SIM-NET 1

01DSNBG - Project: Software-defined communication systems

Project Report

Full Name	Student ID
Anamali Bjorna	322621
Gorji Fatemeh	326721
Jamishi Sadegh	314215
Jasson Olivier	321275

Supervised by: Prof.Pellerey and Dr. Raineri

Politecnico di Torino, July 19, 2024

Contents

Acknowledgments

We would like to express our sincere gratitude to our professors and mentors for their invaluable guidance and support throughout the development of the **Network Reliability Optimization App**. Their expertise and encouragement have been instrumental in the successful completion of this project.
We extend our heartfelt thanks to Professor Franco Pellerey for his continuous support and insightful feedback. We are deeply grateful to Dr. Roberta Raineri for her mentorship and valuable suggestions. We also wish to thank Professor Andrea Carena for his expert advice and encouragement.

1 | Introduction

In the modern digital age, telecommunications networks serve as the backbone for global connectivity and data exchange. These networks, however, are vulnerable to various stochastic events, known as hazards, which can damage data exchange facilities and optical fiber lines, leading to significant disruptions in communication. This project focuses on analyzing and enhancing the reliability of a telecommunications network through rigorous statistical and optimization methods.

1.1 | Network Topology and Modeling

The initial phase of the study concentrates on a predefined network topology, depicted in Figure 1.1.

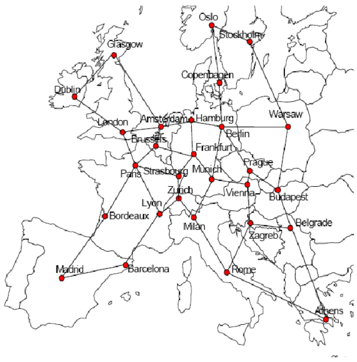

Figure 1.1: Given network topology

This telecommunications network is modeled as a graph to facilitate the analysis of its complex structures and the interactions between various components. In this model, each node represents a data exchange facility, and each link symbolizes an optical fiber line. The topology under consideration consists of 41 links and 28 nodes, as illustrated in Figure 1.2.

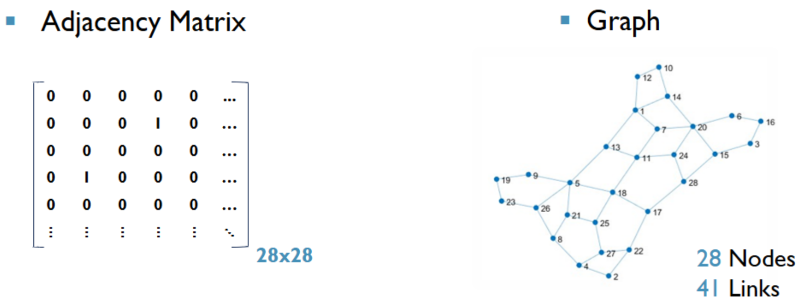

Figure 1.2: Network graph with 28 nodes and 41 links

1.2 | Reliability Measure and Analysis

The primary measure of reliability in this project is the *full connectedness* of the graph. This measure is defined as the percentage of time the network remains fully connected, allowing every node to communicate with every other node, either directly or indirectly, through a series of links. At the end of this report another idea on the measure of the reliability will be proposed, however we will limit the discussion to the introduction of the measure and the results of the simulation.
Throughout the project, the components (nodes and links) are assumed to be **independent**, meaning the failure or repair of one component does not affect the others.

1.3 | Objectives and Methodology

The primary objectives of this project are:

1. **Evaluating Network Reliability:** This involves assessing the reliability of the network by considering potential failures of both nodes and links. Statistical methods are used to model the probability of failures and analyze their impact on network performance. The reliability evaluation is based on the concept of full connectedness, ensuring continuous communication across the network.

2. **Enhancing Network Reliability:** This involves proposing the addition of new links to improve network reliability. Critical points in the network where additional links would most enhance reliability are identified. Various optimization methods, including clustering and diameter algorithms, are employed to determine the most effective links to add. These suggestions enable users to make informed decisions based on budget constraints and available resources.

1.4 | Tools and Techniques

To achieve these objectives, a simulator has been developed, along with a mathematical tool to validate the simulator's results, ensuring accuracy and reliability. The analysis begins by modeling the failure and repair processes of nodes and links using Poisson processes. This approach utilizes exponential distributions to represent the time between failures and repairs. The failure rate (λ) and repair rate (μ) are defined for both nodes and links, with μ being the reciprocal of the mean down time (MDT).
The steady-state probability distributions of the Markov chain are calculated to determine the long-term behavior of the network.

2 | Poisson Processes

The Poisson process is a fundamental concept in probability theory and is widely used in modeling random events over time. It is particularly useful in fields such as telecommunications, traffic engineering, and reliability analysis. In the model, the Poisson process is used to describe the failure and repair rates of network components (nodes and links) [4].

