Measurement and analysis of real network traffic

Costas Courcoubetis and Vasilios A. Siris
University of Crete and Institute of Computer Science, FORTH
P.O. Box 1385, GR 711 10 Heraklion, Crete, Greece
E-mail: {courcou, vsiris}@ics.forth.gr

Technical Report 252, ICS-FORTH, March 1999

Abstract

We present a platform for collecting and analyzing measurements of real network traffic. The measurements are collected by a high performance PC-based monitor, which can collect detailed traffic statistics on a high speed (155 Mbps) link, without disrupting the operation or affecting the performance of the network. The analysis is performed by a set of tools that use results from a recent theory of statistical multiplexing, and have the objective of answering important questions related to the management and dimensioning of networks carrying bursty traffic and guaranteeing some level of performance. Finally, we present case studies demonstrating the application of our analysis tools.

Keywords: Traffic measurements, traffic analysis, network management, network dimensioning, token/leaky bucket

1 Motivation

What combination of users can a network provider accept while ensuring some target level of performance? How many more users can the provider accept if he increases his link capacity or shapes incoming traffic? How can a user select the parameters of the traffic contract or service level agreement (SLA) with his provider? These are some of the questions that will arise with the deployment of networks that, unlike the traditional best-effort Internet, guarantee some level of performance or Quality of Service (QoS). An objective in such networks is the efficient use of network resources shared through statistical multiplexing by a number of bursty users, while ensuring some target level of performance.

Transformation of the Internet to a network that can support performance guarantees is slowly becoming a reality due to advances made by the Integrated Services (intserv), Differentiated Services (diffserv), and Multiprotocol Label Switching (mpls) working groups of the IETF (Internet Engineering Task Force). Deployment of technologies that support differentiated and guaranteed services is underway with the development of the QBone[1] by the Internet2 project. The aforementioned technologies enable, similar to ATM, the creation of services that involve a *traffic contract* or *Service Level Agreement* (SLA) between the user and the network. Furthermore, a single user (e.g., a large organization)

[1] See `http://www.internet2.edu/qos/qbone/`

can have more than one SLAs with the same provider, each for different traffic flows based on, e.g., subnet address or application type. According to such an agreement, the network provides some level of performance for the part of the user's traffic that is within some traffic profile. A widely used descriptor for a user's traffic profile consists of a peak rate and *token* or *leaky bucket*. Although the above capabilities increase the flexibility of networks in handling traffic with different performance requirements, they also increase the complexity of network management and dimensioning.

Traditional approaches to network management such as measuring the average load in intervals of the order of minutes are not adequate, since they fail to capture the burstiness of the traffic, which is important when some level of performance is guaranteed. Approaches based on queueing theory are also inadequate, since they require elaborate traffic models and cannot be effectively applied in the context of large multi-service networks. Furthermore, evidence of self-similar or fractal behavior of network traffic (e.g., see [8, 9]) has rendered traditional traffic source models inadequate.

Finally, technology enables the collection of detailed traffic statistics at very high speeds: 155 Mbps, 622 Mbps, and higher (see [3]). At such high speeds the amount of measurement data is huge. Indeed, a common question posed by both researchers and engineers is which measurements are most important, and how they can be used for network management[2].

Our approach to traffic measurement and analysis

Our approach involves the deployment of a platform for collecting detailed traffic measurements and the development of effective tools for analyzing the measurements and answering questions such as the ones identified in the beginning of this section. The platform is based on a flexible and high performance PC-based monitor, called OC3mon, which can passively collect detailed traffic measurements from a 155 Mbps optical fiber link. Since the station is passive it has no effect on the operation or performance of the network. The analysis tools use results from a recent theory of statistical multiplexing of a large number of bursty traffic streams, while ensuring some level of performance. The approach does not involve traffic models, but relies on actual traffic measurements from which the effective bandwidth of a stream, which is a scalar that measures the relative amount of resources used by the stream, can be estimated. In addition to answering important questions concerning network management, the approach can give guidelines for what traffic statistics, and in particular the time granularity of traffic measurements, are important for evaluating the performance of a link.

The above platform is being deployed for measuring and analyzing traffic on the link connecting the University of Crete network (UCnet) with the Greek Universities network (GUnet). It is also being considered for deployment in the network of the National and Capodistrian University of Athens.

The objective of the paper is to discuss typical questions regarding network management, identifying where networks offering performance guarantees differ

[2]See presentations at the recent Internet Statistics and Metrics Analysis (ISMA) workshop on passive measurement data and analysis at `http://www.caida.org/ISMA/isma9901/index.html`

from traditional best-effort networks, to describe the measurement platform and to discuss the methods and usage of the traffic analysis tools.

The rest of this paper is structured as follows. In Section 2 we identify typical questions concerning network management and dimensioning. In Section 3 we describe the traffic measurement platform. In Section 4 we describe the functionality and methods of the traffic analysis tools, and in Section 5 we present case studies demonstrating the application of the tools. In Section 6 we conclude the paper and identify related research activities we are currently pursuing. In Appendix A we give a detailed description of the methods implemented by the `msa` tool, and in Appendices B and C we show the usage of the `msa` and `lb` tools described in Section 4.

