Active and Passive Network Measurements : A Survey

Venkat Mohan. , Y. R. Janardhan Reddy, K. Kalpana

Computer Science Department, G.Pulla Reddy Engineering College: Autonomous,Kurnool, India

Abstract— **One major question facing operators everywhere is how to be sure that everything goes fine as well as how black holes can be detected in their networks? Passive network monitoring is very suitable for this purpose. It can be used for searching problems of a single network device, a major problem affecting the whole LAN or core network. Passive network monitoring, however, is not just for problem solving, it can also be used for creating network statistics or for measuring network performance. As will be seen in this survey, it is a very powerful tool in everyday network life. Delay or packet loss can be measured with either passive or active means. In this survey, the focus is on both passive and active measurements. The goal of this survey is to introduce the reader to passive and active measurements in data networks.**

Keywords— **Network Measurements, Monitoring Methods, Active Measurements, Passive Measurements, Techniques.**

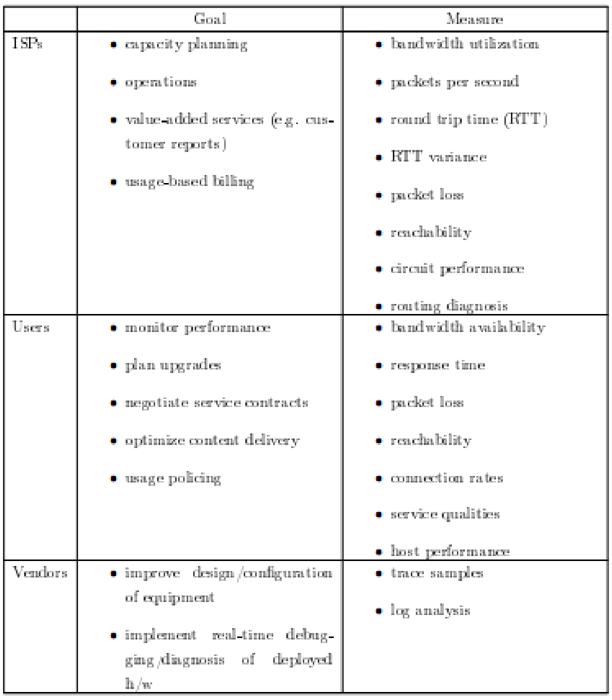
I. Introduction

For network operators it is important to know how well their network performs so that they know what kinds of services they are able to offer to their customers. In addition to measuring performance, network operators use active/passive measurements to troubleshoot their network. In some cases there might be a fault in the network that causes traffic to be routed the wrong way. Generating an artificial traffic flow through the network and inspecting its behavior can help to troubleshoot routing faults.

When introducing a new application or service to a network it is necessary to test the performance of the application before making it available for the users. Active measuring can be used to simulate a large number of users thus it can help in finding out for example how many simultaneous users a web server can service. Passive monitoring in conjunction with active probing (this is called hybrid measurement) can be used in finding out how a new service impacts the network both from the end-user's and the network operator's point of view.

The purpose of network monitoring, as discussed, is to observe and quantify what is happening in the network. With different sizes of magnifying glasses (methods, techniques, and tools) we can observe both the microcosmic and macrocosmic events in time or in state. By gathering data actively or passively from the network, we have a great opportunity towards the following actions [55, 56]:

- Performance tuning: identifying and reducing bottlenecks, balancing resource use, etc.
- Troubleshooting: identifying diagnosing and repairing faults.
- Planning: predicting the scale and required resources.
- Development and design of new technologies: Understanding of current situation in a network, finding trends and directing the development of new technologies.
- Characterization of traffic for providing data for modeling and simulation.
- Understanding and controlling complexity: understanding and interaction between components of the network and to confirm that functioning, innovation and new technologies perform as predicted and required.
- Identification and correction of pathological behavior.



	Goal	Measure
ISPs	• capacity planning • operations • value-added services (e.g. customer reports) • usage-based billing	• bandwidth utilization • packets per second • round trip time (RTT) • RTT variance • packet loss • reachability • circuit performance • routing diagnosis
Users	• monitor performance • plan upgrades • negotiate service contracts • optimize content delivery • usage policing	• bandwidth availability • response time • packet loss • reachability • connection rates • service qualities • host performance
Vendors	• improve design/configuration of equipment • implement real-time debugging/diagnosis of deployed h/w	• trace samples • log analysis

The above table presents three parties: Internet Service Providers (ISPs), users and vendors and gives a brief idea on their respective goals and why these are intended to measure.

ISPs are interested in transferring maximum amount of data at minimum amount of data at minimum costs. In addition, the billing should work properly if the commercial ISP is in question. On the contrary, a user can have a totally different view: he/she usually wants small delay and very low packet loss in end-to-end connections. A user also wants to have persistent connections with full bandwidth as in an agreement between an ISP and a user.

Vendor can be said to be between a user and ISP. For ISPs they try to produce more efficient and cost effective solutions to forward traffic in the network. For users they develop monitoring and measuring software and hardware in order to help a user monitor his/her connections and services.

The rest of document is organized as follows. Section II outlines the basic terms and notions. Section III outlines the passive, active and hybrid measurement techniques along with the differences between passive and active measurements. The active measurements are listed in Section IV and the passive network monitoring methods are framed in the Section V.

II. Basic Terms and Notions

In this section some basic terms and notions related to computer networks are listed.

- *Path*: A sequence of links from a source node S to destination node D is called a (network) path. Also the nodes connecting the links can be considered to be a part of the path.
- *Link Capacity*: The capacity of a link is the maximum transfer rate possible for that link [1]. It must be noted that link capacity is defined per protocol layer. This means that the link capacity on Layer 2 is different from the link capacity on Layer 3 although the physical link is the same. The capacity *C* of an end-to-end path is the minimum link capacity C_i in the path:

$$C = \min C_i ,$$

- *Delay (latency)*: In telecommunications there are several types of delay such as processing delay, propagation delay, queuing delay and transmission delay. In this thesis the notion of delay includes all the mentioned delay types and can be thus called end-to-end delay.

$$D_{E2E} = D_{PROCESSING} + D_{TRANSMISSION} + D_{PROPAGATION} + D_{QUEUING}$$

Processing delay is the sum of delays caused by all the intermediate nodes on the network path processing the packet. A router needs to examine the arriving packet's header to determine where to direct the packet. It also does bit-level error checking to see if the packet is corrupted and it may also process the packet by doing e.g. firewalling, encryption, etc. All these functions the router performs add to the delay caused by the processing. Processing delay mainly occurs on the edge routers of the network.

Transmission delay (or serialization delay) is the time it takes to send out a packet at the bit rate of the link. In other words transmission delay is the amount of time required by a router to push the entire packet onto the link.

$$D_{TRANSMISSION} = \frac{L}{R},$$

Where *L* is the length of the packet and *R* is the transmission rate of the link.

Propagation delay is the time required for the signal to travel from one end of the transmission medium to the other. The delay depends on the physical medium and thus the delay is the distance between two end-points divided by the propagation speed.

$$D_{PROPAGATION} = \frac{d}{\eta c},$$

Where *d* is the distance, *c* is the speed of light and $\eta \leq 1$.

Queuing delay is the amount of time a packet spends inside routers' queues on its way from the source node to the destination node. Queuing delay is proportional to the buffer size and the amount of cross-traffic entering the router.

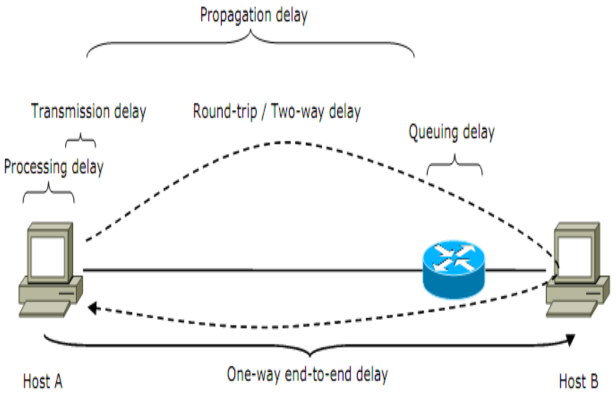


Figure 1. *Delay types.*

Delay measurements produce either one-way or two-way results. One-way delay is the end-to-end delay of a packet from the sending host (Host A in Figure 1 above) to the receiving host (Host B). Two-way delay (or round-trip time, RTT) is the delay of a packet from sender to receiver and back.

- Packet delay variation and inter-arrival time variation (jitter): The variation of packets' one-way delays is called variation (or jitter). The use of the term jitter is nowadays deprecated as it has been used in different meanings by different groups [2].

Instantaneous packet delay variation (PDV) can be calculated from two successive packets' one-way delays:

$$PDV_{INSTANTANEOUS} = D_{n+1} - D_n ,$$

Where Dn+1 and Dn are one-way delays of two consecutive packets.

Delay variation can be caused by congestion in network, routing changes or timing drift. It affects especially real time applications such as VoIP or video streaming where it causes jerkiness in video and breaks in audio. Buffering is used to battle the effects of delay variation: in the receiving end of VoIP-call, packets are buffered and played back after a short delay. This helps the receiver to order and space arriving packets so that the voice stream is continuous and as close to the original as possible.

The variation in the time between packets arriving to a host is called packet inter-arrival time variation (also referred to a jitter). Instantaneous packet inter-arrival (IAT) time can be calculated from two successive packets' arrival times:

$$IAT_{INSTANTANEOUS} = A_{n+1} - A_n ,$$

Where An+1 and An are the arrival times of two consecutive packets.

