

Network Performance Monitoring for Applications Using EXPAND.

Björn Landfeldt*, Aruna Seneviratne*, Bob Melander**, Per Gunningberg**

***Dept. of Electrical Engineering and Telecommunications
The University of New South Wales
Sydney 2052 Australia**

bjornl@ee.unsw.edu.au
a.seneviratne@unsw.edu.au

****Information Technology, Dept. of Computer Systems
Uppsala University
P.O. Box 325, S-75105
Uppsala Sweden**

[melander, perg]@docs.uu.se

Keywords: Network monitoring, wireless links, dynamic window generation, passive monitoring and active probing

Abstract:

As computing becomes more interactive and as networked applications need special system support, it has become increasingly important to manage network resources. An important part of the management lies within monitoring network behaviour. It is important for many applications to be able to estimate system performance in order to select appropriate encoding schemes, adaptation, buffering sizes etc. This paper discusses monitoring within wired and wireless networks and the type of monitoring information needed to support different applications. We suggest a hybrid active/passive monitoring approach with a dynamic time window mechanism and interchangeable filters to extract requested information. The paper also shows our initial experimental results and presents our conclusions.

1 Introduction

A range of new applications for use with the Internet has been introduced in the past few years. Many of these are interactive and use data with real time characteristics. This has led to a diversification in the way the Internet is used, and therefore changed the requirements of the network.

In parallel, there has been a rapid growth in cellular and other mobile communication systems such as wireless LANs, and infra red access nodes. Consequently, it is clear that the natural evolution is to combine the two and provide a wireless Internet, thus providing access to information “at anytime from anywhere”. The wireless Internet will comprise of a fixed IP based core network and multiple overlaid wireless access networks. This will provide users with simultaneous access to several different networks [14]. In the ideal case, this will provide the capability of matching the application and user requirements to network characteristics. For example, a user can interactively request a file download over a low latency cellular network, and let the bulk data transfer occur over a high speed satellite link.

In the wired Internet, resources can be reserved to satisfy the needs of applications. There are several existing technologies capable of doing this, such as ATM, Intserv [15] and Diffserv [16]. However, resource availability cannot be guaranteed in the wireless Internet since wireless links are unreliable by nature. The resource availability is not only governed by the maximum link speed and the traffic load, but is also dependant on external factors such as channel fading and noise interference. In this environment, it is important for applications to be able to determine the status of the network in order to adapt to the current conditions. For example, a video player can select encoding scheme according to the available bandwidth. It is therefore important for these applications to have access to a good monitoring tool to determine the characteristics of the network.

A number of network-monitoring tools have been proposed recently [2]. However, they have been primarily aimed at fixed Internet environments. Consequently, they are not suited for the emerging wireless Internet environment. In this paper, we discuss the requirements for performing network monitoring in wireless environments and suggest a framework that provides flexibility in statistical processing and presentation of monitoring information to enable the choice of network and the configuration of applications. The rest of this paper is organised as follows. Section 2 presents related work and section 3 discusses the requirements applications will have on a monitoring tool. Section 4 then discusses general aspects of network monitoring and section 5 examines the relevance of the proposed monitoring framework. Section 6 describes our experimental set-up and results and finally in section 7 we present our conclusions and future work.

2 Related work

A good survey of network monitoring techniques and the available tools can be found in [2]. These monitoring schemes can be divided into two categories, namely active and passive monitoring.

2.1 Active monitoring schemes

Active monitoring schemes operate by inserting probing traffic into the network. These probes are used to extract information about network characteristics such as available and bottleneck bandwidth, round trip delay etc. Bottleneck bandwidth is the bandwidth of the lowest bandwidth link of the entire transmission path when it is idle. Available bandwidth is the unused bandwidth taken over the path and it varies in time depending on cross traffic. The available bandwidth is always less than or equal to the bottleneck bandwidth.

Bottleneck bandwidth can be detected by a technique called back-to-back or packet-pair probing [6]. This technique measures the time space in-between two arriving packets that are sent back-to-back. If there are no other packets in-between the two probing packets, the space corresponds to the time it takes for the second packet to queue until the first packet is transmitted on the slowest link in the transmission path. Therefore, it is proportional to the bottleneck bandwidth. Several monitoring tools use this technique [5,6,7,11]. For example, Pathchar [7] uses this method in combination with traceroute [3], in order to derive per-hop information about the network.

