# A Passive Available Bandwidth Estimation Methodology [1]

Albert Cabellos-Aparicio [a,*] John Thompson [b]
Francisco J. Garcia [c] Jordi Domingo-Pascual [a]

[a] *Universitat Politècnica de Catalunya, D. d'Arquitectura de Computadors, c/ Jordi Girona 1-3, D6-118, 08034 Barcelona, Spain*

[b] *Institute for Digital Communications, University of Edinburgh, Kings Buildings, Edinburgh, EH9 3JL, UK.*

[c] *Agilent Technologies, Agilent Laboratories Scotland, Edinburgh, EH3 976, UK*

**Abstract**

The Available Bandwidth (AB) of an end-to-end path is its remaining capacity and it is an important metric for several applications such as overlay routing and P2P networking. That is why many AB estimation tools have been published recently. Most of these tools use the Probe Rate Model, which requires sending packet trains at a rate matching the AB. Its main issue is that it congests the path under measurement. We present a different approach: a novel passive methodology to estimate the AB that does not introduce probe traffic. Our methodology, intended to be applied between two separate nodes, estimates the path's AB by analyzing specific parameters of the traffic exchanged. The main challenge is that we cannot rely on any given rate of this traffic. Therefore we rely on a different model, the Utilization Model. In this paper we present our passive methodology and a tool (PKBest) based on it. We evaluate its applicability and accuracy using public NLANR data traces. Our results -more than 300Gb- show that our tool is more accurate than pathChirp, a state-of-the-art active PRM-based tool. At the best of the authors' knowledge this is the first passive AB estimation methodology.

*Key words:* Passive, Measurement, Available Bandwidth

* Corresponding author.
*Email addresses:* acabello@ac.upc.edu (Albert Cabellos-Aparicio), john.thompson@ed.ac.uk (John Thompson), frankie_garcia@agilent.com (Francisco J. Garcia), jordid@ac.upc.edu (Jordi Domingo-Pascual).

[1] This work has been supported by the CONTENT research project.

## 1 Introduction

The Available Bandwidth (AB) of an end-to-end path is its remaining capacity, that is, the amount of traffic that can be sent along the path without congesting it. Recently the area of end-to-end AB estimation has attracted considerable interest. This is because the AB is an important metric for several applications such as overlay routing, P2P file transfers, server selection and inter-domain path monitoring among others. As a result several estimation techniques and tools based on active measurements have been developed.

Most of the proposed tools designed to estimate the AB fall into two categories: the *Probe Rate Model* (PRM) and the *Probe Gap Model* (PGM). The first model uses packet trains (a sequence of consecutive probe packets) and it is based on the concept of *self-induced congestion*. Informally if one sends a packet train at a rate lower than the AB, then the arrival rate of the packet train at the receiver will match the rate at the sender. However if the sending rate is greater or equal than the AB, then the packet train will congest the queues along the path and the receiving rate will be lower than the sending rate. Tools such as TOPP [2], PathLoad [3], IGI/PTR [4], pathchirp [5] and BART [7] use this model. The second model (PGM) uses packet pairs and relies on the differences of the input and output time gaps of the probe packets. However it has been shown recently that this model can underestimate the AB under certain conditions [20].

The PRM model has been used in many AB estimation tools and it has been shown as very accurate. However it suffers from one basic problem: PRM-based tools must send probe traffic at a rate equal or greater than the AB. This will fill the queues along the path congesting it. This means that, for each estimation, a PRM-based tool congests the measured path during a certain period of time. In fact *A.Shriram* showed recently in [16] that tools such as PathLoad [3] (a PRM-based tool) can significantly impact the response time of TCP connections.

In this paper we present a different approach, a passive available bandwidth estimation methodology. This is able to estimate the AB of a given path without introducing probe traffic. Our methodology is intended to be applied between two separate nodes (Figure 1): the sender and the receiver nodes. We aim to estimate the available bandwidth on the path by analyzing specific parameters of the already existing traffic exchanged between both nodes. Throughout the paper we will refer to this traffic as *data-traffic*.

The main challenge of our passive methodology is that we cannot rely on the PRM or the PGM model. Both models require sending probe-traffic at a rate matching the AB. That is why our tool takes base on a different model. *K.*

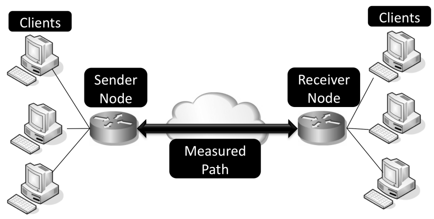



Fig. 1. The passive AB estimation architecture

*Harfoush* presented in [8] a mathematical model that shows that there is a linear relation between the utilization of a multi-hop path and the rate of the probe traffic. This model -intended for active tools- states that, if we inject probe traffic at a particular rate, the utilization increases linearly to this rate. Thus we can estimate the AB by sending probe traffic at an increasing rate. The rate of the packet train that estimates that the path is fully loaded (i.e. utilization=1) is the AB.

The main challenge of our methodology is that it cannot rely on any given rate of the *data-traffic*. Therefore instead of searching for this particular rate, we estimate the linear equation (rate vs. utilization). Once the equation has been estimated, the AB can be computed as the point where the utilization=1. In order to this we rely on Kalman Filtering [11]. The Kalman Filters are an efficient recursive filter that estimates the state of a linear system from a series of noisy measurements. These noisy measurements are, in fact, extracted from the *data-traffic*. Each measurement provides us an estimation of the utilization that, in turn, feeds the Kalman Filters to estimate the linear equation.

First we present the related work (Sect. 2) and then our methodology (Sect. 3). Afterwards we research the specific conditions that probe traffic has to fulfill in order to estimate the utilization optimally (Sect. 4). Then we investigate if such conditions are fulfilled by the Internet traffic (Sect. 5). As we will see in the paper we have analyzed several NLANR data traces that show that our methodology is feasible.

Finally we design a tool based on our methodology: PKBest and we evaluate its accuracy through simulation (sec. 6). Additionally we compare the accuracy of PKBest with that of pathChirp (a state-of-the-art PRM-based tool). Our results, more than 300Gb, show that the accuracy of our tool is higher to that of pathChirp. Considering all the scenarios evaluated, the mean relative error of our tool is 0.12 while for pathChirp it is 0.30.

## 2 Related Work

The area of AB estimation has attracted much attention recently and many tools have been published. As has been stated above, most of these tools use either the Probe Rate Model or the Probe Gap Model. However, at the best of the author's knowledge, there is very little research regarding passive AB estimation methodologies or tools.

First *M. Zangrilli* presented in [21] a one-sided, passive, PRM-based tool that uses TCP packets to estimate the AB. This tool uses the timestamps of data and ACKs packets to calculate round-trip times and then applies the PRM model. Second *C. Man* presented in [22] ImTCP, a new version of TCP that uses the arrival intervals of ACK packets as packet pairs to produce estimations using the Probe Gap Model. Both tools have the same issue, TCP cannot guarantee any given rate or pattern. This means that they are only able to produce estimations if the AB is similar to the actual TCP throughput.

Finally *S. Katti* presented in [23] MultiQ, a passive *capacity* measurement tool suitable for large-scale studies of Internet path characteristics. MultiQ is the first passive tool able to discover the capacity of multiple congested links along a path from a single flow trace and it is based on a modified version of the Probe Gap Model. Although this tool does not estimate the AB, both tools (MultiQ and PKBest) are truly passive and use NLANR data traces to evaluate their accuracy.

## 3 Methodology

In this subsection we present our methodology intended to passively estimate the Available Bandwidth (AB). First we detail the mathematical model, next we present how we apply the Kalman Filter to the model and finally we discuss the estimator of the utilization used.

