

Network Simulation: Tasks and Methods of Their Solution

Nina A. Podolskaya

Abstract—Simulation of networks is an actual task for different areas of informational technologies. It may be used for network control algorithms evaluation, for software validation, for network performance estimation etc. Task of network simulation is usually reduced to the task of network protocols implementation. Peculiarity of network simulation is that network protocols are already implemented in operating system so these protocols not only can be implemented in simulation software but also their implementation in operating system may be re-used. The latest way implies network devices inclusion into simulation process. These network devices may be real or virtual. All these opportunities and their combinations provide a variety of possible methods for network simulation task resolution. The choice of the method is to be defined by different parameters: desired simulation accuracy, desired speed of simulation, the resources available etc. This paper describes different network simulation tasks arisen in semiconductor industry and the methods chosen by the authors for these tasks resolution.

Index Terms—Network simulation, network control algorithm, network protocol stack, WiFi, TUN/TAP.

I. INTRODUCTION

Task of network simulation arises from very different areas of information technologies because simulators let measure parameters of the network that does not exist in reality.

For example while configuring of the network consisting of huge number of computers it is more simple to check parameters of the concrete configuration on simulator and to change configuration if these parameters don't correspond to the expectations, than to solve problems of the network in reality by re-routing wires and by re-installing switches and routers.

One more example of simulation task appearance is necessity of validation of network software that is to be run on non-existing computer platform. This task emerges in the process of development software (firmware, BIOS, drivers) for new computer platform that is still under design. It is well known that for shortening production time development and testing of software for new computer platform are performed in parallel with this new platform design and production. As development of simulator is much faster than corresponding hardware production software developed for running on this hardware is to be validated on the simulator.

Thus the task of simulation of the network arises from the necessity of network parameters estimation in case of lack of network devices or in case of difficulties of these devices usage.

Networking systems are divided into layers that form stack of protocols. According to the model of Open System Interconnection (OSI) project of the International Organization for Standardization (ISO) [1] the hierarchy of the network protocol stack layers is the following:

- 1) Application
- 2) Presentation
- 3) Session
- 4) Transport
- 5) Network
- 6) Data link layer
- 7) Physical layer

The upper layer is application layer. Data sent by application into the network goes down subsequently through all layers of the protocol stack and at the physical layer proceeds to the stack of network protocols of the other operating system. Then it subsequently goes up through the other stack from the physical layer and reaches the application for which it was destined.

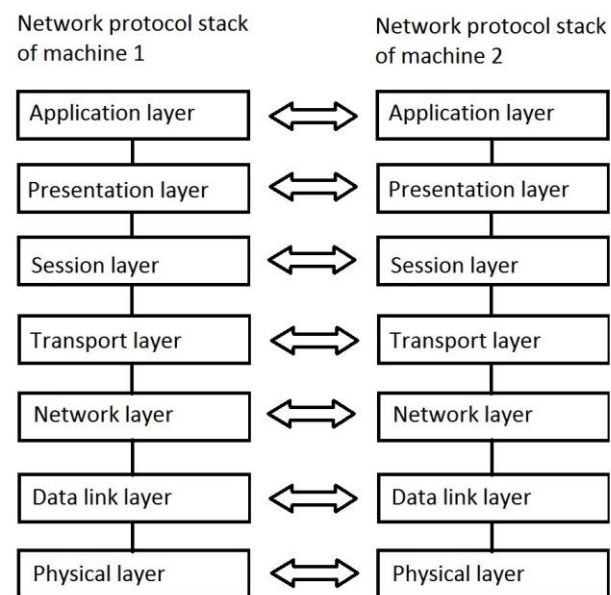


Fig. 1. Network protocol stack layers interactions between two operating systems.

In this way network communication between two instances of operating systems appears to be an interaction of corresponding layers of network protocol stacks (Fig. 1). Thus two applications running on two different machines communicate with each other on application level that is the upper level of network protocol stack. Operating systems are responsible for transmission of the data sent by one of the applications and destined to the other application. During the transmission process data packet goes down through all the layers of the network protocol stack. At some of these layers it may be cut to several pieces that are more adjacent for this layer, and also may get extra data which contains information

that helps two operating systems transmit data properly.

