Application of Cosmos Tool in the Education on Communication Networks

Robert Inkret, Marko Lacković* and Marije Ljolje
Faculty of Electrical Engineering and Computing, Unska 3, HR-10000 Zagreb, Croatia
**École Polytechnique Fédérale de Lausanne, CH-1015 Lausanne, Switzerland*

Abstract
The paper presents application of Cosmos framework in the education on communication networks. It gives a brief overview of Cosmos and main steps in application development. Of the four presented application modules three illustrate the concepts at different layers of communication networks. These are optical burst switching, gigabit Ethernet and TCP. An application that illustrates availability and reliability-related concepts is presented.

1 Introduction

When teaching students about different concepts applied in communication networks, their visual experience is of great value. This experience can be provided either by real systems or software aids. Application of real systems has many disadvantages. For example, building a system set-up on laboratory scale can be unaffordable. In addition, configuring such a setup is commonly impractical and industrious. Finally, measurements on real systems usually produce large amount of data to be analyzed, most of which are irrelevant to the studying of the concepts under consideration.

Clearly, a hands-on experience is often invaluable to the teaching process. However, in many scenarios only visual educational aids are acceptable due to different practical and financial reasons. The main advantage of such tools is that they allow for an abstraction of only relevant details and hiding of all irrelevant details.

Our paper presents four educational tools. The first three deal with the protocols at different communication layers. We start with a physical layer and the tool that enable analysis of optical burst switching (OBS), different signaling and scheduling algorithms, and protocols used within OBS paradigm. Then we move on to the 2nd layer and Gigabit Ethernet, and finally conclude with the layer 4 transmission control protocol (TCP). At the end of the paper we also present a tool for teaching availability and reliability-related concepts.

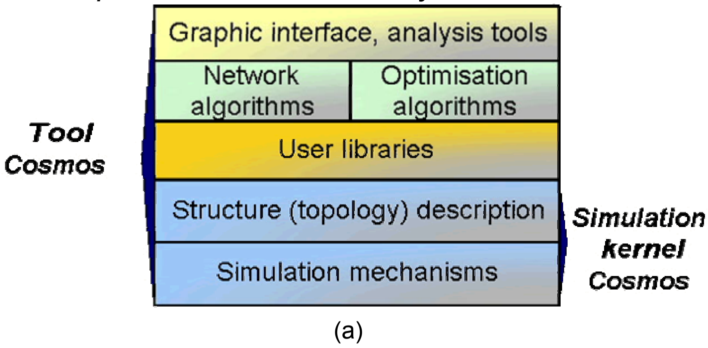
All presented educational tools are built on top of Cosmos software development framework [1]. This is why we start with a brief overview of Cosmos.

2 Cosmos architecture

Cosmos is the abbreviation for "Complex System Modeling and Simulation". Figure 2-1(a) shows its structure. The main parts of Cosmos are:

- *Simulation kernel* - provides minimal functionality necessary for network and model (component) description and simulation.

- *User libraries* - contain user provided descriptions of simulation models (components). User libraries facilitate designing in that the user does not have to provide model (component) description.
- *Libraries of network and optimization algorithms* are used for network analysis and optimization.
- *Graphic interface and analysis tools* are essential for a user-friendly tool.



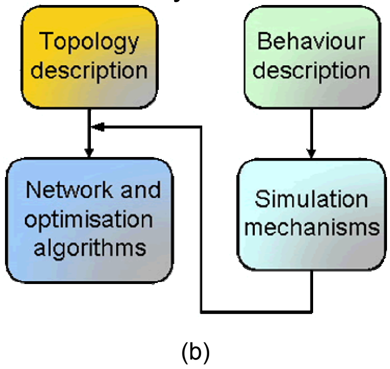


Figure 2-1 Structure of Cosmos tool (a) and steps in network representation in Cosmos (b)

There are two main steps in describing a network using *Cosmos*: description of topology and of behavior. Topology description defines a passive network structure, while behavior description defines a simulation model. Topology itself is sufficient for some analysis and optimization, but simulation requires both steps. Simulation results can serve as input data for the analysis and optimization algorithms (Figure 2-1(b)).