### *3.1 Mathematical Model*

The utilization of a queue $i$ in a single-hop scenario is:

$$u_i = 1 - \pi_i \tag{1}$$

Where $\pi$ is the probability that the queue is void. Most of the existing AB measurement tools rely on using a *constant-rate fluid cross-traffic* model. This model assumes that the cross-traffic has infinitely small packet size and arrives

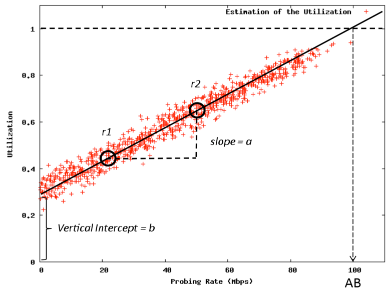



Fig. 2. The mathematical model

at the hop at a constant rate. In fact *X. Liu* showed in [19] that the tools that use the *constant-rate fluid* model can underestimate the AB under certain conditions. However eq. 1 does not make any assumptions about the nature of the cross-traffic.

If we transmit probe traffic at a rate $r$ through this link, then the utilization can be expressed as:

$$u_i(r) = min(1, u_i + \frac{r}{C_i}) \tag{2}$$

Where $C_i$ is the capacity of link $i$. For the multi-hop case *K. Harfoush* showed in [8] a first order approximation of eq. 2:

$$u(r) \approx min(1, ar + b) \tag{3}$$

Where $a$ and $b$ are constants. This equation states that there is a linear relation between the utilization of a path and the rate of the probe traffic sent. Figure 2 shows that, as the probe traffic rate increases so does the utilization (linearly). At a certain rate $r_{ab}$, the utilization will reach 1 (the path is fully loaded) then, the AB of the path is $r_{ab}$.

This means that with this model we can estimate the AB without sending the probe traffic at the AB rate. Once the linear equation (eq. 3) has been estimated the AB can be computed as:

$$AB = \frac{1 - \widetilde{b}}{\widetilde{a}} \tag{4}$$

### *3.2 The Kalman Filters*

Our tool uses a Kalman Filter (KF) to estimate the linear equation (eq. 3). The KFs are able to estimate a system defined by a state vector $x$, affected by an input $u$ through noisy measurements. In our case the network is our system and the noisy measurements are the estimations of the utilization. The system is also affected by a noise $w$ and the measurements have a noise $v$. Then the system is governed by the linear stochastic difference equation:

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1} \tag{5}$$

With a measurement $z$ that is:

$$z_k = Hx_k + v_k \tag{6}$$

Where the subscript $k$ refers to the discrete time and $A$ relates the state of the previous time step $(k-1)$ with the state of the new time step. Similarly $B$ relates the control input to the state $x$ while $H$ relates the state with the measurement. Then the KF estimates the process by using a form of feedback control: the filter estimates the process state at some time and then obtains feedback in the form of (noisy) measurements. The KF algorithm has two steps, in the first step ("time update“) the filter projects forward in time the state of the system and obtains an *a priori* estimate. In the second step ("measurement update") the filter uses a new measurement to correct the *a priori* estimate to produce an improved *a posteriori* estimate. After each time and measurement update pair, the process is repeated with the previous *a posteriori* estimates used to project or predict the new *a priori* estimates. This recursive nature is one of the main advantages of the Kalman Filters. The KFs assume that the system is linear and that the system noise $w$ and the measurement noise $v$ are Gaussian and independent. We refer the reader to [11] for further details on Kalman Filtering.

In our case the state vector $x$ that describes the system represents our linear model (the parameters of the sloping straight line from eq. 3):

$$x = \begin{pmatrix} a \\ b \end{pmatrix} \tag{7}$$

As it has been seen in the previous subsection our system is linear. We drop the input $u$ (and consequently $B$) because in our particular case the network is affected by the intensity of our probe traffic *and* the cross-traffic. As we cannot estimate the intensity of the cross-traffic we do not use this particular parameter. In addition we drop $A$ (i.e. $A = I$) because the state of the previous time step of the network will be the same as the state of the new time step.

Thus the following equation governs our system:

$$x_k = x_{k-1} + w_{k-1} \tag{8}$$

The measurements are governed by equation 6. We define $H$ as:

$$H = \begin{bmatrix} r & 1 \end{bmatrix} \tag{9}$$

This way the measurements $z$ (eq. 6) are seen by the KF as the actual utilization of the system under our probe-traffic load. The measurements $z$ are in fact the estimations of the utilization. Finally the predictor equations defined by the KFs in our particular case are:

$$\widetilde{x}_k^- = \widetilde{x}_{k-1} \tag{10}$$

$$P_k^- = P_{k-1}A^T + Q \tag{11}$$

And the corrector equations are:

$$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1} \tag{12}$$

$$\widetilde{x}_k = \widetilde{x}_k^- + K_k(z_k - H\widetilde{x}_k^-) \tag{13}$$

$$P_k = (I - K_k H)P_k^- \tag{14}$$

Where the "minus" superscript refers to the *a priori* estimates (before the measurement correction). $P$ is the estimate of the error covariance matrix, its value will be updated by the KF each time step. $K$ is the Kalman gain, a very important parameter of the KF. This gain is computed (in each time step) in eq. 12 and weights the new measurement with the *a priori* estimate in eq. 13. Finally $Q$ and $R$ represent the process and measurement noise covariance respectively. $Q$, the process noise covariance, is a 2x2 matrix that represents the variability of the system. This value must be set manually and it is a key parameter when considering the behavior of the KF. A high $Q$ means that the KF will consider the prediction as less accurate while the measurements will be considered as very accurate. Therefore the KFs will set the Kalman gain accordingly and each new measurement will be weighted heavier. Low values for $Q$ mean the opposite. We will come back to this in the results' section.

### *3.3 Estimating the Utilization*

Eq. 1 defines the utilization as the probability that there is at least one packet in the queues of the path. In order to estimate the utilization of a path the authors in [8] suggest sending a packet train (a sequence of probe packets), end to end, and compute the fraction of packets that have experienced queuing delay along the path. Probe packets are time stamped at the sender and at the receiver. Then the minimum one-way delay of the set of packets is computed. This minimum delay corresponds to the delay suffered by a packet that has not encountered queuing delay. Therefore the fraction of packets with a greater delay than the minimum delay is the fraction of packets that suffered queuing delay. Let $D = \{d_1 \cdots d_N\}$ be the set of one-way delays suffered by the packets of the train. Then the utilization is estimated as:

$$\widetilde{u} = \frac{\|\{d_i > min\{D\} | d_i \in D\}\|}{\|D\|} \tag{15}$$

## 4 Analysis of the Probe Traffic

As we have seen if we can estimate the utilization of a path we can apply the KF and estimate the AB through eq. 4. The utilization of the path is estimated through eq. 15. In this section we investigate which are the optimal parameters of the packet trains to estimate it. In the next section we analyze if such packet trains are present in Internet traffic.

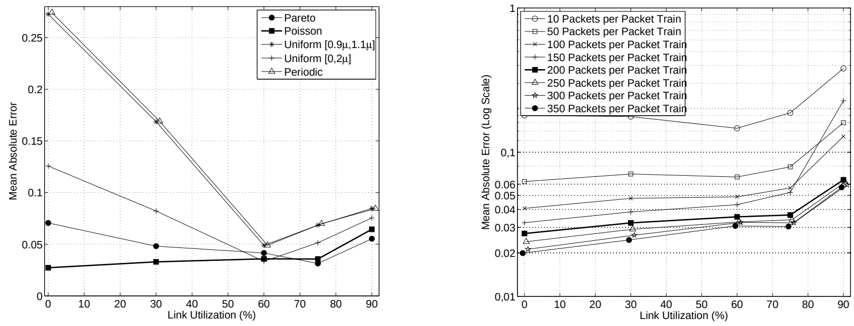
### *4.1 Distributions*

First we are going to investigate which is the optimal distribution of the inter-departure times of the packets within a packet train. According to the PASTA [13] property, if a packet train is sent with exponential inter-departure times, the packets arriving at the queuing system will sample the system queues, on average, as an outside observer would, at an arbitrary point in time. However *F. Baccelli* showed in [14] that Poisson probes are not unique in their ability to sample without bias. That is why we evaluate other distributions in order to analyze which one samples the queues better for our particular estimator. All the experiments have been carried out using NS2.

