Handling Link Failure Dependencies in Micro Mobility Network Reliability Modeling

Győző Góдор, Máté Szalay, Sándor Imre

Department of Telecommunications,
Budapest University of Technology and Economics,
Magyar tudósok körútja 2, Budapest, Hungary H-1117
{godorgy, szalaym, imre}@hit.bme.hu
http://mcl.hit.bme.hu/

Abstract: We have developed a program that helps in reliability modeling of tree topology micro mobility networks. The use of this program is restricted to networks where link errors are independent. In this paper we present a method how this program can be used to compute the reliability of networks where link failures are not completely independent. By using our method we examine the effect of link failure dependencies on the reliability of micro mobility networks. We compare the reliabilities of network pairs where the topologies and link error probabilities are the same, the only difference is that in one of the networks link failures are completely independent from each other, in the other one they are not.

1. INTRODUCTION

Nowadays three separate areas: telecommunications, computer sciences and media are converging towards a common so-called “infocom” network. The common aspect of the trends lies in the network layer, where IP will be the common base of the systems. Therefore there is an increasing need for IP mobility. The mobility provided by IETF Mobile IP [1] cannot properly fulfill the requirements in an always on scenario; hence micro mobility is needed to extend macro mobility. There are several micro mobility protocol recommendations introduced in the literature, see [2, 3, 4, 5]. Most of them are based on a physical or logical tree topology network.

The most important and severe weakness of tree topology is its poor reliability. An approach for modeling tree topology micro mobility networks were introduced earlier in [8]. A reliability function was defined and an algorithm was given to compute the reliability function of any tree topology micro mobility network.

In this paper we present a program called TreeRel. Any tree topology micro mobility network can be built by this program, and the performance distribution function of the networks can be computed based on our algorithm.

Although TreeRel handles networks where link failures are independent, in this paper we present a method to compute the performance distribution functions of networks with dependent links.

This paper is organized as follows: In Section 2 an overview of reliability modeling is given. TreeRel, our reliability modeling program is introduced in Section 3. Section 4 deals with dependent link failures in micro mobility networks. The method is explained in Subsection 4.1, the effect of link dependencies and link error probability on reliability are examined in Subsection 4.2 and 4.3. Finally, in Section 5 we draw the conclusions.

2. RELATED WORK

As we model micro mobility networks, we assume that all the traffic is between the gateway of the network (root of the tree) and one of the access points (leaf of the tree).

A simple graph model is used for the networks. Vertices represent network nodes, edges represent links. In our model only links break down, nodes are completely reliably. Our model is still general in the sense that in a tree topology micro mobility network, the effect of a node breakdown is equivalent to the effect of a link breakdown. In a given state, some of the links are up and some of them are down. Our reliability measure will be the following: The performance of the network in a given state is the number of base stations that can reach the backbone. The relative performance is the proportion of the base stations that can reach the backbone among all of the base stations. This relative performance is the reliability measure we are going to use. The relative performance is between 0 and 1. One minus the relative performance is called unavailable capacity. We are going to make plots of probability versus aggregated unavailable capacity. This means that on the x axis the unavailable capacity runs from 0% (no failures) to 100% (complete breakdown), and the plot at a given value shows the probability of the unavailable capacity being equal to or
greater than the given value. At 0% this probability is 1, because the unavailable capacity is always 0% or more [6, 7]. Logarithmic scale can be used for the probability axis. This function is called the performance distribution function in this paper. A lower running plot means lower probability, thus better reliability, and the most important part of the plots is the leftmost part; it covers the best states with the lowest unavailable capacity, and it has the highest probabilities. So the property of the performance function of a “reliable” network is a low leftside.

Earlier in [8] we have defined an algorithm to compute the performance function of any tree topology. The algorithm is simple and efficient, uses two operations to build up the tree topology from simple building blocks, and computes the function recursively while building. This algorithm computes the exact performance function of tree topology micro mobility networks, where link errors are independent from each other.

This algorithm has been implemented in a program called TreeRel.

3. THE TREEREL PROGRAM

The TreeRel simulation program [10] runs under Microsoft Windows Operating System. It was developed entirely in C++ using Microsoft Visual Studio 6.0 [9]. A program written in C++ is more flexible and runs more quickly than programs generated by general purpose simulation tools.

The simulation related and operating system related parts of the source code are in separate modules. This makes porting to other platforms (e.g. Unix/Linux) much easier. The flexible inner data structures make it easy to extend the program to handle different topologies.

Tree topologies are stored in files using standard XML format. The names of the XML tags are specific for the program and the meaning is easy to find out from the name. Using the output of our program as the input for another should not be a problem with basic XML knowledge.

