Gest. Prod., São Carlos, Ahead of Print, 2018
https://doi.org/10.1590/0104-530X1429-14

GESTÃO & PRODUÇÃO
ISSN 0104-530X (Print)
ISSN 1806-9649 (Online)

Description, modeling and simulation of the congested link of a university

Caracterização, modelagem e simulação de enlace congestionado de uma universidade

Mariana Vazquez[1]
Reinaldo Morabito[2]
Cesar Marcondes[3]

Abstract: This paper presents a performance evaluation study of a congested communication network in a university. Queuing often occurs in this network due to high demand and disposal of requests at peak times, resulting directly in decreased service quality for its users, who have their connections delayed or interrupted and cannot access the web pages. The goal of the present work was to characterize this congestion problem empirically, capturing packet traces in logs and studying delays and losses, and to model and analyze this system based on the queuing theory and discrete event simulation to evaluate its performance, identify bottlenecks and proposing solutions to improve the quality of service. The study showed that, in situations of peak traffic, simulation was the quantitative approach that produced results closer to the empirical ones, as the analytical queuing models studied did not provide good approximations, mostly by not reflecting the control exercised by the Transmission Control Protocol (TCP).
Keywords: Performance evaluation; Simulation; Queuing theory; Internet network; Congestion control.

Resumo: *Este trabalho apresenta um estudo de avaliação de desempenho de uma rede de comunicação congestionada de uma Universidade. Nesta rede, é comum a formação de filas, devido à alta demanda e ao descarte de requisições em horários de pico, resultando diretamente na queda da qualidade de serviço aos usuários, que têm suas conexões demoradas ou interrompidas e não conseguem acessar páginas na web. O objetivo deste trabalho foi caracterizar o problema de congestionamento empiricamente, capturando rastros de pacotes em logs e estudando atrasos e perdas, e modelar e analisar esse sistema, com base nas teorias de filas e simulação discreta, para avaliar seu desempenho, identificar gargalos e propor soluções para a melhoria da qualidade de serviço. O estudo mostrou que em situações de pico de tráfego no sistema, a abordagem quantitativa que produziu resultados mais próximos do empírico foi a simulação, pois os modelos analíticos de filas estudados não forneceram boas aproximações, principalmente por não refletirem o controle exercido pelo protocolo TCP.*
Palavras-chave: *Análise de desempenho; Simulação; Teoria de filas; Redes de Internet; Controle de congestionamento.*

1. Introduction

The explosive growth of the Internet has led to an increase in demand for different types of services in packet-switched networks. These networks no longer exclusively handle data, but enable the transmission of live digital audio and video with a quality close to that of circuit-switched networks. For this to be possible, the network must be able to offer services that meet certain quality standards in terms of bandwidth, delay, packet delay variation, and packet loss. The evolution of fiber optic transmission technology has allowed the speed of networks to increase rapidly. However, the problem of congestion, when transmission demand is approaching the transmission capacity, has not been satisfactorily resolved. This has become one of the main barriers to achieving the quality required by the above-mentioned services. The use of the Internet as

[1] Departamento de Análise e Desenvolvimento de Sistemas, Centro Estadual de Educação Tecnológica Paula Souza – CEETEPS, Faculdade de Tecnologia de Americana – FATEC, Rua Emílio de Menezes, s/n, CEP 13469-111, Americana, SP, Brasil, e-mail: vazquez.prof@gmail.com

[2] Departamento de Engenharia de Produção, Universidade Federal de São Carlos – UFSCar, Rodovia Washington Luís, Km 235, CEP 13565-905, São Carlos, SP, Brasil, e-mail: morabito@ufscar.br

[3] Departamento de Computação, Universidade Federal de São Carlos – UFSCar, Rodovia Washington Luís, Km 235, CEP 13565-905, São Carlos, SP, Brasil, e-mail: marcondes@dc.ufscar.br

Received Apr. 30, 2014 - Accepted Mar. 24, 2015

Financial support: This research was partially supported by CAPES and CNPq.

a multiservice network has influenced the proposition of Quality-of-Service (QoS) mechanisms, since the Internet was designed to provide services that initially did not require bandwidth guarantee, delay limits, and delay variation (jitter) (Pinto et al., 2003): as initially envisioned, it followed best-effort delivery.

In Brazil, the Internet connectivity at the federal level (federal institutions) is provided by the Brazilian National Research and Education Network (RNP), which is responsible for providing links to universities. The Federal University of São Carlos (referred to in this study simply as the "University") is one recipient, but updating its links takes time and can cause congestion problems: this situation was the object of the present research. During the empirical evaluation reported in this paper, the University's Internet connection was saturated at peak times, with rates close to the maximum link speed of 155Mbps.

This congestion condition is rarely described in the empirical literature using quantitative measures, such as those presented here. A few studies (Lee et al., 2011) present an unconventional view based on quantitative measurements, where the degradation of a link operating at 100% of its capacity (at peak times) does not correspond directly to the rate of packet loss. The latter usually proves to be small, but directly influences the QoS offered to the user, since the user notices the added delay in the transmissions, which have their Round-Trip Time (RTT) increased.

Evaluating network performance can also be important when considering whether to install a new service that will result in heavy and intense traffic. The evaluation can determine whether the network will support this new service, and if not, can help in creating a new topology (Brito, 2012). Hence, it is necessary to characterize and then model traffic, and analyze system performance, in order to establish metrics such as response time (total, per component); throughput; use; scalability; availability; and reliability.

The objective of this work is to empirically characterize the congestion problem of the studied University, by capturing packet traces in logs (event registration process) and studying delays and losses, as well as modeling and analyzing this system based on queuing and simulation approaches. In most scenarios, routers have a single First In First Out (FIFO) queue, which collects incoming packets and de-queues them as quickly as the hardware allows. The sizes of these queues are usually dependent on the link capacity multiplied by a round-trip time of 200ms (Srikant, 2004). Because Internet network links transfer serialized data, the queue is required to manage the packets in the output.

The basic idea underlying the analysis is that if capacity and average flow are known, it is possible to compute the average delay of a packet on a given link by applying the queue theory. However, due to the complexity of incoming traffic, it is often not possible to accurately determine this by using simpler Markovian models, such as the M/M /1/ k queue (Poisson arrival/service process and service times are exponentially distributed, one server and queue capacity equal to k). If this model represents the problem well, it is possible to easily and accurately calculate the average end-to-end delay, for example, by convolving queues of all links (Kleinrock, 1975). However, the M/ M/1/ k model is not suitable for the analysis of the University system in the present case study.

