Analysis of End-to-End Techniques for Bottleneck Bandwidth & Path Capacity Estimation

T.Edwan, B.Ghita and X.Wang

Network Research Group, University of Plymouth, United Kingdom
e-mail: info@network-research-group.org

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 resultant 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 which is different than the previous work done in this field that relies too much on statistical methods. The new technique was validated using the ns2 network simulator, several topologies and traffic sources were tried. The post-simulation analysis was done by the programs and Unix/Linux Bourne (and Bash) Shell scripts developed for this purpose. In addition to that, some 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 packets inter arrival times of these flows. The document also presents the results of real traffic experiments conducted in an attempt to infer both the bottleneck bandwidth and the capacity of the path using a passive approach.

1. Introduction

With the evolution of the Internet, and the maturity of its infrastructure, along with the increase of network traffic, efforts now are redirected to services associated with it and how it can be improved. One major parameter that has a direct effect on these services is the bandwidth. A lot of effort has been done in the past few years by several researchers trying to find an adequate way to estimate the bandwidth and define several terminologies that will give better understanding of what exactly to measure and at the same time serve as a base for future work.

The necessity of an adequate technique for measuring the bandwidth can be seen from two different perspectives. Firstly, the applications and protocols, whether they are involved in file transfer and content delivery or they are involved in real-time streaming media, they require accurate measurement of the bandwidth, as shown by Kiwior et al (2004).This is crucial especially in future when we will be dealing with multimedia applications, for example an application unaware about the available(or bottleneck) bandwidth can stream a 5GB video over a19.2Kbps cellular data link or send a text-only version of a web site over a 100Mbps link. Knowledge of the bandwidth along a path allows an application to avoid such mistakes by adapting the size and quality of its content, (Fox 1996), or by choosing a web server or proxy with higher bandwidth than its replicas, (Stemm 1999).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(QoS) (Harfoush et al 2003).

There are two well known approaches for measuring the bandwidth, one is to measure it hop-by-hop which is considered to be inefficient, and the other is end-to-end which requires only two nodes (the sender and the receiver). End-to-end approach in turn can be used in two different ways : active (intrusive) or passive (non-intrusive), the former is seen to have some problems, particularly it is slow and because of the packet injection to the network (that will compete with the original traffic and might also inject more load in to the network) is considered inaccurate, moreover the Internet traffic is not static, so the measurement using this way (active) will give an indication of the bandwidth just over a certain time interval which is in this case the measurement time interval. On the other hand non-intrusive (passive) methods have less measurement overhead. They basically depend on capturing existing traffic in the path of interest and try to estimate the bandwidth by inferring traffic patterns. Passive techniques can be used to analyse huge amount of traffic captured for several years, detect the trends and the evolution of bottlenecks. It is also believed, Katti et al. (2004), that passive techniques fit large scale network traffic measurements better than active methods that suffer from probing overhead, taking into account that we are always interested in real large scale network traffic as this is the case for the Internet.

First in section 0, the paper gives an idea about what is the bottleneck and what should be measured, then in section 0 it focuses on the passive approach and the new technique used in the estimation of bottleneck bandwidth. In section 0, a discussion of the experiments that were conducted to validate the new technique is provided. An example of inferring the bottleneck bandwidth and the capacity of the path using a passive approach in a real traffic environment is presented in section 46. Finally, the paper concludes with the limitations of the new technique (section 0), the alternatives for improvement (section 0) and an overall conclusion (section 0).

2. Bottleneck link and bandwidth

Some researchers define the bandwidth of a link as 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 it can be defined as the ideal bandwidth of the lowest bandwidth link on the route between two hosts, as shown by Thepvilojanapong et al (2002). However this seems to be ideal because we always have traffic along our path. A more practical definition of this link is the link that experiences a significant queuing, so it is the congested link and it is not necessary to be the link with the minimum capacity in an end-to-end network path (Katabi and Blake 2001). In fact a bottleneck occurs when a congestion occurs and this takes place when data arrives on a large capacity link (like a fast LAN) and gets send out to a smaller link (like a slower WAN) or when multiple inputs streams arrive at router whose output capacity is less than the sum of the inputs (Stevens 1994). At both cases the queue will build up. Figure 1 illustrates a typical scenario.