We evaluate a range of different distributions by sending 104 packet trains (of 200 packets and 1500 bytes as packet size) at different rates through a single link fed with Poisson packet arrivals. The cross-traffic packet sizes are distributed as in the Internet (see [15] for details): 50% (40 bytes), 10% (576 bytes) and 40% (1500 bytes). It is worth noting here that in the evaluation of our tool (Section 6) we use also Pareto distributed cross-traffic.

The experiment is performed with different packet trains distributions: Periodic, Poisson, Uniform ($[0.9\mu, 1.1\mu]$), Uniform ($[0\mu, 2\mu]$) and Pareto ($\alpha$ index = 1.16) and with different link loads (utilization=$\{0, 0.3, 0.6, 0.75, 0.9\}$). For each packet train we have computed the *absolute* error when estimating the utilization.

As Figure 3(a) shows the distribution that minimizes the error when estimating the utilization is the Poisson distribution whereas the worst one is the Periodic distribution. In fact the exponentially distributed packet trains are




(a) Mean absolute error for different distributions and link loads

(b) Mean absolute error for different packet trains lengths and link loads

Fig. 3. Analysis of the Probe Traffic

not severely impacted by the load of the link and the mean error is always below 0.07.

The periodic and Uniform distributions ($[0.9\mu, 1.1\mu]$) create "constant" packet trains where the inter-departure time of the packets is very similar. The Probe Rate Model roughly describes the behavior of these packet trains. If the rate of the packet train is above the AB, then almost all the packets are queued and thus, the utilization is overestimated. If the rate is below the AB, then the packets do not congest the tight link queue and thus, the utilization is underestimated. Regarding the Pareto and Uniform ($[0\mu, 2\mu]$) distributions, Figure 3(a) shows that they are more accurate under high link loads. This is because these distributions have also some "periodicity". When the link load is at 90%, the cross-traffic rate congests the tight link queue and almost all the packets are queued. This means that the utilization, for this very special case, is accurately estimated.

From this experiments we can conclude that the optimal inter-departure time of the trains is Poisson. This is an encouraging result since the probability of finding exponentially distributed packet trains in the Internet traffic is higher than for other distributions.

### *4.2 Length*

In this subsection we evaluate the optimal packet train length. We have sent packet trains using different lengths {10,50,100,150,200,250,300,350} in a single-hop scenario. The link is loaded with cross-traffic {0,0.3,0.6,0.75,0.9} and the packet trains are sent at different rates, ranging from 0.1Mbps to the

AB rate. In this case, the distribution is fixed to Poisson and the packet size is fixed to 1500 bytes. The rest of the simulation parameters are the same than in the previous experiment.

Figure 3(b) (note that the X-Axis uses a log-scale for clarity) shows the results of the experiments. Packet trains with a length lower or equal to 150 packets suffer from a large error when estimating the utilization.

Regarding packet trains longer than 150 packets, the mean error is bounded to 0.06. In addition the accuracy is not significantly impacted by the utilization of the link under study. In fact the error is slightly reduced as the packet train length increases. However it is important to remark that long packet trains suffer from one basic problem, which is that the utilization may change during the transmission of the train and this may lead to incorrect estimates. Thus there is a tradeoff between the accuracy of the estimations and the duration of the measurement. Longer packet trains (250, 300, 350...) have slightly less error, however as the figure shows, this extra accuracy is not justified since the duration of the measurement increases dramatically (i.e. packet trains of 250 packets last 20% longer than packet trains of 200 packets). In addition we have to take into account that larger trains have less probability of being present in Internet traffic. That is why we believe that the optimal length for our trains is 200 packets.

### *4.3 Size*

At this point we have concluded that the optimal packet inter-departure time distribution is Poisson and the optimal length is 200. The last parameter to evaluate is the packet size. Obviously we cannot rely on any given packet size or distribution. That is why in this subsection we analyze the impact of the random packet sizes in our packet trains. We have used the same parameters for this simulation than for the previous one, but the distribution is now fixed to Poisson, the length to 200 packets and the packet sizes are randomly chosen (uniformly) between 40 and 1500 bytes.

Figure 4 shows that random packet sizes do not impact the accuracy of the packet trains. By analyzing the results we notice that, for small rates (less than 5Mbps), and when the link is near congestion (90%), the estimation of the utilization is inaccurate. This can be seen in the long tail of the CDF. This is because for such small rates, with random packet sizes, the packet train has very few packets able to congest the link and thus, the utilization is underestimated.

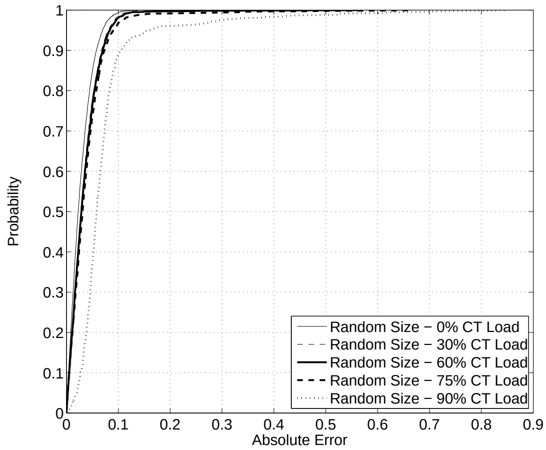



Fig. 4. Absolute error when estimating the utilization with random packet sizes

### *4.4 Design of PKBest*

Figure 1 presents the main architecture of PKBest. Our tool is deployed between a sender and a receiver node that exchange *data-traffic*. The receiver needs the delay of $\rho$ consecutive packets. Then it processes these packets and extracts "valid" packet trains (i.e. packet trains of 200 packets which are exponentially distributed). Then for each "valid" packet train, it estimates the utilization, the results are fed into the Kalman Filter, and it computes the AB. Since our methodology does not need the actual delay, but the *relatives* delays, it does not require clock synchronization.

It is important to remark that the delays of the packets can be obtained in several ways. First a generic passive measurements infrastructure can be used [33]. This type of architectures deploy two points of capture, one at the sender node and one at the receiving node. The points of capture, for each captured packet, send a hash and the timestamp of the packet to a central processing unit. This unit matches the hashes and extracts the delays of the packets. A different approach can be used using In-Line Measurements [24]. This method defines an IPv6 extension header that includes a timestamp and that can be used to compute several QoS metrics. This can be easily ported to IPv4 and a timestamp can be added in a special header after the transport header or in a new IP header using tunneling. The receiver node always removes these headers. Since some IP packets already have the largest size (1500 bytes in Ethernet links) we cannot always add a new header. In these cases the timestamps can be included into the next packet that is smaller than the MTU (along with a special identifier).