Queuing: In packet networks queues are used to mitigate the effects of bursty traffic. A router can process only one packet at a time. If packets arrive on a router faster than the router can process them, the packets are put into a queue. The packets wait in the queue until the router has enough time to process them. If the queue is full and still more packets arrive to, the router the packets arriving are dropped (this is main cause of packet loss).

Packet loss, loss period and loss distance: When a packet is sent from host A to host B and the packet never arrives to B, the packet is lost. This is called packet loss. It is not reasonable to keep on waiting for a packet forever so usually there is some kind of timeout mechanism that discards the packet if it takes too long to reach the other end of the network. This way a packet can be declared lost even if it would reach B at some point.

Packet loss can occur because of several reasons: a packet can be discarded in router because of buffer overflow or because the arriving packet is corrupted, the packet can be accidentally misrouted or be lost because of a link failure or wireless channel errors. Faulty or misconfigured equipment can also cause packet loss. Some congestion control or avoidance mechanisms (such as RED) can cause packet loss intentionally to trigger TCP window size reductions.

Loss period and loss distance are two important notions that are closely tied to packet loss. Loss period is the length of a packet loss event in successive lost packets. The period starts when a packet is lost and a preceding packet is received and ends when a packet is received and the preceding packet is lost. Loss distance is the difference in sequence numbers of two consecutively lost packets that may have received packet between them (see figure 2) [3].

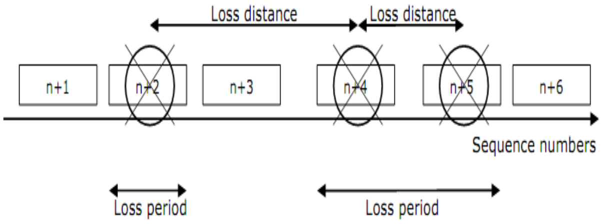


Figure 2. *Packet loss distance and period.*

Packet loss distribution can have a varying impact on video and voice applications. How lost packets are distributed can change the way packet loss degrades, for example, a voice stream. If there are long loss periods during a VoIP call, the voice codec cannot use previously received data packets to "fill in the blanks" and thus the quality of the voice stream is seriously degraded. On the other hand if the lost packets are distributed more widely (shorter loss periods more often), the codec can use history data to replace the missing packets and the degradation is not necessarily as severe.

- *Throughput*: Throughput is a measure of how much data is transferred across a link or a network in a certain time. Usually throughput is measured in bits per second or bytes per second.
- *Available Bandwidth*: The available bandwidth of a link is the unused capacity of a link at a certain time period. If C_i is the capacity of a link and u_i is the average utilization of the link (thus the link transmits C_iu_i bits) during time period T, then the available bandwidth for the link is A_i:

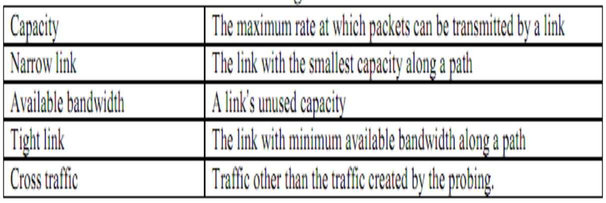
$$A_i = (1 - u_i)C_i \, .$$

From this we get the available bandwidth of a path of N hops:

$$A = \min_{i=1,\dots,N} A_i \, .$$

Table 1 lists terms and notions related to available bandwidth measurement. These notions are later used below when presenting mechanism for active bandwidth measurement.

Table 1. *Terms and notions relating to available bandwidth measurement.*



Capacity	The maximum rate at which packets can be transmitted by a link
Narrow link	The link with the smallest capacity along a path
Available bandwidth	A link's unused capacity
Tight link	The link with minimum available bandwidth along a path
Cross traffic	Traffic other than the traffic created by the probing.

- *Bulk Transfer Capacity*: [4] defines the Bulk Transfer Capacity (BTC) metric as follows:

$$BTC = \frac{sent\_databits}{elapsed\_time} ,$$

Where *sent_databits* represents the number of unique data bits sent (unique in the sense that header bits and retransmissions are not included). BTC is a measure of TCP (or some other congestion aware transport protocol) connection's maximum obtainable throughput. It must be noted that since BTC is TCP-specific and it cannot be as such compared with the available bandwidth metric.

- *Goodput*: In this survey goodput means the effective throughput experienced by a user and in this sense goodput can also be called as application level throughput. Goodput is a measure of how many user data bits per time unit (usually seconds) can be forwarded by a network or system. Goodput can be calculated by subtracting all header overhead and retransmissions from throughput.

  A good example of goodput is a file transfer where a user downloads a file from a remote server. In this case goodput is the file size divided by time it takes for the file to download completely. If the measured throughput during the file transfer is 100 kbps, the goodput can, for example, be only 90 kbps because of header overhead and retransmissions.
- *Probes*: Special probe packets are used in active measurements: a probe is inserted into the network and the response is recorded and analyzed. A probe packet is an artificial packet that can be almost of any type depending on the information wanted from the measurement. A simple example of a probe packet could be a small UDP packet that contains only a timestamp and little or no payload at all. This type of probe could be used in delay measurements or to measure VoIP performance.

Probe packets and their properties should be selected carefully so that they represent the actual network traffic as well as possible. For example, when measuring network delay the use of ICM P packets is not a good choice since they are put to lower priority in most routers and thus are not treated as normal traffic. UDP packets should be used instead to get a more realistic view of the network delay. Also such things as packet size and sending rate can be issues.

- *Metrics:* A metric is a quantity related to the performance and reliability of the internet. It can also be said to be a generic indicator of how the network performs. One single measurement result of a metric is called a singleton metric, a set of distinct measurement results (singletons) is called as a sample metric and a statistic calculated over a sample metric is called a statistic metric [5].

  For example, a single active UDP echo test run between two hosts produces a round-trip time result that is considered a singleton metric. The same test repeated for *n* times in a row produces a sample metric. The mean of all measured round-trip values in the previous sample metric can be defined as a statistic metric.

  The IETF IP Performance Metrics (IPPM) working group has proposed several metrics and procedures for accurately measuring and documenting the metrics. The following metrics have been published in a series of RFCs:

  i. Connectivity (RFC 2678)
  ii. One-way Delay (RFC 2679)
  iii. One-way Packet Loss (RFC 2680)
  iv. Round-trip Delay (RFC 2681)
  v. One-way Loss Pattern Sample (RFC 3357)
  vi. IP Packet Delay Variation (RFC 3393)
  vii. Packet Reordering Metrics (RFC 4737)

  Other metrics such as Bulk Transport Capacity and Link Bandwidth Capacity are being developed by the IPPM.

- *Intrusiveness*: Active network measurement creates an additional load on the measured network and thus uses some of the available bandwidth. Intrusiveness is the property of a measurement tool that describes how much of the available bandwidth the tool consumes. For example, a tool or mechanism that consumes 90% of the available bandwidth on a network path can be considered intrusive. A tool that generates small UDP-packets to measure RTT every now and then can hardly be said intrusive (assuming that the available bandwidth of the path is not exceptionally low). According to [6] an active measurement tool or technique can be considered intrusive if its average probing load on the network during a measurement is significant when compared to the available bandwidth in the path.

III. Active Measurements

In this section, we outline the passive, active, and hybrid measurements followed by the discussion of passive vs. active measurements.

A. *Passive Network Measurements*

In passive network measurements data is gathered by passively listening to network traffic for example by using (optical) link splitters or hubs to duplicate a link's traffic (figure 3) or by monitoring buffers in routers. Most of modern devices have some sort of built-in passive measurement mechanisms like RMON which can be used to gather different types of data from the devices such as the number of sent bytes, lost packets and other interface statistics. These built-in mechanisms usually produce only highly aggregated data and thus provide only little information on the network state or traffic behavior. Data created by these mechanisms can often be fetched by using the SNMP protocol. Another mechanism is IPFIX [8] which gathers IP flow data and then pushes it to pre-configured receivers e.g. a central monitoring server.

Results acquired from passive measurements rang from bandwidth usage and protocol distribution to intrusion detection. Ethereal (nowadays called Wireshark) and tcpdump are among the most used passive measurement tools.

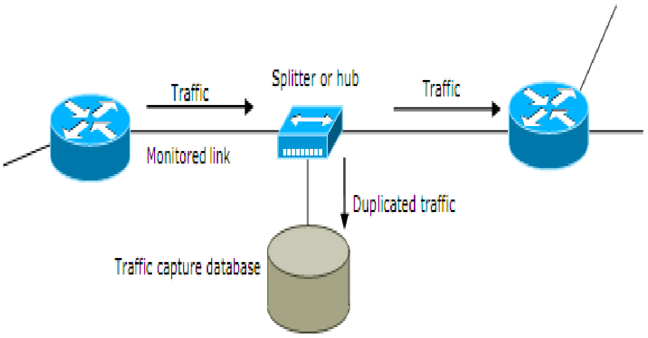


Figure 3. *An example of a passive network measurement.*

The main problem of passive measurements is the amount of data that is generated. If we assume a gigabit link with a utilization of 60% (on IP-layer) and an average packet size of 300 bytes, then the capture rate is about 250000 packets per second. The traffic rate is 75 mebibytes (MiBs) per second and thus the storage space needed for one hour trace is 270000 mebibytes (=270 gibibytes).

If there are several capture points in the network, the amount of captured data is going to a problem. Depending on the type of measurements, several compression methods are available: all irrelevant data could be removed from the captured packets including the payload and some of the header fields.

