# Active Probing Based End To End Internet Path Metrics Estimation Tool

#V.Suryanarayana, K.Chandra Sekaran
*Department of CSE, MIT, Manipal,vytla_suryax@yahoo.co.in*
*Department of CSE, NITK, Surathkal, kch@nitk.ac.in*

*Abstract*— ***This paper considers the problem of measuring the various network path metrics such as Connectivity, Capacity, bandwidth, available bandwidth etc. of a network path using probing-based methods. To that end, we propose a new model called an Integrated Metrics Measurement Model (IMMM) to measure all possible performance metrics. It will be designed to probe the network non-intrusively so that the measurements will not disturb the stability of the network. The experimental results of our partial implementation of the proposed model are presented.***

*Keywords*— Active Probing, Available Bandwidth,, Capacity, Internet Measurements, Network Metrics, Quality of Service.

## 1. INTRODUCTION

*a) Networks and Internets*

A network is a collection of hosts connected together so that they can exchange information. Hosts in a network communicate using a mutually-agreed network protocols. An internet is a collection of networks with links between them so as to allow hosts on any one of the networks to communicate with hosts on any network within that internet [1] [2]. Routers are devices with links to more than one network; they forward packets back and forth between the networks.

The existing packet switched network model do not provide end-to-end performance and quality of service estimates which are as reliable as those of traditional models for circuit switched networks, leaving end-to-end performance measurements as the only really reliable source of this crucial information. Consequently end-to-end measurements became the dominant measurement technique used in the Internet [3]. Internet applications themselves rely upon and are dependant on the feedback of network performance information. As a simple example TCP calculates its retransmission time by estimating end-to-end Round Trip Time (RTT) between the involved end-hosts.

Network researchers have adopted two distinct approaches to data collection [6]. The first approach is that of passively observing and recording network traffic. The second approach uses an active measurement system to inject probe traffic into the network and then extrapolate the performance of the network from the performance of the injected traffic.

One drawback of passive measurements is that they generate a large amount of measurement data. The key advantage of active measurement is its great flexibility: packet probes can be chosen at will (within limits), and many different kinds of probe traffic selected, including packet streams which mimic different traffic types. So, active measurements are a convenient means to estimate various network metrics for ordinary users because it does not require router access. Although passive network monitoring methods such as simple network management protocol (SNMP) can provide detailed statistics on network elements (routers and switches) such as physical bandwidth (capacity) and utilization, they unfortunately require special access privileges which are not usually available to ordinary users [1].

As the result of the increased importance of end-to-end measures, the lack of standardization in the area of measurements and the fast pace of development in the Internet, the traditional focus of measurement efforts has shifted from the core to the edge of the network , and the measurements are typically performed by designated measurement hosts located at the network edges rather then by network elements inside the core .

*b) Some Important Network Metrics*

Latency: [5] Latency is a time delay while one waits for something to happen. A widely used measure of network latency is round-trip time (RTT), the time for a packet to make the round trip from a client to a server and back. Many component times contribute to network latency such as transport time, queuing time, and server response time.

Packet Loss: [4] [5] by default, Internet packet transport works on a best-effort basis. Routers make every reasonable effort to forward packets, but may drop them depending on the router's immediate local conditions. Network packet loss is the fraction of packets lost in transit from client to server and back during a specified time interval, expressed as a percentage of the packets sent to the server during that interval.

Throughput: [1] it is the rate at which data is sent through the network, usually expressed in bits per second (bps), bytes per second (Bps) or packets per second (pps).

Bandwidth-Related Metrics: [1] some important bandwidth related metrics are capacity and available bandwidth.

Capacity: [1] [7] capacity of a hop is defined as the maximum possible IP layer transfer rate at that hop. So we define the capacity of a hop as the bit rate, measured at the IP layer, at which the hop can transfer MTU-sized IP packets. Extending this definition to a network path, the capacity C of an end-to-end path is the maximum IP layer rate the path can transfer from source to sink. The minimum link capacity in

the path determines the end-to-end capacity C. The hop with the minimum capacity is the narrow link on the path.

Available Bandwidth:[1] [2] The available bandwidth of a link relates to the unused or spare capacity of the link during a certain time period. The available bandwidth of a link additionally depends on the traffic load at that link, and is typically a time-varying metric.

*c) History of Internet Measurements*

Sending probe packets to networks is a common way to actively measure network metrics. For each algorithm, there are different methods used to probe the network: single packet, multiple packets (packet trains) etc. There are several techniques for using these methods, such as varying the packet size, dispersion, spacing and so on [2].

