IGMPS, a new tool for estimating end-to-end available bandwidth in IP network paths

Ahmed Ait Ali, Francis Lepage

IGMPS, a New Tool for Estimating End-to-End Available Bandwidth in IP Network Paths

Ahmed Ait Ali, Francis Lepage
CRAN - Nancy University BP 239
54500 Vandoeuvre Cedex
France
e-mail: {ahmed.aitali, francis.lepage}@cran.uhp-nancy.fr

***Abstract*— This paper presents a new end-to-end available bandwidth measurement tool called IGMPS (Improved Gap Model using Packet Size parameter). IGMPS is a lightweight tool based on a new deterministic model of packet delays derived from the Probe Gap Model which uses packet pair technique for actively assessing bottleneck link characteristics. In a previous work that presented a detailed analysis of the capabilities and limitations of packet dispersion techniques, it is showed that probing packet size parameter has a considerable effect on the measurement accuracy. Based on this insight, the model implemented in IGMPS takes into account this parameter and introduce it in available bandwidth estimation formula. Through measurements on several network testbed configurations, we evaluated IGMPS in terms of accuracy, convergence time and intrusiveness and found that IGMPS provides available bandwidth estimates with high accuracy. We investigated the effect of several deciding factors on IGMPS measurement accuracy and found that when the probing packet size is equal or close enough to the cross traffic packet size, IGMPS estimates available bandwidth with a very high accuracy. The comparison of IGMPS to Spruce showed that our tool provides better quality of measurements and it clearly outperforms Spruce.**

***Keywords*— Available bandwidth, Network measurement, Measurement techniques and tools.**

I. INTRODUCTION

The available bandwidth of a link is the unused capacity of that link during a certain time period. At any specific instant in time, a link is either transmitting at the full link capacity or it is idle, so the instantaneous utilisation of a link is either 0 or 1. the definition of instantaneous available bandwidth is meaningless, and a certain time-averaging is needed. The available bandwidth $A_i(t,T)$ of a link i at time t is defined as:

$$A_i(t,T) = \frac{1}{T}\int_t^{t+T} C_i\left(1-u_i\left(t\right)\right)dt$$

Where C_i is the link's capacity and $u_i(t)$ is the percentage of link utilization. The available bandwidth of a path P made of N links $LI_1, LI_2, ...LI_N$ (no routing changes or multipath forwarding occur during the measurement) is defined as:

$$A = \min_{i=1...N}\left\{C_i\left(1-u_i\right)\right\} = \min_{i=1...N} A_i$$

To measure available bandwidth many different active probing techniques have been developed. The packet pair technique is one of the most popular of them. The basic idea of packet pair paradigm is that the sender sends pairs of packets to a receiver. The pair packets are sent close enough together in time to cause the packets to queue together at the bottleneck link. By measuring the changes in the packet spacing, the receiver can estimate the amount of cross traffic that enter between the tow packets at the bottleneck link and then infers bandwidth proprieties of the network path [2], [9], [15].

In this paper, we develope a new deterministic model of packet pair delays for end-to-end available bandwidth estimation in IP network paths. This model captures the relationship between the amount of cross traffic and the packet pair gap taking into account the size of each probing pair packet. Based on this deterministic model and considering a particular case obtained when the two packets of the probing pairs are of the same size, we developed an active measurement tool called IGMPS (Improved Gap Model using Packet Size parameter) that estimates the end-to-end available bandwidth at the bottleneck link of the path. Based on an experimental analysis in an isolated testbed configuration, we present IGMPS measurement results that verify the tool behavior and illustrate its accuracy. Our experiments show that introducing the packet size parameter in the probe gap model formula improves considerably the accuracy of the measurements. We show that using probe packets whose length is equal or close enough to cross traffic packet size increases the tool accuracy.

In [1], authors have compared several available bandwidth measurement tools and concluded that Spruce is the tool that offers the best performance. Based on this insight, we compare IGMPS to Spruce and show that our tool achieves better performance.

This paper is organized as follows. First, we discuss the related work in section II. In section III, we introduce our deterministic model to estimate available bandwidth based on the probe gap model paradigm. The IGMPS implementation issue is introduced in section IV. In section V, we present our performance evaluation methodology and the first results on IGMPS measurements. We discuss the effect of the used probe parameters in section VI. We compare IGMPS performance to Spruce in section VII and we conclude in section VIII.

II. RELATED WORK

To assess and monitor bandwidth metrics, many measurement techniques and methodologies have been developed and

used in several existing software tools. This section aims to describe the most commonly used techniques to measure end-to-end available bandwidth in network paths and provides some examples of tools using them such as Spruce, Pathload, IGI and Pathchirp.

Available bandwidth estimation techniques can be divided into tow categories: self induced congestion based techniques and cross traffic estimation based techniques. Self-induced congestion based techniques assume FIFO queuing at all routers along the path, cross traffic follows a fluid model and average rates of cross traffic change slowly. If a source sends probes to a destination at a rate R less than A, probes will experience similar delays. On the other hand, if R is greater than A, probes will queue in the network and experience increasing delays. This technique is based on the observation that the delays of successive probing packets increase when the probing rate exceeds the available bandwidth in the path. It consists in probing the network at different rates and detecting (at the destination) the point at which delays start to increase. At this point, probing rate is equal to the available bandwidth. Pathload [3], [12], and PathChirp [5] implement this technique.

Pathload uses the network self congestion paradigm called SLoPS (Self Loading Periodic Streams) for available bandwidth estimation. Network congestion is obtained by sending packet streams to a destination with increasing bit rate. When the trend of the packet's one-way delays is found to be increasing the congestion is detected and available bandwidth is evaluated. Pathload algorithm converges by dichotomy to the available bandwidth value. Pathchirp uses the same method as Pathload. It sends exponential flight pattern of probes called a "chirp" for causing the self induced congestion on the network path. By rapidly increasing the probing rate within each chirp, Pathchirp obtains a rich set of information from which it dynamically estimates available bandwidth. The main advantage of using exponentially spaced chirps is the reduction of the probing traffic load.

Cross traffic estimation based techniques assume that the capacity C of the path is known (it can be measured easily with one of the capacity measurement tools such as Nettimer [7], Cprobe [8], ...etc) and the bottleneck link is both the narrow and the tight link along the path. These techniques and tools are based on the Probe Gap model (PGM) [2] that consists in capturing the relationship between the dispersion of a packet-pair and the cross traffic rate C_T at the bottleneck link of a path [9]. They begin by estimating the cross traffic at the bottleneck and then compute the available bandwidth as the difference between the path capacity and the cross traffic rate:$A = C - C_T$ These techniques are implemented in IGI [4] and Spruce [2].

IGI uses a sequence of packet pair trains. It sends each train with increasing inter-packet dispersion until it reaches the turning point. This point is obtained when the initial gap at the sender is equal to the output gap at the receiver. This method assumes that at the turning point, the noise introduced by cross traffic becomes zero mean and the probing rate is equal to the available bandwidth of the path. On the other hand, Spruce implements the Probe Gap Model (PGM). It sends a Poisson process of packet pairs to a destination. By measuring packet dispersion at the receiver, Spruce collects individual samples Ai of available bandwidth:

$$A_i = C\left(1 - \frac{y_i - x}{x}\right)$$

Where x is the initial inter-packet dispersion at the sender and y_i is the measured inter-packet dispersion at the receiver. The algorithm averages samples A_i to obtain a running estimate of the available bandwidth:

$$A = \sum_{i=1}^{n} \frac{A_i}{n}$$

To reduce the probing traffic load, Spruce adjusts the average inter-packet gap to ensure that the probe rate is a minimum of 240 kb/s an 5% of the end-to-end path capacity.

