Bottleneck Bandwidth Estimation Using Frequency Analysis

T. Edwan, B. Ghita, X. Wang
School of Computing, Communications and Electronics,
University of Plymouth, UK.
e-mail: edwantalal@hotmail.com, {bghita, xwang}@plymouth.ac.uk

***Abstract*—This paper presents a new passive technique for estimating the bottleneck bandwidth based on transferring the Gaussian kernel density estimation of the packets inter arrival times to the frequency domain. The resulting spectrum contains information about the transmission time of the bottleneck link and can reveal information about multiple bottlenecks if they exist along the end-to-end path. The advantage of the technique is that it provides a model that can be manipulated by the digital signal processing methods and, unlike prior work in the area, it relies less on statistical methods. The proposed technique was validated using the `ns2` network simulator on several topologies and traffic sources. Further experiments were conducted to test the strength of the patterns between flows that share a bottleneck by applying K-means algorithm to cluster the average packet inter-arrival times of these flows. The paper also presents a set of results from real traffic experiments conducted in order to infer both the bottleneck bandwidth and the capacity of the path using a passive approach.**

I. INTRODUCTION

Due to the major expansion of the Internet infrastructure, users currently witness a very wide range of access/aggregation/core network speeds. In this diverse environment, an increasing amount of ongoing research focuses on bandwidth estimation. A large number of studies were performed in the past few years, summarised in (CAIDA 2007) in order to find an adequate and accurate method to estimate the bandwidth and define a terminology to give a better interpretation of what is measurable and, at the same time, serve as a base for future work. In parallel, a large number of associated tools were released (Cottrell 2007), starting with `pathchar` (Jacobson 1997), followed by `Clink` (Downey 1999), `Nettimer`(Lai and Baker 2001), `pchar` (Mah 2005), and `pathneck` (Hu et al 2005). The necessity of an adequate technique for measuring the bandwidth can be seen from two different perspectives. Firstly, from the applications and protocols point of view, whether involved in file transfer, content delivery, or real-time streaming media, they require accurate measurement of the bandwidth (Kiwior et al. 2004). This will become particularly necessary in the future, with the increased usage of multimedia applications, typically unaware of the available(or bottleneck) bandwidth. Knowledge of the total and available bandwidth along a path will allow an application to avoid bandwidth under-/over-provisioning by adapting the size and quality of its content (Claffy and Dovrolis 2004). Secondly, network operators are also concerned with traffic engineering, routing, network capacity, and network troubleshooting issues, in addition to other issues regarding the verification of Service Level Agreements and Quality of service(Harfoush et al. 2003).

There are two well known approaches for measuring the bandwidth: hop-by-hop, which is rather inefficient and bandwidth-intensive, or end-to-end, involving only two nodes acting as sender and receiver. The end-to-end approach can be either active (intrusive) or passive (non-intrusive). The former alternative is perceived to have inherent problems, relating to the fact that it requires packet injection, which will compete with existing traffic and might add to existing congestion. Also, the method is considered inaccurate; due to the dynamic behaviour of the Internet, the measurements will give an indication of the bandwidth just over a the measurement time interval. On the other hand, non-intrusive (passive) methods have less measurement overhead, as they depend on capturing existing traffic in the path of interest and try to estimate the bandwidth by inferring traffic patterns. Passive techniques can therefore be used to analyse large amounts of captured traffic and observe the trends and the evolution of bandwidth. In addition, passive techniques fit large scale mesh-network traffic measurements better than active methods that suffer from probing overhead (Katti et al. 2004).

The paper starts in section § II with an overview of the concept of bottleneck bandwidth and what should be measured; the discussion then focuses in § III on the passive approach and the proposed technique used for estimating of bottleneck bandwidth. Section § IV then describes the validation experiments performed. The attempts to infer the bottleneck bandwidth and the capacity of the path using a passive approach in a real traffic environment are presented in section § V. Finally, the paper concludes in § VI - § VII by discussing the limitations of the new technique, how it can be improved in future and an overall conclusion of the study.