In this paper we consider this latter case for our methodology. The amount of timestamped packets by the sender node ($\rho$) must be enough to extract at least

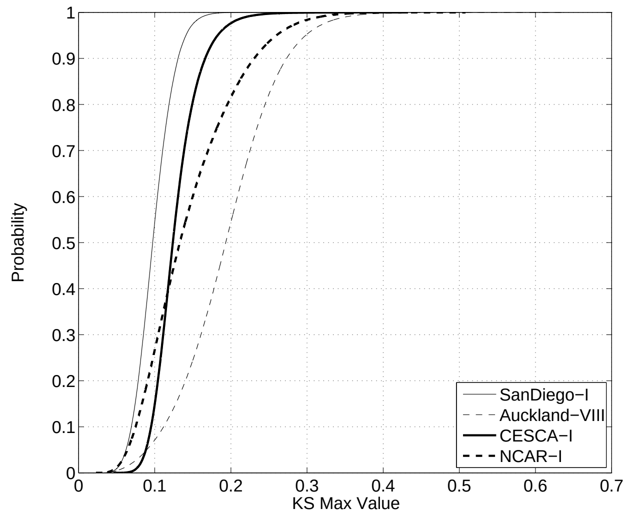



Fig. 5. KS Test for the NLANR traces

one "valid" packet train at the receiver node. In the next section we analyze the number of "valid" packet trains present in several Internet data-traces and we discuss possible values for $\rho$.

## 5 Applicability and Limitations of our Methodology

In this section we analyze the number of "valid" packet trains present in Internet traffic. Additionally we evaluate the main limitations of our methodology.

### *5.1 Analysis of public data traces*

In order to study if our tool will find such packet trains in real traces we have analyzed four different public NLANR (National Laboratory for Applied Network Research) data traces [29]. These traces, well-known in the passive measurements research community, have been collected from a variety of links at different research networks. The traces are public and were anonymized.

Specifically we have analyzed the CESCA-I, SanDiego-I, NCAR-I and Auckland-VIII data traces. It is important to note that among all the public data traces, we have analyzed all the traces that contained contiguous packets. Sampled traces are not useful for our study.

We have processed the traces in the following way. We consider a packet train as 200 consecutive packets within each trace. For each packet train, we determine if it can be considered as exponentially distributed and we compute the number of "valid" packet trains per second.

Figure 5 shows a CDF of the Kolmogorov-Smirnov (KS) test [26] against a theoretical exponential distribution for each packet train. The critical value for this test, with a 95% confidence interval, is 0.096. This means that packet trains with a KS value below or equal 0.096 can be considered as exponentially distributed and thus, valid for our methodology. As the figure shows all the traces contain exponentially distributed packet trains: Auckland-VIII (8.7%), CESCA-I (17.6%), NCAR-I (26.5%) and SanDiego-I (53.2%). This variation of these results may arise from the different types of applications that each link is supporting.

Figure 5 also allows us to discuss the value of $\rho$. For the SanDiego-I data trace if we timestamp 400 packets we will likely have at least one valid packet train. For the NCAR-I and CESCA-I we should timestamp 1000 packets. Finally for the Auckland-VIII trace we should timestamp around 20000 packets.

Table 1
Ratio of "valid" packet trains per second

| Data Trace | Min | Mean | Max | Std.Dev |
|---|---|---|---|---|
| Auckland-VIII | 0.001 | 0.070 | 3.376 | 0.031 |
| CESCA-I | 0.084 | 0.231 | 0.633 | 0.051 |
| NCAR-I | 0.004 | 0.095 | 0.874 | 0.072 |
| SanDiego-I | 0.13 | 0.037 | 1.523 | 0.036 |

Regarding the ratio of "valid" packet trains per second, Table 1 shows the results. As the table shows, in the worst case, the Auckland-VIII data trace, our tool would find a "valid" packet train each 3.376s. This means that our tool can produce an estimation (roughly) each 3 seconds. We believe that this resolution is enough for many applications. For instance PathLoad [3] (considered as one of the most accurate tools [16]), produces an estimation each 10-100 seconds, depending on the scenario [16]. Taking into consideration that the ratio of "valid" trains per second is high, we use 4 "valid" trains to produce an estimation.

### *5.2 Evaluation of the Limitations*

In the previous subsection we have analyzed different data traces and evaluated the number of "valid" packet trains present. The remaining parameter to evaluate is the rate of these packet trains. This is a key parameter when considering the accuracy of our methodology. Let's consider figure 6 as an

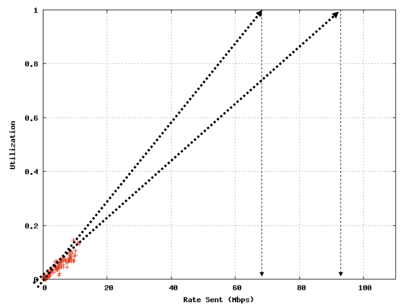



Fig. 6. Limitation of our tool

example. In this case our tool is processing several packet trains to estimate the utilization. As eq. 3 states (and the figure shows) there is a linear relation between the rate of these packet trains and the estimated utilization. Our tool feeds the Kalman Filter with the estimations of the utilization and the rates. In turn the Kalman Filter estimates $a$ and $b$ (eq. 3). Finally using eq. 4, we estimate the AB. The main concern, in this case, is that the rates of these packet trains are very low (0-10Mbps in the figure) compared to the actual AB (80Mbps). This means that the error when estimating $a$ and $b$ will be "projected" to the AB point, producing a larger error. We can conclude that, the closer we operate to the actual AB, the more accurate the estimations will be.

In this subsection we evaluate the relation between the distance to the AB of the rates of the packet trains and the achieved accuracy. First we need to consider all the possible path scenarios. In Section 3.1 we assumed that any path loaded with any cross-traffic can be modeled using eq. 3. This means that any scenario can be represented by two parameters: $a$ are $b$. The valid range of values for $b$ is $[0,1]$ since a negative utilization is not possible. Consequently given an AB of $\alpha$, $a = \frac{(1-b)}{\alpha}$. Thus all the possible paths are given by the following equation:

$$g(\alpha) = \{(a,b)|b = [0,1] \bigwedge a = \frac{(1-b)}{\alpha}\} \tag{16}$$

Second we need to model the rates of the received packet trains. This is the same as modeling the bandwidth of aggregated traffic. This bandwidth is assumed as Gaussian in [30] based on the measurements of [31] (and the references therein). Specifically the measurements show that the vertical aggregation of at least 25 users, with an aggregate average traffic rate of 25Mbps, is a good fit with the Gaussian model in time scales that are longer than 128msec.

Finally we need to model the error of the estimations of the utilization. Considering all the experiments carried out in Subsection 4.3, the error can be modeled as a Gaussian distribution. Specifically it can be modeled as $N(0, 0.03)$.

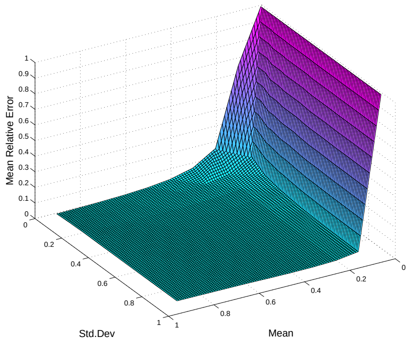
Having modeled the path, the rates of the packet trains and the error of the estimations of the utilization, we perform the following experiment using Matlab. We consider the paths where $\alpha = 100$ and $b = \{0, 0.3, 0.6, 0.75, 0.9\}$. For each path, we fed the Kalman Filters with packet trains at rates distributed as $N(\mu, \sigma)$ We consider the following ranges: $\mu = [0.1\alpha, \alpha]$ and $\sigma = [0.1\mu, \mu]$. It is important to note that, for each rate, we compute the estimation of the utilization affected by an error modeled as: $N(0, 0.03)$.

Figure 7 presents the results of the experiments. The x-axis represent the mean of the rates while the y-axis its standard deviation. Note that the mean is related to the AB (real mean=$\mu \times \alpha$), similarly the standard deviation is related to the mean (real std.dev=$\sigma \times \mu$) Finally the z-axis presents the accuracy of the final estimations of the AB. Specifically we present the mean relative error ($\epsilon = \min(1, \frac{abs(\widetilde{ab} - \alpha)}{\alpha})$) of the accuracy. We plot a figure for each value of $b$.