2 Questions concerning network management

Next we discuss some typical questions concerning the management and dimensioning of networks that carry bursty traffic and guarantee some level of performance. Our discussion considers a single link which is statistically shared by a number of bursty traffic streams of various types (e.g., web, video, and voice traffic). The performance or Quality of Service (QoS) measure that we consider is the probability of traffic being delayed more than some maximum value[3]. Finally, the link has some amount of resources (capacity and buffer), and services packets according to some scheduling discipline (e.g., First-Come-First-Served - FCFS, priorities).

Question 1: Resource usage

What is the amount of resources used by each traffic stream? For an aggregate traffic flow, what is the contribution of each individual flow (e.g., from a subnet or protocol) to the total amount of resources required for the aggregate flow?

Traditionally, the above questions are answered by measuring the load produced by each flow in intervals of the order of minutes. Such an approach is appropriate for best-effort networks, but is inaccurate for networks with QoS guarantees. For such networks, resource usage depends not only on the statistics of the stream, but also on the multiplexing that occurs at the link.

As an example, consider a link that multiplexes three types of traffic streams while ensuring some QoS, Figure 1. All three stream types have the same average rate. However, type (a) is smooth (constant rate) whereas the other two are bursty with different duration of "on" and "off" periods. What amount of resources does each stream type require? Both theory and experimentation indicate that the answer to this question depends on the link resources and the characteristics of the multiplexed traffic. Hence, when the capacity of the link is small, it might be the case that resource usage for stream type (a) is smaller than that for type (b), which in turn is smaller than that for type (c). For a larger capacity, it might be the case that resource usage for stream types (a) and (b) are the same, and smaller than that for type (c). Finally, for even larger capacities, it might happen that all three stream types have the same resource usage.

[3] The proportion of time a buffer of size B is full (probability of overflow) is equivalent to the probability of traffic being delayed more than $D = B/C$ in an infinite buffer served at rate C.

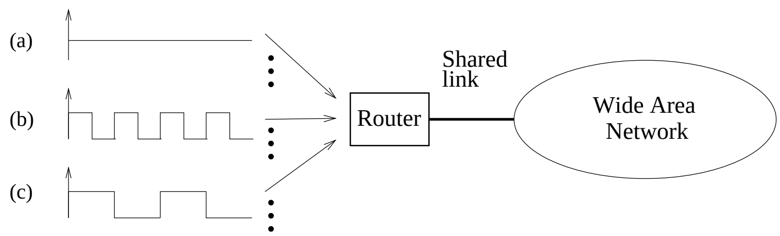

Figure 1: A stream's resource usage depends on the context, i.e., the link resources and the characteristics of the other traffic it is multiplexed with.

Question 2: Traffic measurements

What is the size of the time interval for traffic measurements (number of bytes or cells) so that all the statistics that affect a link's performance are captured?

At one extreme, one can capture traces of individual packets. In addition to producing a very large amount of data, particularly for high speed links, such fine statistics can contain more information than what is necessary for evaluating the performance of a link. An alternative is to measure the load in consecutive intervals of the same length. A longer measurement interval results in less amount of measurement data, but also in less statistical information, since fluctuations are averaged within each interval. The above question refers to the maximum length of the time interval so that all the statistics that affect a link's performance are captured.

Question 3: Acceptance region and link utilization

Given the link resources (capacity and buffer), what combination of traffic types can be multiplexed in the link while ensuring a target QoS? The combination of traffic types that can be accepted form the *acceptance region*.

A related question is the following: Given the link resources and the traffic mix (traffic types and percentage of each type), what is the maximum link utilization that can be achieved (equivalently, the maximum number of streams that can be multiplexed), while ensuring a target QoS? How much can the link's utilization increase when its resources increase by some amount?

Question 4: Quality of Service

Given the link resources, the traffic mix, and the link utilization (number of multiplexed streams), what is the level of QoS offered? How much does the QoS improve when the link's resources increase by some amount?

Question 5: Resource dimensioning

Given the number and combination of traffic streams, what is the minimum amount of resources (capacity and buffer) required to guarantee a target QoS?

Question 6: Traffic shaping

What effect does traffic shaping have on the link's multiplexing capability?

Question 7: Scheduling discipline

What effect does the scheduling discipline have on the link's multiplexing capability?

Question 8: Leaky bucket parameter selection

What leaky or token bucket parameters should be selected for a particular flow? How does traffic shaping affect the value of these parameters?

3 Traffic measurement platform

The proposed measurement platform is based on the OC3mon [1], which was developed by MCI's vBNS engineering team in cooperation with researchers from the National Laboratory for Applied Network Research (NLANR). The OC3mon is a high performance statistics collection station based on a Pentium PC equipped with two Fore ATM cards. The topology for the deployment of OC3mon for measuring traffic traversing the link connecting the University of Crete network (UCnet) with the Greek Universities network (GUnet) is shown in Figure 2. All IP traffic to and from the wide area network is handled by the router, hence traverses the 155 Mbps optical link connecting the switch and the router.

