A Survey on Network Measurement: Concepts, Techniques, and Tools

Kim Ervasti
Department of Computer Science
University of Helsinki
Email: kim.ervasti@gmail.com

***Abstract*—Communication networks are constantly growing in both size and complexity. The traffic demands are increasing while new network applications emerge on daily basis. Network measurements are the only way to gain understanding on how the networks are working and whether network resources are running low. This paper surveys the field of network measurement. The aim of the paper is to offer an all-around understanding over the different aspects of the subject: what network measurement means, what and why do we need to measure and how do we perform the measurements.**

I. Introduction

Within the last two decades, the user base of Internet has rapidly grown [1]. The same has happened to enterprise intra networks as companies are shifting toward the idea of paperless offices. As a consequence, the amount of network traffic has massively increased [1]. To keep up with the ever increasing traffic demands, networks are constantly growing in both size and complexity [1]. Internet backbones are continuously being upgraded to provide greater capacity and new network technologies are being utilized. The network traffic is diverse in nature and the behavior continuously changing [2]. New network services and applications are frequently emerging. For example, Internet's Web traffic grew from zero to 80% of all traffic during 1995-2000 [2]. Currently Web traffic's proportion is decreasing as file transfer and streaming service traffic are taking over.

Measuring networks is the only effective way to qualify and quantify how networks are being used and how networks are behaving [3]. Knowing the network behavior is critical to diagnose network problems and performance issues [4]. In addition, the behavior gives useful insights for developing future network applications and services.

However, neither the Internet architecture nor the protocols used have been designed with measurements as a priority [4], [2]. Performing network measurements to achieve accurate results is a difficult task. Simple measurement tools tend to produce limited and inaccurate results [4]. No single measurement technique alone produces accurate understanding of networks state, performance and operation [3]. To produce accurate results, complex tools and environments specialized in measuring are needed [4]. In addition, attention must be paid on solutions to minimize the effects of measurements to the network: the additional traffic to the network, increased load to the hardware and applications, and the extra maintenance costs for measurement environments [1].

This paper surveys the art of network measurement. The rest of the paper is organized as follows: section II introduces the measurement metrics that are important for understanding the network behavior. A background and a purpose is presented for each metric. Section III introduces the basic principles on how network measurements can be executed. Section IV introduces common ways to classify measurement methods. Section V introduces a few more definitions to complete the background knowledge on the subject. Section VI introduces existing tools, platforms and techniques to measure each of the introduced metrics. Section VII presents future work in the field. Section VIII concludes the survey.

II. Measurement metrics

To determine the performance of the network, multiple different measurement metrics are to be measured. The most typical metrics are connectivity, latency, packet loss rate, bandwidth and throughput [5]. In addition, traffic behavior analyses can be made. These metrics are shortly introduced as follows.

Connectivity, also known as reachability, is the most basic measurement metric [6]. Connectivity determines whether two hosts can establish a connection between each other through the network.

In networking, **latency** is the time it takes for a packet to arrive from the source host to the destination host. Latency is also referred as network delay [6]. The total end-to-end network delay is actually a product of several different types of delays [6]: processing delay, transmission delay, propagation delay and queuing delay. Processing delay [6] is the total time it takes for all the routers in the network path to process the packet. In IP network, the processing includes decrementing packet's Time to Live variable by one, check that the IP header checksum matches the header and to decide where the packet is forwarded. Transmission delay [6] is the time it takes to send the packet through the link. The time is affected by the bit rate of the link. Propagation delay is the time it takes for the physical signal to travel over the transmission medium [6]. Queuing delay is the time the packet spends in buffers of the routers in the network path [6].

Latency can be measured either as one-way or two-day delay [6]. Two-way delay, also known as round-trip time (RTT), is easier to measure as only a single host needs to measure the time between the initial packet and the response.

To measure one-way delay, both the source and the destination hosts have to cooperate and be synchronized [2]. One-way delay is an important affecting factor for performance in many applications [2].

Packet loss rate is the rate at which packets are being lost in their transit from the source to the host. Packet being lost means that the packet does not arrive to the intended destination [6]. To keep the network application usable, there is often some kind of timeout mechanism for lost packets to trigger retransmission. If the retransmission timeout is reached before the packet arrives, the packet is regarded as lost even if it eventually would arrive to the destination [6]. Packet loss is an important metric, because it may have a dramatic impact on the performance of an application [4].