Normal compression methods can be used to remove redundancy from packets (for example gzip can be used to further reduce the required storage space) [9]. Also, traffic sampling can be used when full traffic analysis is not required. Sampling can drastically reduce the amount of storage space needed but it has some drawbacks (difficulty of flow analysis [9]) and not all sampling methods produce good results [10]. Different sampling methods are discussed in [11].

If only the IP and transport layer headers were stored (40 bytes per packet), the example calculation above would yield a traffic rate of 10 megabytes per second and 36 gigabytes of storage space required for a one hour trace.

The analysis of the captured data is also an issue; on-line analysis is difficult because of the large amount of data. If the capture is made from an operational network, there are privacy issues that needed to be taken into account. This means that the captured traffic has to be modified in such a way that the IP- addresses are anonymized and the payload data has to be removed. A short discussion about the sensitivity of IP header fields and a method to anonymize packets is given in [12].

There are some advantages in passive measurements over active measurements. Passive methods do not create additional traffic thus they do not disturb the network and they provide an accurate representation of the network traffic.

B. Active Measurements

Active measurements generate special probe packets that are sent over the network to, for example, measure the time it takes for the packet to reach the other end of the network (one-way delay), the available capacity of a network path or the response time of an application. Unlike passive measurements, active measurements generate additional network traffic so they may possibly disturb the normal traffic flow. This is why active measurements have to be carefully planned before execution and usually the bandwidth reserved for the probe packets is limited to fewer than 5 percent of the path's total capacity. This is the case in most SLA-measurements where the measurement is done constantly meaning the test traffic and customer traffic shares the same bandwidth.

Some methods (e.g. SLOPS, see Section IV) used for measuring the available bandwidth on a path consist in sending probe packets at an increasing rate and recording the rate at which the probe's delays start to rise (meaning the packets are being queued at some point) [13]. These methods will cause perturbations in the normal traffic flow although the perturbations are usually short. Heisenberg's (Werner Karl Heisenberg, December 5, 1901 – February 1, 1976, Germany) uncertainty principle can be interpreted to state that the act of measurement itself introduces (an irreducible) uncertainty to the measurements [14]. This is true in the case of active network measurements and especially in active packet-loss measurements, where the probe packets may cause congestion and therefore packet-loss. Passive measurement does not have this issue as no additional traffic is inserted into the measured system.

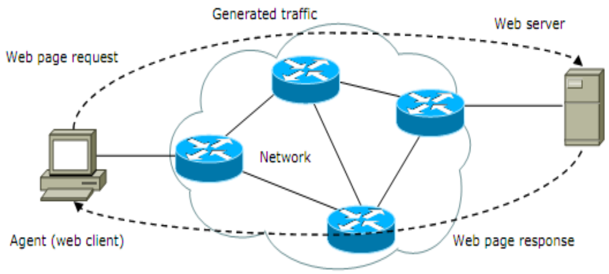


Figure 4. *An example of active network measurement.*

Active measurements do not require huge amounts of storage space and they can be used to measure things that are not possible by using passive measurements. Also, when using active probing, there are no privacy issues since the data used does not contain any private information. All active probe packets are artificial i.e. they are generated on demand and thus they usually contain only random bits as payload. The example presented in Figure 4 shows how active probing can be used to measure the response time of a web server. A measurement device or a software agent installed on a normal PC sends web page requests across a network and records the response time.

The most well known active measurement tools are probably traceroute and ping which are built in to most operating systems. These two tools will be presented in more details later

C. Hybrid Measurements

Combining active and passive measurements is called hybrid measurement. An example of a hybrid measurement (Figure 5) could be a scenario where active probes are sent over a network and their progress is monitored by passive means during the measurement. This type of arrangement allows the measurer to track the path of the probes and record the intermediate and end-to-end delays. This is something that is not possible by doing only active probing.

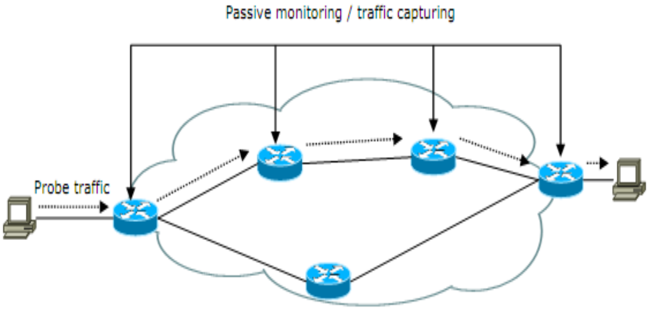


Figure 5. *An example of a hybrid measurement.*

The above scenario required that the measurer has administrative access to the intermediate routers and is thus not suitable to Internet scale measurements. It must be noted that since hybrid measurements use both passive and active means, they share all the same issues as passive and active measurements

D. Active vs. Passive Measurements

Active and passive measurements produce different kinds of information and the results do not necessarily correlate well. A more complete picture of the health of a network can be gained by combining results from both active and passive measurements (hybrid measurements). Although the focus in this thesis is on active measurements, differences in active and passive measurements will also be discussed briefly.

Passive measurements are best suited to situations where capture points can be freely selected. This is true in situations where the whole network is owned and operated by a single organization (eg.. corporate premises networks). This allows traffic to be captured from any point on the path from the sender to the receiver.In situations where it is not possible to select capture points freely, active measurements have to be used. This is often the case when measuring delay performance of a VPN which is carried over multiple ISPs.

Active measurements can be made over a network path that has parts which are not controlled by the measurer.

When it comes to accuracy of measurements, passive methods are often more accurate. For example packet loss can be measured very accurately by monitoring router buffers along the network path. Also, available bandwidth can be accurately measured by monitoring link usage on routers. Both above mentioned measurements are difficult to do accurately with active probing.

IV. Active Measurements

In this section, we list out active measurement mechanisms, methodologies and tools.

A. Layer 2 Measurements

Measurement mechanisms and techniques on the link layer are presented in the section. Traditionally link layer measurement has been minimal, but now as Ethernet technologies are being more widely deployed in the carrier level, Ethernet measurement and troubleshooting tools are becoming more important.

Ethernet OAM: Operations, administration and maintenance (OAM) protocols for Ethernet provide operators the same troubleshooting tools for Ethernet that they have been using on the IP layer. These tools include Continuity Check, Link Trace and Loopback Messages. Continuity Check (CC) messages are used as a heart beat signal to detect connectivity between two endpoints. Link trace message are sent to trace a path hop by hop between two endpoints. This is the Ethernet equivalent for the Traceroute tool on IP layer. Loopback message functionality is similar to ICMP Ping. Its function is to test for connectivity between two endpoints. [47].

UDLD: Unidirectional Link Detection (UDLD) is a Layer 2 mechanism to detect unidirectional Ethernet fiber or copper links but it can also detect for example mis-wirings, interface and media converter faults. A unidirectional link is a situation where a normal bidirectional Ethernet link loses it capability to either transmit or receive data from the Ethernet port at the other end of the link. This kind of fault can cause different types of problems in a network such as spanning-tree topology loops or malfunctioning of other protocols. UDLD monitors the physical configuration of the cables and detects whenever a unidirectional link exist. In the case UDLD detects a unidirectional link, it shuts down the affected port and create an alert for the network administrator. UDLD works with Layer 1 mechanisms to determine a link's physical status and also to detect the existence of physical and logical unidirectional connections. Auto-negotiation is one of these Layer 1 mechanisms: it takes care of physical signaling and fault detection at Layer 1. UDLD performs mutual neighbor identification and neighbor acknowledgment on top of the Logical Link Control (LLC) layer. This makes it possible for UDLD to discover logical one-way miscommunication between neighbors even if a physical layer mechanism has reported the communication to be bidirectional. To be able to detect faults and misconfigurations UDLD uses two mechanisms. The first mechanism is used to advertise a port's identity with hello-packets and to learn the identities of its neighbors. These identities are kept in a neighbor database for a defined time interval (time-to-live, TTL) after which they are considered old and removed. The second mechanism periodically sends UDLD echo messages to its neighbors UDLD enable ports. If the packets are not echoed back in a specific time, the link is considered unidirectional and the report is shut down. [48].

Link Layer (physical) topology discovery: Several proprietary solutions to Layer 2-discovery (e.g. Cisco Discovery Protocol) exist. These solutions are device manufacturer dependent and do not work in heterogeneous network environments. There are also some automatic link layer topology discovery algorithms proposed by the research community [49], [50]. There has been some talk in IEEE 802.1 working group to develop a physical topology discovery protocol [51] but nothing has been standardized yet.

B. Layer 2+ to Layer 4 Measurements

In this section we outline Layer2+, Layer 3 and Layer 4 active measurement mechanisms are presented. The list includes mechanisms built into routing hardware, measurement tools developed by the research community and general measurement techniques. Note that here the term "Layer 2+" means that the mechanism or technique is on top of Layer 2 but not on Layer 3. e.g. MPLS.

The mechanisms and methods are presented in such order that lower layer methods are presented first.

MPLS/LSP-Ping: LSP-Ping [52] is intended as a diagnostic tool for operators to isolate faults in MPLS networks and especially to detect synchronization problems between the data and control planes. It works in two modes: *ping* mode and *traceroute* mode. These two modes correspond to the ICMP *ping* and *traceroute* tools used in IP networks for connectivity tests (*ping*), path tracing and fault isolation (*traceroute*).

