Reliability Evaluation Software for Tree Topology Micro Mobility Networks

Győző Gódor
gyozoke@mlabdial.hit.bme.hu

János Novák
nj203@hszk.bme.hu

Máté Szalay
szalaym@hit.bme.hu

Sándor Imre
imre@hit.bme.hu

Department of Telecommunications,
Budapest University of Technology and Economics,
Budapest, Hungary

Abstract – **Telecommunications, computer sciences and media of today seem to converge to an IP network. Not only IP backbone will be used but also IP access networks. At the same time there is an increasing need for mobility. MobileIP can not provide fast handovers in an always-on scenario. Therefore IP micro mobility solutions are needed. IP micro mobility networks have several special requirements. They have to provide fast handovers and special routing is needed. Most of the IP micro mobility solutions are based upon a tree topology network. The most important weakness of the tree topology networks is vulnerability. After introducing a reliability model for micro mobility networks and an algorithm to compute the reliability of tree topology micro mobility networks we present our program called TreeRel. This program is able to compute the reliability of any tree topology micro mobility network based upon our algorithm. The program is introduced through two examples.**

Index terms – **micro mobility, reliability.**

I. 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 MobileIP [1, 2] 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 [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 [10]. 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 that uses the algorithm to compute the reliability function of any tree topology network.
This paper is organized as follows:

In Section II an overview of Micro Mobility solutions and reliability modeling is given. In Section III we present an algorithm which can be used for efficient calculation of the performance distribution function of any tree topology network. In Section IV our new program called TreeRel is introduced. Using this program arbitrary tree network topologies can be built and their performance function can be calculated. Examples and screenshots provide better explanation of the program. Conclusions and future planes are summarized in Section V.

II. Related Work

A. IP Micro Mobility

The Internet and most telecommunication networks have or will have an IP backbone. In the future IP will be taken to the terminals. At the same time user requirements are changing, there is a growing need for security and mobility. The standard IP mobility solution, Mobile IP (Mobile IPv4 or Mobile IPv6) is not suitable for an always-on scenario with frequent handovers [2]. When handovers are frequent, micro mobility has to be used. Micro mobility provides local mobility within a well-defined area, for example in an access network. While the mobile equipment stays in the same micro mobility domain handovers are handled locally, the Mobile IP home agent or the corresponding nodes are not notified, as the IP address of the mobile node is not changed.
There are several published IP micro mobility protocol recommendations, see [3, 4, 5]. Most of them are designed for

IPv4 but with modifications they can be applied in IPv6 networks, too.

In a micro mobility network the positions of the mobile nodes have to be stored in a database to route packets correctly. Because of routing considerations almost all micro mobility solutions are based on a tree topology network. The root of the tree functions as the gateway to the IP backbone, and the leaves of the tree are the base stations, see Figure 1.

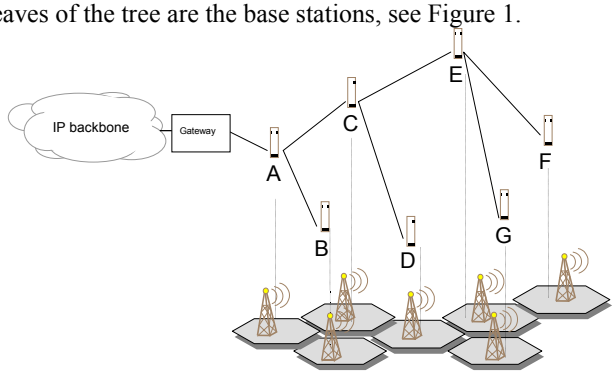

Figure 1. Tree topology micro mobility network

The tree topology suits most of the requirements of a micro mobility network (e.g. efficient routing, scalability), the chief disadvantage is, however, the poor reliability [10]. If a link or node breaks down, a whole subtree is separated from the network. If the gateway breaks down, the whole micro mobility network is separated from the IP backbone.

B. Reliability

Reliability modeling and analysis can be briefly summarized as follows [6, 7, 8]:

- definition of adequate reliability measures,
- determination of the possible states of the network,
- determination of the impact of failures on reliability measures.