II. BOTTLENECK LINK AND BANDWIDTH

A generic definition of the bottleneck bandwidth of a link is the maximum transmission rate that could be achieved between two hosts at the endpoints of a given path in the absence of any competing traffic.(Harfoush et al. 2003), Or the ideal bandwidth of the slowest link on the route between two hosts (Thepvilojanapong et al. 2002). Above definitions describe the ideal case, as a path will rarely, if ever, be free of traffic. A more practical definition of the bandwidth for a path will

1095-2055/07/$25.00 ©2007 IEEE

consider an element of queuing, not necessarily on the link with the minimum capacity for an end-to-end network path (Katabi and Blake 2001). In fact a *bottleneck* occurs when a congestion occurs either due to uneven connectivity or when arrival rates are higher than the output capacity (Stevens 1994). In both cases the queue will build up, as illustrated in Fig 1. If the path is symmetric **R-1** and **R-3** will be the same router

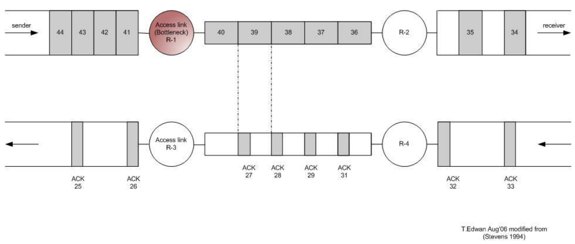


Fig. 1: Congestion case.

as are **R-2** and **R-4**. If we a assume that the packets are arriving to **R-2** from a WAN and the router will then pass them to a LAN, the packets will maintain the same spacing as they did on the WAN on the left of **R-2**. In the same manner the spacing of the ACKs on their way back is the same as the spacing of the slowest link in the path. Virtually all current bandwidth inference methods are based on this packet inter-arrival spacing when estimating the capacity of the link.

III. Passive Approach

The proposed approach of this study relates to the automation of equally spaced gaps detection mentioned in (Katti et al. 2004). These equally spaced gaps appear in the probability density function of the packets inter arrival times and they are multiples of the transmission time (the time to transmit one packet) of the bottleneck link. Rather than using the pdf of the equally spaced gaps (as in `multiQ` (Katti et al. 2004)), the proposed method performs Gaussian kernel estimation for the packets inter arrival times on the flows of interest. The result can then be transformed to the frequency domain in order to detect the repetition of the mode spikes, which will map to the transmission time of the packets on the bottleneck link. Note that, theoretically, more patterns (frequencies) can also be detected[1]. Before applying this approach and in an attempt to study how the packet inter arrival times for certain flows behave in the existence of cross traffic, K-means clustering was applied to cluster the average inter arrival time of each pair in a group of TCP flows available at the receiver.

A. Classification of flows using K-means

Within the unsupervised artificial neural networks area, algorithms like k-means are typically used for clustering. Clustering here means to group the objects based on a certain feature into a number of groups by minimising the sum of squares of distances between data and the corresponding cluster centroid. As a result, the data is classified into a K-clusters, with all the data in one cluster being very similar (relatively close in values).

The preferred procedure for this study was to start by generating all non repetitive combinations of a group of flows available at the receiver, where a flow is defined by the (`src IP`, `src port`, `dst IP`, `dst port`) quadruple (Lai and Baker 2001). The number of non repetitive combinations can be calculated from the following equation:

$$\binom{n}{k} = \frac{n!}{k!(n-k)!} \tag{1}$$

where : n is the number of flows and $k = 2$ (pairs of flows). After that the average inter arrival time for each pair of flows was calculated, then the k-means algorithm was applied to cluster the samples and minimise the sum of squares (or the Euclidean distance) within a cluster, according to the following equation (Webb 2002) :

$$S_W = \frac{1}{n} \sum_{j=1}^{g} \sum_{n}^{i=1} z_{ji}(x_i - m_j)(x_i - m_j)^T \tag{2}$$

Several clusters alternatives where evaluated to investigate how strong is the relation between the flows that do share a bottleneck. The aim was to see if it is possible to detect a shared bottleneck directly from packet inter arrival times.

