http://dx.doi.org/10.1109/UKSIM.2011.99

Visual Feature Extraction for Classification of Packet-Sequence Dispersion Signatures and Identification of Bottleneck Parameters

Martin J. Tunnicliffe, Mehri Hosseinpour

Faculty of Computing, Information Systems and Mathematics,
Kingston University,
Kingston-on-Thames, Surrey, UK, KT1 2EE.
e-mail: M.J.Tunnicliffe@king.ac.uk

***Abstract* - As an alternative to pdf approximation methods to identify modes within the dispersion data of a packet-pair network probing experiment, the classical Hough transform is used to identify straight lines within the output vs. input dispersion plot. These lines can then be mapped to the independence, rate, and distribution signatures of the different links within that path.**

Keywords - Queuing Networks; Bandwidth Probing; Visual Feature Extraction.

I. Introduction

End-to-end packet-pair dispersion has been studied extensively since it was first introduced in the late 1990's [1]. Under the ideal conditions the principle is very straightforward: Sequences of two or more closely spaced packets are injected at the source and collected at the sink, and the increase in packet separation (dispersion) is used to determine bottleneck parameters. However multiple tandem links and cross-traffic introduce additional signatures which require statistical filtering algorithms. Here we investigate a novel approach based on the Hough transform used to identify geometrical features in visual images.

II. Dispersion Signatures

Figure 1 shows a portion of a network path model consisting of tandem chain of FIFO queuing systems, each independently loaded with cross-traffic. When of closely spaced packets (S and S' bits) pass along the chain, their dispersion Δ is modified at each hop:

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

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 in (1) ("accumulation signature" [2]) can be suppressed by making the packets equal size. If there is no cross traffic on the link, then (1) becomes:

$$\Delta_{n+1} = \max\left(\Delta_n, \frac{S + H_n}{B_n} + G_n\right) \qquad (2)$$

where H_n (bits) and G_n (seconds) are the link-layer header and inter-frame gap for link n. For large Δ_n the input and output dispersions are equal ("independence signature" [2]). As the input dispersion decreases the output dispersion saturates at a value which does not depend on Δ_n, but on the link parameters: This is the "rate signature" [2].

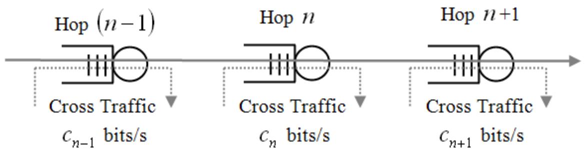

Figure 1. Network path consisting of a chain of queue systems, each carrying independent cross-traffic.

If the initial input dispersion is Δ_{in} seconds then (assuming no cross-traffic on *any* of the links) the output dispersion $\Delta_{out} = \max\left(\Delta_{in}, G_b + \left(S + H_b\right)/B_b\right)$ seconds, where G_b, H_b and B_b are the respective inter-frame gap (seconds), link-layer header (bits) and bit-rate (bits per second) of the bottleneck or "narrow" link. However since the effects of inter-frame gap and header are externally indistinguishable it is convenient to lump them into a single parameter $G'_b = G_b + H_b/B_b$ seconds which we term the "effective inter-packet gap". Thus:

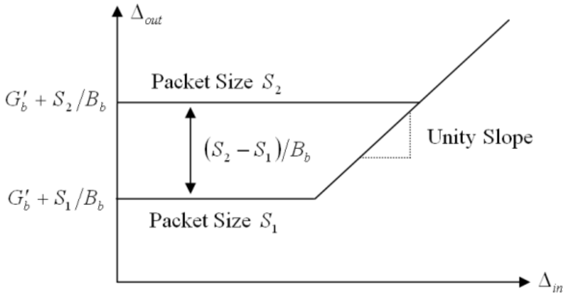

Figure 2: Idealized graph of output vs. input dispersion for two different packet sizes. Without cross-traffic this is dictated by the path bottleneck.