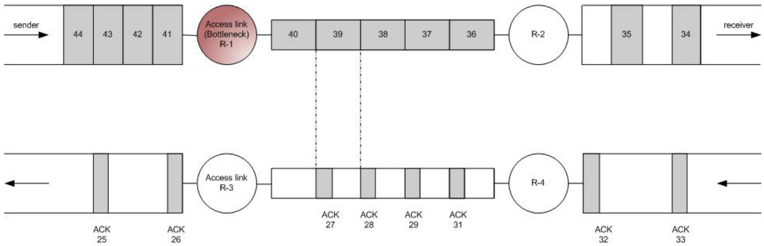


Figure 1: Congestion case

If the path is symmetric, $\mathbf{R_1}$ and $\mathbf{R_3}$ will be the same router as are $\mathbf{R_2}$ and $\mathbf{R_4}$. If we a assume that the packets are arriving to $\mathbf{R_2}$ 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 $\mathbf{R_2}$. 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. This seems to be logical but it is very important when trying to infer the graphs of the packet inter arrival times in the passive approaches.

3. Non-intrusive bottleneck bandwidth estimation

This section discusses the proposed approach, which relates to the automation of equally spaced gaps detection mentioned by Katti et al (2004). These equally spaced gaps are the located between the probability density function of the packets inter arrival times, and they are usually 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*, a passive tool developed at the M.I.T. (Katti et al 2004)), the method proposed by this study is to perform Gaussian kernel estimation on the packets inter arrival times of the flows of interest then transform the result 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 if there is more than one pattern (frequency) it can also be detected (usually the most congested bottleneck dominates the pattern). 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.

3.1 Classification of flows using K-means

In Pattern Recognition and unsupervised learning of neural networks-means, 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 and this is done by minimising the sum of squares of distances between data and the corresponding cluster centroid. As a result the data is classified into K-clusters. All the data in one cluster is very similar (relatively close in values). The procedure is to first take all non repetitive combinations of a group of flows available at the receiver, the flow here is defined as the IP address and port number. The number of non repetitive combinations can be calculated from the following equation:

$$\binom{n}{k} = \frac{n!}{K!\,(n-k)!} \quad (1)$$

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

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

Several clustering methods where applied to evaluate how strong is the relation between the flows that do share a bottleneck. The aim of this was to see if it is possible to detect a shared bottleneck directly from packet inter arrival times.

3.2 Novel Technique for estimating bottleneck bandwidth

The new technique for estimating the bottleneck bandwidth proposed in this paper is based on transferring the Gaussian Kernel Density Estimation of the packets 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 extension to histograms whose disadvantages provide the motivation for kernel estimators. 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). So the problems with histograms are that they 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. KDE is expresses mathematically as:

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

where K is the kernel function. There are various choices among the kernel function, however several of them 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 signals. Any repeated pattern in the original graph will give a frequency bump in the new graph from which the transmission time of the bottleneck link can be determined. The Gaussian kernel function was used (Cosine function may produce better results) and 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 bump in the frequency domain (if there is a bottleneck).

The final result after applying the KDE on inter arrival times would be an 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. The assumption was that the damping frequency (the bump in the frequency domain) will directly map to the transmission time of the packets, in other words it will give an indication if there is a bottleneck and an estimate of its bandwidth. The damped sinusoidal is expressed as:

$$f(x) = exp(-\alpha x)[sin(\omega_b x)]u(x) \Leftrightarrow F(\omega) = \frac{\omega_b}{(a + j\omega)^2 + \omega_b^2}, a > 0$$

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).

4. Validation

4. 1 K-means

A simple bottleneck simulation scenario was used: nine sources connected to one destination using transmission control protocol TCP through three bottlenecks *B1, B2, B3* with 1.5Mbps, 2Mbps, and 2.5Mbps respectively. Each of the nine sources is connected to its gateway by a 10Mbps link and each three share a bottleneck. Following the simulation, the flows were separated by first calculating all possible non-repetitive combinations according to equation (1), followed by calculation of the K-means clusters.