B. Estimating bottleneck bandwidth

The technique for estimating the bottleneck bandwidth proposed in this paper is based on transferring the Gaussian kernel density estimation of the packet inter arrival times to the frequency domain. Kernel density estimation is a standard technique for constructing an estimate of a probability density function from measurements of the random variable. In fact kernel estimators are an extension to histograms designed to overcome their granularity issues. In a histogram, it is important to consider the width of the bins (equal sub-intervals in which the whole data interval is divided) and the end points of the bins (where each of the bins start). Regardless of the choice, histograms are not smooth and they depend on the width of the bins and the end points of the bins. These problems can be alleviated by using kernel density estimators which will provide the required smoothing. (Mishra 2006). KDE is expresses mathematically as :

$$f_x = \frac{1}{n} \sum_{i=1}^{n} k\left(\frac{x - x_i}{h}\right) \tag{3}$$

where K is the kernel function. Although there are several choices for the kernel function, they typically have $\int K(t)dt = 1$ and have peaks at the centre (at each point). Because the points here represent the modes, it was decided to adjust the kernel function so that the overall graph will look very similar to a sinusoidal plot. By doing this it is possible to transfer the new graph to the frequency domain by performing Fourier transforms and still obtain all the frequencies in our "signal" in a manner reassembling the ordinary analysis of electrical

[1]Usually the most congested bottleneck dominates the pattern.

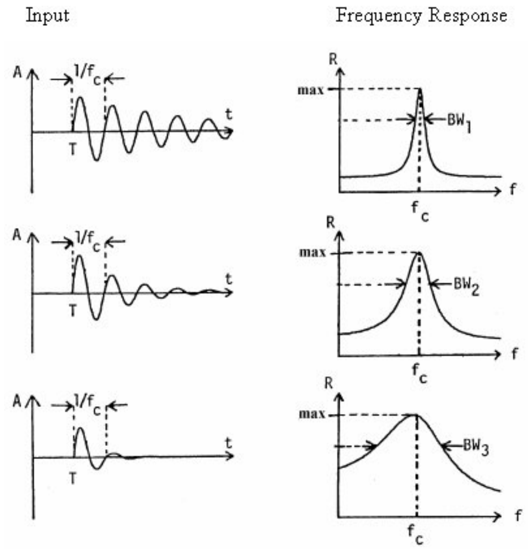


Fig. 2: Low, medium and high sinusoidal damped functions and their frequency responses

signals. Any repeated pattern in the original graph will give a frequency peak in the new graph, from which the transmission time of the bottleneck link can be determined. The Gaussian kernel function was used; the bandwidth h should be adjusted for not to overestimate the density, we can look at this as a resolution problem : adjust the variable h until you get a peak in the frequency domain (if there is a bottleneck).

The final result after applying the KDE on inter arrival times would be equally separated modes that usually decrease in amplitude as we move far from the global mode; this would be similar to a damped sinusoidal plot, as it can be seen in Figure 2. The assumption was that the damping frequency (the peak in the frequency domain) will directly map to the transmission time of the packets, giving an indication of a bottleneck and an estimate of its bandwidth. The damped sinusoidal is expressed as :

$$f(x) = \exp(-ax)[sin(\omega_b x)]u(x), a > 0. \tag{4}$$

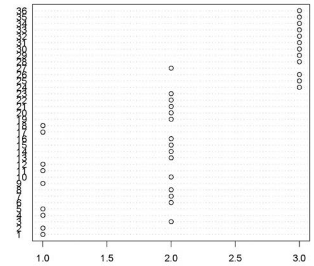
$$F(\omega) = \frac{\omega_b}{(a + j\omega)^2 + \omega_b^2}, a > 0. \tag{5}$$

The motivation for this is to propose a model for the passive approach that can be ported to any digital signal processing tool for better analysis (such as filtering techniques) and, from an implementation point of view, it can be easily implemented as a hardware device (bandwidth analyser).