Cosmos simulation kernel provides basic elements for network topology description and the framework for behavior description. Simulation mechanisms implemented in kernel determine the calls of predefined functions from the kernel, while a user has to provide the body for these functions i.e. behavior (model) description.

3 OBS application module

3.1 Introduction and motivation

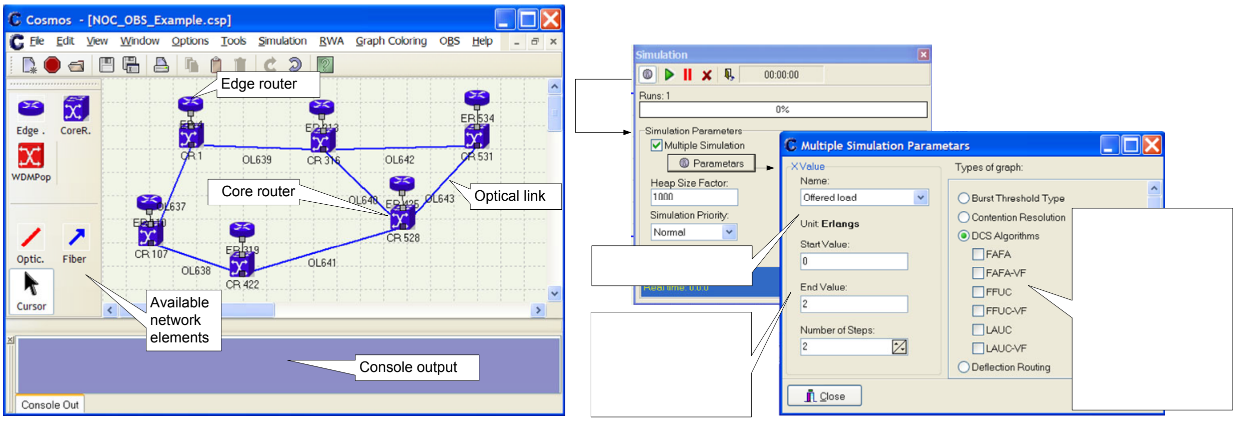
In optical burst switching, transmission of control information is separated from data transmission. This avoids bottlenecks in the control plane raised by limited switching rate in the electrical domain. A fundamental transport unit, data burst, is composed of a number of packets. Control information needed for routing data burst to a destination is located in a separate packet called BHC (Burst Header Control) or BHP (Burst Header Packet) which is routed over the control part of a network. OBS is based on the one-way reservation protocol; before sending data burst, BHC packet is sent to a destination in order to reserve capacity along the way, and data burst is sent without waiting for the acknowledgment of successful reservation process. Data travels through a network only in the optical domain.

At first, the objective of OBS application module was to thoroughly investigate OBS architecture and other related issues. Afterwards, it became obvious that the application had its use in the educational process as well.

3.2 Architecture and application usage

Architecture of OBS network elements implemented in the application is described in [5]. Figure 3-1(a) shows graphical user interface of OBS application and simple OBS network.

In OBS application, a network node consists of a *core router* and *edge router*. Core routers, which are interconnected by optical links, represent a backbone of OBS network, while the edge routers represent the access networks.