LSP-Ping's main use is to verify that packets belonging to a certain FEC really go through the path that they are supposed to. This is done by sending an MPLS Echo Request packet through the same path as all the other packets belonging to this FEC. In ping mode the echo request packets are forwarded just like any other packet in the FEC and once they reach the egress router they are sent to the control plane of the egress router. The control plane checks if the egress router is actually the egress point for the packet's FEC. In the *traceroute* mode the echo packets are sent to the control plane of each LSR along the path to see if the LSR is a valid transit LSR for the packet's intended path. Transit LSRs return information that can be used to check if the forwarding on the router matches what the routing protocols determined as the path for this packet (control plane check against the data plane)

MPLS echo request packets are routed based on the label stack so the IP address of the receiving end is never used in the forwarding decision. This means that the sender of the echo request packet does not have to know the IP address of the egress router. To prevent packets from causing confusion in the network in case of LSP failure, the destination IP address should be selected from the 127/8 address range (internal host loopback address, localhost) [53]. This way the packets that happen to drop out from the LSP are not IP forwarded but are silently discarded instead [54]

Juniper Real Time Performance Monitor (RPM): RPM [55] is an active measurement mechanism built into Juniper routers to actively monitor the performance of network paths between two or more Juniper devices. By sending a constant flow of probes routers can monitor for example the level of delay inside a VPN. Main use for RPM is performance monitoring on Layers 3 and 4 and it can also be used to generate SNMP traps on SLA violations. So, for example if the delay level inside a VPN rises above some predetermined value, then an alarm is generated. Alarm generating thresholds can be configured so that the monitoring and analysis of the measurement results are simplified. All results can be directly used from the CLI, fetched via SNMP or exported to external network management applications.

RPM supports RFC 2925 MIB (Management Information Base) with extensions. The RFC defines a MIB for performing remote ping, *traceroute* and IP or DNS lookup operations at remote hosts meaning that a Juniper router can be used to initiate one of the mentioned operations on another Juniper router.

The following types of probes are supported by RPM with Differentiated Services Code Point (DSCP) marking:

i. ICMP Echo
ii. ICMP Timestamp
iii. HTTP Get
iv. UDP Echo
v. TCP Connections

The probe packets can be given a priority over regular data packets on input interfaces in which case the probes can reach their destination even if there is congestion. Such results as minimum, maximum and average round-trip time, RTT delay variation and standard deviation, number of probes sent and percentage of lost probes are produced by the probes

Cisco Service Assurance Agent /IOS IP Service Level Agreements: Formerly known as the Service Assurance Agent (SAA) the Cisco IOS IP SLAs [56] is much like its Juniper counterpart RPM. It is a built in feature of the Cisco IOS devices allowing active probing and thus active monitoring. The probes have several configurable options such as UDP/TCP port numbers, ToS field, VRF instance, source and destination IP addresses and web URL. Since IP SLAs is Layer 2 transport independent it can be configured to run end-to-end over a heterogeneous network.

IP SLAs allow the collection of the following performance metrics:

i. One-way delay
ii. Round-trip delay
iii. Delay variation
iv. Packet loss
v. Packet ordering
vi. Voice quality scoring
vii. Network resource availability
viii. Application performance
ix. Server response time

The data collected by the probes can be accessed via CLI or SNMP MIBs and it can be used by third party performance monitoring applications.

Active network layer topology discovery: With the speed networks are growing and changing today getting a clear picture of a network's topology is becoming more and more difficult. Topology information is valuable for network resource managers and administrators planning server placements

Researchers also need topology information to simulate networks. Different tools and methods have been proposed for active network topology discovery [57], [58], [59]. Most of these tools are based on SNMP or traceroute-like methods (sending hop-limited packets to a destination address and waiting for an ICMP message indicating IP TTL expiration).

Reachability / Ping: One of the most basic active network measurements is testing if a certain host is reachable available). This can be done easily by sending an ICMP echo_request (ICMP Type 8) packet to the target host which then elicits an ICMP echo_response. The most well known reach-ability testing tool is the ping tool originally written by Mike Muuss in 1983. Its usefulness and simplicity has allowed it to rise to a status where it is built in to nearly every operating system. In addition to measuring reach-ability ping can also be used to estimate (measure) round-trip delay and packet-loss.

Even ping has its problems. Many ISPs have begun to filter out ICMP echo requests because of growing number of Internet worms using them to search for potential targets Also, some hosts do not reply to echo requests in purpose to hide their presence. These facts diminish the usefulness of the ping tool, but most of the time it still is the most valuable tool a network engineer has when performing troubleshooting.

Route discovery / Traceroute: Finding out what route a packet takes on its way through a network can be done by exploiting the time-to-live (TTL) field of the IP header. The TTL field on an IP packet is decremented every time the packet is processed by a router. When the TTL counter of an IP packet reaches zero, the packet is dropped and an ICMP TTL Expired –message is sent back to the sender. By sending packets with increasing TTL fields (starting from 1) each router on the path can be elicited to send an expiration message thus all routers can be identified. The method described here was first used in the famous *traceroute* program written by Van Jacobson [60]. Known problems exist in the *traceroute* method as presented by Vern Paxson in [61]:

The method assumes that all intermediate routers send ICMP messages while this is not true in all cases as some routers are configured not to send or reply to ICMP messages because of security concerns.

Layer 2 devices are transparent to the method: switches and different link layer technologies cannot discovered with *traceroute*.

The traceroute tool is similar to the ping tool in its popularity as it is built in to most of the current operating systems. Unfortunately, it suffers even more than ping from the filtering issues since not all routers reply to ICMP messages. Often traceroute returns only the first few routers on the path.

Path MTU discovery: The largest packet size that can be sent on to a link without fragmenting the packet is called the Maximum Transmission Unit (MTU). An arbitrary path between two nodes in a network may have links that have different MTUs. The smallest MTU on the path between these two nodes is the Path MTU (PMTU). When sending large amounts of data across a network it is efficient to use the largest MTU possible; using a smaller packet size would waste resources. RFC 1191 [62] defines one ICMP based Path MTU discovery mechanism. This mechanism has several problems which are discussed in RFC 2923 [63].

The mechanism defined in RFC 1191 uses the IP header's Don't Fragment bit to discover the PMTU of a path. A source node first assumes that the PMTU is the MTU of the first hop. With the DF bit set in every packet, the node starts to send traffic to the destination node. If a router on the path notices that the datagram cannot be sent to a next hop without fragmentation, the router drops the packet and sends an ICMP Destination Unreachable message with the code "fragmentation needed and DF set" back to the source node [64].When a source node receives these messages, it automatically reduces the size of the packets and thus the PMTU until it receives no more error messages. However, the source must never reduce its PMTU estimate below 68 octets, since, according to RFC 791, every router must be able to send packets of 68 bytes without fragmenting them. [62].

Increases in the PMTU can be detected by periodically sending packets with increased PMTU (e.g. by setting the PMTU back to the MTU of the first hop). Since this will most likely result in more of above mentioned ICMP messages, it is recommended that the testing is done infrequently. Decreases of the PMTU are detected by ICMP "fragmentation needed and DF set" messages.

Available bandwidth measurement methods and tools: Some applications benefit from knowing the amount of bandwidth available on a network path so that they can adapt their sending rate and share the bandwidth more fairly. Such applications include multimedia content adaptation, dynamic server selection, peer-to-peer applications and congestion control transports. Measuring (or rather, estimating) available bandwidth with active probing is required when all routers along a network path are not controlled by the measurer (passive measurement methods cannot be used).

When measuring available bandwidth by probing, it must be noted that all current methods merely give approximations of the current bandwidth usage of a path. The available methods used are not very accurate especially when used to measure high bandwidth links [6].

There are four major techniques that are used when estimating available bandwidth. A brief overview of these techniques is given here; a more thorough review can be found for example in [6].

i. *Variable Packet Size (VPS)* technique attempts to estimate the capacity of each link (hop) along a path. VPS does this by sending different sized probe packets from the source node to all nodes along the path and measuring the RTT to each hop as a function of packet size. The inverse of the RTT vs. packet size slope is the capacity estimate of a hop. The minimum of all link capacity estimates is the end-to-end path capacity. This method was first used by Bellovin [65] in 1992 and later by V. Jacobson in the pathchar-tool [66]. It was later used in such tools as Clink and Pchar [67], [68].

ii. *Packet Pair/Train Dispersion (PPTD)* [6] technique estimates the end-to-end capacity of a path. It does this by sending multiple identical (in terms of size) packets back-to-back and by measuring the dispersion of the packets at the receiver side. The narrow link on the path causes an increase in the dispersion of the packets. This increase can be used to estimate the capacity of the narrow link. The difference in packet pair and packet train techniques is that the latter uses multiple packets while the former uses only a pair of packets. The dispersion of a packet train (or pair) is the time measured from the last bit of the first packet to the last bit of the last packet. Such tools as bprobe, nettimer, pathrate and sprobe implement the PPTD methodology [69], [70], [71], [72]. Self-Loading Periodic Streams (SLOPS) technique [13] measures the end-to-end available bandwidth of a path. The operating principle is to send sequences of equal sized packets at an increasing rate and to monitor the one-way delay variations experienced by the packets. An increase in delay indicates congestion on the path's tight link. SLoPS uses an iterative binary search -like method to find the optimal sending rate i.e. the rate that does not cause queuing and yet is able to fully utilize the path's available bandwidth.

iii. Trains of Packet Pairs (TOPP) [74] is another end-to-end available bandwidth measuring technique. The TOPP method is much like the SLOPS method but instead of just estimating the available bandwidth it is also able to estimate the tight link on the path. TOPP adjusts its sending rate linearly. Pathload, pathChirp and IGI are tools that use either the SLOPS or the TOPP method to measure the end-to-end available bandwidth [75], [76]. Comparative analyses of available bandwidth measurement tools and methodologies are presented in [1], [77].