The packet loss rate is often very different for the same path in opposite directions [4]. With simple echo-reply measurement tools like Ping, it is not possible to define whether the original echo packet was lost or the response. To measure forward loss (source to destination) and reverse loss (destination to source) separately, more advanced tools are needed. Loss asymmetry is important because many protocols have a different level of fault tolerance for forward and reverse directions. For example, TCP is far more tolerant for losing an acknowledgment packet than a data packet [4].

Bandwidth is a two-fold term [7]: depending on the context, the term is used to describe either the physical link capacity in terms of signaling or the maximum actual data rate of a specific network link or a path can transfer. To the data rate in a link or a path, there are three measurable metrics associated: total **capacity**, **available bandwidth** and **bulk transfer capacity** (BTC) [7]. Total capacity refers to the maximum possible bandwidth the link or a path can transfer. Available bandwidth refers to maximum unused bandwidth of a link or path during a certain time period. Bulk transfer capacity refers to achievable throughput of an established TCP connection.

Bandwidth is an important factor for network performance, because for many data-intensive applications, like file transfers and streaming services, bandwidth directly defines the application performance [7]. High bandwidth is also often related to low latency [7] in such, that packets spend less time in buffers of bottleneck links. All three listed metrics are important. The relevance depends on the application.

Throughput is a measure for amount of data actually being transferred across a link or network at a certain time [6]. Throughput is defined as bits per second or bytes per second.

Traffic is not exactly a metric, but something that can be collected from the network and analyzed. Traffic analysis is used to determine the composition of the current traffic in the network [8]. Information is useful for capacity planning, traffic accounting and network security. In addition to traffic behavioral analyses, several other measurement metrics can be determined by the analysis: for example packet loss, available bandwidth and throughput [6].

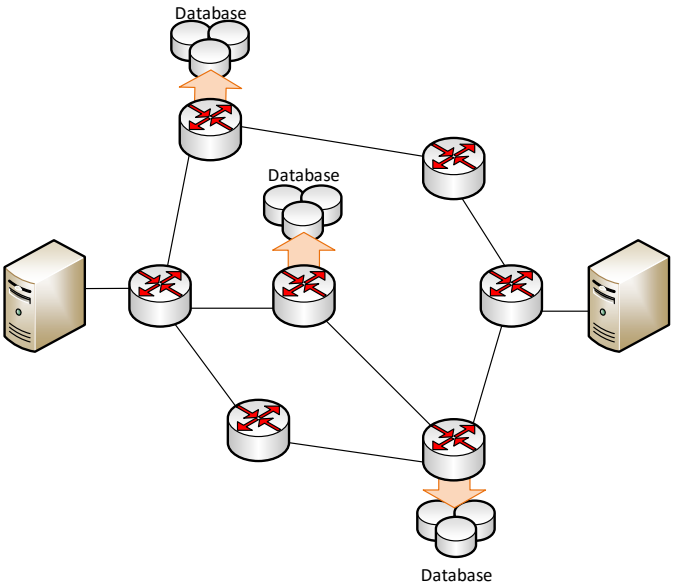

Fig. 1. An example of passive measurements

III. Basic concepts of network measurement

This section introduces the different approaches to measure the network. Network measurement methods are classified as being a passive measurement method or an active measurement method [9]. A measurement can also be a combination of these two, a hybrid measurement.

A. *Passive measurements*

In passive network measurements, network traffic is being collected and analyzed [5]. In passive measurements, no new traffic is introduced or generated to the network [1], [5]. As a result, measurements have no impact on network performance and therefore measurement analysis. Data collection is achieved by setting up measurement devices in key locations of the network.

Figure 1 shows an example on how passive measurements can be performed in a network. In the example, several routers are collecting network traffic to a database. In addition to only storing the traffic, real-time analysis can be made.

