A non-invasive method for link upgrade planning using coarse-grained measurements

D.J. Leith, C. Kellett, P. Clifford, J. Epperlein, R.N. Shorten,
Hamilton Institute, NUI Maynooth, Ireland

***Abstract*— A basic problem faced by network operators concerns the provisioning of bandwidth to meet quality of service (QoS) requirements. In the network core, the preferred solution is simply to overprovision link bandwidth. We propose a new approach to making link upgrade decisions based only on readily available coarse SNMP measurements.**

I. Introduction

A fundamental issue in the design and management of wired networks is the selection of the bandwidth to be installed on each link, or *link dimensioning*. In the network core, the method of choice is to over-provision links so that there is sufficient capacity for all requests [8], [7], [3], [5]. While numerous approaches have been studied in the literature to provide QoS, including the development of protocols and queuing strategies (DiffServ, IntServ for example), the case for over-provisioning is compelling. Benefits of over-provisioning include: simplicity, low management overhead, good performance (low latency, loss) and redundancy (spare capacity allows fail-over between over-provisioned links). A drawback to over-provisioning is that resources are then under-utilised.

Link upgrade decisions are typically based on experimentally validated stochastic models of network traffic, and simple rules-of-thumb that are derived from these models. For instance one such 'rule' is to provision networks such that the link capacity should always exceed the "mean five minute demand plus 50%"[8]. However, it is widely accepted that provisioning links in this manner suffers from many drawbacks. In particular, while mean utilisation over some interval is an easily obtained statistic, it can nevertheless be an unreliable one upon which to base a link upgrade trigger. Firstly, the decision to trigger an upgrade is a function of not only mean traffic patterns, but also a function of traffic fluctuations over shorter time scales, and of user QoS requirements. Conventional approaches to capacity planning attempt to use network models parameterised via measurements over long time scales. The models predict the capacity such that QoS requirements can be met over short time scales. Clearly, this requires accurate traffic models, and it is questionable as to whether existing models capture network behaviour sufficiently to allow such an approach to function effectively (due to the stochastic nature of network traffic and network configurations). Given these reservations, an alternative approach is to monitor traffic behaviour over very short time-scales. However, such an approach is not possible using SNMP measurements, and it may be be very costly to develop dedicated probes to measure traffic over very short time scales.

In this paper we seek to infer network traffic fluctuations at the ingress of core routers over short time scales without using traffic models and using only coarse five-minute averages that are available as standard (SNMP) on routers. Our basic idea is to manipulate droptail buffer sizes[1] to actively probe the condition of a link; by adjusting the buffer size to maintain a very small loss rate (e.g., 10^{-3} to 10^{-4}), we can infer qualitative behaviour of traffic fluctuations over short (sub 5 minute) time-scales. We demonstrate that coarse five minute average statistics are more than adequate for this regulation task and that the onset of link under-provisioning corresponds to an increase in buffering on a link beyond a pre-specified threshold. An increasing trend in the buffer size required to maintain a fixed loss rate therefore provides a clear, principled indicator of link under-provisioning.

To our knowledge such a strategy for detecting the onset of link under-provisioning is entirely new and opens up many interesting possibilities. A key feature of the proposed approach is that it avoids the need to introduce dedicated measurement probes and/or perform internal router changes or hardware upgrades, yet provides a method of triggering link upgrades that uses mean traffic load as well as inferred information on the amplitude of traffic bursts. Furthermore, our approach does not require a model of network traffic.