The framework presented in [5] consists of two parts. The first part, b-probe, similarly to the other tools above, only provides information about the bottleneck bandwidth using packet-pair. The second part, c-probe, provides an estimate of the available bandwidth of a network. This is done by sending a “short train” (several packets send back-to-back) of ICMP echo packets and measuring the time in-between the first and last packet, that will eventually be received at the sender side.

A more common way of estimating the available bandwidth is to simulate a bulk data transfer. Ttcp [8] makes an estimation of the available bandwidth based on data from a TCP bulk data transfer. Treno [9] uses ICMP echo packets with flow and congestion control in order to simulate a TCP bulk data transfer, and from that derive the available bandwidth. Compared to probing that measures the distance between packets, both these schemes introduce significant traffic into the network. It should also be noted that these methods measure the average bandwidth over a period, and that the instantaneous bandwidth may vary significantly during this period.

The advantage of using the bottleneck bandwidth is that the value stays fixed as long as the routing path is not changed. The disadvantage is that the bottleneck value, though stable, does not reflect the actual load on a link and thus, may be an over-optimistic metric. Furthermore, bottleneck bandwidth estimation does not detect any degradation in throughput that might occur due to heavy CPU load in a server. We therefore believe that the available bandwidth is more useful for applications. Bottleneck bandwidth can be used for routing purposes, choice of network and for obtaining background statistical information that can be used for estimating bandwidth.

Active monitoring provides an accurate method of determining network characteristics. However, the monitoring traffic competes with application data flows for network resources and active probing is therefore not scalable, especially for resource scarce environments. Furthermore, the delays associated with obtaining necessary results may be large, thus making it unsuitable for use with reactive QoS managers such as USA. The manager reacts to user dissatisfaction and need to present configuration suggestions to the user within an “acceptable” period of time.

2.2 Passive Monitoring

Passive schemes overcome the disadvantages of active schemes associated with overheads and delay by monitoring streams in progress and from them, deriving the current network status, rather than introducing monitoring traffic. SPAND [1] is a monitoring scheme that is based on the principles of passive monitoring. It extends the basic passive monitoring by providing facilities for sharing of measurement results among hosts in order to increase the accuracy. SPAND operates through a centralised server to which applications may report the measured parameters of a flow. This server is accessible by any application to obtain an estimate of the bandwidth to a certain host. The tool relies on the assumption that hosts within a certain region are likely to have the same path to a remote host and hence can share information. Furthermore,

it assumes that network performance is stable so that reported results at the server are valid for a “reasonable” period of time.

Although the last assumption is valid in the current fixed Internet environments as shown in [4], it is not valid in a wireless Internet environment. The performance of a wireless access network not only depends on the current load, but it also depends on other factors such as multi-path fading, other forms of interference, and the transmitted power. These factors in turn depend on the geographic location of the mobile host. Consequently, sharing measurement results collected by mobile hosts are likely to lead to unreliable estimates. Therefore, shared passive monitoring cannot be directly used in the wireless Internet environment.

3 Monitoring for applications

A network-monitoring tool should be versatile, in its ability to present information, thus supporting different applications as described earlier. Figure 1 shows an available bandwidth measurement, which illustrates the need for flexibility when presenting the results to the application.

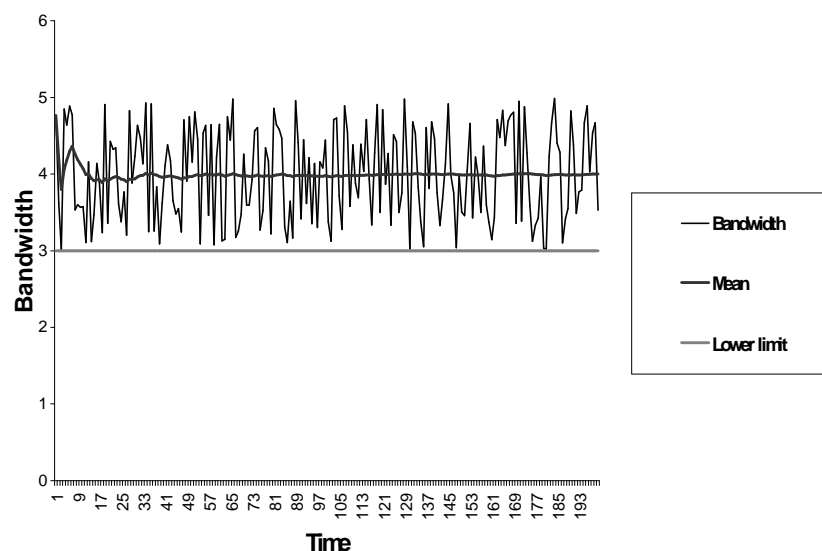


Figure 1. Example of varying available bandwidth

The figure shows bandwidth varying with a “bounded noise” factor, the mean value of the bandwidth and an effective “lower limit” which is the minimum measured bandwidth. Assume that the video conferencing tool in our example wants an estimate of the available bandwidth. The mean value might be a good estimate if the application is adaptive and can compensate for the noise, for example by using a play-out buffer. If not, the “lower limit” estimates the least available bandwidth the application will be expected to obtain from the network and the encoding scheme can be selected depending on this limit.