The packets arriving at the link are stored in the router's buffer to await transmission; however, if the arrival rate is excessive, congestion occurs, and the router starts the discard process. To ensure that data will behave correctly in the presence of congestion, the protocols used for Internet data transmission include end-to-end congestion control mechanisms that automatically reduce the rate at which data is transmitted when congestion is detected. One of the most used protocols is TCP (Transmission Control Protocol), which transmits in bursts (thousands of files transmitted in a few milliseconds), in order to send as many links as possible.

This dynamic between router and TCP can be described by the fluid control technique; however, this technique has limitations because it does not capture other protocols, such as UDP, DNS, and image protocols in general. On the other hand, queuing theory is more generic, and any protocol can be used (Bu & Towsley, 2001). The search for solutions to congestion problems motivates research in the areas of network planning, management, and design. All these areas are supported by modeling and traffic characterization (Van Woensel et al., 2010). The process of characterization and modeling of traffic are preponderant points in the evolution of telecommunications networks. A simple and accurate modeling of the traffic can allow the understanding of a physical network problem as a mathematical/statistical problem, with a consequently simpler solution. Modeling also allows the realization of performance simulations on a network before its implementation, and the correction and validation of parameters during its existence.

Performance evaluation models of queuing systems have been applied using analytical methods (exact and approximate), and simulation and related techniques. In some cases, these models are used in an integrated way. It is common to apply an analytical model to reduce the set of possible system configurations, and identify the parameters that most affect its performance: and then, using an experimental simulation model, determine which of the evaluated configurations is best (Leung & Suri, 1990). Thus, simulation can play an important role in investigating how the system behaves, confronting experimentation, measurement, and analysis. Of course, all four approaches (simulation, experimentation, measurement, and analysis) are necessary, each playing a fundamental role. Measurement is necessary for reality verification and for challenging implicit assumptions. Experiments are crucial to dealing with implementation issues, which at first may seem trivial, but often introduce unexpected complexities. Experimentation also plays a key role in exploring new environments, and how Internet protocols should operate in them (Banks et al., 2010).

This work is organized into six sections including this Introduction. Section 2 presents the study of real traffic characterization of the University network and the measurements made with the use of specific software. Section 3 presents the statistical analysis of the data collected in this study. Section 4 describes the development of simulation models for application in the University network. Section 5 evaluates the results of the performance measures. Section 6 presents the conclusions of this research, together with the future perspectives of the work.

2 Data collection and processing methodology

To conduct this empirical study of congestion on an academic network, we had access to data from the University, an academic community of approximately 12,000 users, which sees Internet usage reach more than 90% of capacity during certain peak periods of each day. Initially, the environment used by the University is described. From the topological point of view, the university network is physically organized as an extended star. At the core of the star is the General Information Office (SIn), which houses the core equipment of the network and the edge router.

It should be noted that the equipment used for University Routing at the time (May/2012) was a CISCO 7200 model with a 155Mbps Synchronous Digital Hierarchy (SDH) interface with the operator, and therefore the Internet. After the study of the network mapping, it was verified that this router was the bottleneck on the Internet network, since the main Internet connection arrived in the SIn with a capacity of 155 Mbps and went out to the departments (connected directly to the router) with capacity of 1 Gbps. The topology at this time, with the main departments directly connected to the router, can be visualized in Figure 1, with the output of the Internet represented as a cloud. Within the topology of the campus, it is noted that the star's main links are 1Gbps connections, and the internal branches (partially described in figure 1) have connections at 100Mbps.

From the point of view of the 155Mbps link with the operator, the Internet is delivered through the Brazilian National Research Network (RNP). The RNP backbone, not shown in Figure 1, has 10Gbps links between states, and in the case of the University, the 155 Mbps link is the bottleneck. Several monitoring tools have been used, which complement each other

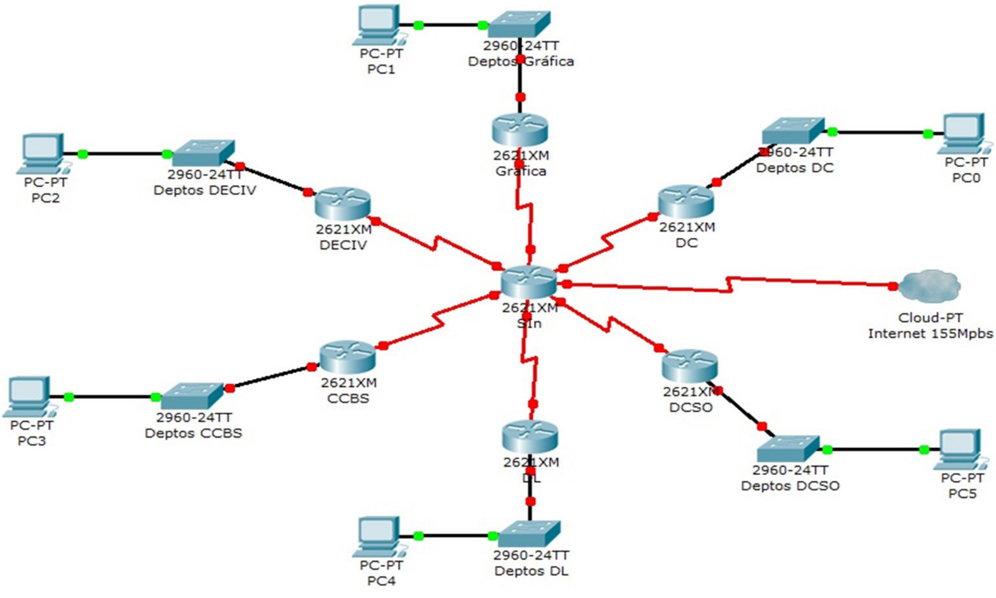

Figure 1. University network topology, with the departments that have routers.

by observing several aspects simultaneously. As an example, the delay of a single stream was captured and processed by TCPDump (used for collection of files in the network) and Wireshark software (used for forensic analysis), while aggregated information about the links (such as total usage) was measured by the Nagios and MRTG network monitoring software programs.

