https://dx.doi.org/10.1109/LCN.2011.6115546

Extracting the Independence Signature from Packet Pair Dispersion Data in a Closed-Loop System for End-to-End Bandwidth Probing

Martin Tunnicliffe, Mehri Hosseinpour
Faculty of Computing, Information Systems and Mathematics
Kingston University
Kingston-on-Thames, UK
{M.J.Tunnicliffe, M.Hosseinpour}@kingston.ac.uk

***Abstract* - A receiver-based end-to-end scheme for packet-pair end-to-end probing is explored. Dispersion data collected by an agent process at the receiver end of the monitored path is reported back to the controlling manager process at the transmitter via the return path allowing the inter-packet gaps to be dynamically adjusted for optimum measurement. We concentrate particularly on the "independence signature" monitored through the "long" inter-packet gaps (between successive packet pairs) and compared with the corresponding feature in the "short" inter-packet data. A system of dynamic closed-loop monitoring is devised which correctly identifies the narrow-link bottleneck.**

Keywords–Bandwidth; probing; end-to-end.

I. INTRODUCTION

End-to-end packet-pair dispersion has been studied extensively since it was first introduced in the late 1990's [1]. Although many refinements to this technique have been devised over the years [e.g. 2,3] the basic principle remains the same: Sequences of two or more closely spaced packets are injected at the source and collected at the sink end of a path, and the distribution of the measured packet separation (or dispersion) is used to determine the bottleneck parameters.

However multiple links under cross-traffic introduce additional signatures unrelated to the bottleneck, which require interpretation. Here we explore an end-to-end monitoring scheme by which an agent process at the receiver end measures inter-packet dispersion and returns this information to the manager process at the transmitter. The transmitter in turn adjusts the input gap so as to optimize the measurement.

The remainder of the paper is organized as follows: section II describes the technical background of the work by introducing the independence and rate signatures for an unloaded path, how link parameters are identified from these signatures, the effect of cross traffic and the subsequent emergence of distribution signatures. The new proposed scheme is outlined and tested in section III: the simulation test bed used to evaluate this scheme is described and the mechanism is demonstrated in "open loop" mode. An automated learning mechanism is then applied, to allow the system to adjust to current state of the independence signature. Finally the control loop is closed, showing how dynamic adjustment of the transmitter's parameters based upon information fed back from the receiver can cause the system to home in on an estimate of the narrow link bandwidth. Finally section IV presents general conclusions and suggestions for future work.

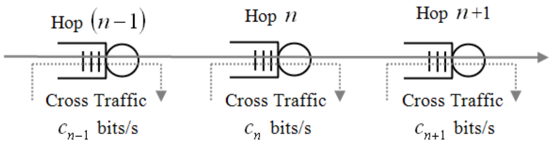

Figure 1. Network path consisting of a tandem chain of independently loaded queuing systems [6].

II. TECHNICAL BACKGROUND

A. *The Independence and Rate Signatures*

In 2002 Pásztor and Veitch [4,5] identified four distinct "signatures" which appear within the packet dispersion data. Their approach was to consider a tandem chain of store-and-forward nodes, each modeled as an independently loaded First-In-First-Out (FIFO) queuing system (Fig. 1). When a pair of probe packets (containing S and S' bits respectively) passes along the chain, their temporal separation before and after each hop are related by the equation:

$$\Delta_{n+1} = \Delta_n + (S - S')/B_n + (\hat{w}'_n - \hat{w}_n) \qquad (1)$$

where B_n is the bandwidth of link *n* and $\hat{w}_n$ and $\hat{w}'_n$ are the waiting times experienced by the two packets. The second term is the "accumulation signature" which can easily be suppressed by making $S = S'$. If there is no cross-traffic passing through the link then $\hat{w}_n = 0$ and $\hat{w}'_n = \max[(S + H_n)/B_n + G_n - \Delta_n, 0]$ where H_n (bits) and G_n (seconds) are the link-layer header and inter-frame gap for hop n. Under these assumptions (1) can be re-written:

$$\Delta_{n+1} = \max[\Delta_n, (S + H_n)/B_n + G_n]. \qquad (2)$$

When $\Delta_n \geq (S + H_n)/B_n + G_n$ the input and output dispersions are equal, giving rise to the so-called "independence signature". Conversely if $\Delta_n \leq (S + H_n)/B_n + G_n$ then Δ_{n+1} saturates at a value which depends not on Δ_n but on the link parameters. This is the "rate signature" from which the bottleneck link parameters may be computed.

