Hosseinpour, M and Tunnicliffe, M.J. (2010) Packet-pair behavior in wired and 802.11-type wireless connection and the use of data clustering algorithms for dispersion-mode tracking. In Biljanovic, Petar, Skala, Karolj, Golubic, Stjepan, Bogunovic, Nikola, Ribaric, Slobodan, Cicin-Sain, Marina, Cisic, Dragan, Hutinski, Zeljko, Baranovic, Mirta, Mauher, Mladen and Pletikosa, Marko, (eds.) (2010) MIPRO 2010 : 33rd Internation Convention on Information and Communication Technology, Electronics and Microelectronics : Proceedings. Rijeka, Croatia : Croatian Society for Information and Communication Technology, Electronics and Microelectronics. pp. 539-543. ISBN 9789532330502

http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5533452&isnumber=5533310

Packet-Pair Behavior in Wired and 802.11-type Wireless Connection and the use of Data Clustering Algorithms for Dispersion-mode Tracking

M. Hosseinpour, M. J. Tunnicliffe
Faculty of Computing, Information Systems and Mathematics, Kingston University,
Kingston-on-Thames, Surrey, KT1 2EE. +20-85472000+62674 M.J.Tunnicliffe@king.ac.uk

Abstract - Packet-pair bandwidth probing in wired-cum-wireless network paths was tested and analyzed in a C++ simulation environment using link models verified alongside Opnet results. Some major differences were noted between these results and those of pure wired scenarios investigated in earlier work. Attempts were made to use a dynamic Gaussian-mix algorithm to identify data clusters within the bandwidth distribution.

I. INTRODUCTION

Techniques for estimating the bandwidth of a network path tend to rely on assumptions about network behavior which, while generally true in wired networks of switches and routers, may not necessarily be true in the case of broadband access and wireless networks. The techniques themselves (summarized by Prasad et al. [1]) may be classified according to what they actually measure: (i) The individual link bandwidths vs. the end-to-end path capacity and (ii) the maximum potential throughput vs. the throughput available to a specific user.

Here we consider one particular technique: Packet Pair/Train Dispersion (PPTD) probing aims to measure maximum end-to-end capacity by injecting multiple pairs (or trains) of identical-sized probing packets whose resulting dispersion provides an estimate the path capacity. Under the simplest assumptions, if two probe packets are introduced Δ_{in} seconds apart and emerge Δ_{out} seconds apart then if no cross-traffic interferes:

$$\Delta_{out} = \max\left(\Delta_{in}, P/l\right) \qquad (1)$$

where P is packet size in bits and l the smallest link capacity (bits/s) in the path (the *narrow link*.)

However, several factors combine to complicate this simple picture: Firstly cross-traffic may delay one or both of the probing packets: When the first packet is delayed more than the second, the dispersion is increased, causing a bandwidth underestimation. Similarly if the second packet experiences the greater delay then the bandwidth is overestimated. The "true" bandwidth stands as a local node within the dispersion distribution surrounded by spurious cross-traffic nodes which may change their positions and sizes as the cross-traffic varies. Secondly the packet transmission time is not the only cause of latency within a network link. The processing of link-layer headers, as well as inter-frame spacing may introduce further delays which are both statistically variable and independent of packet size [2].

In an earlier paper [3] we considered the use of two techniques to track node behavior: A modified version of the Kernel Density method [4] the Gaussian Mixture Model (GMM) borrowed from the field of machine vision [5]. However, the network environments tested were simplistic, and assumed ideal queuing behavior at each node. In the current paper we apply the same Gaussian Mixture technique to a more realistic simulation representing Ethernet and Wi-Fi connections.

II. MODELING LINK

The main simulation tool used for this work was based on the C++ classes developed in [6] using node models representing Ethernet and 802.11 wireless connections. Figure 1 shows the basic model operation: The packet processing time consists of the time to process the network-layer packet and data-link header fields (including the trailer and preamble) and an inter-frame gap (IFG) which has both fixed and random components. The objects were parameterised so as to mimic the observed behaviour of Opnet simulations of Ethernet and Wireless links (see Figures 2-5).

