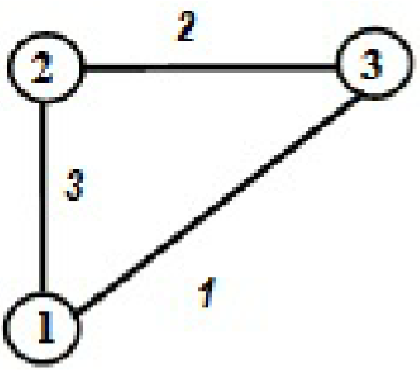

Figure 2: Complete undirected graph with 3 nodes

Number of links down	Possible configuration of links	System condition
0	[1, 1, 1]	Up
1	[0, 1, 1]	Up
	[1, 0, 1]	Up
	[1, 1, 0]	Up
2	[0, 0, 1]	Down
	[1, 0, 0]	Down
	[0, 1, 0]	Down
3	[0, 0, 0]	Down

In the above, each 0 indicates that the link is down and 1 indicates that the link is up. Probability that the link is up is 'p' and probability that the link is down is '1 - p'.

Reliability of the system is computed by summing the product of probabilities of all the valid configurations. From the above table, configurations in which the graph is connected are 111, 011, 101, 110.
Hence, reliability of the above system is computed as:

Reliability = p*p*p + (1 - p) *p*p + p* (1 - p) *p + p*p* (1 - p)

2. Algorithm

The algorithm to compute the network reliability is as follows
1) Create a graph_matrix to represent links within the network. The value graph_matrix [i] [j] denotes the edge from node i to node j. If the node exists, then the value is 1 else the value is 0.
2) Total number of combinations are 2^{10} = 1024 as there are 10 edges/links. To generate links corresponding to the 1024 states, we use 10bit binary representation of the state and assign each bit to the state of each link. For this we convert the decimal numbers from 0 to 1023 to their corresponding 10 - digit binary format.
3) In the obtained binary format, 0 indicates that the link is down, and 1 indicates that the link is up. Thus, the binary representations would be interpreted as follows:
   - 20: 0000010100 implies 2 links are up and 8 links are down
   - 0: 0000000000 implies that all links are down
   - 63: 0000011111: implies 5 links are up and 5 links are down
4) We check whether the graph is connected or not for each combination of links using the graph_is_connected () function. Because only if the graph is connected, it is operational and only then is reliability computed.
5) For all the connected graphs we find the reliability of the state as the product of reliabilities (pi) for links for which the matrix value is 1 and (1 - pi) for links for which the matrix value is 0.
6) The total reliability is obtained by adding the products of probabilities obtained for systems that operational. For example, for a combination of 1010011111, network reliability would be calculated as p* (1 - p) *p* (1 - p) * (1 - p) *p*p*p*p*p + products of probabilities of other combinations that are operational.

Pseudo Code

Pseudo Code for finding relation between Network Reliability and parameter 'p'

Input:
- Network Topology: which is a complete undirected graph with 5 nodes
- Reliability Parameter 'p': which changes from 0.05 to 1 in steps of 0.05
- 10 - digit random number: which is 2021455516

Output:
- Network Reliability of the system vs Parameter 'p'

Pseudo Code:

Note: from 'p' and the10 - digit random number we find the reliability of each link using the formula pi=p[di/3] and the value is shown in table - 1.
1) Create Network Topology
2) Initialize id = 2021455516 and p - value = 0.05
3) Compute combinations of links which is a list of lists with each inner list representing each combination of link states.
4) Initialize reliability_list to an empty list and state_reliability = 0
5) While p≤1:
   a) Initialize network_reliability = 0
   b) For i in range [0, 1023]:

Check if the graph is connected for that possible combination of links

If the graph is connected:
Initialize state_reliability = 1
For j in range [0, 9]:

Calculate reliability of each link
if link is up:
temp = p_i
else:
temp = 1 - p_i
state_reliability = state_reliability * temp
network_reliability = network_reliability + state_reliability
a. Append network_reliability to reliability_list
b. p = p + 0.5
2. print results and plot the graph