One fundamental challenge in passive measurements is the amount of data collected [6]. Also the equipment needed for collecting and analyzing the data can be very expensive, because they have to be able to process and save all the collected data [1]. High performance equipment is needed with particular focus on the speed of main memory [8]. Therefore, it is essential for passive measurements to minimize both the amount of measuring devices and the amount of data collected while maintaining the accuracy of measurements adequate [1].

The amount of collected and stored data can be decreased in several ways [6]. When network packets are collected, data that is irrelevant for network measuring can be removed. For example by removing packet payloads, in other words by only saving the packet headers, the amount of saved data is rapidly decreased. Be removing the payload, also privacy issues

related to collecting network traffic are mitigated. Regular compression methods, for example gzip, can also be applied to remove redundant information from collected data [6].

One way to decrease the amount of collected traffic data is simply to leave some packets out. For a traffic analysis to be adequately accurate, not all packets are needed. Merely a part of the traffic is enough to acquire the big picture [1]. Traffic data can be selected for collection by filtering, classification and sampling [1].

In **filtering** packets are collected based on a specific property of the packet [1]. For example, only packets with a certain protocol, or destination port number can be selected. In **classification** packets can be classified to classes based on a certain property [1]. Only class statistics are calculated and stored.

In **sampling** packets are collected statistically [9]. Not all data is collected, but more or less random based packet selection is made [1]. There are several techniques for traffic sampling, often divided into two categories [9]: typical sampling techniques and specific algorithm based techniques. Typical sampling techniques contain systematic, random, stratified and probabilistic sampling [8], [1]. Systematic sampling selects packets or objects from the data flow every N intervals [8]. N can be either a time interval (time-based sampling) [1] or the amount of packets or other collectable data objects (regular sampling). Random sampling selects each packet or data object independently from each other and with certain random rules [8]. If the random rule is simply the probability of 1/N, then the technique is also called probabilistic sampling [1]. Stratified sampling divides the data flow into several groups or types, and then selects a single sample from every group with a same probability [8]. Algorithm based techniques are more complex but are specialized in traffic sampling.

The fundamental problem in sampling is to minimize the amount of collected data and to collect a representative subset of the relevant packets [10]. There is no single best and most useful statistical sampling method [9]. The choice of the sampling method depends on the application and on the amount of and type of data collected. The selection of the best or even a good sampling method for the use case is a difficult task [6], [9]. Another downside of sampling is that flow analyses gets more difficult or even impossible in some cases [6]. When sampling the traffic, packets belonging to a certain flow of interest might get sampled out in the data collection phase and results as black holes in the knowledge about that particular flow.

Passive measurements are a good choice when measurement locations can be freely selected [6]. For this, having an administrative access and control over the whole network is a requirement. In general, passive measurements are way more complex than active measurements [5]. Passive measurement can only measure and analyze a part of the network traffic. Therefore the accuracy of measurements depends highly on the performance of the measurement probes and on the choices on which statistical and analytical methods are deployed [5].

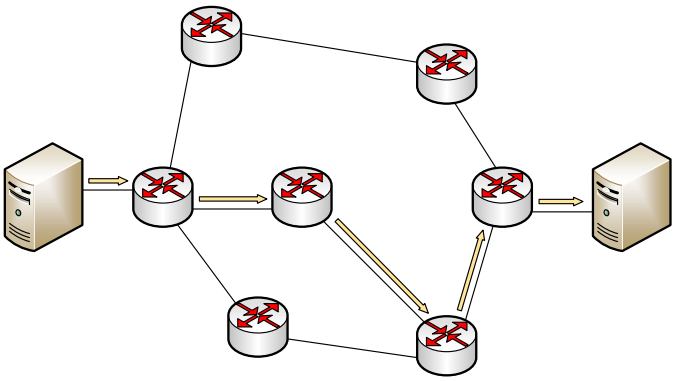

Fig. 2. An example of an active measurement

B. Active measurements

Active measurements are performed by pushing probe packets into the network [5], [6]. Probe packets are specifically designed for the measurement task at hand. The flow of packets is then used to analyze network performance. For example, throughput of the packet flow can be used to estimate network bandwidth. Delay of the packets can be used to analyze end-to-end latency of the network.

Figure 2 illustrates an example of how an active measurement can be performed in a network. In the figure, an end-to-end type measurement is being executed between two servers.

