International Telecommunications Symposium – ITS2002, Natal, Brazil

HTTP Traffic Modeling: Development and Application

Kleber V. Cardoso, José F. de Rezende
Universidade Federal do Rio de Janeiro, Rio de Janeiro RJ, Brasil

Abstract—**This paper presents a new HTTP traffic model based on the aggregation concept. The model development, evaluation and application are shown. In addition to the basic HTTP traffic characteristics, the traffic model has an easy and accurate load control. Some examples are provided to present the traffic model usage.**

I. Introduction

In the last years, web has maintained its status of the Internet killer application and there is not clues that the situation will change soon. The HTTP, responsible to transfer web content, dominates the traffic traces. According to recent statistics from CAIDA [1], HTTP typically represents from 47% to 69% of the bytes sent over the Internet. The number of services and the amount of information available in the web keeps growing and this looks like to be a dominant trend for some years. First, because web is a suitable application for any kind of service which is based on text and graphics. Second, HTTP is adequate to transfer different types of files, from small Java applets to huge non-stream videos. Third, and most important, web has become a kind of universal interface. The simple and friendly "look and feel" of the web pages have allowed different services and information to be widely available to almost any system regardless the hardware or the operating system.

Within this context, it is important to understand how HTTP traffic behaviors in order to understand and make improvements in the Internet. One way to do this is developing and using HTTP traffic models. Many works have been made in this area [2], [3], [4], [5], [6], using different approaches in the development. The majority proposes models that describe a common web client behavior [2], [5], [6], with different levels of details. A small number of works [3], [4] focuses on the behavior of a group or aggregation of web clients, which has as main advantage the simplicity. These works do not present precise methods to control the network load generated by their models. In many cases this is a wanted characteristic since it can represent a control over network condition. In many situations, the lack of examples precludes people to utilize the existing models, which drives to repeated job on traffic model development. This paper proposes improvements in these subjects.

This paper proposes a new HTTP model, which is based on the concept of aggregated behavior and presents two main advantages: small number of parameters and easy and precise load control. The development and features of the model are detailed. The procedures for evaluation of the model properties are also shown. Examples of the model utilization are shown and results are presented and discussed.

Kleber Vieira Cardoso and José F. de Rezende are in the Grupo de Teleinformática e Automação, COPPE, Programa de Engenharia Elétrica, Universidade Federal do Rio de Janeiro, Rio de Janeiro - RJ, Brasil, 21.945-970, Phone: +55 21 2260 5010. E-mails: {kleber,rezende}@gta.ufrj.br.

This paper is organized as follows. Section 2 reviews the main techniques used in traffic modeling. Section 3 presents a new HTTP aggregation model and its features. Section 4 shows some uses of the generator based on the proposed traffic model. At last, in the section 5, conclusions and final comments are drawn.

II. HTTP Traffic Modeling

Simulation is a widely used tool for computer networks evaluation, but it is important to have suitable traffic models to get useful results. The majority of works about web traffic modeling has concentrated on developing client models, which focus on the behavior of individual web clients. Other approach is to model the aggregation behavior of several web clients, i.e., an aggregation model. Both models have advantages and shortcomings. The client model is able to capture more details of the application, so it is in some sense a better mimic. However, this higher level of detail brings more complexity to the model because it demands the understanding and configuration of more parameters. In some situations the level of detail does not help in the evaluation, since many of the details simply does not matter.

The aggregation model is generally a coarser approximation of the real traffic. In spite of this, its simplicity allows it to simulate some conditions and identifying behaviors that are difficult with client models. In addition, client models tend to consume more computing resources than aggregation models when representing a large number of web clients in a simulation environment. In both kinds of model an important issue is the choice of application's characteristics that are desired, since they are the focus of the model development. Some examples of these characteristics are burstiness, network load, long-range dependency, etc.

A model (aggregation or client) utilizes parameters to reproduce certain properties of the web application. Some examples of parameters are transfer size, interval between pages, number of objects per page, etc. To describe these parameters two approaches are used: one based on real traffic samples and other analytic. The models created using these approaches are known as structural models [7], since they try to characterize the traffic nature.