2.1 | Definition and Properties

A Poisson process is a type of stochastic process that models a series of events occurring randomly over a continuous time interval. The process is characterized by the following properties:
1. **Independence**: The occurrence of each event is independent of the occurrence of previous events. This implies that the time between successive events (inter-arrival times) is independently and identically distributed.
2. **Stationarity**: The probability of an event occurring in a given time interval depends only on the length of the interval, not on its position in time. Specifically, the number of events occurring in an interval of length t follows a Poisson distribution with parameter λt, where λ is the rate of the process.
3. **Memorylessness**: The process has no memory of past events. Formally, the time until the next event occurs follows an exponential distribution, which is characterized by the memoryless property. This means that the probability of an event occurring in the next t units of time is the same regardless of how much time has already elapsed.

2.2 | Mathematical Formulation

Let $N(t)$ denote the number of events that have occurred by time t. The Poisson process with rate λ has the following properties:
- The number of events in any time interval of length t is Poisson distributed:

$$P(N(t) = k) = \frac{(\lambda t)^k e^{-\lambda t}}{k!}, \quad k = 0, 1, 2, \ldots$$

- The time between successive events (inter-arrival times) is exponentially distributed with parameter λ:

$$f(t) = \lambda e^{-\lambda t}, \quad t \geq 0$$

This implies that the mean time between events is $\frac{1}{\lambda}$.

2.3 | Application to Network Reliability

In our network model, we assume that both node failures and link failures follow Poisson processes. This assumption allows us to model the time between failures and repairs using exponential distributions. Specifically:

- λ_N: Node failure rate (failures per unit time).
- μ_N: Node repair rate (repairs per unit time, where $\mu_N = \frac{1}{\mathrm{MDT}_N}$, with MDT_N being the mean down time for a node).
- λ_L: Link failure rate (failures per unit time).
- μ_L: Link repair rate (repairs per unit time, where $\mu_L = \frac{1}{\mathrm{MDT}_L}$, with MDT_L being the mean down time for a link).

In this context, the repair rate μ is the reciprocal of the mean down time (MDT). For example, if the unit of time is one month and μ is 2, this implies that the time it takes for the node or link to be repaired is $\frac{1}{2}$ months.
These rates define the transition rates in the continuous-time Markov chain model used to analyze the network's reliability (chapter 3). The transition rate matrix incorporates these rates to describe the probabilities of moving from one state to another in the network.

2.4 | Discretization Approach

To simulate the occurrence of failures and repairs in a telecommunications network, the Poisson process, which typically operates in continuous time, needs to be adapted for discrete time steps. This involves breaking down the continuous time into small, manageable intervals and determining the likelihood of events (failures or repairs) within each interval. In a Poisson process, events occur continuously and independently at a constant average rate λ (for failures) or μ (for repairs). The time between consecutive events follows an exponential distribution. For simulation purposes, these continuous processes must be discretized.
The total simulation time, specified as "time" in the simulator input, is divided into several discrete intervals, each defined by the time step Δt. The number of these intervals, or steps, is given by:

$$\text{step} = \frac{\text{time}}{\Delta t}$$

The key idea is to approximate the continuous Poisson process by evaluating the probability of an event occurring within each discrete time step.
Given a time step Δt, if the product $\lambda \cdot \Delta t$ is sufficiently small (typically below 0.1), the probability of a failure event occurring within this interval can be approximated by:

$$p = \lambda \cdot \Delta t$$

Similarly, the probability of a repair event occurring within the same interval is:

$$\text{rep} = \mu \cdot \Delta t$$

In the simulator, these probabilities are calculated as:

$$p = \frac{\lambda \cdot \text{time}}{\text{step}}$$

$$\text{rep} = \frac{\mu \cdot \text{time}}{\text{step}}$$

2.4.1 | Tuning of the parameters

In order to have a good performance of the simulator, we need to compromise between the running time and the precision of the measures. Out of the four parameters of the Poisson process modeling the network (λ_{node}, λ_{link}, μ_{node}, μ_{link}), we consider that μ_{link} is always the biggest, as it is the inverse of the expected reparation time of a link. The standard value for μ_{node} is 2. It means that, for *rep* to be smaller than 0.1, Δt has to be smaller than 0.05. For a running time of 300 months, it means a number of steps of 6000. Depending on the link selection method, this might be very computationally demanding. Therefore, depending on the computing time, summarized later in table 5.1, we decided to accept a value of rep smaller than 0.2 instead of 0.1. This adjustment effectively halves the computation time while maintaining an acceptable level of precision.

2.4.2 | Practical Interpretation

- **Failure Probability** p: This represents the likelihood that a node or link will fail within a given discrete time step. It is derived from the failure rate λ and the length of the time step Δt.
- **Repair Probability rep**: This indicates the probability that a failed node or link will be repaired within the time step, based on the repair rate μ and the time step length Δt.