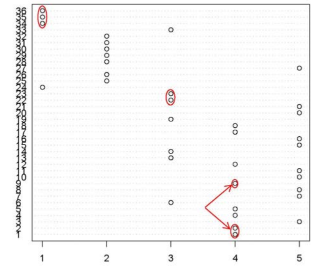
IV. Validation

A. K-means

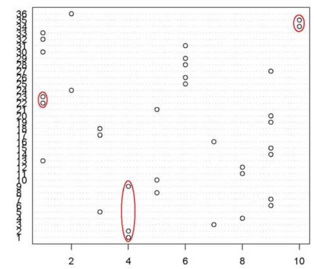
A simple bottleneck scenario consisted of nine TCP sources S_1-S_9 were connected to one sink through three links L_1, L_2, L_3 as follows:$S_1 - S_3$ connected to $L1$ (1.5Mbps), $S_4 - S_6$ connected to $L2$(2Mbps), and $S_6 - S_9$ connected to $L3$ (2.5Mbps). Each of the nine sources was connected to its corresponding gateway thorugh a 10Mbps link and



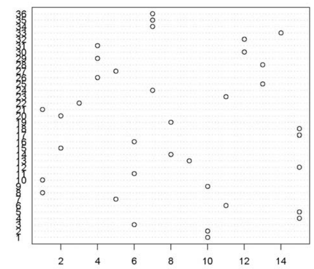
(a) 3-clusters.



(b) 5-clusters.



(c) 10-clusters.



(d) 15-clusters.

Fig. 3: Different k-means clusters. {x-axis : number of clusters, y-axis : flow group number} Note that we have 36 groups, each contains two flows, this can be calculated from equation-1

each three share a bottleneck. The simulation time was 22 seconds. The flows were then grouped by calculating all possible non-repetitive combinations of the available flows, according to equation 1, for which the K-means clusters were calculated. Figure 3b and Figure 3c show k-means clustering results for 5 and 10 clusters. As it can be seen from these graphs, some patterns persist but several are lost especially when increasing the number of clusters. For example when the pairs were K-grouped into five clusters, groups $G_{34} - [S_7, S_8]$, $G_35 - [S_7, S_9]$, and $G_{36}[S_8, S_9]$ did cluster correctly but with an additional group, $G_{24} - [S_4, S_7]$ that should not be in this cluster. Note that S_4 belongs to the 2Mbps bottleneck and S_7 belongs to the 2.5Mbps bottleneck, while the 1.5Mbps bottleneck did not contribute to the error. Also, groups $G_1 - [S_1, S_2]$, $G_2 - [S_1, S_3]$, and $G_9 - [S_2, S_3]$ all share the 1.5Mbps bottleneck and they have a strong pattern that persist even when the number of clusters is increased to 15 and 20 (not shown in the picture). On the other hand $G_{22}[S_4, S_5], G23[S_4, S_6]$ and $G27[S_5, S_6]$ (that belong to the middle bottleneck) were severely affected by the flows from the other two bottleneck as they are both close to its data rate. From this discussion it can be observed that the closer the bottlenecks bandwidth values are, the more difficult to separate their flows and thus the more difficult to detect them. The second observation is that at higher bandwidth values, the clustering error increases, while lower bandwidth allows