There are two fundamental downsides to active measurements. First, the probe packets used for measuring the network, also introduce additional traffic to the network [5]. Second, as probe packets consume network resources, the measurement process itself might affect the measurement results by degrading the performance of the network [5].

On the other hand, active measurements do not require large amounts of disk space to collect network packet data [6] as only the results need to be stored. By not collecting network packet data, network traffic related privacy and information security concerns are greatly reduced [6]. As passive measurements depends on existing traffic in the network, active measurement might be the only mean to measure a specific link or a path between two specific hosts if there happens to be no existing spontaneous traffic in that link or path.

One of the key objectives in active measurements is to find the minimum number of probe packets that are able to measure all links in the network [1]. To measure all the network links efficiently, also the selection of the best probe host candidates, that is hosts that function as beacons for the packets, is important part of the problem. The problem has been found NP-hard [1].

C. Hybrid measurements

Hybrid measurements combine the active and the passive measurement methods [6]. Typically, in hybrid measurements probe packets are sent to the network and their progress in the network is then monitored with passive traffic collection. Hybrid measurements enable to track the path of the packets

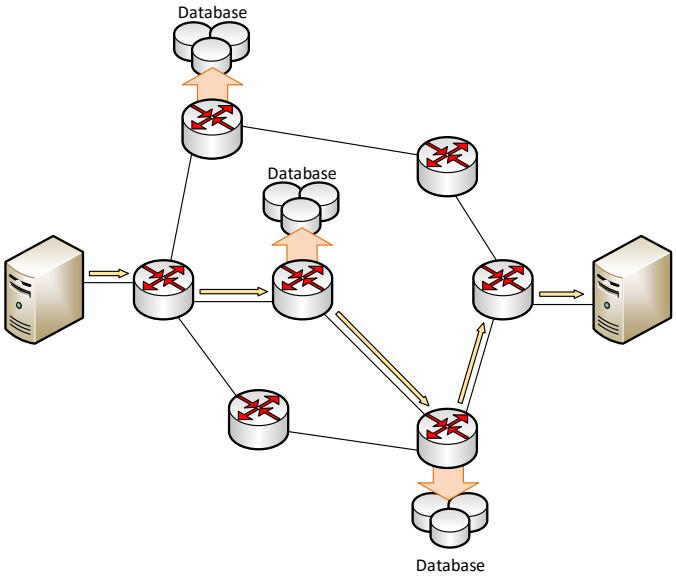

Fig. 3. An example of hybrid measurements

and measure both end-to-end and intermediate link delays. Hybrid measurements combines both the capabilities and the downsides of the active and passive methods. The downsides are increased traffic in the network and the costs of complex data collecting infrastructure.

Figure 3 illustrates an example on how passive and active measurements can be performed simultaneously in the network. In the example, all network traffic is being collected and analyzed at all times using several routers to gain understanding on how the network is being used. In addition, a specific path performance is measured when needed with an active end-to-end type measurement. For example, an active measurement can be used to test the network connectivity if there is no spontaneous traffic in the network. This way, it can be assured that the network is not silent because of network dysfunction.

IV. Measurement classifications

All measurements are classified as being an active, a passive or a hybrid measurement, but measurements can also be classified in several other ways based on other charasteristics and criterion [2]. This section introduces other common classifications.

A. Edge and interior measurements

Measurement methods can be classified to edge and interior measurements according to where the measurement devices are located [5]. The edge measurements are executed by edge hosts of the network. Edge measurements cover end-to-end performance of the network services. Interior network measurements are executed in network routers. Interior measurements are passive in nature. Interior measurements are used to analyze traffic flows and commonly used routes in the network.

B. Cooperative and non-cooperative measurements

Measurements are classified as cooperative or non-cooperative measurements based on whether the network devices cooperate with the measurements or not [5]. In cooperative measurements commonly routers participate in the measurement by enabling route analysis and detailed network segment analysis [5]. In non-cooperative measurements, measurements are performed in end-to-end fashion. Even the destination node might node cooperate with the measurement by having no active measurement application enabled, but merely responding with standard ICMP messages for example.

