On the Reduction of the Available Bandwidth Estimation Error Through Clustering with K-means

Cesar D. Guerrero
Universidad Autonoma de Bucaramanga -UNAB
Avenida 42 No. 48 - 11
Bucaramanga - Colombia
Email: cguerrer@unab.edu.co

Dixon Salcedo Morillo
Universidad de la Costa - CUC
St. 58 # 55 - 66
Barranquilla - Colombia
Email: dsalcedo2@cuc.edu.co

***Abstract*—There are different tools to estimate the end to end available bandwidth (AB). These tools use techniques which send pairs of packets to the network and observe changes in dispersion or propagation delays to infer the value of the AB. Given the fractal nature of Internet traffic, these observations are prompt to errors affecting the accuracy of the estimation. This article presents the application of a clustering technique to reduce the estimation error due to wrong observations of the available bandwidth in the network. The clustering technique used is K-means which is applied to a tool called *Traceband* that is originally based on a Hidden Markov Model to perform the estimation. It is shown that using K-means in Traceband can improve its accuracy in 67.45% when the cross traffic is about 70% of the end-to-end capacity.**

I. Introduction

The estimation of the available bandwidth from end to end on a network connection can be used to evaluate or improve the performance of network applications [1] and [2]. Prasad et al. [3] indicate that the available bandwidth can be used to optimize the end to end network performance, routing in overlay networks, and Peer to Peer network file sharing. This value can be also used to define the Quality of Service of the network; to change the transmission rate according to the availability of the channel in a transport layer protocol; to perform traffic engineering in network management applications; to achieve the performance needed in content delivery networks, including multimedia (streaming) in overlapping virtual networks; among other functions of different network applications.

The available bandwidth of an end-to-end path is a time-varying metric related to the individual utilization of each link throughout the path. Defining T as the *averaging timescale* of the available bandwidth [3], the average utilization for a sample during T, is given by

$$u_i(t, t+T) = \frac{1}{T}\int_t^{t+T} u_i(s)ds \quad (1)$$

where $0 \leq u_i(t, t+T) \leq 1$. For a link *i* with capacity C_i, the AB of the link in the interval (*t,t+T*) can be defined as the average non-utilized capacity during the time T. That is,

$$AB_i(t, t+T) = C_i[1 - u_i(t, t+T)] \quad (2)$$

978-1-4673-5080-8/12/$31.00 © 2012 IEEE

For an end-to-end path with *H* hops, the available bandwidth is given by the link with minimum non-utilized capacity in the path. That is, $AB(t, t+T) = min_{i=1..H} AB_i(t, t+T)$. In the literature, the link with the minimum capacity is called the *narrow link* and the link with the minimum available bandwidth is called the *tight link*, which is considered the bottleneck of the path and the link that determines the end-to-end available bandwidth.

Monitoring and estimation of the available bandwidth has grown from a regular activity of a network administrator (expert or amateur) to a more complex research area where mathematical models or applied statistics are implemented in estimations tools, techniques and algorithms. Two main available bandwidth estimation approaches have been reported. The first approach is called the *Probe Gap Model* (PGM) which bases the estimation on the gap dispersion between two consecutive probing packets at the receiver. That dispersion is used to estimate the amount of cross-traffic in the tight link during T which is subtracted from the Capacity to estimate the AB in the path. Examples of tools in this category are *Spruce* [4], *Delphi* [5], *IGI* [6], and *Traceband* [7]. The second approach called *Probe Rate Model* (PRM) is based on the idea of *induced congestion*, in which the turning point (available bandwidth) is determined by the variation in the probing packet rate from sender to receiver. *Pathload* [8], *TOPP* [9], and *Pathchirp* [10] are examples of tools utilizing this approach. Each tool has trade offs in terms of the estimation accuracy, convergence time and intrusiveness of the tool to perform the estimation.

One of these tools, called *Traceband*, uses the concept of Hidden Markov Chains to model the network and perform the estimation based on a Probe Gap Model approach [7]. Traceband has shown to perform fast estimations without increasing the probing packet overhead in the network. However, its accuracy still need to be improved as shown in [11] and [7]. Traceband average error goes around 10% for a 30% congested link. This article presents a variant of Traceband based on a clustering technique called K-means, that is shown to reduce the estimation error of the tool when the network is highly congested.