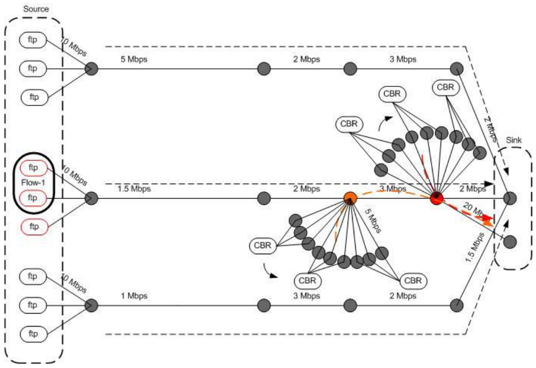


Fig. 4: Simulation topology

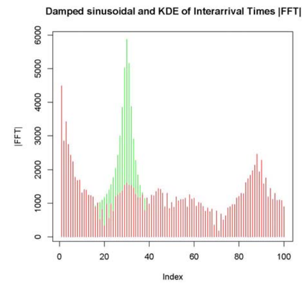
strong patterns to persist

B. Estimating bottleneck bandwidth in frequency domain

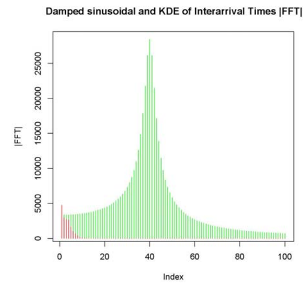
To validate the proposed technique of converting the kernel density "signal"[2] to the frequency domain, the simulation scenario in Figure 4 was constructed: nine TCP sources connected through 10Mbps to three links with different bandwidth values. Further, each path included five hops including a combination of bottleneck links and cross-traffic sources. The middle path contains 4 links from the gateway to the sink(which is the observer's point, in this case the receiver), 1.5Mbps, 2Mbps, 3Mbps, 2Mbps. At hop 3 and 4 there are 9 cross traffic sources for each, they are using the same path as our packets (path persistent) and are connected to a different sink by 20Mbps link. A mixture of constant bit rate and Pareto pdf sources were used for cross-traffic. At the end of the simulation, packets were extracted from a group of two flows (Flow-1 in Figure 4) then the packet inter arrival times were computed, followed by the kernel density estimation. The results were then transferred to the frequency domain by applying Fast Fourier Transforms on the resultant "signal". Figures 5a - 5c depicts some of the results when the kernel bandwidth was changed. For the sake of comparison, a damped sinusoidal function was also plotted in the same KDE plot because, as mentioned in section § III-B, the assumption was that there might be a correlation between this function and the KDE. The period of this function was chosen as the transmission time of the bottleneck (3Mbps in this case) for a 1040 bytes packet [3], which is $T = 2.773ms \rightarrow f_b \approx 360.58Hz \rightarrow \omega_b = 2\pi(360.58) = 2265.6$ In the first case in Figure 5a a small kernel bandwidth was chosen which results in an under smoothed high noise signal that can be seen as a wide spectrum in the figure. In comparison, when the kernel bandwidth was increased by nearly the factor of 10, the signal was lost except for the low frequency envelope. Finally, when a kernel bandwidth was chosen between the two extremes, analysis produced a smoothed signal, highly correlated to the reference signal. Note in Figure 5c the first peak and even

[2]The term signal is used as an analogy with the electrical signal

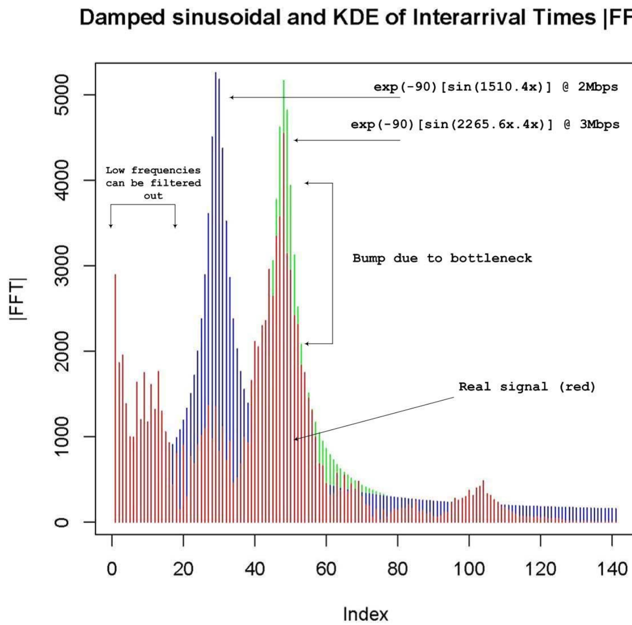
[3]The 1000 bytes of data and 40 bytes of TCP/IP headers



(a) bw = 0.0001.



(b) bw = 0.007.

