УДК 004.7:519.2

ANALYSIS OF NETWORK PERFORMANCE UNDER SELFSIMILAR SYSTEM LOADING BY COMPUTER SIMULATION

L.O. Kirichenko[1], T.A. Radivilova[2]

[1]KNURE, Kharkov, Ukraine, ludmila@kture.kharkov.ua
[2]KNURE, Kharkov, Ukraine, lmd@kture.kharkov.ua

The simulation have shown that management of selfsimilar traffic allows to improve quality of network service and avoid overflow of the buffer memory.

SELFSIMILAR DATA, TRAFFIC MODEL, SIMULATING COMPUTER NETWORKS.

Introduction

Numerous researches of processes in a network have shown that statistical characteristics of the traffic have property of time scale invariance (selfsimilarity). The reasons of such effect are features of distribution of files on servers, their sizes, and typical behavior of users. There were found, that data flows that initially not showing selfsimilarity properties, having passed processing on main servers and active network elements, obtains attributes of selfsimilarity. The selfsimilar traffic has the special structure kept on many scales. There are always a number of extremely large surges at rather small average level of the traffic. These surges cause significant delays and losses of packages, even when the total loading of all streams are more less than maximal values. In a classical case for Poisson stream buffers of an average size will be enough. The queue can be formed in short-term prospect, but for the long period buffer will be cleared. However in case of selfsimilar traffic queues have more greater length [1-3].

The traffic in computer networks with high servicing factor shows properties of selfsimilarity. Because of it the fast overload of devices' buffers is possible even with small servicing factor. Especially it happens if the buffer's size has been calculated for loading with Poisson streams' distributions.

For the majority of networks actual that incoming load can exceed one which can be served even at optimum routing. Thus without restrictions of the incoming traffic, queue on the most loaded lines will grow without limit, and eventually will exceed the sizes of buffers in corresponding units. This can cause the situation, when incoming packets will be ignored and thus will have to be transmitted again, that leads to irrational expenditure of network resources.

So, increasing load of network will lead to channel capacity decrease and information delays growing. It is obvious, that number of lost packets should be reduced as much as possible to improve network quality. The simplest, but also the most expensive method is channel bandwidth widening. More cheap, but also quite effective results can be obtained by using methods of switching, routing and information streams management. Also some special methods of queuing are applicable for this problem. These methods can be broken into three major groups: special queuing strategies, traffic shaping and speed limitation [1].

Queuing strategies. Queues are usually formed only in case of occupied interface. In opposite case packets are retransmitted without any additional processing. All standard queues use FIFO(first in first out) principle: packet that came earlier will be transmitted first. If queue is overloaded and new packets are incoming, the tail drop is take place. More complicated way is to use several queues. Packets are classified according to user needs and then are sort over corresponding queues. When interface is able to transmit packet, special algorithms are used to select queue.

Traffic shaping. While shaping count of traffic for the interface is take place. Shaping can be applied to the whole traffic or only to those packets, which matches some criteria. This happens both in case of free interface and in case of full queue. When traffic exceeds some user-defined value all incoming packets are pushed into queue and delayed. Thus, used network channel capacity is limited by some value.

Speed limitation. This method is quite similar to shaping. The difference is that abuse traffic is processed separate from usual, by user-defined rules. The most widely used method is deletion of excessive traffic, while some other methods exist, such as priority decreasing in IP-header.

1. Switching of information flows

There're several methods of flow switching: circuit switching that can be done both on logical and physical layers, message switching, packet switching [1, 6]. Circuit switching is the persisting of channel that connects some net abonent on overall connection time. The main disadvantages of circuit switching method are: inability of speed and code transformation, that causes necessity of similar hardware usage; organization of multicast and circular transmitting can become very tricky; the lost of requests for connection in case of free channel absence; low channel loading in cases, when the probability of packet lost must be decreased greatly.

Network with message switching method implemented are almost free of these lacks. Here the incoming message is moved to memory and then transmitted into outgoing channel. Such network, especially with

high channel usage, needs significant number of memorizing devises in switching nodes.

Packet switching method has the same advantages as message switching, but requires much less memory, because packet size is significantly smaller then message size.

2. Routing and intencity management

There are different criteria of routing methods classification which used in information systems. One of it is the classification by management centralization: centralized, distributed, zone. [1, 4].

In centralized methods route choosing is held in central control point. Distributed methods cause route to be chosen by each joint. Joints can interact with each other to tell some system information.