Lai and Baker present a hybrid technique called packet tailgating in [70]. Tailgating combines VPS and packet pair techniques to measure the end-to-end capacity of a path in two phases. The first phase, called the sigma phase, measures the characteristics of the whole path, while the second phase, called the tailgating phase, measures the characteristics of each hop individually.

The idea in tailgating is to send a large packet (tailgated) followed by a small packet (tailgater) for each link on the path. The larger packet's Time to Live (TTL) is set to expire on the link under measurement. The smaller packet will continuously queue behind the larger packet until the larger packet's TTL expires in which case the tailgater will continue to destination without queuing thus capturing the important timing information. It is assumed that the larger packet will not be queued, while the smaller packet is always queued after the larger one.

Packet tailgating is less intrusive than the previously mentioned techniques. STAB is a lightweight tool that combines self-induced congestion, packet tailgating and packet chirps to measure and locate tight links [78].

Bulk transfer capacity: IPerf, TReno and Cap tools implement the BTC measurement methodology [79], [80],[81]. IPerf measures BTC by establishing a TCP connection to a selected host and trying to send data as fast as possible. It uses the TCP implementation of the underlying operating system (e.g. Windows, Linux). TReno tool emulates TCP by using low TTL UDP or ICMP Echo packets: probe packets elicit TTL Expired ICMP packets from the receiving host thus simulating TCP ACKs. Cap also uses UDP packets to emulate TCP but instead of using ICMP to simulate ACKs, it sends UDP packets from the receiving end every time it receives a packet.

TReno is a non-cooperative tool meaning that it does not require software to be installed to the receiving end. IPerf and Cap are both cooperative thus they require software to be installed on both ends of the measurement.

IPMP: The Internet Protocol Measurement Protocol (IPMP) is a proposition to create a protocol that is designed purely for active network measurements. The protocol is basically an echo protocol allowing routers to participate in the measurement by inserting path information in the probe packets. IPMP can be used to measure one-way and round-trip delay, packet loss and one-way path length. [35]

OWAMP: One Way Active Measurement Protocol (OWAMP) defined in RFC 4656 [82] aims to provide an interoperable high precision mechanism to measure one-way delay in Internet environment. OWAMP has been designed with security in mind: the protocol traffic is hard to detect (plain UDP packets) and manipulate which makes it more difficult for others to interfere with the measurements. Test traffic can be encrypted which makes it impossible for attackers to alter the timestamps undetectably. Authentication is also supported by adding an HMAC (a keyed Hash Message Authentication Code) code to control messages.

The OWAMP architecture is separated to different roles in order for to it be more flexible.

Five roles are defined in the RFC:

i. Session-Sender: the sending host of the test session.
ii. Session-Receiver: the receiving host of the test session.
iii. Server: manages the test sessions, configures per-session states in the session endpoints, and returns the results of a test session.
iv. Control-Client: initiates requests for test sessions, triggers the start or termination of test sessions.
v. Fetch-Client: initiates requests to fetch the results of completed test sessions.

Figure 9 shows a simple example of how a test could be set up. Host A plays the roles of a control-client, fetch-Client and session-sender, while Host B acts as a server and a session-Receiver. This way there is no need for other devices to take part in the measurement except for the two endpoints.

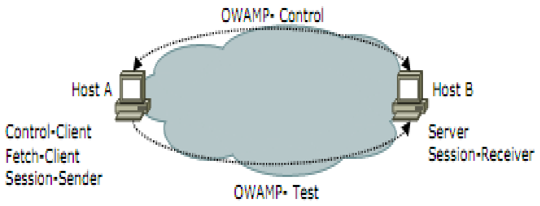


Figure 9. *An example of a simple OWAMP test setup where several roles are played by one end host.*

The OWAMP protocol is divided into two separate parts (protocols): the control part and be used to initiate, start or stop a session or to fetch test results from the test receiver. The test protocol, layered over UDP, handles the sending of test packets between the sender and receiver using the IP addresses and port numbers negotiated during the session initialization.

The principle of operation in OWAMP is simple: the test packets are sent from the sender to the receiver and the packets' timestamps (send and receive times), sequence numbers and TTLs are recorded on arrival.

As OWAMP measures the one-way delay by comparing the timestamps on the sender's and receiver's end, it is clear that the clocks of both the sender and the receiver have to be synchronized.

Two implementations of OWAMP have been made to date: Internet2's OWAMP [83] and JOWAMP [84]. The developers of J-OWAMP report successful testing of interoperability of these two implementations in [85].

TWAMP: While OWAMP is aimed at measuring one-way delay the Two-way Active Measurement Protocol (TWAMP) adds two-way or round-trip measurement capabilities to the OWAMP methodology and architecture. TWAMP also consists of two inter-related protocols: the control and test protocols. The TWAMP protocol is still in draft status [86].

The TWAMP architecture is similar to OWAMP's but with some exceptions. The Session Receiver is replaced by the Session-Reflector which is capable of creating and sending test packets when it receives test packets from a Session-Sender. Unlike the Session-Receiver it does not collect any information from the test packets as round-trip delay information is available only after the reflected test packet has been received by the Session-Sender.

Another exception is that the Server component does not have the capability to return the results of a test session as the Session-Reflector it is associated with does not collect any results. Consequently, this means that there is no need for a Fetch-Client and thus it does not exist in the TWAMP architecture.

Again, one host can play one or more of the roles. An example of a minimal setup is presented in the figure below (Figure 10) where Host A initiates the measurement and Host B reflects the received test packets.

The TWAMP Internet draft specifies also a lighter version of TWAMP called the Session Sender are performed by the sending host and the role of Session-Reflector by the responding host thus there is no need for the TWAMP control protocol.

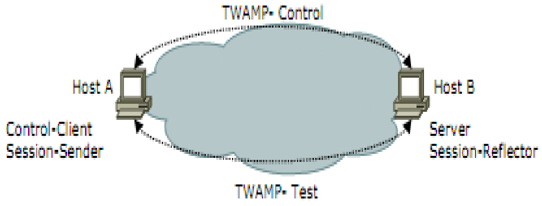


Figure 10. *A simple example of a TWAMP test setup where multiple roles are played by one entity.*

The Control-Client establishes a test session with the Server through non-standard means since they are located on the same host (this means has not yet been defined to date by the working group). Once the session is established, the sender starts to send test packets to the responder who then reflects them back so that the sender can collect round-trip time data.

Brix Networks have announced [87] that they have made an implementation of the TWAMP draft and successfully tested it with another implementation by Allied Telesyn. According to the driving force behind the TWAMP project, Kaynam Hedayat from Brix Networks, there are also other parties developing their implementations of TWAMP.

V. Passive Monitoring Methods

This section offers a closer view to network monitoring. The first part concerns the traffic and which are the most important fields of different protocols in the area of network monitoring. The second part is a discussion about where to monitor traffic in a case of troubleshooting networks.

A. Traffic

Today pure Internet Protocol (IP) networks or connections over core networks with customer data are almost non-existent. It is preferable to transfer customer data over encrypted Virtual Private Networks (VPNs) and Multiprotocol Label Switched (MPLS) networks. This section discusses first the IP model and which fields of the IP protocol are important in the troubleshooting process. After introducing the most basic IP model, it proceeds to the more complex MPLS troubleshooting. The section ends with a few words about encrypted VPNs and their troubleshooting.

IP Traffic – The 5-layer TCP/IP Model

Communication protocols used in the Internet can be divided into distinct hierarchical structures. The most general models are the 7-layered Open Systems Interconnection (OSI) model and the 5-layered TCP/IP model. This survey concentrates on the latter. In Figure 3.1 the TCP/IP model is shown with common protocols. A sand-glass illustrates in the figure the amount of protocols on each level.

When traffic is studied with passive monitoring, the layers of the greatest interest are the three uppermost: the network, transport and application layers. The next three sections of this paper discuss these layers in a greater detail.

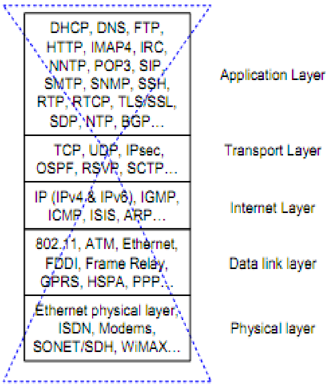

DHCP, DNS, FTP, HTTP, IMAP4, IRC, NNTP, POP3, SIP, SMTP, SNMP, SSH, RTP, RTCP, TLS/SSL, SDP, NTP, BGP...
Application Layer
TCP, UDP, IPsec, OSPF, RSVP, SCTP...
Transport Layer
IP (IPv4 & IPv6), IGMP, ICMP, ISIS, ARP...
Internet Layer
802.11, ATM, Ethernet, FDDI, Frame Relay, GPRS, HSPA, PPP...
Data link layer
Ethernet physical layer, ISDN, Modems, SONET/SDH, WiMAX...
Physical layer

Figure 3.1: The TCP/IP model [Tan02] shown with common protocols.

Network Layer

In the TCP/IP model, IP acts as a network protocol. There are two versions of the IP protocols - version 4 and version 6. The first is still the most commonly used. The Ipv4 datagram header structure is illustrated in Figure 3.2 [56]. The IPv6 datagram header structure is shown in [DH98].

Transport Layer

The transport layer resides at the top of the network layer. Its purpose is to manage the higher level functionalities of communication. In this layer there are two protocols which are used mostly: Transmission Control Protocol (TCP) and User Datagram Protocol (UDP).