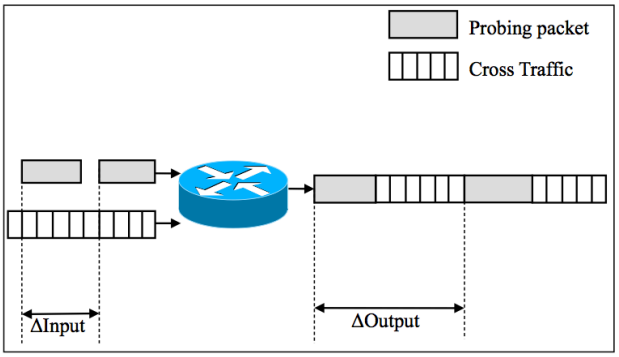

Fig. 1. Dispersion of probing packets due to cross traffic

This document is organized as follows. Section II shows a general description of Traceband. Section III presents k-means as the clustering technique to be used to filter wrong estimations of the available bandwidth. Section IV presents the modification performed to the original version of Traceband in order to embed a K-means algorithm as a way to reduce the estimation error of the tool. Section V compares the performance of Traceband with HMM and Traceband with K-means using a fully controlled testbed with synthetically generated traffic. Finally, Section VI presents the conclusions of the paper.

II. Traceband

Traceband [7] is an available bandwidth estimation tool that builds a Hidden Markov Model of the available bandwidth in the end-to-end connection. Traceband uses the Probe Gap Model to perform its estimation. This model uses the information obtained when a packet pair is sent to the network and its inter-departure time is affected by cross traffic arriving to intermediate nodes. The information is collected at the reception of the packet pairs and after measuring its inter-arrival time. More specifically, a packet pair is sent from host to host (end-to-end) with a certain space Δ_{Input} and due to cross traffic, that space between packets is modified (Δ_{Output}) when they reach the destination (See Figure 1). Assuming a single bottleneck in the end-to-end path and fluid traffic, the available bandwidth can be calculated by Equation 3 where C is the capacity of the end-to-end path. See [2], [1], [12], and [3] for a more detailed explanation of available bandwidth estimation methods and tools.

$$AB = C \times \left\{1 - \frac{\Delta Output - \Delta Input}{\Delta Input}\right\} \quad (3)$$

It is shown in [7] that Traceband is a fast tool that introduces low overhead to the network and that has a relatively low estimation error (around 10%). This error is close to other tools estimation errors according to the literature. Since *Traceband* has shown to perform fast and low-overhead estimations with similar accuracy when compared to other tools, this paper is focused on using a clustering technique to reduce traceband estimation error while keeping its overhead and estimation time unchanged. Intuitively, since every single estimation is affected by difficulties in the network and in the hosts [13], a set of estimations can be grouped to find a representative estimate to be averaged with other similar estimates. This *clustering* is shown in the following sections to be an effective way to filter wrong single estimates of the available bandwidth.

III. Clustering

A clustering algorithm creates a partition of data sets into clusters or subsets. Each element in the cluster has a common characteristic or pattern with other cluster elements. Clustering is computationally simpler than other grouping techniques such as the construction of a dendrogram with large data sets. A key step in the clustering algorithm is the selection of the membership criteria which also determines the number of groups to be created. According to Dubes et al. [14] groups can be defined by optimizing a criterion function as a result of running several times the clustering algorithm with different starting states. Jain et al. [15] shows the steps involved in a typical pattern clustering activity: 1) pattern representation, 2) pattern proximity measure, 3) clustering, 4) data abstraction, and 5) assessment of output. The loop mentioned by Dubes goes from step 3 back to step 1.