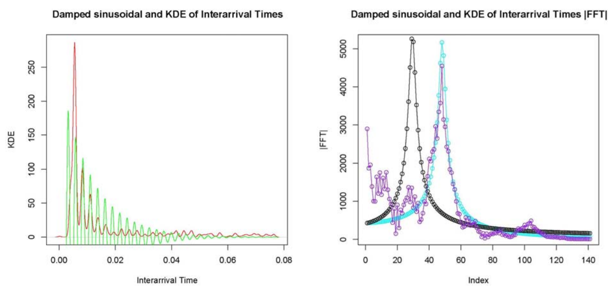


(c) bw = 0.0008.

Fig. 5: Effect of changing the kernel bandwidth : Under smoothed(wide frequency spectrum), Over smoothed(loss of signal except the low frequency envelop) and normally smoothed (the signal has a bump at the bottleneck frequency[reciprocal of transmission time]).

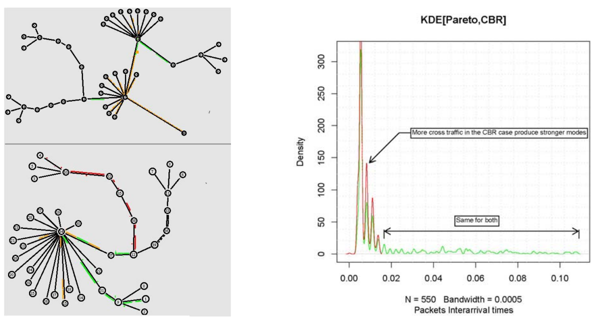
the tail of the rest of the plot matches the reference signal. Also note the persistent low frequency pattern in all graphs that corresponds to the envelope of the signal; this can be removed by applying filtering techniques as it does not contain any significant information in this case.

Several alternatives were used to benchmark the new technique, for example, it was observed that interchanging Pareto and constant bit rate cross traffic has no effect on the overall pattern; the only observable difference is that the cross traffic is less intense than in the constant bit rate case and thus the modes still exist in the same positions, with their same separation but with smaller amplitudes. This can be clearly seen Figure 7b, which shows the results of the Pareto sources being applied to model (interactive) Internet traffic flows. This heavy tailed pdf can be expressed as: $f(x) = ax^{1-a}, a >$



(a) Correlation between KDE and damped sinusoidal
(b) Frequency response for KDE and damped sinusoidal

Fig. 6: New technique for estimating bottleneck bandwidth in frequency domain



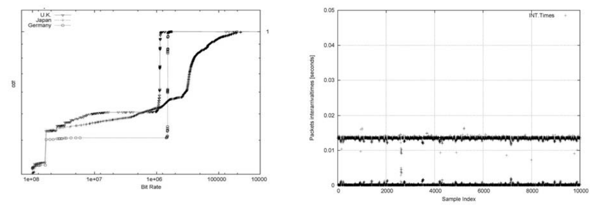
(a) ns2 simulation topology
(b) KDE of inter arrival times for Pareto(green) and CBR(red) sources

Fig. 7: Effect of applying Pareto instead of CBR

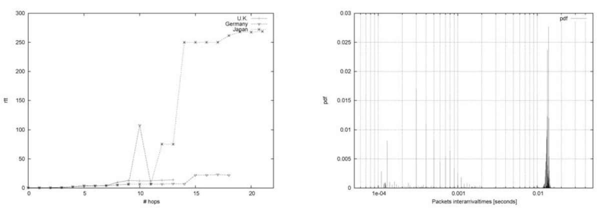
$0, 1 \leq x \leq \infty$, where a is the shape parameter; when $a \to 1$, it gives rise to self-similar traffic, whereas $a \to 2$ will lead to fractal properties similar with exponential traffic. To balance these extreme cases, a was set to 1.5.