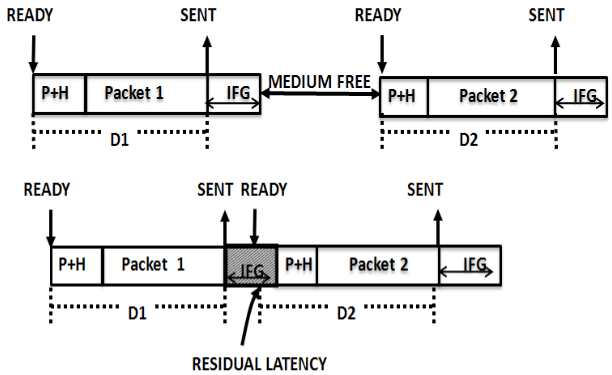

Fig. 1. Representation of generic link model implemented in C++ class.

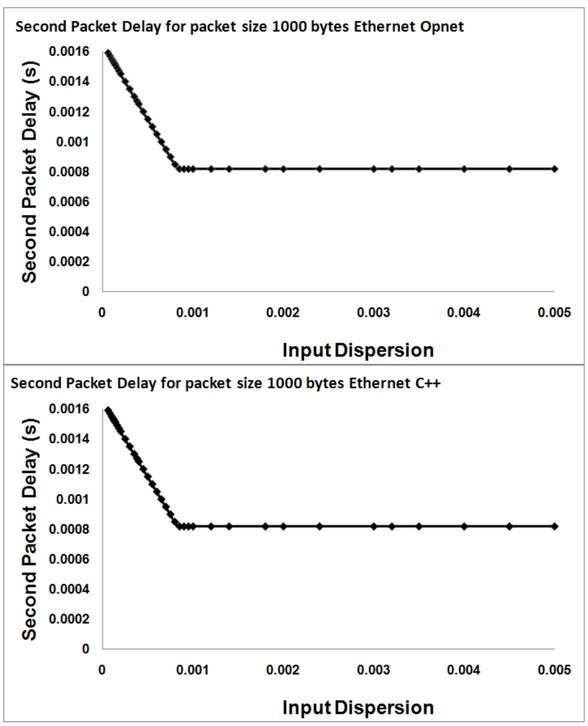

Fig. 2. Second Packet Delay with Probe Packet 100 bytes (Ethernet). Comparison between Opnet and C++ model.

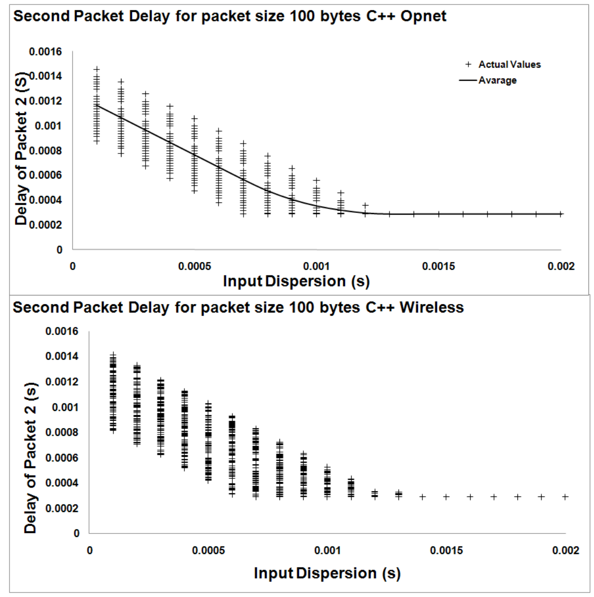

Fig. 3 Second Packet Delay with Probe Packet 100 bytes (Wireless). Comparison between Opnet and C++ model.