The use of real traffic samples consists of describing a certain application parameter through a set of predefined values which are collected from a real network environment. The main advantage of this method is the easy of implementation and accurate representation of a known system. However, this approach treats the generated traffic as a "black-box". In addition, the generator traffic based on this kind of model becomes hard to set up since new conditions or variable demands are not easy to configure.

The analytic approach lies in the use of probability distributions to describe a certain parameter. A probability distribution tells how a sequence of random values behaves, provided that there is available enough number of samples. When the distribution is known, it allows to generate new and different sequences of values following such distribution. The main drawback of this approach is the difficulty found in identifying and configuring the distribution that describes adequately the sequences of random values of the application parameters.

A third approach can be included, which consists of using known abstract processes to try to capture only the statistical traffic properties with independence of the subjacent mechanisms of traffic generation. This approach is efficient and quite simple to implement. Moreover, this approach is useful when specific features are of interest. For example, self-similarity can be easy reproduced by a fBm process (fractal Brownian motion). However, this sort of method does not take into account important factors from the traffic profile and neglects elements such as the congestion control of TCP, which is an important feature of HTTP traffic. Models based on this approach are known as "behaviourist" [7].

III. The Traffic Model

Following the proposal presented in the last section, a good start point for model development is to establish its objective, i.e. what application it will describe and what the focus or features are intended for the model. So it is important to establish a profile before beginning the model development. Despite the simplicity, this methodology can minimize the development time, since it has a clear focus and try to avoid unnecessary complexities.

In this work, it was established that the model should have few parameters. In addition, the traffic generator based on the model would be used as input to bottleneck links. Thus, the model could ignore several details related to individual web clients since the appropriated aggregation behavior was kept.

Traffic generators, sometimes called workload generator, generally do not have a simple way to adjust the load. It is common to use the mean load generated by a client or a set of them to a specific network configuration. To vary the load, the number of clients are varied. However, if the network configuration is changed then it is necessary to recompute the new load. Moreover, in this conventional way, the mean load is measured during all simulation time and measurement in short intervals can be always far from the mean. Thus the objectives of the model are an easy way to adjust load and samples near to the mean in time intervals shorter than the whole simulation time.

According to queue theory [8], the concept of load or utilization factor can be written as

$$\rho = R/C$$

in which
R - (work) arrival rate, and
C - maximum rate or system capacity.

The work that a new customer [1] brings to the system is equal to service time it requires. So if system has a unique server (e.g. a bottleneck link router) the load can be rewritten as
$\rho = \lambda\overline{x}$
where
λ - mean arrival rate of customers, and
$\overline{x}$ - mean service time.
Considering the context of HTTP traffic and bottleneck link, the last equation can be modified to $\overline{x} = \overline{L}/C$
in which
$\overline{L}$ - mean transfer size, and
C - maximum rate or system capacity or link capacity.
Thus,

$$\rho = \lambda\overline{L}/C \tag{1}$$

ρ is main adjustment parameter and is used to choose different load conditions. ρ describes the time percentage that the system is busy given a measurement window. C is fixed to a certain network configuration. $\overline{L}$ controls the mean size of web transfer. $\overline{L}$ may describe the size of a web page/object if the interest is HTTP/1.0 without keep-alive or the size of group of pages and objects if HTTP/1.0 with keep-alive or HTTP/1.1 are the protocols of interest. At last, λ describes the connection arrival rate, which varies according to $\overline{L}$ in order to accomplish the established ρ. Thus the arrival rate can be written as
$\lambda = \rho C/\overline{L}$
Since
$T = 1/\lambda$
describes the interval between connection arrivals, then

$$T = \overline{L}/\rho C \tag{2}$$

The distributions that describe the parameter $\overline{L}$ have been widely studied [3], [9], [6], [10] and there is some convergence. The majority agrees on a heavy-tail distribution to describe this parameter and examples of configuration are in table I. The label information is used for future citations of the distributions.

TABLE I
Some distributions that describe the parameter $\overline{L}$.