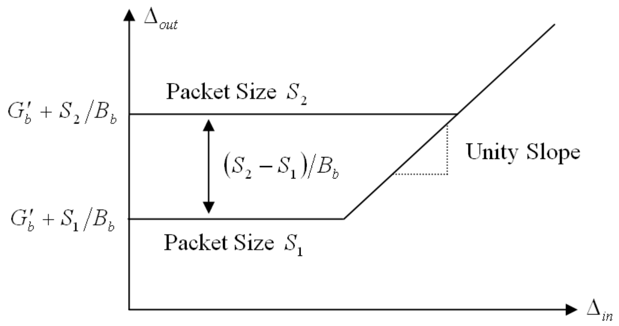

Figure 2. Idealized graph of output vs. input dispersion for two different packet sizes. Without cross-traffic this is dictated by the bottleneck [6].

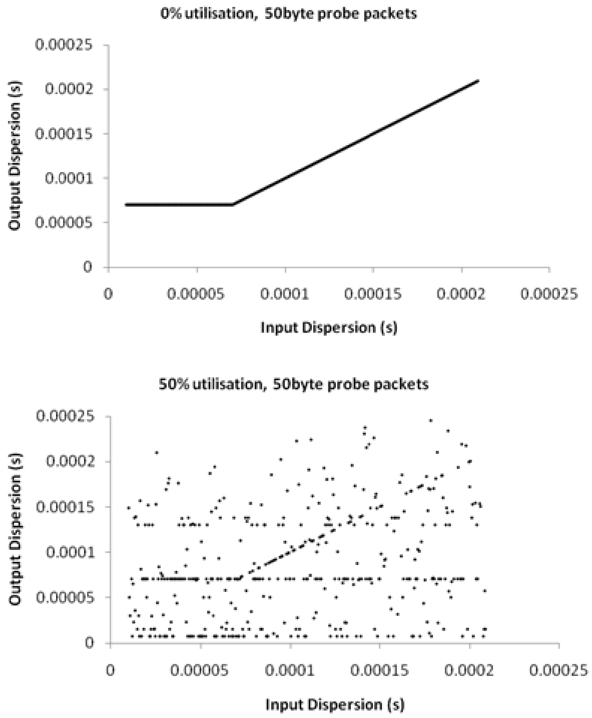

Figure 3. Simulated graphs of output vs. input dispersion for a two-hop network path with and without cross-traffic [6].

B. Identification of Link Parameters

If no cross-traffic interferes anywhere along the path, an input dispersion Δ_{in} seconds gives rise to an output dispersion

$$\Delta_{out} = \max\left(\Delta_{in}, G_b + (S + H_b)/B_b\right) \qquad (3)$$

where G_b, H_b and B_b are respectively the inter-frame gap (seconds), link-layer header (bits) and bit-rate (bits per second) of the bottleneck or "narrow link" of the path. However, since the effects of inter-frame gap and header are externally indistinguishable it is convenient to lump them into a single parameter $G'_b = G_b + H_b/B_b$, the "effective inter-frame gap". Thus $\Delta_{out} = \max\left(\Delta_{in}, G'_b + S/B_b\right)$ and by varying S and Δ_{in} and observing Δ_{out} the values of B_b and G'_b can be determined, as shown in Fig. 2. The maximum effective throughput for a packet size *s* bits then becomes [6]