2.1 Description of the tools and applications

Single flow analysis: The tools and their applications are described in more detail next.

TCPDump: TCPDump was used with filter settings to capture the transmission of a single stream through segmentation by source and destination IP addresses, and source and destination TCP port addresses. TCPDump has proven useful for examining and evaluating retransmission and window management operations of TCP implementations (Ostermann, 2012). **Wireshark**: In addition to TCPDump, the Wireshark graphical capture tool was used for more complex analyses, since this software can capture details of the packages such as addressing (source and destination IPs); file size; transmission time; resend; loss; and at a higher level, it can also access the contents of the archives (Orebaugh et al, 2006).

Analysis of aggregate information of the link: In order to share the experience observed by a single flow in a shared queue of the outbound link, macroscopic analysis tools were also used together with the previous microscopic analysis. The University has a Nagios-based monitoring infrastructure (Harlan, 2003) and MRTG (Oetiker, 1998).

Nagios: This was used to monitor services of different protocols (such as SMTP, and POP3) and also the computational resources of network equipment (such as processor load, temperature, and interface drop rate).

MRTG (Multi Router Traffic Grapher): Designed to assess the aggregate output link monitoring to the Internet, the MRTG was used to monitor data by collecting statistics consolidated by the equipment every 5 minutes. An SNMP program was then triggered to extract this information and display last minute, 24 hour, 1 month, and annual data usage graphs. For the collection environment, periods of great congestion were used, as presented by MRTG.

By using TCPDump and Wireshark, all network traffic was collected at 5-minute intervals, while Nagios and MRTG provided information about the bandwidth conditions during these collections, as shown in the graphs in Figures 2, 3, and 4. The graphs show bandwidth and losses placed side-by-side, connection errors, and link utilization.

Figure 2 shows that inbound traffic, from the Internet to the campus, is highly saturated at 127.33 Mbps during one of the collections. The 155 Mbps link, from the payload standpoint, can transmit data at just over 140 Mbps by calculating header exclusion. Thus, at collection time, the use of the link was at 90.95%. Figures 3 and 4 show the results of packet loss at different scales. Figure 3 shows the losses measured at intervals of 2 hours, with a mean of 3 errors/second. It can also be noticed that the daytime pattern generates more traffic, and consequently more errors, between 10:00 h and 18:00 h. Figure 4 shows a detail of the errors, in time intervals of 20 min, between 10:00 h and 13:00 h.

2.2 Collection methodology

For data collection, the following scenario was stipulated: (a) collection of files from the network every 5 minutes, up to a total of 30 minutes, using

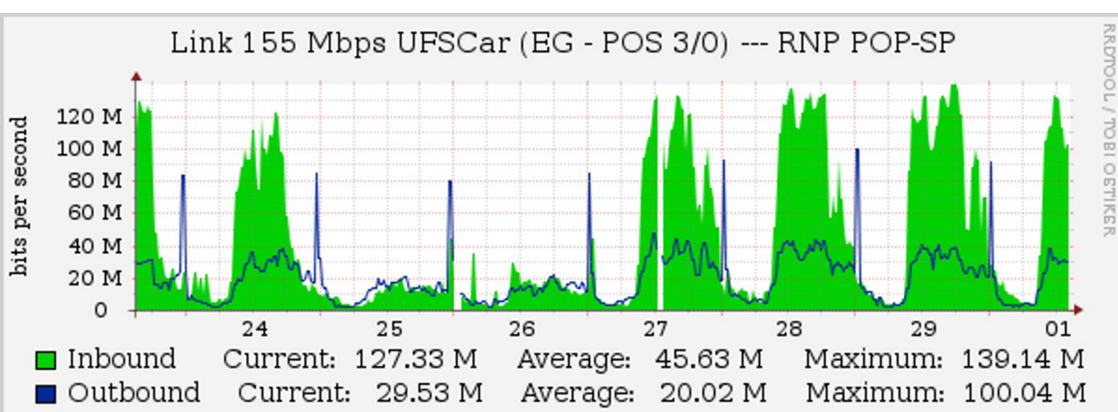

Figure 2. Use of the University network on month days.

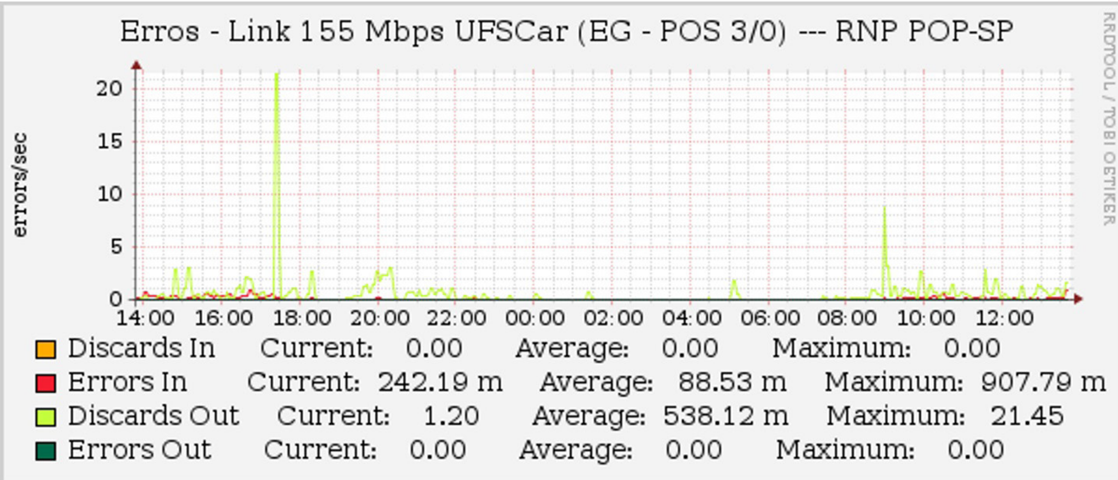

Figure 3. Connection errors in 2-hour time periods.