A graph model of the network will be used. The edges of the graph represent the links connecting the nodes. In our model only links break down, the nodes are totally reliable. This makes the model much simpler and in a tree topology micro mobility network the breakdown of a node has exactly the same effect as the breakdown of the link that connects that node to its parent. A binary model will be used for unreliable links. The links have two states: up (working) state, and down (broken) state. In our simple model all the links have an independent (and very low) probability of being in the down state.

Reliability means that faults in the system do not degrade the performance of the system too much. To formalize this statement a performance function is introduced as a reliability measure. This performance function tells the performance of the system in a state, [6].

Maximum performance is the value of the performance function in the faultless state of the system. If the performance of the system in a given state is divided by the maximum performance, we get the relative performance.

Given the probability of the states of the network and the reliability measure there are several methods to get the expected value or the distribution of the relative performance. Some of these methods are exact, some are stochastic and some give only upper and lower bounds for the mean value [8].

A reliability measure and an algorithm for getting the exact distribution of the performance for tree topology networks were introduced in [9].

III. THE ALGORITHM

A. The Reliability Measure

The general graph model of the network was given in Section II. In a given state some of the links are up and some of them are down. Our reliability measure for tree topology micro mobility networks will be the following: The performance of the network in a given state is the number of base stations that can reach the backbone.

Let B denote the number of base stations in the network. B^* is the number of base stations that can reach the backbone. The performance is:

$$Perf = B^*$$

The maximum performance is obviously the number of base stations in the network.

$$Perf_{\max} = B$$

The relative performance is the proportion of the base stations that can reach the backbone among all the base stations.

$$Perf_r = \frac{B^*}{B}$$

Figure 2 shows a 4-level binary tree topology network. It has 8 base stations and 7 routers including the gateway.

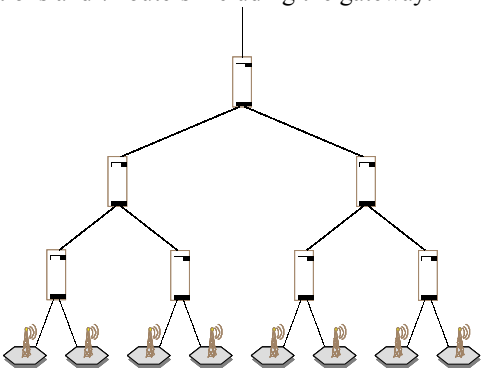

Figure 2. A binary tree topology network

If the link at the first level (the link connecting the gateway to the core network) breaks down, the relative performance is 0, because none of the base stations can reach the backbone. If there is a one-link failure at the fourth level (lowest level link, connecting a base station to a router), the relative performance is 7/8 as only 7 out of 8 base stations can reach the backbone.

B. The Distribution Function

For general tree topology micro mobility networks with the reliability measure described earlier we constructed an algorithm in order to calculate the exact distribution function of the relative performance.

First we define the performance vector $\mathbf{m}=(m_0,...,m_b)$ for each tree topology network where b refers to the number of base stations in the network. The length of the vector is $(b+1)$ and the elements are indexed from 0 to b. The element m_i equals to the probability of exactly i base stations being able to reach the backbone. For example the performance vector of the network Figure 2 with a link error probability of 0.1 is the following:

m=(0.1110, 0.0063, 0.0341, 0.0519, 0.1325, 0.0911, 0.1841, 0.1830, 0.2059)

Note that the sum of the elements of **m** has to equal to 1 and that the distribution function of the performance can be computed easily from this vector, as this vector is actually the density function of performance.

Now we define a recursive algorithm for computing the performance vector of any tree topology network. Complex tree topologies can be built up recursively from simple building blocks.

We apply two basic operations of the algorithm for tree topology networks and for their vectors.

The first operation is joining the roots of two trees to get one tree. Figure 3 shows an example for this operation. The performance vector of the resulting tree is the convolution of the two performance vectors.