In order to highlight the need for versatility, table 1 gives a non-exhaustive list of the information different applications may require from a network-monitoring tool. A user running an FTP session may want to estimate the time of large file transfers and the associated monetary cost and battery usage. The mean available bandwidth is sufficient for this. A file transfer is elastic and therefore information about delay and jitter become irrelevant. In contrast, when using a web browser, the response time becomes one of the significant characteristics, therefore the monitoring should indicate the expected packet delay.

<i>Type of application</i>	<i>Bandwidth</i>	<i>Delay</i>	<i>Jitter</i>	<i>MTU</i>
<i>Elastic (FTP)</i>	Yes	No	No	No
<i>Elastic Interactive (HTTP, Telnet)</i>	Yes	Yes	No	No
<i>Real Time (Stored Video)</i>	Yes	No	Yes	Yes
<i>Real Time Interactive (Video Conference)</i>	Yes	Yes	Yes	Yes

Table 1. Example of information required by different application types

As shown in table 1, stored video players need different information, e.g. mean available bandwidth for choosing the appropriate encoding and information about jitter in order to determine the play-out buffer size. Video applications may want to know the path Maximum Transmission Unit, MTU in order to be able to use Application Layer Framing, ALF [10].

4 Hybrid Passive and Active Monitoring Framework

It is clear from the above discussion that neither active, nor passive monitoring can be directly used for monitoring a wireless Internet. However, it is possible to use a hybrid scheme that uses passive monitoring for the wired segments, and active monitoring for the wireless segments coupled with a dynamic windowing mechanism which accounts to the rapid fluctuations. The proposed scheme is such a hybrid scheme, which provides a universal monitoring tool for the emerging wireless Internet environment.

4.1 Hybrid Monitoring

In the proposed scheme, as shown in Figure 2, the wired and wireless parts are treated independently. The passive component uses an extended SPAND server - EXPAND. The EXPAND server thus provides support for active monitoring of the wireless access networks as described below, in addition to the passive monitoring of the fixed network segment.

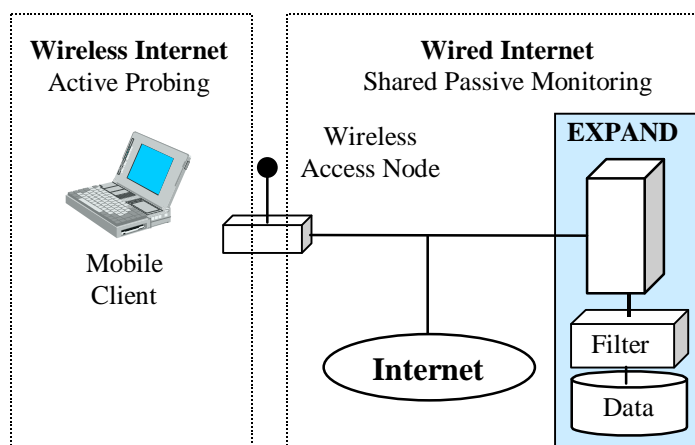


Figure 2. The hybrid passive and active monitoring approach

The hosts attached to the wireless segments will send active probe packets to the EXPAND server. These probe packets will request information from the EXPAND server about the connection via the wired segment to the destination. The EXPAND server will respond with the requested information. The request-response packet pair in turn will be used by the requesting host as active probes to determine the characteristics of the wireless access network. Finally, the requesting host will use the results from the EXPAND server about the fixed segment of the network, and the information about the wireless segment of the network, gathered from the request-response packet pair, to estimate the end-to-end characteristics of the connection.