In [1], authors have compared and analyzed the performance of Spruce, Pathload, IGI and Pathchirp in an isolated testbed configuration. Under the conditions of their experiments authors found that Spruce is the fastest, the most accurate tool and one of the least intrusive.

III. Available bandwidth characterization

In this section we explain in detail our deterministic model based on the packet pair methodology to model the available bandwidth in a multiple link path. Before starting, we model first the path capacity based on two different delay equations and then according to the probe gap model we give a formula that models the end-to-end available bandwidth in the path. Our model is based on the packet pair paradigm, it is then imperative to make the following assumptions:

1) Cross traffic follows a fluid model and changes slowly.
2) The routers of the path use FIFO-queuing service and are store-and-forward.

Notations and different parameters used in the remain of this paper are defined in table 1:

Table 1. Parameter definitions.

Parameter	Parameter definition
n	hop length of the path
d_i	latency of the link i
C_i	capacity of the link i
s^k	size of the packet k
t_l^k	arrival time of the packet k to the link l
q_l^k	queuing delay of the packet k at the link l
b	bottleneck link

According to the reasoning given in [10], the path capacity model is derived from two delay equations: the arrival time equation and the queuing delay equation.

The arrival time equation predicts the time needed for a packet to travel across l link in the path before reaching the

destination node. The arrival time t_l^k of a packet k at the link l is given using the following equation:

$$t_l^k = t_0^k + \sum_{i=0}^{l-1} \left(\frac{s^k}{C_i} + d_i + q_i^k \right) \tag{1}$$

Were t_0^k is the transmission time and $(0 < i < l)$.

The queuing delay q_l^k is given by the following equation:

$$q_l^k = max\left(0, t_{l+1}^{k-1} - d_l - t_l^k\right) \tag{2}$$

Using equation (1) et (2), the packet delay in a multiple link path is expressed by:

$$t_n^k = t_0^k + \sum_{i=0}^{n-1} \left(\frac{s^k}{C_i} + d_i + max\left(0, t_{i+1}^{k-1} - d_i - t_i^k\right) \right) \tag{3}$$

Assuming that there is one bottleneck link (with capacity $C_b = C$) and no queuing occurs elsewhere. The second part of the equation (3) can be split into the time to travel from the source to the bottleneck, the queuing time at the bottleneck and the time to travel from the bottleneck to destination. Equation (3) becomes then :

$$\begin{aligned} t_n^k = \left[t_0^k + \sum_{i=0}^{b-1} \left(\frac{s^k}{C_i} + d_i \right) \right] + \left[\frac{s^k}{C} + t_{b+1}^{k-1} - t_b^k \right] \\ + \left[\sum_{i=b+1}^{n-1} \left(\frac{s^k}{C_i} + d_i \right) \right] \end{aligned} \tag{4}$$

With $(0 < i < n)$.

Since the queuing occurs only at the bottleneck link and using (3) we obtain

$$t_0^k + \sum_{i=0}^{b-1} \left(\frac{s^k}{C_i} + d_i \right) = t_b^k$$

Substituting in (4) we obtain:

$$t_n^k = t_b^k + \frac{s^k}{C} + t_{b+1}^{k-1} - t_b^k + \sum_{i=b+1}^{n-1} \left(\frac{s^k}{C_i} + d_i \right)$$

$$t_n^k = \left(\frac{s^k}{C} \right) + t_{b+1}^{k-1} + \sum_{i=b+1}^{n-1} \left(\frac{s^k}{C_i} + d_i \right) \tag{5}$$

Using (3), t_{b+1}^{k-1} is given as:

$$t_{b+1}^{k-1} = t_0^{k-1} + \sum_{i=0}^{b} \left(\frac{s^{k-1}}{C_i} + d_i \right)$$

Equation (5) becomes then:

$$t_n^k = \left(\frac{s^k}{C} \right) + \sum_{i=0}^{b} \left(\frac{s^{k-1}}{C_i} + d_i \right) + t_0^{k-1} + \sum_{i=b+1}^{n-1} \left(\frac{s^k}{C_i} + d_i \right)$$

$$\begin{aligned} t_n^k = \left(\frac{s^{k-1}}{C} \right) + \sum_{i=0}^{b-1} \left(\frac{s^{k-1}}{C_i} \right) + \sum_{i=b}^{n-1} \left(\frac{s^k}{C_i} \right) \\ + t_0^{k-1} + \sum_{i=0}^{n-1} (d_i) \end{aligned} \tag{6}$$

Since we have

$$\sum_{i=b+1}^{n-1} \left(\frac{s^k}{C_i} \right) = \sum_{i=0}^{n-1} \left(\frac{s^k}{C_i} \right) - \sum_{i=0}^{b} \left(\frac{s^k}{C_i} \right)$$

then by substituting in (6) we obtain:

$$\begin{aligned} t_n^k = \left(\frac{s^k}{C} \right) + \left(s^{k-1} - s^k \right) \sum_{i=0}^{b} \left(\frac{1}{C_i} \right) + \sum_{i=0}^{n-1} \left(\frac{s^k}{C_i} \right) \\ + \sum_{i=0}^{n-1} (d_i) + t_0^{k-1} \end{aligned}$$

$$\begin{aligned} t_n^k = \left(\frac{s^{k-1}}{C} \right) + \left(s^{k-1} - s^k \right) \sum_{i=0}^{b-1} \left(\frac{1}{C_i} \right) + \sum_{i=0}^{n-1} \left(\frac{s^k}{C_i} \right) \\ + \sum_{i=0}^{n-1} (d_i) + t_0^{k-1} \end{aligned} \tag{7}$$

For more compact notation, we define the following variable:

$$d^l = \sum_{i=0}^{l} d_i \qquad \frac{1}{C^l} = \sum_{i=0}^{l} \left(\frac{1}{C_i} \right)$$

From (7) we derive the capacity C as :

$$C = \frac{s^{k-1}}{t_n^k + \frac{(s^k - s^{k-1})}{C^{b-1}} - \frac{s^k}{C^{n-1}} - t_0^{k-1} - d^l} \tag{8}$$

Equation (8) gives the bottleneck link capacity in the path. It is expressed using several parameters which directly influence the estimated capacity C. In the following, we give a generic formula to estimate available bandwidth on the end-to-end path. Then, by experiments in sections V and VII, we show that the obtained model provides better measurement results.

A. Available bandwidth model

The packet pair technique consists in capturing the relationship between the inter-packet gap of a packet pair and the cross traffic rate at the bottleneck link. A probe pair is sent from a sender with a time gap $\Delta_{in} = t_0^k - t_0^{k-1}$, and reaches the receiver with a time gap $\Delta_{out} = t_n^k - t_n^{k-1}$.