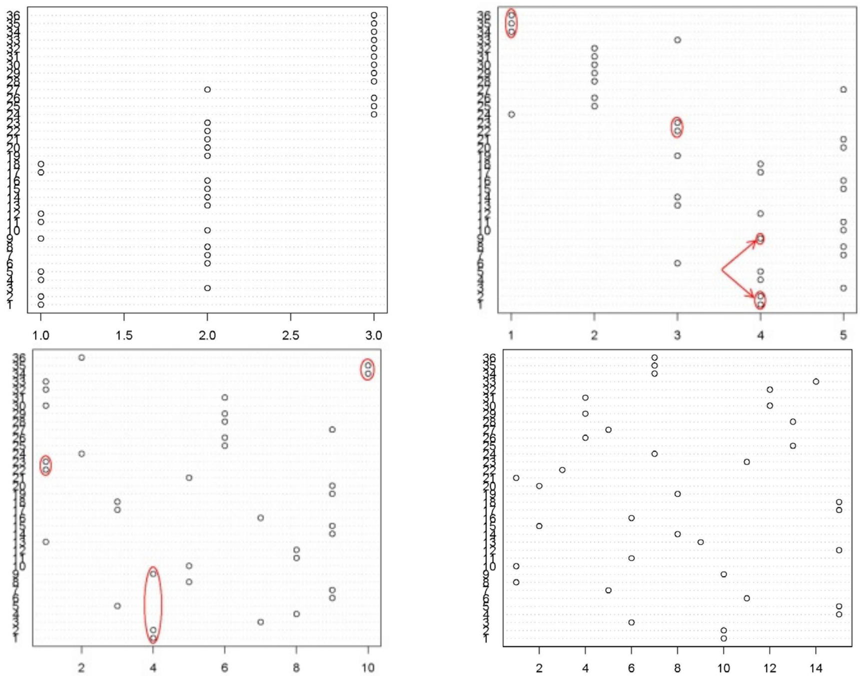


Figure 2 - Resulting clusters when using different K-means clustering sizes

As it can be seen from the above graphs, some patterns persist but a lot are lost especially when increasing the number of clusters. For example when the number of clusters is 5, groups 34-[7,8], 35-[7,9], 36-[8,9] did cluster correctly but with additional group 24-[4,7] that should not be in this cluster. Notice here that flow 4 belongs to the 2Mbps bottleneck and flow 7 belongs to the 2.5Mbps bottleneck and the first bottleneck (1.5Mbps) did not contribute to the error. Also Notice that groups 1-[1,2], 2-[1,3],9-[2,3] all share a bottleneck and they have a strong pattern that persist even when the number of clusters is increased to 15 clusters and 20 (not shown). Also notice that groups 1,2,9 all share the bottleneck with the least bandwidth (1.5 Mbps) compared with other bottlenecks (2 and 2.5 Mbps). On the other hand groups 22,23 and 27 (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 bandwidths are the more difficult to separate their flows and thus the more difficult to detect them. The second observation is at higher bandwidths the clustering error increase (failed to cluster correctly) while at lower bandwidths strong patterns persist, this seems to be logical since at higher bandwidths the transmission times are small and close to each other in contrast to lower bandwidth bottlenecks.

4.2 Estimating bottleneck bandwidth in frequency domain

To validate the new technique of converting our kernel density signal (the term signal is used as an analogy with the electrical signal) to the frequency domain, the simulation scenario in Figure 3 was constructed : 9 FTP sources, each three belongs to a path and they are connected to their gateways via 10Mbps links, each path contains 5 hops. 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, and 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. First constant bit rate cross traffic sources were used then Pareto pdf sources were tried. In some cases the sources were modified for not to generate traffic all at the same time, but rather they will fire their packets three by three.(first three start, then after a while the other three and so on). The simulation time was 22 seconds. At the end of the simulation, our program extracts packets from a group of two flows (Flow-1 in Figure 3) and computes the packets inter arrival times then the kernel density estimation and transfer the results to the frequency domain by applying Fast Fourier Transforms on the resultant ``signal''. Figure 4 (a-f) depicts some of the results when the kernel bandwidth was changed.