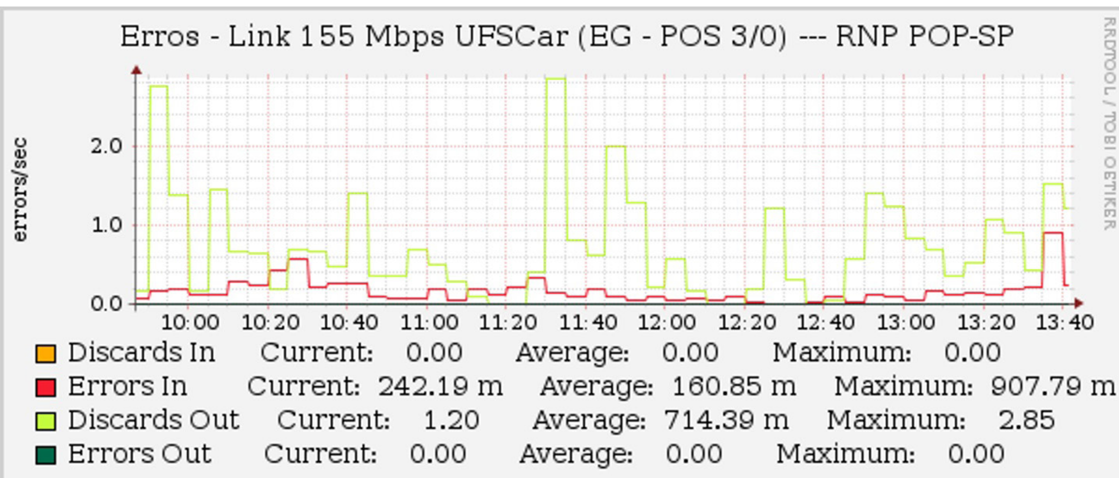

Figure 4. Connection errors in 20-min time periods

TCPDump software; (b) injection of network traffic (active measurement), and downloading in and out of the network (passive measurement); (c) Injection of Internet Control Message Protocol (ICMP) packets to measure performance inside and outside the network; and (d) generation of graphs between nodes evaluated with Nagios and Multi Router Traffic Grapher (MRTG).

Passive Measurement: The passive measurement method analyzes the performance of networks using passive devices, so-called because they do not interfere with network traffic as they perform their measurements. These devices only observe the current traffic passing through observation points which are triggered periodically to collect the information. This is how the performance and status of the network are analyzed. Measuring performance using passive measurement does not increase network traffic at the time of measurement, as this technique uses real traffic. On the other hand, it is necessary to access the medium to collect the data, and the alarms generate traffic, which in some cases can be substantial. The amount of data collected with passive measurement can be robust, especially if flow analysis or information gathering requires observation of all data packets traveling on the network.

Passive measurements are valuable for evaluating performance when the aim is to detect the network problem, but there is a limitation in emulating error frames or isolating the exact location of the problem. Another problem is security. This type of performance measurement solution needs to access information from all packets to characterize the state of the network, thereby compromising users' privacy.

During the collection period, passive measurements were performed at times of low traffic (03:00 h) and high traffic, where congestion was more likely to occur. The records in the logs of thousands of flows showed that, on average, daytime flows experienced a substantial increase in round trip time (RTT) when compared to night flows. However, due to the complex nature of relating the RTT and the location of congestion, it was decided to perform active measurements, presented next.

Active Measurement: Active measurement is a method for analyzing network performance, and is intended to inject test packets into the network, or send packets to servers and applications, in order to measure network performance by evaluating how test packets behave during traffic. However, this method has the disadvantage of adding extra traffic which is not part of the normal behavior of the network: it is considered artificial traffic that could emulate a transaction by a user. The volume and other parameters of the additional traffic are fully adjustable, with the additional traffic being small compared to the total traffic, and sufficient to perform significant measurements.

Measuring network performance using active measurements provides explicit control in generating packets for carrying out measurements. This control includes the nature of the generated traffic; sampling techniques; timing; frequency; scheduling; size and type of packages (which may vary in order to simulate various types of applications); statistical quality; path; and functions chosen for monitoring. The state of being active allows testing what is wanted when necessary. Simulating and verifying traffic becomes a simple task if QoS and service level agreements are met.

As the University network is a monitored network, as discussed in the previous section, active measurement tests were performed using either the ICMP request/reply protocol, which provides error reports, or performing file transfers that passed through the network bottleneck at typical peak times, such as between 14:00 h and 15:00 h. After the mapping of network utilization at peak times, the departments that had more traffic, and consequently greater network use, were observed, according to the passive analysis detailed above. Nagios software provides usage percentages dynamically.

Therefore, indirect measurements were made in a sampling manner in the Department of Computing (which has the highest utilization rates in the University), through 6 collections of 5 minutes of traffic capture at peak hours (with TCPDump), in October 2012. Next, using the Wireshark software, it was possible to identify and separate the streams by source and destination IPs, and also identify packet losses. The measurements were performed indirectly using the Wireshark and TCPDump software tools to collect the network packet stream, and the MRTG to generate the interface graphs. Nagios network management software, along with MRTG, graphically generate statistics using the Simple Network Management Protocol (SNMP), which functions as a client-server. In this way, we are able to monitor the performance of the network and observe how an interface behaves through the analysis of the dump files. No type of control is exerted on the router.

3 Statistical analysis

This section shows the procedures used for the treatment and analysis of the data collected at the University. The highest RTT (Round-Trip Time) on the network being analyzed is 8ms (the maximum time a packet takes to go from client to server and back again, determined by the Ping command-line utility, an operating system utility that uses the Internet Control Message Protocol (ICMP) to test connectivity between devices). The Ping command informs the round-trip time of packets, the quantity of packets transmitted, quantity received, and the loss percentage.

Data obtained from Wireshark was statistically analyzed by two commercial software programs: Arena InputAnalyzer and Bestfit. Heavy-tailed distributions (i.e. those that, when compared to normal or Gaussian distributions, present a much larger quantity of data over a "long tail") are commonly encountered when analyzing input data from a queueing system (Rodríguez-Dagnino, 2004). According to Crovella et al. (1998), the distributions of the sizes of files that travel on the Internet, including files requested by the users, files transmitted by the network, the duration of the transmission of the files and the number of files are characterized as heavy-tailed distributions stored on the servers (queues).

According to Banks et al. (2010), the use of Internet-based telephony systems such voice over IP (VOIP) prompted the development of new models that suggest the use of heavy-tailed distributions such as Pareto and Weibull. The use of faxes and connections to the Internet radically transformed the statistical behavior of traffic, and the Poisson process no longer corresponded to reality. The work of Willinger & Paxson, (1998) shows that packet arrival processes on the Internet are not Poisson processes. Since the variance in the size of the transferred files is very large, the heavy-tailed distributions are quite adequate for this representation.