Δ_{out} is the time taken by the bottleneck to transmit the second packet of the pair and the cross traffic that arrived during Δ_{in} . Using the Probe Gap model, the available bandwidth at the bottleneck link is estimated as:

$$A = C\left(1 - \frac{\Delta_{out} - \Delta_{in}}{\Delta_{in}}\right) \tag{9}$$

Substituting (8) in (9) we obtain:

$$A = \frac{s^{k-1}\left(2t_0^k + t_n^{k-1} - 2t_0^{k-1} - t_n^k\right)}{\left(t_0^k - t_0^{k-1}\right)\left[t_n^k + \frac{(s^k - s^{k-1})}{C^{b-1}} - \frac{s^k}{C^{n-1}} - t_0^{k-1} - d^l\right]} \quad (10)$$

The one-way delay (OWD) of a probing packet in the path is the accumulation of the total transmission delays $\sum_{i=0}^{n-1}\left(\frac{s^k}{C_i}\right)$, the link queuing delay q_i and the link latency d_i:

$$OWD = \sum_{i=0}^{n-1}\left(\frac{s^k}{C_i}\right) + \sum_{i=0}^{n-1} q_i + \sum_{i=0}^{n-1} d_i$$

$$OWD = \sum_{i=0}^{n-1}\left(\frac{s^k}{C_i}\right) + \sum_{i=0}^{n-1} q_i + d^l$$

then

$$d^l = \sum_{i=0}^{n-1} d_i = OWD - \sum_{i=0}^{n-1}\left(\frac{s^k}{C_i}\right) - \sum_{i=0}^{n-1} q_i \quad (11)$$

Assuming that there is only one queuing link which is the bottleneck link and no queuing occurs elsewhere, then:

$\sum_{i=0}^{n-1} q_i = q_b$ where q_b represents the queuing delay at the bottleneck link. Equation (11) becomes:

$$\sum_{i=0}^{n-1} d_i = OWD - \sum_{i=0}^{n-1}\left(\frac{s^k}{C_i}\right) - q_b$$

In the previous assumptions we assumed that the second packet of the probe pair arrives to the bottleneck before the departure of the first packet. Since that, we can model the delays of the two packets as shown in the Figure 1.

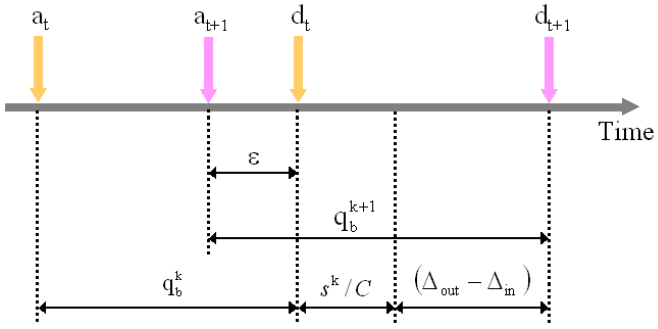

Fig. 1. Modeling packets arrival and departure in the bottleneck queue. a_i is the arrival time, d_i the departure time and q_i the queuing delay.

From this figure we can define the following equation:

$$q_b^k = \epsilon + \left(\frac{s^{k-1}}{C}\right) + (\Delta_{out} - \Delta_{in}) \quad (12)$$

Assuming that ϵ is too small so we can set it equal to 0, then equation (12) becomes:

$$q_b^k = \left(\frac{s^{k-1}}{C}\right) + (\Delta_{out} - \Delta_{in}) \quad (13)$$

Substituting in (11) using (13) we obtain:

$$\sum_{i=0}^{n-1} d_i = OWD - \sum_{i=0}^{n-1}\left(\frac{s^k}{C_i}\right) - \left(\frac{s^{k-1}}{C}\right) - \left(t_n^k - t_n^{k-1} - t_0^k + t_0^{k-1}\right)$$

The one-way delay OWD is given as:

$$OWD = t_n^k - t_0^k$$

Then, equation (10) becomes:

$$d^l = \sum_{i=0}^{n-1} d_i = t_n^{k-1} - t_0^{k-1} - \sum_{i=0}^{n-1}\left(\frac{s^k}{C_i}\right) - \left(\frac{s^{k-1}}{C}\right) \quad (14)$$

Substituting in (10) with (14) we obtain

$$A = \frac{s^{k-1}\left(2t_0^k + t_n^{k-1} - 2t_0^{k-1} - t_n^k\right)}{\left(t_0^k - t_0^{k-1}\right)\left[\frac{(s^k - s^{k-1})}{C^{b-1}} + \frac{s^{k-1}}{C} + \left(t_n^k - t_n^{k-1}\right)\right]} \quad (15)$$

In formula (15), the end-to-end available bandwidth in the path is defined as a function of several deciding parameters such as packet size, the bottleneck link capacity and packet departure and arrival times. If we consider the particular case where the two packets of the pair have equal sizes ($s^k = s^{k-1} = S$), then the end-to-end available bandwidth of the path is given by:

$$A = \frac{S\left(2\Delta_{in} - \Delta_{out}\right)}{(\Delta_{in})\left[\left(\frac{S}{C}\right) + \Delta_{out}\right]} \quad (16)$$

Equation (16) is implemented in our measurement tool. The design and performance evaluation of this tool are described in the following sections.

IV. IGMPS Design and Implementation

Compared to the Probe Gap Model described in [2], our model takes into account the packet size parameter that is very important when measuring available bandwidth and that impacts heavily on the results.

Formula (16) is a particular case of the equation (15). It is obtained by setting the first packet size equal to second packet size on the packet pair, it defines the end-to-end available bandwidth at the bottleneck link. We have implemented this formula in a measurement tool called IGMPS (Improved Gap Model using Packet Size parameter) developed in C language under Linux environment.

IGMPS is a one-way measurement tool, it is composed of two distinct parts, the sender program and the receiver program. Assuming that the bottleneck capacity C is known (it can be estimated using one of the well known capacity measurement tools such as Nettimer [7], Cprobe [8], ...etc) IGMPS sets the initial inter-packet gap Δ_{in} at the sender and measures the output gap Δ_{out} at the receiver.

IGMPS sets the pair inter-packet dispersion at the sender equal to the transmission time of 1500B data packet on the bottleneck link (for instance, in 10Mb/s path the inter-packet

gap is about 1.2 ms). This inter-packet dispersion time is sufficient to ensure that cross traffic packets arrive at the bottleneck queue between the tow probe packets of a pair. This will ensure that the bottleneck queue does not empty between the departure of the first and the second packet of the pair. The initial inter-packet gap must not be too large to ensure that the two packets of a pair queue together at the bottleneck link. At the receiver, IGMPS measures the output inter-packet gap that is equal to the transmission time of 1500B data packet and cross traffic packets that are arrived at the bottleneck queue between the first and the second packet of the pair. Analyzing this information and knowing both the bottleneck link capacity and the size of each probing packet, IGMPS calculates available bandwidth and gives one sample measurement following formula (16).

IGMPS probe packets travel one-way from the sender to the receiver. The IGMPS sender sends a Poisson process of packet-pairs of 1500B UDP packets. So the dispersion between to pairs follows an exponentially distributed function with an average much larger than Δ_{in} (about 100 ms). The receiver performs the available bandwidth estimation and echo back the result to the sender. The Poisson process sampling makes IGMPS to be non-intrusive compared to certain tools that use packet trains instead of sending a sequence of probing packet pairs, such as Pathload [3] and Pathchirp [5].

To cope with the burstiness of cross traffic and to improve the measurement accuracy, IGMPS performs a sequence of k packet-pair measurements and computes the available bandwidth at time t as the average of the last k sample measurements. IGMPS sets the default value of k to 100.

V. Performance evaluation

This section shows some measurement results and validates IGMPS in terms of accuracy, response time and intrusiveness. Afterwards, it is compared to Spruce that uses the Probe Gap Model. For that purpose, we ran several experiments on an isolated testbed configuration using IGMPS to measure available bandwidth of the network path. The topology is shown in Figure 2 where Ps and Pd are respectively the probing source and destination. This path comports three Cisco 1700 series routers. The capacity of the bottleneck situated at the second router, is 10 Mb/s and all of the other links have a 100 Mb/s bandwidth. IGMPS measurement tool consists of separate user-level sender and receiver parts.