For example if the link error probability is 0.01, the performance vectors of the networks in Figure 3 are:

$\mathbf{m}^{(1)}$=(0.0001, 0.0004, 0.0200, 0.0380, 0.9415)

$\mathbf{m}^{(2)}$=(0.0001, 0.0198, 0.9801)

m=(0.0000, 0.0000, 0.0001, 0.0008, 0.0204, 0.0559, 0.9227)

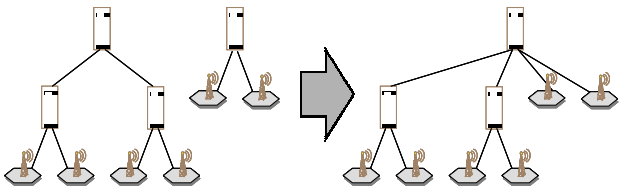

Figure 3. The first operation type: joining the roots

The second operation is adding an edge (a link) to the root of a tree. So the tree will have a new root. Figure 4 shows an example for this second operation. Let p be the error probability of this new link. What happens to the performance vector **m** of the network? If the new link is in up state the performance distribution is the same as it was without the new link; if the new link is down, the performance is zero. To get the new performance vector **m'** of the new network **m** has to be multiplied by *(*1-*p)* then m_0 has to be incremented by p.

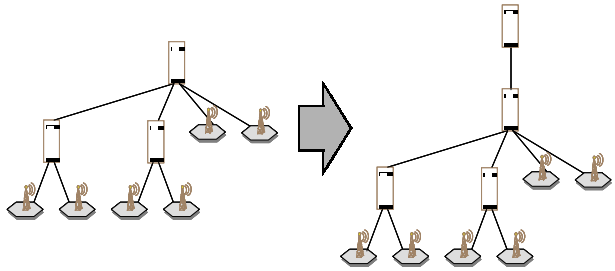

Figure 4. The second operation type: adding an edge to the root

By means of these two operations all tree topology networks can obviously be built from basic building blocks. The basic building block is one base station that is connected to the backbone by a link. If the error probability of this link is p, the performance vector is (p, 1-p).

Figure 5 shows how a general network can be built from these basic building blocks.

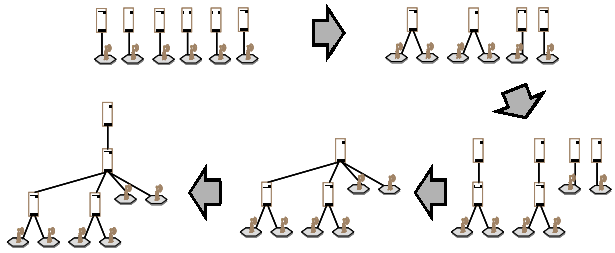

Figure 5. Building a network using the two operations

The reliability function of the network can be plotted, see Figure 6. The (vertical) "Probability" axis is logarithmic, the (horizontal) "Unavailable Capacity" axis is linear.

Better reliability means lower probability for the same unavailable capacity. Lower probability means lower plot. So a lower plot indicates a better reliability.

IV. THE TREEREL PROGRAM

One of the most important issues of network planning is reliability. Therefore much emphasis should be put on its research. However, by analytical methods we can obtain results in case of simple network structures. In case of more complex networks analytical calculation is not applicable thus we have to apply computer simulations.

It is favorable if we can approximate reliability even at the planning phase to avoid having to rebuild the network. With the use of our simulation program the costs can be significantly reduced.
Based on the already introduced algorithm we can easily define the reliability of a tree topology network. This method is applied in our program, which we present in Section III in details and its operation is introduced through two examples.

A. About the simulation program

The TreeRel simulation program [15] runs under Microsoft Windows Operation System. It has been developed entirely in C++ using Microsoft Visual Studio 6.0 [11]. A program written in C++ is more flexible and runs more quickly than a program generated by some 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 develop the program to handle different topologies.
Tree topologies are stored in files with standard XML format [13]. 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. Any XML parser can import these files. Our program uses an XML parser called TinyXML, written by Lee Thomason in C++ [14].
The random variable generation algorithms are based upon [12].
Not only the tree topology can be saved, 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.