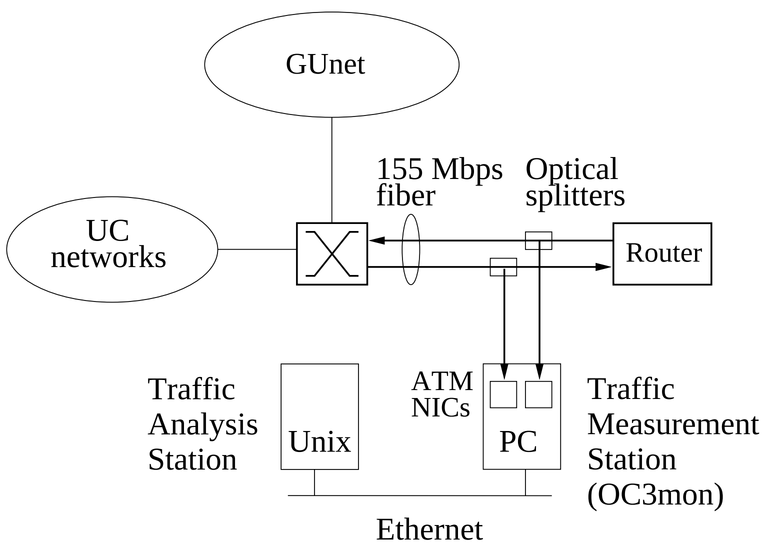

Figure 2: Measurement platform based on the OC3mon. Optical splitters guide a percentage of the light traveling over the two fibers to the receive ports of the two ATM cards, enabling measurements on both directions of the switch-router link.

Two optical splitters divert a percentage of the optical signal from the pair of fibers of the 155 Mbps link to the two ATM cards (one for each direction). The ATM cards run a modified firmware that supports a number of collection modes such as first, first and second, or first, second, and third cell of each packet, all cells without payloads, all cells, etc. Furthermore, the OC3mon is accompanied by a set of flexible, publically available software tools (Coral [3]), which can be adapted and/or extended to particular needs. Among others, the OC3mon can gather statistics based on VCI/VPI at the ATM level or source/destination addresses and port numbers at the IP level. Such flow granularity allows us to measure resource usage for departments, subnets or individual workstations at the IP level, or connections at the ATM level.[4]

Unlike the OC3mon, router-based statistics collection (e.g., NetFlow equipped Cisco routers) has an effect on the router's packet forwarding speed. Furthermore, router-based monitoring tends to be inflexible and cannot support the continuous collection of detailed measurements, which are required to capture the burstiness of traffic.

The main analysis and storage of traffic measurements is performed by a high performance Unix workstation (*Traffic Analysis Station*), using the tools that we describe in the next section. The Unix operating system was selected to facilitate remote access of the analysis tools and traces.

4 Traffic analysis tools

Currently our analysis is performed by two primary tools: the first, called `msa`, implements the analysis based on the effective bandwidths and the many sources asymptotic (discussed in Appendix A), and the second, called `lb`, computes the token (or leaky) bucket parameters. Both programs take input from a trace file which contains the load (number of bytes or cells) in consecutive intervals (epochs) of fixed duration. A sample trace file is shown below:

```
# epoch_in_msecs = 40
# bits_per_info_unit = 424
65
4
5
8
...
```

The first two lines indicate the duration of the epoch and the size (in bits) of the units of load, respectively. Hence, according to the above trace the stream produced 65 cells (since one cell contains 424 bits) in the first epoch ($0 - 40$ msec), 4 cells in the second epoch ($40 - 80$ msec), etc.

Both tools, along with a detailed manual and other related information, have been made publically available at [10]. Furthermore, we are currently developing a web based interface that enables the use of the tools over the Internet[5].

In addition to the two primary tools, we have written other supportive programs (e.g., for traffic smoothing) and scripts for executing the same function for a range of link parameters. The scripts enable us, for example, to run the function for computing the maximum load for a range of buffer sizes, thus producing the data for plotting the maximum load as a function of buffer size (e.g., see Section 5.2).

In the next two subsections we describe the functionality and methods of the `msa` and `lb` tools.

[4] See [11, 2] for a presentation and discussion of traffic measurements collected by the OC3mon.

[5] A prototype will be available at `http://www.ics.forth.gr/netgroup/msa/interface/`

4.1 The `msa` tool

The `msa` tool is based on a recent theory of multiplexing a large number of bursty traffic streams, while guaranteeing some level of performance [7, 6, 4]. This work considers the effective bandwidth of a stream as a measure of resource usage of that stream. The effective bandwidth of a stream depends on the context of the stream (see the related discussion of Question 1 in Section 2) through only two parameters s, t. These parameters can be taken to characterize the operating point of a link. Furthermore, periods of the day during which the traffic mix remains relatively constant can be characterized by a single pair, which can be computed off-line from traffic measurements [5]. A detailed description of the methods implemented by the `msa` tool is contained in Appendix A.

The `msa` tool has the following four functions:

1. Calculates the effective bandwidth (measure of resource usage) for a given link operating point (expressed through parameters s, t) and traffic mix (source types and percentage of each type).
2. Calculates the buffer overflow probability (BOP) for a given link capacity, buffer, number of sources, and traffic mix.
3. Calculates the maximum load (equivalently, the maximum number of sources) for a link of given capacity, buffer, and traffic mix, while ensuring a maximum buffer overflow probability.
4. Calculates the minimum buffer size for a link of given capacity, number of sources, and traffic mix, such that a maximum buffer overflow probability is satisfied.