V. PDFs and Real traffic

In order to see how much information can a pdf of a packet inter arrival times contains about the path capacity and the bottleneck bandwidth, several experiments were conducted in a real environment. The UoP network was used to download a large file (116.5 MB) from three different repositories : Keihanna (Japan), Duesseldorf (Germany), and Kent (UK). 10,000 packets were captured during the transfer from each destination to a trace file using `tcpdump` (tcpdump 2007). The packets were then analysed, pdfs and cdfs were calculated and plotted on the same graph. The following tools were also used for comparative analysis: `Clink`, `pchar`, `pathchar`, `pathneck`. By examining Figure 8a it can be seen that, for the Duesseldorf trace, most of the packets arrived spaced by a time nearly equal to the 1.25 μs, giving a rate of 800Kbps. This should give an indication about the path capacity (the same applies for the other plots). Inferring the path capacity is not a straightforward task. In the simplest approximation, bandwidth can be calculated from the minimum inter arrival times for back-to-back packets. A more advanced alternative



(a) CDF of the three traces
(b) Packet inter arrival times



(c) RTT of the three traces
(d) PDF.

Fig. 8: Real traffic analysis

is to extract the value from the global mode in the distribution of packet inter arrival times. For maximum accuracy, however, the analysis should include an examination of the pdf and the location of the bumps as an alternative approach (Katabi and Blake 2001), however the global mode here is very clear (no noisy samples) and most of the packets arrived at this rate.

The pdf plot must be considered in order to infer the bottleneck. An important point to note here is that the network interface card used throughout the real network experiments was a 100Mbps Fast Ethernet, connected to the University of Plymouth (UoP) network; further, the UoP infrastructure includes a 1Gbps backbone and is connected to the Internet via a 620Mbps link, physically implemented with four 155Mbps STM-1 lines. A per-flow traffic splitting policy was assumed, so the observed bottleneck[4] speed would be 155Mbps rather than 620Mbps if significant queuing occurs on that link. For the Kent download, the pdfs were equally separated by 0.1 ms; assuming a packet size of 1500 bytes, the resulting bottleneck bandwidth estimate is 120 Mbps. See Figure 8d Note that the spacing between the modes was considered and not the packets inter arrival times which in our case cannot exceed the 100 Mbps network card rate.

Comparing the results with other tools, and by looking at the first hop estimation, both `Clink` and `pchar` estimate the total lines bandwidth 620 Mbps, in addition to that the hop queuing was at its maximum at the first hops and decreased to zero at the third hop which means that access links are more likely to have queuing. `pathneck` was used to determine the choke points along the path. When considering the download from Keihanna, it was found that the packet inter arrival times

[4]It was found that bottleneck usually occur at the access links because of the significant queuing, this can be seen from the results of pathneck were all choke points occur at the last hops except for one case.

give a high fluctuating pattern, as seen from the cdf and RTT graphs. In the cdf graph it is still possible to estimate the path capacity as the global mode.

VI. Limitations

Applying the K-means approach to group flows that share a bottleneck was just to sense how strong is the relation between these flows without using the *entropy* as a decision rule as mentioned in (Katabi and Blake 2001). However, trying to cluster the average inter arrival times for each pair of flows did reveal a strong relationship for the analysed cases. In fact, some flows could still be grouped in one cluster even when the number of clusters was increased to ten. The analysis was less successful for the 2Mbps bottleneck; the 1.5Mbps and 2.5Mbps values experienced by the concurrent flows appeared to be too close (and symmetric) to the 2Mbps value to cluster correctly.

The results also indicated that, if additional perturbing modes exist between the equally spaced modes, the pattern will be disturbed, leading to errors (typically overestimation) in the frequency measurement. Finally, timestamp-related errors (clock resolution, drift, and skew) can introduce further errors in the measurement.

VII. Conclusion

Bottleneck bandwidth inference represented one of the topical research areas over recent years. A number of end-to-end measurement methods, both active and passive were proposed by prior studies. This paper proposes a novel passive technique to estimate bottleneck bandwidth by applying clustering and digital signal processing methods to the packet inter-arrival times. The translation of the analysis environment to the frequency domain opens the avenue for further digital signal processing techniques to be applied. Both simulation and real network tests produced promising results, indicating that the method can accurately cluster flows following similar paths and determine the bottleneck bandwidth spectrum. Still, the proposed model does exhibit certain limitations and would benefit from further testing in a real traffic environment.