One of the most popular, simpler and widely used clustering methods is K-means [**?**]. The K-means algorithm, developed by MacQueen in 1967, follows a simple procedure for classifying a set of objects in a given number K of clusters. In the algorithm, the membership of any element is determined by its proximity to the cluster center (called centroid). That centroid is the average of all elements in the cluster. The algorithm can be depicted in five steps:

1) Select the number of K clusters.
2) Randomly create K groups and determine the K centroids.
3) Determine the centroid each element belongs to (that whose distance to the element is the nearest).
4) Calculate the new K clusters and centroids.

```
Read tm; /* Traceband total running time */
do
{
    Receive a train of |O|=30 packet pairs;
    conv_thresh=0.0001;
    For each packet pair at time t=1,..,T
        { rel_dispersion_t= ξ_t = (Δ_out- Δ_in) / Δ_in; }
    Observ_seq = O = {ξ_1, ξ_2, ... , ξ_T};
    K =  |O|/4;  // Initial number of clusters
    new_O = k_means(O, |O|, K, conv_thresh);
    For each new_ξ_t in new_O
        { AB_est= C*abs(1- new_ξ_t); }
    Print AB=mean(C*abs(1- new_ξ_t));
} while running_time <= tm;
```

Fig. 2. Traceband with K-means receiver pseudo code.

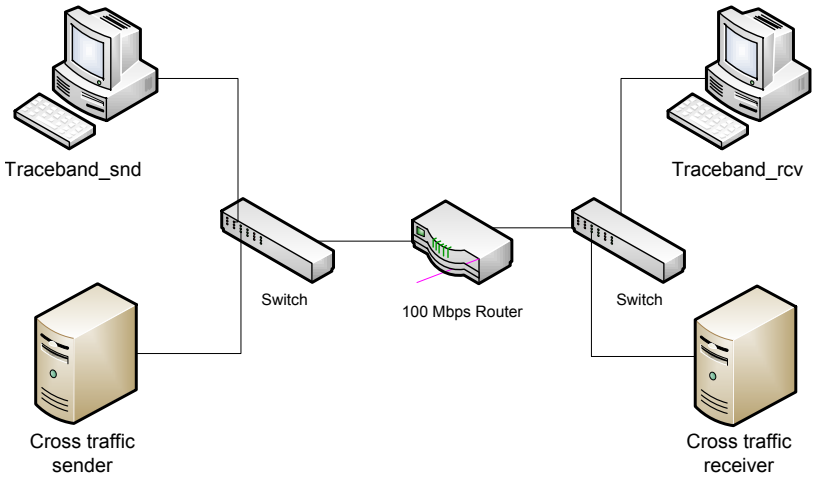

Fig. 3. Testbed used in the performance evaluation of Traceband with K-means and with HMM.

5) Repeat steps 3 and 4 until the clusters does not significantly change (or other convergence criteria)

However, the K-means algorithm has some disadvantages [**?**]. On one hand, the optimal clustering depends on the selection of the initial groups or centroids. On the other hand, the convergence criteria is not guaranteed and for large data sets the number of iterations to converge can be very large with the corresponding computational complexity.

There are different applications where K-means can be implemented. As it is shown in [16], K-means can be used in data mining, knowledge discovery, data compression and vector quantization, and pattern recognition and pattern classification. It is specially useful in medicine applications such as image processing for diagnosis, finding patterns in mortality rates from different types of cancer, among other.

IV. Traceband with K-means

The motivation of using a clustering technique in Traceband was to reduce the error caused by incorrect single estimates of the available bandwidth. The hypothesis was that by clustering them and adjusting their values to a common pattern would help to obtain average values with less variability. Since Traceband was written in ANSI C, an implementation of K-means by Roger Zhang [17] was inserted within Traceband code. Given that Traceband performs the estimation at the receiver side of the application [7], the K-means algorithm was part of that code as it is depicted in Figure 2. The pseudo code shown is similar to the original Traceband based on the Hidden Markov Model and shown in [7]. A train of 30 packet pairs is sent from sender to receiver. The observation sequence is calculated at the receiver as the relative dispersion of the packet pairs according to the fraction shown as part of Equation 3. Those observations are clustered in groups and are sent to the clustering algorithm which returns new observation values that are adjusted according to the cluster pattern. With the new values, a single averaged available bandwidth estimation is calculated. Traceband is repeatedly run during *tm* seconds to provide consecutive monitoring of the available bandwidth.