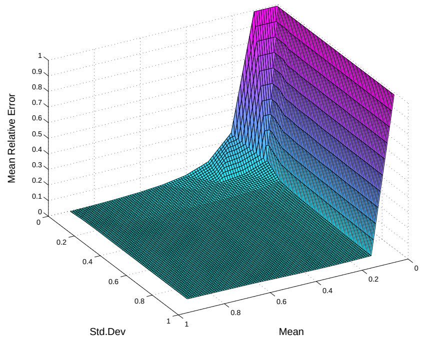
For low values of $b$ the figure shows that the accuracy is not affected by $\sigma$ unless it is 0. This is because even a small amount of variability in the rates of the packet trains is enough for the Kalman Filters. Regarding the mean, as expected, it impacts the accuracy of the estimation in certain cases. When the path is below 60% and the mean rate above 20% of the AB, then the achieved accuracy is bounded at 0.01. However as the utilization increases (75% and 90%) the error, for small mean values increases. In this case the accuracy is bounded at 0.01 when the mean rate is above 50% of the AB. In fact in theses cases (75% and 90%) $\sigma$ also affects the achieved accuracy. This is also an expected results since a larger $\sigma$ means that some trains are sent at a closer rate to the AB.

This analysis helps us identifying the main limitations of our methodology. When the path is not congested (below 60%) almost any packet train rates are valid to achieve good accuracy. When the path is near congestion (above 75%), then the packet train rates must be around 50% of the AB. It is worth to note that, when the path is near congestion the AB decreases. For instance when a 100Mbps link is near congestion, let's say 75%, the AB is 25Mbps. This means that the mean rate of our packet trains should be around 12.5Mbps. Thus, as the path utilization increases, the AB decreases and the required rates of our packet trains also decrease.

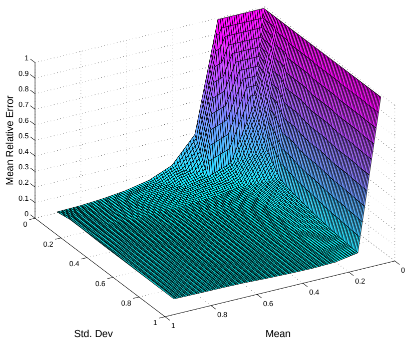
Finally table 2 shows the mean rates of the "valid" packet trains present in each data trace. Since these traces have been collected during large periods of time (days) they show a very large variability regarding the rate.



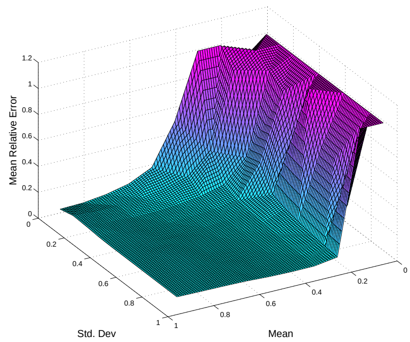

(a) u=0%



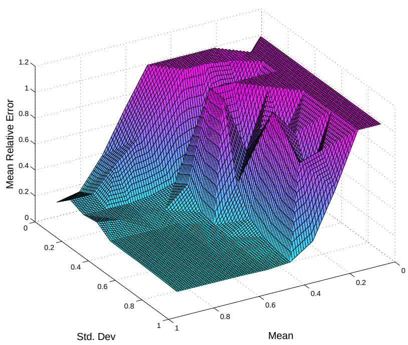

(b) u=30%




(c) u=60%




(d) u=75%




(e) u=90%

Fig. 7. Evaluation of the limitations of our methodology

Table 2
Rates of the "valid" packet trains (in Mbps)

| Data Trace | Min | Mean | Max | Std.Dev |
|---|---|---|---|---|
| Auckland-VIII | 0.01 | 7.67 | 330.77 | 10.38 |
| CESCA-I | 84.38 | 230.61 | 633.63 | 50.89 |
| NCAR-I | 4.21 | 95.87 | 870.90 | 75.32 |
| SanDiego-I | 13.34 | 37.39 | 152.15 | 108.17 |

## 6 Evaluation of the Accuracy

In this section we present a complete evaluation of the accuracy of our tool. In order to choose the evaluation scenarios we follow the methodology presented in [28] where the authors compare the performance of different AB estimation tools. Our evaluation is based on this methodology and on the evaluation of PathLoad [3]. Specifically we consider the cross-traffic load similarly to [3] while our evaluation scenarios are close to the ones depicted in [28].

In all the experiments each link is loaded with three different cross-traffic flows with Pareto inter-departure times ($\alpha = 1.19$). Each flow uses different packet sizes (40, 576 and 1500 bytes). The amount of packets of each size is distributed as in the Internet [15]: 50% (40 bytes), 10% (576 bytes) and 40% (1500 bytes). We have chosen Pareto-distributed cross-traffic because it has been shown that $N$ Pareto flows mimics the burstiness behavior of the Internet traffic [25]. In addition we repeat the experiments using the same setup but with exponentially distributed cross-traffic for comparison. Each experiment is run for 600 seconds. In the last case of the evaluation (Section 6.5), we use NLANR data traces as cross-traffic. With this latter case we aim to test our tool under a highly realistic scenario.

Regarding the data traces we test our tool using the NCAR-I and the CESCA-I traces. According to the parameters evaluated in Section 5 the most suitable trace for our tool is the SanDiego-I while the less suitable is the Auckland-VIII. That is why we choose these two traces to evaluate the accuracy of our tool. Since $\rho$ does not affect the accuracy of our methodology but the amount of estimations per second, we timestamp all the packets (i.e. $\rho = \infty$). The value for the process covariance matrix $Q$ (of the KFs) used in the evaluation is:

$$Q = \begin{bmatrix} 10^{-6} & 10^{-7} \\ 10^{-7} & 10^{-2} \end{bmatrix} \tag{17}$$

Finally for each experiment we compare the achieved accuracy of our tool

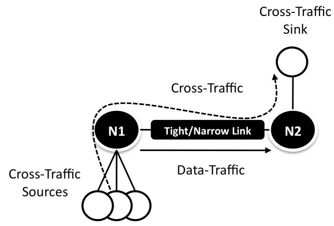



Fig. 8. Single-Hop, Single-Bottleneck, same tight and narrow link scenario

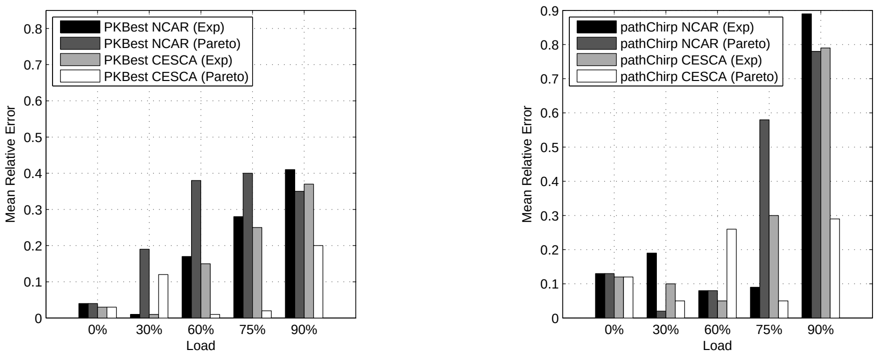
with that of pathChirp [5] (considered as a state-of-the-art tool [28]). We choose pathChirp because it is one of the less intrusive tools [28]. Since this is also one of the main advantages of our tool we believe that this is a fair comparison. pathChirp is evaluated under exactly the same scenarios and the same cross-traffic: the Pareto cross-traffic and the NLANR data trace. We have used the publically available implementation of pathChirp [32]. The authors of pathChirp claim that the parameters of pathChirp do not need to be set because it adapts them to the situation automatically. Therefore we use pathChirp's default parameters throughout the evaluation.