VIII. Future work

This proposed technique lends itself to extension in several directions. One possible area for further research is to consider more complex scenarios and to test the new technique in real traffic environments for longer periods. Another issue is to consider the effects of packet loss and delay, which may impact on the estimation results. Using the proposed technique to produce long-term "traffic spectrum" is essential for studying bottleneck bandwidth trends and network growth. It is recognised that implementing the method in real-time is still a challenging issue, especially with the necessary "tuning" needed for the bandwidth of the kernel density which should be done by the user during the measurement. It is envisaged that, in order to produce actual estimates from the resulting graphs, the method could be further linked to an artificial neural network to provide the interpretation of the obtained distributions.

Another potential direction for further research should consider the translation of the KDE to the frequency domain. Although the method yielded good results in a controlled simulation environment, it would benefit from more testing in large scale environments. In addition, the choice of the kernel function could potentially lead to better results. As a possible refinement, making the bandwidth of the kernel function variable may change the original "signal" dramatically and this may have a a great impact on the frequency response, although it will still indicate if there is a bottleneck or not based on the existence of the peak this may change the estimated rate.

IX. References

- Claffy, K., Dovrolis, C. (2004),'Bandwidth Estimation: Measurement Methodologies and Applications' http://www.scidac.org/March2004/ascr_net_2.html.
- CAIDA(2007), Cooperactive Association for Internet Data Analysis.[Online]http://www.caida.org/.
- Cottrell, L., (2007), 'Network Monitoring Tools', http://www.slac.stanford.edu/xorg/nmtf/nmtf-tools.html
- Downey, A. B., (1999), 'Clink: a tool for estimating Internet link characteristics', :http://allendowney.com/research/clink/
- Harfoush, K., Bestavros, A., Byers, J.(2003)'Measuring Bottleneck Bandwidth of Targeted Path Segments', Technical Report: 2001-016, Boston University, 2001.
- Hu, N., Steenkiste, P., Li, ,E.L., Mao, Z.M. and Wang, J. (2005), 'A measurement study of Internet bottlenecks', Proceedings of INFOCOM 2005, Pittsburg, US, March 2005
- Jacobson, V. (1997), *pathchar*, :ftp://ftp.ee.lbl.gov/pathchar
- Katabi, D., Blake, C.(2001)'Inferring Congestion Sharing and Path Characteristics from Packet Interarrival Times', MIT/LCS Technical Memo 828, June 2001
- Katti, S., Katabi, D., Blake, C., Kohler, E. and Strauss, J.(2004)'M&M: A Passive Toolkit for Measuring Tracking, and Correlating Path Characteristics', Technical Report 945, MIT CSAIL, 2004. .
- Kiwior, D., Kingston, J., Spratt, A.(2004)'PATHMON, A Methodology for Determining Available Bandwidth over an Unknown Network', IEEE/Sarnoff Symposium on Advances in Wired and Wireless Communication, 26-27 Apr 2004, pp 27 - 30
- Lai, K., Baker, M.(2001)'Nettimer: A Tool for Measuring Bottleneck Link Bandwidth', Proceedings of the USENIX Symposium on Internet Technologies and Systems, March 2001
- Mah, B.A., (2005), *Pchar*, http://www.kitchenlab.org/www/bmah/Software/pchar/
- Mishra (2006), 'Kernel Density Estimators', tutorial, http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL _COPIES/AV0405/MISHRA/kde.html.
- Stevens, W.R.(1994)*TCP/IP Illustrated, Volume 1*, Addison Wesley, ISBN 0201633469.
- tcpdump (2007), *tcpdump*,protocol packet capture and dumper program,http://www.tcpdump.org/
- ns2 (2007), The *ns2* Network Simulator,http://www.isi.edu/nsnam/ns/
- Thepvilojanapong, N., Tobe, Y. and Sezaki, K.(2002)'One-way Delay Measurement and Bottleneck Bandwidth Estimation', The Tenth DPS Workshop, IPSJ Symposium Series, 2002
- Webb, A.(2002)*Statistical Pattern Recognition* 2nd Edition, John Wiley& Sons Ltd, The Atrium, Southern Gate, Chichester, West Sussex PO198SQ, England.