According to described mechanism of network operating the task of network simulation is usually reduced to the task of implementation of number of network protocols. Peculiarity of network protocol simulation in comparison of any other objects of simulation is that network protocol is already implemented in operating system, namely in OS network protocol stack. Therefore each protocol may be implemented in two following ways: 1) entirely in software or 2) by using this protocol implementation in operating system. Lack of the first method is its unreliability due to possible difference with its implementation in operating system. The second method provides more reliable results but presence of significant number of outsider frames impedes the data analysis in simulator and also implies non-determinism of simulator work.

This paper considers examples of both methods usage and suggests an alternative approach to the problem consisting in combining these two methods.

II. SOFTWARE IMPLEMENTATION OF NETWORK PROTOCOLS

An example of software simulation of network connections is simulator ns-2 [2]. Timing of data motion in network is simulated in this tool. In ns-2 all protocols of network stack TCP/IP are implemented from view of time and duration of frames transmission. Network protocol stack is implemented in this simulator by the following way:

1-2. Application and presentation layers are implemented as particular case of network connection between applications: connection between FTP server and FTP client.

3-4. Session and transport layers: TCP and UDP network protocols are implemented.

5. Network layer: attaching IP protocol header to the frame is modeled on this layer simply by frame size increase.

6. Data link layer: 802.3, 802.11 and other protocols are implemented.

7. Physical layer: two-ray and other radio signal propagation modes are implemented.

Ns-2 simulator is usually utilized for estimation of network performance or evaluation of traffic control algorithms. This simulator provides very high speed of simulation as all network communications are performed inside one application and also because no actual data transmission modeled in it. This is why the authors used ns-2 for measurement of performance in wireless networks.

Performance of data transmission through wireless networks is very sensitive to the state of media: radio fading causes the loss of data packets and retransmission of lost packets degrades data throughput.

Data losses may be reduced for example by adjusting the transmission power or by transmission rate or by transmitted frames size.

Transmission rate may influence WiFi network performance in different ways. The lower the transmission rate, the higher the transmission reliability, but the fewer data packets can be transmitted for the same time interval. And vice versa, the higher the transmission rate, the more data packets can be transmitted for the same time interval, but the lower the transmission reliability (i.e., the higher the

probability of data damage, and hence the probability of packet retransmission). For every level of radio fading there exists the optimal transmission rate — the rate that provides us with the best transmission throughput. A rate-adaptation algorithm should maintain this optimal rate.

The authors of this paper used ns-2 for evaluation of different rate adaptation algorithms in 802.11 networks (WiFi) working indoor.

For these purposes it was necessary to implement physical level of network protocol stack in ns-2 as rate adaptation has sense only in conditions of radio fading while transmission media simulated in ns-2 is free of radio noise.

Radio fading parameters were measured for different environments and published in the work [3]. The task of indoor radio fading simulation was solved by the authors together with F. Sherstyuk [4] basing of these measured radio fading parameters. The result of the solution was simulated values of radio noise level in indoor media.

These values of radio noise level were embedded in source codes of ns-2 simulator: they were used for defining of Signal-to-Noise Ratio (SNR) in the given moment of time. It should be noticed that in ns-2 signal level depends on the distance between transmitting and receiving stations.

Using experimental data of error rate probability during frame transmission at the given level of SNR the probability of error in received frame was calculated. The event of the error in received frame was modeled using random numbers generator. If error is present the frame reception is not acknowledged and transmitting station has to re-transmit the frame.

Described improvement of ns-2 let use it for rate adaptation algorithms estimation. The source codes of the algorithms were embedded into source codes of ns-2 and network connection performance was measured for each of the algorithms (Fig. 2).