B. The functions of the program

The program has a user-friendly graphical interface. Figure 6 shows a screenshot of the program. Each of the network elements is indicated by a different image. The ‘big computer’ on the top left represents the gateway. ‘Little computers’ symbolize the Nodes, whereas the ‘antennas’ symbolize the Base Stations.

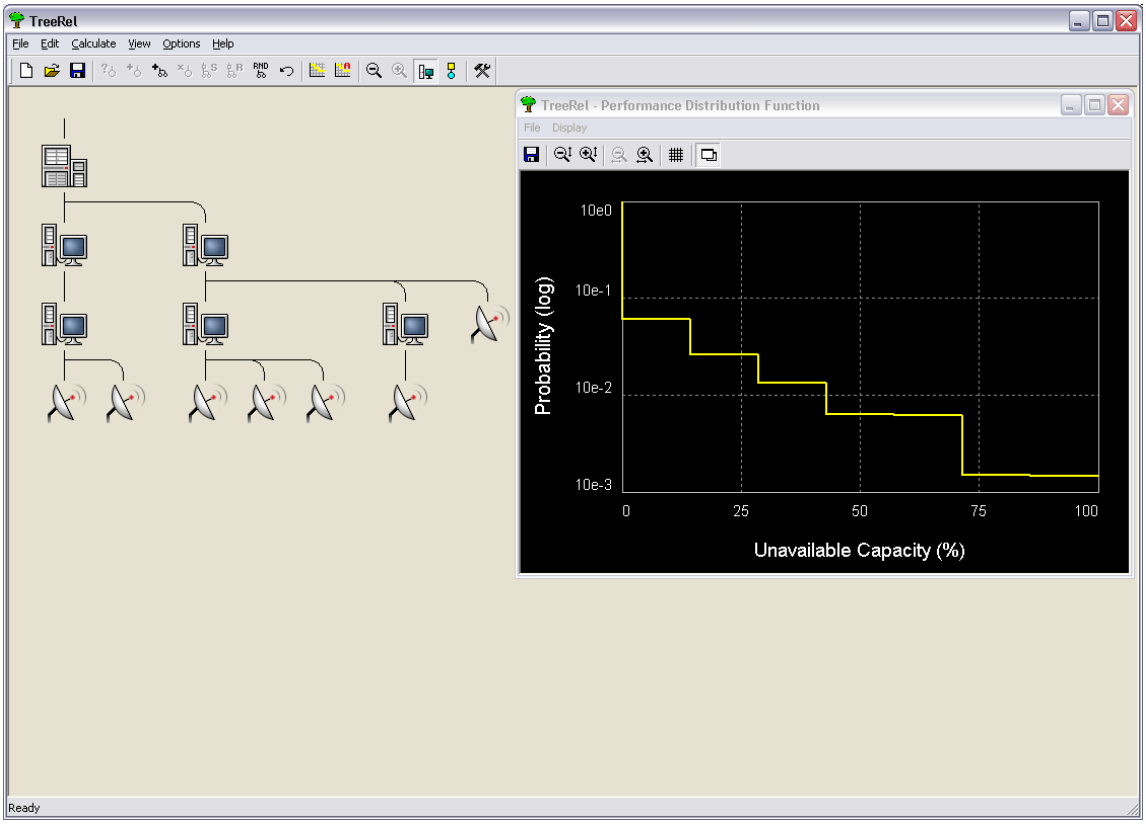

Figure 6. The graphical interface of TreeRel program

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 symmetric subtrees or manual building. In case we choose the automatic generation, we have to set the following parameters: maximal depth, minimal and maximal branching. Parameters can be set in the **Edit → Generate Symmetric Subtree,** and the **Edit → Generate Random Subtree** menus. Figure 7a and 7b show the corresponding dialog boxes.

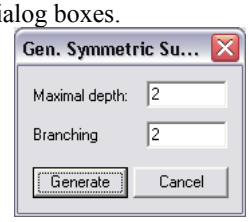

Figure 7a. Generate Symmetric Subtree

Figure 7b. Generate Random Subtree

Link error probabilities can be set in two different ways: By automatic generation or by manual setting. In case 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 distribution can be set manually.