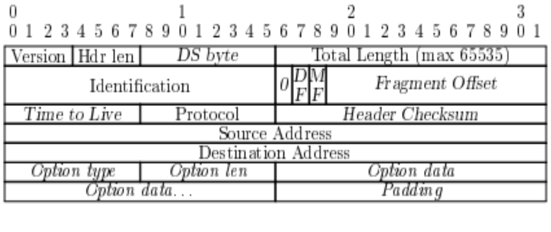


Figure 3.2: The IP datagram (version 4) header structure [Pos81b].

TCP header structure is illustrated in Figure.3.3

As TCP is a connection-oriented octet-streaming protocol, an ACK1 packet is sent to the sender for signing for every TCP packet when it has reached its destination. A connection is established every time with a SYN2 packet and it is cleared properly with a FIN3 or RST4 packet. TCP is used for communication that needs to be reliable, for application layer protocols such as Secure Shell (SSH), Hypertext Transfer Protocol (HTTP) and File Transfer Protocol (FTP).

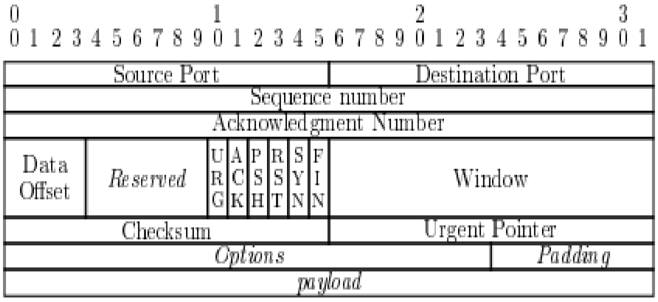


Figure 3.3: The TCP header structure [Pos81c].

UDP is a connectionless message-based protocol. It is a more simple protocol than TCP - there are not any acknowledgements to guarantee an end-to-end connection. Additionally, UDP is lacking any congestion avoidance and control mechanisms. UDP is primarily used for traffic which is time sensitive, for example Network Time Protocol (NTP), Real-Time Transport Protocol (RTP) and Domain Name Service (DNS). The UDP header structure is shown in Figure.3.4.

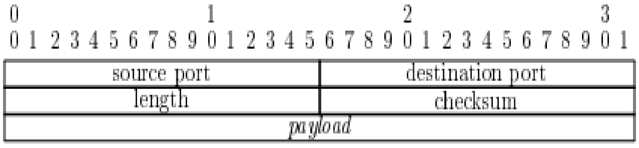


Figure 3.4: The UDP header structure [Pos80].

Application Layer

In the previous section mentioned protocols e.g. HTTP, FTTP and RTP are application layer protocols. They usually run on the top of TCP or UDP. Application layer protocols are for more specific use - for each service there is an own protocol for only that use.

Purpose of different protocols in Passive Monitoring:

On the different levels of TCP/IP model there are a lot of information which is useful for passive monitoring. IP resides on the network layer. From the IP header we can use the source and destination address fields, and in addition, the IP protocol field. The most important fields from the transport layer are the source and destination port ones. With these used application layer protocol can be studied. A list of relation between commonly used port numbers and application layer protocols is maintained by Internet Assigned Numbers Authority (IANA) [58]. In addition, the same organization maintains a list of the IP protocol numbers [59].

Now we have so called 5-tuple - five fields of data, with which every packet can be recognized. With these fields a flow can also be identified.

Traffic Flows

A flow is a series of packets traveling from source to destination [60]. It is unidirectional. Sometimes a flow is defined as bidirectional (packets to both direction belongs to the same flow). IP routing is generally asymmetric; therefore bidirectional flow study in the middle of core network may be impossible.

A flow is defined by Quittek et al. with following words [61]: "A flow is a set of packets passing an observation point in the network during a certain time interval t. All packets belonging to a particular flow have a set of common properties derived from the data contained in the packet and from the packet treatment at the observation point". For example, the 5-tuple can be used as a property in defining a flow. There do exist also other definitions of the term 'flow' being used by the Internet community.

A timeout can be a separator for different flows, for example if there is t seconds between packet A and packet B with the same 5-tuple when reaching the destination, we can assume them belonging to different flows. First t has to be defined, it can be anything between 0 and ∞, and the start and end time of a flow is defined by a data analyzer. But according to a study by Jain et al. it is usually 60 or 64 seconds [60].

With some protocols it is able to utilize their special features, functions, or behaving in defining a flow. For example, with TCP a flow can be defined as groups of packets, whose first packet is SYN and the last is a FIN or an RST packet.

MPLS: Measuring MPLS is a special problem, since there is no end-to-end identifier–there is no unique source/destination pair in header like in IP packets. Header for a path changes on traversing a node, hence it changes over every physical link. This makes it difficult to track.

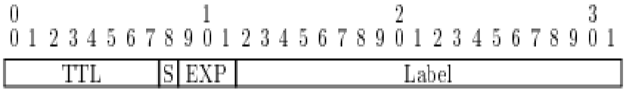


Figure 3.5: The MPLS header structure [RTF+01].

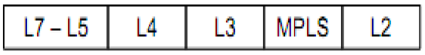


Figure 3.6: MPLS layer is located in between the OSI layers 2 and 3.

Figure 3.5 represents how an MPLS shim header looks like. The MPLS header is a 32-bit length header conformed by four parts: 20 bits are used for the label, 3 bits for experimental functions (nowadays for Class of Service (CoS) use), 1 bit for stack functions (the S field) and 8 bits for the time-to-live field (TTL). In Figure 3.6 we can see how an MPLS layer is located – it is between the OSI layers 2 (data link layer) and 3 (network layer). Sometimes it is said that MPLS is located on the layer 2.5.

Virtual Private Networks

Virtual Private Network (VPN) is a method to connect two private networks together over a public network (Internet) in such a way that these networks function as if they were connected physically to each other.

Generally, there is a frame inside of a frame. But when encryption is in use, the inner frame is encrypted. Then the whole data is hidden from testing and troubleshooting. Encrypted VPNs can be made by using several encryption methods: IPSec5, SSL VPN6. Basically, using VPN does not necessarily mean that encryption is in use. It can also function just as a normal IP pipe between two LANs.

MPLS VPNs: MPLS VPN can be built both on the layer 2 or the layer 3. This VPN is not encrypted, which means that we have a possibility to take a look inside of the whole packet if necessary. There are a number of failure points in MPLS VPN networks that can be monitored with passive monitoring. For example, the following faults can be detected:

- MPLS VPN label allocation verification. An idea is to monitor VPN labels and then confirm them to the label allocation plan. For example, in changes to the network topology, this method is able to check that Provider Edge (PE) routers work properly.
- Resource reservation. By monitoring an acquired metric of different VPN labels in links and mapping this to reservation allocation of different VPNs it is possible to observe how well resource allocation works. The simplest metric to monitor is the amount of traffic, since this requires only one monitoring point. For example, monitoring the delay needs more monitoring points and more computation. But in reality, using passive monitoring methods to follow resource reservation situation on links would be almost impossible to implement

Since there are no end-to-end identifiers in MPLS, some work needs to be done in order to find an end-to-end path in a network. This happens in a way that Provider (P) and Provider Edge (PE) routers in a network can be interrogated periodically by SNMP queries on LSP connections and LDP or RSVP connection status of the routers. The LSP connection information can be obtained from the LSR MIB in each node. By using the cross-connect and other tables in the MIB, incoming labels and interfaces are able to be mapped to egress labels and interfaces. A database from this information can be created consisting of all the LSPs between all the routers in the network. Links in the database can be created to illustrate end-to-end connections between PE routers.

Encrypted VPNs

Encryption causes some problems from the perspective of application level troubleshooting. Now internal IP packet is encrypted, so all the important data concerning of troubleshooting of application level is hidden. We can conclude something for example from packet length e.g. VoIP traffic can be observed from the data quite reliable. But for network troubleshooting encryption does not affect at all, since the IP header is not encrypted.

Depending on protocol, Security Parameter Index (SPI) and Internet Key Exchange (IKE) negotiations can be observed and one can create some statistics based on these. For example we know that packets with the same SPI value belong to the same user, IP address or subnets depending on encryption rules. Thus we can make some statistics for this kind of flows: Inter-Arrival Time (IAT), the amount of transferred data, packet loss by examining packet sequence numbers.

Header vs. Packet information: Usually information obtained from the headers is enough for analyzing or troubleshooting. In some cases we, however, need the information located in data part of packets. For example routing information of routing protocols – what kind of subnets are they advertising. Now IP address and port number in the header do not tell which virtual hosts will be used, this information is only located in the data part of a packet. Choice of capturing of the whole packet or just headers also affects on the amount of captured data. And, usually passive monitoring needs no special specification or decisions before starting capturing but at this point it is needed: it is needed to decide whether to capture the whole packet or just the headers. For example, in the tcpdump program this is done by defining how many bytes are captured – in this case it should be known which media and protocol stack are in use to capture the right amount of bytes from each packet.

B. Monitoring Traffic

Ideally, passive monitoring points should be attached to links where the greatest and widest sample of traffic can be observed, where packets travelling in both directions between servers and clients are visible and where routing variations have minimal impact [54]. This way we can ensure that we have a lot of traffic and we see a lot of "normal" events. But more, however, can be learnt if monitoring points are in selective places, like in links which are connected to some specific sites (e.g. campus area, server farms, large modem banks, the interfaces of interesting devices). These kind of places can produce interesting information and comparisons.

Monitoring at Single Point and at Multiple Points:

The single point monitoring system is well suited for monitoring the performance of Local Area Networks (LANs) where only one point is connected to WAN or larger network. There are, however, some limitations for single point monitoring – there is no possibility to handle time issues. Only RTT can be obtained from such as protocols which have bidirectional communication. Generally, it can be said that with single point measurements it is possible to perform count of different events, calculate throughput and distribution of different protocols.