The above-mentioned analyses confirmed that the arrival and service rates follow heavy-tailed distributions. The sample used was 2000 values (time intervals between arrivals and service times). According to the adherence tests performed by Arena Input Analyzer and Bestfit statistical software packages (Chi-square, Kolmogorov-Smirnof, Anderson-Darling), the Lognormal, Loglogistic, Weibull, and Pareto distributions alternated as adhering most to the data, both for the time intervals between arrivals and for the times of service.

According to Chwif's practical method of calculating sample sizes (Chwif & Medina, 2010), 1600 values would suffice, with 95% confidence. For 99% confidence, the required sample size would be 1838, with 3% sample error. The population was 574,821 packs, within 5 minutes of collection. According to the sample inserted in the Arena Input Analyzer software, the Lognormal distribution for the time between arrivals, with a mean of 0.45 ms and a standard deviation of 0.90 ms, was obtained as the best approximation. For the service times, a Lognormal distribution was also obtained, with mean values 0.47 ms and standard deviation 0.43 ms.

Under heavy traffic conditions, the statistically estimated values were: input rate $\lambda = 2.222$ (arrivals per millisecond), with the mean of the intervals between arrivals of $E(x) = 0.45$ms, where x is the time interval variable between arrivals; and service rate $\mu = 2.127$ (requests processed per millisecond), since the average service time is $E(s) = 0.47$ms, where s is the service time variable of each request. Thus, the average level of utilization is $\rho = \lambda/\mu = 1.047$, higher than 1, which verifies the heavy traffic condition. The values of the variance and standard deviation of the variable x were V (x) = 0.80, and σ (x) = 0.89, respectively; for the variable s, we have: $V(x) = 0.80$ e $\sigma(x) = 0.89$. These values were obtained with the Arena InputAnalyzer and Bestfit.

3.1 Balance of the sample - Statistical Process Control (SPC)

The SPC method (Mahajan & Ingalls, 2004) is a graphical technique used in environments that require multiple replications, and aims to evaluate if the sample collected for system simulation is in equilibrium. This method can be described in four steps:

I. Make n replicate runs of the dummy model, where each replication will have the same size m, to define the values of Ymn and the means of $\bar{Y}m$;

II. Partition the means into x groups of size k, where the means will be represented by group $\bar{Y}(k)b$ para b=1,2,...,x. The size of these groups should be proposed in such a way that it ensures that the means per group are accepted by the Anderson-Darling adherence test for normality and the Von Neuman correlation test (Mahajan & Ingalls, 2004). A minimum of 20 groups is advisable. After dividing the groups, the time series is determined by:

$$Y(k) = [\bar{Y}(k)1,\ \bar{Y}(k)2,\ ...,\ \bar{Y}(k)x]$$

III. Generate the control limits for these time series, considering the estimate of population mean (μ) and standard deviation (σ) from the last half of the series $Y(k)$. After the calculation of the standard deviation and the mean, the control limit (CL) is determined by the formula:

$$CL = \frac{\mu \pm z\sigma}{\sqrt{\frac{b}{2}}},\ \text{where } z = 1, 2, \text{ and } 3 \tag{1}$$

IV. Construct the control chart using the three limits found. By means of the following four steps, the control output is verified:

- Presence of points outside the control limit of 3σ;
- Two points outside the control limit of 2σ of three consecutive;
- Four points outside the control limit of 1σ of five consecutive;
- Presence of eight consecutive points above the average or eight consecutive points below the mean.

3.2 Procedures performed with the University sample

The sample of 2000 values of the interval between arrivals was divided into 100 groups of 20 elements, resulting in a series of 100 elements (*Y1, Y2, ..., Y100*). The mean between the mean Y50 and the mean was calculated, which resulted in the mean value *M (k)* of 0.000427797 (in ms). The resulting standard deviation (σ (k)) between elements Y1001 to Y2000 of the sample was 0.000877976 (in ms). Thus, considering the three intervals (control bands) of z (-1,1), (-2,2) and (-3,3); the value of b = 100 (groups); $k = 20$ (quantity of values for each group);

μ (k) = 0.000427797; and σ (k) = 0.0008779760, the following control limits were established by Equation 1: *CL*[-1.1] = (0.0003036324. 0.0005519616); *CL*[-2.2] = (0.0001794679. 0.0006761261); and *CL*[-3.3] = (0.0000553033, 0.0008002907). The validity of the four steps above was verified in the sample data at the three control limits, and the sample was validated for all of them. In this way, we verified that the sample analyzed is in equilibrium.

4. Simulation Modeling

4.1 Simulation

Rockwell Automation's Arena tool was used to build the simulation model. It is a general-purpose simulation package, which can be used to simulate discrete and continuous systems (Banks et al., 2010). Simulation models are constructed with graphical objects called modules (which define the basic logic of the system and its physical components, such as machines, operators, etc.). Modules are represented by icons associated with input data in a dialog window. These icons are connected to represent the flow of entities. The modules are organized into collections called "templates", which allow the modeling of various types of applications

In the modeling process, resources, queues, process logic, and system data were also added. Arena uses the SIMAN simulation language. The Arena Input Analyzer automates the process of selecting the most appropriate distribution and its parameters for representing existing data, such as process and time between arrivals. The Output Analyzer automates the comparison of several system design and configuration alternatives to model for the simulation. For the development of the simulation model, it is necessary to calculate the following values: number of replications, warm-up period, and the size of replications (time).