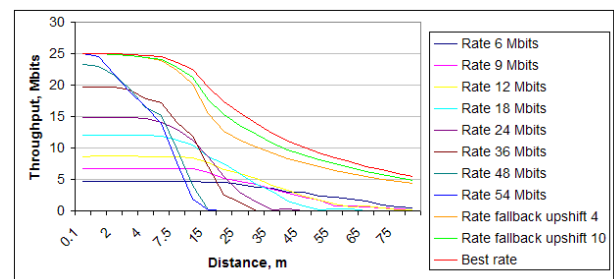


Fig. 2. Graphs of network connection throughput at constant transmission rates and with different rate adaptation algorithms usage depending on the distance between WiFi stations.

The following rate adaptation algorithms estimation drawn on Fig. 2:

- 1) The “rate fallback” algorithm decreases the transmission rate after two packet losses and increases the rate after a defined (4 or 10) number of successful transmissions.
- 2) “The best rate (clairvoyant)” algorithm chooses the transmission rate for which the transmission throughput is the best, i.e., the probable duration of packet transmission (including duration of transmission of packet reception acknowledgement and the intervals between packets) is minimal.

Simulation results let estimate “rate fallback” algorithm

and consider it good enough approaching “the best rate” algorithm results. As a result of this conclusion rate fallback algorithm with upshift 10 was embedded into wireless card firmware.

III. OS PROTOCOLS IMPLEMENTATION USAGE

An example of usage in simulator of network protocols implementation in operating system is simulator of computer platform developed with participation of the authors. This simulator was designed for validation of BIOS and drivers developed for new computer platform before this platform production.

Simulation of network connection is needed in computer platform simulator for network software validation. For example usage of simulated network connection makes it possible to check if BIOS initializes network devices properly. Also it can be checked if driver works properly that was designed for verification of network hardware included in this (future) platform. And also interaction between the platform and network devices can be verified on simulator.

Usage of OS network protocols implementation implies that network data flow goes through operating system protocol stack and through the network devices that are installed on the computer. This means that simulation software runs on two computers connected by real network. It transmits and receives data that passes through network protocol stack of operating system on one computer, then through network connection and then through network protocol stack of operating system on the other computer.

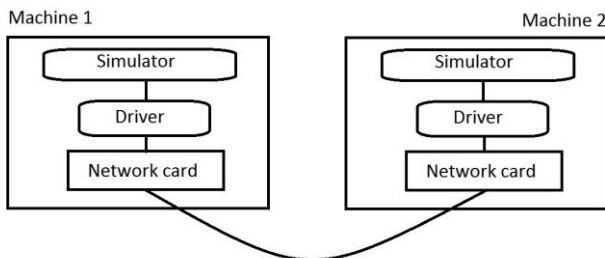


Fig. 3. Two simulators network communication through real network cards.

In Fig. 3 two instances of platform simulator are drawn that are running on different computers. Each of them assigned certain IP address that also reported to the other instance of simulator. Then each of simulators connects to previously installed driver reporting its IP address to this driver. After that transmission of data packet is simulated by transmission of these packets to real network with real network devices usage and also by usage of implementation of network protocols in operating system.

This configuration of network simulator is used for example for task of software validation on computer platform simulator. In this way simulated network corresponds to real network implementation. This let use real network for connection between two instances of platform simulator.

The advantage of this method is its reliability for network software validation. In this mode all functionality of the software that is to be validated is used and thus tested.

The drawback of this method is complexity of the software debugging. The peculiarity of network software running in

operating system is that delay in the packet transmission or reception caused problems in the network operation. Debugging of simulator becomes very difficult because delay in frame transmission caused by pausing simulator in debugger makes operating system consider the frame as lost which prevents normal work of simulated network.

One more drawback of this approach is possibly huge amount of the outsider data transferred through the real network. This impedes the data analysis in simulator. For example this makes data transmission logs enormous that makes their processing difficult.

Also when we have many simulated machines connected with a network it requires special techniques to maintain determinism and to enable saving state of the whole network. These techniques usage is difficult in conditions of real network.

IV. COMBINED METHOD

In case when we are going to use network protocol implementation in operating system but hardware network devices for some reasons can not be used for network simulation we can try to use some kind of virtual network device for network data transmission.