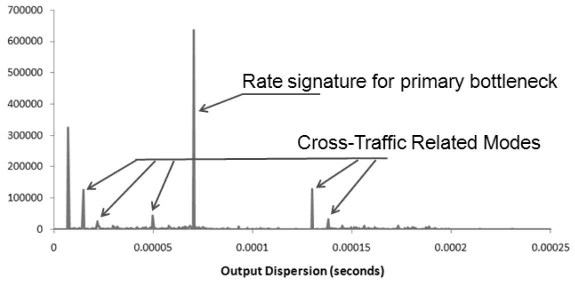

Figure 3: Graph of output dispersion for a loaded two-node network path, showing rate signature and cross-traffic modes.

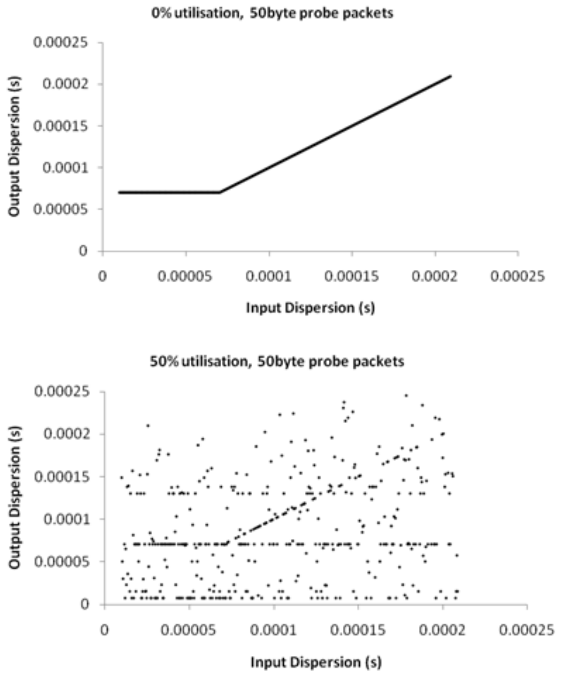

Figure 4. Simulated graphs of output vs. input dispersion for a two-hop network path with and without cross-traffic. Input dispersion is randomized.

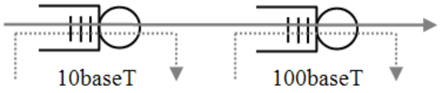

Figure 5. Simulated network path. Cross traffic rate is adjustable to give different utilization levels.

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

By varying S and Δ_{in} and observing Δ_{out} the values of B_b and G_b' can be determined (see Fig. 2) and thus the maximum effective data-rate can be computed from the formula:

$$B_{eff}(s) = B_b s / \left(B_b G_b' + s\right)\ . \tag{4}$$

where s is the data packet size in bits.

When cross-traffic is added the picture becomes considerably more complex. Cross-traffic packets become inserted between probe packets, forcing them apart and giving rise to the “distribution signature” [2]. If all the probed links have constant bandwidth, header size and inter-frame spacing and the probe-packets are relatively small compared to the average cross-traffic packets, the output dispersion follows a random distribution with many sharply defined modes (Fig. 3). Only one of these modes corresponds to the simple model described above, and must therefore be isolated by some filtering mechanism. (In wireless networks, variable inter-frame gaps introduce disperse distributions which are more problematic [3].) While the distribution’s “primary mode” encodes the all-important bottleneck parameters, other nodes carry useful information about network topology and traffic profiles which impact upon the available bandwidth. There is therefore a need to identify classify and record the development of each distribution peak.

III. Interpretation of Data

The mode-positions in Fig. 3 are dynamic, and shift as the input dispersion and packet-size is changed. In an earlier paper [4] the authors attempted to use spacio-temporal kernel functions and Gaussian tracking algorithms originally developed for foreground classification in moving images. Here however we take a different approach: ignoring the distributions themselves we revisit the Δ_{out} vs. Δ_{in} plot previously discussed (Fig. 4).