### *6.1 Single-Hop*

In the first set of experiments we evaluate our tool in a single-hop scenario (Figure 8), this scenario represents paths on which our methodology is likely to encounter only a single congested link. In this set the cross-traffic has different loads at the bottleneck link {0,0.3,0.6,0.75,0.9} and the capacity of the bottleneck link is set accordingly {500Mbps,714Mbps,1250Mbps,2Gbps,5Gbps}.

Figure 9 shows the mean relative error. Unless noted otherwise we compute the error as ($\epsilon = \min(1, \frac{abs(\widetilde{ab} - ab)}{ab})$). As the figure shows the Pareto cross traffic has not a noticeable impact on the accuracy of our tool (compared to the exponential cross-traffic). Our tool is based on a model that does not rely on a constant-fluid cross traffic (see Section 3.1) and it is able to deal with the burstiness of the cross-traffic. Regarding the *data-traffic*, PKBest achieves a slightly higher accuracy when operating with the CESCA-I data trace. This is because, as has already been mentioned in Section 5, the mean rates of the CESCA-I trace are higher.

The intensity of the cross-traffic at the tight link impacts the accuracy of PKBest. When the utilization of the tight link is high the slope of the linear eq. 4 is close to 0, and the error of the estimations is "projected". This has been analyzed and evaluated in Section 5. Regarding pathChirp's accuracy it is severely impacted by the cross-traffic intensity. When the tight link is at 90% of its capacity the error of pathChirp's estimations are close to 1. In [28], pathChirp was evaluated under a similar scenario, specifically with a link load




(a) PKBest's results

(b) pathChirp's results

Fig. 9. Results for the single-Hop, single-Bottleneck, same tight and narrow link scenario

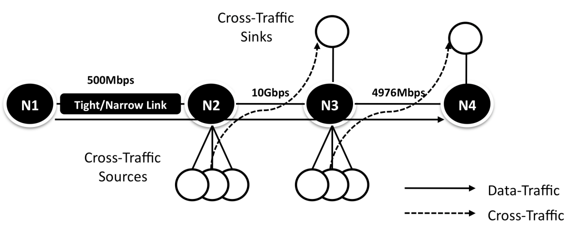


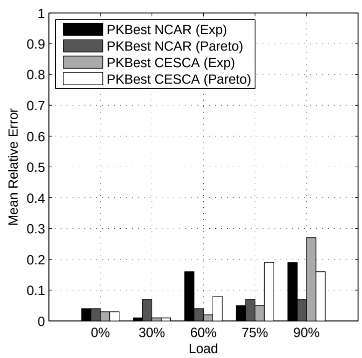

Fig. 10. Multi-Hop, Single-Bottleneck, same tight and narrow link scenario

of 53%, and the mean relative error was between 0.1 and 0.25. In our case the closest scenario is when the link load is at 60% of its capacity, and the achieved accuracy of pathChirp in our case is between 0.08 and 0.27. Thus our results agree with that of [28].

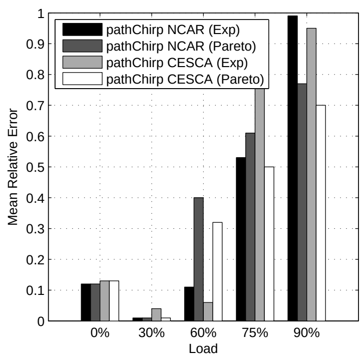
### *6.2 Multi-Hop, Single Bottleneck, same tight and narrow link*

In the second set of experiments we use a multi-hop scenario. Almost all the paths on the Internet are multi-hop. In this case the experiments are intended to evaluate the accuracy of our tool when it is affected by non-tight link cross-traffic. In this case the tight link is located between N1-N2, the load at the links N2-N3 and N3-N4 varies: {0,0.3,0.6,0.75,0.9}. The AB is set to 500Mbps.

Figure 11 shows the results for this scenario. PKBest's estimates match the AB and the error is bounded to 0.27. Non-tight link cross traffic has not a noticeable impact in PKBest's estimates. This is because our model sees this cross traffic as a minor increase of $b$. However pathChirp is severely affected, especially when the non-tight link cross-traffic intensity is high. It is worth noting that these cases can be considered as extreme scenarios.




(a) PKBest's results




(b) pathChirp's results

Fig. 11. Results for the multi-Hop, single-Bottleneck, same tight and narrow link scenario

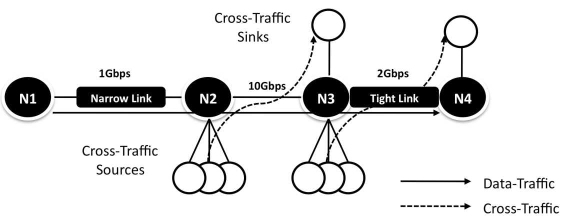
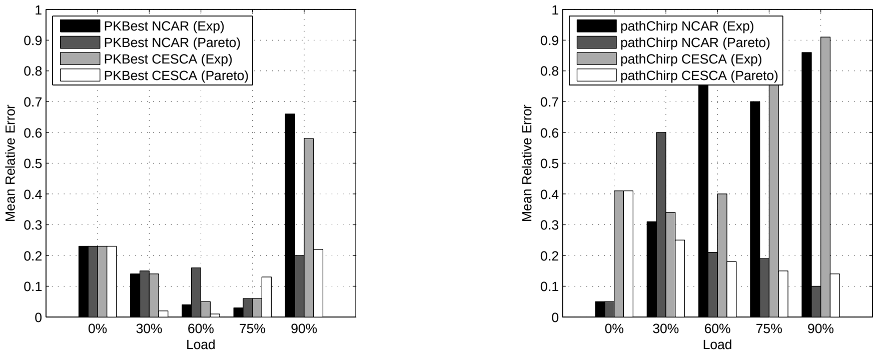



Fig. 12. Multi-Hop, Single-Bottleneck, different tight and narrow link scenario

### *6.3 Multi-Hop, Single Bottleneck, different tight and narrow link*

Many AB estimation tools have different accuracy depending on the location of the tight and narrow link. In this experiment we evaluate our tool in a scenario with different tight and narrow link (figure 12). This case may be very common on the Internet since an ISP access link that is shared among a large user population may have a lower AB. In this case the narrow link is between N1-N2 and the tight link is at N3-N4. The load at the link N2-N4 varies {0,0.3,0.6,0.75,0.9} while the utilization of the tight link is always 0.625 and thus, the AB=750Mbps.

Again figure 13 shows the results of these experiments. We can see that PKBest's estimates agree with the AB. This scenario does not affect the accuracy of our tool because the load at the tight link is reasonable low (62.5%) and, as we have seen earlier, it is not affected by non-tight link cross-traffic. Regarding pathChirp's it shows a larger error. A similar scenario has also been evaluated in [28] and the error of pathChirp increased by a factor of 2-3 compared to the scenario of figure 8. Again our results agree with the evaluation of pathChirp presented in [28].