$$B_{eff}(s) = B_b s / \left(B_b G'_b + s\right). \qquad (4)$$

C. Cross Traffic Signatures

In a loaded network cross-traffic often causes the probe packets to arrive later than expected, making the output dispersion sometimes longer and sometimes shorter (depending on which packet was delayed the most) than (3) predicts. In addition to this, cross-traffic packets can become inserted between successive probe packets, obliterating any idle-time between the two and giving rise to the so-called "distribution signature". Fig. 3 shows some typical results obtained using a simple 2-hop path and randomizing the input gap uniformly between 10 and 200µs. Without cross-traffic (0% utilization) the graph agrees with (3) ($(S + H_n)/B_n + G_n \approx 70$µs), but when cross-traffic is added (50% utilization) a number of other signatures appear: the additional horizontal lines indicate distribution signatures of the two nodes, while the narrow-link's rate signature now continues above the 70µs cut-off due to compression of the inter-packet gap at the upstream node.

In two earlier papers [6-8] we explored various techniques of identifying signatures within this dispersion data, including using the classical Hough Transform [6] to detect probable linear features. In the Hough method each data-point (x, y) is mapped to a function $r(\theta) = x\cos\theta + y\sin\theta$ (where r and θ are the quantities indicated in Fig. 4) representing the set of lines to which it may belong. If $r(\theta)$ is computed across the range $\theta \in (0, \pi)$ then the most frequently visited regions in the (r, θ) plane correspond to the most probable straight lines.

However, in the current scenario this proved something of a detour, since the most prominent lines are either horizontal, giving rise to $r = \Delta_{out}$ for $\theta = 90°$, or else (as in the case of the independence signature) at 45º to the horizontal. In this latter case $r = 0$ for $\theta = 135°$ which is trivially identical to $\Delta_{out} - \Delta_{in} = 0$. In the current paper we use a closed-loop system to detect this independence signature, the onset of which indicates the bottleneck.

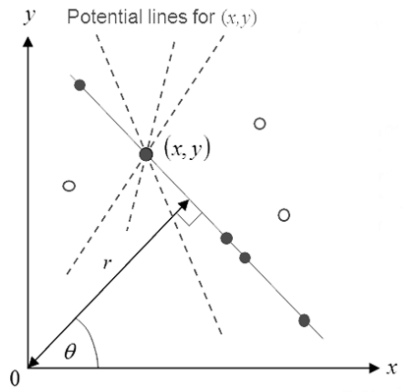

Figure 4. Classical Hough Transform, showing potential straight-lines for a cloud of data-points [6].

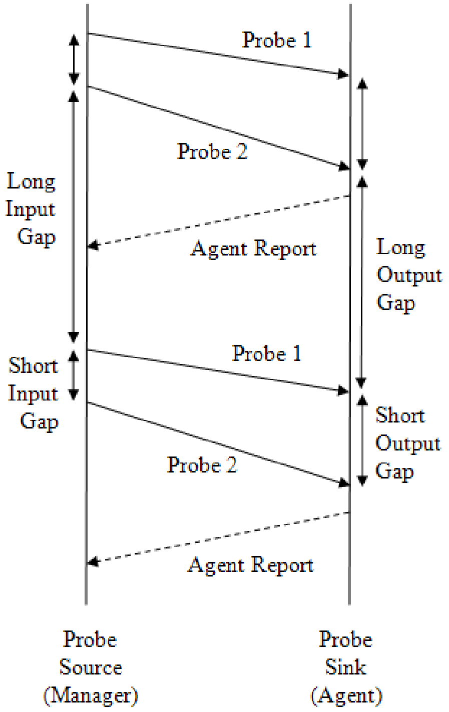

Figure 5. Timing diagram for simulated scheme.

III. The Proposed Scheme

A. *The Basic Architecture*

Fig. 5 shows a timing diagram for the proposed scheme. The manager process at the probe source sends at regular intervals pairs of equal-sized packets to the receiver who notes their temporal separation (output dispersion). Both the "long-gap" (the time between successive pairs) and the "short-gap" (the time between packets within a pair) are reported back to the sender. In a real system this "report back" mechanism could itself form a probe-pair process, allowing the properties of the link to be monitored simultaneously in both directions. However, the current experiment does not have this refinement, which may be investigated in future papers.