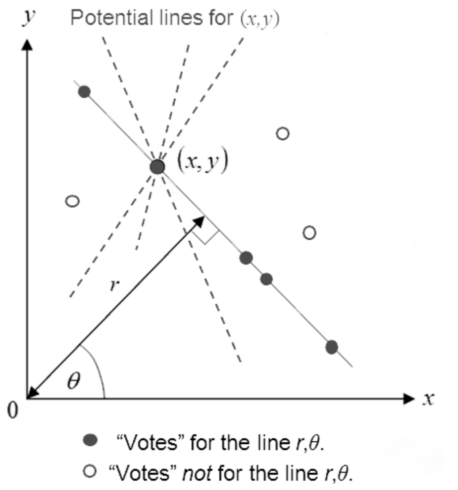
y
Potential lines for (x,y)
(x,y)
r
θ
0
x
● "Votes" for the line r,θ.
○ "Votes" *not* for the line r,θ.

Figure 6. Hough Transform illustration. The plane is mapped to an accumulator array whose members are incremented when that region of the plane is visited.

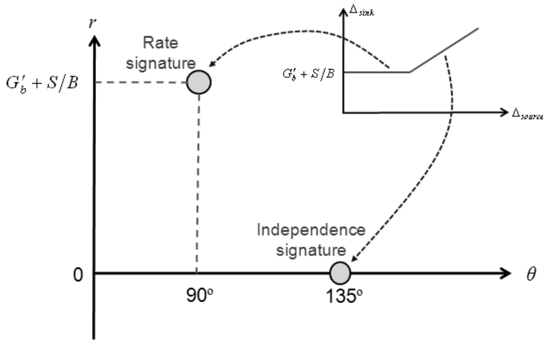

Figure 7. Expected mapping of ideal dispersion graph with corresponding Hough plane.

Figure 4 shows some results for a simulated two-hop network path, containing one 10baseT and one 100BaseT Ethernet link (Fig. 5), with the input dispersion uniformly randomized. Without cross-traffic the graph resembles Fig. 2, but when the utilization is increased to 50% the distribution modes appear as lines in the in the two-dimensional scatter-plot. (Cross-traffic packet sizes followed the same profile used in [4].)

Plane geometrical features such as lines, circles and ellipses can be identified by the Hough Transform [5], whose classical form detects probable straight lines amongst discrete data-points, and was originally developed for the automated analysis of cloud-chamber images. The Hough Transform is based upon a very simple concept: 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. 6) which represents the set of all possible lines passing through the point (x, y). If this function is computed for *all* data-points across the range $\theta \in (0, \pi)$, the most often visited regions in the (r, θ) plane must correspond to the most probable straight-line features within the data. In practice this is achieved by mapping r and θ to discrete integers and using these to increment the contents of a two-dimensional "accumulator" array. Finally the local maxima within the accumulator are detected and mapped back to their corresponding (r, θ) pairs.

There are a few important caveats: Firstly the accumulator must be large enough to eliminate significant quantization error, yet small enough that a significant number of "votes" fall into each relevant bin. A single line may in practice correspond to a cluster of closely-spaced maxima which must be grouped together for analytical purposes. Secondly the raw transform gives no indication of the special extent of the lines, which must be obtained by referring back to the data.

IV. Results

Figure 7 shows the idealized mapping of the rate and independence signatures to their corresponding points in the Hough plane, and Fig. 8 shows the corresponding results obtained from a simulation without cross-traffic. It is clear that the major peaks correspond to the points where the signatures are expected to lie. Figs. 9 and 10 show how the Hough data changes as the utilization is increased. At 15% the rate and independence signatures of the 10baseT link (the bottleneck) are still visible, as well as a the 10baseT's distribution signature. Also the rate and distribution signatures of the 100baseT link are beginning to become visible. (The independence signature of the 100baseT is merged with that of the 10baseT, and is not independently visible.) When the utilization is increased to 70% (Fig. 10) the Hough data is mostly dominated by the 100baseT link, the 10baseT's signatures appearing as secondary peaks.

