A-PRIORI FLOW BANDWIDTH ESTIMATES FOR DYNAMIC BANDWIDTH ALLOCATION IN ISP ACCESS LINKS

J. Aracil and D. Morató
Public University of Navarra
Automatics and Computer Science Department
Campus Arrosadía s/n, 31006 Pamplona, Spain
email: {javier.aracil,daniel.morato}@unavarra.es
Phone: +34 948 169733. Fax: +34 948 168924

ABSTRACT

In this paper we study a-priori bandwidth estimation algorithms for TCP flows. An RTT-based bandwidth allocator is proposed, which outperforms a broad class of peak-rate and static allocation flow switching solutions. Our findings suggest that a-priori bandwidth estimation (i.e, before the TCP data transfer phase takes place) is indeed feasible and serves to design simple, yet efficient, dynamic bandwidth allocation rules for ISP access links.

1. Introduction and problem statement

Both the Internet access and backbone are suffering a continuous upgrade process in order to provide users with higher speeds. While in the access network ADSL or HFC technologies are improving access rates dramatically a parallel effort is being done in the backbone, which is incorporating optical technologies in the range of Gbps. Bridging the gap between access and backbone Internet Service Providers (ISPs) provide residential users with Internet connectivity through an *access link*, as shown in figure 1.

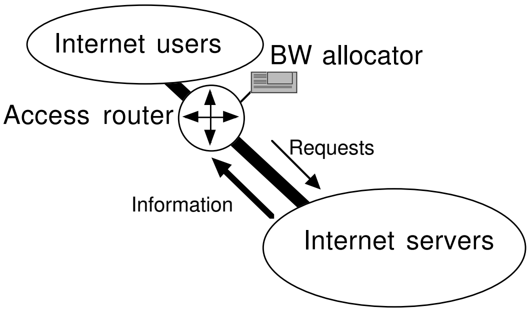

Figure 1: ISP network setup

An access router serves as the internetworking unit between the access subnetwork and the Internet backbone, providing address translation functions and others such as policing and accounting functions in order to shape user traffic to the access link. In fact, due to the skyrocketing increase in network bandwidth at both access and backbone network the access link bandwidth is a most valuable resource, with strong impact in the ISP revenue. Such access link is either hired to a major operator or, being the operator an ISP itself, hired to a second operator through peering agreements.

Usually, a flat rate scheme is adopted for billing access bandwidth. Namely, regardless of the traffic volume the ISP will be charged for a fixed amount of bandwidth with no time of day variation. Such fixed amount of bandwidth is normally approximated with the aid of network measurements, for example bandwidth usage reported by the SNMP agent at the access router. Since Internet traffic shows an extreme burstiness a constant bandwidth allocation is clearly inefficient. On the other hand, even if a constant bandwidth allocation scheme is adopted it turns out that in most cases the absolute value of such constant bandwidth is determined with heuristics. Such heuristics aim at maximizing statistical multiplexing gain while providing a good quality of service to the end users and are normally based on ISP experience, rather than any other analytical consideration. This is mainly due to the lack of dimensioning rules for the Internet case, whose traffic presents not only self-similar features [1, 2] but also strong non-stationarity [3].

Thus, due to the strong input traffic variability, which complicates matters for off-line link dimensioning, dynamic bandwidth allocation schemes seem better suited to optimize bandwidth usage at the access link. Such dynamic bandwidth allocation schemes can be classified as pro-active or reactive. Both schemes are complementary since the performance of reactive schemes, such as for instance the ATM Available Bit Rate service class, is highly dependent on the input traffic burstiness and propagation delay in the link. A pro-active scheme, which adapts the network resources to a *traffic prediction* may serve to the purpose of reducing uncertainty to the bandwidth allocator, which anticipates resource usage before the actual traffic is produced.

Regarding our specific case study of an access link that is providing a large population of users with connectivity to the Internet we note that coordination between multiple hops is not feasible. Indeed, due to the heterogeneity and large number of possible destinations to which the user may have access, resource allocation on an end-to-end basis is still far from being realized in practice. Therefore, the access router may only perform bandwidth scheduling in the access link, which can be considered as an isolated subsystem with an input traffic generated by a large number of servers in the best-effort Internet. Such Internet servers will primarily produce an input traffic which is composed by TCP flows coming from the WWW [4]. In such scenario, per-flow allocation schemes are pro-active dynamic bandwidth allocation mechanisms that provide bandwidth on demand on a per-flow or flow bundle basis. Once the flow requested bandwidth is known the link capacity is increased in order to meet a certain quality of service. Such per-flow allocation schemes are in accordance with emerging label switching protocols, such as the IETF Multiprotocol Label Switching [5, 6] which considers traffic flows and not packets as the atomic unit to perform bandwidth allotment. On the other hand, a number of router manufacturers are currently developing *layer 4* switching solutions which provide resources on a per-flow basis so that the increasing demand for differentiated QoS in the Internet can be met.