By using these probabilities, the simulator can step through each time interval and probabilistically determine whether a failure or repair occurs for each component (node or link) of the network.

3 | Markov Chains and Steady-State Analysis

A Markov chain [3] is a stochastic process that describes a sequence of possible events where the probability of each event depends only on the state attained in the previous event. This property is known as the Markov property or memorylessness.
In the context of the developed model, a Markov chain is used to represent the states of the network (nodes and link) and the transitions between these states due to failures and repairs. The states of the system can include scenarios such as both nodes and the link being operational, one node failing, the link failing, or simultaneous failures of nodes and the link.

3.1 | Transition Matrix

The transition matrix, also known as the rate matrix or generator matrix, is a fundamental component of a Markov chain. It is a square matrix where each element P_{ij} represents the transition rate from state i to state j. For continuous-time Markov chains, the transition rates are defined based on the failure and repair rates of the components (nodes and link).
For example, in the simple topology consisting of 2 nodes and 1 link, the transition matrix P might be represented as:

$$P = \begin{pmatrix} -\lambda & \lambda \\ \mu & -\mu \end{pmatrix}$$

where λ is the failure rate of the link, and μ is the repair rate of the link. Similar matrices can be constructed for node failures and for simultaneous node and link failures as we will see later.

3.2 | Steady-State Probability Distributions

The steady-state probability distribution of a Markov chain represents the long-term behavior of the system, where the probabilities of being in each state remain constant over time. There are two primary methods to determine the steady-state probabilities:

1. Solving the Kolmogorov Equations The Kolmogorov forward equations (or differential equations) describe the time evolution of the state probabilities. For a transition matrix P, the Kolmogorov equations are given by:

$$\dot{P}(t) = P(t) \cdot P$$

By solving these differential equations, the steady-state probabilities can be found as $t \to \infty$.

2. Solving the Equilibrium Equation The equilibrium equation approach involves finding a probability distribution P_∞ that satisfies the equilibrium condition:

$$P_\infty = P_\infty \cdot P$$

This can be rewritten as:

$$P_\infty \cdot (I - P) = 0$$

where I is the identity matrix. Additionally, the sum of the steady-state probabilities must equal 1:

$$\sum_i P_{i,\infty} = 1$$

This set of linear equations can be solved to obtain the steady-state probability distribution.
Both methods ultimately yield the same steady-state probabilities, providing a comprehensive understanding of the network's reliability and behavior under different failure conditions.

3.3 | Application to the Model

In this project, the steady-state probability distributions are calculated for different failure scenarios using both the Kolmogorov equations and the equilibrium equation. This dual approach ensures the accuracy and robustness of the model, allowing for a thorough analysis of the network's reliability.
By leveraging the power of Markov chains and these analytical methods, the model effectively evaluates the long-term performance of the telecommunications network.

4 | Mathematical Model

To validate the simulator, a mathematical model was developed to evaluate the steady-state probability of full connectedness in a simple topology consisting of only 2 nodes and 1 link, all **independent** of each other. This model considers three distinct cases: link failure, node failure, and the simultaneous occurrence of both node and link failures. For each case, the probability P_0 is the probability that the graph is fully connected.
For the simulator validation, it is important to see the steady-state (asymptotic) behavior of the probability distributions. Therefore, we are interested in evaluating the following limit:

$$\lim_{t \to \infty} P_0(t)$$

which we will denote as $P_{i,\infty}$ where "i" corresponds to the number of the state.

4.1 | Link Failure

In the case of link failure, the states of the system can be described as follows:

State	Description
0	Link is operational
1	Link has failed

Table 4.1: Link Failure States

The transition rate matrix P is defined as:

$$P = \begin{pmatrix} -\lambda & \lambda \\ \mu & -\mu \end{pmatrix}$$

The Kolmogorov equation, which describes the time evolution of the state probabilities, is given by:

$$\dot{P}(t) = \frac{dP(t)}{dt} = P(t) \cdot P$$

This expands to the following system of differential equations:

$$\begin{pmatrix} \dot{P}_0(t) \\ \dot{P}_1(t) \end{pmatrix} = \begin{pmatrix} P_0(t) & P_1(t) \end{pmatrix} \cdot \begin{pmatrix} -\lambda & \lambda \\ \mu & -\mu \end{pmatrix}$$

Solving these equations, we get:

$$\dot{P}_0(t) = -\lambda P_0(t) + \mu P_1(t) \tag{4.1}$$

$$\dot{P}_1(t) = \lambda P_0(t) - \mu P_1(t) \tag{4.2}$$

With the initial conditions:

$$P_1(0) = 0, \quad P_0(0) = 1$$

Using MATLAB, the solutions for $P_0(t)$ and $P_1(t)$ are found to be:

$$P_0(t) = \frac{\mu}{\lambda + \mu} + \frac{\lambda}{\lambda + \mu} e^{-(\lambda+\mu)t}$$

$$P_1(t) = 1 - P_0(t) = \frac{\lambda}{\lambda + \mu} - \frac{\lambda}{\lambda + \mu} e^{-(\lambda+\mu)t}$$