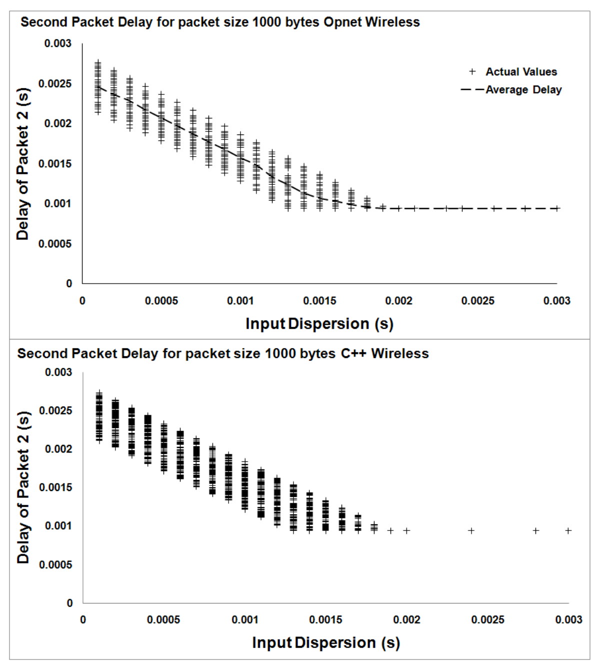

Fig. 4. Second Packet Delay with Probe Packet 1000 bytes (Wireless). Comparison between Opnet and C++ model.

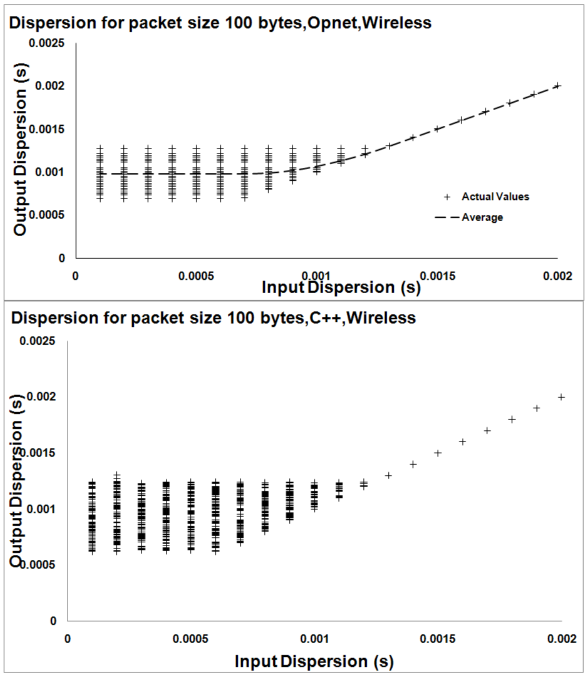

Fig. 5. Output Dispersion probe packet 100 byte: Opnet and C++ model.

The model parameters are the raw speed of the link *S* (bit/s), the header, trailer and preamble H (bits), the Inter-frame Gap (IFG) t_{ifg} (seconds) with a fixed component t_{fix} and a variable component t_{var} (wireless only) which was assumed to follow a uniform distribution. Figure 1 shows how the IFG interferes with dispersion: In a sub-congested link the medium is free and we have a clear gap between 2 packets. In a congested link the second packet has to wait for the IFG from the first packet to expire before it can be serviced. The accessible throughput of the link is therefore given by

$$l = \frac{P}{\frac{H+P}{S} + \overline{t_{ifg}}} \quad \text{bits/s} \qquad (2)$$

Where *P* is the packet size (bits), *S* is the raw speed of the link (bits/s) and $\overline{t_{ifg}}$ is the average inter-frame gap (the fixed component plus half the variable component). For the wired Ethernet link (which represented 10BaseT) the accessible throughput for 1000 byte packets was 9.634Mbit/s, while for the Wireless link (802.11b) with 11Mbit/s raw bandwidth, the corresponding value was 5.043Mbit/s. Thus when the two links are combined in tandem, the wireless provides the bottleneck link.

III. MODELING WIRED-CUM-WIRELESS NETWORK SCENARIOS

Having established C++ objects to represent 10BaseT and 802.11b links (as modeled by Opnet), these were combined to form the wired-cum-wireless scenarios shown in Figure 6, representing the wireless "last mile" and "first mile" configurations. Figure 7 shows a typical distribution of bandwidth estimates based on packet dispersions obtained from the last-mile simulation: (P / Δ_{out}) The flat feature to the right of the histogram represents the true bottleneck bandwidth spread over a range associated with the variable inter-frame gap. The spurious peaks to the left represent dispersions associated with cross-traffic in the upstream wired link.