The implementation with K-means requires an initial number of clusters which is set to a fourth part of the number of observations. This is because we expect to cluster at least four groups of observations (with their corresponding centriods) from the beginning. The algorithm convergence criteria is set to 0.0001 as a threshold error between interactions.

V. Performance Evaluation

In this section, Traceband with K-means and Traceband with HMM (Hidden Markov Model) are evaluated and compared. The evaluation is performed using a completely controlled environment shown in Figure 3. This testbed is made of two Linux computers hosting both traceband versions and identified as *Traceband_snd* and *Traceband_rcv* for the sending and receiver sides of the application respectively. There are also two Linux computers which congest the network with synthetically generated cross traffic using an open source tool called MGEN [18]. Cross traffic is generated from *Cross traffic sender* to *Cross traffic receiver*. As it is shown in Figure 3, there are two networks connected by a 100 Mbps router.

To evaluate the performance Traceband with k-means against Traceband with HMM, three different scenarios were set in a 70% congested network: one using a Poisson distribution of cross traffic, one using a periodic (uniform) distribution and one using bursty cross-traffic where the length of the bursts and the burst interarrival times are both exponentially distributed with averages of 5 and 10 seconds, respectively. Both Traceband and MGEN generate UDP traffic with packet sizes of 1490 bytes.

Using the described scenarios, ten experiments where run for each scenario and the accuracy of each version of the tool was estimated as the relative error according to Equation 4, where m_{AB} is the value given by Traceband and μ_{AB} is the

Tool	Periodic	Poisson	Burst
% of error in Traceband with HMM	-13.54%	-13.57%	-25.11%
% of error in Traceband with K-means	-4.21%	-1.27%	-14.13%
% of improvement	68.91%	90.66%	42.77%

TABLE I
AVERAGE ESTIMATION ERROR OF TRACEBAND WITH HMM AND WITH K-MEANS FOR A 70% CONGESTED PATH.

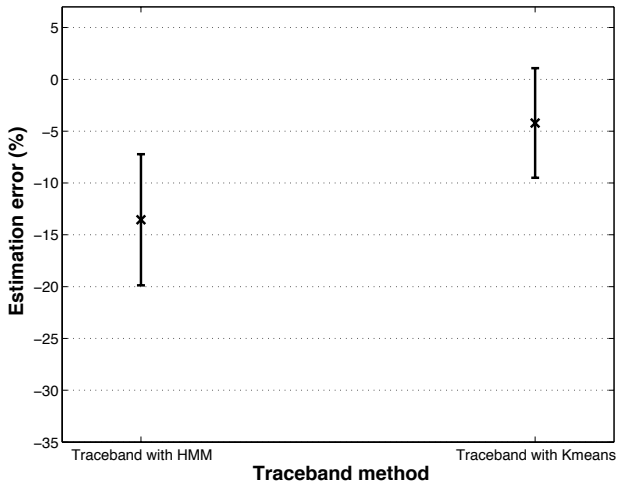

Fig. 4. Traceband estimation error for a 70% congested path with periodic traffic.

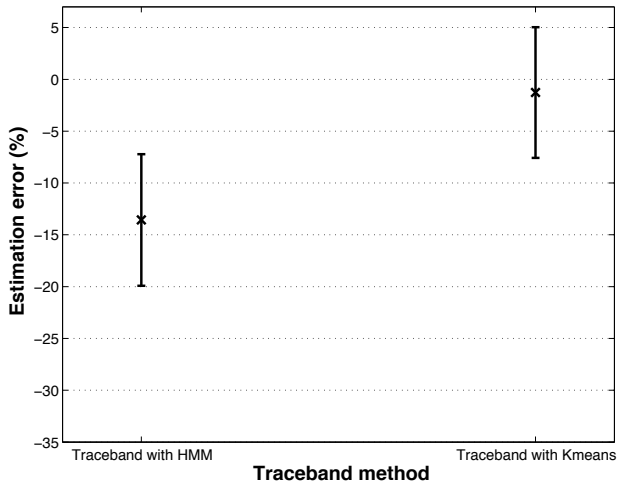

Fig. 5. Traceband estimation error for a 70% congested path with Poisson traffic.