B. *The Simulation Test Bed*

Fig. 6 shows the simulated network, which was implemented using the queuing-network classes previously developed in C++ by one of the authors [9]. All components are 100BaseT Fast Ethernet, with the exception of the switch connecting CT Sources 2 and 6 which is 10BaseT. This is the Narrow Link, whose properties dictate the maximum throughput of the path.

Probe packets are all 50 bytes in length and are transmitted 0.5 seconds apart. Cross traffic consists of 60, 148, 500 and 1500 byte packets, which constitute 4.77, 2.18, 9.5 and 82.92% of the aggregate data volume respectively. (This is a very common profile, which we have used in many earlier papers [6-8].) All cross-traffic arrivals are exponentially distributed, though in practice the output buffers of their sources have a traffic-shaping effect which spreads the more closely-spaced packets apart.

The cross-traffic sinks are not modeled in detail; cross-traffic packets are merely deleted once they part company with the probe traffic. Average utilization of all four switches was set to approximately 30%. In addition to the jitter due to queuing, an additional Gaussian jitter was introduced to represent random fluctuations in the clock-speeds at both ends of the link. This was set to an RMS value of 10ns (one tenth of a bit time for 10BaseT) to represent a worst-case scenario.

C. *Identifying Independence*

Figure 7 shows the typical dispersion increases for measurement cycle *j*:

$$y_j^{short} = \Delta_{out(j)}^{short} - \Delta_{in(j)}^{short} \qquad (5)$$

$$y_j^{long} = \Delta_{out(j)}^{long} - \Delta_{in(j)}^{long} \qquad (6)$$

obtained during a linear increase in the short input gap Δ_{in}^{short} between 10 and 180μs. Notice that although the long-gap is sufficient to eliminate causal linkage between the two packet delays, y_j^{long} does not show a zero component when the input gap is below the bottleneck value (around 70μs). This is because the long gap is artificially shortened when the first packet in the previous pair pushes the second packet forwards in time. However, we can cancel this effect by subtracting the previous increase in the short gap, i.e.

$$z_j^{long} = y_j^{long} - y_{j-1}^{short}. \qquad (7)$$

Fig. 8 shows some typical results, indicating the restored independence signature in the long-gap data.

To make good use of the long dispersion-gap data, it needs to be compared with the corresponding short-gap data which is probing dynamically for the independence signature and hence the bottleneck.