By running the above functions for a range of link parameters we can compute (i) the buffer overflow probability (BOP) as a function of buffer size (for fixed capacity, traffic mix and load), (ii) the maximum load as a function of buffer size (for fixed capacity, overflow probability, and traffic mix), see Section 5.2, and (iii) the acceptance region (for fixed capacity, buffer, and overflow probability), see Section 5.3.

4.2 The `lb` tool

The `lb` tool computes the token (or leaky) bucket parameters for a particular trace file when all traffic is to be conforming or when some percentage of the traffic is allowed to be non-conforming. A token bucket (see Figure 3) consists of a token pool of size b (bucket size) which fills at rate r (token rate) measured in tokens per second.[6] The token bucket is used to police a traffic stream in the following way: While the token pool is non-empty, one token is removed from the token pool for each conforming cell. On the other hand, if the token pool is empty when a cell arrives, then the cell is non-conforming.

For a particular trace, there is not a single pair of token bucket parameters, but a set of such pairs which form an *indifference curve*, Figure 4. The indifference curve of a stream is convex, it intersects the rate axis (horizontal axis

[6] Our description refers to a network with fixed packet size (such as ATM). It's use in a network with variable size packets differs in that the token rate is measured in bytes per second, and for each conforming packet the bucket is emptied by an amount equal to the size (number of bytes) of the conforming packet.

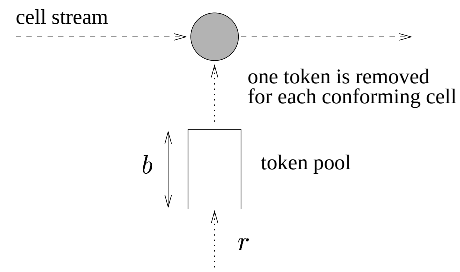

Figure 3: Token bucket policing. A cell is conforming if when it arrives the token pool is not empty (and one token is removed from the pool), else it is non-conforming.

in Figure 4) at the peak rate p of the stream, and it increases abruptly as the token rate approaches the mean rate m of the stream.

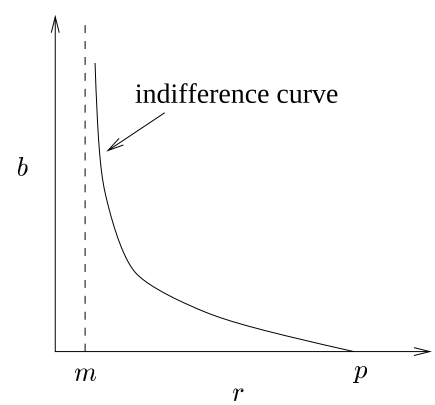

Figure 4: The indifference curve is convex, intersects the rate (horizontal) axis at the stream's peak rate p, and increases abruptly as the token rate approaches the stream's mean rate m.

The indifference curve of a stream can be numerically computed in the following way: For values of r ranging from the mean rate to the peak rate, we emulate a single queue with infinite buffer being fed with traffic from the trace file, and find the maximum backlog b such that all or a minimum percentage of the traffic is conforming. With the above procedure we obtain the pairs (r, b) that belong to the trace's indifference curve.

If all traffic is to be conforming, then the maximum backlog b can be found in a single pass of the trace file. However, when some percentage of the traffic is allowed to be non-conforming, then for each value of r we must search for the minimum value of b such that the percentage of backlogged traffic above b is equal to the percentage of non-conforming traffic. Calculating this percentage requires, for each value of b searched, one pass of the trace file. Because the percentage of non-conforming traffic is monotone with b, the minimum b can be found using a binary search.

5 Examples of traffic analysis

In this section we present some case studies demonstrating the application of our analysis tools to answer some of the questions posed in Section 2.[7] In particular, we present experimental results investigating the following issues:

- Dependence of a stream's resource usage on the context of the stream (in particular, the buffer size).
- Effect of link resources (in particular, the buffer size) and traffic shaping on the multiplexing capability of a link.
- Acceptance region for the case of two traffic types.
- Computation of token bucket parameters, and effect of traffic shaping.

The traces[8] used in the experiments are the Bellcore Ethernet WAN trace [8], and the Lawrence Berkeley Laboratory (LBL) TCP WAN trace [9].

5.1 Dependence of resource usage on the buffer size

Table 1 shows the effective bandwidth (resource usage) when a stream is multiplexed in links with different buffer sizes, demonstrating that resource usage depends on the context of a stream (the buffer size in our particular case).

Buffer size (x10^3 Bytes)	Effective bandwidth (Kbps)
17	8.5
43	7.5
100	7.0
200	6.9

Table 1: Resource usage depends on the context of the stream (buffer size in the table). $C = 34$ Mbps, $BOP \leq 10^{-6}$, Bellcore trace.

5.2 Maximum link utilization and effect of traffic shaping

Figure 5 shows the effect of traffic shaping on the maximum link utilization, when a maximum overflow probability is to be guaranteed. Shaping is performed by evenly spacing the traffic in consecutive time intervals of length d (shaping delay). The figure shows that the effect of traffic shaping on the multiplexing capability of a link decreases for larger buffer sizes.

Figure 5 also shows that the same increase of the buffer size does not produce the same increase of the maximum utilization. In particular, an increase of the buffer has a larger effect for smaller buffer sizes.