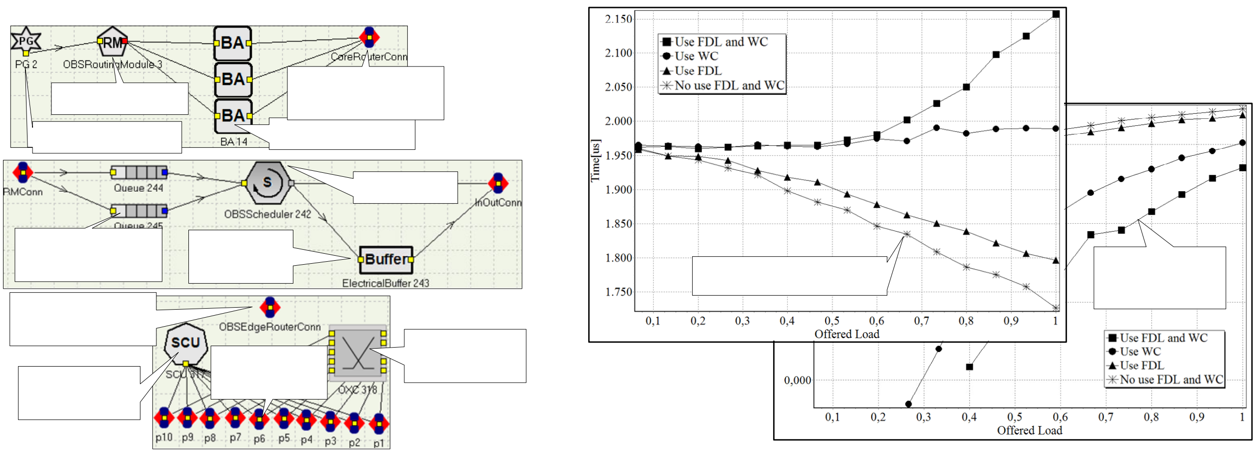


(a) (b)

Figure 3-1 GUI of OBS application module (a) and simulation settings (b)

3.2.1 Network elements

Edge router, with the inner structure shown in Figure 3-2(a), has four components: *packet generator* (simulates traffic source), *routing module* (performs packet routing), *Burst Assembler* (BA, creates bursts from a packet stream) and *Input-output port* (for connection to the core router). BA, located inside the edge router, has a complex structure, shown in Figure 3-2(b).



(d)

Figure 3-2 Inner structure of edge router (a), burst assembler (b) and core router (c)

BA is further composed of three modules; *CoS queues* (collecting packets based on the class of service), *Scheduler* (responsible for burst creation and resources reservation) and *electrical buffer* (keeps created bursts).

Core router, structured as shown in Figure 3-2(c), comprises *Switching Control Unit – SCU* (responsible for processing BHC packet and resource allocation), *Optical Cross Connect* (OXC), and *Connection ports* (core router connections with optical links).

3.3 Simulation and analysis

OBS application offers possibility of setting-up various network configurations. This is achieved in OBS main system property window. Also, it is possible to compare and analyze configurations through the event-driven simulation. Some of the network options that can be analyzed are: different contention resolution schemes, different data channel scheduling algorithms (algorithms for allocation of data channels to bursts), and different network element structures.

In addition to the listed options, simulation analysis can be run for different parameters, such as network load, burst size, optical buffer size, etc. (Figure 3-2(d)). Simulation parameters are chosen in the window shown in Figure 3-1(b).

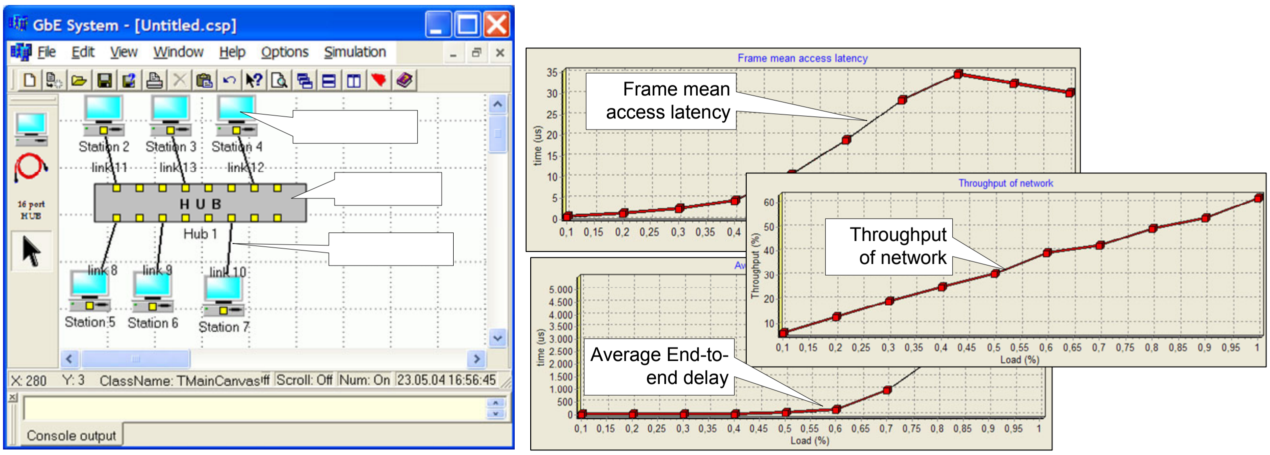
4 Gigabit Ethernet application module

4.1 Introduction and motivation

Gigabit Ethernet (GbE) application is probably the best example of how easy it is to develop graphical application within Cosmos framework. The application was built during a half-semestral student project (a month and a half). The goal was to illustrate and analyze the extensions to the original CSMA/CD protocol used in Ethernet standards in the gigabit environment. The application also offers performance analysis.