Number of replications: In order for the results of a simulation to be reliable and to approximate real-world conditions, it is necessary to run the model more than once, according to the desired confidence interval (Pegden et al, 1990). A confidence interval is a range of values with a (1-*a*) probability of including the true value of the variable to be analyzed, where (1-*a*) is the confidence level and *a* is the allowed error in the probability of the appearance of the variable's actual value in the confidence interval. For example, if a 5% error of the true value of the variable is allowed to be in the range, *a* = 0.05: that is, there is a 95% probability of the real value being contained in the interval (Mahajan & Ingalls, 2004). In order to calculate the confidence interval, it is necessary to obtain the mean (μ) and the half-interval (*h*), the range being limited by μ-h, μ+h]. Having set the confidence level, it is possible to determine the tabulated value of the standard normal random variable (z), which will be used to calculate the number of replication runs. The following formula is then applied:

$$n = \left(\frac{100 z \sigma}{r \mu} \right)^2 \tag{2}$$

with *n* being the number of replication runs, *z* the standard variable value, σ the standard deviation, *r* the required precision, and μ the mean in the sample. If the *n* value is decimal, round to the next integer value. In the constructed model, the mean (μ) was 0.45 ms, the standard deviation (σ) was 0.90 ms, and the required precision (*r*) was 5. Using the 95% confidence level, the z value is 1.96 (tabulated value) and the number of replication runs is 6,146.56, obtained by Equation 2, rounded up to 6,147 replication runs. At a 99% confidence level, the *z* value is 2.58 (tabulated value) and the number of replications obtained with (2) becomes 10,651 replication runs.

Warm-up: The Warm-up period was obtained through the Arena Output Analyzer, by observing the simulation with graphic animation and statistical reports, which indicated that after 5,200 ms the arrival process reaches equilibrium.

Modeling in Arena: The modeling was elaborated in the Arena software with the implementation of six modules of the bars of the components (Basic Process and Advanced Process), according to Figure 5. In the module "Arrivals" we consider the Lognormal Distribution (0.409, 0.839), with the respective parameters provided by the Arena Input Analyzer. The "Decide 1" module evaluates the number of requests arriving at the router. The router's buffer capacity (B =rtt*C=8ms*1Gbps [109 bits per sec] = 8000000 bits) is up to 1,297 (8000000 bits / 6166 bits) requests queueing (router's input port), considering the average request size of 6166 bits. The value of 8ms comes from measurement, and represents the exact RTT of the network (Srikant, 2004).

The "Decide 1" module makes the following evaluation: if there are already 1,297 requests in the router queue, the next request will go through a "counter" module, called "Record 2" in the modeling (so that the quantity of lost requisitions is presented in the ARENA statistical reports). Next it will proceed to the "Loss" module. If there are not already 1,297 requests in the queue, the request will proceed to the "RoutingRequest" block (Service Process), and in the sequence it will go to the "Exit" block and be considered delivered. The RoutingRequest block is the Service Process block. An empirical distribution was considered (since none of the existing theoretical distributions passed the adherence tests), with the

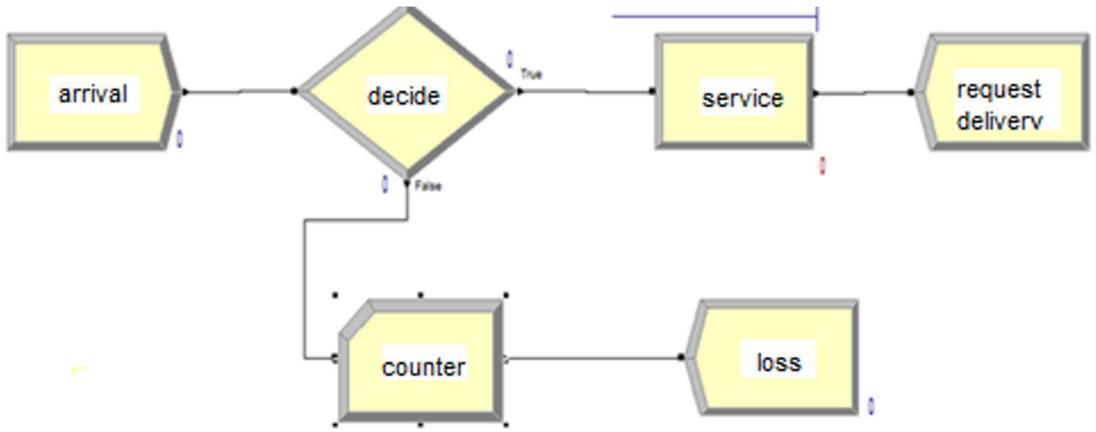

Figure 5. Modeling in ARENA.

Probability and Cumulative Distribution Function parameters provided by the Arena Input Analyzer.

4.2 *M*/*M*/1/*k* and *G*/*G*/1 analytical models

The G/G/1 (general arrival and general service, one server) can be considered a possible analytical queueing model to address congested links. It was chosen because the results of the statistical analyses pointed to non-exponential distributions, both for the arrival process and the service process in the network, and this model considers that these processes follow general distributions. Although there is a limitation of the router's buffer (1,297 requests, on average), for simplicity's sake we considered the G/G/1 model with no queue capacity—which is well developed in the literature (Whitt, 1983), (Arenales et al., 2007)—with easily computable closed-formulas for some performance measures. The Krämer and Langenbach-Belz (Equation 3), and Buzacott and Shanthikumar approximations (Buzacott & Shanthikumar, 1993, 1993) (Equations 4 and 5) were used to calculate the average number of requests in the link (these are approximations with FIFO queue discipline):

$$E(L)=\rho+\frac{\left(\rho^2\left(C_x^2-C_s^2\right)\right)}{\left(2(1-\rho)\right)} \quad \text{Kraemer e Lagenbach-Belz (K-L)} \tag{3}$$

$$E(L)=\rho+\frac{\left(\rho^2\left(1+C_s^2\right)\left(C_x^2+\rho^2C_s^2\right)\right)}{\left(2(1-\rho)\left(1+\rho^2C_s^2\right)\right)} \quad \text{Buzacott and Shantikumar (BS-1)} \tag{4}$$

$$E(L)=\rho+\frac{\left(\rho^2\left(1+C_s^2\right)(2-\rho)C_x^2+\left(\rho^2C_x^2\right)\right)}{\left(2(1-\rho)\left(2-\rho+\rho C_s^2\right)\right)} \quad \text{Buzacott and Shantikumar (BS-2)} \tag{5}$$

where C^2_x e C^2_s are the quadratic coefficients of variation of the random variables x and s, being defined from $E(x)$, $E(s)$, $V(x)$, and $V(s)$, as (Equations 6 and 7):

$$C_x^2=\frac{V(x)}{E(x)^2} \tag{6}$$

$$C_s^2=\frac{V(s)}{E(s)^2} \tag{7}$$

We also analyzed the application of the classical queue model $M/M/1/k$ (Kleinrock, 1975; Gross & Harris, 1998; Arenales et al., 2007) (Equation 8):

$$E(L)=\frac{\rho}{1-\rho}-\frac{(k+1)\rho^{k+1}}{1-\rho^{k+1}} \tag{8}$$

For the sample data, we have that C^2x = 3.99 and C^2_S =0.83 using (6) and (7), indicating that the distribution of the interval between arrivals has a high variability. With the sample values of ρ, λ, and μ and the computed value of $E(L)$, calculated by the approximations in (3), (4), (5), and (8), one can easily apply Little's formula to obtain other system performance measures (Gross & Harris, 1998; Arenales et al., 2007).