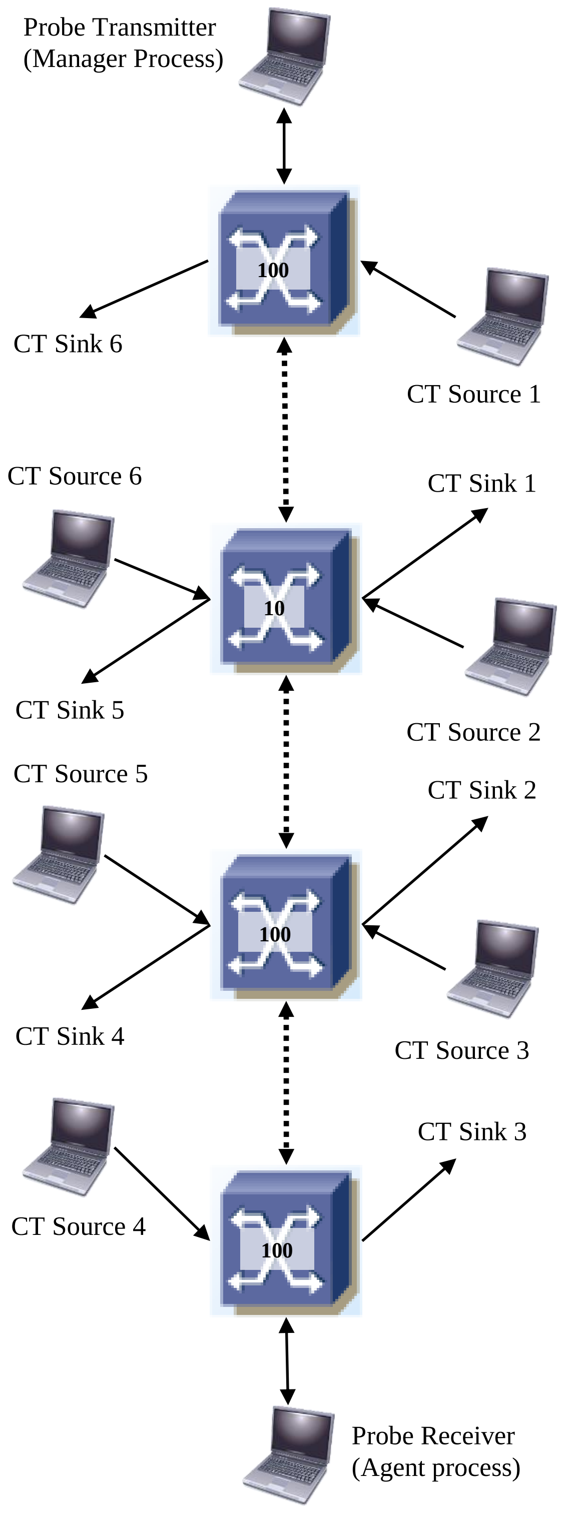

Figure 6: Simulated network path, representing 10/100 Ethernet. All devices are 100BaseT Fast Ethernet with the exception of the switch connecting CT Sources 2 and 6 which represents the 10BaseT narrow link. The three shared links are shown as broken lines.

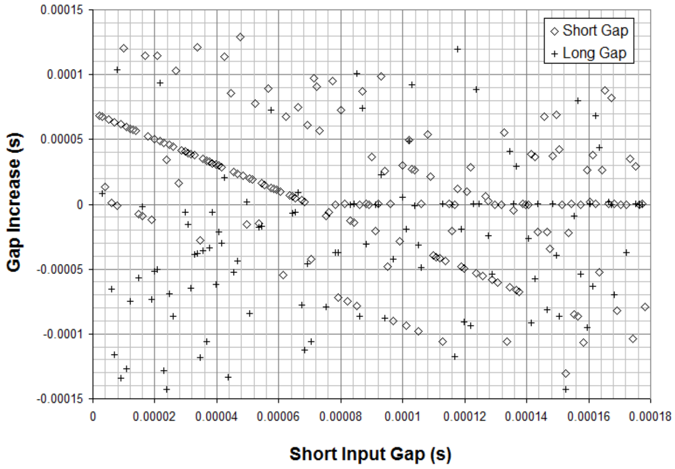

Figure 7: Increases in the short and long input gaps reported by the agent to the manager during a linear sweep of the short input gap over 100 seconds. The long input gap only shows an independence signature above the bottleneck gap.

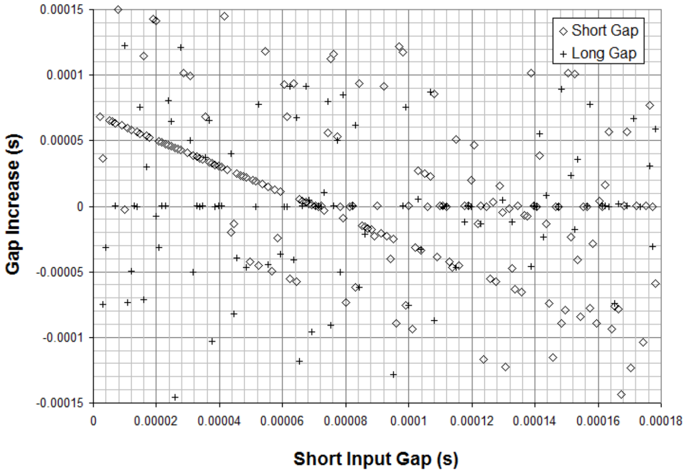

Figure 8: Increases in the short and long input gaps reported by the agent to the manager during a linear sweep of the short input gap over 100 seconds. Here the long input gap is adjusted by adding the dispersion increase measured for the previous short gap.