In order to provide such flow-switching solutions a clear understanding of the input flows statistical features is in order. More precisely, per-flow allocation demands an a-priori approximate knowledge of the link resources being consumed by a certain flow. A number of recent studies clearly show that most Internet traffic flows are TCP connections coming from the WWW [7, 8, 4]. The traffic trace recorded from our University IP over ATM access link, which is analyzed in this paper (February 2000) [4], shows that the TCP traffic percentage equals 99% in bytes transmitted, 82.8% of which (96.9% of the total number of TCP connections) are WWW connections, out of a sample of 1,029,350 TCP connections. Such high percentage is also reported in a number of measurements performed in a wide variety of academic and industrial settings [7, 8, 4].

Consequently, we note that in order to realize pro-active bandwidth allocation in the Internet case the bandwidth allocator should be provided with the offered throughput of the incoming TCP connections. More specifically, being r_i the offered throughput of TCP connection i we are seeking for a function f such that

$$r_i = f(\alpha_1, \ldots, \alpha_n) \tag{1}$$

being $\alpha_1, \ldots \alpha_n$ a number of connection parameters which are known by the scheduler *a priori*, namely right after the connection has been established but *before* the actual data transfer takes place. For example, one of such parameters could the be the connection size, provided by the HTTP protocol, or server IP address or client advertised window size. We note that the exact estimation of r_i is clearly infeasible since the access link is connected to the best-effort Internet. Thus, due to the many possible different destinations, each of which possibly showing a different bottleneck link in the path from client to server, there is considerable uncertainty in the achieved throughput for a given TCP connection.

In this paper we address the issue of a-priori estimation of TCP connection throughput (r_i) in an access link serving a generic population of users connected to the Internet. Applying the concept of average mutual information between random variables we note that the simple RTT (Round Trip Time) estimate observed in the initial SYN-SYN handshakes suffices to reduce r_i uncertainty significantly. In much less extent, connection size or connection initiation time-of-day also provide information about the connection attained throughput. Based on such findings we consider the a-priori (conditional) probability density of the throughput r_i conditioned to a certain RTT. A simple static allocator which assigns the conditional probability density mean plus n time the variance for different values of n yields significant bandwidth savings in comparison to a peak rate allocator of window size divided by RTT. Most interestingly, since the bandwidth allocator is based on a set of simple rules inferred from an a-priori RTT distribution our results assess the feasibility of neuro-fuzzy schemes that actually work with no a-priori knowledge by learning the RTT distribution and applying dimensioning rules on an standalone basis, thus allowing for fully automated pro-active bandwidth allocation schemes which adapt to any possible users population and access link topology.

1.1. Network scenario and measurement tool

Our traffic traces are obtained from the ATM Permanent Virtual Circuit (PVC) that links Public University of Navarra to the core router of the Spanish academic network (*RedIris* [1]) in Madrid. The Peak Cell Rate (PCR) of the circuit is limited to 4 Mbps and the transmission rate in the optical fiber is 155 Mbps. We note that the scenario under analysis is a representative example of a number of very common network configurations. For example, the most Spanish Internet Service Providers (ISPs) hire ATM PVC links to the operators in order to provide customers with access to the Internet. On the other hand, measurements are not constrained by a predetermined set of destinations but represent a real example of a very large sample of users accessing random destinations in the Internet.

[1] http://www.rediris.es

Furthermore, we carefully check the utilization factor of the ATM PVC and note that never reaches 50% in intervals of 15 min. during the measurement campaign. This sanity check is performed to ensure that the results represent a general Internet case. Namely, different connections are facing different bottleneck links according to the destination, but the results are not correlated by a potential bottleneck link in the access. Finally, the wealth of data in the trace provides a strong confidence level in the obtained results. Measurements comprise one day worth of data starting Monday 14/02/2000 at midnight, making a total of 1029350 TCP connections (16375793 IP packets).