4.2 Architecture and application usage

Main window of GbE application module is given in Figure 4-1(a), showing the working area of GbE system in which simple GbE LAN is constructed and composed of GbE Hub and GbE network stations. All stations are connected to the hub via optical link module (component). The function of optical link in the application is to introduce propagation delay. Properties windows are used to set MAC address, offered load, length of a link and other parameters.



(a) (b)

Figure 4-1 Main GUI window of GbE module (a) and simulation results (b)

Simulation analysis includes monitoring of various parameters (e.g. network throughput, end-to-end delay, collision probability and buffer mean waiting time versus network load (Figure 4-1(b)).

5 TCP application module

5.1 Introduction and motivation

For several reasons transport protocol is often considered the most important part of the whole network architecture. Firstly, it is central in data exchange between the remote hosts. Transport protocol represents the lowest peer-to-peer connection. Its main responsibilities are to provide a reliable service, to relieve upper layers from worrying about communication and architectural details of the lower layers, and, on the other hand, to convert data to a form convenient for transportation across the network.

There are several issues every transport protocol should include in order to provide the expected service. These are: addressing, multiplexing, flow control and connection establishment/termination.

The purpose of TCP application module (also known as TCP simulator) [4] in our work was to serve as an attractive teaching tool for explaining main TCP features to the undergraduate students with basic knowledge of network architecture and the role of transportation protocol in it.

5.2 Application Architecture

The main window of TCP module (Figure 5-1) contains a network as seen from TCP layer. The example in Figure 5-1 (left window) shows four stations (computers), fully interconnected and making a mesh. The connections represent all possible TCP connections opened, used and closed during simulation.

The only module that can be created with the toolbar is TCP station, with a communication stack inside. It comprises two modules – *TCP application* and *TCP entity* (Figure 5-1, central window). TCP application serves as traffic generator for TCP and a destination for traffic egress from TCP.

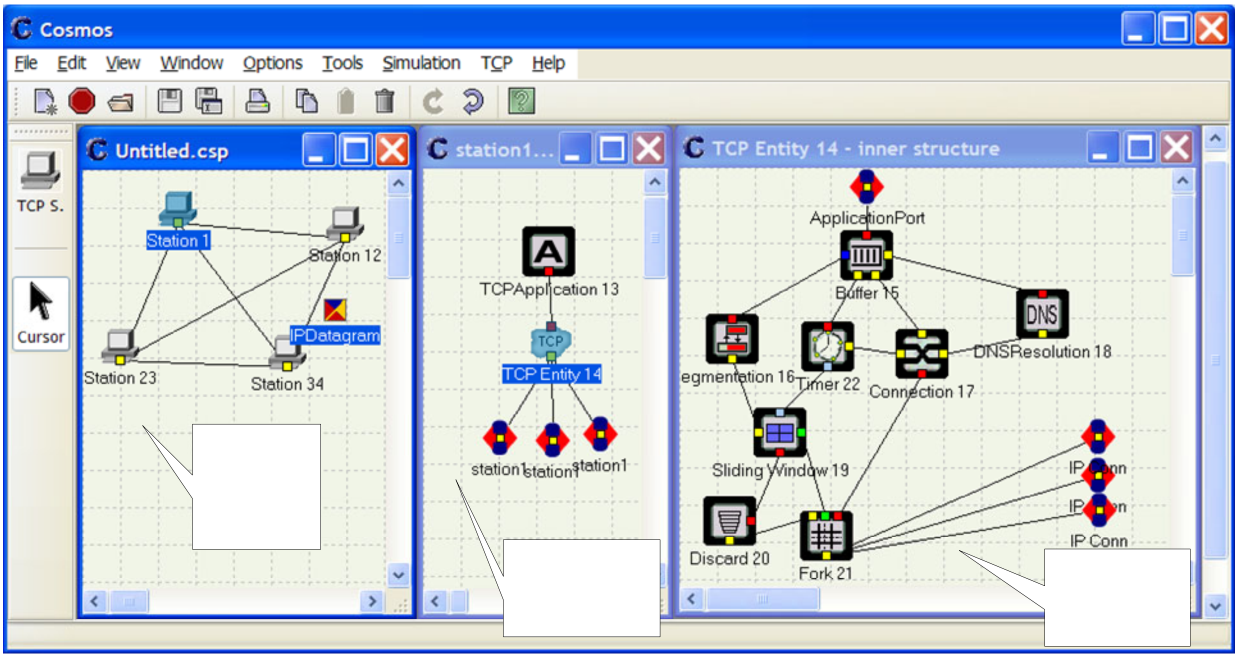
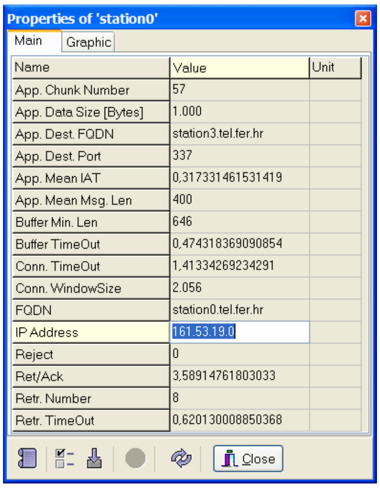