The sender part is set up on Ps and the receiver part on Pd. Cs and Cd are used to generate cross traffic using D-ITG traffic generator tool [16] . The traffic analyzer Ethereal is installed both on P_d and C_d. It is used to report measurement traffic load and convergence times of IGMPS and also to verify the cross traffic throughput generated by the traffic generator. In order to evaluate the accuracy and performance of our tool, we considerate three different scenarios regarding the generated cross traffic pattern. In the first and second scenarios, the cross traffic is respectively a constant bit rate UDP and TCP traffic generated with constant packet size. However, in the third

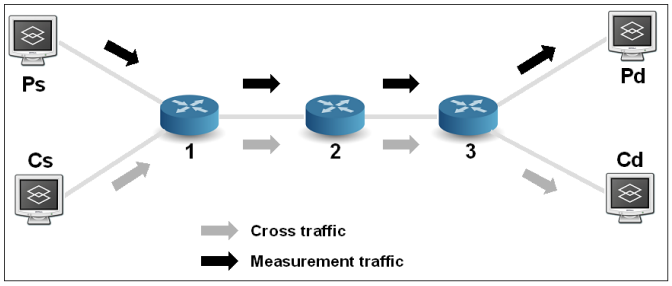

Fig. 2. Available Bandwidth Measurement Testbed.

scenario we use a TCP traffic generated with an exponentially distributed Inter Departure Time (IDT) and a constant packet size.

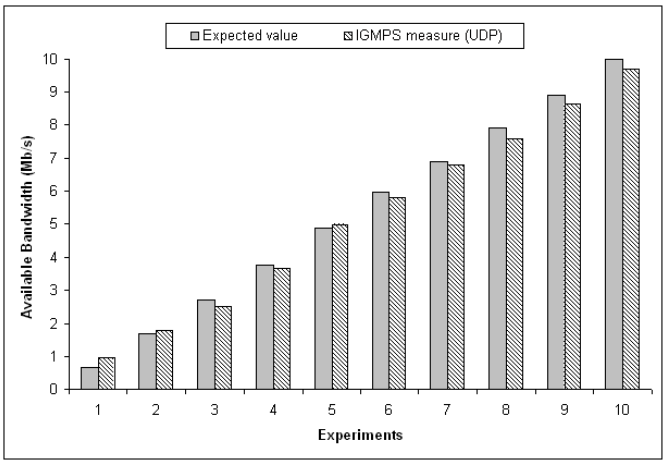

Fig. 3. IGMPS measurement results in 10 Mb/s path with UDP cross traffic.

The data presented in this section is collected using IGMPS tool for available bandwidth measurement along the path (Ps, Pd). Varying the cross traffic rate from 0 to 10 Mb/s in a 10 Mb/s path will make the available bandwidth vary from 10 to 0 Mb/s. Under each scenario, The experiments are undertaken by increasing the cross traffic rate with 1 Mb/s increments in each measurement session. A total of 30 experiments were run for each value of the available bandwidth. The measurements were collected and the averages of the results obtained with IGMPS for each cross traffic pattern are presented in figure 3, figure 4 and figure 5. Figure 3 shows IGMPS results in an end-to-end path with a periodic UDP cross traffic. This figure shows that IGMPS closely tracks the available bandwidth and correctly responds to the cross traffic variation.

IGMPS results with a periodic TCP cross traffic are depicted in figure 4. This figure shows that with such cross traffic profile, IGMPS measures the available bandwidth with high accuracy and responds correctly to the cross traffic variation. IGMPS results under the third scenario that considers an end-to-end path with exponentially distributed IDT are depicted in figure 5. As like as the first and second scenarios, results obtained in this case show that IGMPS achieves good perfor-

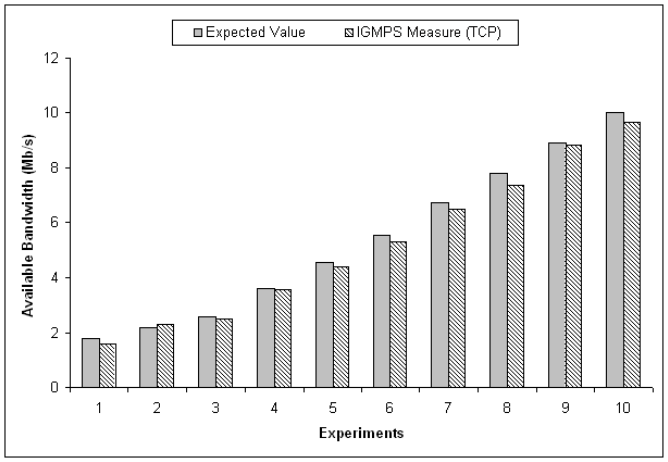

Fig. 4. IGMPS measurement results in 10 Mb/s path with TCP cross traffic.

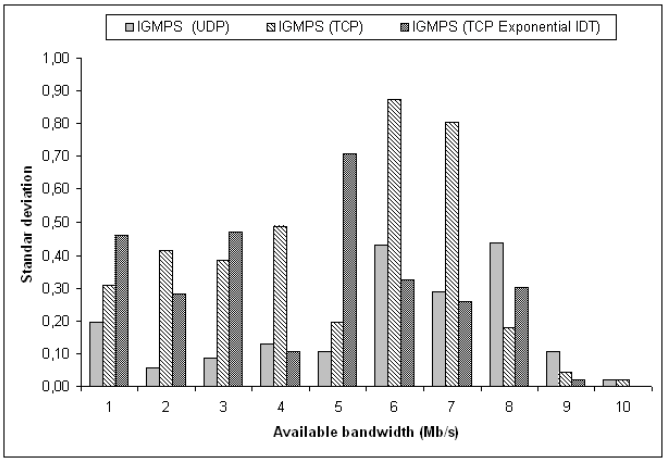

Fig. 6. The standard deviation of IGMPS measurements with UDP and TCP scenarios .

mance in terms of measurement accuracy.

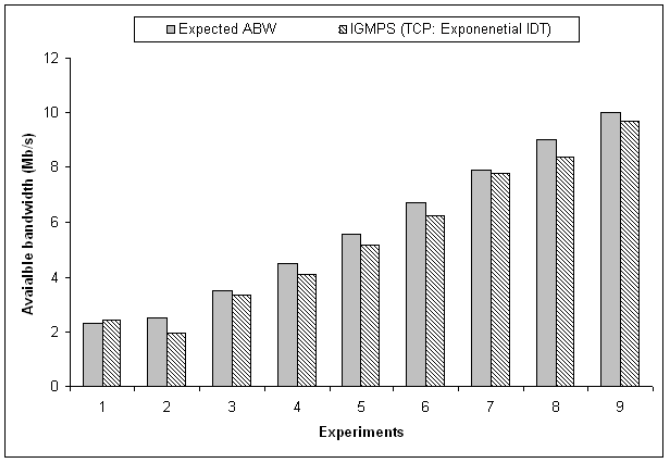

Fig. 5. IGMPS measurement results in 10 Mb/s path with TCP cross traffic and Exponentially distributed IDT.