The remainder of this paper is organized as follows: in section II we analyze possible a-priori connection parameters which affect connection throughput. In section III we present the statistical features of our RTT-estimate and the conditional probability densities of the attained throughput r_i. Section IV is devoted to performance comparison of peak rate versus RTT-based bandwidth scheduler. Finally, section IV presents the conclusions and future work.

2. Estimating connection throughput with a-priori connection parameters

We consider the case of an access link in which clients are accessing random destinations in the Internet and we wish to estimate connection throughput. While clients are connected by the access router to the access link the server is located elsewhere in the Internet. Thus, the ISP bandwidth allocator is a sub-component of the access router. For our particular network configuration we note that the downstream from servers to clients is indeed dominant (in a ratio 9:1 [4]), whereas the upstream is primarily devoted to transport of ACK packets. Thus, we will only consider the link downstream as the bandwidth-constrained segment that requires pro-active bandwidth allocation.

In order to study the influence of a-priori connection parameters in the attained connection throughput a list of candidate parameters is in order. By a-priori connection parameters we refer to those parameters which are known right after the TCP connection establishment phase. We now perform a preliminary qualitative analysis in order to filter out some of the candidate parameters. Let us first consider connection independent parameters such as connection initiation time or link load. Both are related since link load fluctuates depending on time of day. For a lightly loaded link such as the one considered in this paper we believe that the bottleneck link is elsewhere and thus we consider connection initiation time only. Since most users access servers in the US we observe better congestion conditions in the early morning hours, which coincide with late night hours in the US.

On the other hand, connection dependent parameters such as destination and origin IP address, RTT-estimate and connection size also influence connection performance. The origin IP address provides very scarce information regarding connection throughput, since our users are connected to 10/100 Mbps LAN within the University campus. Such LANs are connected through transparent bridges mostly and have similar load conditions resulting in no difference between the different network points of attachment corresponding to the various origin IP addresses. Both IP-address and RTT-estimate provide information about server location, which clearly influences connection performance. However, it becomes impractical to infer server location with destination IP-address for a twofold reason: first, the IP-address does not necessarily relate to the number of hops and congested links the packet has to go through in the path from client to server; secondly, the server "distance" in terms of bandwidth and delay are dependent on instantaneous network congestion, together with dynamic routing procedures in the Internet whereas the IP address is fixed. Finally, since the number of possible IP addresses is very large the practical implementation of a bandwidth scheduler based on destination IP-address is difficult to realize. Thus, the RTT estimate serves better to the purpose of characterizing the path conditions between client and server. On the other hand, connection size also provides information about the dynamic behavior of the TCP connection. While a large connection is more likely to reach steady-state a relatively small connection will never leave the slow-start phase, thus producing less throughput in equal RTT conditions.

Other parameters such as packet loss probability along the path or bottleneck link bandwidth cannot be estimated a-priori, even though they would naturally provide information about connection throughput [9]. Thus, our candidate parameters set is reduced to three elements: size, which can be easily obtained from the HTTP response from the server, connection initiation time-of-day and RTT-estimate.

Regarding our RTT-estimate, figure 2 shows the RTT-estimate in a sample HTTP connection. Such RTT estimate is performed with the initial SYN-SYN handshake. Specifically, we consider the time elapsed from the detection of the SYN from the client to the first segment (ACK to the previous SYN) from the server. We note several advantages of such estimate: first, the server response time has *no contribution at all* in such RTT estimate since the ACK in response to the client initial SYN is sent by the TCP layer. Such server response time, which does not affect the RTT estimate, is rather significant for *cold* HTTP servers [10], i. e. servers being accessed from the client after an idle period. Even though there is also a *cold-route* effect which does affect our RTT estimate we note from [10] that the cold-server effect is significantly more important. Thus, our RTT estimate provides a RTT-interval value with minimal error.