[7] Since the measurement platform is not yet deployed, our case studies will necessarily involve traffic measurements taken from other locations.

[8] Available from the Internet Traffic Archive at `http://www.acm.org/sigcomm/ITA/`

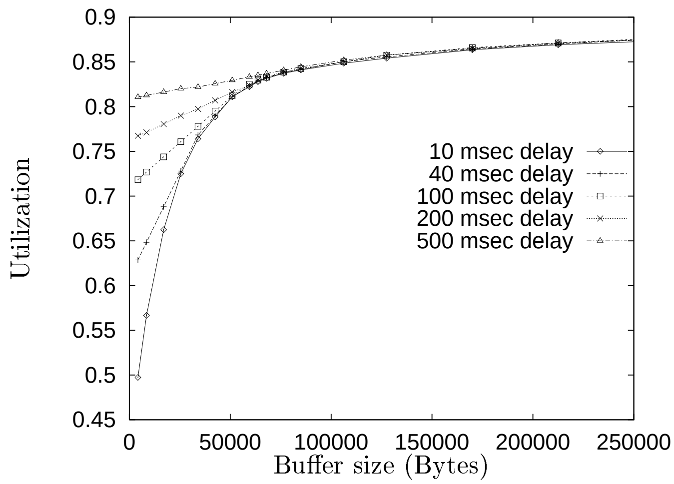

Figure 5: Maximum utilization as a function of buffer size for various shaping delays. Observe that the effect of traffic shaping on the multiplexing capability of a link decreases as the buffer size increases. $C = 34$ Mbps, $BOP \leq 10^{-6}$, Bellcore trace.

5.3 Acceptance region

Figure 6 shows the acceptance region for the case of two source types. Observe that the boundary of the acceptance region, although close, is not exactly linear. This indicates that the effective bandwidth, which is a relative measure of the amount of resources used by each traffic type, depends on the traffic mix. An important issue is the shape of the boundary for other traffic types, and if and how well it can be approximated by a single line.

5.4 Indifference curve and effect of traffic shaping

Figure 7 shows the indifference curve for various shaping delays. Observe that traffic shaping affects mostly the lower-right portion of the indifference curve.

6 Concluding remarks

We have presented a platform for measurement and analysis of real network traffic. The measurements are performed by a low-cost PC-based monitor (OC3mon), which can collect detailed traffic statistics on a high speed (155 Mbps) link, without disrupting the operation or affecting the performance of the network. This platform is currently being deployed at the University of Crete network. The traffic analysis tools involve a number of programs targeted at answering important questions concerning the management and dimensioning of networks carrying bursty traffic and guaranteeing some level of performance. A subset of the analysis tools, along with other information regarding the underlying theory, are available at [10].

In addition to further analysis of network traffic, ongoing research and development activities related to the above work are proceeding in a number of

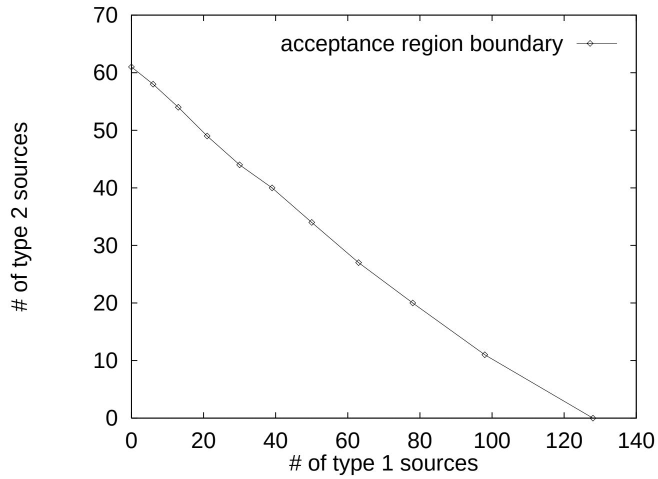

Figure 6: Acceptance region for two source types. Observe that the boundary of the acceptance region is not linear, indicating that the effective bandwidth, which is a relative measure of the amount of resources used by each traffic type, depends on the traffic mix. $C = 34$ Mbps, $B = 42500$ Bytes, $BOP \leq 10^{-6}$. Source type 1 consists of 32 independent flows carrying Bellcore traffic, and source type 2 consists of a single flow carrying LBL traffic.

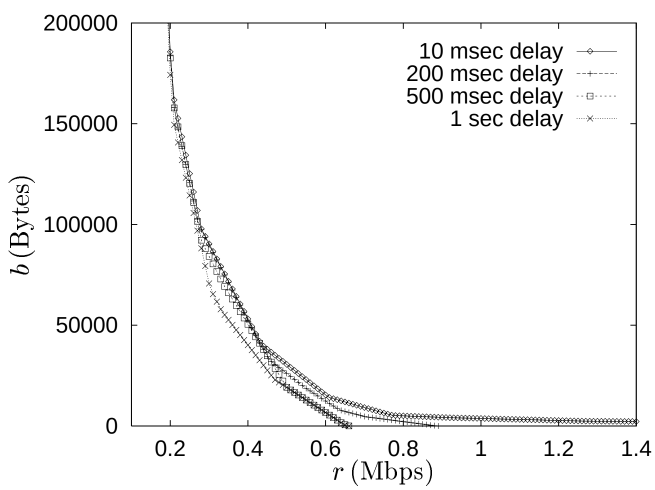

Figure 7: Indifference curve for various shaping delays. Observe that traffic shaping affects mostly the lower-right portion of the indifference curve. Bellcore trace.

directions. One direction is the development of a flexible interactive environment, accessible from the web, that enables researchers to access and experiment with our analysis tools. Such an environment, enhanced with additional methods, can serve both as a teaching tool for hand-on experience with traffic engineering methods, and as a tool to assist managers in operating and dimensioning their networks more efficiently. Finally, another direction involves using the platform for experimenting in areas such as charging for network services.