The standard deviation values of IGMPS measurements according to each scenario are shown in figure 6. These histograms show that the averages of available bandwidth measurements in the second and the third scenarios are given with high standard deviation values, which means that IGMPS hugely underestimates or overestimates the available bandwidth in some cases. Since the experiments are performed under the same conditions for the two scenarios, results obtained on the standard deviation show that IGMPS presents a low measurement repeatability. however, in the first scenario, IGMPS presents reasonable standard deviation values.

Figure 7 shows IGMPS relative measurement error under each scenario. the relative error is defined as:

$$rel\_err = \frac{|abw\_exp - abw\_est|}{abw\_exp}$$

Where abw_est is the available bandwidth estimates generated by IGMPS tool, and abw_exp is the expected available bandwidth obtained by varying the cross traffic in the path. From figure 8, we observe that, under the first scenario (UDP cross traffic), with the exception of the first estimate, the measurement error on IGMPS is below 10% and in most cases the error is less than 5%. Results show that IGMPS is very accurate when the bottleneck link utilization is low. However, when the bottleneck link utilization is high, IGMPS produces less accurate measurements that are over-estimated by around 48%. Under the second scenario, IGMPS performs more accurate measurements. It presents a very low relative error that is less than 5% in almost cases. However, Under the third scenario (TCP cross traffic with exponentially distributed IDT), IGMPS is slightly less accurate with a relative error that is about 10%.

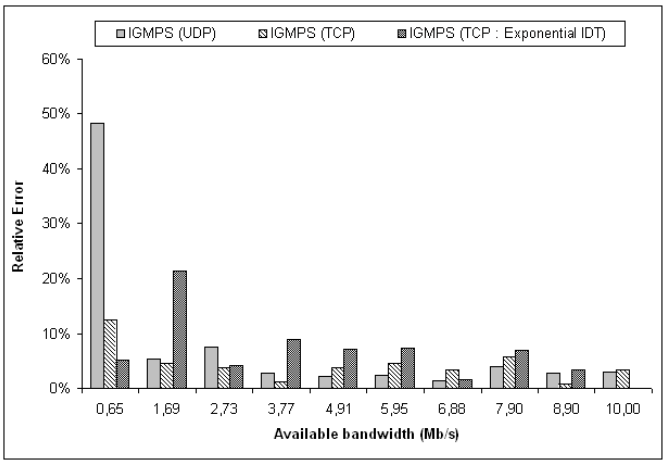

Fig. 7. IGMPS accuracy represented by the relative error. With the exception of the two first estimates IGMPS is very accurate.

A. Intrusiveness and response time

To highlight IGMPS performance in terms of convergence time and measurement traffic load, we compare it to the

well known tool Pathload. This tool uses a self-congestion technique to infer the available bandwidth of the stressed end-to-end path. This technique results generally in high convergence time and huge measurement traffic load.

Available bandwidth is a parameter that varies over time. It is therefore essential to measure it as fast as possible. To test IGMPS we repeat the experiments 30 times. The convergence time is therefore the average of the 30 measurements response time. Each IGMPS response time is the necessary time to send and process k packet pairs (in this case $k = 100$), so this convergence time closely depends on the length of the probing pair train. Results of IGMPS convergence times are depicted in figure 8. this figure shows that, compared to Pathload, IGMPS presents a short and stable convergence time which is about 10s.

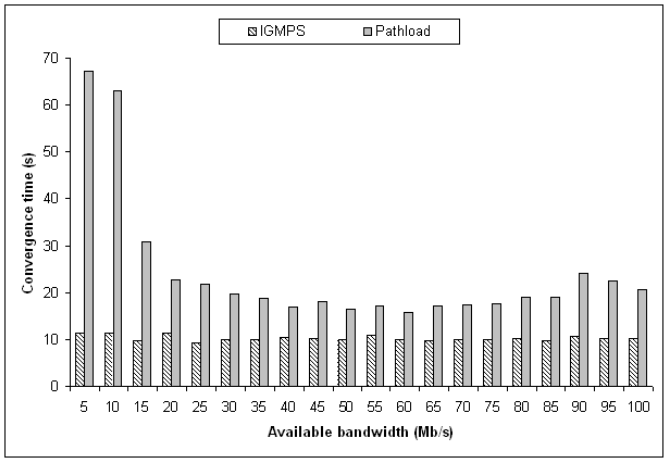

Fig. 8. IGMPS response time (in seconds) compared to Pthload convergence time. The convergence time of IGMPS is stable and it is about 10s.

In some cases IGMPS is 7 times faster than Pathload. The long measurement time of Pathload is due to its convergence algorithm. Indeed, Pathload monitors changes in the one way delay of the probing packets in order to determine the relationship between probing speed and available bandwidth. Pathload uses a dichotomic convergence algorithm, so the convergence process is slow down when the probing packets experience different levels of congestion which leads to obtain long probing times as shown in figure 8. In contrast, the convergence of IGMPS is determined directly by the number of probing packet pairs transmitted from the sender to the receiver which makes IGMPS convergence time closely depends on the parameter k.

We define tool intrusiveness as the ratio of the average tool measurement traffic load to the path capacity. The measurement traffic load depends on the probe packet size and on the number of probe packet pairs sent from the source to the destination. In our experiments, IGMPS sends 100 packet pairs to the receiver. Each packet is of 972 Bytes length. The measurement traffic load is illustrated in figure 9. This figure shows that IGMPS generates constant and low traffic load that is only about 250Kb/s. Compared to IGMPS, Pathload

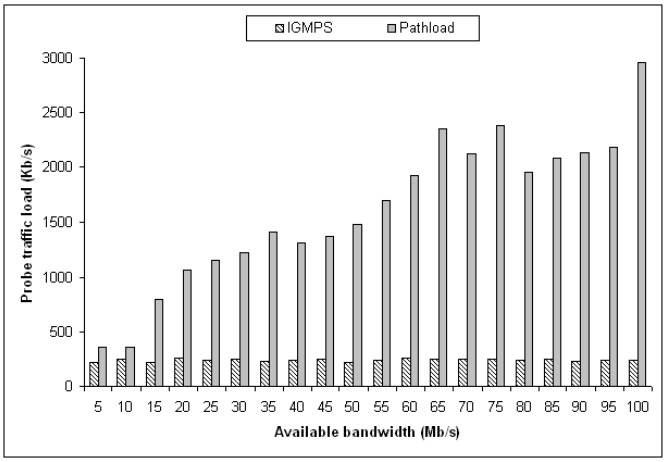

Fig. 9. IGMPS intrusiveness represented by the probe traffic load. IGMPS generates constant and low measurement traffic load that is about 250 Kb/s.

generates much more high traffic load which makes it, in some cases, about 12 times much more intrusive than IGMPS. Pathload intrusiveness is due to the SLoPS algorithm [3] that attempts to occupy all the available bandwidth to extract path characteristics. To resume, the comparison of IGMPS and Pathload showed that the former is faster and much less intrusive.

VI. Effect of IGMPS probing parameters

In this section we study the impact of the variation of some IGMPS parameters on estimation performance. (such as the probing packet size and the number of pairs in a measurement sequence).

A. Probing packet size

To study the impact of the packet size on the measurement accuracy, experiments are carried out using IGMPS with variable probing packet size ranging from 100 to 1500 bytes, results are shown in Figure 10. In this 3D figure, the y-axis that is labeled ABW represents the expected available bandwidth, the x-axis represents the probe packet size and the z-axis represents the available bandwidth measurement obtained using IGMPS.