In order to evaluate dependence of throughput versus size, connection initiation time-of-day and RTT-estimate we choose the concept of average mutual information between random variables. Let $H(X)$ denote the en-

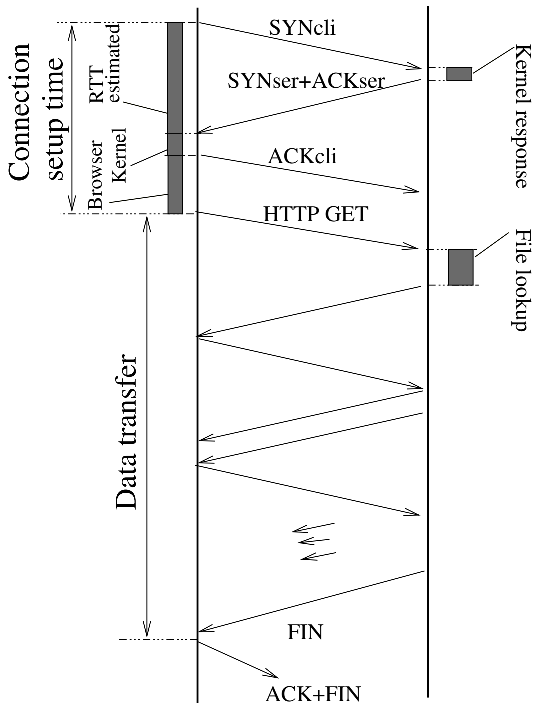

Figure 2: RTT estimate

tropy of the random variable X with possible outcomes $x_1, \ldots, x_n$ with probabilities $p(x_1), \ldots, p(x_n)$ respectively. Such entropy $H(x)$ is given by:

$$H(X) = \sum_{i=1}^{n} p(x_i) log(1/P(x_i)) \quad (2)$$

Let $H(X/Y)$ be the (conditional) entropy of X conditioned to Y with possible outcomes $y_1, \ldots, y_m$ with probabilities $p(y_1), \ldots p(y_m)$ respectively, which is defined by:

$$H(X/Y) = \sum_{i=1}^{n} \sum_{j=1}^{m} P(x_i, y_j) log(1/P(x_i, y_j)) \quad (3)$$

The average mutual information between X and Y is given by:

$$I(X;Y) = H(X) - H(X/Y) \quad (4)$$

If we interpret $H(X/Y)$ as the average amount of uncertainty in X after observing Y and $H(X)$ as the average amount of uncertainty prior to the observation then $I(X;Y)$ is the average information gained per value of X per known value of Y. Thus, the average mutual information provides a simple framework to evaluate dependencies between several random variables. We note that if X and Y are independent then it turns out that $H(X/Y) = H(X)$ and the average mutual information between X and Y equals zero.

In order to evaluate the average mutual information between throughput and RTT-estimate, connection initiation time-of-day and size we take discrete versions of the abovementioned random variables. To this end, we take 1000 bps bins for the throughput, 100 ms. bins for the RTT-estimate, 10 Kbyte bins for the size and hourly intervals for the connection initiation time of day. Table 1 presents the average mutual information and conditional entropy of the throughput r_i:

$H(r_i/RTT)$	$H(r_i/time)$	$H(r_i/size)$
3.986	4.515	4.711
$I(r_i, RTT)$	$I(r_i, time)$	$I(r_i, size)$
0.860	0.331	0.136

Table 1: Information and conditional entropy of r_i

The entropy of the (unconditioned) random variable r_i, $H(r_i)$, equals 4.847 bits. We note that the RTT-estimate has a major contribution in reducing the uncertainty about the throughput r_i, followed by time-of-day connection initiation and finally size. The average mutual information between throughput and time-of-day takes into account 24 hours in the day-worth trace of TCP packets (see section 1.1.). A closer examination of the data reveals that most connections take place during daytime. If we actually restrict our analysis to daytime the information provided by the time-of-day about the throughput decays to 0.129 bits. Since during night time the access link is very lightly loaded it makes no sense to use a bandwidth scheduler for such a link.

Regarding average mutual information between throughput and size we note that large connections do not necessarily achieve higher throughput, even though they are more likely to leave the slow start phase and enter steady-state regime. As a conclusion, we note that the RTT-estimate is the variable that achieves the maximum information about connection throughput.

3. RTT-estimate and attained throughput distributions

In this section we focus on the analysis of the RTT-estimate, as an a-priori variable that serves to reduce uncertainty in the TCP connection attained throughput. We divide the RTTs into 100 ms intervals $(j100, (j+1)100]$ where the index j takes values in the range $0, \ldots, 22$. Let us denote the interval j by RTT_j. We now consider the (conditional) densities of the throughput r_i conditioned to the RTT region RTT_j in figure 3.