4.2 Dynamic Windowing

The raw measurement provided by the split-monitoring scheme in itself, is insufficient to obtain a good estimate of the network characteristics. Firstly, in order to obtain a good estimate the raw data needs to be filtered to suit the application requesting the data. Secondly, as the characteristics of the wireless access segment will vary significantly, it will not be possible to use simple statistical techniques. Therefore, the proposed scheme uses a dynamic windowing mechanism to filter the raw data obtained through the split-monitoring scheme. The dynamic windowing scheme exploits the fact that the Internet displays quasi-stable characteristics [4]. This will be true even for the wireless networks as these networks will be used to access information from stationary locations, rather than whilst moving, i.e. from a meeting room, a laboratory or a client's premises. The emerging wireless LAN environment contains mechanisms for reducing the data rate when the signal quality drops, thus giving different link speeds at different locations within the network [12, 13]. The quasi-stable behaviour also applies to cellular networks. Thus, within these networks, the data rate stays stable within a region and the only difference between the wireless and wired segments will be the period of stability.

Considering this, the proposed dynamic windowing scheme attempts to determine a step change in the network characteristics, and then uses a simple statistical analysis of the data from the step in order to determine the characteristics. This is schematically shown in Figure 3.

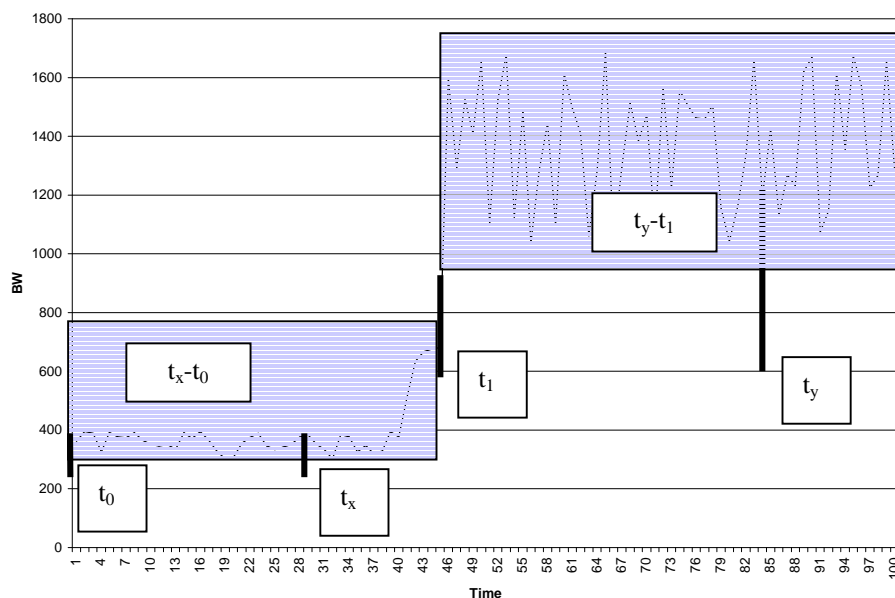


Figure 3. Dynamic window data regions

At time $t = t_x$, $0 < t_x < t_1$, the data set used for calculation is between t_x and t_0 . At time t_y ; $t_1 < t_y$, i.e. when the step change has been detected, the data set used for the calculation is between t_y and t_1 and the data between t_0 and t_1 is discarded.

Considering the above, the windowing scheme consists of two parts. A mechanism for detecting step changes in instantaneous measured data and a statistical filtering mechanism. The step changes are determined by the standard deviation of the data normalised by the mean value. The standard deviation indicates the variability of the measurements and will therefore reflect the step change. The normalisation is done so that the variability can be quantified. In order to find the step, we define a threshold value δ , and compare it against the cumulative standard deviation to mean ratio, i.e.

$$\text{A step change is found if } \delta \leq \frac{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2}{\frac{1}{n} \sum_{i=1}^n x_i} \text{ is true.}$$

The statistical filtering mechanism then filters the data as described below and produces an approximation of the requested parameters, i.e. RTT, bandwidth or jitter.

4.3 Filtering

The filtering phase should be versatile to support the need of different applications. USA will need a good estimate of the available bandwidth and delay. Other applications might need different information. For example, a video player might want to know jitter and mean available bandwidth values in order to determine the size of the play-out buffer. Interactive applications might want to know delay and an estimate of the periods with the lowest bandwidth for selecting an appropriate encoding scheme, since a play-out buffer cannot be used with an interactive application. Therefore, in the proposed scheme the filters are interchangeable, and specified by the requesting application as a part of the signalling while probing the wireless access network.