expected AB value.

$$error = \left| \frac{m_{AB} - \mu_{AB}}{\mu_{AB}} \right| \times 100\% \quad (4)$$

Table I shows the results as averages of the ten experiments run for each network scenario. It is important to note that each experiment is by itself an average of AB estimates generated by single packet pairs. Those single estimates are the ones clustered to generate each one of the ten results. According to the Table, Traceband with K-means performs better than Traceband with HMM for all types of cross traffic. When averaging the percentage of improvement for all scenarios, it can be said that Traceband with K-means produces 67.45% better estimations than the original version of Traceband when the network is highly congested (at 70% of the en-to-end capacity). Negative values indicate that Traceband always overestimates the bandwidth availability. As expected, the estimation error with Bursty cross traffic is the higher than with Poisson and Periodic traffic. However, the fact that the average available bandwidth when using Poisson traffic is better than when using Periodic traffic, was one reason to perform a deeper analysis by observing the variability of the results and plotting confidence intervals. To do that, the t-student distribution was used to calculate and plot confidence intervals of 95% for each scenario.

4 Figure 4 shows the intervals when the path is congested with a 70% periodic cross traffic. In both cases it is shown that the intervals are similar which means that the variability of the results are almost the same. It is shown in the graph that Traceband with K-means has an interval that includes 0% estimation error which means that using that technique the tool can perform error-free estimations. This behavior does not happen with the original Traceband in this scenario.

Figure 5 shows the intervals when the path is congested with a 70% Poisson cross traffic. Similar to the experiments in periodic traffic, the variability of the estimation error is very similar in both versions of Traceband with a the worst average value in the case of Traceband with HMM. When

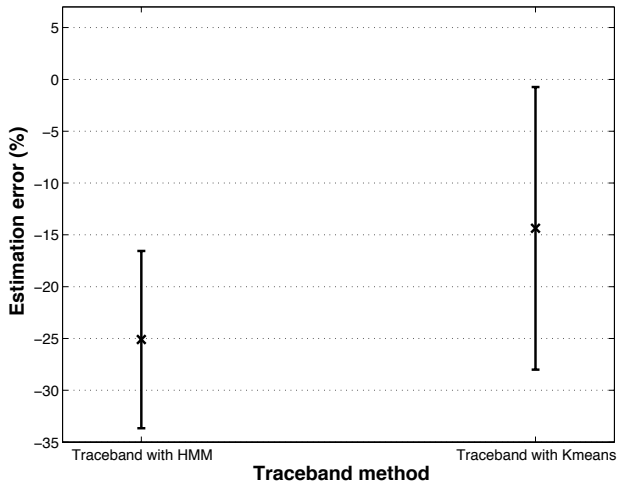

Fig. 6. Traceband estimation error for a 70% congested path with bursty traffic.

comparing the variability of the estimations using Traceband with K-means in Figures 4 and 5, it can be observed that although the average estimation error in the case of Poisson cross traffic is better that in the case of periodic traffic, the confidence intervals show that both estimation errors are statistically almost the same.

Finally, Figure 6 shows the intervals when the path is congested with a 70% bursty cross-traffic where the length of the bursts and the burst inter-arrival times are both exponentially distributed with averages of 5 and 10 seconds, respectively. In this case, although the estimation error is improved by Traceband with K-means, its variability is higher. In the worst case, however, the estimation is around 5 units better than Traceband with HMM. The best estimation error (lower limit in the interval) is close to zero in Traceband with K-means and close to 15% in Traceband with HMM. Therefore, although the variability is higher, the accuracy is considerably improved in the case of Traceband with K-means.

VI. Conclusions

It has been shown that under highly congested end-to-end paths, Traceband based on a clustering technique called K-means performs, in terms of estimation error, on average 67.45% better that the original Traceband based on a Hidden Markov Model approach. On average, the estimation error under a 70% congested path does not go further than 25% when estimating with HMM and 15% when estimating with K-means. The best traffic scenario is that of Poisson traffic and the worst is that of Bursty traffic (which is the closest scenario to real Internet connections).