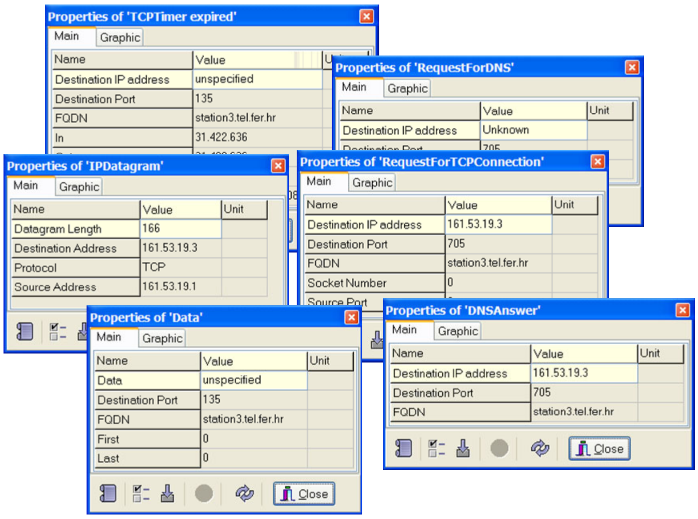
Figure 5-1 - User interface for simulator parameterization

A destination station is determined for each source station, implying that each will make one TCP connection during simulation. Besides FQDN (Fully Qualified Domain Name destination), the higher layer parameterization includes a destination application port and application traffic description (the number of chunks to be generated and the average chunk length) (Figure 5-2(a)).

Other TCP properties include initial window sizes (the sliding window mechanism) that are to be advertised by the senders, optimal data sizes that are to be transferred using specified connections, and the number of retransmissions of the same segment before closing up connection. The failures are simulated using segment rejection percentage (percentage of segments rejected by the receiver).



(a)



(b)

Figure 5-2 (a) Properties window for a TCP station and (b) properties windows of various messages that can be viewed by pausing the simulation

TCP itself was modeled using 8 modules (right window, Figure 5-1): *Buffer* (stores the input data stream and performs data aggregation/division), *DNS Resolution* (converts FQDN destination to IP address), *Segmentation* (adds header and pseudo header to the data of appropriate size that arrived from a buffer), *Timer* (implements five timers: timer for buffer, 2MSL timer used during connection termination, connection establishment timer, acknowledgement timer and retransmission timer), *Connection* (performs connection establishment), *Sliding window* (implements the sliding window protocol (flow control)), *Discard* (deletes TCP segments delivered to a wrong host or the segments damaged during transportation) and *Fork* (directs the input traffic to appropriate module analyzing the segment's flags and detecting connection phase).