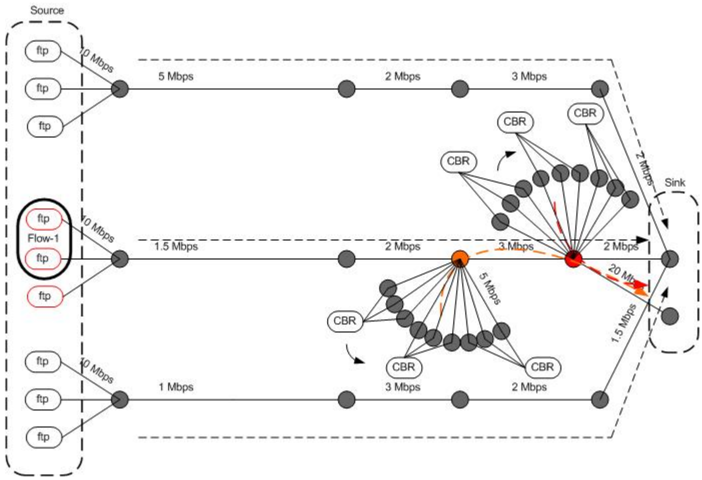
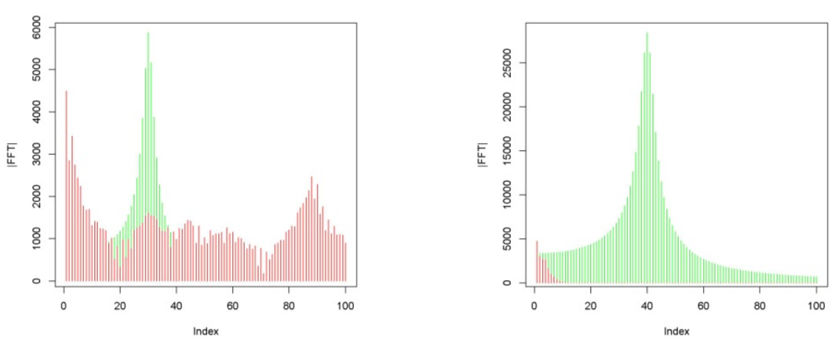


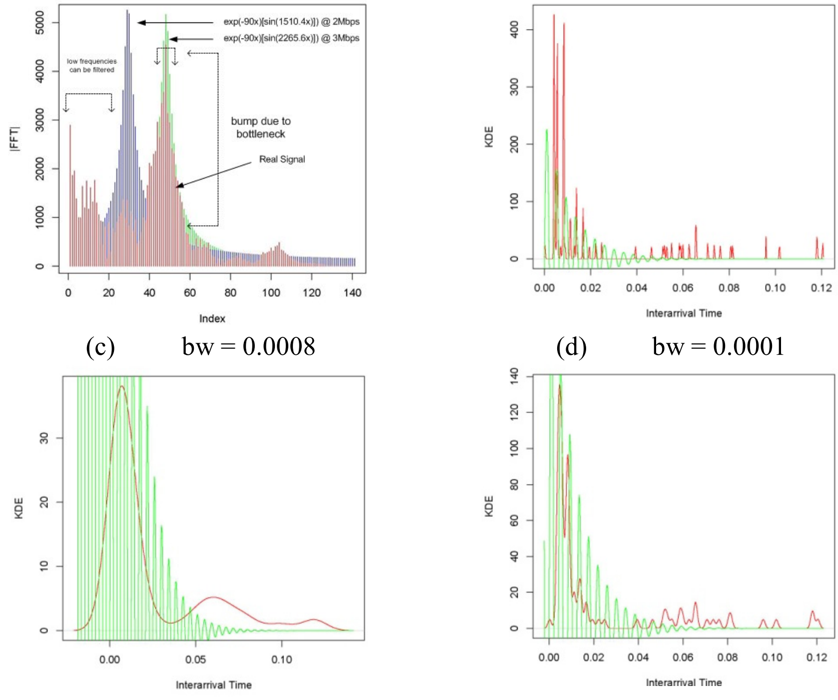
Figure 3 - Simulation topology

For the sake of comparison a damped sinusoidal function was also plotted in the same KDE plot, because as mentioned in section 0 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 (2Mbps in this case), this is the time to transmit a packet of 1040 byte. The packet size was 1000 byte and 40 bytes for the TCP-IP headers}, giving $T = 2.773\ ms \rightarrow f_b \approx 360.58Hz \rightarrow \omega_b = 2\pi(360.58) = 2265.6$. In the first case in Figure 4d a small kernel bandwidth was chosen which results in an under smoothed high noise signal which can be seen as a wide spectrum in the figure. While in the second case when the kernel bandwidth was increased by nearly the factor of 10, the signal was lost except for the low frequency envelop. Finally when a kernel bandwidth was chosen between the two extremes a normally smoothed signal was produced which is highly correlated to the reference signal. Notice in Figure 4 the first bump and even the tail of the rest of the plot nearly matches our reference signal. Also notice the persistent low frequency pattern in all graphs that corresponds to the envelop of our signal, this can be removed by applying filtering techniques as it does not contain any significant information in this case.