Pseudo code for finding Network Reliability vs Random Errors

Input:
- Network Topology: which is a complete undirected graph with 5 nodes
- Reliability Parameter ‘p’ = 0.9
- 10 - digit random number: which is 2021455516
- Number of random states to be flipped k (chosen from 0 to 20 in steps of 1)

Output:
- Network Reliability of the system vs Parameter ‘k’

Pseudo Code:
Note: K number values are chosen randomly within the range [0, 1023] and the corresponding system condition is flipped that is if the system is up change it to down and vice versa.
1) Initialize link reliability and experiment_list to empty list
2) For i in range [0, 99]:

Initialize k = 0, p=0.9

Compute combinations array/list which is the binary representation of numbers from 0 to 1023

While k ≤ 20:
For i in range [0, k - 1]:
Choose random integer between [0, 1023]
If the graph formed from that link is connected:
Flip the system to down state
Else:
Flip the statem to up state
Compute the network reliability as we did in previous pseudo code
Append the results into the reliability_list
a. K = k + 1
2. Print results and plot the graphs

3. Results and Analysis

Network reliability vs Parameter - p:

Network Reliability for corresponding "p" value

p	Reliability
0.05	0.00033146557037324557
0.1	0.0032751484276600036
0.15	0.012917305459435301
0.2	0.03419254964224006
0.25	0.07174613326787949
0.3	0.12874792215293993
0.35	0.20595226181267925
0.4	0.30122823909376
0.45	0.4096731141438961
0.5	0.5242919921874999
0.55	0.6371048703498238
0.6	0.7404516030873617
0.65	0.8282238559741396
0.7	0.896769691664141
0.75	0.9452889785170555
0.8	0.9756522997350401
0.85	0.9917070569194887
0.9	0.9982476382980605
0.95	0.9998831236842189

Figure 3: p values and corresponding Reliability

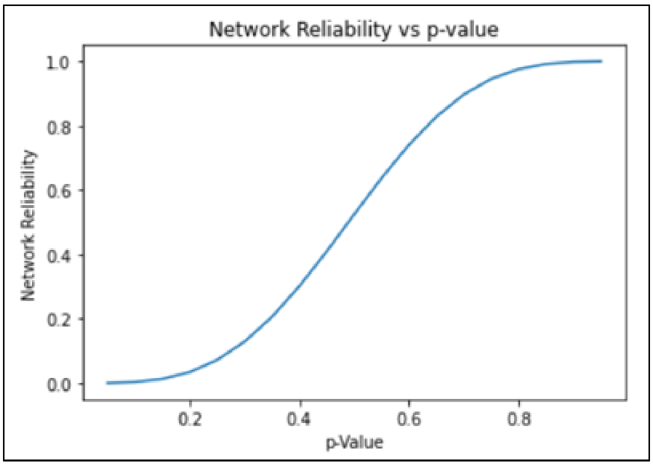

Figure 4: Network Reliability vs p - value

Analysis:
- As we can see that the reliability of the network increases with increase in p - value and reaches a constant value after p increases beyond 0.8. This can also be concluded theoretically as the total reliability of a network depends on the reliability of each link and with increase in reliability of each link the total network reliability increases.
- Hence Network reliability is directly proportional to the link reliability ‘p’

Network Reliability vs parameter – k

Network Reliability vs parameter-k

k	Reliability
0	0.9982476382980582
1	0.9974932503865981
2	0.9949429030524576
3	0.992162967389165
4	0.9886315808548541
5	0.9799518845904082
6	0.9699560415575796
7	0.9661914381447313
8	0.9574992591911244
9	0.9506400080169592
10	0.9415579697058728
11	0.9306342095291309
12	0.9199903082246993
13	0.9096859835534856
14	0.8988365627104797
15	0.8875211974277115
16	0.8746288744597374
17	0.8624810109255787
18	0.8454121158726885
19	0.8273508116446243