Distribution	Configuration	Label	Reference
Pareto	mean - 4100 shape - 1.95	HTTP-1	[10]
Pareto	mean - 4100 shape - 1.35	HTTP-2	[10]
Lognormal	mean - 4827 std. dev. - 41008	HTTP-3	[3]
hybrid: Pareto - 7%, Lognormal - 93%	mean - 1463000 shape - 1.1	HTTP-4	[4]
	mean - 27600 std. dev. - 59714		
hybrid: Pareto - 12%, Lognormal - 88%	mean - 10558 shape - 1.383	HTTP-5	[11]
	mean - 7247 std. dev. - 28765		

A. *Study of Network Load*

In this paper, the system will be always a router, but many concepts can be extended to other network equipments such as switches. Based on this, it is important to define what network load means. Sometimes, the network load refers to a bandwidth

[1] To avoid confusion with the word **client** that is used to refer to web software

use, which can vary from 0 to 100% of all output link throughput. A more accurate approach is to consider the load as the time use of the router. To measure the network load as the time use of the system (router) is necessary to establish a measurement interval or window. This interval is a time quantity in which the network load is measured. Initially, the measurement interval can be arbitrary and vary from milliseconds until the whole simulation time. However, as it has been said, many times is useful to have time interval smaller than the whole simulation.

Based on these concepts and on equation 1, ρ represents the effective network load by occupying the system during a percentage of time of a certain measurement interval. E.g., if $\rho = 0,9$ (90%) and the measurement interval is 10 seconds, the system should be in use during 90% of this time, i.e., 9 seconds. Two main issues arise about the model.

First, it was established that in measurement intervals smaller than the whole simulation time, the load should get close to the mean value. This would be affected by transfer sizes because short-term transfers could fulfil smaller measurement intervals, while long-term transfers can present large intervals. In addition, HTTP uses TCP as transport protocol, which causes the transfers to happen in variable rates. This would create "distortions" on the sequences of system use. This way, the measurement interval would be affected again.

Second, the protocol TCP is reliable and retransmit lost packets mainly due to buffer overflows. If more than a copy of a packet pass by the network point where the load is being measured, load would be higher than ρ. By another hand, since the model is designed to bottleneck links, measurements are took place at this point, an thus the losses should occur at arriving in the buffer and duplicates would not account.

To evaluate the previous issues, simulations were done to verify the relation between ρ and the measured mean load in different intervals. The methodology applied in this work was based on [12]. In the simulations were used traffic sources with transfer sizes of 1, 5, 50, 500 and 5000 KBytes, and also sizes which obey the distributions shown in table I.

The topology chosen to make the simulations is a kind of dumb-bell and is presented in figure 1. The buffer size of the bottleneck link follows the suggested by [13], which is $2 * Bw * RTT$, where $Bw = C$ and RTT is the longest round trip time in the network. TCP Reno was the implementation choice, since it is still one of most used. The queue discipline is FIFO and queue management is Drop Tail, i.e. the traditional configuration of a router. Packets come from s_1 and s_2 belong to two different traffic classes, that is, they are marked differently. In this section this is not take into account since there is no packet differentiation. This configuration is used to keep an uniform environment in all experiments, including the ones that have packet differentiation.

Figures 2(a) and 2(b) present the results when ρ is based on a bottleneck link of C and the link really has this capacity. Simulations with bottleneck link of $10C$ were also run in order to verify loads beyond 100%. The results were similar to the ones presented by figures 2(a) and 2(b), thus they are not shown in this work. In figures 2(a) and 2(b) the load is a mean which is taken after the first 50 seconds until the end of the simulation that happens in 500 seconds. The beginning of the simulation is discarded in order to eliminate the transient.