(a) bw = 0.0001

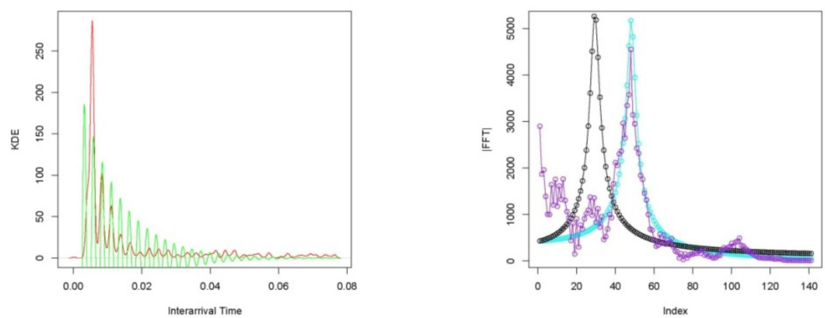
(b) bw = 0.007



(c) bw = 0.0008 (d) bw = 0.0001

(e) bw = 0.007 (f) bw = 0.0008

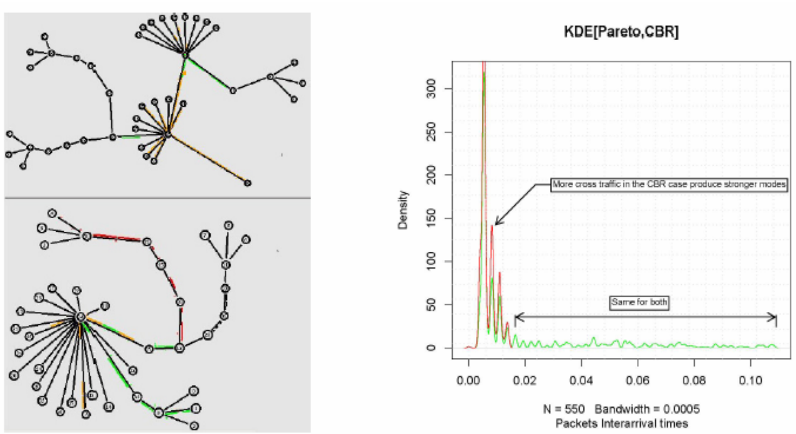
Figure 4 - Effect of changing the kernel bandwidth: Under smoothed (wide frequency spectrum), Over smoothed (loss of signal except the low frequency envelop) and normally smoothed (the signal has a bump at the bottleneck frequency reciprocal of transmission time)



(a) KDE and damped sinusoidal correlation (b) KDE and damped sinusoidal – frequency response

Figure 5 - New technique for estimating the bottleneck bandwidth in the frequency domain

Several alternatives were used to benchmark the proposed method, for example when applying Pareto cross traffic instead of constant bit rate it was found that this has no effect on the overall pattern, the only difference was that the cross traffic is less intense than in the constant bit rate case and thus the modes still exist in their same places with their same separation but with less amplitude. This can be clearly seen Figure Figure 6b. The Pareto pdf sources were applied to model (interactive) Internet traffic flows. And because the traffic is bursty, transmission only takes place during on periods. This heavy tailed pdf can be expressed as: $f(x) = ax^{1-a}, a > 0,1 \le x \le \infty$, where *a* is the shape parameter, when it is near to one it gives rise to self-similar traffic, whereas a near to two, has similar fractal properties to exponential traffic. The default 1.5 was used. In general terms, a heavy tailed distribution can give rise to very large file sizes.

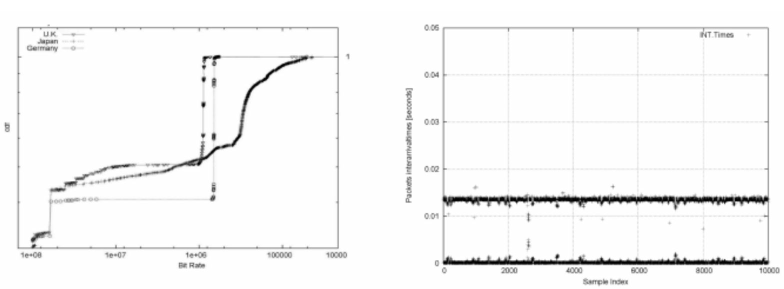


(a) ns2 simulation topology (b) KDE interarrival times for Pareto (green) and CBR (red) sources