Ping is the earliest (1980) and simplest network measurement tool based on a single packet probe [8]. It measures the round trip time (RTT) based on the time to forward a single packet and the time to get an Internet Control Message Protocol (ICMP) reply packet. Ping results can be used to estimate network congestion by analyzing the RTT variation. Developed in 1988, traceroute used a similar mechanism to measure RTT and router addresses on each hop. In the same year, ttcp provided a method using a User Datagram Protocol (UDP) stream to obtain a majority of the path bandwidth in order to estimate the path bandwidth in a highly intrusive manner. In 1991, netest-1 (netest version 1) used a burst methodology, assuming that a UDP burst can gain most of the bandwidth in 10 RTTs if most of the competing network traffic is Transmission Control Protocol (TCP). The maximum burst size was set between 0.5- 1 second. Netest-1 repeats the same test in every 5 seconds with a short UDP/TCP burst instead of a continuous UDP stream.

The methods used by both ttcp (UDP mode) and netest-1[8] are based on packet trains. Pathchar [8] was released in 1997, used a variable packet size (VPS) algorithm to measure link capacity (physical bandwidth). Since 2001, many bandwidth estimation tools have been released, most of them designed to measure available bandwidth, and most based on packet dispersion. Nettimer [1] [2] is for estimating the narrow link (a router or switch that has the lowest capacity along a path) physical bandwidth. Pathload [2] estimates the available bandwidth. NCS and netest rev. 2 are designed to measure bandwidth as well as the achievable throughput and other important network characteristics. Two new tools, pathchirp and igi [8], are also for available bandwidth measurement. All of these tools are active measurement tools because they send packets into the network in order to make a measurement. Some tools, such as ping, are not intrusive. Packet-train-based tools, such as ttcp, can be very intrusive, sending a large number of packets into the network and possibly pushing other traffic aside. Packet trains that are too long can also cause router queues to overflow.

*d) Statistical Filtering*

Once after understanding what network metrics exist and how to measure them, we must decide how to present the results using statistical techniques such as Mean, Median, and Percentiles

*e) Applications of Internet Measurements*

The main applications of Internet measurements are topology, performance, management and routing. Network traffic measurements are essential if network operators are to monitor and maintain the quality of services, and are also fundamental for strategically important activities like network planning. However, measurements are not only at the heart of operating existing networks but are the key to the continuous improvement of the performance and quality of networking technologies as a whole.

The Cooperative Association of Internet Data Analysis (CAIDA) [8] is currently using skitter to monitor the Internet's topology, using probes sent from multiple sources to a large number, tens of thousands, of destinations located all around the globe. Performance focuses on the analysis of end-to-end behavior and on the diagnosis of network problems. These efforts typically include the collection of end-to-end packet loss, delay and round trip time statistics most often performed by injecting test traffic into the network. Routing measurements provide insight into the dynamics of routing protocols and routing table updates. These are of great importance as the reliability and robustness of the Internet depends on the stability and efficiency of routing.

## 2. INTEGRATED METRICS MEASUREMENT MODEL (IMMM)

Currently existing tools have got some drawbacks including portability issues, poor GUI, and poor presentation of observed data. To obtain the performance of a given path, it is required to run many tools and lot of packets has to be injected. It is also required to modify the existing algorithms to get the best estimates of the metrics by considering various overheads in the measurements.

In view of the above facts, we have proposed a new model for measuring various network metrics mentioned above. The model is known as Integrated Metrics Measurement Model (IMMM) and shown in figure1.

This model contains different modules named Packet Probing module, Overheads Estimating module, Statistical Filtering module, and Metrics Estimation module. Along with these modules there is an Input Data file and an output database file. The functions of various modules are as follows:

*a) Input Data File*: The input data file is used to store the information such as destination host addresses. Measurements can be made to several end hosts simultaneously using threads by specifying their IP addresses or domain names in this file.

*b) Packet Probing module*: This module is used to generate probing packets as per the user choice. User is allowed to set various parameters such as type of probing protocol (ICMP, TCP or UDP), number of individual packets, number of

packet pairs, number of packet trains, maximum size of the packet, type of packets etc. This module is also responsible for receiving probing packets and for each packet received; it stores the observed data such as Round Trip Time, dispersion, arrival time, packet size, packet losses etc. in a database. This database is referred as raw database as some overheads are included in those observations.