Interestingly, figure 3 presents several bell-shaped regions, showing strong dependence of throughput with RTT for small RTT values. For larger RTT values we note that dependence with RTT is not so strong. Large RTT values correspond to congested links in the Internet, possibly with severe loss and high server response times, which influence connection throughput significantly. Let us now denote by μ_j and σ_j the mean and variance of the (conditional) throughput distributions conditioned to RTT_j respectively. In an ideal case, with no losses, no

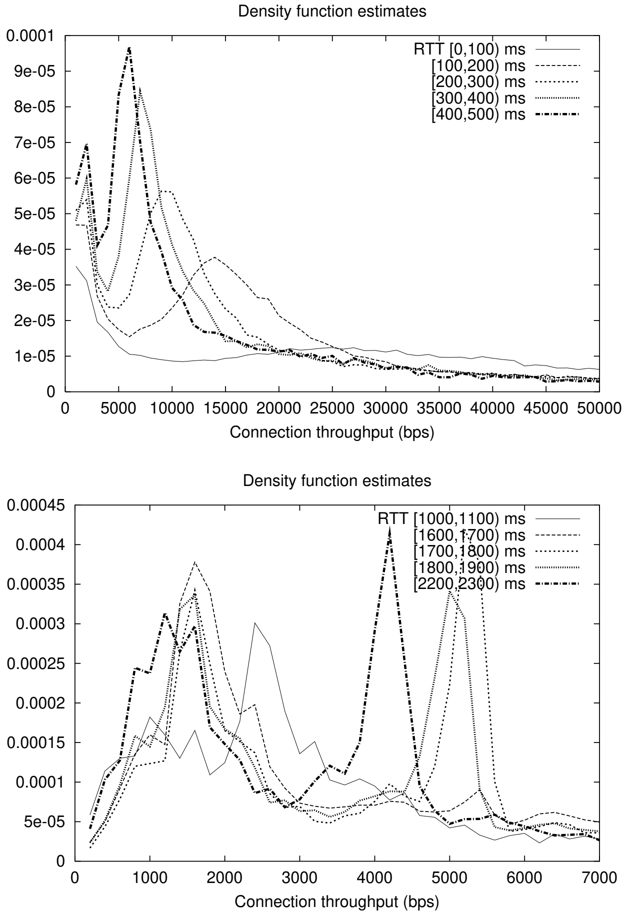

Figure 3: Conditional throughput distribution (bottom: large RTT, top: small RTT)

delay variation and negligible server response time, the network would behave as a deterministic system with a one to one correspondence between throughput and RTT. However, the throughput value which is actually achieved in a real case is a random variable. Having an a-priori characterization of the conditional throughput means and variances, we note that the TCP throughput is amenable for prediction based on the knowledge of the connection RTT, which can be determined *before* the connection data transfer phase starts (see figure 2). Indeed, the next section presents a practical RTT-based TCP bandwidth allocator.

4. Performance evaluation of an RTT-based bandwidth allocator

Flow switching techniques appear as a promising solution to maximize bandwidth efficiency and provide differentiated QoS in the Internet. In this section we present a flow bandwidth allocator that is based on the RTT-estimate presented in the previous section. In order to compare the performance of the proposed allocator we evaluate the bandwidth wastage percentage and unsatisfied bandwidth percentage for three different allocators:

- RTT-based bandwidth allocator: which takes the RTT estimate (RTT_i) as an input and delivers a bandwidth allocation for the TCP flow i, based on the (conditional) probability density of throughput conditioned to RTT.
- Peak rate allocator: which takes client advertised window size for flow i (W_i) and RTT estimate (RTT_i) as an input and delivers a peak rate allocation which is equal to W_i/RTT_i. Note that this is actually the theoretical peak rate which may be achieved by a TCP connection with advertised flow control window W_i.
- Static allocation: We consider the class of static bandwidth allocators which deliver a constant bandwidth per flow.

The RTT-based allocator derives the bandwidth assignment rules from figure 3. More precisely, we divide the RTT into 22 regions corresponding to the intervals $(j100, (j+1)100]$, where j takes values in the range $0, \ldots, 22$ ms, which we name RTT_j and consider the (conditional) probability density $P(r_i/RTT_j)$, which is depicted in figure 3, being μ_j and σ_j the corresponding means and variances. As a first approximation we derive the following bandwidth assignment rule:

For each RTT corresponding to flow i, select the RTT interval RTT_j such that $RTT \in RTT_j$. The bandwidth assignment for flow i equals $r_i = \mu_j + n\sigma_j$ being n a constant

We note that selecting a value for the parameter n is a trade-off between maximizing statistical multiplexing gain and quality of service assigned to the flow. Small values of n will actually provide maximum bandwidth efficiency, at cost of extra buffers. On the other hand, note that the RTT-based bandwidth allocator is a very simple automata, amenable for on-line bandwidth allotment in high-speed links.