The main observation from Figure 10 is that, when the link utilization is high and when the probing packet size varies between 100 and 700 bytes IGMPS hugely over-estimates the available bandwidth. However, it under-estimates the available bandwidth when the probing packet size is ranging from 1000 to 1300 bytes. On the other hand, when the link utilization is low, the measurement accuracy proportionally increases with the probing packet size. In both the two cases, we observe that when the packet size is between 800 and 1000 or between 1400 and 1500 bytes, IGMPS tool achieves the most accurate measurements.

As explained in section VI (IGMPS Design and Implementation) the inter-packet gap is the time delay between

the departure of the first and the second packet of the pair. This gap is function of two different parameters. The first parameter is the inter-packet dispersion set to the necessary time to the bottleneck link to transmit 1500 bytes of data, and the second parameter is the size of the second packet (in IGMPS algorithm the two packets of the pair are of equal size). The initial inter-packet gap depends then both on the packet size and on the bottleneck link capacity. With small packet size or high bottleneck link capacity, we will have small inter-packet gaps.

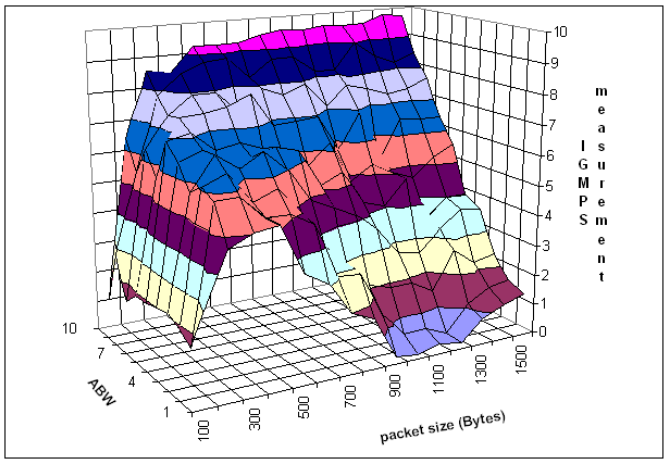

Fig. 10. Effect of IGMPS probing packet size on available bandwidth measurement with cros traffic packet size of 972 bytes.

In our experiments we have a constant bottleneck link capacity of 10Mb/s which leads to have an inter-packet dispersion of 1.2ms. The initial gap Δ_{in} depends then only on the packet size parameter. With small packet size, the resulting probe inter-packet gaps are more sensitive to interference and harder to measure accurately (especially for user-level applications). Indeed, in such cases, IGMPS has tendency to send packets with larger inter-packet dispersion ($\Delta'_{in} > \Delta_{in}$). Larger the dispersion is, the higher the amount of cross traffic packet pair will be. At the receiver, the available bandwidth is calculated using formula (16) which considers Δ_{in} parameter instead of Δ'_{in}. The disregard of the possible errors on the initial gap in IGMPS algorithm leads to underestimate the cross traffic and then to overestimate the available bandwidth in the considered end-to-end path. The measurement inaccuracy is accentuated and affects the results significantly when the link utilization is high. Results presented in figure 10 are obtained using a cross traffic with 972 bytes packet size. This figure shows that IGMPS achieves nice convergence when the probing packet size is between 900 and 1000 bytes. Based on this observation, we think that when the probing packet size is close enough to cross traffic packet size, the probe packet pair interacts more significantly with cross traffic packets.

In order to confirm this hypothesis, other experiments using smaller cross traffic packets are undertaken. In this scenario, the cross traffic injected in the network path is an UDP traffic with 460 Bytes sized packets. The results of the available bandwidth measurement are summarized in figure 11. under high link utilization, IGMPS overestimates the available bandwidth when it uses small packets (packet length between 100 and 400 Bytes) and underestimates this latter when the probe packet size is between700 and 1200 Bytes. However, when the the probe packet size is between 400 and 500 Bytes (which is very close to cross traffic packet length : 460 Bytes) IGMPS presents a nice convergence. Figure 11 shows that, also when the probe packet size is between 1300 and 1500 Bytes IGMPS measures the available bandwidth with good accuracy. These results are in agreement with those obtained in the first experiments and they show that our hypothesis is plausible enough to be considered. This section presents preliminary results. To definitely confirm our hypothesis, much more experiments under various scenarios are needed.

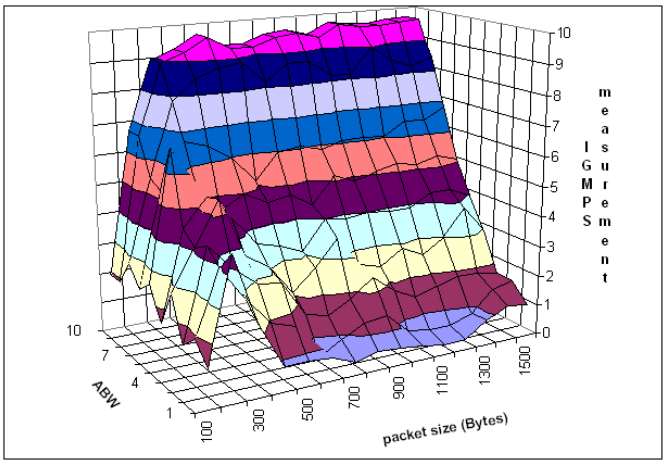

Fig. 11. Effect of IGMPS probing packet size on available bandwidth measurement with crosse traffic packet size of 460 bytes.

In IGMPS implementation, the default value of probing packet size is set to the maximum non-fragmented packet size, i.e the path Maximum Transmission Unit (MTU) size that is equal to 1500 bytes in our case. The reason of this choice is that a larger packet size leads to wider inter-packet dispersion that is easier to measure, more robust to queuing delay noise, and less sensitive to the timestamping resolution at the receiver.