With multiple points monitoring it is possible to expand the amount of metrics which can be obtained from captured data. Time-related things, such as delay and jitter can be calculated. In addition, it is possible to study the behavior of traffic flows and changes in used routes. The greater the number of monitoring points in the network, the better and more precisely different events can be observed. However, it is good to remember that the greater number of monitoring points in the network, the more complicated and more time-consuming analyzing the data gets. The traffic matrix can even be calculated from single point monitoring data. Then the matrix is only able to present the amount of traffic or other events between different source-destination pairs, for example. But by calculating traffic matrices from data got from multiple point monitoring measurements, we can also present the used path between source-destination pairs. Events can be, for example, delay (OWD not possible in single point monitoring), the amount of traffic, and availability of connection. The traffic matrix is a required input in many network management and traffic engineering tasks, where typically the traffic volumes are assumed to be known. However, in reality, they are seldom readily obtainable, but have to be estimated. The estimators use as input the available information, namely link load measurements and routing information. [62]

Time-related metrics need an accurate clock at every monitoring point in order to get reliable results. Clock synchronization can be made with the Network Time Protocol (NTP) [63], the Global Positioning System (GPS) or Code Division Multiple Access (CDMA). The clock synchronization in a computer cluster with GPS is discussed in [66].

A Cisco Whitepaper presents that the accuracy of an NTP adjusted clock over a WAN network is within a 10 millisecond level and a 1 millisecond level in a LAN network.

In GPS the accuracy of the PPS signal is about 10 µs. A more accurate network time protocol, Precision Time Protocol (PTP), has been developed by the IEEE (IEEE 1588). The purpose of PTP is to make possible to bring as accurate time as it is in Synchronous Digital Hierarchy (SDH) and Plesiochronous Digital Hierarchy (PDH) networks over the Ethernet networks. PTP is designed for local systems requiring very high accuracies beyond those attainable using NTP [63]. As a disadvantage the protocol needs support of all the network devices to work. Nieminen measured the accuracy of PTP in his Master's thesis [64]: he found that the accuracy for a network of only one empty straight Ethernet cable is approximately 50 ns, for a network with a pass-through device it is 100 ns, and for a network with a hub and meaningful load it is 500 ns.

A Finnish company called Flexibilis7 has tested their own PTP devices. They managed to get the accuracy of 2 ns for two devices with one single Ethernet link (1 Gbps optical fiber), and the accuracy of 3 ns for four devices in a chain connected with three Ethernet links. [67]

The complexity of the measuring process increases when we are moving from single point measurements to multipoint measurements as can be seen in Figure 3.7. Active measurements are generally regarded as being more complex than passive ones [68]. Reasons for this increased complexity are, for instance: handling of captured data, clock synchronization, or making cross analysis over all the captured data. And following a certain packet from a customer to another with multipoint measurements can be sometimes difficult – for example, if a measurement point is located in an encrypted environment (VPN etc.), then there is no identifier available which could be followed through a network in measurement points under a time window.

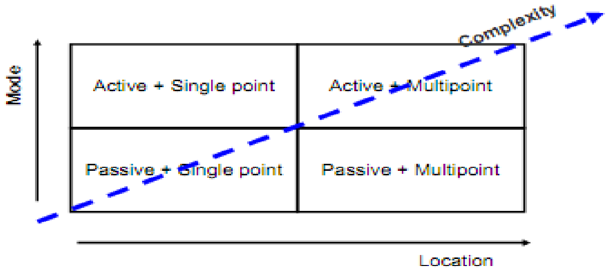


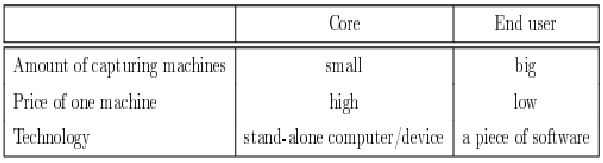
Figure 3.7: Complexity of measurements [Ilv07].

Measuring at Core vs. Measuring on the End user side: There are usually three options to put probes to measure network traffic. The first is measuring at core network – putting probes in core links or routers to capture data from high speed links. This requires permissions and the access equipment bay of operators. The second is to measure on the customer side. The third option is to measure traffic with the user's computer. In this case it is only needed to install measuring software for end user's computer.

The need for these three options is different. The end-to-end performance of a single user tells how well the whole chain between end points is working, but this can tell quite little or nothing about single networks or devices inside the chain.

But end users do not have any possibility to study the behavior of some routing protocol with measurements, whereas this is "a piece of cake" in the core network. If something has to be measured, you have to know first that there is a possibility that the measurable item is visible also in that place where it is measured. In every case it is necessary to consider carefully what and where to measure. In Table 3.1 the measurements of these three options is compared.

Table 3.1: Comparison of capturing data at core or on the customer side.



	Core	End user
Amount of capturing machines	small	big
Price of one machine	high	low
Technology	stand-alone computer/device	a piece of software

C. *Multicast Traffic*

Multicast traffic monitoring based on capturing passively traffic is studied, for example, by Walz et al. in the article "A practical multicast monitoring scheme" [69] and Al-Shaer et al. in the article "MRMON: remote multicast monitoring" [70]. The basic idea in these solutions is to first put capturing points between a sender and a network device (a switch, a hub or a router) and second between receivers and a network device receiving multicast traffic. This is illustrated in Figure 3.8. Every paper in this area has this same idea, they have only been extended with some data analyzing functionalities etc.

Why to use then passive monitoring methods when active approach provides a useful means to investigate particular problems? There are a couple of reasons for this. First, active monitoring is not effectively useful for diagnosing intermittent or short-lived problems. In addition, active approach requires a significant amount of bandwidth for tracking user/group activities and intermittent problems. On the other hand, passive monitoring offers more monitoring information on multicast operations with no traffic overhead. [70]

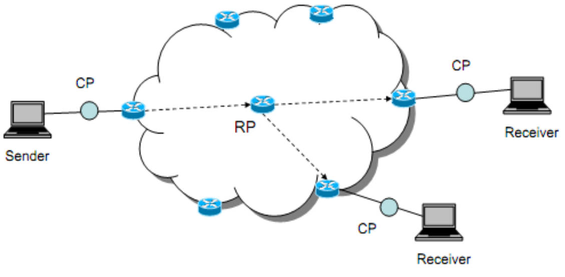


Figure 3.8: The basic idea to measure multicast traffic passively. CP means a capturing point and RP means a rendevouz point.

Figure 3.8: The basic idea to measure multicast traffic passively. CP means a capturing point and RP means a rendevouz point.

In hybrid solutions we can send traffic to a test multicast group and passively capture the data sent and measure, for example, delay parameters. In this way we can measure periodically multicast traffic and to test that it works. Then we can put test receivers at some points and make measurements and analysis between these points and a test sender.

In passive monitoring the measurements can be taken if traffic exists in a certain multicast group.

REFERENCES

[1] Ahmed Ait Ali, Fabien Michaut, Francis Lepage, "End-to-End Available Bandwidth Measurement Tools: A Comparative Evaluation of Performances", IPS-MoMe 2006.

[2] C. Demichelis and P. Chimento, "IP Packet Delay Variation Metric for IP performance Metrics (IIPM)", RFC 3393, November 2002.

[3] Fabien Michaut, Francis Lepage, "Application-oriented network metrology: metrics and active measurement tools", IEEE Communications Surveys & Tutorials Second Quarter 2005, Vol. 7, No.2

[4]] M. Mathis, M. Allman, "A Framework for Defining Empirical Bulk Transfer Capacity Metrics", RFC 3148, July 2001.

[5] Vern Paxson et al., "Framework for IP Performance Metrics", RFC2330, May 1998

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

[7] P. Barford, J. Sommers, "Comparing Probe and Router-based Methods for Measuring Packet Loss", IEEE Internet Computing - Special issue on Measuring the Internet, 2004.

[8] B. Claise, "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of IP Traffic Flow Information", RFC 5101, January 2008 http://www.ietf.org/rfc/rfc5101.txt

[9] Vern Paxson et al., "Framework for IP Performance Metrics", RFC2330, May 1998.

[10] Nick Duffield, "Sampling for Passive Internet Measurement: A Review", Statistical Science 2004, Vol. 19, No. 3, 472–498 http://www.projecteuclid.org/Dienst/UI/1.0/Summarize/euclid.ss/1110999311

[11] M. Peuhkuri, "A Method to Compress and Anonymize Packet Traces", Proceedings of ACM SIGCOMM Internet Measurement Workshop 2001 http://www.imconf.net/imw-2001/imw2001-papers/32.pdf.

[12] M. Jain and C. Dovrolis, "End-to-End Available Bandwidth: Measurement Methodology, Dynamics, and Relation with TCP Throughput", IEEE/ACM Transactions on Networking, vol.11, no. 4, pp. 537-549, Aug. 2003.

[13] M. Roughan, "Fundamental bounds on the accuracy of network performance measurements", Proceedings of the 2005 ACM SIGMETRICS international Conference on Measurement and Modeling of Computer Systems (Banff, Alberta, Canada, June 06 - 10, 2005), SIGMETRICS '05, ACM, New York, 253-264.

[14] A. McGregor and M. Luckie, "IP Measurement Protocol (IPMP)", Internet draft, November 2003, http://watt.nlanr.net/AMP/IPMP/draft-mcgregor-ipmp-03.txt,

[15] M. McFarland, S. Salam, R. Checker, "Ethernet OAM: key enabler for carrier class metro ethernet services", IEEE Communications Magazine, vol.43, no.11, pp. 152-157, Nov. 2005.