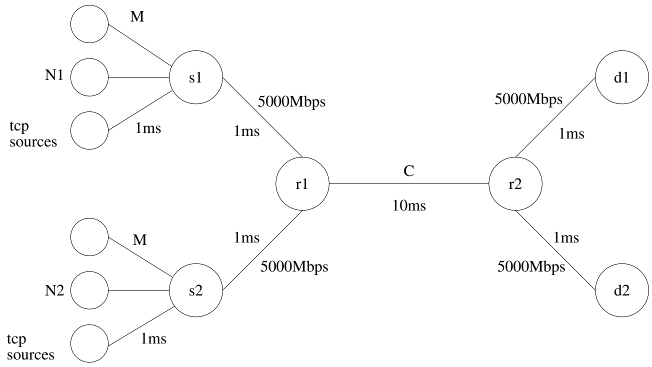
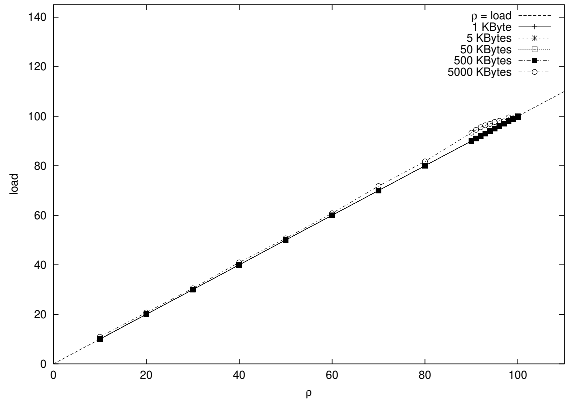
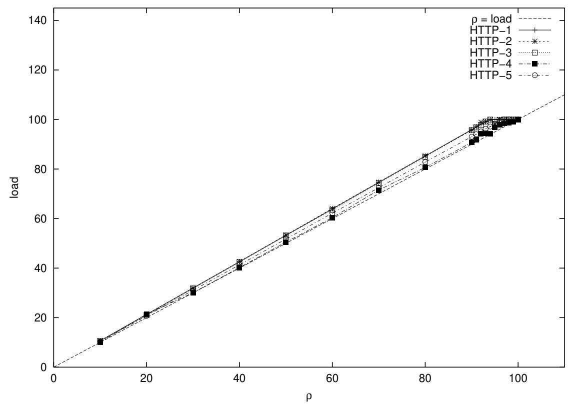


Fig. 1. Topology.



(a) Fixed size transfers.



(b) HTTP transfers.

Fig. 2. Load and ρ relationship in a bottleneck link of capacity of C.

Figure 2(a) shows that transfer sizes of 1, 5, 50 and 500 KBytes present a good match for ρ and load. By another hand, transfer sizes of 5000 KBytes exhibit a difference between ρ and load from 70% when the link capacity is C. Experiments have shown that the reason is bust behavior of TCP combined with his rate control. Since this transfer size is large enough to maintain the system in use for a long time, the TCP has the opportunity to rise the transmission window beyond the bottleneck link capacity which makes buffer overflows. Losses demand retransmissions and decrease the goodput. This makes transfers to be stretched along the time. Thus, system gets a "debt" of idle time which it can not vanish until the simulation end because it happens abruptly when the simulation time reaches 500 seconds.

Table II helps understanding the phenomenon described. This table gives additional information about the traffic behavior when ρ is varied from 70% to 90%. $\rho = 70\%$ was chosen, since it is the start point of the "distortion", according to figure 2(a). In addition, were used two buffer configurations, one equal to figure 2(a) and other 10 times bigger. The last configuration intend to offer enough buffer space to accept long bursts. The table exhibits a significant higher number of losses when the buffer is B. It can be also noted a trend to increase in the number of simultaneous transfers as ρ rises. The time without active transfers is longer with buffer $10B$ than with B, which makes mean load not match ρ. Simulations have shown that $\rho = load$ when buffer is $10B$ as is the case when bottleneck is $10C$. The results are not presented here due to size limitations.

TABLE II
DETAILS OF 5MB TRANSFER SIZE WITH DIFFERENT BUFFER VALUES.