After the desired network topology is built, the reliability measure can be computed in the **Calculate → Performance Distribution Function** menu. The result, the Performance Distributed Function is shown in a separate window (see Figure 6). 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.

When **Options → Auto Refresh Graph** is enabled, the changes in the network affect the performance graph instantly; the button does not have to be pressed each time.

We can get to the previous states by using the Undo button (undo depth can be specified).

The capabilities of the program are explained in more detail through two examples.

C. Example - Reliability of Binary Tree Topology Network

In the first example we examine the effect of the depth of a tree on reliability. It's a very simple example, which can be solved even by thinking logically.

Figure 8 shows the starting network, which is a symmetric tree, the maximum depth is 2 and the branching is 2. It is a 2 deep binary tree. In this tree topology error probability is the same for all the links: 0.01.

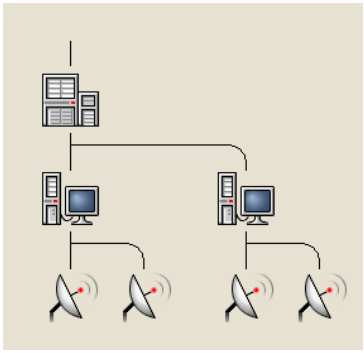

Figure 8. The first example's starting network.

Figures 9 and 10 show the extended networks. Link error probability is the same, they are binary trees with depths of 3 and 4.

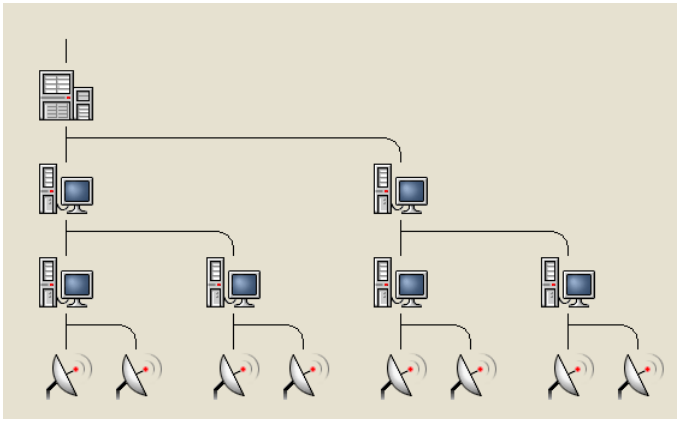

Figure 9. The 3 level deep network

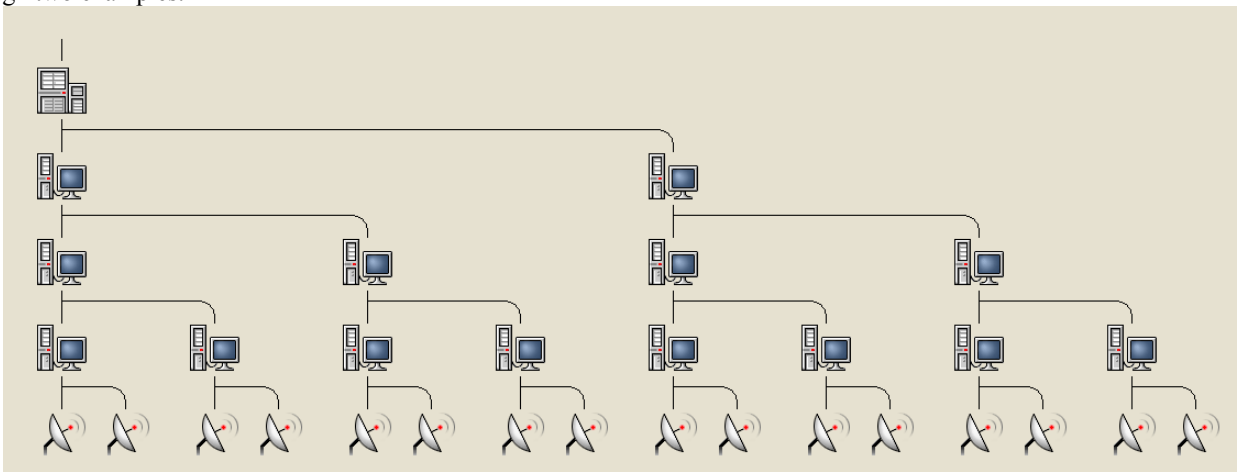

Figure 10. The 4 level deep network

We illustrated the distribution functions of the three networks in one diagram (see Figure 11) in order to display them in a comparative way.