The filters will approximate the data set within the region found by the dynamic windowing scheme. The region will typically consist of a trend and some noise. Since the traffic over the Internet is not yet fully modelled, it is impossible to determine which approximation is the most suitable. Initially we propose to approximate the data set with a curve, using the least square method in order to take advantage of the extra information the trend provides.

5 Experimental results

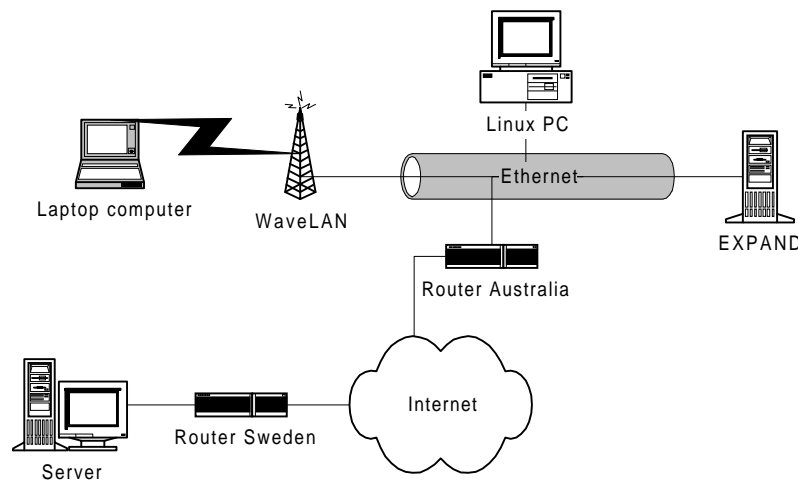


Figure 4. The experimental test-bed

To verify the viability of the above framework we conducted several tests within the experimental test-bed shown in Figure 4.

The WaveLAN network covered one floor of the Electrical Engineering building at the University of New South Wales, and provided varying SNR at different locations. The wireless tests consisted of a person moving with a laptop within the wireless LAN, measuring the available bandwidth to the PC from fixed locations. The wired tests were carried out over the Internet between the PC in Australia and the server in Sweden

5.1 Wireless experiments

Our wireless tests aimed to investigate the traffic behaviour over the wireless link. Figure 5 shows a typical plot of available bandwidth measured over the wireless LAN while moving around. In the figure the client moved towards the cell boundary where the bandwidth dropped significantly. The client then turned and moved towards the centre again and the bandwidth rapidly increased to the maximum value. In our measurements, we could consistently see this behaviour.

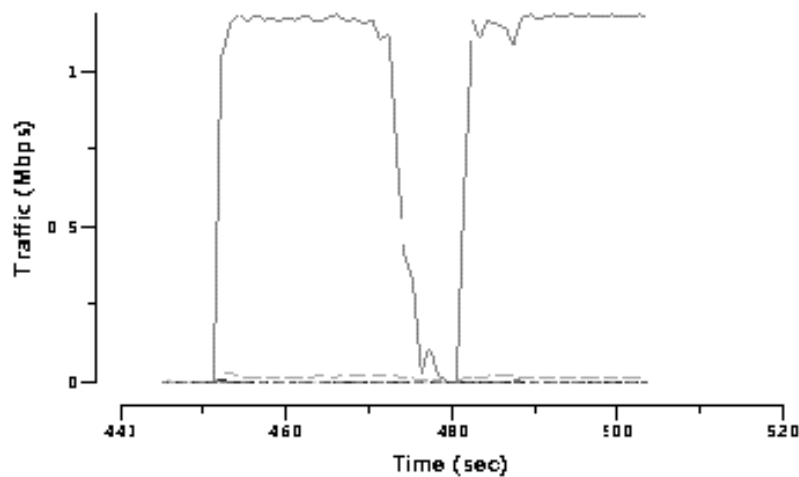


Figure 5. Bandwidth measurements while moving in a wireless LAN.

Because of the stable behaviour within a cell and rapid changes at the boundaries, the need for a windowing scheme is accentuated. The stable behaviour also means historical data is useful even within a wireless network.

5.2 Dynamic window experiments

We conducted a series of measurements between Australia and Sweden in order to investigate the viability of the dynamic window scheme. Figure 6 shows one measurement of the round trip time (RTT) measured between Australia and Sweden during 4 hours. The following example demonstrates the performance improvement of the dynamic window scheme.

As can be seen the measurements vary significantly with time. There is a significant change at around 8 am when most people come in to work and start to read mail, newspapers etc. The round trip time reaches a peak at around 8.30 am and starts to decay slowly until 9 am after which it drops down to its previous level.