(a) PKBest's results (b) pathChirp's results

Fig. 13. Results for the multi-Hop, single-Bottleneck, different tight and narrow link scenario

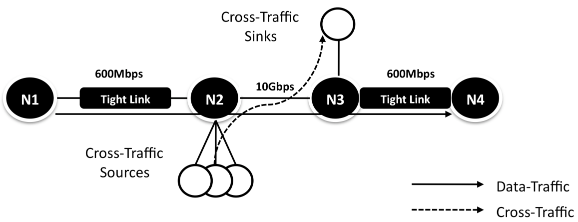
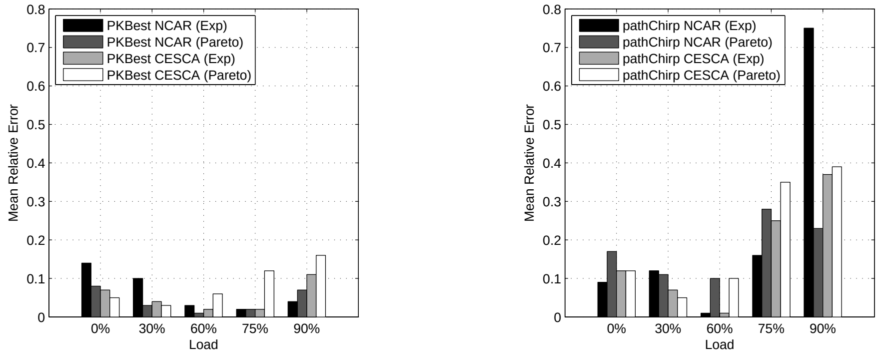



Fig. 14. Multi-Hop, Multiple-Bottleneck, two potential tight links

### *6.4 Multi-Hop, Multiple Bottleneck, two potential tight links*

Finally most ABET assume the existence of only a single congested link on the path. For instance Spruce [25] (a PGM-based tool), needs the *a priori* knowledge of the capacity of the tight link. Therefore it may not perform well in this scenario. Additionally it is conjectured that PRM-based tools might underestimate the AB in the presence of multiple bottleneck links [25]. In order to study this scenario we simulate the topology of figure 14. With this setup we evaluate our tool in an scenario with one narrow link and two potential tight-links. On average the latter is the "tigher link" but the *data-traffic* experiences queuing at both links. In this case the utilization of the link N2-N3 is {0,0.3,0.6,0.75,0.9} and the utilization of the latter tight link is 0.083.

Figure 15 shows the results for these experiments. In this case PKBest shows a very good accuracy and the error is bounded to 0.17. In this scenario both 'tight links' have a low load and, as noted before, the non-tight link cross-traffic does not affect PKBest's accuracy. pathChirp's estimates also match the AB when the non-tight link is not congested.




(a) PKBest's results

(b) pathChirp's results

Fig. 15. Results for the multi-Hop, multiple-Bottleneck, two potential tight links

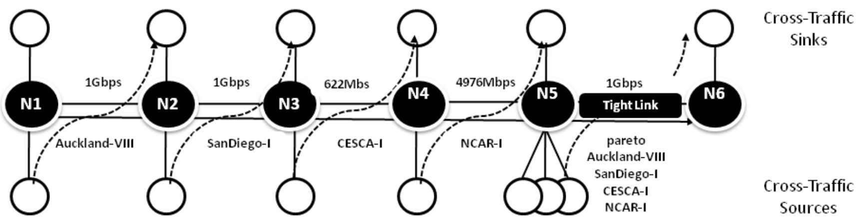



Fig. 16. NLANR data traces as cross-traffic

### *6.5 NLANR data traces as cross-traffic*

Finally in this set of experiments we setup a highly realistic environment (Figure 16). We evaluate the accuracy of PKBest in a path with 5 hops where each hop is loaded with a different NLANR trace as cross-traffic. This set of experiments is intended to evaluate the performance of our tool with realistic cross-traffic.

In this case each link is loaded with an NLANR trace as cross-traffic. The last link is the tight link and it is loaded either with Pareto or a NLANR data trace. Since some of the NLANR data traces have a low rate, we multiply the inter-departure times of the packets by a scale factor to increase its rate. This way the utilization in the last link is always around 50%. Note that the NLANR traces have a variable rate, and that the AB varies over the time. We run this experiment for 600 seconds and the AB=400Mbps. We use the CESCA-I data trace.

Table 3 shows the results of this set of experiments. As the table shows the mean estimations of the AB match with the real value (400Mbps), in fact the mean relative error is below 0.10 in all the cases. The table also shows that the achieved accuracy is similar when operating with realistic cross-traffic or

Table 3
Statistics of the results (Values in Mbps)

| Data Trace | Min | Mean | Max | Std.Dev |
|---|---|---|---|---|
| Auckland-VIII | 182.38 | 358.31 | 798.97 | 73.79 |
| CESCA-I | 184.00 | 386.15 | 823.97 | 86.30 |
| NCAR-I | 185.08 | 387.24 | 813.03 | 87.21 |
| SanDiego-I | 185.23 | 384.93 | 839.91 | 85.13 |
| Pareto | 163.87 | 374.82 | 821.72 | 95.21 |

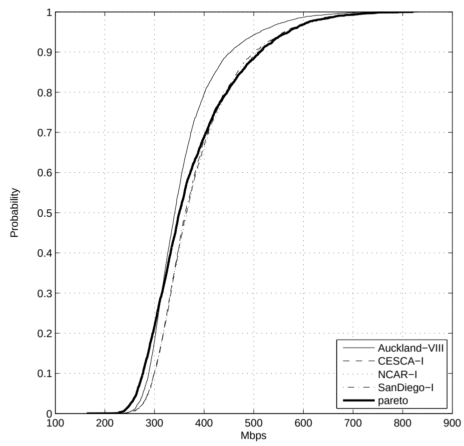



Fig. 17. NLANR traces as cross-traffic (Results)

with Pareto cross-traffic. This is an expected result since it has been shown that $N$ Pareto flows mimics the burstiness behavior of the Internet traffic [25].

The figure 17 shows a CDF of the estimations, for all the cases. The figure shows that the behavior of our tool is very similar when operating through the different types of cross-traffic. The variability of the estimations is due to several factors. First the AB changes during the experiment (as noted previously). Second the configuration of the process covariance matrix $Q$ and finally the error when estimating the utilization. As it has been stated before a high $Q$ means that the KF will consider the prediction as less accurate while the measurements will be considered as very accurate. This means that a high $Q$ helps the KFs to better track changes of the AB, but produces less stable estimations. A low $Q$ means the opposite. Throughout the paper we have used a high value for $Q$ (eq. 17).

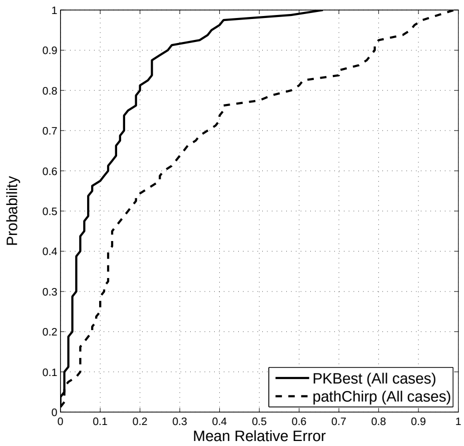



Fig. 18. Summary of Results

### *6.6 Summary of the Results*

This subsection presents a summary of the results. The figure 18 shows a CDF of the mean relative error of both tools (PKBest and pathChirp) considering all the cases. As the figure shows the mean relative error for PKBest is 0.12 while for pathChirp is 0.30. For both tools the maximum error (0.66 for PKBest and 0.99 for pathChirp) occurs when the tight link is near congestion.

## 7 Conclusions

The Available Bandwidth (AB) is one of the most important metrics in the area of network measurements. Many tools have been published and most of them use the Probe Rate Model. This model sends probe traffic at a rate similar (or even larger) than the AB. This leads to congestion and reduces the performance of the path under measurement.

In this paper we have presented an original *passive* methodology to estimate the AB of an end-to-end path. Our methodology, intended to be applied between two separate nodes, estimates the path's AB without introducing probe traffic. Instead of injecting probe traffic, it relies on specific parameters of the traffic exchanged between both nodes.