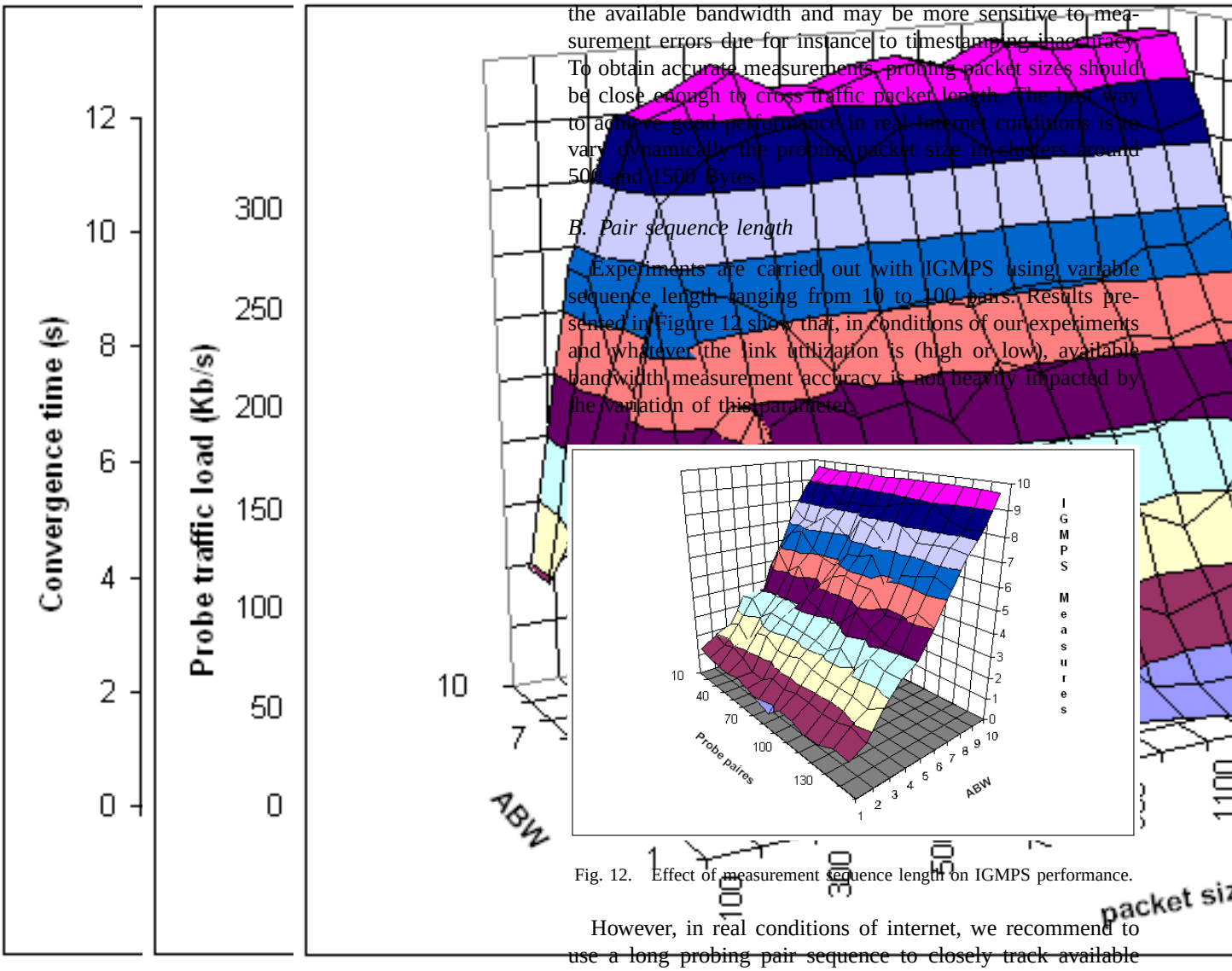

Fig. 12. Effect of measurement sequence length on IGMPS performance.

The packet size distribution in Internet has clusters around 40 Bytes, 500 Bytes and 1500 Bytes. In [17] authors show that the most of Internet packets are less than 552 Bytes length. They show also the predominance of small packets with common size of 44 Bytes and 552 bytes (TCP Maximum segment size) which include TCP acknowledgement segments, TCP control segments such as SYN, FIN and RTS packets. Based on this insight, we think that it is necessary to improve IGMPS algorithm by dynamically varying the probing packet size during the measurement. Since IGMPS sends the receiver with k packet pairs, as a future work, we plan to use (for instance) $k/2$ pairs with 400 Bytes sized packets, $k/4$ pairs with 600 Bytes sized packets and finally, $k/4$ pairs with 1500 Bytes sized packets to interact well with Internet traffic packets. We will not use probe packets that are less than 400 Bytes length because the gaps generated in this case are harder to measure accurately and therefore, IGMPS obtains inaccurate measurements of the available bandwidth. Our conclusion is that in general, small probing packets sizes may overestimates the available bandwidth and may be more sensitive to measurement errors due for instance to timestamping inaccuracy. To obtain accurate measurements, probing packet sizes should be close enough to cross traffic packet length. The best way to achieve good performance in real Internet conditions is to vary dynamically the probing packet size in clusters around 500 and 1500 Bytes.

B. Pair sequence length

Experiments are carried out with IGMPS using variable sequence length ranging from 10 to 100 pairs. Results presented in Figure 12 show that, in conditions of our experiments and whatever the link utilization is (high or low), available bandwidth measurement accuracy is not heavily impacted by the variation of this parameter.

However, in real conditions of internet, we recommend to use a long probing pair sequence to closely track available bandwidth. Indeed, using shorter pair sequence leads to get a wider range of available bandwidth estimates. The reason is that the cross traffic along the path is bursty and with a shorter pair sequence we obtain a shorter sampling interval, which leads to see a wider range of estimates. On the other hand, Figures 13 and Figure 14 show that pair sequence length has a large impact on the cost of IGMPS algorithm. This parameter directly affect the amount of measurement traffic generated by the tool and its convergence time. Indeed, both IGMPS response time and intrusiveness proportionally increase with the number of pairs in a measurement sequence.

VII. Comparison of IGMPS to Spruce tool

Lab experiments to compare the performance of IGMPS to existing tools (Spruce, Pathload, IGI and Pathchirp) are carried out in the same network configuration shown in Figure 2 but due to space constrain we can not provide the results of the all experiments of the comparative analysis. So, in this article we focus our interest on Spruce because, as IGMPS, this tool uses the packet pair technique too and, as mentioned

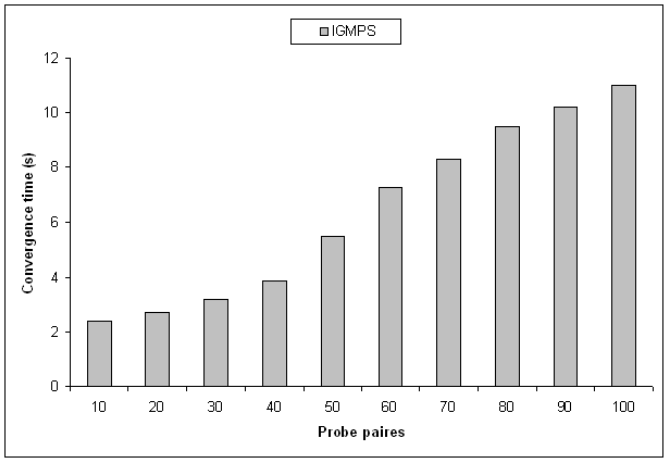

Fig. 13. Effect of measurement sequence length on IGMPS response time.

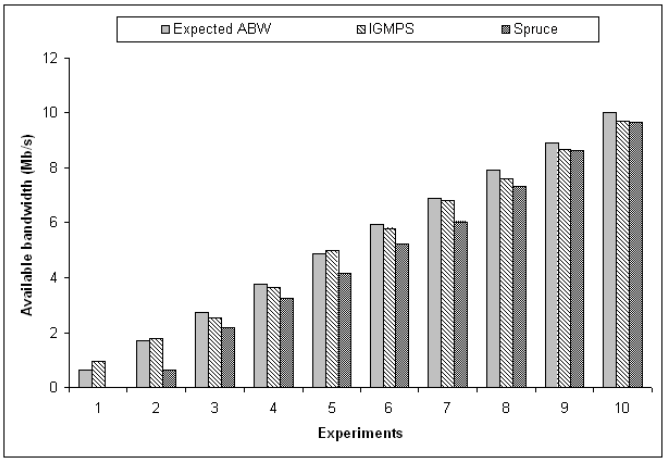

Fig. 15. Comparison of IGMPS and Spruce performance in 10 Mb/s path.

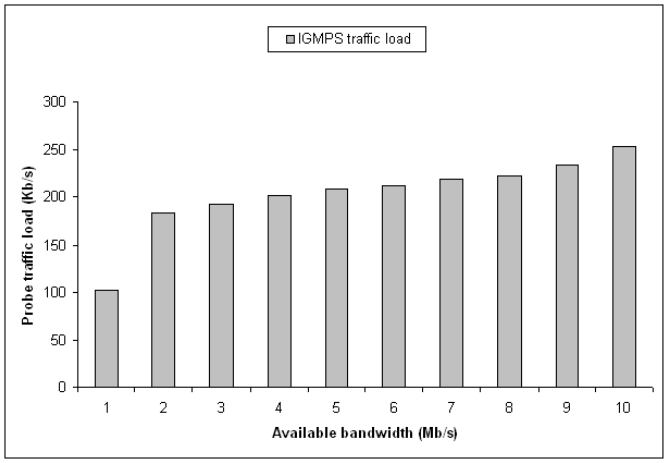

Fig. 14. Effect of measurement sequence length on IGMPS traffic load.