At steady state (as $t \to \infty$), the probabilities are:

$$P_{0,\infty} = \frac{\mu}{\lambda + \mu}, \quad P_{1,\infty} = \frac{\lambda}{\lambda + \mu}$$

This steady-state result can be confirmed by solving the equilibrium equation:

$$P_\infty = P_\infty \cdot P$$

Rearranging gives:

$$P_\infty - P_\infty \cdot P = 0$$

$$P_\infty(I - P) = 0$$

Where:

$$\begin{pmatrix} P_{0,\infty} & P_{1,\infty} \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - \begin{pmatrix} 1-\lambda & \lambda \\ \mu & 1-\mu \end{pmatrix} = 0$$

$$\begin{pmatrix} P_{0,\infty} & P_{1,\infty} \end{pmatrix} \cdot \begin{pmatrix} \lambda & -\lambda \\ -\mu & \mu \end{pmatrix} = 0$$

Given that:

$$P_{0,\infty} + P_{1,\infty} = 1$$

Solving for $P_{0,\infty}$ and $P_{1,\infty}$ gives:

$$-\lambda P_0 + \mu(1 - P_0) = 0$$

$$(\lambda + \mu)P_0 = \mu$$

$$P_{0,\infty} = \frac{\mu}{\lambda + \mu}$$

$$P_{1,\infty} = 1 - P_{0,\infty} = \frac{\lambda}{\lambda + \mu}$$

An important aspect of this result is that the ratio:

$$\frac{\mu}{\lambda + \mu}$$

is known as the *availability* of a component, which represents the long-term proportion of time that the component is operational. Similarly the ratio $\frac{\lambda}{\lambda+\mu}$ is referred to as the *unavailability* of the network component. This concept is referenced in reliability engineering literature [2], highlighting the validity of the obtained result in this context.

4.2 | Node Failure

In the case of node failure, the states of the system can be described as follows:

State	Description
0	Both nodes are operational
1	Node 1 is operational, Node 2 has failed
2	Node 2 is operational, Node 1 has failed
3	Both nodes have failed

Table 4.2: Node Failure States

The transition rate matrix P is defined as:

$$P = \begin{pmatrix} 1-2\lambda & \lambda & \lambda & 0 \\ \mu & 1-(\lambda+\mu) & 0 & \lambda \\ \mu & 0 & 1-(\lambda+\mu) & \lambda \\ 0 & \mu & \mu & 1-2\mu \end{pmatrix}$$

The steady-state probabilities P_∞ satisfy:

$$P_\infty = P_\infty \cdot P$$

$$P_\infty - P_\infty \cdot P = 0$$

$$P_\infty (I - P) = 0$$

This leads to the following system of equations:

$$\begin{pmatrix} P_{0,\infty} & P_{1,\infty} & P_{2,\infty} & P_{3,\infty} \end{pmatrix} \cdot \begin{pmatrix} 2\lambda & -\lambda & -\lambda & 0 \\ -\mu & \lambda+\mu & 0 & -\lambda \\ -\mu & 0 & \lambda+\mu & -\lambda \\ 0 & -\mu & -\mu & 2\mu \end{pmatrix} = 0$$

We can solve the linear system of equations.
Solving for $P_{0,\infty}$ we obtain:

$$P_{0,\infty} = \frac{\mu^2}{(\lambda+\mu)^2}$$

The other probabilities are:

$$P_{1,\infty} = P_{2,\infty} = \frac{\lambda\mu}{(\lambda+\mu)^2}$$

$$P_{3,\infty} = \frac{\lambda^2}{(\lambda+\mu)^2}$$

4.3 | Link and Node Failure

In the case of both link and node failure, the following table cointains the details of the 8 states of the system :

State	Link	Node 1	Node 2
0	UP	UP	UP
1	DOWN	UP	UP
2	UP	UP	DOWN
3	UP	DOWN	UP
4	DOWN	DOWN	UP
5	DOWN	UP	DOWN
6	DOWN	DOWN	DOWN
7	UP	DOWN	DOWN

Table 4.3: Link and Node Failure States

The transition rate matrix P is given by:

$$P = \begin{bmatrix} -3\lambda & \lambda & \lambda & \lambda & 0 & 0 & 0 & 0 \\ \mu & -(2\lambda+\mu) & 0 & 0 & \lambda & \lambda & 0 & 0 \\ \mu & 0 & -(2\lambda+\mu) & 0 & 0 & \lambda & 0 & \lambda \\ \mu & 0 & 0 & -(2\lambda+\mu) & \lambda & 0 & 0 & \lambda \\ 0 & \mu & 0 & \mu & -(\lambda+2\mu) & 0 & \lambda & 0 \\ 0 & \mu & \mu & 0 & 0 & -(\lambda+2\mu) & \lambda & 0 \\ 0 & 0 & 0 & 0 & \mu & \mu & -3\mu & \mu \\ 0 & 0 & \mu & \mu & 0 & 0 & \lambda & -(\lambda+2\mu) \end{bmatrix}$$