II. Related work

The problem of provisioning link capacity to provide statistical guarantees for network QoS is not new. Much of the recent work on this topic is motivated by the fact that real network traffic cannot be characterised in a simple fashion, a fact that invalidates many provisioning rules based on assumed traffic arrival processes [6]. To the best of our knowledge, the most recent work appears to have been carried out by Fraleigh, Tobagi and Diot [5], by Van de Meent and his co-authors [8], [3] and in the context of the Cisco supported start-up Corvil Networks [1]. The work presented by these authors represents to varying degrees a direct extension of the work first presented by Kleinrock [6]. Fraleigh, Tobagi, and Diot [5] use real measurements of backbone traffic to construct statistical models from which link provisioning rules are derived. Van de Meent [8], [7] uses real measurements to argue that the (typical) assumed relationship between mean

[1] In this work, buffer size refers to the maximum allowable queue size of a droptail buffer and buffer occupancy refers to the number of packets in the queue at any given instant.

and peak rate is not supported by evidence from real measurements, and proposes others which he claims are more realistic. We note that to be effective, both of these approaches require stationarity of network traffic, and accurate statistical models; nonstationarity of traffic or modeling error potentially invalidates the rules proposed by these authors. The work by the Cisco supported start-up Corvil Networks is proprietary in nature and thus difficult to quantify. However, it appears that their approach is based upon fine-grained (millisecond level) traffic measurements obtained using network probes.

III. Non-Invasive Monitoring for Link Provisioning

Our basic idea is a very simple one. Rather than measuring average load directly, we seek to find the buffer size that gives a pre-specified level of packet loss. This gives us information on the peak traffic level; more precisely, how often a certain level is achieved over some interval. By modifying the buffer size, we also track the maximum buffering delays being experienced by individual packets, as well as the maximum jitter being introduced into packet trains by the router under study. Our proposed approach therefore is to regularly adjust the buffer size to be the smallest value that maintains the packet loss rate at a specified level. Note that the target loss rate is chosen to be very small (e.g., 10^{-3} to 10^{-4}) so as to minimise the impact of the algorithm on network end-users.

Detailed Algorithm : We assume that the average loss rate over some time interval τ is measured; e.g. this might be obtained from five minute SNMP data. Let $q(k)$ denote the buffer size over the k'th interval and $\lambda(q(k), k)$ the corresponding measured loss rate.

1. Set $q(0)$ to an initial value. Set $k = 0$
2. Measure $\lambda(q(k), k)$.
3. Compare $\lambda(q(k), k)$ with λ^*. If $\lambda(q(k), k) > \lambda^*$ then adjust the buffer size such that $q(k+1) > q(k)$. If $\lambda(q(k), k) < \lambda^*$ then adjust the buffer size such that $q(k+1) < q(k)$. If $\lambda(q(k), k) = \lambda^*$, set $q(k+1) = q(k)$. Set $k = k + 1$.
4. Goto Step 2.

We stress that there are many methods that can be employed to update the buffer size in Step 3. In this paper we update according to: $q(k+1) = q(k) + Ke$, where e is the difference between the measured and target loss rates and K is a scalar.

Remark: We note that in environments with many traffic sources and a reasonable degree of traffic multiplexing (such as occurs in network core routers), small buffer sizes do not have a detrimental effect on router throughput (see, e.g., [2]).

IV. Simulations

We present three simulation examples generated using the *ns-2* network simulator. The first two examples demonstrate that using coarse-grained measurements of the buffer size, and the proposed algorithm, one can capture both short-term traffic fluctuations (i.e., traffic variance) and increases in traffic intensity (i.e., traffic mean). The third example presents

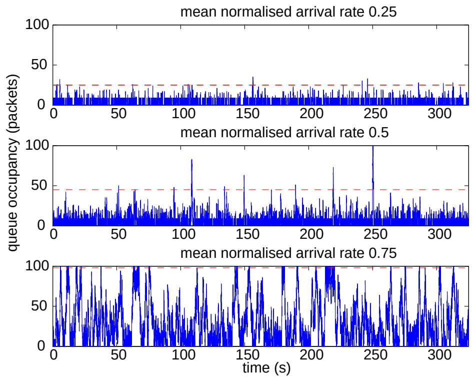

Fig. 1. Illustrating buffering behaviour for exponential arrivals of increasing intensity. Dashed lines mark the buffer size corresponding to a packet loss rate of 10^{-4}.

a synthetic "growing network" to show that the proposed algorithm provides a principled trigger for link upgrades.