	Losses (pkts)		No transfers (secs)		Simult. transfers	
ρ	B	10B	B	10B	B	10B
70	29679	0	36.01	100.01	2	2
75	21353	0	29.25	70.49	2	2
80	22854	0	0	51.30	5	2
85	21790	0	0	23.41	5	2
90	22256	0	0	1.31	6	2

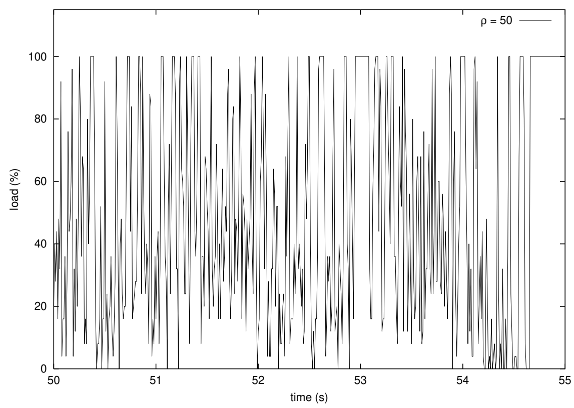
Since large transfer sizes can disturb the relation between ρ and load, it is important to evaluate in which rate this values appear in HTTP traffic. The evaluation was based on the transfer sizes distribution used in previous works widely cited. Table III exhibits the percentile of some typical transfer sizes. In this table is described the results of 100 sequences, with 100 thousand sample values each one. The distribution were based on [3], [4], [10] and [11]. As it can be seen, the long-term transfers happen rarely. It was also observed that these long-term transfers take place in a sparse manner.

TABLE III
PERCENTILE OF SOME DISTRIBUTIONS.

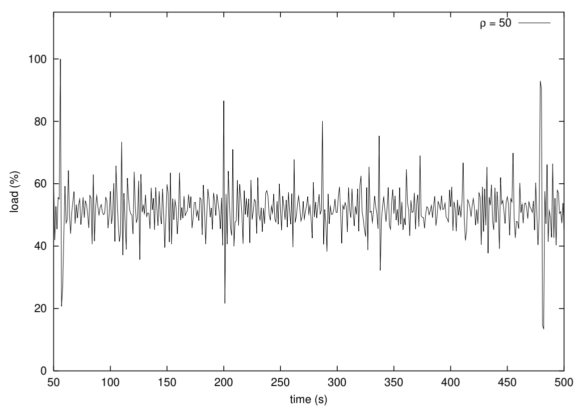
	$< 5K$	$< 50K$	$< 500K$	$< 5M$
HTTP-1	83.30%	99.81%	99.99%	100.00%
HTTP-2	87.63%	99.45%	99.97%	99.99%
HTTP-3	85.38%	98.48%	99.94%	99.99%
HTTP-4	26.20%	86.63%	99.78%	99.96%
HTTP-5	73.18%	97.66%	99.96%	99.99%

Figure 2(b) shows always $\rho < load$. Actually, the difference is small but for the sake of accuracy a detailed analysis was made. Surprisely, the reason is only the NS simulator. In NS, the creation of each traffic source demands the configuration of the packet size. This packet size is fixed and does not change whatever the size of data to be sent. So, if the transfer size is a multiple of the packet size then there is a good match between ρ and load. By another hand, there is always a packet which carries less data than it can and a padding is used to complete the size. This puts more bits in the network than is was previously established by ρ. E.g., if the packet size is 1000 Bytes and the transfer size is 1200 Bytes, then 2 packets of 1000 Bytes will be transmitted. It were run simulations that show this results, but they are not exhibited here due to size constraints.

Another important part of the model is the measurement interval, since it was established as a model objective. To analyze how load varies in different measurement intervals, the mean was measured from 10 milliseconds until 10 seconds. These values are representative because they describe the ability of the model in controlling the load under different time intervals. Figures 3(a) and 3(b) show the mean load behavior when $\rho = 50\%$ and measurement intervals of 10 milliseconds and 1 second. In these measurement intervals, the mean load presents significant variation. Figure 4 exhibits the mean load for different values of ρ and interval of 10 seconds. Under these intervals, the mean load presents a close match to ρ.



(a) Mean load variation under 10ms time interval.



(b) Mean load variation under 1s time interval.

Fig. 3. Mean load variation.