The main challenge of our methodology is that we cannot rely on any given rate of this traffic. That is why we base our methodology on a different model, the Utilization Model. This model states that there is a linear relation between the rate of the traffic and the utilization of the path. We estimate this linear equation using Kalman Filtering and we "project" it to the AB point (i.e.

utilization=1). Therefore if we can estimate the utilization at different rates, we can compute the AB.

In the paper we have analyzed which are the specific conditions that the traffic exchanged has to fulfill to estimate the utilization. Our conclusions show that 200 consecutives packets, with exponential inter-departure times, estimate the utilization with a mean absolute error of 0.1 (in 90% of the cases). We consider the traffic that meets these conditions as "valid" packet trains. Then we have investigated whether such packet trains are actually present in real Internet traffic. We have analyzed four NLANR public data-traces (60Gb in total). Our analysis shows that, considering all the traces and grouping the packets into packet trains, roughly 25% of packet trains are "valid". Additionally we have analyzed the amount of "valid" packet trains per second and (on average) there is a "valid" packet train each 0.433 seconds.

The main limitation of our methodology is the distance between the rate of the "valid" packet trains and the AB. The closer we operate to the AB rate, the more accurate the estimations will be. This is because the error of the estimations of the linear equation is "projected" to the AB point. We have evaluated this limitation modeling the path, the rates of trains and the error. Our analysis shows that when the path is not congested (below 60% of its capacity), almost any packet train rates are valid to achieve good accuracy. When the path is near congestion (above 75%), then the packet train rates must be around 50% of the AB.

Finally we have designed a tool based on our methodology: PKBest (Passive Kalman-Based estimation). We have evaluated its accuracy through simulation and compared it to that of pathChirp (a state-of-the-art PRM-based active tool). Our evaluation methodology is based on [28] where the authors presented a methodology to compare the performance of AB estimation tools. Our results show that PKBest's estimation agree with the AB and, considering all the cases, the mean relative error of PKBest is 0.12 (for pathChirp is 0.30). The evaluation has shown that our methodology is not affected by non-tight link cross-traffic, different tight and narrow link, or even multiple tight links. When the tight link is near congestion (above 75%), the error of the estimations is higher. This affects also pathChirp and, in general, PRM-based tools [28].

## References


[1] V. J. Ribeiro, M. Coates, R. H. Riedi, S. Sarvotham, and R. G. Baraniuk. "Multifractal cross traffic estimation" In Proc. of ITC spec, 2000

[2] B. Melander, M. Bjorkman, and P. Gunningberg, "A New End-to-End Probing

and Analysis Method for Estimating Bandwidth Bottlenecks” In Global Internet Symposium, 2000.

[3] M. Jain and C. Dovrolis, ”Pathload: A Measurement Tool for End-to-End Available Bandwidth” In Passive and Active Measurements, Fort Collins, CO, March 2002.

[4] N. Hu and P. Steenkiste, ”Evaluation and Characterization of Available Bandwidth Techniques” IEEE JSAC Special Issue in Internet and WWW Measurement, Mapping, and Modeling, 2003.

[5] V. J. Ribeiro, R. H. Riedi, R. G. Baraniuk, J. Navratil, and L. Cottrell, ”pathChirp: Efficient Available Bandwidth Estimation for Network Paths” In Passive and Active Measurement Workshop, 2003.

[6] J. strauss et al, ”A Measurement Study of Available Bandwidth Estimation tools” in Proc. Internet Measurement Conference (IMC) 2003.

[7] Svante Ekelin et al, Real-Time Measurement of End-to-End Available Bandwidth using Kalman Filtering, NOMS 2006

[8] Mradula Neiginhal et al, ”Measuring Bandwidth Signatures of Network Paths” IFIP Networking 2007

[9] Yu Cheng et al, ”New Exploration of Packet-Pair Probing of Available Bandwidth Estimation and Traffic Characterization” IEEE ICC 2007

[10] C. Dovrolis et al, ”Packet-Dispersion Techniques and a Capacity-Estimation Methodology” IEEE/ACM Transactions On Networking, Vol. 12, No. 6, 2004

[11] Rudolph Emil Kalman, ”A New Approach to Linear Filtering and Prediction Problems” Transactions of the ASME–Journal of Basic Engineering, 1960

[12] Svante Ekelin et al. ”Real-Time Measurement of End-to-End Available Bandwidth using Kalman Filtering” IEEE NOMS 2006

[13] R.W. Wolff, ”Poisson arrivals see time averages” Operations Research 30 (1982), pp. 223-231.

[14] F. Baccelli et al, ”The Role of the PASTA property in Network Measurement” SIGCOMM 2006.

[15] Cooperative Association for Internet Data Analyisis NASA Ames Internet Exchange Packet Length Distributions

[16] Alok Shriram and Jasleen Kaur, ”Empirical Evaluation of Techniques for Measuring Available Bandwidth” IEEE INFOCOM 2007.

[17] Erik Hartikainen and Svante Ekelin, ”Tuning the Temporal Characteristics of a Kalman-Filter Method for End-to-End Bandwidth Estimation” IEEE E2EMON 2006

[18] R.Prasad and M.Jain, ”Effects of Interrupt Coalescence on Network Measurements” Passive and Active Measurements (PAM) conference, April 2004.

[19] Xiliang Liu, Kaliappa Ravindran, and Dmitri Loguinov, "A Queueing-Theoretic Foundation of Available Bandwidth Estimation: Single-Hop Analysis" IEEE/ACM Transactions on Networking, 2007

[20] L.Lao and M.Y.Sanadidi, "The Probe Gap Model can Underestimate the Available Bandwidth of Multihop Paths" In the ACM SIGCOMM Computer Communications Review, October 2006.

[21] Cao Le Thanh Man et al, "A Merged Inline Measurement Method for Capacity and Available Bandwidth" PAM 2005

[22] M. Zangrilli et al, "pplying Principles of Active Available Bandwidth Algorithms to Passive TCP traces" PAM 2005

[23] S.Katti et al, "MultiQ: Automated Detection of Multiple Bottleneck Capacities Along a Path" IMC 2004

[24] D. P. Pezaros, D. Hutchison, R. D. Gardner, F. J.Garcia, J. S. Sventek, "Inline Measurements: A Native Measurement Technique for IPv6 Networks" INCC 2004

[25] Z. Zhang, V.J. Ribeiro, S.B. Moon, C. Diot, "Small-time scaling behaviors of Internet backbone traffic: an empirical study" IEEE INFOCOM 2003

[26] NIST/SEMATECH e-Handbook of Statistical Methods, http://www.itl.nist.gov/div898/handbook/ Sec. 1.3.5.6

[27] M.Jain, C.Dovrolis, "Pathload: A measurement tool for end-to-end available bandwidth" PAM 2002

[28] Alok Shriram et al, "Empirical Evaluation of Techniques for Measuring Available Bandwidth" IEEE INFOCOM 2007

[29] Passive Measurement and Analysis (PMA), http://pma.nlanr.net/Special/ (Last seen on 19th March 2008)

[30] M. Jain and C. Dovrolis, "End-to-End Estimation of the Available Bandwidth Variation Range" ACM SIGMETRICS Conference, June 2005

[31] J. Kilpi and I. Norros, "Testing the Gaussian Approximation of Aggregate Traffic" ACM/USENIX Internet Measurement Workshop (IMW), 2002

[32] pathChrip Software, http://www.spin.rice.edu/Software/pathChirp/ (Last seen on 19th March 2008)

[33] Rene Serral-Gracia, Albert Cabellos-Aparicio, Jordi Domingo-Pascual, "Network performance assessment using adaptive traffic sampling" IFIP Networking 2008