[16] M.Foschiano, "UniDirectional Link Detection (UDLD) Protocol", Internet draft(informational), draft-foschiano-udld-01.txt, February 2006.

[17] Hwa-Chun, Hsin-Liang Lai, Shou-Chuan Lai, "Automatic link layer topology discovery of IP networks", IEEE International Conference on Communications (ICC '99), vol.2, no., pp.1034-1038 vol.2, 1999.

[18] R. Black, A. Donnelly, C. Fournet, "Ethernet topology discovery without network assistance", Proceedings of the 12th IEEE International Conference on Network Protocols,vol., no., pp. 328-339, 5-8 Oct. 2004.

[19] "IEEE Interim DP Discussion", http://www.ieee802.org/1/files/public/docs2002/IEEE%20Interim%20DP%20Discussion.pdf

[20] K.Kompella, G.Swallow, “Detecting Multi-Protocol Label Switched (MPLS) Data Plane Failures”, RFC4379, Feb 2006

[21] R. Braden, “Requirements for Internet Hosts - Communication Layers”, RFC1122, October 1989

[22] F. Baker, “Requirements for IP Version 4 Routers”, RFC1812, June 1995

[23] Juniper configuration guide, available online http://www.juniper.net/techpubs/software/junos/junos82/swconfig82-services/html/rpm-overview.html#1019604

[24] Cisco configuration guide, available online, http://www.cisco.com/en/USproducts/ps6350/products_configuration_guide_chapter09186a0080441596.html

[25] B. Huffaker, D. Plummer, D. Moore, and k claffy, "Topology discovery by active probing," in Proceedings of the 2002 Symposium on Applications and the Internet (SAINT) Workshops, available online, http://www.caida.org/publications/papers/2002/SkitterOverview/skitter_overview.pdf

[26] M. Luckie, K. Cho, and B. Owens, “Inferring and debugging path MTU discovery failures”, In Proceedings of the Internet Measurement Conference 2005 on Internet Measurement Conference (Berkeley, CA, October 19 - 21, 2005).

[27] Ramesh Govindan, Hongsuda Tangmunarunkit, “Heuristics for Internet Map Discovery”, Proceedings of INFOCOM 2000, Volume 3, on page 1371-1380

[28] Van Jacobson, Traceroute UNIX manual page, Traceroute source available online, ftp://ftp.ee.lbl.gov/traceroute.tar.gz

[29] V. Paxson, “Measurements and Analysis of End-to-End Internet Dynamics”, PhD thesis, Computer Science Division, University of California, Berkeley, April 1997

[30] J. Mogul, S. Deering, “Path MTU Discovery”, RFC 1191, November 1990

[31] K. Lahey, “TCP Problems with Path MTU Discovery”, RFC2933, September 2000

[32] J. Postel. “Internet Control Message Protocol”, RFC 792, SRI Network Information Center, September, 1981.

[33] S. M. Bellovin, “A Best-Case Network Performance Model”, February 1992, AT&T Bell Laboratories

[34] Van Jacobson, “Pathchar: a Tool to Infer Characteristics of Internet Paths”, 1997.

[35] http://allendowney.com/research/clink/

[36] http://www.kitchenlab.org/www/bmah/Software/pchar/

[37] R. Carter, M. Crovella, “Measuring Bottleneck Link Speed in Packet-Switched Networks”, Technical Report, Boston University.

[38] Kevin Lai and Mary Baker, “Measuring Link Bandwidths Using a Deterministic Model of Packet Delay”, SIGGCOMM 2000, http://www.sigcomm.org/sigcomm2000/conf/paper/sigcomm2000-8-3.ps.gz.

[39] C. Dovrolis, P. Ramanathan, D. Moore, “What do packet dispersion techniques measure?”, Proceedings of INFOCOM 2001, 20th Annual Joint Conference of the IEEE Computer and Communications Societies, IEEE, vol.2, no., pp.905-914 vol.2, 2001 .

[40] Stefan Saroiu, P. Krishna Gummadi, Steven D. Gribble, “SProbe: A Fast Technique for Measuring Bottleneck Bandwidth in Uncooperative Environments”, Proceedings of INFOCOM 2002, June 2002.

[41] Bob Melander, Mats Björkman, Per Gunningberg, “A new end-to-end probing and analysis method for estimating bandwidth bottlenecks”, Global Telecommunications Conference, 2000. GLOBECOM '00. IEEE, vol.1, pp. 415-420, 2000.

[42] M. Jain, C. Dovrolis, “Pathload: A measurement tool for end-to-end available bandwidth”, Proceedings of Passive and Active Measurements (PAM) Workshop, March 2002.

[43] V. Ribeiro, R. Riedi, R. Baraniuk, J. Navratil, and L. Cottrell, "pathChirp: Efficient Available Bandwidth Estimation for Network Paths", Proceedings of Passive and Active Measurements (PAM) workshop, April 2003.

[44] Federico Montesino-Pouzols,”Comparative Analysis of Active Bandwidth Estimation Tools”, IPS-MoMe 2004, http://www.pam2004.org/papers/200.pdf.

[45] V.J. Ribeiro, R.H. Riedi and R.G. Baraniuk, “Spatio-temporal available bandwidth estimation with STAB”, SIGMETRICS Perform. Eval. Rev. 32, 1 (Jun. 2004), 394-395.

[46] IPerf network measurement tool homepage, http://dast.nlanr.net/Projects/Iperf/.

[47] Mark Allman, “Measuring end-to-end bulk transfer capacity”, Proceedings of the ACM SIGCOMM Internet Measurement Workshop, San Francisco, CA, November 2001.

[48] Stanislav Shalunov et al., “One Way Active Measurement Protocol”, RFC 4656, September 2006.

[49] Internet2 group, An implementation of OWAMP, http://e2epi.internet2.edu/owamp/

[50] Hélder Veiga et al., A Java implementation of OWAMP, http://www.av.it.pt/jowamp/index.htm

[51] Hélder Veiga et al., “Active traffic monitoring for heterogeneous environments”, 4th International Conference on Networking, ICN'05, April 17-21, 2005 – Reunion Island, http://www.av.it.pt/jowamp/index_files/ICN05_J-WAMP_paper.pdf

[52] K. Hedayat et al., “A Two-way Active Measurement Protocol (TWAMP)”, Internet draft, March 2007

[53] Brix Networks, http://www.brixnet.com

[54] [Hal 03] J. Hall. Multi-layer network monitoring and analysis. Technical report 571, University of Cambridge, Computer Laboratory, Cambridge, United Kingdom, July 2003

[55] [Peu 02] M. Peuhkuri. Internet Tra-c Measurements Aims, Methodology, and Discoveries, 2002

[56] J. Postel. RFC 791: Internet Protocol, September 1981.

[57] J. Postel. RFC 768: User datagram protocol, August 1980.

[58] IANA. IANA (Internet Number Assignment Authority) http://www.iana.orgassignmentsport-numbers, 2006

[59] IANA. IANA (Internet Number Assignment Authority) http://www.iana.orgassignmentsprotocol-numbers, April 2008

[60] R. Jain and S. Routhier. Packet Trains-Measurements and a New Model for Computer Network Trace.

[61] J. Quittek, T. Zseby, G. Carle, and S. Zander. Tra-c Flow Measurements within IP Networks: Requirements, Technologies, and Standardization. In SAINT-W '02: Proceedings of the 2002 Symposium on Applications and the Internet (SAINT) Workshops, page 97, Washington, DC, USA, 2002. IEEE Computer Society.

[62] I. Juva. Tra-c Matrix Estimation in the Internet: Measurement Analysis, Estimation Methods and Applications. Doctoral dissertation, Apr. 2008.

[63] John C. Eidson. Measurement, Control, and Communication Using IEEE 1588 (Advances in Industrial Control). Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[64] J. Nieminen. “Synchronization of Next Generation Wireless Communication Systems”. Master's thesis, Helsinki University of Technology TKK, October 2007

[65] D. L. Mills. RFC 958: Network time Protocol (NTP).

[66] A. Gröhn. Clock Synchronisation of a Computer Test Network (Testiverkon kellosynkronointi). Master's thesis, Networking Laboratory, Helsinki University of Technology TKK, October 2004.

[67] T. Koskiahde, J. Kujala, and T. Norolampi. A Sensor Network Architecture for Military and Crisis Management. 2008 International IEEE Symposium on Precision Clock Synchronization for Measurement, Control and Communication. September 22-26, 2008, University of Michigan, Ann Arbor, Michigan, USA, 2008.

[68] M. Ilvesmäki. Lecture slides for the S-38.3183 Internet trace measurements and measurement analysis: Introduction and basics of Internet measurements course. http://www.netlab.hut/opetus.net/s383183/k07/lectures/intro.pdf, 2007.

[69] J. Walz and B. Levine. A practical multicast monitoring scheme.

[70] E. Al-Shaer and Y. Tang. MRMON: remote multicast monitoring. Network Operations and Management Symposium, 2004. NOMS 2004. IEEE/IFIP, 2004.

[71] C. Dovrolis, P. Ramanathan, D. Moore, “What do packet dispersion techniques measure?”, Proceedings of INFOCOM 2001, Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies, IEEE , vol.2, no., pp.905-914 vol.2, 2001

[72] Stefan Saroiu, P. Krishna Gummadi, Steven D. Gribble, “SProbe: A Fast Technique for Measuring Bottleneck Bandwidth in Uncooperative Environments”, Proceedings of INFOCOM 2002, June 2002