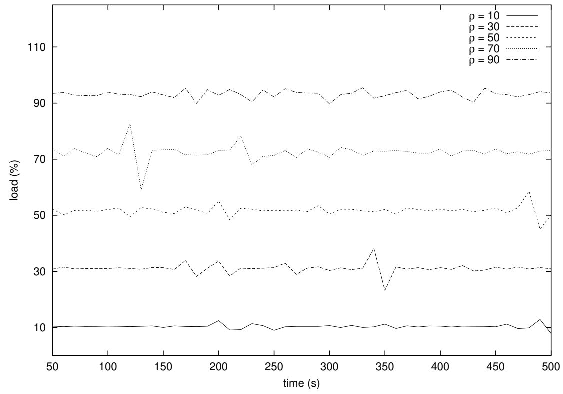


Fig. 4. Mean load variation under 10s time interval.

B. Simultaneous Connections

The model presents some interesting characteristics related to number of simultaneous transfers or connections. Initially, the model was based on HTTP/1.0, but by choosing suitable values for distribution of transfer sizes, HTTP/1.1 can be also resembled. Thus, the following evaluation will consider each HTTP transfer as a TCP connection, even though it can be modeling more than a page/object per connection. The use intended to the model does not care about this simplification.

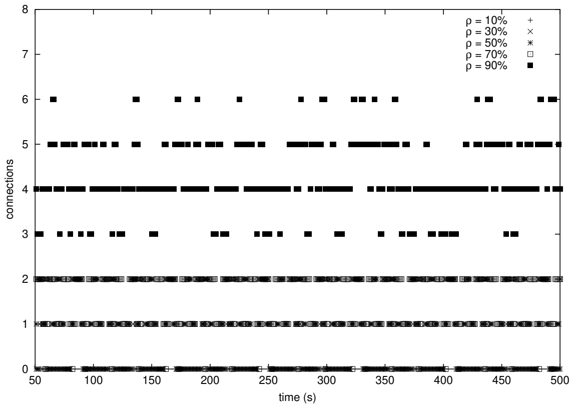
TCP protocol gives a special contribution in the way number of simultaneous connections varies. First, thanks to slow-start algorithm, sequences of short-term transfers tend to have a high level of overlapping. Second, slow-start and congestion-avoidance algorithms help to increase the number of simultaneous connections as ρ rises. Figures 5(a) and 5(b) illustrate these situations. Figure 6 summarizes results of fixed size and HTTP transfers. It can be seen that as ρ gets closer to 100% the number of simultaneous transfers increases significantly, in special for HTTP transfers.

C. Self-similarity

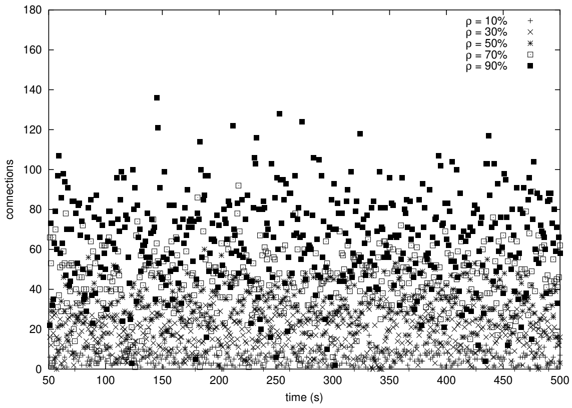
In computer networks, asymptotically second-order self-similarity can be summarized as the property of having observable bursts on several (or all) time scales. Self-similarity is mainly evaluated by the Hurst parameter (H), which is described in the following interval: $0.5 < H < 1$. As $H \to 1$, the degree of self-similarity increases.

Some works have highlighted the existence and consequences of self-similarity in web traffic [9], [10]. The interest on self-similar process arises due to the consequences on network behavior. It has been shown that self-similarity can affect, in some extent, the buffers of network components and then increase loss rate.

In this context, it is important that an HTTP traffic model presents self-similarity if an experiment demands. The proposed model was evaluated and and sample result is presented in figure 7. The Hurst parameter was measured by the wavelet estimator introduced in [14] with minor modification to exhibit H as part of the graphic's title. The figure shows that traffic presented strong self-similarity ($H = 0.889$) with bursts varying from a few to hundreds of seconds, i.e. two orders of magnitude.