The figure shows RTT estimations using three different methods, mean value, least square linear approximation and least square linear approximation using the dynamic window scheme. In order to find the edges of the dynamic windows we used the mean to standard deviation formula described above and compared it to a δ -value of 0.15. Determining the appropriate δ -value is still

an open research issue. For now, our experiences show that a value between 0.15 and 0.20 makes the scheme perform well.

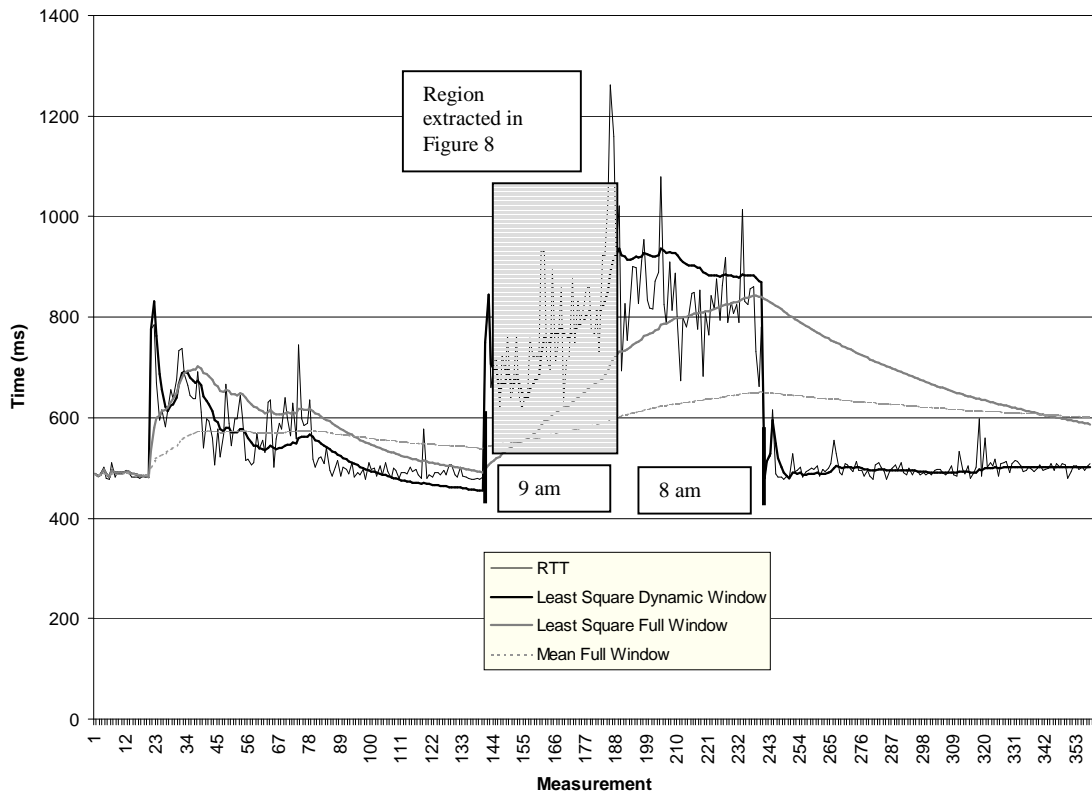


Figure 6. RTT estimation between Australia and Sweden using three different methods