The system of equations can be solved with MATLAB to obtain the following steady-state probability distributions:

$$P_\infty = \left(\frac{\mu^3}{(\lambda+\mu)^3}, \quad \frac{\lambda\mu^2}{(\lambda+\mu)^3}, \quad \frac{\lambda\mu^2}{(\lambda+\mu)^3}, \quad \frac{\lambda\mu^2}{(\lambda+\mu)^3}, \quad \frac{\mu\lambda^2}{(\lambda+\mu)^3}, \quad \frac{\mu\lambda^2}{(\lambda+\mu)^3}, \quad \frac{\mu\lambda^2}{(\lambda+\mu)^3}, \quad \frac{\lambda^3}{(\lambda+\mu)^3} \right)$$

Similarly to the previous cases, the probability of the graph being fully connected as $t \to \infty$ is given by the product of the availabilities[2] of the 3 components :

$$P_{0,\infty} = \frac{\mu^3}{(\lambda + \mu)^3}$$

If different failure and repair rates are considered for nodes and links, the expression would be modified accordingly. Distinguishing between the failure rate (λ_N) and repair rate (μ_N) for nodes, and the failure rate (λ_L) and repair rate (μ_L) for links, the probability of the graph being fully connected as $t \to \infty$ becomes:

$$P_{0,\infty} = \frac{\mu_N^2 \cdot \mu_L}{(\lambda_N + \mu_N)^2 \cdot (\lambda_L + \mu_L)} = Avail(Node_1) \times Avail(Node_2) \times Avail(Link)$$

4.4 | Simulator Validation

To validate the simulator , its results are compared with those obtained from the mathematical model. The simulator runs multiple iterations to estimate the reliability of the network under the assumption of independent failures of nodes and links.

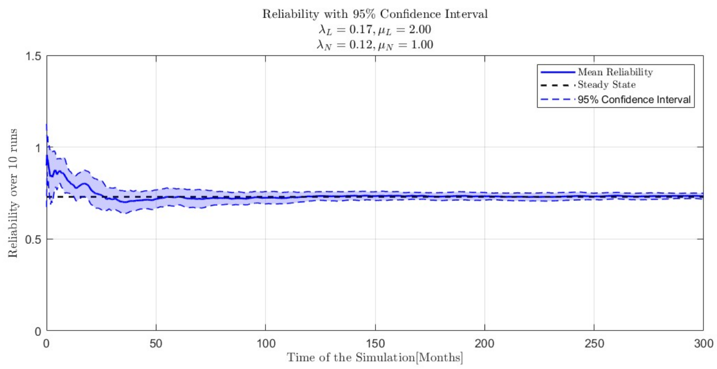

Figure 4.1: Simulator validation for link and node failure

The parameter values used in the simulation are as follows: $\lambda_{\text{Link}} = \frac{1}{6}$, indicating a link failure rate of one failure every six months; $\mu_{\text{Link}} = 2$, representing a mean down time (MDT) of 0.5 months for link repairs; $\lambda_{\text{Nodes}} = \frac{1}{8}$, indicating a node failure rate of one failure every eight months; and $\mu_{\text{Nodes}} = 1$, representing a mean down time (MDT) of one month for node repairs. The steady-state probability $P_{0,\infty}$ calculated using the parameter values by using the formula obtained in section 4.3 is:

$$P_{0,\infty} = \frac{\mu_N^2 \cdot \mu_L}{(\lambda_N + \mu_N)^2 \cdot (\lambda_L + \mu_L)} \approx 0.7293$$

Thus, the steady-state probability of the network being fully connected is approximately 72.93%.
Figure 4.1 provides a visual comparison between the simulator results and the mathematical model. The X-axis represents the time span of the simulation in months, up to 300 months, while the Y-axis shows the reliability of the network, which is the percentage of time the network remains fully connected. The blue solid line represents the mean reliability calculated over 10 simulation runs, and the dashed line indicates the steady-state reliability as predicted by the mathematical model. The shaded area represents the 95% confidence interval for the reliability, providing a measure of the variability in the simulation results.
The simulator was run 10 times to account for the stochastic nature of node and link failures and repairs. Multiple runs help in averaging out randomness and obtaining a more accurate estimate of the network's reliability. The reliability converges quickly due to the high repair rates relative to failure rates, which ensures that the network can recover rapidly from failures, maintaining a high level of connectivity over time.

