

© 2009 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

<https://dx.doi.org/10.1109/UKSIM.2009.74>

Real-Time Tracking of Packet-Pair Dispersion Nodes using the Kernel-Density and Gaussian-Mixture Models

M. Hosseinpour, M. J. Tunncliffe

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

Abstract

A brief simulation study of real-time packet-dispersion mode-tracking using the Gaussian-Mix Model (originally devised for real-time background classification in moving pictures) and an adaptation of the Kernel-Density Estimator is presented. The simulated environment consisted of two FIFO store-and-forward nodes where the probe packets interact with Poisson and Pareto-generated cross-traffic with a range of packet sizes. The two models produced broadly similar results, able to track node activity under the dynamically changing conditions associated with the Pareto cross-traffic. The Gaussian model sometimes replaced the primary mode with a double peak, which disappeared when some of the model's parameters were changed.

1. Introduction

Techniques for estimating the bandwidth of a network path (summarized by Prasad et al. [1]) may be classified two ways: (i) Whether they measure the bandwidths of individual network links or the end-to-end capacity of a path, and (ii) whether they measure the maximum potential capacity or the bandwidth available to a particular user. Here we consider one particular technique: Packet Pair/Train Dispersion (PPTD) probing, which aims to measure maximum end-to-end capacity.

PPTD injects pairs (or trains) of closely spaced probing packets, whose resulting dispersion provides an estimate the path capacity. Suppose that two probe packets are introduced into the path Δ_{in} seconds apart and emerge Δ_{out} seconds apart at the other end. If no cross-traffic interferes then

$$\Delta_{out} = \max(\Delta_{in}, S/l) \quad (1)$$

where S is the packet size in bits and l is the smallest link capacity (bits/s) within the path. (The

latter is usually called the *narrow link*, not to be confused with the *tight link* - the hop with the smallest available bandwidth.)

Cross-traffic complicates this simple picture by delaying one or both of the probing packets, thus interfering with the dispersion mechanism. When the first packet is delayed more than the second the dispersion is increased, causing an underestimation of the narrow-link bandwidth. Similarly if the second packet experiences greater delay than the first, the bandwidth is overestimated. Thus the "true" bandwidth stands as a local node within the dispersion distribution, surrounded by spurious cross-traffic nodes which must be statistically filtered. Furthermore, these nodes change their positions and sizes as the cross-traffic conditions change dynamically over time.

For the remainder of this paper we consider two models for tracking the activity of nodes in real-time. The first of these - the Gaussian Mixture Model (GMM) - is borrowed from the field of machine vision where it was developed to classify background activity in moving images [2]. The second model - the Kernel Density Estimator (KDE) - is an adaptation of a technique investigated by Lai and Baker [3]. Both models are tested using both Poisson and Pareto cross-traffic (the latter based on the ON/OFF scheme described in [4]) in a simulated network environment written in C++ [5]. Figure 1 shows the two-hop network topology employed, and Table I shows the distribution of packet-sizes within the cross-traffic.

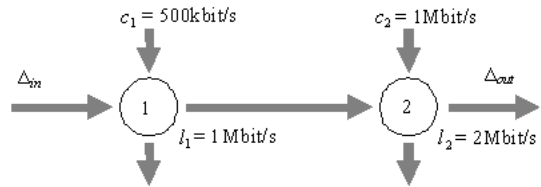


Figure 1. Simulated network topology. Five hundred pairs of probe packets, 1500 bytes each were transmitted 0.1ms apart over 50s. Initial packet dispersion was 1ms.

Table I: Composition of Cross Traffic [6]