In order to evaluate the bandwidth allocator performance we define the following variables: r'_i is the connection attained throughput *a-posteriori*, namely the observed throughput with no bandwidth allocation at all, which is obtained from the traffic trace. Since the access link under analysis is very lightly loaded, thus far from being the bottleneck link, the former variable indicates the maximum achieved throughput in the access link for any particular TCP flow. Let r_i be the bandwidth assigned by the allocator. Note that r_i is determined *a-priori*, i. e. before the connection data transfer takes place. We consider the set of flows i for which $r'_i > r_i$ and define *unsatisfied bandwidth percentage* Δ_i as:

$$\Delta_i = \frac{r'_i - r_i}{r'_i} \tag{5}$$

For those connections with $r'_i \leq r_i$ the variable Δ_i takes the value zero by definition. Note that Δ_i depends on the value of n, the larger the n the smaller the unsatisfied bandwidth percentage for a particular flow. Figure 4 represents the *average* unsatisfied bandwidth percentage versus n, averaging Δ_i over the entire set of flows in the trace.

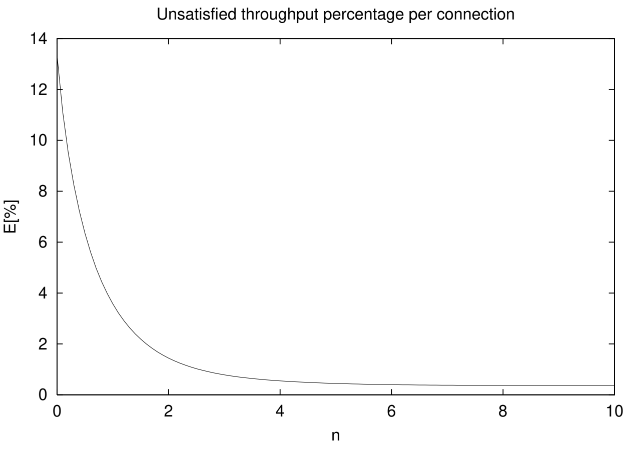

Figure 4: Average unsatisfied throughput percentage

We note that a value of $n = 2$ achieves an average unsatisfied bandwidth in the vicinity of 1.7%. Now, consider the set of flows for which $r_i > r'_i$ and define the *bandwidth wastage percentage* o_i as:

$$o_i = \frac{r_i - r'_i}{r_i} \quad (6)$$

For those connections with $r_i \leq r'_i$ the variable Δ_i takes the value zero by definition. Figure 5 represents the *average* bandwidth wastage percentage versus n. We note that a value of n in the interval $(0, 2)$ provides a wastage between 40% and 73% and a unsatisfied throughput between 13% and 1.7% respectively.

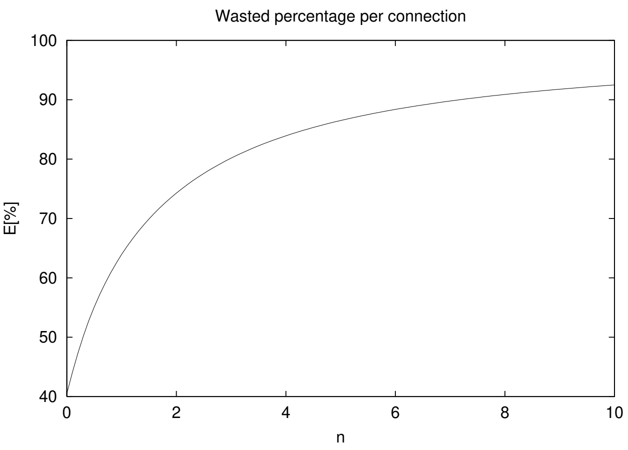

Figure 5: Average bandwidth wastage percentage

Comparing to the peak rate allocator, the average wastage percentage grows up to 94.36%. On the other hand, the unsatisfied bandwidth percentage equals 0.11%. We note that the RTT-based allocator provides a higher degree of flexibility by allowing different values of n that translate into a range of scheduler operating points which can be selected depending on the desired link utilization and quality of service, providing a bandwidth wastage which is below the peak rate assignment. Finally, we consider the class of static allocators (constant bandwidth per flow) which achieve a certain average bandwidth wastage percentage target and compare to the RTT-based allocator. Figure 6 shows the average unsatisfied bandwidth percentage versus the average bandwidth wastage percentage in log-scale in the y-axis. The results show that the RTT-based allocator outperforms any possible static bandwidth allocator, since for the same bandwidth wastage percentage the unsatisfied bandwidth percentage is lower.