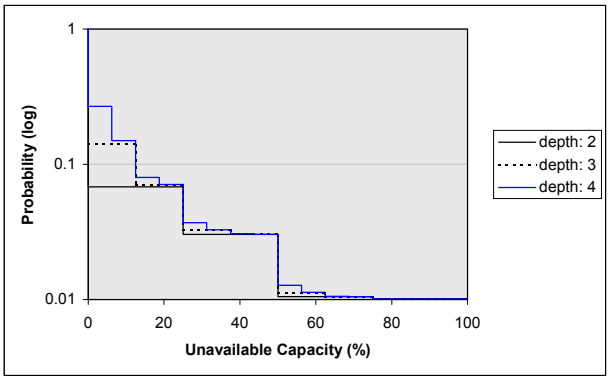

Figure 11. Reliability functions of 2, 3 and 4 deep binary trees

If the depth of the network is increased, the reliability is decreased (i.e. the plot is moved up). In fact, it is not surprising since by making the tree deeper, base stations moved further away from the gateway.
The aim of this example was to prove that our model is suitable, and the program works perfectly.

D. Example - More Complex Tree Topology

The second example is much more complex than the first one. This network is a random tree topology, the depth is 3 and the link error probability is normally distributed ($m = 0.01$ and $\sigma = 0.005$), see Figure 12. Thus the calculation of the reliability function of the starting network is very difficult without the program.
Consider a smaller subnet (see Figure 13.). This subnet would be attached to one of the three numbered nodes of the network of Figure 12. We would like to find the new network with the best reliability features. Manual calculations would be complicated and time-consuming since each case we have to calculate performance distribution function for the new network. However, the simulation program can solve it in a few minutes (including the building of the networks).

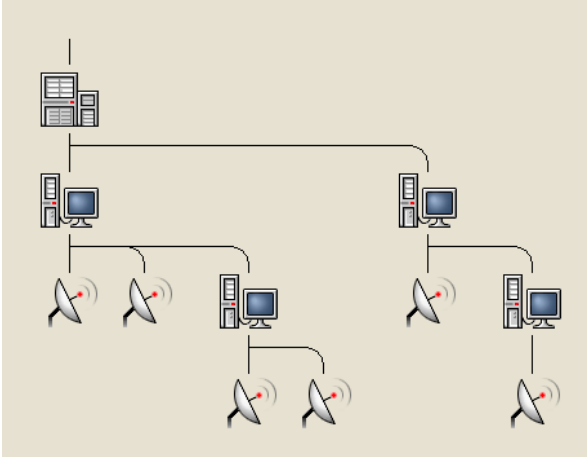

Figure 13. Network to be attached

For illustration we show the network which is attached to the second node (see Figure 14).