Packet Size (bytes)	% Total Packets
60	46
148	11
500	11
1500	32

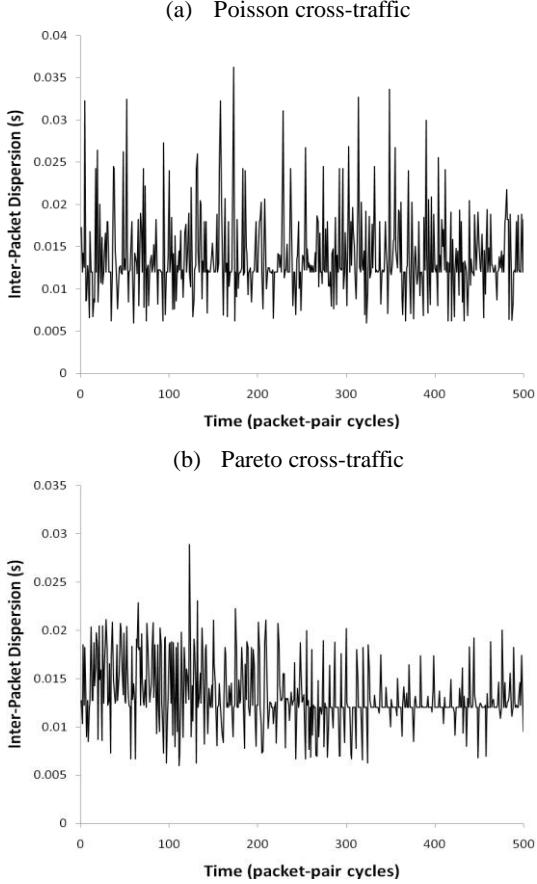


Figure 2. Typical dispersion profiles for the two-node network path of Figure 2 using the packet-size profile of Table I. Pareto streams had mean ON-time 30s, minimum ON-time 10s and a mean (Poisson) OFF-time of 50s.

2. The Gaussian-Mixture Model (GMM)

Suppose we represent the history of the output dispersion Δ_{out} as $\{\Delta_1, \Delta_2, \dots, \Delta_t\}$, where t is time expressed as the number of packet-pair transmissions since the experiment began. (Figure 2 shows typical profiles, obtained using the simulated network path of Figure 1.) Now suppose we represent the probability density

function for Δ_t as a weighted sum of K Gaussian distributions:

$$f(\Delta_t) \approx \sum_{i=1}^K \omega_{i,t} \cdot \eta(\Delta_t | \mu_{i,t}, \sigma_{i,t}) \quad (2)$$

where

$$\eta(\Delta | \mu, \sigma) = \frac{1}{\sigma \sqrt{2\pi}} \exp \left[-\frac{(\Delta - \mu)^2}{2\sigma^2} \right] \quad (3)$$

and $\omega_{i,t}$ represents the probabilistic weighting of the Gaussian component i at time t . Following Stauffer and Grimson [2] we classify a dispersion measurement Δ_t as belonging to distribution i if and only 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 $\Delta_t / 20$ and weighting probability 0.01.

If k represents the distribution selected for a particular dispersion then the weightings are adjusted according to the rule

$$\omega_{i,t} = \begin{cases} (1 - \alpha)\omega_{i,t-1} + \alpha; & i = k \\ (1 - \alpha)\omega_{i,t-1}; & i \neq k \end{cases} \quad (4)$$

where α is the learning rate (which we set to 0.02 for a response time-constant of $1/0.02=50$ packet-pair cycles, i.e. one tenth of the experiment's time-span) 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.

$$\mu_{k,t} = (1 - \rho)\mu_{k,t-1} + \rho\Delta_t \quad (5)$$

$$\sigma_{k,t} = \sqrt{(1 - \rho)\sigma_{k,t-1}^2 + \rho(\Delta_t - \mu_{k,t})^2} \quad (6)$$

where ρ is the learning rate adjusted according to the degree to which the new measurement fits the distribution. Stauffer and Grimson [2] used $\rho = \alpha \cdot \eta(\Delta_t | \mu_{k,t}, \sigma_{k,t})$, but this

creates problems for very narrow distributions requiring $\rho > 1$. Here we use a “compromise” formula which ensures that ρ never exceeds α :

$$\rho = \alpha \cdot \sigma_{i,t} \sqrt{2\pi} \cdot \eta(\Delta_t | \mu_{k,t}, \sigma_{k,t}). \quad (7)$$

3. The Kernel-Density Estimator (KDE)

Kernel-Density Estimation is a well known technique for estimating the probability density function of a random variable from a finite set of observations. It uses a “kernel function” $K(x)$, typically a Gaussian or symmetrical triangular function with zero mean, unity width/standard deviation and a unity integral between $\pm\infty$. The estimated distribution is

$$f(\Delta_t) \approx \frac{1}{t} \sum_{i=1}^t \frac{1}{h} K\left(\frac{\Delta_t - \Delta_i}{h}\right) \quad (8)$$

where t is the number of observations and h is the “kernel width” (loosely speaking, the minimum distance between Δ and observation Δ_i in order that the observation should affect the distribution value at Δ .) Lai and Baker [2] used this approach to identify the dominant node of a dispersion distribution, which was assumed to represent the narrow-link capacity. (This is not necessarily the case [1].) However, it requires an arbitrarily selected value for h : Too small an h merely reproduces the observations as a series of spikes, while too large an h obscures genuine data-clustering.

A small modification to the kernel-density model allows it to respond dynamically in a similar manner to the Gaussian-mix model: We replace the constant kernel width with a value which increases exponentially with the antiquity of the observation being included:

$$f(\Delta_t) \approx \frac{1}{t} \sum_{i=1}^t \frac{1}{h(1-\beta)^{i-t}} K\left(\frac{\Delta - \Delta_i}{h(1-\beta)^{i-t}}\right) \quad (8)$$

where h is the *maximum* kernel width and β governs the rate of change to new information.