***Example 1* - <u>Traffic intensity and buffer size :</u>** We first illustrate that long-term traffic trends are also reflected in buffer behaviour. Figure 1 shows *ns-2* simulations of bursty traffic arriving at a network router for different intensities; i.e., traffic corresponding to exponential arrivals with different mean normalised arrival rates. Figure 1 shows buffer occupancy as well as offline calculated buffer sizes that correspond to a packet loss rate of 10^{-4}. It can be seen that, although the traffic is bursty, a course-grained measurement accurately reflects increasing load.

***Example 2* - <u>Short-term fluctuations and buffer size :</u>** To demonstrate that the approach does indeed capture relevant short time-scale traffic behaviour, we show that the algorithm is able to distinguish between two scenarios with the same mean traffic level, but with very different short-term traffic fluctuations. Our set-up is as follows. We use *ns-2* to simulate a core link of 7Mbits. The traffic arrivals are CBR-Traffic with mean 5Mbits, first with oscillations about the mean of amplitude 2.1Mbits, and halfway through the experiment of 2.3Mbits. The controller uses a logarithmic scaling of the error, and uses a first order filter ($\bar{\lambda}(k) = 0.8\lambda(k) + 0.2\bar{\lambda}(k-1)$) to estimate average loss, as well as the integral controller described above. We set $K = 0.1$ and the target loss rate is 0.001. Note the loss rate is chosen to be relatively high to illustrate the effectiveness of the approach using modest computation. As can be seen in Figure 2, the buffer's behaviour distinguishes these traffic arrivals and identifies the situation corresponding to the traffic with larger short-term fluctuations with a larger buffer size.

***Example 3* - <u>Traffic growth:</u>** We consider a network link that aggregates traffic from n access links (Figure 3). Each access link carries multiple web sessions with a wide range of connection sizes (we use the web traffic generator in *ns-2* with traffic parameters taken from [9]). We simulate growth in network usage by increasing the number of active links over time. While in a real network such growth might occur over

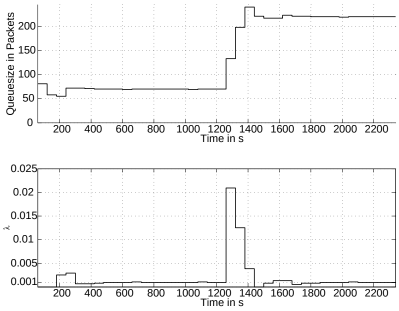

Fig. 2. Buffer size and loss rate evolution for Example 2. Traffic variance changes at $t = 1250s$. λ denotes loss rate.

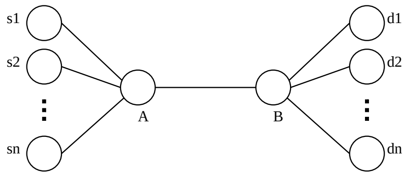

Fig. 3. Network topology: aggregation of n 128Kbs access links $s_1 - s_n$ onto 10Mbs backhaul link AB. Link AB is enabled with algorithm. Each access link carries multiple web sessions generating bursty, on-off traffic.

a period covering many months, we consider an accelerated example to illustrate performance where a new link is added every 30 seconds. We make use of the previously described integral controller with $K = 2.5$ and a target loss rate of 10^{-4}. These values are chosen using standard techniques from control theory and is beyond the scope of the present paper; see [4] for details. The resulting buffer size evolution is shown in Figure 4. The rising trend in the buffer size required to maintain the desired loss rate as new links are activated is clearly evident and could be used to provide a direct trigger for a decision to increase link provisioning.

One important question relates to the sensitivity of the algorithm to the measurement sampling interval. The impact of changing the measurement sampling interval τ is illustrated in Figure 5. It can be seen that decreasing τ (i) increases responsiveness to shorter-term changes in traffic patterns while (ii) also decreasing the accuracy of the measured loss estimate (increasing the level of noise). Increasing τ decreases tracking of short-term changes while improving measurement accuracy. Nonetheless, qualitatively, the algorithm appears to be relatively insensitive to sampling rate.