V. Conclusions and Future Work

This brief paper demonstrates the use of the Hough algorithm to interpret packet-pair dispersion data. The simulations are based on a relatively simple scenario, consisting of two queuing systems connected in tandem, representing 10baseT and a 100baseT Ethernet links. The results show the expected signatures of both links, though their levels of significance depend upon the utilization of the network links.

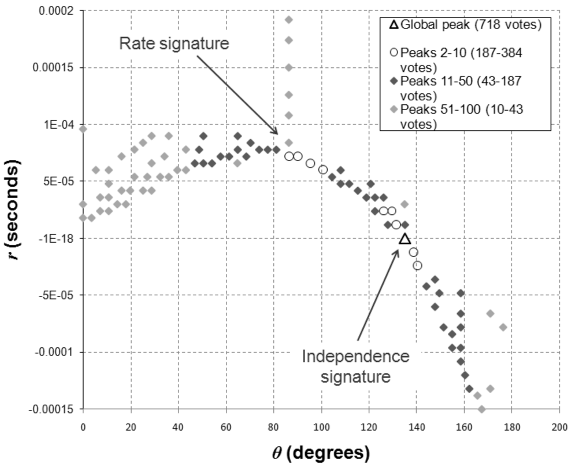

Figure 8. Hough transform peaks obtained from simulation with 0% cross-traffic utilization.

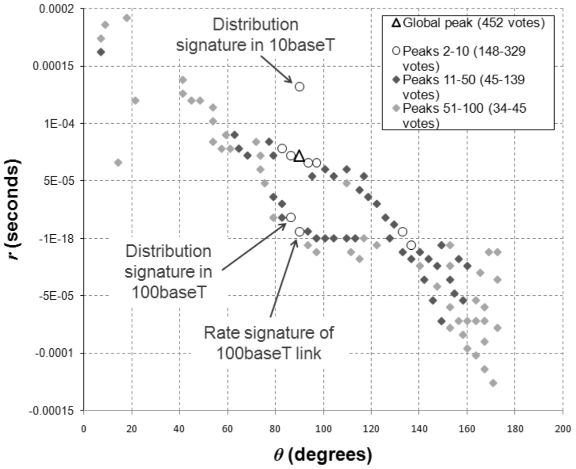

Figure 9. Hough transform peaks obtained from simulation with 15% cross-traffic utilization.

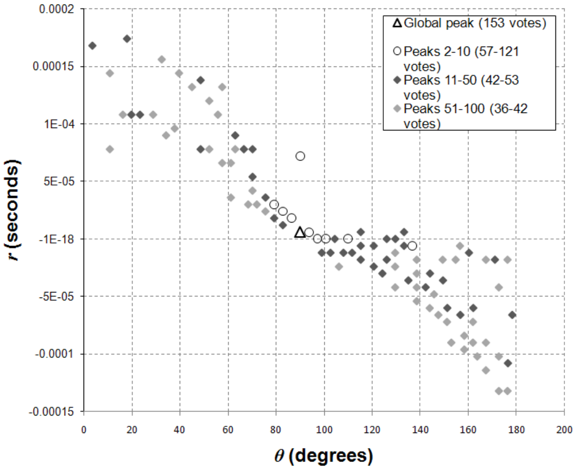

Figure 10. Hough transform peaks obtained from simulation with 70% cross-traffic utilization.

So far the only independent variable in the experiment is the input dispersion, which varies according to a uniform random distribution. Future work will investigate other variables such as the relative probe packet sizes, numbers of packets in a stream etc.

REFERENCES

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

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

[3] 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.

[4] 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.

[5] E.R.Davies, “Machine Vision: Theory, Algorithms, Practicalities” Academic Press, London, pp.191-206, 1990.