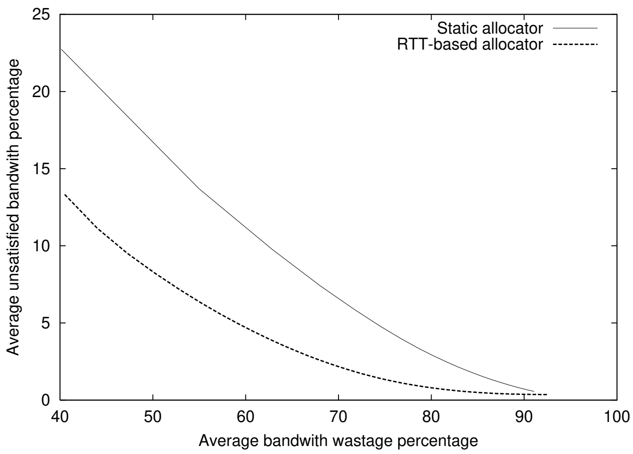

Figure 6: Comparison of RTT-based and static allocation

5. Conclusions and future work

In this paper we show that it is indeed possible to perform a-priori TCP connection bandwidth estimates for dimensioning (bandwidth allocation) purposes. An RTT-based bandwidth allocator is proposed, which outperforms a broad range of peak rate and static bandwidth allocators.

Such RTT-based is built upon a set of simple, yet efficient, bandwidth assignment rules which consider mean and variance of conditional throughput distributions. Such dimensioning rules are achieved by observation of a large traffic trace and set *manually* in the bandwidth scheduler. Thus, the dimensioning rules apply to the link under analysis but not to every possible link. However, having shown that a simple flow bandwidth estimate is feasible we wonder whether automatic derivation of dimensioning rules can also be implemented. In order to do so, and recalling equation 1 (section I) we propose the analysis of other possible estimators of the form:

$$r_i = f(\alpha_1, \ldots, \alpha_n) \tag{7}$$

where f is an estimator function which may be derived on-line and not by prior observation of recorded data as is done in this study. Interestingly, the findings of this paper indicate that there are a number of variables which actually provide information about r_i and pave the way for further analysis of other estimation approaches such as neuro-fuzzy estimators, which are the subject of our present and future research.

References

[1] V. Paxson and S. Floyd. "Wide Area Traffic: The Failure of Poisson Modeling". *IEEE/ACM Transactions on Networking*, 3(3):226–244, June 1995.

[2] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson. "On the Self-Similar Nature of Ethernet Traffic (Extended Version)". *IEEE/ACM Transactions on Networking*, 2(1):1–15, February 1994.

[3] V. Paxson. *Measurement and analysis of end-to-end Internet dynamics*. PhD thesis, University of California, Berkeley, 1997.

[4] J. Aracil, D. Morató, and M. Izal. "Analysis of Internet Services in IP over ATM networks". *IEEE Communications Magazine*, 37(12):92–97, December 1999.

[5] IEEE Communications Magazine, special issue on MPLS, December 1999.

[6] Y. Rekhter, B. Davie, D. Katz, E. Rosen, and G. Swallow. "Tag switching architecture overview". RFC 2105, February 1998.

[7] G. J. Miller K. Thompson and R. Wilder. "Wide-Area Internet Traffic Patterns and Characteristics". *IEEE Network*, pages 10–23, November/December 1997.

[8] M. E. Crovella and A. Bestavros. "Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes". *IEEE/ACM Transactions on Networking*, 5(6):835–846, December 1997.

[9] M. Mathis, J. Semske, J. Mahdavi, and T. Ott. "The macroscopic behavior of TCP congestion aviodance algorithm". *Computer Communication Review*, 27(3), July 1997.

[10] E. Cohen and H. Kaplan. "Prefetching the means for document transfer: A new approach for reducing web latency". In *IEEE INFOCOM 00*, Tel Aviv, Israel, 2000.