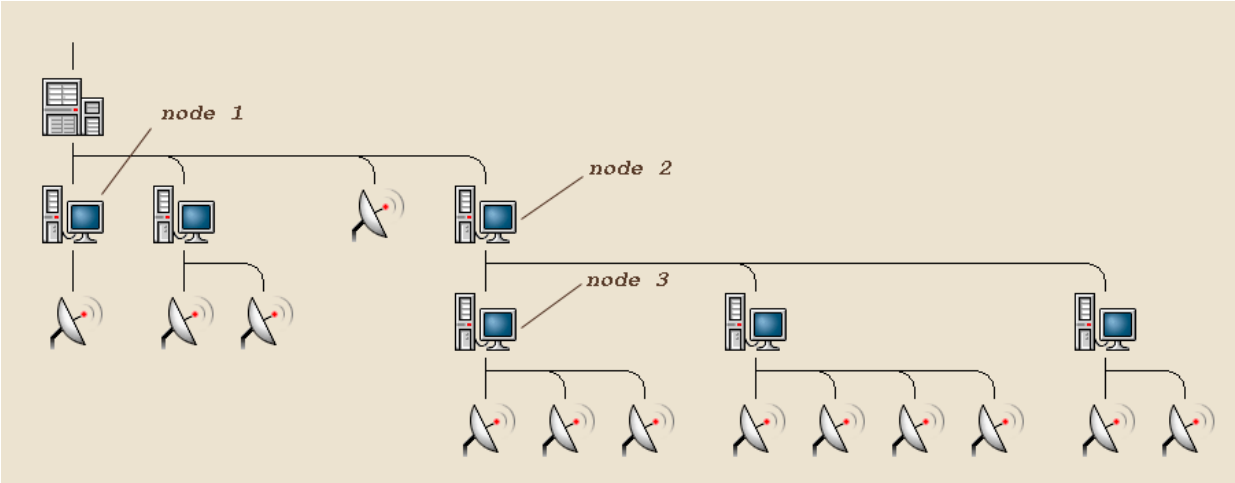

Figure 12. The main network of the second example

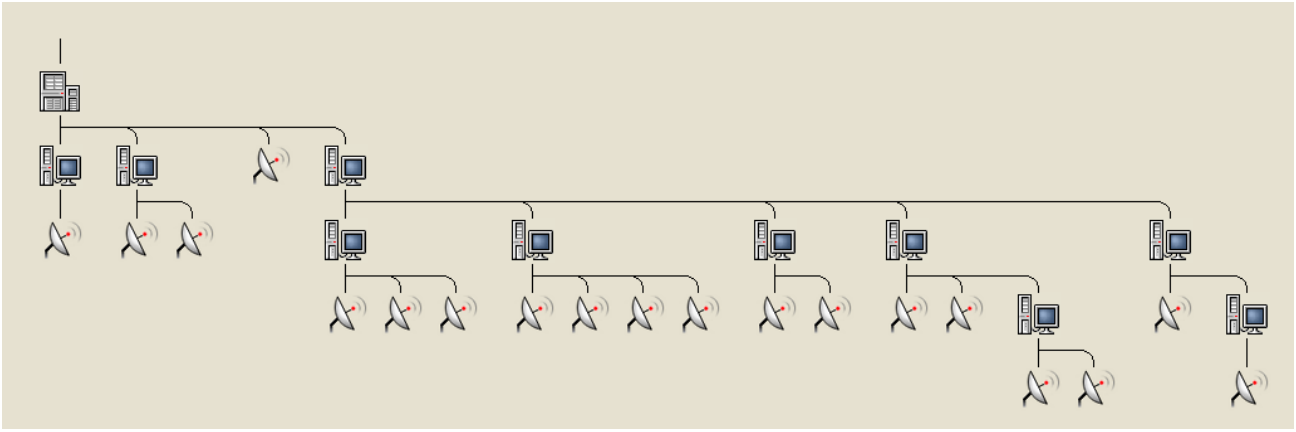

Figure 14. Small network attached to node 2

The performance distribution function of all the three networks is depicted in Figure 15.

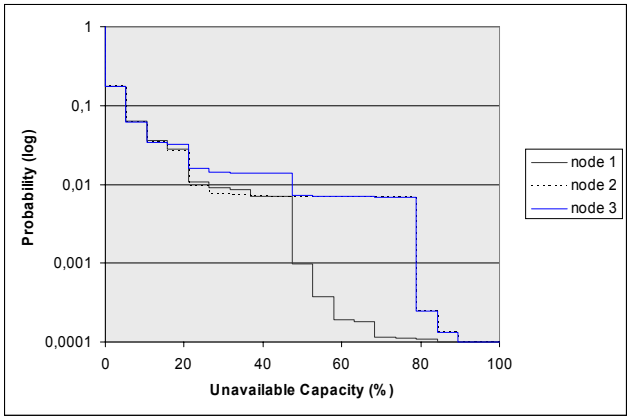

Figure 15. Reliability functions when the small network is attached to node 1., 2. or 3.