Not only the tree topology, but also the distribution function of the relative performance can be saved in text files. Other programs (such as Microsoft Excel or Mathworks MATLAB) can import these results and make statistics or various plots.

The program has a user-friendly graphical interface. Any tree topology network can be built by using the program. We have two alternatives for building the desired network: automatic generation of random or structured subtrees or manual building.

Link error probabilities can be set in two different ways: by automatic generation or by manual setting. If the link error probabilities are generated automatically, we can choose between two types of distributions: uniform distribution or normal (Gaussian) distribution. Of course, parameters of the distributions can be set manually.

After the desired network topology is built, the reliability measure can be computed. The result, the Performance Distributed Function is shown in a separate window. The diagram can be made smaller, larger; it can be zoomed horizontally and vertically thus the most important parts can be examined. The results can be stored in text files for the later processing.

4. DEPENDENT LINK FAILURE

The TreeRel program cannot handle dependent link failures yet. Although this is a very important part of network planning. For example some base stations or routers are different points of the network topology but their geographical positions are similar (perhaps they are by the side of a fault line). Thus these nodes break down together.

We represent a method with the help of which the dependency between links can be managed easily. For the sake of simplicity we examine a network where exactly two links are dependent, the others are independent from each other. These two links break down together, so either they are both up or they are both down.

4.1 Introducing the Method

Firstly, we compute the performance distribution function of the network where all link error probabilities are independent. This is very simple using the TreeRel program.

The dependency between the two nodes can be realized in the following way:

Since nodes “A” and “B” are dependent, these break down together. Both of them are up or both of them are down at the same time. Therefore, firstly, we calculate the performance vector of the network where these links are down (the link error probability of these links are 1); this vector is $m_{\text{down}}$. After that we calculate the performance vector of the network where these links are up (the link error probability of these links are 0); this vector is $m_{\text{up}}$. And finally these two performance vectors have to be combined.

The two performance vectors $m_{\text{down}}$ and $m_{\text{up}}$ have to be summed in the following way:

The $m_{\text{up}}$ vector has to be multiplied by $(1-p)$ (the probability of being up), and the $m_{\text{down}}$ vector has to be multiplied by $p$ (the probability of being down).

Because of the performance distribution function of the network where all links are independent and the performance distribution function of the network where two links are dependent are very similar, it can be hard to decide which of
the two networks is better than the other. Therefore we composed the difference of these two performance distribution functions.

We subtracted the performance distribution function of the independent network from the performance distribution function of the dependent network. As a lower plot indicates better reliability, in case the function runs below zero the network with dependent links is more reliable than the network which has just independent links. And if the function runs above zero, the network where all links are independent is better than the other.

4.2 Dependent Failure Analysis

The analyzed network is shown in Figure 1, which is a structured tree, the maximum depth is 10 and the branching factor is 2. It is a 10 deep binary tree.

In this experiment all link error probabilities are set to the same value.

Three different classes of cases were examined:
- one of the two dependent links is in the subtree below the other one
- two dependent links at the same depth (10)
- two dependent links at different depths

One of the Two Dependent Links Is in the Subtree Below the Other One

It is easy to see that in this case the dependency has no effect on reliability. Since, if the two dependent links break down, the whole subtree becomes unreachable. Thus the breakdown of the link which is in the subtree, does not have influence on the reliability of the whole network.

Two Dependent Links at the Same Depth

There were ten test networks in this class. The position of one of the dependent links was fixed, the other was at the same depth. All dependent links are at the 10th level. The fixed link was the bottom left one, denoted by “A0” in Figure 1.

The ten test networks are the following:
- network1 is the network where the two dependent links are “A0” and “A1”
- network2 is the network where the two dependent links are “A0” and “A2”
- network3 is the network where the two dependent links are “A0” and “A3”
- network10 is the network where the two dependent links are “A0” and “A10”

We computed the performance distribution functions of these networks using TreeRel. We also calculated the performance distribution function of the network where all links are independent. The link error probability is the same for all links: 0.01.

We illustrated the difference of the distribution functions of the ten networks from the all-independent one in one diagram (see Figure 1) in order to display them in a comparative way.

Since the performance distribution functions of all these ten networks run very close to each other we illustrate the plots of network1 and network10 (see Figure 2). So it can be said that if the two dependent links are at the same depth, the reliability of the whole network does not depend on the positions of the dependent links.