C. Single point and multipoint measurements

Measurements can be classified as single point or multipoint measurements according to the amount of devices performing the measurement [2], [5]. In a large network, more measurement devices are needed to gather more detailed information about the network. Larger the network is, the less we can analyze the network in detail by using only end-to-end measurements.

D. Other characteristics

A little less official but common ways to characterize network measurement techniques are the following four divisions [2]: 1) a passive measurement can collect and analyze either a particular traffic flow or general behavior of all traffic. 2) The measurement can be continuous or performed on demand. 3) A measurement can be direct or indirect, the latter meaning that only end-to-end measurements are made and the interior network conditions are infered by analyzing the results. 4) The measurement can be unidirectional or bidirectional, meaning that either the behavior of one direction of the network path or behavior of both forward and reverse paths are measured.

V. Additional measurement definitions

To give more complete understanding on the field of network measurement, a few more definitions need a deeper introduction.

A. Network Tomography

Network tomography is a study, where the interior network performance is measured indirectly, by estimating measurement data collected by edge hosts of the network [11]. Network tomography uses the same principles as tomography image reconstruction in imaging [11]: an object's resonance is measured from the surface from multiple measurement points, and the interior structure is analyzed by combining the data. Network tomography can include either active or passive measurement [11]. To produce the estimate in active way, several end-to-end measurements are executed in the network. For example, the measurement metric can be packet loss ratio or traffic throughput from end-to-end. Each end-to-end measurement resemble a slice of the network, similarly to tomography imaging. Measurement results are used to create an *origin-destination traffic matrix* [2]. In origin-destination

traffic matrix each end-to-end measurement is an origin-destination pair and presented as a matrix element. In passive measurement, slices are measured by collecting existing traffic data. After the matrix is created, a statistical algorithm is used to analyze and infer the interior links by combining the slice information in the matrix.

Because the network tomography uses edge measurement data to estimate all interior link capacities, the method substantially reduces the amount of probe packets needed in active measurement or reduces the amount of measurement probes needed for collecting data in passive mode [5]. Network tomography is particularly useful in situations where there is no administrative access of to the interior network and the interior network is non-cooperative. The downsides of the method are high computing complexity of the analysis and the inaccuracy of the estimates [5].

B. Measurement probe

A measurement probe is a network device that performs network measurements or collects measurement data [5]. Probes are able to measure the network in both active and passive fashion. A probe that is measuring the network actively by sending probe packets, is often also referred as a beacon [1]. Probes are controlled and monitored with a centralized network monitoring system. The control connection is typically created as an encrypted TCP connection [5]. Network firewalls or tunnels are configured accordingly, so that the network monitoring system is able control the probes and collected measurement data from the probes located in multiple different networks. Transmissions of measurement data from the probes to the centralized monitoring system may in some cases consume considerable network bandwidth [11].

VI. Measurement tools, platforms and techniques

Countless tools, platforms and techniques have been introduced under the theme of network measurement. This section introduces some of the most known and cited ones. Each measurement metric introduced in section II is covered here by one or more concrete example on how to measure them in practice.

Ping and **Traceroute** are probably the most well known network measurement tools [6]. They are both ICMP based, and operate with active measurements.

Ping measures *connectivity*, *round-trip time* and *packet loss rate* between two hosts [6], [2]. Ping works by sending an ICMP echo request to the target host. The target host then replies with ICMP echo reply. The host executing the measurement often sends out several ICMP echo request packets one by one. If one or more replies are received, there is at least partial connectivity. A round-trip time can be calculated from the time request was sent to the time when reply was received. Packet loss rate is estimated by dividing the amount of lost replies with the total amount of requests sent. Ping is very simple, yet useful tool that can be found in almost every operating system today [6].

TraceRoute resolves the *path* between two hosts and returns a single sample round-trip time of each hop of the route [6]. Traceroute does this by sending ICMP or UDP packets to destination host of the path. Time to Live (TTL) value in the IP header is increased for every packet. The first packet has TTL of 1 and value is increased until a packet reaches the destination host. All routers forwarding the packet will decrease this value by one. If the value reaches zero, the packet is dropped and ICMP TTL Expired message is sent back to the sender host. As the measurement progresses, one by one routers within the path will drop the packets and send back ICMP TTL Expired messages to the source host. This way, for each router an IP address can be discovered and a round-trip time be measured.