When looking at 95% confidence intervals, a bursty traffic scenario shows a higher variability in the estimation error when using Traceband with K-means than the variability shown in Traceband with HMM. In spite of that, the worst case estimation error is better in the K-means version than the worst case in the HMM version.

This work shows that by using a clustering technique on an available bandwidth estimation tool, the estimation error can be reduced. This fact opens possibilities to implement a more complex clustering method or to implement k-means in other estimation tools in order to overcome the effect of wrong observations made by probing packets sent to the network.

References

[1] C. D. Guerrero and M. A. Labrador, "Experimental and analytical evaluation of available bandwidth estimation tools," in *Proceedings of the IEEE Local Computer Networks*, 2006, pp. 710–717.

[2] ——, "On the applicability of available bandwidth estimation techniques and tools," *Computer Communications*, vol. 33, no. 1, pp. 11–22, 2010.

[3] R. Prasad, C. Dovrolis, M. Murray, and K. Claffy, "Bandwidth estimation: metrics, measurement techniques, and tools," *IEEE Network*, vol. 17, no. 6, pp. 27–35, 2003.

[4] J. Strauss, D. Katabi, and F. Kaashoek, "A measurement study of available bandwidth estimation tools," in *Proceedings of the 3rd ACM SIGCOMM conference on Internet Measurement*, 2003, pp. 39–44.

[5] V. Ribeiro, M. Coates, R. Riedi, S. Sarvotham, B. Hendricks, and R. Baraniuk, "Multifractal cross-traffic estimation," in *Proceedings of the ITC Conference on IP Traffic, Modeling and Management*, 2002.

[6] N. Hu and P. Steenkiste, "Evaluation and characterization of available bandwidth probing techniques," *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 6, pp. 879–894, 2003.

[7] C. D. Guerrero and M. A. Labrador, "Traceband: A fast, low overhead and accurate tool for available bandwidth estimation and monitoring," *Computer Networks*, vol. 54, no. 6, pp. 977–990, 2010, 1389-1286.

[8] M. Jain and C. Dovrolis, "Pathload: A measurement tool for end-to-end available bandwidth," in *Proceedings of the 3rd Passive and Active Measurements Workshop*, vol. 11, 2002, pp. 14–25.

[9] B. Melander, M. Bjorkman, and P. Gunningberg, "A new end-to-end probing and analysis method for estimating bandwidth bottlenecks," in *Proceedings of the IEEE Global Telecommunications Conference*, vol. 1, San Francisco, CA, USA, 2000, pp. 415–420.

[10] V. J. Ribeiro, R. H. Riedi, R. G. Baraniuk, J. Navratil, and L. Cottrell, "pathchirp: Efficient available bandwidth estimation for network paths," in *Proceedings of the 4th Passive and Active Measurements Workshop*, vol. 2, 2003.

[11] C. D. Guerrero and M. A. Labrador, "A hidden markov model approach to available bandwidth estimation and monitoring," in *Proceedings of the Internet Network Management Workshop*, 2008.

[12] M. Jain and C. Dovrolis, "End-to-end available bandwidth: measurement methodology, dynamics, and relation with tcp throughput," *IEEE/ACM Transactions on Networking*, vol. 11, no. 4, pp. 537–549, 2003.

[13] C. D. Guerrero, *Available Bandwidth Estimation: A Hidden Markov Model Approach*. LAP LAMBERT, 2010.

[14] R. Dubes, "How many clusters are best?-an experiment," *Pattern Recognition*, vol. 20, no. 6, pp. 645–663, 1987.

[15] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review," *ACM Computing Surveys*, vol. 31, no. 3, pp. 264–323, 1999.

[16] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "An efficient k-means clustering algorithm: analysis and implementation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 881–892, 2002.

[17] R. Zhang, "K-means clustering," 2005.

[18] B. Adamson and S. Gallavan, "Mgen," 1997. [Online]. Available: http://cs.itd.nrl.navy.mil/work/mgen/index.php