5 | Link addition strategy

The goal of this project is to upgrade the reliability of a network, with the prime solution being to add a new link in the network. In this part, we will focus on different strategies that can be adopted to determine the best link, For most of them, the idea is to add one link to the original network from a collection of candidates, measure the reliability with the simulator of the updated network, and repeat for all candidates. The link that improves the reliability the most is the best suggestion. The strategies differ in the initial collection of candidates.

5.1 | Brute Force

One fundamental approach to enhancing network reliability is the brute force method. This involves exhaustively testing all possible new links in the network to identify the optimal addition that maximizes reliability.
Mathematically, let $G = (V, E)$ represent the graph of the network, where V is the set of nodes and E is the set of edges (links). To enhance the network, we consider adding a new link e between every pair of nodes (u, v) such that $u, v \in V$ and $e \notin E$.
The brute force approach involves the following steps:

- **Generate All Possible Links:** Enumerate all possible pairs of nodes (u, v) that do not currently have a link. Since the graph is undirected, each pair is considered only once. This results in $\binom{n}{2} - |E|$ possible new links, where n is the number of nodes.
- **Evaluate Each Potential Link:** For each potential new link $e = (u, v)$, calculate the network's reliability if e were added. The reliability R is defined as the percentage of time the network remains fully connected:
$$R = \frac{\text{Time network is fully connected}}{\text{Total simulation time}} \times 100\%$$
- **Select the Optimal Link:** Identify the link e that maximizes the reliability improvement ΔR:
$$\Delta R = R(G + e) - R(G)$$
where $R(G + e)$ is the reliability of the network with the new link added, and $R(G)$ is the reliability of the original network.

The complexity of this brute force method is $O\left(\frac{n^2}{2}\right)$, reflecting the need to evaluate each of the $\binom{n}{2}$ possible links in an undirected graph. While this approach guarantees the identification of the optimal link, it is computationally intensive, especially for large networks.

$$\text{Complexity:} \quad O\left(\frac{n^2}{2}\right) \tag{5.1}$$

Despite its high computational cost, the brute force method serves as a benchmark, ensuring that no potential link is overlooked and providing the best possible improvement in network reliability.

5.2 | Clustering Strategy

Clustering is an effective technique used to enhance network reliability by identifying groups of nodes that can be connected to improve the overall robustness of the network. This project employs the K-means clustering algorithm to partition the network and determine the optimal links to add between clusters, thereby reducing the complexity compared to the brute force method.
The clustering strategy involves the following steps:

- **Partition the Network into Clusters:** The network is represented as a graph G, and the shortest path distances between all pairs of nodes are calculated to form a distance matrix D. The K-means clustering algorithm is applied to this distance matrix to partition the nodes into $k = 2$ clusters using the cityblock (Manhattan) distance metric.
- **Evaluate Potential New Links:** Every potential new link between nodes belonging to different clusters is considered. This involves evaluating all possible node pairs where one node belongs to one cluster and the other node belongs to the other cluster. By focusing only on inter-cluster links, the number of potential links to evaluate is significantly reduced.

- **Calculate Network Reliability:** For each potential link addition, the network's reliability is calculated. This involves simulating the network with the new link and assessing the improvement in full connectedness. The reliability measure used is the percentage of time the network remains fully connected over multiple simulation iterations.
- **Select Optimal Link:** The link that provides the highest increase in network reliability is selected and added to the network, ensuring that the addition maximizes the overall robustness of the network.

The distance matrix D is defined as:

$$D_{ij} = \text{shortest path distance between nodes } i \text{ and } j$$

The K-means clustering algorithm partitions the nodes into two clusters by minimizing the within-cluster sum of squares (WCSS) using the cityblock distance metric:

$$\text{minimize} \sum_{k=1}^{2} \sum_{i \in C_k} \|D_i - \mu_k\|^2_{\text{cityblock}}$$

where C_k is the set of nodes in cluster k and μ_k is the mean distance of cluster k.
The centroids are initialized using the (default) *k-means++* method [1], which works as follows:

- Choose the first centroid randomly from the data points.
- For each subsequent centroid, choose a data point from the remaining points with a probability proportional to its distance squared from the nearest existing centroid.

More information can be found at: MathWorks documentation.
The algorithm then iteratively assigns each node to the nearest centroid using the cityblock distance and updates the centroids based on the mean of the nodes assigned to each cluster until convergence.
After clustering, potential new links between nodes in different clusters are evaluated. The optimal link is determined by the reliability improvement as for the Brute-Force method. The complexity is given by:

$$\text{Complexity:} \quad \frac{n^2}{4} \quad \text{where} \quad n = \text{number of nodes}$$

The reduced complexity arises because the clustering strategy considers only potential links between nodes in different clusters, effectively halving the number of link evaluations compared to the brute force method, which considers all possible node pairs.