Both Ping and Traceroute have a common downside: ICMP filtering [6], [4]. Some Internet Service Providers and various NAT devices tend to filter out ICMP messages for security reasons. For example, Internet worms search for potential targets by using Ping. For the same reason, many Internet hosts ignore ICMP messages to hide their presence. In Internet routers, incoming ICMP messages are often totally ignored and the routers might be configured to not send out any ICMP replies. In addition to ICMP filtering problems, link level devices like switches, are not discoverable with ICMP messaging. Also, when measuring packet loss rate, Ping merely measures total packet loss ratio and ignores the important factor of loss asymmetry [4]. Measuring one-way delay requires more advanced approach.

Sting [4] is a measurement tool specialized in measuring *one-way packet loss rates* on both directions. Sting utilizes TCP for the measurement and TCP error control to gain understanding whether packets are dropped in forward or backward direction.

Sting measures forward loss by first sending out a set of TCP data packets to the receiver host. In the second phase, Sting sends one more packet and observes whether the receiver acknowledges the packet with sequence number one higher than the last data packet. From this information, Sting can conclude that all packets have arrived to the receiver. If the acknowledgement has a sequence number that is lower than than the last data packet, the packet with the sequence number acknowledged can be declared as lost. The lost packet is then retransferred. Then the sequence number of the new acknowledgement message is again observed. This is continued until all sequences have been acknowledged. The amount of lost segments is used to calculate the forward packet loss rate.

Figure 4 shows a partial example scenario of TCP segments transferred in a Sting forward packet loss measurement. In the first phase, Sting sends eight segments of which two get lost in the transit. In the second phase Sting sends one more segment (segment 9) and waits for the acknowledgement. The receiver acknowledges segment 4 from which Sting knows that segment 4 has been lost. The segment is then retransmitted and another acknowledgement is then observed. After resending the segment 6, everything will be acknowledged up to this point and the measurement ends concluding two segments out

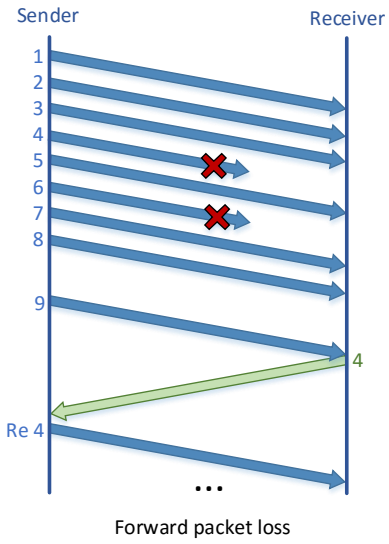

Fig. 4. An example scenario of TCP segments transferred in Sting

of eight was lost.

Measuring the reverse loss rate is more problematic [4]. The problem is that TCP does not acknowledge every packet but uses the *delayed acknowledgement* technique that skips unnecessary acknowledgements to save bandwidth. A way to to solve the problem is to force acknowledgement for every packet by waiting long enough after sending each data packet, until the receiver acknowledges it [4]. The downside of the solution is a long measurement runtime. Another method to force the receiver to acknowledge every packet is to exploit the *fast retransmit* technique that modern TCP implementations use [4]. Fast retransmit algorithm specifies that if the receiver immediately acknowledges the last orderly received segment, when an out of order segment is received. The technique is used speed up the retransmission of possibly lost segments. In this case, the sender sends all data packets out of order, to receive equal amount of acknowledgement messages from the receiver.

OWAMP (One-way Active Measurement Protocol) and **TWAMP** (Two-way Active Measurement Protocol) are measurement protocols for measuring *network delay* reliably and in high-precision [6]. OWAMP is specialized in measuring one-way delay between two hosts. A client and a server are set up on the two hosts. When measurement starts, the client sends UDP probe packets to the server. Every packet contains the timestamp when the packet was sent. The server then compares the timestamp to the receiving time on the server and calculates the delay. The clocks of the hosts obviously need to be synchronized. To finish the measurement, the server returns the delay measurement results to the client.