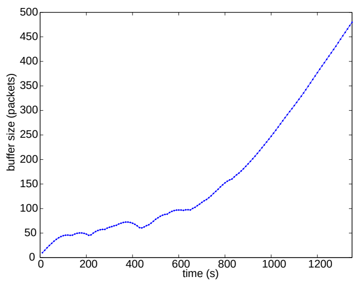

Fig. 4. Buffer size time-history as the number of active access links is increased by 1 every 20s to simulate accelerated demand growth. Algorithm parameters: target packet loss rate $\lambda^* = 10^{-4}$, control gain $K = 2.5$, sampling interval $\tau = 10s$, buffer limits q_{min}=10 packets, q_{max}=500 packets. Network topology as shown in Figure 3.

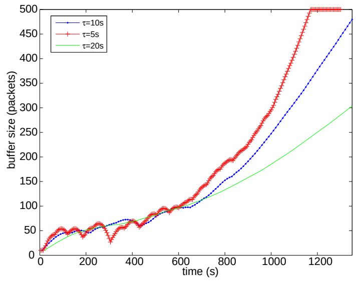

Fig. 5. Impact of choice of sampling interval τ. Algorithm parameters: target packet loss rate $\lambda^* = 10^{-4}$, control gain $K = 2.5$, buffer limits q_{min}=10 packets, q_{max}=500 packets. Network topology as shown in Figure 3.

Remark: The proposed algorithm can be implemented in an entirely non-invasive manner. No router hardware or software changes are required. Loss rate measurements can be obtained from standard SNMP loss count data and adjustment of buffer sizes is widely supported on core routers and can be achieved via remote scripting `ssh` console access. In addition, it is computationally much simpler than model based methods, or methods based on fine grained measurements on a micro-second level. At each coarse grained step, monitoring involves a simple addition only (the control update).

V. Conclusions

In this paper we propose a new approach to making link upgrade decisions. We demonstrate that it is possible to use the buffer itself for fine-grained sensing of the level of link QoS via the introduction of a slow buffer size feedback loop.

References

[1] Corvil networks. http://www.corvil.com.

[2] G. Appenzeller, I. Keslassy, and N. McKeown. Sizing router buffers. In *ACM SIGCOMM 2004*, Portland, Oregon, USA, August 2004.

[3] S. Bessler and P. Reichl. A network provisioning scheme based on decentralized bandwidth auctions. In TELECOMMUNICATIONS PLANNING: Innovations in Pricing, Network Design and Management, Springer, Operations Research/Computer Science Interfaces Series, ISBN 0-387-29222-5, 2006.

[4] J. Epperlein. A control-based approach to core link provisioning, 2007. Studienarbeit.

[5] C. Fraleigh, F. Tobagi, and C. Diot. Provisioning ip backbone networks to support latency sensitive traffic. In Proc. IEEE INFOCOM, 2003.

[6] C. Kleinrock. Queueing systems: Volume 2. John Wiley and Sons, 1976.

[7] M. Mandjes and R. van de Meent. Inferring traffic burstiness by sampling the buffer occupancy. In Proceedings of NETWORKING 2005: 4th International IFIP-TC6 Networking Conference, 2005.

[8] R. van de Meent, A. Pras, M. Mandjes, H. van den Berg, and L. Nieuwenhuis. Trac measurements for link dimensioning. In Proceedings of the 14th IFIP/IEEE Workshop on Distributed Systems: Operations and Management, 2003.

[9] W. Willinger, M. S. Taqqu, R. Sherman, and D. V. Wilson. Self-similarity through high-variability: statistical analysis of Ethernet LAN traffic at the source level. *IEEE/ACM Transactions on Networking*, 5(1):71–86, 1997.