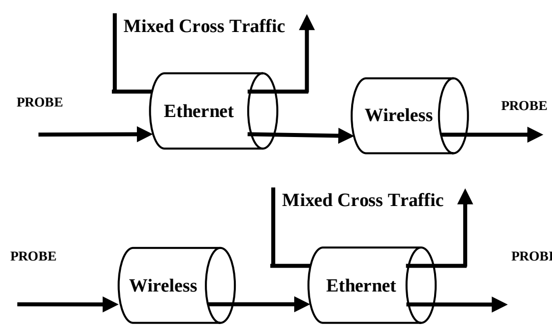

Fig. 6. Simulated wired-cum-wireless scenarios using "last mile" and "first mile" wireless bottlenecks.

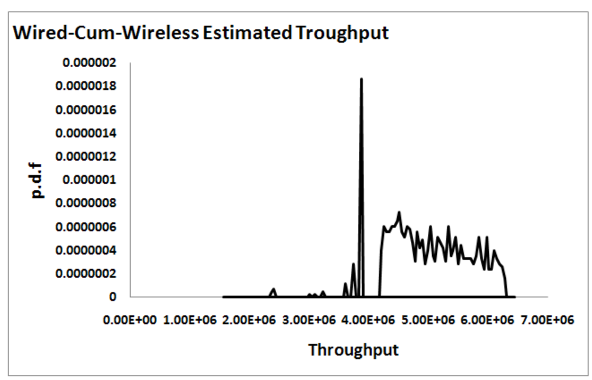

Fig. 7. Typical dispersion profile for wired-cum-wireless simulation.

IV. THE GAUSSIAN-MIXTURE MODEL

The histogram distribution shown in Fig.7 was based upon 10,000 packet pairs spaced 1s apart, thus representing nearly 3 hours of real time. In order to obtain bandwidth information in shorter periods than this, we have investigated techniques for estimating data clusters (or modes) within the results. One such method is the EM algorithm [7] which uses a mix of Gaussian components to represent a multimodal distribution, but this is itself computationally costly. A more rapid Gaussian-mix technique devised by Stauffer and Grimson [5] was investigated in an earlier paper [3] and is applied here.

Suppose we represent the history of the output dispersion Δ_{out} as $\{\Delta_1, \Delta_2 ... \Delta_t\}$, where t is time expressed as the number of packet-pair transmissions since the experiment began. Now suppose we represent the probability density function for Δ_t as a weighted sum of K Gaussian distributions:

$$f(\Delta_t) = \sum_{i=1}^{K} \omega_{i,t} \cdot \eta(\Delta_t | \mu_{i,t}, \sigma_{i,t}) \tag{3}$$

Where

$$\eta(\Delta|\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{(\Delta-\mu)^2}{2\sigma^2}\right] \tag{4}$$

and $\omega_{i,t}$ represents the probabilistic weighting of the Gaussian component i at time t. We classify a dispersion measurement Δ_t as belonging to distribution i if $|\Delta_t - \mu_{i,t}| / \sigma_{i,t} < 2.5$. In the case of multiple matches the closest match is selected and if *no* existing distribution matches a new Gaussian is created with a mean of Δ_t, standard deviation 0.1Mbit/s and weighting probability 0.01. If k represents the distribution selected for a particular dispersion then the weightings are adjusted according to the rule

$$\omega_{i,t} = \begin{cases} (1-\alpha)\omega_{i,t-1} + \alpha\,; & i = k \\ (1-\alpha)\omega_{i,t-1}\,; & i \neq k \end{cases} \tag{5}$$

where α is the learning rate (which we set to 0.01). and renormalize such that the weightings again sum to unity. Adjustments to $\mu_{i,t}$ and $\sigma_{i,t}$ are applied only to the matched distribution, i.e.

$$\mu_{k,t} = (1-\rho)\mu_{k,t-1} + \rho\Delta_t \tag{6}$$

$$\sigma_{k,t} = \sqrt{(1-\rho)\sigma_{k,t-1}^2 + \rho(\Delta_t - \mu_{k,t})^2} \tag{7}$$

where ρ is the learning rate adjusted according to the degree to which the new measurement fits the distribution, given by:

$$\rho = \alpha \cdot \exp\left[-\frac{(\Delta - \mu_{i,t})^2}{2\sigma_{i,t}^{\;2}}\right]. \tag{8}$$