As it can be seen in Figure 2, the difference between the performance distribution functions of the dependent and the independent networks is very small. The difference is several orders of magnitude smaller than the link error probability. And the graphs fluctuate around zero. Therefore it cannot be stated that one network is better in reliability than the other, but the most important part of the diagram for us is the left
side, and there the graph is negative, so it can be said that the networks with two dependent links are more reliable.

**Two Dependent Links at Different Depths**

There were ten test networks in this class, too. The position of one of the dependent links was fixed, the other one was placed at different depths. The fixed link was the bottom left. It is called “B0” (see Figure 1).

The ten test networks are the following:
- network1 is the network where the two dependent links are “B0” and “B1”
- network2 is the network where the two dependent links are “B0” and “B2”
- network3 is the network where the two dependent links are “B0” and “B3”
- network10 is the network where the two dependent links are “B0” and “B10”

We computed the difference between the performance distribution functions of the dependent and the independent networks and we illustrated these in one diagram (see Figure 3) in order to display them in a comparative way. In this case the link error probability is also the same for all links: 0.01.

![Figure 3 - The two dependent links at different depths when p=0.01](image)

It can be seen in Figure 3 that the graphs, which illustrate the difference between the performance distribution functions of the dependent and the independent networks, do not run on each other. Each graph has a negative and a positive peak. The higher the dependent link is, the bigger the peak is and it is shifted more towards the higher unavailable capacities.

As the most important part of the diagram is the left side, consequently it can be said that the higher the dependent link is, the more reliable the network is. So if the two dependent links are at different depths, the reliability of the whole network is depended on the positions of the dependent links.

As it can be seen in Figure 3, the difference between the performance distribution functions of the dependent and the independent networks is very little. The difference is several orders of magnitude smaller than the link error probability. However, in contrast with that case when the two dependent links were at the same depth, the graphs do not fluctuate that much.

Since the negative part of each graph is on the left, it can be said that the networks with two dependent links are more reliable than the network which has just independent links.

**4.3 Introducing the Method**

In this section we examine how link error probability affects the reliability of the dependent and the independent networks. We also consider the changes in the characteristic of the graph which represents the difference between the performance distribution functions of the dependent and the independent networks.

The analyzed network is the same as in Section 4.2 (see Figure 1).

**Two Dependent Links at the Same Depth**

We computed the performance distribution function of the dependent and the independent networks when the link error probability is 0.01 and when it is 0.001, and computed the difference between the performance distribution functions of the dependent and the independent networks in the two cases.

In the case of network10 and when the link error probability is 0.01, the difference between the performance distribution functions of the dependent and the independent networks is shown in Figure 2 (network10).

If the link error probability is 0.001, the difference between the performance distribution functions of the dependent and the independent networks is shown in Figure 4.

![Figure 4 - The two dependent links at different depths when p=0.001](image)

As it can be seen in Figure 2 and Figure 4 the two plots are very similar to each other. The difference between the two...
graphs means that if the link error probability is lower, the fluctuation withers away faster.

In case the link error probability is less the peaks are higher and narrower.

As it can be seen the change of the link error probability has small influence on the plots. The characteristics are very similar to each other.

**Two Dependent Links at Different Depths**

We computed the performance distribution function of the dependent and the independent networks when the link error probability is 0.01 and when it is 0.001. And we computed the difference between the performance distribution functions of the dependent and the independent networks in the two cases.

In the case of network10 and when the link error probability is 0.01, the difference between the performance distribution functions of the dependent and the independent networks is shown in Figure 3 (network10).

If the link error probability is 0.001, the difference between the performance distribution functions of the dependent and independent networks is shown in Figure 5.

![The two dependent links at different depths when p=0.001](image)

**Figure 5 - The two dependent links at different depths when p=0.001**

As it can be seen in Figure 3 and Figure 5, the characteristics of the two curves are very similar to each other. The difference between the two graphs is that if the link error probability is lower, the peak is lankier and the fluctuation withers away faster.

Therefore those characteristics that we have discovered do not derive from the special choice of link error probability, but from the nature of the problem.

5. **CONCLUSIONS AND FUTURE WORK**

In this paper we have presented our algorithm to compute the performance function of tree topology micro mobility networks, and our program that implements this algorithm. Although this program only handles networks with independent link failures, we have shown a method how this program can be used to compute the performance function in case of simple link dependencies. Using this method we have examined the effect of link dependencies on the reliability of tree topology micro mobility networks.

Our future work will include a more automated solution for handling simple dependencies, the handling of more complex and more general dependencies within tree topology networks, and link failure dependency handling in other topologies.

The project is supported by EU project Phoenix and OTKA F042590

**REFERENCES**