References

[1] J. Apisdorf, K. Claffy, K. Thompson, and R. Wilder. OC3MON: Flexible, affordable, high-performance statistics collection. In *Proc. of INET'97*, 1997. Available at `http://www.nlanr.net/NA/Oc3mon/`.

[2] K. Claffy, G. Miller, and K. Thompson. The nature of the beast: recent traffic measurements from an Internet backbone. In *Proc. of INET'98*, 1998. Available at `http://www.caida.org/Papers/Inet98/index.html`.

[3] Coral. Coral - flexible, affordable, high performance network statistics collection. `http://www.caida.org/Tools/Coral/`.

[4] C. Courcoubetis, F. P. Kelly, and R. Weber. Measurement-based usage charges in communications networks. Technical Report 1997-19, Statistical Laboratory, University of Cambridge, 1997. To appear in Operations Research.

[5] C. Courcoubetis, V. A. Siris, and G. D. Stamoulis. Application of the many sources asymptotic and effective bandwidths to traffic engineering. *Telecommunication Systems, to appear*, 1999.

[6] C. Courcoubetis and R. Weber. Buffer overflow asymptotics for a switch handling many traffic sources. *J. Appl. Prob.*, 33:886–903, 1996.

[7] F. P. Kelly. Notes on effective bandwidths. In F. P. Kelly, S. Zachary, and I. Zeidins, editors, *Stochastic Networks: Theory and Applications*, pages 141–168. Oxford University Press, 1996.

[8] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson. On the self-similar nature of ethernet traffic. In *Proc. of ACM SIGCOMM'93*, pages 183–193, Ithaca, NY, USA, September 1993.

[9] V. Paxson and S. Floyd. Wide-area traffic: The failure of Poisson modeling. *IEEE/ACM Trans. on Networking*, 3(3):226–244, June 1995.

[10] V. A. Siris. Large deviation techniques for traffic engineering. `http://www.ics.forth.gr/netgroup/msa/`.

[11] K. Thomson, G. J. Miller, and R. Wilder. Wide-area Internet traffic patterns and characteristics. *IEEE Network*, November/December 1997.

A Effective bandwidths and many sources asymptotic: theory and application

Next we present the theory and application of the effective bandwidths and the many sources asymptotic [7, 6, 4, 5]. The effective bandwidth of a stream of type j that produces load $X_j[0,t]$ in a time interval t is defined as

$$\alpha_j(s,t) = \frac{1}{st} \log E\left[e^{sX_j[0,t]}\right], \tag{1}$$

where the parameters s and t characterize a link's operating point and depend on the context of the source, i.e., the link resources and the characteristics of the multiplexed traffic. Specifically, the space parameter s (measured in, e.g., Mbit^{-1}) indicates the degree of multiplexing and depends, among others, on the size of the peak rate of the multiplexed streams relative to the link capacity: For links with capacity much larger than the peak rate of the multiplexed streams, s tends to zero and the effective bandwidth approaches the mean rate, while for links with capacity not much larger than the peak of the streams, s is large and the effective bandwidth approaches the peak rate $X[0,t]/t$, measured over an interval of duration t. On the other hand, the time parameter t (measured in, e.g., seconds) corresponds to the most probable duration of the buffer busy period prior to overflow.

Computation of the buffer overflow probability (BOP)

If N streams are multiplexed in a link with capacity C and buffer B, and ρ_j is the percentage of streams of type j, then the parameters s, t can be computed from the following formula

$$NI = \inf_t \sup_s F(s,t), \tag{2}$$

where

$$F(s,t) = s(B + Ct) - stN \sum_j \rho_j \alpha_j(s,t).$$

The buffer overflow probability (BOP) can then be approximated by

$$BOP \approx e^{-NI}. \tag{3}$$

The many sources asymptotic can be improved using the Bahadur-Rao theorem. With the Bahadur-Rao improvement the buffer overflow probability is approximated by

$$BOP \approx e^{-NI - \frac{1}{2}\log(4\pi NI)},$$

where NI is computed as before using (2).

Let T be the total duration of the trace. The expectation in (1) can be approximated by the empirical average, hence the effective bandwidth can be approximated by (for simplicity we assume that T is integer multiples of t)

$$\tilde{\alpha}_j(s,t) = \frac{1}{st} \log\left[\frac{1}{T/t} \sum_{i=1}^{T/t} e^{sX_j[(i-1)t, it]}\right].$$

The solution of (2) involves two optimization procedures: the first consists of finding, for a fixed value t, the maximum $F^*(t) = \max_s F(s,t)$, and the second consists of finding the minimum $NI = \min_t F^*(t)$.