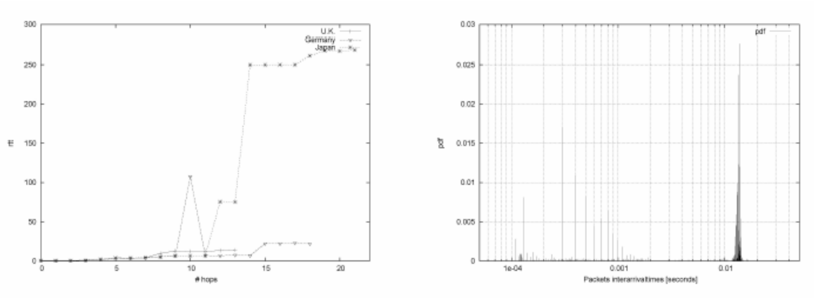
Figure 6 - Effect of applying Pareto vs. CBR

5. PDFs and Real traffic

In order to see how much information the pdf of the packet interarrival times contains on the path capacity and the bottleneck bandwidth, several experiments were conducted in real environment. The UoP network was used to download a file (116.5 MB) from different repositories: [Keihanna, Japan Asia], [Duesseldorf, Germany Europe], [Kent, UK Europe]. 10,000 packets were dumped to a trace file using *tcpdump* from each destination, the packets then were analysed, pdfs and cdfs were calculated and plotted at the same graph.



(a) CDF of the three traces (b) Packet interarrival times



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

Figure 7 - Real traffic analysis

By examining Figure 7a it can be seen that in the case of [Duesseldorf, Germany] most of the packets arrived spaced by a time nearly equal to the 1.25µs which gives a rate of 800Kbps. This should give an indication about the path capacity (the same applies for the other plots), although inferring the path capacity is not a simple task, some look at it as the minimum inter arrival times for back-to-back packets others see it as the global mode in the distribution of packet inter arrival times, while some researchers see that both can give wrong results and suggest 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.

Now to infer the bottleneck the pdf plot must be considered, An important point to note here is that the network interface card for the machine was 100Mbps Fast Ethernet, while the total link capacity was 620Mbps with a physical implementation of 4 STM-1 lines of 155Mbps each. And per-flow basis traffic splitting was assumed so the bottleneck speed would be 155Mbps rather than 620Mbps if there is a significant queuing at that link. It was found that bottleneck usually occur at the access links because of the significant queuing. Consider the [Kent, UK] case were the pdf are equally separated by 0.1 ms - assuming a packet size of 1500 bytes, this gives 120 Mbps as a bottleneck bandwidth estimate. Figure 7d, shows 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 path to [Keihanna, Japan Asia] it was found that the packets inter arrival times give a high fluctuating pattern this can be seen from the cdf and RTT graphs. In the cdf graph it is still possible to estimate the path capacity as the global mode.

6. 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 the strong relationship for limited cases with the bottleneck bandwidths used, in fact some flows can still be grouped in one cluster even when the number of clusters was increased to ten, but the rest did not show this relation this is because the averages of other flows are still close to the flows of our interest (that share a bottleneck) and fail to cluster correctly. On the other hand applying our new technique (the KDE in the frequency domain) in a controlled simulation environment did produce the expected results but still needs more testing in large scale environments. In addition to that the idea of using a certain kernel function like the Gaussian or the cosine with varying bandwidth is still foggy. Particularly making the bandwidth of the kernel function variable may change the original signal (the term signal is used as an analogy with the electrical signal) dramatically and this may have 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 bump this may change the estimated rate. One more thing to mention about this new technique is that if for any reason an additional perturbing modes exist between the equally spaced modes the pattern will be disturbed and this will contribute to the error in the frequency measurement. Particularly this will overestimate the frequency. Finally, other parameters like the clock resolution (*tcpdump* timestamps errors), delay can still have their effects on the measurement.

7. Future work

This project lends itself to extension in more than one direction; one of these directions is to consider more complex scenarios and testing the new technique in real traffic environments for longer periods. Another issue is to consider the effects of packet loss and delay which may have impacts on the estimation results. Studying the resultant traffic ``spectrum'' for longer periods that can be obtained by the new technique is an important issue that must be considered because this will reveal a lot of information about the bottleneck evolution and the network growth. Implementing the method in real-time is still a challenging issue especially with the necessary adjustments needed for the bandwidth of the kernel density which should be done by the user during the measurement. In fact, we must keep in mind that the output of this method is a visual information that need to be inferred logically by the user which means that the output cannot be provided directly to a non-human (non-intelligent) system (an application for example), thus involving artificial intelligence techniques (like Artificial Neural Networks) might be considered in the future.