(a) Fixed transfer size of 5M.



(b) HTTP-5.

Fig. 5. Number of simultaneous connections.

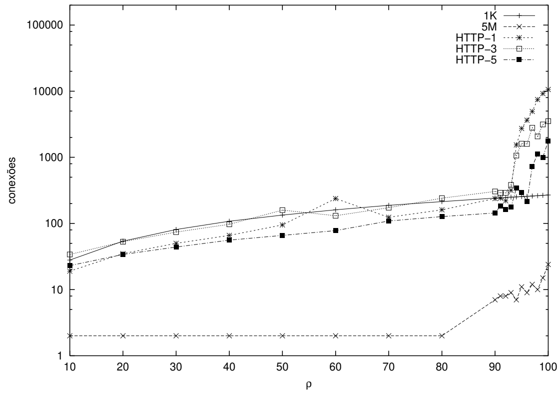


Fig. 6. Number of simultaneous connections under different loads.

IV. Model Application

Since the aggregation model presented in this paper was intended to be used as input to bottleneck links, some specific uses are adequate. Bottleneck links are basically routers, switches or similar equipments. In this kind of network elements, some mechanisms, policies and disciplines are of interest. Examples

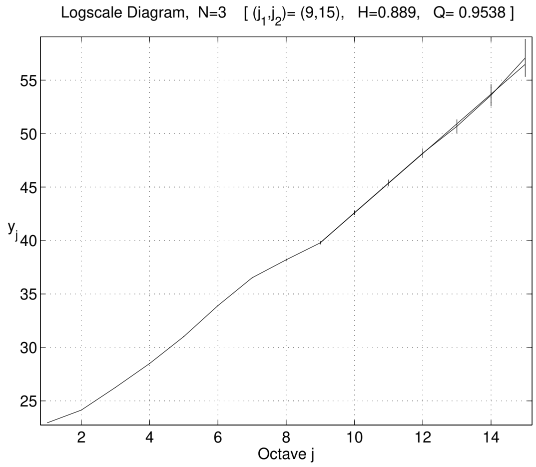


Fig. 7. Wavelet analysis of cumulative work process of HTTP-1 with $\rho = 95\%$.

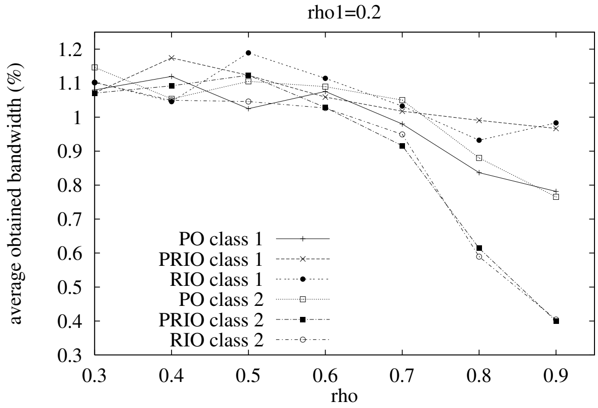


Fig. 8. Bandwidth of classes 1 and 2 as a function of network load.

of these are:

- queue management, e.g. RED, BLUE, REM, etc.;
- schedulers, e.g. WFQ, GPS, etc.;
- markers, shapers, etc.

The model was developed in such way that is easy to generate traffic for one or several classes under a unique load control. This is useful in evaluation of quality of service architectures such as DiffServ.

The model can also be used to evaluate congestion control algorithms since it present simple controls for load, transfer sizes and number of connections. In addition, the model can be applied as background traffic, which helps the evaluation of the influence of web traffic on other kinds of traffic.

Figure 8 shows an example of the model application. The simulation intended to evaluate the effectiveness of selective discard mechanisms for HTTP transfers. In order to assess the sensitivity of the discarding mechanisms, class 1 load was kept constant and the total load was increased up to 0.9. A desirable result would be the remaining of class 1 to keep performance constant with the increase of the total load. As can be seen in figure 8, class 1 obtained bandwidth does not significantly change under PRIO (Push-out RIO) and RIO (RED with In/Out bit) policies. It is worth noting that the joint use of push-out and RIO does not offer any improvement to both classes. PO (Push-Out) is the most sensitive mechanism to the load increase, and it does not significantly differ the priority classes. Detailed comments about this experiment and some other uses of the model can be viewed in [15].