5 Results

Analysis of the sample data collected at the University shows that a large part of the requests (46.05%) use Transmission Control Protocol (TCP). This protocol makes several attempts to transmit requests, until the timeout is reached and the request is discarded. Thus, each retry attempt generates duplicate requests. In the sample analyzed, all attempts to send requests represent 10.4% of the sample. However, discounting 4.6% of duplicated requests, there is a 5.8% real loss of requests (discarded), observed with the Wireshark software.

The statistical analysis performed on the collected sample showed that the Lognormal distribution was adherent to the arrival process, with a quadratic error of 0.00612, but in the service process none of the 26 theoretical distributions evaluated in Arena's Bestfit and Input Analyzer passed the adherence tests. Some studies in the literature report this difficulty in Internet modeling (Floyd & Paxson, 2001), mainly due to the traffic control exercised by the TCP family. However, as this study focuses on the modeling of a single link—not the entire Internet—it was important to verify the extent of this problem, also described by Willinger & Paxson (1998).

TCP uses end-to-end congestion control, which means that the sender limits or increases the rate of data transmission to the TCP connection based on the sender's assessment of perceived network congestion. This is why TCP is described as self-regulated. The TCP connection is composed of a receiver buffer, a sender buffer, and several variables, including a congestion window (cwnd), which limits the packet sending rate of a sender's TCP. At the beginning of each RTT (round-trip time), the sender sends its packets according to the size of the established cwnd, and at the end receives acknowledgment for the data, a signal that all packets have been sent correctly. When a duplicated loss event or three ACKs (acknowledgements of receipt) occur, the sender reduces its cwnd by using the so-called multiplicative decrease, thereby halving the cwnd. However, there is a minimum threshold for the size of this window, which is 1 maximum segment size (MSS).

TCP recognizes that there is no congestion in the network when it receives ACKs, so it increases the cwnd slowly at each round-trip time (additive increase). This TCP behavior of increasing the congestion window slowly, and then halving it sharply, engenders behavior which when graphed resembles sawtooth diagrams. During the start of a TCP connection, the slow start phase occurs when the sender transmits at a small rate (usually 1 MSS), and then increases its rate exponentially by doubling the cwnd value at each round-trip time until a loss event occurs. The TCP sender can also go through the slow start phase after a time-out event, setting the congestion window to 1 MSS and increasing exponentially until cwnd reaches half the value it had before the event (threshold) (Kurose & Ross, 2006). This TCP control procedure influences the modeling and simulation processes.

The simulation model was implemented in Arena software, as previously described, considering the Lognormal distribution in the arrival process and an empirical distribution in the service process. The loss of requests for simulation was 12.12%, which can be considered a reasonably accurate estimate, since in the sample, this loss without the discount of resends and duplications was 10.4% (in the simulation, the resubmission and duplication process cannot detected). It is worth mentioning that the control of congestion exercised by TCP should cause a loss of 10.4%, but only caused an actual loss of 5.8%.

Regarding the analysis of the analytical queue model G/G/1, the value of E (L) - average number of requests in the link, calculated with approximate closed formulas, was compared with the simulated E (L) value. The results of this comparison show that the approximations used for the G/G/1 model are not adequate, since the simulated E (L) was 1280.7, while in the analytical model this value was 238.0, as calculated by the approximation KL in (3); 239.37, as calculated by the approximation B-S1 in (4); and 221.10, as calculated by the approximation B-S2 in (5). This is probably because these approximations do not behave well in congested systems that perform self-adjustments as the system tends to saturation (heavy traffic), through the FIFO disciplines based on the TCP protocol (different from the basic FIFO discipline considered in the approximations of the G/G/1 model).

As part of an analytical comparison, E (L) was also calculated using the basic M/ M/1/k queue model, using the expression E (L) in (8). Again, the result E (L) = 98.99 was very far from the simulated value. In other words, this model, in spite of considering the limitation of the queue size (unlike the G/G/1 model), also does not adequately represent the system in terms of the assumed assumptions of the Markovian arrival and service processes, and the FIFO discipline, without self-adjustment. This result reinforces reports from the literature that this model may not be adequate for the analysis of these heavy traffic systems, although it has been presented in previous studies as an alternative to calculate the average end-to-end delay by convolving queues of all links (Kleinrock, 1975).

Given that the loss at peak times is approximately 6% (sample size), based on the experiments carried out we conclude that the University network is reasonably well sized for the service processes at the time of data collection, as well as for the number of users using the link. In terms of QoS in the Network, although the packet loss is, on average, relatively small (5.8%) in departments, at peak times this percentage is much higher, more than double. This was observed by injecting traffic and measuring within each department, and between departments. These experiments were carried out in the Department of

Computing and in the Department of Production Engineering of the University, and between the two. This problem can be explained by the fact that many access machines still have low capacity 100Mbps Internet network cards.

6 Final considerations

This work analyzed the performance of a University communication network dealing with link congestion in peak periods. The system's bottleneck was identified and the congestion problem was empirically characterized through data collection and analysis, capture of traces of log packets, and the study of delays and losses. For the analysis of system performance, we used models based on discrete simulation and also some analytical queueing models. In the heavy traffic situations of this system, when the use of the link is close to 100% of its capacity, the approach that best characterized the data traffic and produced good estimates for the performance measures of the system was the simulation. In terms of loss of requests, without discounting the data, the simulation model implemented in the Arena software obtained values close to those sampled in the system. As this model does not explicitly consider the control exercised by the TCP module, a difference between these results was expected, in addition to random sampling deviations. A prospect of QoS in the network was presented, as well as the proposal of solutions for its improvement.