Fig. 9 shows cumulative frequency plot for long gap dispersion increase compared with the corresponding short-gap data for an input gap above the bottleneck value (110µs). Note that the while both pass (as should be expected) through median (0,0.5), the short-gap increase lacks rotational symmetry about this point, and also shows a much smaller variance. This is firstly because the short-gap output dispersion cannot fall below the corresponding input dispersion, and secondly because any dispersion increases must be caused by cross-traffic arriving during the restricted time between the two packets. More traffic has the potential to arrive during the long-gap, and thus the output dispersion can increase much further.

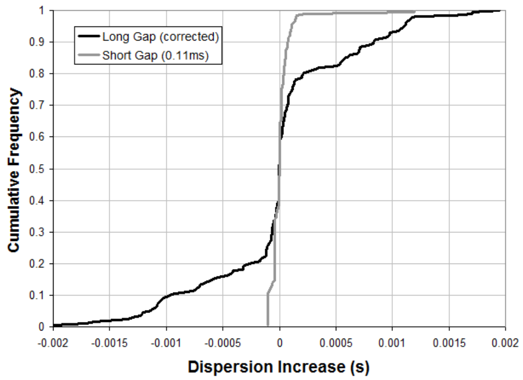

Figure 9: Dispersion increase distributions for long and short gaps, for short input gap above the bottleneck value. (Long gap=0.5s.)

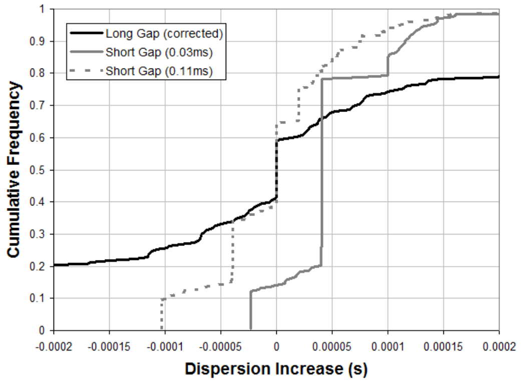

Figure 10: Expanded dispersion increase distributions for long and short gaps, for short input dispersion above and below the bottleneck threshold. (Long gap=0.5s.)

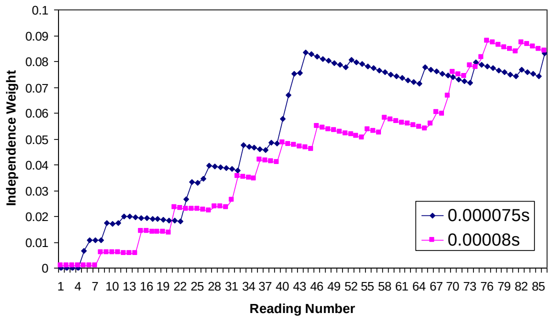

Figure 11. Typical profiles of independence weight vs. number of short-gap readings, using a learning rate of 0.01.

Fig. 10 shows an expanded version of Fig. 9, showing that the compact central modes of the two distributions (representing events where no cross-traffic has interfered, and any residual dispersion variation is due to elemental clock jitter) are almost identical. To this we compare some data obtained using a short input gap below the bottleneck threshold, namely 30µs. The central mode of *this* distribution lies distinctly to the right of the independence signature, indicating that this is either a rate or a distribution signature. (It is in fact the rate signature of the bottleneck. Distribution signatures of different nodes are also visible.)