The maximization $F^*(t) = \max_s F(s,t)$ can be numerically solved in an efficient manner by taking into account that the logarithmic moment generating function $st\alpha_j(s,t) = \log E[e^{sX_j[0,t]}]$ is convex in s, whereas $s(B + Ct)$ is linear in s . Due to this, $F_t(s) = F(s,t)$ is a unimodal function of s and the maximizer is unique. Hence, to find $F^*(t) = \max_s F(s,t) = \max_s F_t(s)$ one can start from an initial "uncertainty" interval $[s_a, s_b]$ that contains the maximum (for this to be the case it is sufficient that for some $x \in [s_a, s_b]$ we have $F_t(x) > F_t(s_a)$ and $F_t(x) > F_t(s_b)$), and decrease the uncertainty interval using a *golden section* search as follows:

1. Given the interval $[s_a, s_b]$, two trial points s_l, s_r are selected such that $s_r - s_a = s_b - s_l = g(s_b - s_a)$, where $g = 0.618$ is (approximately) the *golden ratio*.

2. We evaluate $F_t(s_l)$ and $F_t(s_r)$, and identify three cases:

   (a) if $F_t(s_l) > F_t(s_r)$ the interval becomes $[s_a, s_r]$,

   (b) if $F_t(s_l) < F_t(s_r)$ the interval becomes $[s_l, s_b]$, and

   (c) if $F_t(s_l) = F_t(s_r)$ the interval becomes $[s_l, s_r]$.

3. Steps 1 and 2 are repeated until the uncertainty interval has length less than some small value ϵ.

The golden section search is the limit (for a large number of steps) of the *Fibonacci search*, which minimizes the maximum number of steps needed to reduce the uncertainty interval to some prescribed length.

Unlike the maximization procedure for $\max_s F(s,t)$, there is no general property for $F^*(t)$ that we can take advantage of in order to perform the minimization $\min_t F^*(t)$ in an efficient manner.[9] For this reason, the minimization is solved by linearly searching the values of t in the interval $[0, \kappa\tau]$ with granularity equal to one epoch τ. The value of κ is determined empirically and depends on the buffer size: the extremizing value of t is larger for larger buffer sizes.

The above describe the numerical solution of the `infsup` formula (2) which, using (3), implements the second functionality of the `msa` tool described on page 7.

To select the time granularity of traffic measurements (i.e., the length of the interval for measuring the number of bytes or cells) so that all statistics that are important for buffer overflow are captured, one proceeds as follows. The minimizing value of t in (2) is computed as described above. Obtaining the value $t = \tau$ (the duration of one epoch) indicates that buffer overflow occurs on time scales less than (or equal) to τ. In this case the load measurements are too coarse, and the measurement interval must be decreased so that the minimizing value of t is a few times τ.

[9]Furthermore, experiments have shown that $F^*(t)$ can have more than one local maxima.

Computation of the maximum number of sources

Now we describe the implementation of the third functionality of the `msa` tool described on page 7. The maximum utilization, or equivalently the maximum number of streams, for a given amount of link resources (capacity C and buffer B), traffic mix $\vec{\rho} = \{\rho_1, \rho_2, \ldots\}$, and target overflow probability $e^{-\gamma}$ can be computed by solving the following equation

$$N = \inf_t \sup_s G(s,t)\,, \tag{4}$$

where

$$G(s,t) = \frac{s(B+Ct)-\gamma}{st\sum_j \rho_j \alpha_j(s,t)}\,.$$

As was the case for $F(s,t)$ in (2), $G(s,t)$ is also a unimodal function of s with a unique maximizer, hence $G^*(t) = \max_s G(s,t)$ can be solved using a golden section search.

To use the Bahadur-Rao improvement, the maximum number of multiplexed sources is computed using (4) after replacing γ with $\gamma_{\text{B-R}} = \gamma - \frac{1/2\log(4\pi\gamma)}{1+1/(2\gamma)}$.

Computation of the minimum buffer size

The fourth functionality of the `msa` tool, i.e., the computation of the minimum buffer size such that a maximum buffer overflow probability is guaranteed can be computed by solving the following equation:

$$N = \sup_t \inf_s K(s,t)\,, \tag{5}$$

where

$$K(s,t) = \frac{stN\sum_j \rho_j \alpha_j + \gamma}{s} - Ct.$$

As was the case for $F(s,t)$ in (2), $G(s,t)$ is also a unimodal function of s with a unique minimizer, hence $K^*(t) = \min_s K(s,t)$ can be solved using a golden section search.

To use the Bahadur-Rao improvement, in (5) we replace γ with $\gamma_{\text{B-R}} = \gamma - \frac{1/2\log(4\pi\gamma)}{1+1/(2\gamma)}$.