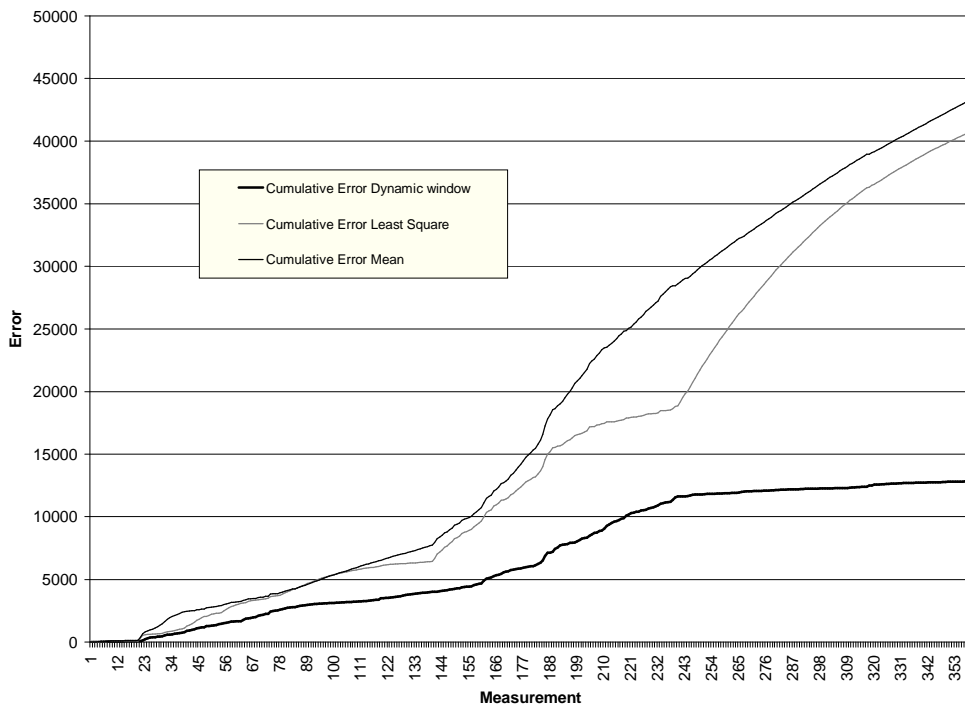


Figure 7 Cumulative Error of estimated and actual RTT values

The δ -value of 0.15 results in the dynamic windowing scheme finding four different regions with values 1-21, 22-141, 142-241 and 242-359. The figure clearly shows that the least square linear approximation with dynamic windowing estimates the RTT better than the other two methods. In order to highlight the performance difference of the three methods, Figure 7 shows the cumulative error between estimated and actual RTT values. Again, the least square linear approximation with dynamic windowing performs much better than the other two methods.

5.3 Filtering experiments

We have made preliminary experiments with different methods for filtering the data after we have applied the dynamic window scheme. From our measurements, we have found that the traffic over the fixed Internet show quasi-stable properties as stated previously. We have also found that within the stable periods there are often slowly growing or decaying trends. These trends provide additional information to a weighed value, when estimating parameters such as RTT and bandwidth. We therefore ran experiments using different methods to see how well they captured these trends. Our measurements show that a median or mean calculation wrongly estimates the value when there is a trend in the data set.

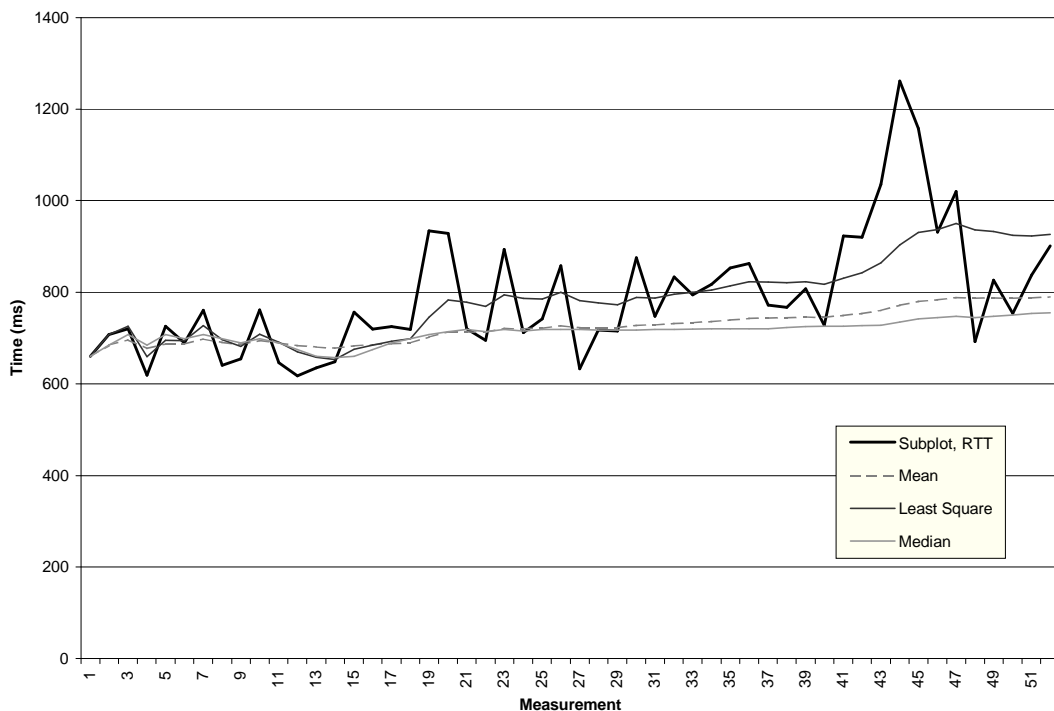


Figure 8. RTT subplot

As an example, Figure 8 shows the region 144-197 extracted from the data in Figure 6. The figure also shows the mean, median and least squares linear estimation of the RTT value. It can be seen in the figure how the median and mean calculations underestimate the trend whereas the linear approximation better captures it. Until the Internet traffic has been mathematically modelled properly, it is impossible to determine the best method of estimation. We can however conclude that mean and median values are not suitable for estimating values such as RTT and bandwidth over a period since they do not accurately capture the trends.