As for the analytical modeling, the approximations used for the G/G/1 queue model were not adequate, since the model considers neither limitations in the size of the buffer (in the network studied the queue capacity is limited), nor in the control of the TCP environment, which resends and discards requests in the queue after some waiting time. In the same way, the well-known M/M/1/k Markovian model was also not adequate, because the processes of arrival and service in the studied system do not follow exponential distributions, and this model also neglects the TCP behavior.

This study was an exploratory investigation, and further analyses should be conducted involving other communication networks in universities and organizations, to better evaluate the performance and effectiveness of the methods used here. Specific studies to add the dependence of the service processes on the arrival process of the data packets, due to the control of the TCP protocol, will be fundamental for a better characterization of the data traffic in communication networks. Another relevant future research direction involves the study and development of more elaborate and appropriate analytical queueing theory models to represent these networks. These should explicitly consider control of the TCP protocol, and use stage and phase-like methods (Kleinrock, 1975; Neuts, 1989), for instance, to better represent the variability of the arrival and processing processes of the data packets and the limitations of the buffers involved.

Acknowledgments

The authors would like to express their gratitude to the two anonymous reviewers for their helpful comments; to CAPES for its financial support; and to the Department of Computing and the General Office of Informatics (SIn) of the University. Special thanks to analysts Marcelo Duarte, who provided relevant data about the logical and physical structure of the University's Internet Network, together with data collection at the SIn, and Gleise Segatto, who carried out the data collection in the Department of Computing.

References

Arenales, M., Armentano, V., Morabito, R., & Yanasse, H. (2007). *Pesquisa operacional*. Rio de Janeiro: Elsevier.

Banks, J., Carson, J. S., Nelson, B. L., & Nicol, D. M. (2010). *Discrete-event system simulation*. (5th ed.). New Jersey: Prentice-Hall.

Brito, S. H. B. (2012). *Laboratório de tecnologia Cisco em infraestrutura de redes*. São Paulo: Novatec.

Bu, T., & Towsley, D. F. (2001). Fixed point approximations for TCP behavior in an AQM Network. In *Proceedings of the 2001 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems - SIGMETRICS'01* (pp. 216-225). New York: ACM.

Buzacott, J., & Shanthikumar, J. (1993). *Stochastic models of manufacturing systems*. Nova Jersey: Prentice Hall.

Chwif, L., & Medina, A. C. (2010). *Modelagem e simulação de eventos discretos*. 3. ed. São Paulo: Bravarte.

Crovella, M. E., Taqqu, M. S., & Bestravos, A. (1998). Heavy-tailed probability distributions in the World Wide Web. In R. J. Adler, R. E. Feldman & M. S. Taqqu, (Eds.), *Practical guide to heavy tails: statistical techniques and applications* (pp. 3-25). Boston: Birkhäuser.

Floyd, S., Paxson, V. (2001). Difficulties in simulating the internet. IEEE-ACM Transactions on Networking, 9(4), 392-403.

Gross, D., & Harris, C. (1998). *Fundamentals of queueing theory*. 3. ed. New York: John Wiley.

Harlan, R. C. (2003). Network management with Nagios. Linux Journal, 2003(111), 3.

Kleinrock, L. (1975). *Queueing systems: theory* (Vol. 1). New York: John Wiley & Sons.

Kurose, J. F., & Ross, K. W. (2006). *Redes de computadores e Internet*. São Paulo: Pearson Addison-Wesley.

Lee, C., Lee, D. K., Yi, Y., & Moon, S. (2011). Operating a network link at 100%. In *12th Passive Active Measurement Conference (PAM)*. Atlanta.

Leung, Y., & Suri, R. (1990). Performance evaluation of discrete manufacturing systems. IEEE Control Systems Magazine, 10(4), 77-86.

Mahajan, P. S., & Ingalls, R. G. (2004). Evaluation of methods used to detect warm-up period in steady state simulation. In Ingalls, R. G., Rossetti, M. D., Smith, J. S., & Peters, B. A. (Orgs.), *Proceedings of the 2004 Winter Simulation Conference* (pp. 663-671). Los Alamitos: IEEE Computer Society.

Neuts, M. F. (1989). *Structured stochastic matrices of M/G/1 type and their applications*. New York: Marcel Dekker.

Oetiker, T. (1998). MRTG: The Multi Router Traffic Grapher. In *Proceedings of the 12th Conference on Systems Administration* (pp. 141-148). Berkeley: USENIX Association.

Orebaugh, A., Ramirez, G., & Beale, J. (2006). *Wireshark & ethereal network protocol analyzer toolkit*. New York: Elsevier Science. 448 p.

Ostermann, S. *tcptrace TCP dump-file analysis tool*. 2012. Recuperado em 7 de outubro de 2013, de http://www.tcptrace.org/.

Pegden, C. D., Shannon, R. E., & Sadowski, R. P. (1990). *Introduction to simulation using SIMAN*. (2nd ed.). New York: McGraw-Hill.

Pinto, R. P., Guimarães, E. G., Cardozo, E., & Magalhães, M. F. (2003). Incorporação de qualidade de serviço em aplicações telemáticas. In *21º Simpósio Brasileiro de Redes de Computadores* (pp. 331-346). Natal.

Rodríguez-Dagnino, R. M. (2004). On the Pareto/M/c and Pareto/M/1/K Queues. In *Proceedings of the 2nd SPIE 5598*. Performance, Quality of Service, and Control of Next-Generation Communication Networks (pp. 183-193). Philadelphia, 2, 183-193. http://dx.doi.org/10.1117/12.570535.

Srikant, R. (2004). *The mathematics of internet congestion control*. Boston: Birkhäuser. http://dx.doi.org/10.1007/978-0-8176-8216-3.

Van Woensel, T., Andriansyah, R., Cruz, F. R. B., Smith, J. M. G., & Kerbache, L. (2010). Buffer and server allocation in general multi-server queueing networks. International Transactions in Operational Research, 17(2), 257-286. http://dx.doi.org/10.1111/j.1475-3995.2009.00746.x.

Whitt, W. (1983). The queueing network analyzer. The Bell System Technical Journal, 62(9), 2779-2815. http://dx.doi.org/10.1002/j.1538-7305.1983.tb03204.x.

Willinger, W., & Paxson, V. (1998). Where mathematics meets the internet. Notices of the AMS, 45(8), 961-970.