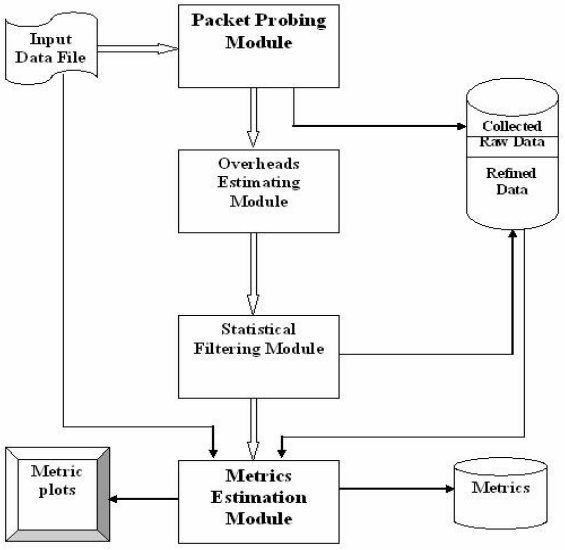



Figure1: Integrated Metrics Measurement Model (IMMM)

*c) Overheads Estimation module:* This module takes care of overheads involved in the measurement process. Overheads include average time taken to perform various system calls, router queuing delays, and server response time. The time to perform a system call affects both the outgoing packet spacing and the time to get timestamps for incoming packets. The syscall time is measured for two areas — getting a timestamp and doing I/O. The system call time also affects our ability to increase the system timer resolution.

Some overheads like system call time is almost constant for a given measurement host and is referred as static overheads. Other overheads like router queuing delays and server response time are dynamic in nature and keep varying and are referred as dynamic overheads. This module finds the static overheads before initiating the measurements and estimates the dynamic overheads during the measurement phase.

*d) Statistical Filtering module:* This module is responsible for obtaining a refined data from raw data by performing suitable statistical filtering. The refined data is stored in the database so that it can be used in estimating various metrics.

*e) Metrics Estimation module:* This module is the heart of the IMMM, and is responsible for estimating the end-to-end metrics in question from measurement host to one or more destination hosts.

## 3. IMPLEMENTATION OF IMMM

In our ongoing efforts, we have started implementing the IMMM using VB.NET on Microsoft .NET Framework. The .NET Framework is a development and execution environment that allows different programming languages & libraries to work together seamlessly to create Windows-based applications that are portable, easier to build, manage, deploy, and integrate with other networked systems. The Graphical User Interface (GUI) we have designed for measurements is very user friendly, which is one of the drawbacks of existing UNIX based monitoring tools.

## 4. EXPERIMENTAL RESULTS

The input data file is created with four destination domain names as shown in figure 2. Measurements are conducted from our residence with BSNL dialup connection. Measurement host configuration is HP, Intel Pentium, and 1.73 GHz.

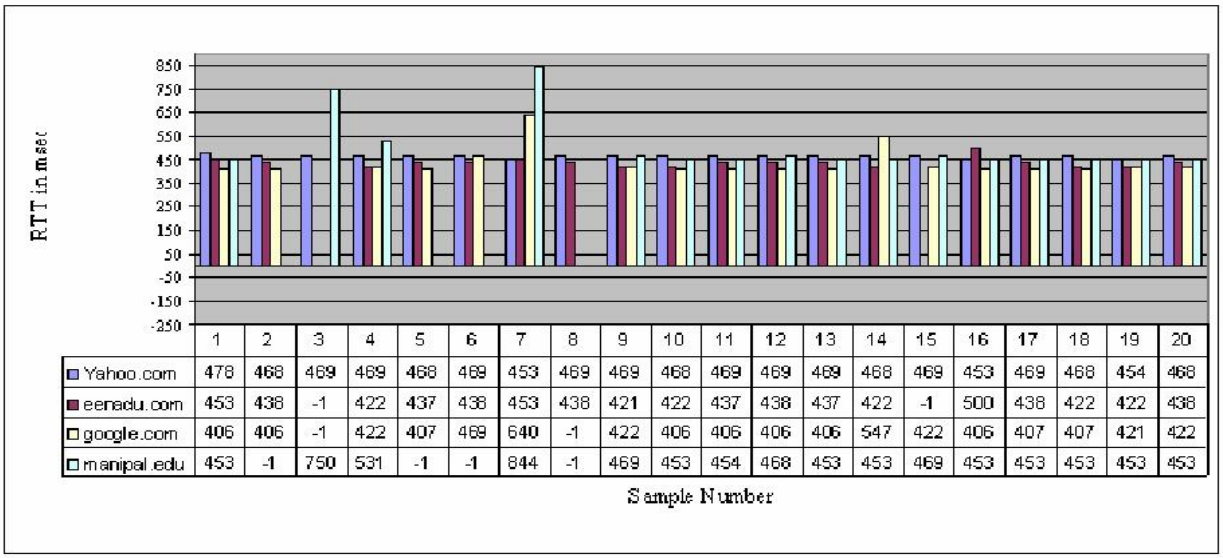
```
www.yahoo.com www.eenadu.net
www.google.com www.manipal.edu
```

Figure 2: Input Data file Contents

The raw 20 samples of RTT data for the given four destinations using a dial up connection is given in figure 3. RTT of -1 indicates a packet loss. Various statistics of RTT and packet loss are shown in figure 4.
The intermediate router IP addresses and the corresponding RTTs in milliseconds for another set of destination hosts (www.msn.com, www.harward.edu, and www.google.com ) obtained using our tool is given in figure 5.