5.3 | Minimal Node degree

The minimal node degree strategy aims to enhance network reliability by focusing on nodes with the fewest connections. This strategy involves identifying nodes with the minimal degree (i.e., the smallest number of direct connections) and evaluating potential link additions between these nodes.
The minimal node degree strategy involves the following steps:

- **Identify Nodes with Minimal Degree:** Determine the degree of each node in the network and identify the set of nodes that have the minimum degree.
- **Evaluate Potential New Links:** Consider all possible new links between nodes with the minimal degree. Since these nodes have the fewest connections, adding links between them is likely to improve their connectivity significantly.
- **Calculate Network Reliability:** For each potential link addition, calculate the network's reliability. The reliability measure used is the percentage of time the network remains fully connected over multiple simulation iterations.
- **Select Optimal Link:** Select the link that provides the highest increase in network reliability and add the selected link to the network, ensuring that the addition maximizes the overall robustness.

The complexity of the minimal node degree strategy depends on the number of nodes with the minimal degree. If m is the number of such nodes, the number of potential new links to evaluate is:

$$nr_{links} = \binom{m}{2}$$

This complexity is generally less than evaluating all possible links in the network, making this strategy more efficient than the brute force method.

5.4 | Diameter reduction

The diameter reduction strategy aims to enhance network reliability by minimizing the longest shortest path between any two nodes in the network. This strategy is particularly effective because reducing the diameter of a graph generally improves its overall connectivity and resilience, making it less susceptible to disconnections.
Why diameter reduction is a good idea?
The diameter of a graph is defined as the longest shortest path between any two nodes. A smaller diameter implies that the network is more compact, which means there are shorter paths between nodes and better redundancy. This results in higher robustness and faster communication within the network.
The diameter reduction strategy involves the following steps:

- **Calculate the Current Diameter:** Determine the current diameter of the graph, which is the longest shortest path between any two nodes.
- **Evaluate Potential New Links:** Consider all possible new links between pairs of nodes whose distance is equal to the diameter.
- **Determine the Impact on Diameter:** For each potential new link, calculate its impact on the graph's diameter. Specifically, determine how much the diameter would be reduced if the link were added.
- **Select Optimal Link:** Select the link that reduces the diameter the most. This link is then added to the network to maximize the reduction in diameter.

Complexity Analysis
The complexity of the diameter reduction strategy is given by:

$$\text{Complexity:} \quad O(1)$$

This complexity arises because the method focuses on adding the link that directly minimizes the diameter without exhaustively evaluating all possible links.

5.5 | Comparison of Link addition methods

Each strategy has its advantages and weaknesses. Depending on the complexity of the method, the trade-off between running time and precision can be exploited in a direction or the other.

- The Brute Force method makes sure that the proposition done is the best in the graph. On the other hand, it is bound to be the most time consuming, and so the least precise in it's computation of the reliability.
- The Clustering method is, from the experience made, always better than the Brute Force, because the proposed link is the same for both methods in the end. It is also quiet time consuming, but a good alternative.
- The Minimal Node degree method is much quicker in terms of computation time. The proposed link can be different from the best proposition made by the Brute Force solution, but the resulting reliability is close enough.
- The Diameter reduction solution is the fastest, as the analysis is only topological. The simulator is only used to measure the effects of it on the reliability of the new network. The major problem of this solution is that it advises to connect two nodes that are topologically far away, but it most often means that they also are physically far away. On the European topology, one such new link proposed is between Oslo and Madrid, which are complicated to connect.

5.5.1 | Running Times

The following table summarizes the running time of each method for the complete topology shown in Fig.1.1. The standard values used are: $\lambda_{\text{link}} = \frac{1}{6}$, $\mu_{\text{link}} = 2$, $\lambda_{\text{node}} = \frac{1}{8}$, $\mu_{\text{node}} = 1$, and $\Delta t = 0.1$, corresponding to 3000 steps over 300 months.

Method	**Initial Reliability (%)**	**New Reliability (%)**	**Time Elapsed (seconds)**
Brute Force	69.42	75.49	133.93
Clustering	69.42	75.49	69.86
Degree	69.42	74.65	11.50
Degree with Clustering	69.42	74.65	7.39
Diameter (with $\Delta t = 0.05$)	69.42	74.65	7.39

Table 5.1: Running time and reliability results for each method.

6 | Another view on the reliability

6.1 | Limits of the measure of reliability

In this project, reliability is defined as the percentage of time the entire graph remains fully connected. This metric is evaluated in discrete time slots, where connectivity is either 0 or 1. However, this approach does not adequately address node failures. When a node fails, it disconnects from the network, automatically reducing graph connectivity to 0 until the node is repaired. Network administrators have limited options in such scenarios, aside from potentially enhancing node robustness.
Nevertheless, when a node fails, other nodes may still be operational and able to communicate directly with each other if alternate paths exist that bypass the failed node. The initial reliability measure fails to consider this network resilience and dependency on individual nodes.