8. Conclusion

A substantial amount of prior research focused on modelling the Internet traffic and understanding the parameters that affect its performance; amongst these parameters, bottleneck bandwidth is one of the critical ones. Two end-to-end approaches were used to estimate bottleneck bandwidth, one is active and the other is passive. This paper promotes the passive techniques by proposing a modification to a passive algorithm. The focus was on passive techniques mainly because of their less overhead of measurement and because they do not have an intrusive nature that may affect the measurement, in addition to their ability to analyse huge traffic captured for years which facilitate the understanding of the bottleneck evolution and the Internet growth. One important outcome of this project is the new model that transfers the analysis environment to the frequency domain were digital signal processing techniques can be widely applied and hardware implementation can be considered. In addition to that the new model provides a new perspective in the field of estimating the bottleneck bandwidth.

9. References

Chiu D.-M. and Jain R., 1989.Analysis of the Increase and Decrease Algorithms for Congestion Avoidance in Computer Networks, *Computer Networks and ISDN Systems*, 17(1):1-14, 1989.

Claffy, K., Dovrolis, C., 2004. Bandwidth Estimation: Measurement Methodologies and Applications, [Online] http://www.scidac.org/March2004/ascr_net_2.html [accessed 01 September 2006]

CAIDA, 2006. Cooperative Association for Internet Data Analysis [Online] http://www.caida.org [accessed 01 September 2006]

Harfoush, K., Bestavros, A., Byers, J., 2003. Measuring Bottleneck Bandwidth of Targeted Path Segments Proceedings of the IEEE INFOCOM 2003, San Francisco, CA, USA, March 30 - April 3, 2003.

Hu, N., Steenkiste, P., Li, E.L., Mao, Z.M. and Wang, J., 2004. Locating Internet Bottlenecks: Algorithms, Measurements, and Implications, *Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*, Oregon, USA

Hu, N., Steenkiste, P., Li, ,E.L., Mao, Z.M. and Wang, J., Pathneck, 2006. [Online] http://www.cs.cmu.edu/~hnn/pathneck/ [accessed 01 September 2006]

Katabi, D., Blake, C., 2001. Inferring Congestion Sharing and Path Characteristics from Packet Interarrival Times http://www.lcs.mit.edu/publications/ pubs/PDF/MIT-LCS-TR-828.PDF [accessed 01 September 2006]

Katti, S., Katabi, D., Blake, C., Kohler, E. and Strauss, J.(2004)`M&M: A Passive Toolkit for Measuring Tracking, and Correlating Path Characteristics, [Online] http://nms.lcs.mit.edu/~dina/MNM/mmdocs/paper.PDF, [accessed 01 September 2006]

Key, P., Massoulie, L., 2003. Fair Internet traffic integration: network flow models and analysis, Statistical Laboratory, University of Cambridge, Cambridge CB3 0WB, UK, http://www.statslab.cam.ac.uk/~frank/PAPERS/kmbk1.PDF, [accessed 01 September 2006]

Kiwior, D., Kingston, J., Spratt, A., 2004. PATHMON, A Methodology for Determining. Available Bandwidth over an Unknown Network, [Online] http://www.mitre.org/work/tech_papers/tech_papers_04/kiwior_pathmon/kiwior_pathmon.PDF [accessed 01 September 2006]

Stevens, W.R., 1994. *TCP/IP Illustrated, Volume 1*, Addison Wesley, ISBN 0201633469.

Teknomo, K., n.d., K-means Clustering Tutorials, [Online]
http://people.revoledu.com/kardi/tutorial/kMean/index.html, [accessed 01 September 2006]

tcpdump, [Online] http://www.tcpdump.org , [accessed 01 September 2006]

Thepvilojanapong, N., Tobe, Y. and Sezaki, K., 2002. One-way Delay Measurement and Bottleneck Bandwidth Estimation, *Joho Shori Gakkai Shinpojiumu Ronbunshu Journal*, vol. 2002, no. 15;pp. 39-44

Webb, A., 2002. *Statistical Pattern Recognition*, 2nd Edition, John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester, West Sussex, England