During simulation of data transfer on TCP layer it is possible to turn on the animation feature displaying all messages included in communication (e.g. IP datagram in Figure 5-1). By pausing simulation it is possible to view the properties of these messages (Figure 5-2(b)) and to get a closer view of how TCP works.

6 TSAR application module

6.1 Introduction and motivation

System availability and reliability are primary parameters that should be taken into consideration when designing and implementing a telecommunication system. Over the past decades, following the growth of consumer needs and dramatic increase of exchanged data, these systems have grown in complexity and price. With TSAR application module [2], our goal was to design and implement a simple, yet flexible, model of a component behavior considering failures and repairs, and to analyze behavior of a system that comprises such simple modules. The tool was named TSAR – *Tool for System Availability and Reliability* analysis.

6.2 Architecture and application Usage

The whole model was implemented as a Markov chain, or a state and transition diagram with transition probabilities, because of availability/reliability (A/R) definitions. Figure 6-1(b) depicts a model comprising three states. Transitions between the states are marked with (*xy*) where *x* denotes initial, and *y* denotes final state.

Figure 6-1(a) represents general system structure. *S* and *T* modules are single state modules that are always working properly and serve as edge points for the system A/R definition. All system modules have at least one direct or indirect connection to *S* and *T* modules. After composing a desired system structure it is possible to perform A/R analysis by simulation or analytically [6].

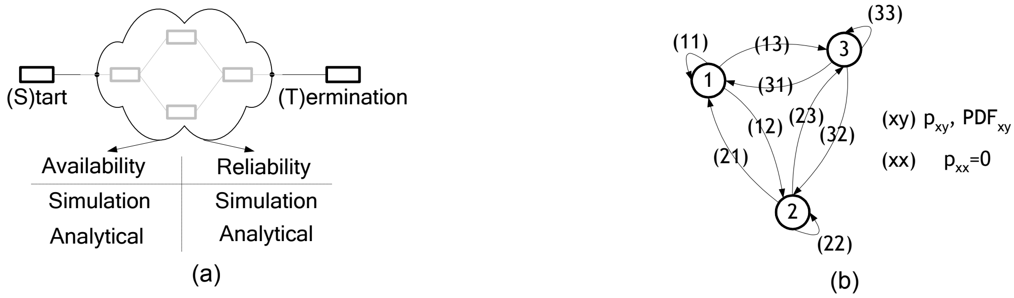


Figure 6-1 General system structure (a) and Markov three state model of module (b)

Figure 6-2(a) shows the main window of TSAR application. The example presented in the figure shows a simple network for which it is necessary to calculate A/R of communication between *S* and *T* nodes. The figure also shows that any module can have a complex inner structure built of available components (Link 7).