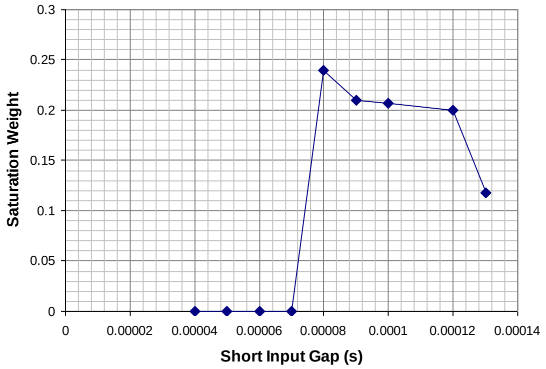

Figure 12. Typical saturation values for independence weight vs. short input gap. The bottleneck value is a little over 70µs.

D. Automated Independence Learning

To work well, this algorithm needs to detect the modal independence signature automatically and track any changes in variance that it may exhibit. For this reason we propose that the manager should continually track the long-gap dispersion increase throughout measurement and maintain an up-to-date model of the independence signature. For each value of z_i^{long} we update the existing estimate σ_{i-1}^2 of the independence signature's variance using the learning rule

$$\sigma_i^2 = (1-\alpha)\,\sigma_{i-1}^2 + \alpha\cdot\left(z_i^{long}\right)^2 \qquad (8)$$

where α is the learning rate (which here we set to 0.01). To limit the information to the true modal signature (where no cross-traffic has interfered) we filter out all values of z_i^{long} with magnitudes below a critical tolerance which we set to 100ns (one bit-time for 10BaseT). We also need a starter value σ_0 which we obtain by finding the standard deviation of the first 10 gaps while keeping $\Delta_{in}^{long} = \Delta_{in}^{short}$=0.5s to ensure *absolute* independence.) Using a Gaussian model, the degree to which a short-gap increase y conforms to the independence signature is $u(y,\sigma) = \exp\left(-y^2/2\sigma\right)$ which provides the input to a second learning process

$$w_j = (1-\alpha)w_{j-1} + \alpha\cdot u\left(y_j,\sigma_i\right) \qquad (9)$$

with $w_0 = 0$, which indicates the dynamically evolving weight of evidence in favor of an independence signature. Fig.11 shows the increase of this weight factor with time for two values of the short input gap slightly higher than the bottleneck value. These values initially increase but approximately saturate, and Fig. 12 shows typical saturation values for short input gaps above and below the bottleneck. The abrupt transition at about 70µs is easily detected and provides the basis for the algorithm discussed next.

E. Closing the Loop

The final stage of this exercise is to make the input gap vary dynamically to track the bottleneck value for the path, which can subsequently be used for bandwidth determination. To this end we make Δ^{short}_{in} increase linearly from an initial value significantly lower than the bottleneck dispersion, and to slow its rate of increase by an amount dependent on the current w_j i.e.

$$\Delta^{short}_{in(j)} = \Delta^{short}_{in(j-1)} + \beta\left(\lambda - w_j\right). \tag{10}$$

By selecting suitable values for the parameters (in this case $\beta = 0.00001$ and $\lambda = 0.1$) the short dispersion can be made to track the bottleneck value at which w_j begins to show rapid increase. Fig. 13 shows some typical results.

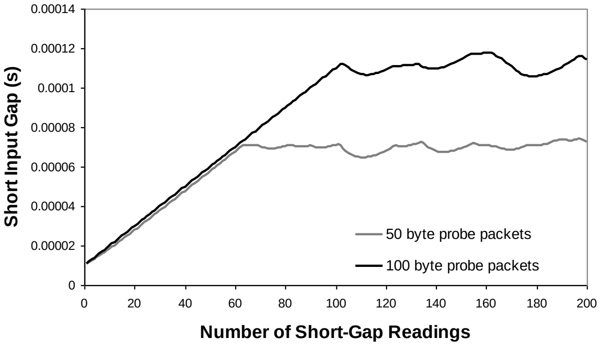

Figure 13: Typical time profiles for the short input gap under closed-loop control, for 50 and 100byte probe packets.