The left side of the graphs is the most important part, because the states there have a much higher probability than other states and those are "best" states of the network where the performance is the highest.
When evaluating or comparing plots, we have to concentrate on the left side.

Comparing each performance distribution function it can be shown that the second one is the most reliable (the left side of the diagram is been significant for us). In this network each node has many subnets thus if one of the nodes breaks down, just a smaller part of the whole network would be unavailable.
In the first case the reliability of the new network can be considered good. A lot of nodes have smaller subnets or leaves thus if such a node breaks down, a smaller part of the whole network would be unavailable, too.
The third network has the worst reliability. Since this network has two nodes, which have a big subnet, so if one of them breaks down, a bigger part of the network would be unavailable.

V. Conclusions and Future Work

We have presented a micro mobility specific reliability measure and introduced an algorithm for its calculation. We introduced a program that uses our algorithm and computes reliability functions of tree topology micro mobility networks.
We introduced the main functionality of the TreeRel program and explained its operation through two examples. However, these examples were used for illustration. This program can be used in a wide range of applications, reliability of micro mobility networks can be evaluated even at the planning phase saving time and money.
In the future other network topologies should be implanted. Future plans also include porting to other platforms (e.g. UNIX/LINUX).

Acknowledgment

The project is supported by OTKA F042590 and COST279

REFERENCES

[1] C. Perkins, "IP Mobility Support", IETF RFC 2002, http://www.ietf.org/rfc/rfc2002.txt, 1996

[2] B. Gloss, C. Hauser, "The IP Micro Mobility Approach", EUNICE 2000, September 2000, Eschende pp. 195-202.

[3] A. T. Campbell, J. Gomez, C. Y. Wan, S. Kim, Z. Turanyi, A. Valko, "Cellular IP", draft-ietf-mobileip-cellularip-00.txt, IETF Internet Draft, 1999

[4] R. Ramjee, T. La Porta, S. Thuel, K. Varadhan, "HAWAII: A Domain-based Approach for Supporting Mobility in Wide-area Wireless Networks", Seventh International Conference on Network Protocols, Toronto, Canada, 1999

[5] Claude Castelluccia, HMIPv6: A Hierarchical Mobile IPv6 Proposal. ACM Mobile Computing and Communication Review (MC2R), April 2000 issue

[6] L. Jereb, "Efficient Reliability Modeling and Analysis of Telecommunication Networks", 6th International Conference on Telecommunication Systems, 1998, Nashville

[7] L. Jereb: "Network Reliability: Models, Measures and Analysis", in Proceedings of the 6th IFIP Workshop on Performance Modelling and Evaluation of ATM Networks, Tutorial Papers, Ilkley, UK, 1998, pp. T02/1-T02/10

[8] C.J. Colbourn, "Reliability issues in telecommunications network planning", in: Telecommunications Network Planning (B. Sanso, ed.) Kluwer Academic Press, 1999, pp. 135- 146.

[9] M.Szalay, S. Imre "Reliability Modeling of Tree Topology IP Micro Mobility Networks", Eunice 2002, Trondheim, Norway, September 2-4, 2002, pp. 63-69.

[10] S. Imre, M. Szalay: "Reliability Considerations of Micro Mobility Networks", DRNC2001, Budapest, Hungary, October 7-10, 2001.

[11] Microsoft Visual C++ 6.0, http://msdn.microsoft.com/visualc/

[12] György Pongor: "Performance Analyzis of Telecom Networks Based on Simulation I.", published by BME, 2001, pp. 17-20, 27-30.

[13] W3C Recommendation: "Extensible Markup Language (XML) 1.0 (Second Edition)", http://www.w3.org/TR/REC-xml, 6 October 2000

[14] Lee Thomason: "Tiny XML XML parser", http://www.sourceforge.net/projects/tinyxml

[15] Győző Gódor, János Novák, Máté Szalay, Sándor Imre: "Reliability Analysis of Tree Topology IP Micro Mobility Networks with Simulation", Student conference, 12 November 2002