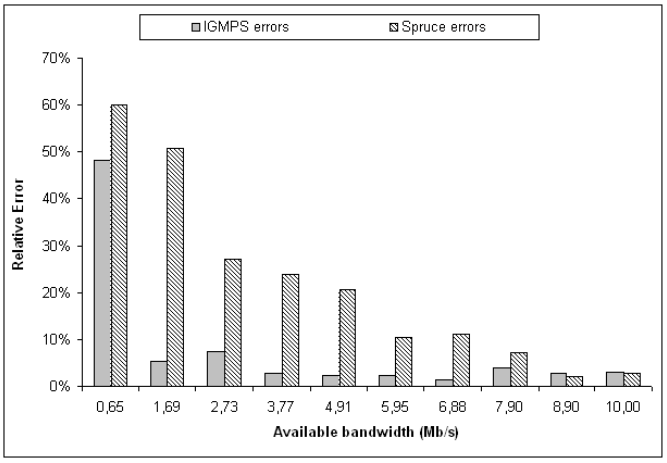

Fig. 16. Comparison of IGMPS and Spruce relative errors in 10 Mb/s path.

in section I, this tool seems to be one of the tools that offers the best performance with regards to the criteria studied in this paper (accuracy, convergence time and intrusiveness). in our experiments, we make available bandwidth ranging from 0 to 10 Mb/s. the results are depicted in Figure 15 and the relative errors of IGMPS and Spruce are illustrated in figure 16.

Results show that IGMPS reacts well to cross traffic variation and closely tracks the available bandwidth. It achieves the measurements with a very high accuracy and its relative measurement error is in some cases less than 3%. Compared to IGMPS, Spruce is less accurate. Indeed, despite the slight inaccuracy of IGMPS when the link utilization is high (in the first estimates), Spruce achieves less accurate estimates than IGMPS in each measurement session. In the conditions of our experiments, IGMPS clearly outperforms Spruce.

These experiments have shown that IGMPS achieves the best performance in terms of measurement accuracy, convergence time and intrusiveness. However, we call attention to three limitations in IGMPS design. The first one is that IGMPS requires access to both the sender part and the receiver part of the end-to-end path which limits its applicability since regular end-users often only have local access. the second one is that the deterministic model implemented in IGMPS assumes that the path routers use FIFO queuing service to schedule packets. This model may not apply to wireless networks where packets are often scheduled using non-FIFO queuing algorithms. Finally, IGMPS algorithm assumes that the cross traffic is fluid and changes slowly. This assumption is not realistic in real Internet conditions.

VIII. Conclusion

In this paper, we presented a deterministic model for estimating end-to-end available bandwidth. This model is based on the packet pair paradigm and takes into account the packet size parameter. Then, we introduced IGMPS, a tool that estimates available bandwidth based on the proposed model and compared its performance to Spruce. Results showed that IGMPS is very accurate and in the conditions of our experiments, it outperforms Spruce. Experiments showed that introducing the packet size parameter in the available

bandwidth formula improves considerably the accuracy of the measurements. Indeed, we showed that when the probing packet sizes are equal or close enough to the cross traffic packet sizes, IGMPS estimates available bandwidth with a very high accuracy. we plan to improve IGMPS measurement accuracy by dynamically varying the probing packet size in a carefully defined interval to interact well with Internet traffic packets.

The study presented in this paper was focused on a small number of criteria which seemed to be the most important. However, this study must be completed by considering other parameters and by evaluating IGMPS under real Internet conditions.

REFERENCES

[1] A. Aitali, F. Michaut, F. Lepage. End-to-end available bandwidth measurment tools: A comprative evaluation of performance. In *IPS-MoMe 2006: 4th International Workshop on Internet Performance, Simulation, Monitoring and Measurement*, Salzburg, Austria, February 27-28, 2006.

[2] J.Strauss, D. Katabi, and F. Kaashoek. A Measurement Study of Available Bandwidth Estimation Tools. In *IMC 2003: The Internet Measurements Conference*, Florida, 2003.

[3] M. Jain, C. Dovrolis. Pathload: a Measurement Tool for Available Bandwidth Estimation. In *Proceeding of PAM'02: Passive and Active Measurment*, Fort Collins, Colorado, 2002.

[4] N. Hu, P. Steenkiste. Evaluation and Characterization of Available Bandwidth Probing Techniques. In *the IEEE JSAC Special Issue in Internet Measurement, Mapping, and Modeling*, 21(6), August 2003.

[5] V. Ribeiro, R. Riedi, R. Baraniuk, J. Navratil, and L. Cottrell. PathChirp: Efficient Available Bandwidth Estimation for Network Paths. In *PAM 2003: Passive and Active Measurement Workshop*, San Diego, California, 2003.

[6] R.Prasad,M .Murray, C. Dovrolis, K. Claffy. Bandwidth Estimation: Metrics, Measurement Techniques, and Tools. *IEEE Network*, November-December 2003.

[7] K. Lai, M. Baker. Nettimer: A Tool for Measuring Bottleneck Link Bandwidth. In *USENIX Symposium on Internet Technologies and Systems*, March 2001.

[8] R. Carter M. Corvella. Measuring bottlenck link speed in packet-switched networks. *Technical report 1996-006*, Boston University, March 1996.

[9] B. Melander, M. Björkman. A New End-to-end Probing and Analysis Method for Estimating Bandwidth Bottlenecks. In *IEEE GLOBCOM'00*, San francisco, california, 2000.

[10] K. Lai, M. Baker. Measuring link bandwidth using a deterministic model of packet delay. In *ACM SIGCOM conference on Applications, Technologies, Architecture, and protocols for computer communications*, Stockholm, Sweden, 2000.

[11] C.Dovrolis, D.Moore and P.Ramanathan. What Do Packet Dispersion Techniques Measure? In *Proceedings of the 2001 INFOCOM*,Anchorage AK, April 2001.

[12] M. Jain, C. Dovrolis. End-to-End Available Bandwidth: Measurement Methodology, Dynamics, and Relation with TCP Throughput. In *IEEE/ACM Transactions in Networking*, August 2003.

[13] V. Paxon. end-to-end internet packet dynamics. In *ACM SIGCOMM*,Cannes, France, 1997.

[14] V. Paxon. *Measurements and analysis of end-to-end internet dynamics*. PhD thesis, University of California, Berkeley, April 1997.

[15] J.-C. Bolot. End-to-end packet delay and loss behavior in the Internet. In *ACM SIGCOMM*, San francisco, california, 1993.

[16] A. Pescapè, S. Avallone, G. Ventre. Analysis and. experimentation of Internet Traffic Generator. In *New2an: Next Generation Teletraffic and Wired/Wireless Advanced Networking*, Saint-Petersburg, Russia, 2004.

[17] k claffy, G. Miller, and K. Thompson. the nature of the beast: recent traffic measurements from an internet backbone. In *Proceedings of ISOC INET 98*, Geneva, Switzerland, July 1998.