6 Conclusion and future work

In this paper, we discussed how network monitoring should be carried out in a mixture of wired and wireless environments. The purpose of this discussion is to form guidelines for an overall framework for network monitoring taking into account the different needs of different applications. We have proposed a split network monitoring approach with passive/active monitoring. Central to this framework is the method of dynamic windowing to find a region with a relevant set of data on which to conduct statistical computations. We have also discussed the need for flexible filtering of the monitoring information in order to derive the different information needed by different applications. Finally, we have conducted initial experiments to verify our framework and to show the advantages of the dynamic window scheme.

Our experimental results show that using the dynamic windowing scheme gave consistently better results than using instantaneous measurements or fixed windows. Instantaneous measurements are not reliable when there is a large fluctuation in the measurements, and it is impossible to determine a correct size for the fixed window. We also believe that the dynamic windowing scheme is sufficient to determine the state of the network and to extract appropriate regions. We therefore conclude that the scheme is a universal tool that applies to all network types.

In the future, we intend to focus on statistical computations of monitoring results. Firstly, we will investigate possible ways of determining the optimal time window size for filtering transient events while still reacting on sustained bandwidth shifts. The combination of statistical properties and application requirements will be of primary interest. Secondly, we will work on statistical methods for use in different filters in order to provide accurate and for applications appropriate estimates of the performance parameters. Thirdly, we will work on implementation issues such as a protocol between a terminal and a SPAND server for specifying filter options etc. Finally, we intend to implement a prototype of the framework in order to evaluate its usefulness.

Acknowledgements

The authors would like to acknowledge Ericsson Radio Systems AB, Sweden and Ericsson Australia for their financial support.

References

- [1] S. Seshan, M. Stemm and R. Katz. "SPAND: Shared Passive Network Performance Discovery". In Proc 1st Usenix Symposium on Internet Technologies and Systems (USITS '97) Monterey, CA December 1997.
- [2] <http://heplibw3.slac.stanford.edu/xorg/nmtf/nmtf-tools.html>.
- [3] W. R. Stevens, TCP/IP Illustrated Volume 1 Addison-Wesley, Reading MA, Nov 1994.
- [4] V. Paxson. "Measurements and Analysis of End-to-End Internet Dynamics. PhD thesis, U. C. Berkeley, May 1996.
- [5] R. L. Carter and M. E. Crovella. "Measuring bottleneck-link speed in packet switched networks". Technical Report BU-CS-96-007, Computer Science Department, Boston University, March 1996.
- [6] S. Keshav. "Packet-Pair Flow Control". IEEE/ACM Transactions on Networking, Feb 1995.

- [7] Pathchar – A Tool to Infer Characteristics in Internet Paths. <ftp://ee.lbl.gov/pathchar>, 1997.
- [8] IEEE RFC 1470 June 1993
- [9] M. Mathis and J. Mahdavi “Diagnosing Internet Congestion with a Transport Layer Performance Tool”. In Proc. INET '96, Montreal Canada, June 1996.
- [10] D. Clark and D. Tannenhouse “Architectural considerations for a new generation of protocols”. In SIGCOM'90 Philadelphia Pennsylvania September 1990.
- [11] K. Lai and M. Baker, "Measuring Bandwidth." Proceedings of IEEE Infocom' 99, March 1999.
- [12] www.3com.com/promotions/wireless
- [13] www.wavelan.com/products
- [14] R. Katz and E. Brewer, “The Case for Wireless Overlaid Networks”, SPIE Multimedia and Networking Conference (MMNC'96) San Jose January 1996.
- [15] R. Braden et al. “Integrated Services in the Internet Architecture: an Overview”, RFC 1633, ISI, MIT and PARC, June 1994.
- [16] K. Nichols, V. Jacobson and L. Zhang, “A two-bit Differential Services Architecture for the Internet”, IETF Internet draft November 1997.