V. Conclusion

In this paper, a new HTTP aggregation model was proposed. The development steps were detailed and the model evaluation shown its benefits and shortcomings. The traffic model presented the ability to reproduce some important HTTP characteristics, which include transfer sizes and self-similarity. To control network load in an easy and precise manner is the main model feature. Examples of how to use the traffic model are also illustrated. As future plans to the traffic model, there are an under-development implementation based on sockets to be used in real network environments, evaluation of new mechanisms and model improvements.

Acknowledgments

This work is supported by CNPq, CAPES, COFECUB e FAPERJ.

References

[1] "CAIDA (Cooperative Association for Internet Data Analysis) - Characterization of Internet traffic loads, segregated by application," http://www.caida.org/analysis/workload/byapplication/.

[2] Bruce Mah, "An empirical model of http network traffic," in *Proc. INFOCOM'97*, Apr. 1997.

[3] Maurizio Molina, Paolo Castelli, and Gianluca Foddis, "Web Traffic Modeling Exploiting TCP Connections' Temporal Clustering through HTML-REDUCE," *IEEE Network Magazine*, vol. 14, no. 3, pp. 46–55, 2000.

[4] Paul Barford and Mark Crovella, "Generating representative web workloads for network and server performance evaluation," in *Proc. ACM SIGMETRICS Conference*, July 1998, pp. 151–160.

[5] Henrik Abrahamsson and Bengt Ahlgren, "Using Empirical Distributions to Characterize Web Client Traffic and to Generate Synthetic Traffic," in *IEEE/Globecomm'00*, San Francisco, Nov. 2000.

[6] Hyoung-Kee Choi and John O. Limb, "A Behavioural Model of Web Traffic," in *International Conference of Networking Protocol 99 (ICNP99)*, 1999.

[7] Eduardo Casilari, Arcadio Reyes, Francisco Javier Gonzalez, Antonio Diaz Estrella, and Francisco Sandoval, "Characterisation of Web Traffic," in *Internet Performance Symposium*, San Antonio, Nov. 2001.

[8] Leonard Kleinrock, *Queueing Systems*, ISBN 0471491101. John Wiley and Sons Inc., 1975.

[9] Mark E. Crovella and Azer Bestavros, "Self-similarity in World Wide Web traffic: Evidence and possible causes," in *Proc. of the ACM SIGMETRICS Conference on Measurement & Modeling of Computer Systems*, Philadelphia, 1996, pp. 160–169.

[10] Kihong Park, Gitae Kim, and Mark Crovella, "On the relationship between file sizes, transport protocols, and self-similar network traffic," in *Proc. IEEE International Conference on Network Protocols*, Oct. 1996, pp. 171–180.

[11] Paul Barford, Azer Bestavros, Adam Bradley, and Mark Crovella, "Changes in Web Client Access patterns: Characteristics and Caching Implications," in *Special Issue on Characterization and Performance Evaluation*, 1999.

[12] Mikkel Christiansen, Kevin Jeffay, David Ott, and F. Donelson Smith, "Tunning RED for Web Traffic," in *Proc. ACM/SIGCOMM'00*, Stockholm, 2000.

[13] "IRTF end2end-interest mailing list archive," ftp://ftp.isi.edu/end2end/end2end-interest-1998.mail.

[14] Darryl Veitch and Patrice Abry, "A Wavelet Based Joint Estimator of the Parameters of Long-Range Dependence," in *IEEE Trans. on Info. Theory, Special Issue on Multiscale Statistical Signal Analysis and its applications*, Apr. 1999.

[15] Kleber Vieira Cardoso, José Ferreira de Rezende, and Nelson L. S. Fonseca, "On the Effectiveness of Push-out Mechanisms for the Discard of TCP Packets," in *IEEE International Conference on Communications*, Apr. 2002.