Centralized routing methods' main idea can be stated as – each joint transmits its state information to the central control point, that builds complete network state. This information is used then to choose optimal routes.

In order to combine advantages of these methods some hybrid methods were created. Here central control point observes the global state, and other joints can react to local traffic and component changes. Each joint can decide by itself what route to choose with respect to local infrastructure.

Other classification of routing methods is based on route-changing property. In static methods route that is defined by pair "sender"–"receiver" is fixed and doesn't depend on traffic fluctuation. It can be changed only in case of some hardware breakage (i.e. network topology change). Such routing method can be recommended only for simplest networks, or when network effectiveness is not a priority, because it can't provide enough channel capacity in cases of incoming traffic variations. Some intermediate place is occupied by quasi-static methods. In opposite to static methods (that have hard-coded routes) and dynamic routes (routes are defined by analyzing current network information) it uses the number of local route tables that are not fixed and is changed (but quite slow) when traffic intensity vitiates.

More to say, all routing method can be split into two groups by such criteria: some mathematical background is used in method for taking optimal route by performance criteria; heuristic methods.

The concrete routing method is usually implemented in network level protocol that manages packets transmission over network to the destination host. Flow control can be necessary on data transmitting between abonent and incoming network node or between two network nodes.

But, none of discussed methods of overload network are prevention provides significant quality of service. Furthermore, they don't use parameters of incoming traffic that actually cause queue creation.

3. The simulation

Simulation Modeling is becoming an increasingly popular method for network performance analysis. Generally, there are two forms of network simulation: analytical modeling and computer simulation. The first is by mathematical analysis that characterizes a network as a set of equations.

The main disadvantage is its over simplistic view of the network and inability to simulate the dynamic nature of a network. Thus, the study of a complex system always requires a discrete event simulation package, which can compute the time that would be associated with real events in a real-life situation. Software simulator is a valuable tool especially for today's network with complex architectures and topologies. Designers can test their new ideas and carry out performance related studies, therefore freed from the burden of the "trial and error" hardware implementations.

One of well-known network simulators is OPNET Modeler. OPNET (Optimized Network Engineering Tool) provides a comprehensive development environment for the specification, simulation and performance analysis of communication networks. A large range of communication systems from a single LAN to global satellite networks can be supported. Discrete event simulations are used as the means of analyzing system performance and their behavior. [5].

The key features of OPNET are summarized here as:

– modeling and Simulation Cycle. OPNET provides powerful tools to assist user to go through three out of the five phases in a design circle(i.e. the building of models, the execution of a simulation and the analysis of the output data);

– hierarchical Modeling. OPNET employs a hierarchical structure to modeling. Each level of the hierarchy describes different aspects of the complete model being simulated;

– specialized in communication networks**.** Detailed library models provide support for existing protocols and allow researchers and developers to either modify these existing models or develop new models of their own;

– automatic simulation generation. OPNET models can be compiled into executable code. An executable discrete-event simulation can be debugged or simply executed, resulting in output data.

In order to construct adequate model the information about modeling objects is needed. The network from the physical point of view is the union of some number of computers, hubs and switches, that are connected via communication lines. Let's see main units of the network:

– computer – source and destination of a signal, with management and data buffering abilities;

– connection lines (wires, hubs) – the delay element;

– switch, modem, server – delay element with management and data caching abilities;

– router – element that manages data flow with data buffering abilities.

Buffering here means the ability to store some data inside device. The main part of system ("executive mechanism") is represented by server and router. From the physical point of view this is buffer for incoming data – capacitance element, packet switching element and outgoing buffers. Figure 1 shows the example of computer network. This network consists of four workstations, server, cisco router, and hubs, connected via wire.

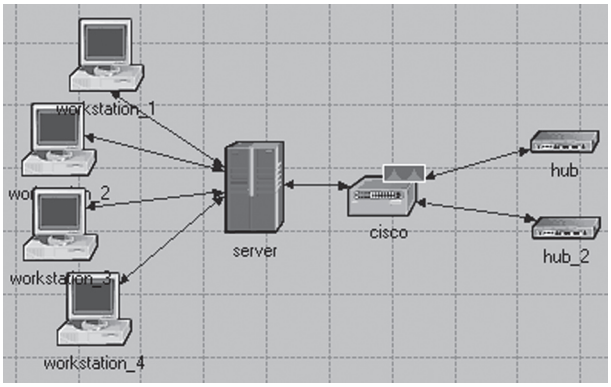

Fig. 1. Example of computer network

While there exists lots of software, that allows to build and simulate different computer networks, there no unified model of selfsimilar traffic, that will provide a way to simulate such traffic with given characteristics.