## 5. CONCLUSION

In this paper we have presented a summary of the internet measurements and current techniques of metrics estimation. The details about the proposed model and its implementation have been presented and discussed. The experimental results of our partial implementation of the proposed model are presented.




| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Yahoo.com | 478 | 468 | 469 | 469 | 468 | 469 | 453 | 469 | 469 | 468 | 469 | 469 | 469 | 468 | 469 | 453 | 469 | 468 | 454 | 468 |
| eenadu.com | 453 | 438 | -1 | 422 | 437 | 438 | 453 | 438 | 421 | 422 | 437 | 438 | 437 | 422 | -1 | 500 | 438 | 422 | 422 | 438 |
| google.com | 406 | 406 | -1 | 422 | 407 | 469 | 640 | -1 | 422 | 406 | 406 | 406 | 406 | 547 | 422 | 406 | 407 | 407 | 421 | 422 |
| manipal.edu | 453 | -1 | 750 | 531 | -1 | -1 | 844 | -1 | 469 | 453 | 454 | 468 | 453 | 453 | 469 | 453 | 453 | 453 | 453 | 453 |

Figure 3: Raw RTT Data

| Destination address | Max RTT | Min RTT | Avg. RTT | No. of packets sent | No. of packets received | % of packet loss |
|---|---|---|---|---|---|---|
| yahoo.com | 478 | 453 | 466.8 | 20 | 20 | 0 |
| eenadu.net | 500 | 421 | 437.9 | 20 | 18 | 10 |
| google.com | 640 | 406 | 434.9 | 20 | 18 | 10 |
| manipal.edu | 844 | 453 | 503.9 | 20 | 4 | 20 |

Figure 4: Statistics of RTT and Packet Loss

| msn.com | | harward.edu | | google.com | |
|---|---|---|---|---|---|
| router_addr | RTT | router_addr | RTT | router_addr | RTT |
| 218.248.47.1 | 0 | 218.248.47.1 | 0 | 218.248.47.1 | 0 |
| 172.24.64.130 | 15 | 172.24.64.130 | 15 | 172.24.64.130 | 16 |
| 218.248.255.6 | 16 | 218.248.255.30 | 16 | 218.248.255.6 | 16 |
| 202.54.185.170 | 546 | 202.54.185.170 | 547 | 203.101.65.181 | 16 |
| 202.54.2.221 | 563 | 59.163.16.146 | 750 | 203.196.8.1 | 578 |
| 202.54.2.14 | 781 | | | 217.6.24.169 | 578 |
| 208.192.178.117 | 578 | | | 217.239.39.230 | 578 |
| 152.63.35.38 | 562 | | | 62.156.139.218 | 594 |
| 152.63.107.254 | 579 | | | 195.219.15.214 | 578 |
| 152.63.107.241 | 578 | | | 195.219.196.9 | 594 |
| 157.130.177.254 | 609 | | | 195.219.196.22 | 578 |
| | | | | 216.6.63.22 | 594 |

Figure5: Intermediate Router IP addresses

## REFERENCES

[1] Ravi S.Prasad , Constantinos Dovrolis, Bandwidth estimation: Metrics, Measurement Techniques, and Tools", IEEE Network, November/December 2003, pp 27-35

[2]Manish Jain and Constantinos Dovrolis, " End_to_End Available Bandwidth: Measurement Methodology, Dynamics, and Relation with TCP Throughput, IEEE/ACM Trans. Networking , vol. 11, August 2003.

[3] RFC 2679, " A One-way Delay Metric for IPPM", G. Almes, S. Kalidindi, M. Zekauskas, Advanced Network & Services,September 1999.

[4] RFC 2680, "A One-way Packet Loss Metric for IPPM", Guy Almes, Sunil Kalidindi, Advanced Network & Services, September 1999

[5] RFC 2681, "A Round-trip Delay Metric for IPPM", G. Almes, S. Kalidindi, M. Zekauskas, Advanced Network & Services, September 1999.

[6] Chuck Fraleigh, NetVMG, Sue Moon, KAIST: "Packet_Level Traffic Measurements from the Sprint IP Backbone ", IEEE Network, Nov/Dec 2003.

[7] C. Dovrolis, P.Ramanathan and D.Moore, "what do packet dispersion techniques measure", proc. IEEE INFOCOM , April 2001.

[8] CAIDA, http://www.caida.org/tools/taxonomy, Oct 2002