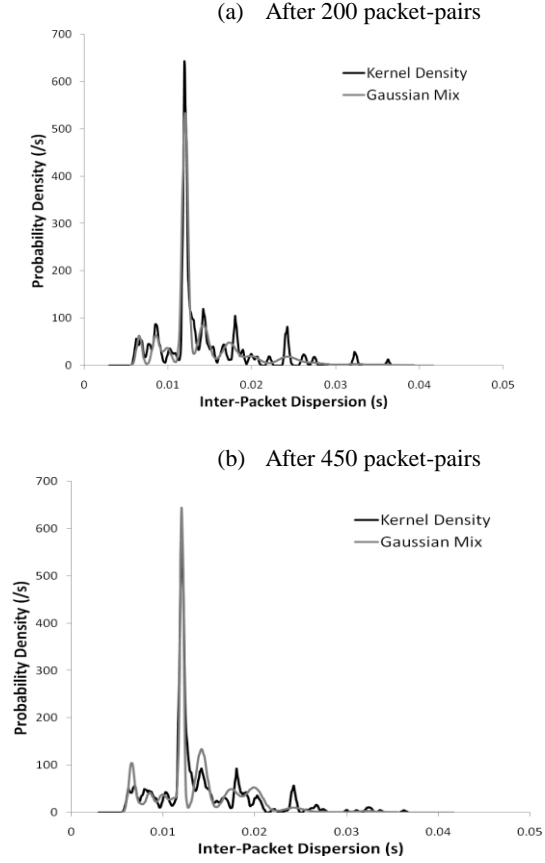


Figure 3. Dispersion probability density functions obtained using the Poisson cross-traffic at two instants during the experiment.

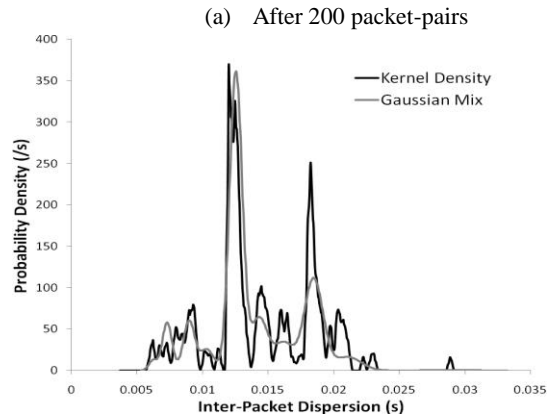
4. Results

Figures 3 and 4 show the probability density functions obtained using both models for three points within the simulation data of Figure 2. For the GMM, no upper limit was set on the number of Gaussians, which was found to vary between 10 and 12. In the KDE model, the maximum kernel width h was set to $1/120^{\text{th}}$ of the maximum observed dispersion range and $\beta = 0.001$. A triangular function was used.

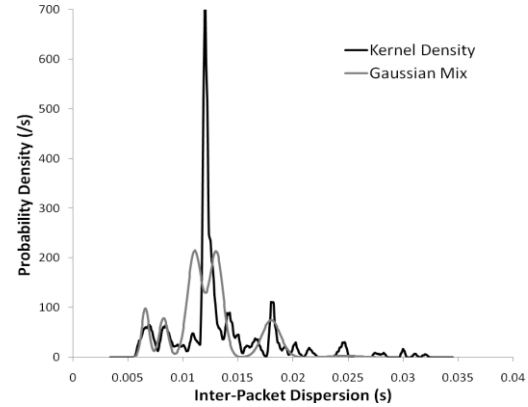
Figure 3(a) and 3(b) are broadly similar, despite being separated in time by 20s. (Figure 2(a) suggests that the Poisson data are a roughly stationary process.) Both models pick out the dominant mode at 12ms, which corresponds to $1500 \times 8 / 0.012 = 1\text{Mbit/s}$, the actual narrow-link bandwidth of the

path. The minor nodes surrounding this peak are to some extent consistent between the two models.

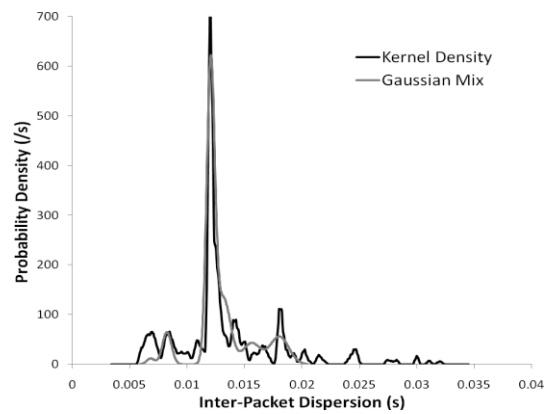
in the distribution there is a fairly close agreement between the peaks of the two models.