6.2 | Proposition for a new measure

The new method proposed here aims at being more complete in its description of the state of the network at each time frame. Instead of measuring the full connectivity of the graph at each time, the new measure of connectivity is the size of the biggest connected component of the graph. By doing this, an isolated node has a lot less impact on the reliability when it crashes than a central node of the network.

6.3 | Example

Consider the following example: the graph contains two distinct clusters: 1, 2, 3, 4 and 5, 6, 7. To enhance reliability, the most beneficial link to add would be one that connects these two clusters. The key consideration is determining the optimal source and destination nodes for this link to maximize the improvement in reliability.

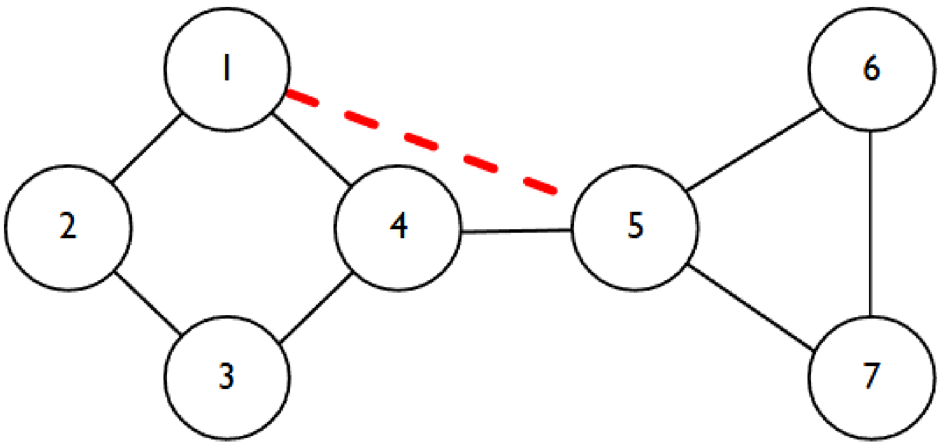

Figure 6.1: Example of link addition proposition

With the first definition of reliability, the first proposition (6.1) might seem acceptable. Yet, as outside designers, it is clear that the network still has a very weak point, since many paths are bound to go through the node 5. If it fails, a good amount of the traffic won't be possible anymore. With the first measure of reliability, the connectivity is null when this node fails, like any other, so it doesn't change anything that traffic could go through another path. It won't make a difference with the second proposition (6.2).
On the other hand, the new measure of reliability will take the weakness of the node into account. When node 5 fails, if everything else works, in the first case the biggest connected component has 4 nodes 1,2,3,4, and for the second case it still has 6 1,2,3,4,6,7. This new measure of connectivity provides better way to propose redundancy.

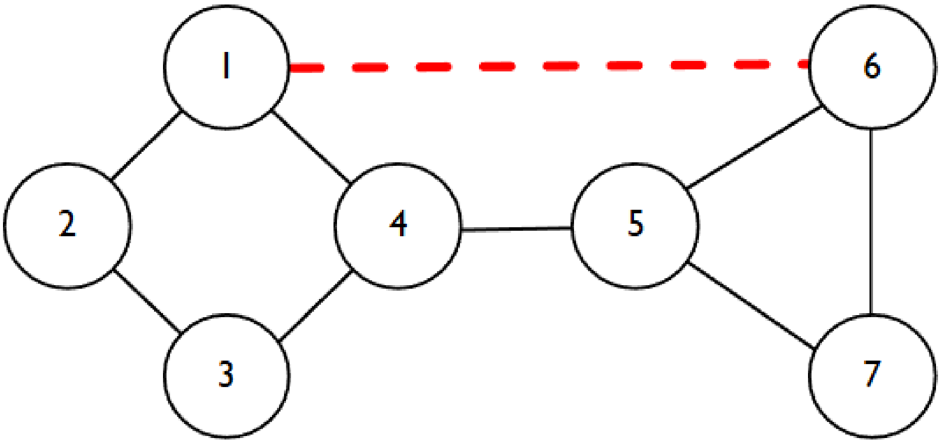

Figure 6.2: Better link addition proposition

With the standard values for the parameters (time = 300 months, steps = 10^5, $\lambda_{node} = \lambda_{link} = 1/6$, $\mu_{node} = \mu_{link} = 2$) the difference in the two measures is obvious. The first network obtains reliabilities of 67.32% and 92.21% with the first and second measure respectively, and the second network got 68.07% and 93.10%. The first measure makes a small difference between the two networks, while the second has a 1% difference.

7 | References

[1] David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding, 2007. Accessed: 2024-07-09.

[2] L.Jereb. *Network Reliability: Models,Measures and Analysis.* Department of Telecommunications, Technical University of Budapest.

[3] Course Slides provided. *Continuous time Markov chains.* Useful material, 2013.

[4] Course Slides provided. *Preliminaries and Counting Processes.* Useful material, 2017.