The results show average saturation values of 69.9μs for the 50 byte probe packets and 111μs for the 100 byte packets. We see from Fig. 2 that the difference between these values must equal to the difference between the packet sizes divided by the bottleneck link bandwidth. Thus the bandwidth is equal to:

$$\frac{(100-50)\times 8}{(111-69.9)\times 10^{-6}} = 9.732 \text{ Mbit/s}$$

and the 10BaseT switch is correctly identified as the narrow link for the path.

IV. Conclusions and Future Work

This paper demonstrates through a simulated experiment a simple scheme whereby a closed loop control of the input gap in a probe pair experiment allows the narrow-link bottleneck to be automatically identified. This technique, though successful in this static scenario under simple Poisson traffic needs further refinement and optimization to ensure its correct operation under a wider range of loading conditions and dynamic changes to the network infrastructure.

In particular more realistic traffic profiles need to be considered, based upon physically captured network traffic. Furthermore, the speed of the system's convergence needs to be improved, such that speedy measurements can be obtained, and changes in the network behaviour can be reliably tracked in real time.

In addition to this, the scheme currently measures only the narrow link capacity rather than the "available bandwidth" of the path. The latter is important for informing resource-aware applications which may need to adjust their compression ratios to accommodate the current bandwidth surplus [10]. However, we believe more information is encoded within the various dispersion signatures, and that further refinement of this scheme may provide useful available bandwidth data.

References

[1] K. Lai, M. Baker, "Measuring Bandwidth", Proc. IEEE INFOCOM, pp.905-14, 1999.

[2] B. Melander, M. Björkman, P. Gunningberg, "A New End-to-End Probing and Analysis Method for Estimating Bandwidth Bottlenecks", IEEE Globecom'00, San Francisco, CA, USA, Nov. 2000.

[3] V.J. Ribeiro, R.H. Reidi, R.G. Baranuiuk, J. Navratil, L. Cottrell, "pathChirp: Efficient Available Bandwidth Estimation for Network Paths", Proc. Passive and Active Measurement Workshop, 2003.

[4] A. Pásztor, D. Veitch, "On the Scope of End-to-End Probing Methods", IEEE Communications Letters, 6(11), pp.509-11, 2002.

[5] A. Pásztor, D. Veitch, "The Packet Size Dependence of Packet Pair Like Methods", IEEE/IFIP International Workshop on Quality of Service (IWQoS), 2002.

[6] M. Hosseinpour, M. Tunnicliffe, "Visual Feature Extraction for Classification of Packet-Sequence Dispersion Signatures and Identification of Bottleneck Parameters", Proc. 13th International Conference on Computer Modelling and Simulation (UKSim 2011), IEEE Computer Society, 30 Mar – 1 April 2009, Cambridge, U.K., pp. 488-91, 2011.

[7] M. Hosseinpour, M. Tunnicliffe, "Packet-pair behavior in wired and 802.11-type wireless connection and the use of data clustering algorithms for dispersion-mode tracking", 33rd MIPRO International Convention, 24 May – 28 May 2010, Opatija, Croatia, pp. 539-543, 2010.

[8] M. Hosseinpour, M. Tunnicliffe, "Real-Time tracking of packet-pair dispersion nodes ssing the Kernel-Density and Gaussian-Mixture models", 11th International Conference on Computer Modelling and Simulation (UKSim 2009), IEEE Computer Society, 25 Mar – 27 Mar 2009, Cambridge, U.K., pp. 548-552, 2009.

[9] M. Tunnicliffe, "The Measurement of Bandwidth: A Simulation Study" in "Modeling, Simulation and Optimisation: Tolerance and Optimal Control", Edited by Cakaj, S., InTech, April, pp. 285-304, 2010.

[10] Y. Yu, I. Cheung, A. Basu, "Optimal Adaptive Bandwidth Monitoring for QoS based Retrieval", IEEE Trans. on Multimedia, vol 5, pp. 466-72, 2003.