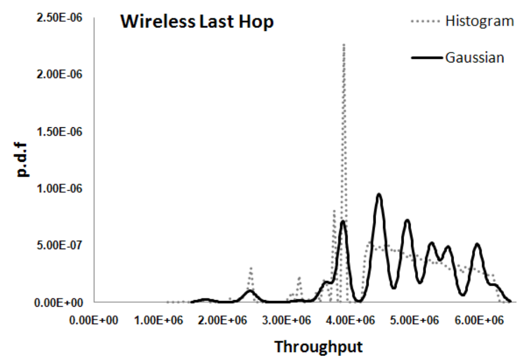

Fig.8. Last-mile dispersion profiles captured by Gaussian Mix model.

V. RESULTS

Figures 8 and 9 show typical results obtained using the Gaussian mix model based on 500 data points (8.3 minutes) compared with the corresponding histogram results based on 10,000 data points (2.78 hours). In Figure 8 the major features of the histogram are captured by the Gaussian model, though the continuous wireless bottleneck feature is transformed into a series of discrete Gaussians. In Figure 9 this continuous feature is less visible as the downstream wired link superimposes spurious cross-traffic related peaks upon the wireless link's distribution. These peaks are captured much more accurately by the Gaussian-mix model.

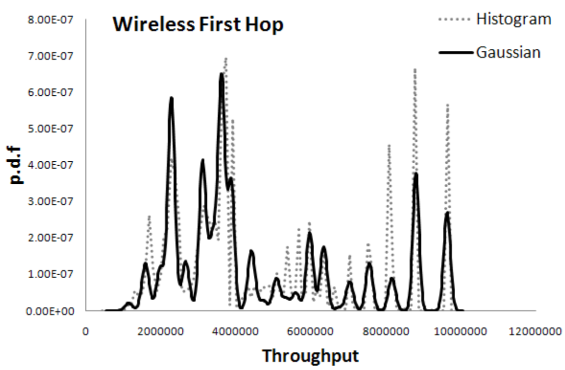

Fig.9. First-mile dispersion profiles captured by Gaussian Mix model.

VI. DISCUSSION AND CONCLUSION

This paper has used simulation to investigate the use of a Gaussian-mixture model to interpret packet-pair dispersion data in a wired-cum-wireless network. The simulation and analysis were performed in C++, though the link models were parameterized and verified by comparison with Opnet results. The results show that unlike the bandwidth modes of wired-links, the continuous features produced by the wireless link are not well represented by Gaussian mixtures.

One feature of the work so far is that the wireless link is assumed to be uncontested by cross-traffic. The presence of cross-traffic is likely to introduce further complications, which will be investigated in future work.

REFERENCES

[1] Prasad, R, Dovrolis,C, Murray,M, Claffy,K, "Bandwidth Estimation: Metrics, Measurement Techniques and Tools", *IEEE Network*, 17(6), pp.27-35, 2003.

[2] M. Hosseinpour, M.J. Tunnicliffe, "TOPP Probing of Network Links with Large Independent Latencies", pp.381-386, PGNet2007.

[3] M. Hosseinpour, M.J. Tunnicliffe, "Real-Time Tracking of Packet-Pair Dispersion Nodes using the Kernel-Density and Gaussian-Mixture Models", pp.548-52. DOI-10.1109/UKSIM.2009.

[4] Lai,K, Baker,M, "Measuring Bandwidth", Proc. IEEE INFOCOM'99, pp.904-14, 1999.

[5] Stauffer,C, Grimson,W.E.L., "Learning Patterns of Activity Using Real-Time Tracking", IEEE Trans. Pattern Analysis and Machine Intelligence, 22(8), pp.747-57, 2000.

[6] Tunnicliffe,M.J., "The Measurement of Bandwidth: A Simulation Study", in "Modelling, Simulation and Optimization", ed. Vedran Kordic (in press) , IN-TECH, 2010.

[7] Tom M.MichellMc-Grow Hill, "Machine Learning", Mc\Graw-Hill, New York,1997 ,page 191-196.