Figure 5: k values and corresponding Reliability

Change in Network Reliability vs k

k	Change in Network Reliability
0	0.0
1	0.00077467365215933
2	0.0017192058449151393
3	0.00359165094784486
4	0.006447318491950127
5	0.012732341296780025
6	0.0165434421640831
7	0.02137570707297487
8	0.025676586465765094
9	0.03304230874545222
10	0.03885807388964546
11	0.0479280536260559
12	0.06043771404563758
13	0.07291825193275758
14	0.08107724159316976
15	0.09400275153454507
16	0.11183764278069741
17	0.1275967740437689
18	0.14040921494506142
19	0.15264000376791953
20	0.16945298080458449

Figure 6: k values and corresponding change in Network Reliability

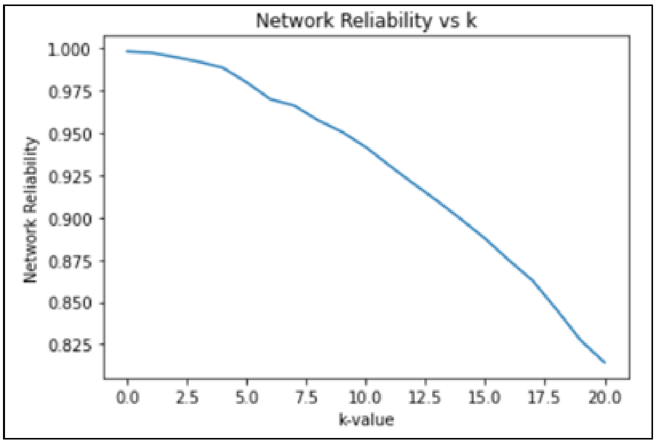

Figure 7: Network Reliability vs k

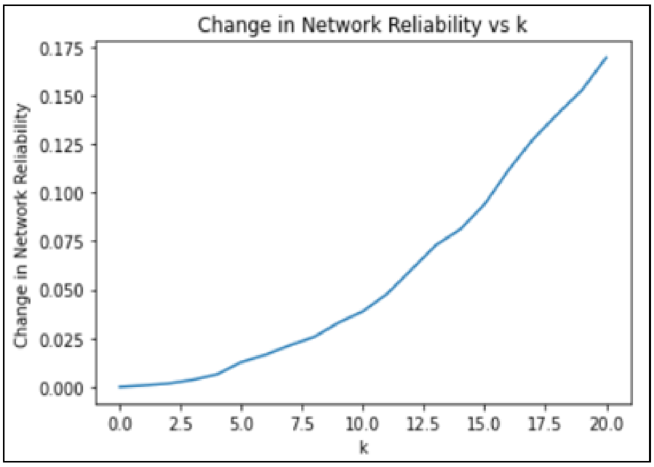

Figure 8: Change in Network Reliability vs k

Analysis

- As we can see the reliability of the network decreases with increase in k - value figure - 7. The k - value is nothing, but the random error introduced into the network, and we can see that with increase in random errors in the network the network reliability decreases.
- Also, from the figure - 8, we can also see that the with increase in k value the change in network reliability is increasing. This can be supported by the statement that with increase in number random errors in the reliability of network decreases more rapidly and hence the change in the network reliability increases.
- Thus, it can be concluded that network reliability is indirectly proportional to the number of random states to be flipped (k) and the change in network reliability is directly proportional to k. This is show in below figure - 7.

4. Conclusion

This study demonstrates how the overall reliability of a network depends on individual link reliabilities. Using exhaustive enumeration, we computed the network reliability for all possible configurations of link failures and observed how different link reliabilities affect the system. The results show that as the reliability of individual links increases, the network becomes more robust. Furthermore, introducing random errors in the configuration state has an impact on system reliability.

References

[1] https: //www.igi - global. com/dictionary/cellular - network - planning/20172
[2] https: //networkx. org/documentation/latest/_downloads/networkx_reference. pdf
[3] https: //www.sciencedirect. com/science/article/pii/S0951832015001702
[4] https: //ceur - ws. org/Vol - 1830/Paper10. pdf
[5] https: //en. wikipedia. org/wiki/Reliability_(computer_networking)
[6] https: //particular. net/blog/the - network - is - reliable
[7] https: //www.sciencedirect. com/topics/computer - science/network - reliability