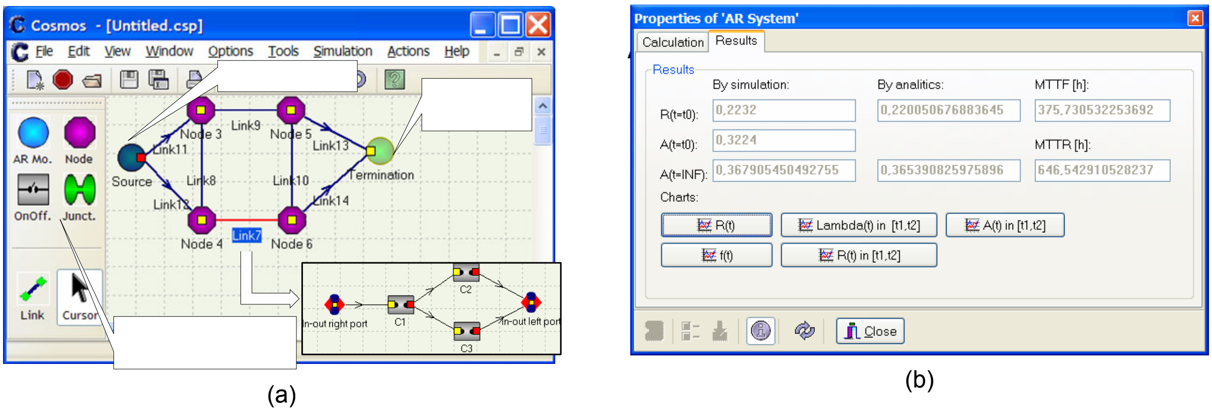
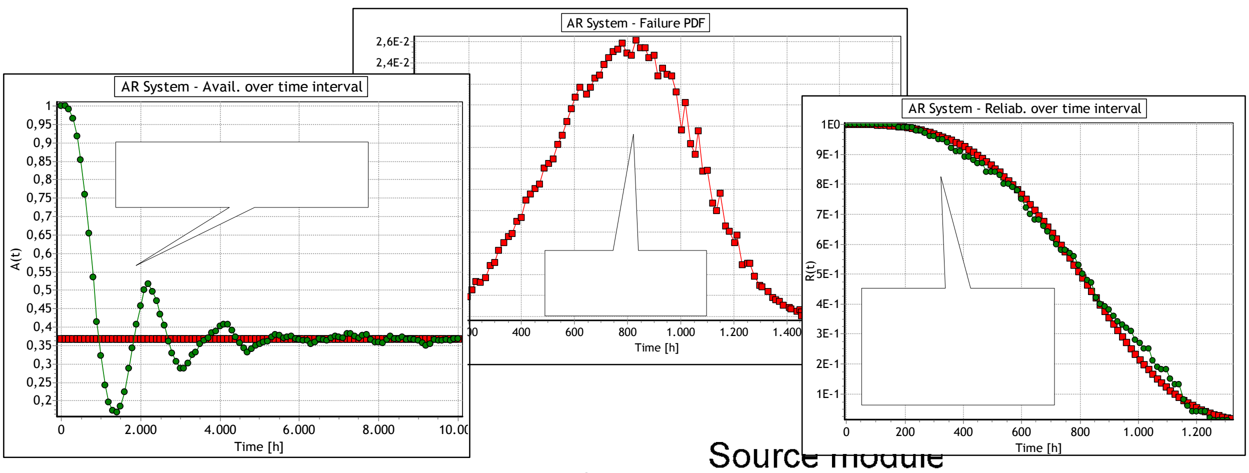


Figure 6-2 Main TSAR window (a) and system properties (b)

From calculation and simulation results it is possible to see how A/R of the system behaves in time, difference between the results obtained analytically and by the applied simulation, and empirically (simulation) obtained distribution of the system's failure and repair time (Figure 6-2 (b) and Figure 6-3).



Source module

Figure 6-3 Resulting charts of simulation and analytical calculation

Termination module

7 Conclusions

In the paper we presented four different educational tools based on Cosmos framework. This framework is built in a way that allows fast and easy creation of different tools for design and simulation. The main steps in the process are structural and behavioral description of the concepts that will be analyzed in the simulation and, finally, if required, the choice of their visual representation.

The ease of application development allows the use of Cosmos framework in different students' projects. The framework is designed for under- and postgraduate students in learning on the ready-to-use examples, in creating own case studies, and at the highest level of tool usage where the students create new applications by implementation of different concepts in communication networks, analyzing performances and comparing different architectures, techniques protocols and protection scenarios.

8 References

[1] M. Lackovic, R. Inkret, Network design, optimization and simulation tool *Cosmos*, *Proceedings of WAON*, June 13-14, 2001, Zagreb, Croatia, pp. 37-44.

[2] M. Lackovic, M. Ljolje, R. Inkret, TSAR - Tool for System Availability and Reliability Analysis, *Proceedings on EUROCON* , Ljubljana, Slovenia, September 22-24, 2003,

[3] J.A. Abraham. An improved algorithm for network reliability, *IEEE Transactions on Reliability*, R-28:58-61, April 1979.

[4] M. Lacković, R. Inkret, M. Mikuc, An approach to education oriented TCP simulation, SoftCOM 2002, Oct. 8-11, 2002, Split-Venice-Ancona-Dubrovnik, pp. 181-185,

[5] Marije Ljolje and Branko Mikac, Architecture and Performances of Optical Networks with Burst Switching, *Proceedings of Mipro 2003, 26th International Convention*, pp. 93-98, Opatija, Croatia, May 2003.

[6] R. Inkret, *Availability Modeling of Multi-Service All-Optical Transmission Network*, PhD thesis, Zagreb, 2004.