(a) Gaussian learning rate 0.01, classification criterion <2.5 standard deviations.



(b) Gaussian learning rate 0.01, classification criterion <1.5 standard deviations.



(c) Gaussian learning rate 0.05, classification criterion <2.5 standard deviations.

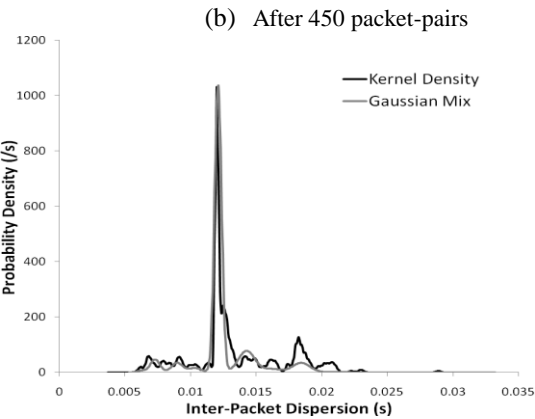
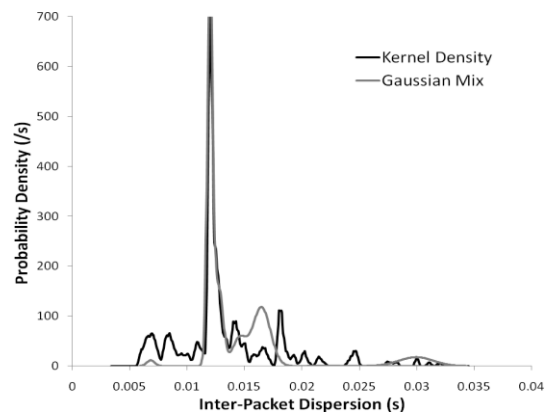


Figure 4. Dispersion probability density functions obtained using the Pareto cross-traffic at two instants during the experiment.

Figure 4 illustrates the adaptability of the two algorithms to the time-variations introduced by the Pareto cross-traffic. Figure 2(b) shows wide variations in dispersion between 1 and 200 packet-pairs, and a much more consistent dispersion (centered around the modal 12ms) leading up to 450 pairs. These differences are reflected in the dispersion profiles of Figure 4(a) and (b), particularly in the much more dominant primary peak in the latter.

Figure 5(a) shows an interesting phenomenon which was sometimes observed in the Gaussian-mixture distributions. (These particular results were obtained using Poisson cross-traffic, using a Gaussian learning rate of 0.01 and an inter-pair time of 10s.) While the primary “bandwidth” peak is identified correctly by the KDE, the Gaussian-mixture model curiously splits the peak into two near-identical components. However, at other points

Figure 5. Misclassification of the primary distribution mode as a double peak (a) removed by using a more stringent classification rule (b) and a faster learning rate (c).

To investigate this anomaly, further analysis was performed on the same data-set. Firstly a more stringent classification criterion was applied: Sample Δ_t was rejected from distribution i unless $|\Delta_t - \mu_{i,t}|/\sigma_{i,t}$ was less than 1.5. (Figure 5(b)) Secondly a faster learning rate was used: $\alpha=0.05$ instead of 0.01 (Figure 5(c)). The results show a single primary mode restored in both cases. It seems most probable that in Figure 5(a) the bandwidth-peak falls midway between - and within the common catchment area of - two neighboring modes, which have developed separately at the expense of the “true” node between them.

5. Conclusions

This paper presents a brief study of real-time mode-tracking using the Gaussian-mix model of Stauffer and Grimson [2] and an adaptation of the kernel-density method employed by Lei and Baker [3]. The two models produced broadly similar results, including the ability to track node activity in real time. However, the Gaussian model sometimes replaces the primary mode with a double peak, though this disappears when some of the model’s parameters are changed.

6. 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] 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.
- [3] Lai,K, Baker,M, “Measuring Bandwidth”, *Proc. IEEE INFOCOM*, pp.905-14, 1999.
- [4] Pitts,J.M. Schormans,J.A. “Introduction to IP and ATM Design and Performance”, Wiley, 2000, p.293.
- [5] Tunnicliffe,M.J., “Stochastic Analysis of the Packet-Pair Bandwidth Probing Event under Heterogeneous Cross-Traffic”, *Proc. 10th International Conference on Computer Modelling and Simulation*, IEEE Computer Society, 2008, pp.691-6.
- [6] Johnsson,A, Melander,B, Björkman,M, “Bandwidth Measurement in Wireless Networks”, *Proc. Mediterranean Ad Hoc Networking Workshop*, Porquerolles, France, 2005.