B Usage of msa

The following is taken from the manual page for msa [10].

```
NAME
msa  -  Numerically  solves  the  supinf  formula and computes the
effective bandwidth, for input from trace files.

SYNOPSIS
(0) msa [-help]
(1) msa data -eb s t
(2) msa data -bop B C N [-t t_start t_final t_step] [-log logfile]
(3) msa data -load | -load_br B C g [-t t_start t_final t_step]
             [-log logfile]
(4) msa data -B | -B_br N C g [-t t_start t_final t_step]
             [-log logfile]

where data = dataset |
             -two dataset1 dataset2 perc2 |
             -two dataset1 dataset2 -N2 n2 |
             -three dataset1 dataset2 dataset3 perc2 perc3 |

DESCRIPTION
Numerically  solves  the  supinf  formula  of  the  many sources
asymptotic  [CW95],  through  which  the  s,t  parameters and BOP
(Buffer  Overflow  Probability)  are computed, and calculates the
related effective bandwidth definition [Kel96], for traffic from a
trace  file.  It  has four basic functions: (1) computes effective
bandwidth  for specific s,t, and traffic mix, (2) computes BOP and
s, t for specific capacity, buffer, number of sources, and traffic
mix,  (3)  computes maximum number of sources and s,t for specific
capacity,  buffer,  target  BOP, and traffic mix, and (4) computes
minimum  buffer  and s,t for specific capacity, number of sources,
target BOP, and traffic mix.

The  traffic  mix  can contain one, two, or three different types,
each described from a trace file. In the case of two or three data
types,  the  percentage  of the second or second and third traffic
type,  respectively,  is  given.  Furthermore, for the case of two
traffic  types, we can fix the number of source of the second type
('-two dataset1 dataset2 -N2 n2' option above).

(1) msa data -eb s t
Computes  the  mean,  peak,  standard  deviation, and eb(s,t) of a
single  traffic  stream for particular values of s (real) in kb^-1
and  t  (integer)  in epochs. The stream can be a "virtual" stream
created from two or three other streams.

EXAMPLE
The  following  command calculates the effective bandwidth for the
star2.tr  file,  and  for  s=.01135 kb^-1 and t=0.16 sec =4 epochs
(since for the star2.tr file 1 epoch=40 msec).

msa star2 -eb .01135 4
```

```
and gives effective bandwidth = 0.30815 Mbps

(2) msa data -bop B C N [-t t_start t_final t_step] [-log logfile]
Numerically solves the supinf formula (computes s, t, and BOP) for
given  buffer,  capacity, number of sources, and traffic mix. B is
in info  units  of  datafile,  C in Mbps, N number of streams, t_*
in epochs.

EXAMPLE
msa star2 -bop 351 149 400

Running the above command gives:
------------------------------------------------------------
B=351 cells (=1.0 msec), C=149 Mbps
N=400, Load=0.70
BOP(many sources)=10^-7.64
BOP(many sources + b-r)=10^-8.86
BOP(many sources + b-r_approx)=10^-8.81
s=0.016532 kb^-1, t=1 epochs (=40.000 msec)
effective bandwidth=0.3153 Mbps
------------------------------------------------------------

(3) msa data -load | -load_br B C g [-t t_start t_final t_step]
        [-log logfile]
Finds  maximum  load such that BOP <= 10^-g. B is in info units of
datafile, C in Mbps, t_* in epochs, and traffic is determined from
data.  To  use  the  many  source  asymptotic with the Bahadur-Rao
improvement, use -load_br instead of -load.

EXAMPLE
msa star2 -load 1405 149 7

Running the above command gives:
------------------------------------------------------------
B=1405 cells (=4.0 msec), C=149 Mbps
N=443, Load=0.78, BOP(many sources)=10^-7.01
s=0.015291 kb^-1, t=1 epochs (=40.000 msec)
effective bandwidth=0.3104 Mbps
------------------------------------------------------------
where  N=443  is  the  maximum  number  of  sources  that  can  be
multiplexed  such  that  the  target overflow probability 10^-7 is
satisfied. The load (=N*mean_rate/C) is 0.78.

(4) msa data -B | -B_br N C g [-t t_start t_final t_step]
        [-log logfile]

Finds  the  minimum  buffer for given capacity, number of sources,
and  traffic  mix  such  that  BOP <= 10^-g. B is in info units of
datafile, C in Mbps, t_* in epochs, and traffic is determined from
data.  To  use  the  many  sources asymptotic with the Bahadur-Rao
improvement, use -B_br instead of -B.
```

C Usage of lb

The following is taken from the manual page for lb [10].

```
NAME
lb  -  computes  the  set  of  token  (or leaky) bucket parameters
of a traffic stream.

SYNOPSIS
lb [-help]
lb data options

where options = -d | -p delta

DESCRIPTION
Computes   token   bucket   parameters   for   a   traffic  stream
in  the case all traffic is conforming (-d option) and in the case
some percentage of the traffic is non-conforming (-p option).

EXAMPLES
lb star2 -d

Computes  token  bucket  parameters for data in 'star2.tr' for the
case  all  traffic  is conforming. The leaky bucket parameters are
stored  in  the  file 'lb_star2_d'. The contents of this file look
like the following:

0 3.46
2 3.44
3 3.42
...
87324 0.28

where  the  first column is the value of the bucket size b in info
units   of  the  star2.tr  file  and  the  second  column  is  the
corresponding    value    of   the   token   rate   r   in   Mbps.

lb star2 -p 4

Computes  token  bucket  parameters for data in 'star2.tr' for the
case  less  than 10^-4 of the traffic is non-conforming. The token
bucket parameters are stored in the file 'lb_star2_p_4'.
```