In this paper the mathematical model of traffic is provided [6]. The input traffic let us treat a certain random process with arbitrary law of distribution. An our task is find a model of random process such that when it passes through the communication channel the queues and probabilities of losses have values similar to those of the real traffic.

In this work we show that the length of a queue is defined by three key parameters of the input traffic: intensity, the Fano parameter F and Hurst's parameter. The Fano parameter F is defined as the ratio of the dispersion of the number of events on the given time interval T to the average of this number: $F(T) = \frac{D[N(T)]}{M[N(T)]}$, where $N(T)$ is the random variable N defines the number of events of the given stream on the interval T. Big values of the Fano parameter correspond to a wide scatter of values in the input stream, which provides queues even at small intensity. Selfsimilarity and long-term dependence of the traffic is indicated traditionally by Hurst's parameter. Hurst's parameter value $H \rangle 0.5$ characterizes long-term dependence of the process. Particularly, it means that high values of the process intensity will be most probably followed with the same high. That does not allow the buffer to be released quickly enough.

The model of traffic presented in this work is selfsimilar random process with discrete time, based on fractal Gaussian noise. It correspond to the certain value of Hurst's parameter. Traffic model is an exponent of Gaussian noise: $Y(t) = b \cdot e^{k \cdot X(t)}$ where X is the realization of fractal Gaussian noise, obtained by random summation method. B, k, - parameters that depends on intensity and Fano parameter.

Obtained mathematical model of selfsimilar traffic allows to make simulation with given buffer size and link channel capacity. The result of research states, that for 80-90% system load buffer size should be greater that traffic intensivity in hundreds times to minimize the losses. Numerical experiment results provide a way to compute the average size of buffer, needed to normal transmission (not more that 7% losses) of traffic with given channel capacity C and obtained parameters of incoming data flow $Buf = f(C, \overline{X}, F, H)$ [7].

If computed buffer size if greater than existing size, the data will be transmitted to reserve channel or router with additional buffer (fig. 2).

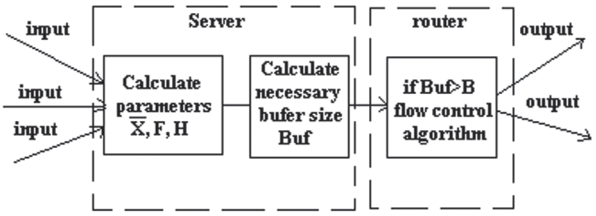

Fig. 2. Traffic management in modeled network

Conclusion

This paper describes a new traffic model we implemented in OPNET Modeler. The simulation shows a way to manage data flow with goal to network node for processing. The management of data allows to improve quality of network service and avoid overflow of the buffer memory in network node. This method is based on finding the maximum of system load by monitoring of incoming traffic.

References: 1. *W. Stollings.* High-speed networks and Internets. Performance and quality of service. New Jersey, 2002. 2. *Vern Paxson and Sally Floyd.* Wide-Area Traffic: The Failure of Poisson Modeling IEEE/ACM Transactions on Networking, vol. 3, № 3, pp. 226-244, June 1995. 3. *Leland W.E., Taqqu M.S., Willinger W., and Wilson D.V.* On the selfsimilarnature of Ethernet traffic, IEEE/ACM Transactions of Networking, 2(1), 1994. pp.1-15. 4. *М. Кульгин.* Практика построения компьютерных сетей. Для профессионалов. 2001, СПб., "Питер". 5. *J. Theunis, P. Leys, J. Potemans , Bart Van den Broeck, E. Van Lil and A. Van de Capelle.* "Advanced Networking Training for Master Students Through OPNET Projects", OPNETWORK 2003, Washington D.C., USA, August 2003. 6. *Кириченко Л.О., Радивилова Т.А.* Исследование влияния самоподобия трафика при проектировании фрагмента сети // Нові технології. – 2007. – № 1-2. – С.124-129. 7. *Кириченко Л.О., Радивилова Т.А.* Управление параметрами сети на основе мониторинга входной нагрузки//Материалы 2-й Международной научной конференции «Современные информационные системы. Проблемы и тенденции развития», Харьков-Туапсе. – 2007. С. 89-90.

Поступила в редколлегию 07.05.2008