An example of such virtual network devices is TUN/TAP driver [5].

TUN and TAP are virtual-network kernel devices that are implemented in software. TAP simulates a data link layer device and operates with data link layer frames such as Ethernet frames. TUN simulates a network layer device and operates with network layer frames such like IP packets.

Data sent by an operating system via a TUN/TAP device are delivered to user-space program which attaches itself to the device. A user-space program may also put packets into a TUN/TAP device. In this case TUN/TAP device passes these packets to the operating-system network stack thus emulating their reception from an external source.

TUN/TAP virtual device was used for network simulation in the work [6] performed by M. Khropov and A. Yampolskiy, students of the Faculty of Mechanics and Mathematics of Moscow State University, with scientific advising of the authors. In this work WiFi connection was simulated using radio fading simulation results obtained in work [4].

User-space program that attaches to TUN/TAP virtual device was developed that models behavior of WiFi connection. The queue of frames was supported in this program and frame transmission delays calculation implemented corresponding to 802.11 protocol and also radio fading simulation results applied to transmission condition for data loss probability finding.

The result of the work is a simulator of WiFi connection which behavior is very close to real WiFi connection behavior.

This simulator was successfully used for estimation of algorithm developed by the same students that controls data flow for WiFi network performance improvement. The results of these estimation proved that the developed algorithm significantly improves WiFi network performance and this algorithm was used in WiFi network driver.

V. CONCLUSION

Task of network simulation is usually reduced to the task of network protocols implementation. The protocols may be implemented in simulation software or operating system network protocols implementation may be re-used. In the latest case network devices are to be used in simulator and these devices may be real or virtual. This variety of possible simulator architectures provides simulator designers with ability to choose simulator architecture that is the most suitable for the task that is to be resolved.

In this paper the authors described the simulator architectures chosen by them for resolution of different tasks arisen in semiconductor industry.

The way of software implementation of network protocols was chosen for the task that claims high speed of simulation and doesn't need high simulation accuracy.

The way of re-usage of network protocols implementation in operating system was used for the task that demands high simulation accuracy.

The way of virtual network devices usage was chosen for the solution designed under conditions that network is to be simulated on the single computer that is when network resources were restricted.

The experience described in this paper shows that there are variety of possible architectures of network simulators and that the choice of the architecture is to be made according to

the demands applied to the solution and also by the resources that are accessible.

REFERENCES

- [1] Publicly Available Standards. (July 30, 2010). [Online]. Available: Standards.iso.org
- [2] The Network Simulator - ns-2. [Online]. Available: <http://www.isi.edu/nsnam/ns/>
- [3] M. Carroll and T. Wysocki, "Fading characteristics for indoor wireless channels at 5 GHz unlicensed bands," in *Proc. SympoTIC '03, the Joint First Workshop on Mobile Future and Symposium on Trends in Communications*, 26-28 October 2003, pp. 102-105.
- [4] N. A. Podolskaya and F. N. Sherstyuk, "The problem of WiFi radio-fading simulation: Solution and applications," *Journal of Mathematical Sciences*, vol. 152, issue 4, pp. 571-577, July 2008.
- [5] Virtual Point-to-Point(TUN) and Ethernet(TAP) devices. [Online]. Available: <http://vtun.sourceforge.net/tun/>
- [6] M. S. Khropov and A. D. Yampolskiy, "About solution of the task of network connection simulation working under 802.11 standard," in *Proc. the 52-th scientific conference of Moscow Institute of Physics and Technology*, part I, vol. 1, 2009, pp. 37-40.



Nina A. Podolskaya was born in USSR in 1963. She received master degree in mathematics from Faculty of Mechanics and Mathematics of Moscow State University in 1985 and received PhD degree in mathematics from Moscow State University in 1993. She worked as a research fellow in MSU, as a senior software engineer for Resonext Communications, Inc., as a staff software engineer for RF Micro Devices, Inc., as a senior member of Technical Staff for Cadence Design Systems Inc, as a senior software engineer for Intel Corporation. Currently she works as a docent for Moscow State University, Moscow.