TWAMP is similar to OWAMP in both methodology and architecture, but adds round-trip measurement capability to the arsenal [6]. To achieve this, both hosts act as clients and servers at the same time. TWAMP has much better measurement accuracy for round-trip time than Ping. In addition, as TWAMP uses UDP, TWAMP lacks all the ICMP originated problems that Ping has.

For measuring *bandwidth* there are four major techniques [7], [6]: **VPS** (Variable packet size), **PPTD** (Packet Pair/Train Dispersion), **SLoPS** (Self-Loading Periodic Streams), **TTOP** (Trains of Packet Pairs).

VPS [7], [6] technique estimates the *capacity* of individual link hops along a network path [7], [6]. In VPS, source node sends different sized packets to all routers along the network path. Round-trip time is then measured for each hop as a function of the packet size. The hop capacity can be estimated from the ratio of change in packet size compared to measured round-trip times.

PPTD [7], [6] technique estimates the end-to-end *capacity* of a path [7], [6]. PPTD estimates the capacity by sending multiple packet pairs to the receiver. The two packets in a pair are sent back-to-back. Size of the packets is a constant. At the receiver, the dispersion introduced to the packet pairs within the network path is measured. The bottleneck link of the path increases the dispersion for the packet pairs. Unfortunately, the technique assumes that at the time of the measurement, the additional network traffic load on the path is very close to zero [7]. Using many packet pairs in the measurement will decrease the measurement error and increase the chance of receiving at least one packet pair without other traffic interfering with the transmission.

SLoPS [12] and TTOP [13] techniques estimates *available bandwidth* of a path [7], [6]. In SLoPS, a measurement probe sends equal sized probing packets at a certain rate to the destination host. Packets are sent as short streams. The destination node measures one-way delay of the packets. After the test, destination host reports the measured delays back to the sender. The sender tests different rates with iterative search similar to binary search. The sender host keeps a silent period between each test stream to decrease the intrusiveness of the measurement. To estimate the available bandwidth, the variation of one-way delay compared to sending rate is monitored. An increase in delay indicates that there is a congestion in the bottleneck link of the path. In other words, that the available bandwidth has been utilized. TOPP [13] is similar to SLoPS with the difference that in addition to available bandwidth, TOPP also estimates the capacity of the bottleneck link. TOPP does this by combining the ideas in SLoPS to PPTD. In TOPP, measurement probe host sends multiple packet pairs to the destination host. Rate is increased linearly until one-way delay starts to increase. In addition to one-way delay, the destination host monitors the dispersion between the packet pairs.

All four techniques are active measurement methods and as such are intrusive in nature [7]. However, all four techniques consumes network bandwidth sparingly in the measurement process [7]. It is estimated that typically less than 10 percent of the available bandwidth is consumed in the measurements by these techniques [7].

As a general drawback, these techniques assume that the measured path remains static and the amount of traffic in

```
host:/home/user/iperf-3.1.3# src/iperf3 -c 192.168.1.113 -p 2048
Connecting to host 192.168.1.113, port 2048

[  4] local 192.168.1.2 port 57564 connected to 192.168.1.113 port 2048
[ ID] Interval           Transfer     Bandwidth       Retr  Cwnd
[  4]   0.00-1.00   sec  67.5 MBytes   566 Mbits/sec    0   62.2 KBytes
[  4]   1.00-2.00   sec  67.9 MBytes   570 Mbits/sec    0   80.6 KBytes
[  4]   2.00-3.00   sec  68.3 MBytes   573 Mbits/sec    0   84.8 KBytes
[  4]   3.00-4.00   sec  72.6 MBytes   609 Mbits/sec    0   90.5 KBytes
[  4]   4.00-5.00   sec  72.6 MBytes   609 Mbits/sec    0   90.5 KBytes
[  4]   5.00-6.00   sec  72.6 MBytes   609 Mbits/sec    0    106 KBytes
[  4]   6.00-7.00   sec  72.5 MBytes   608 Mbits/sec    0    106 KBytes
[  4]   7.00-8.00   sec  72.6 MBytes   609 Mbits/sec    0    106 KBytes
[  4]   8.00-9.00   sec  72.3 MBytes   606 Mbits/sec    0    106 KBytes
[  4]   9.00-10.00  sec  72.3 MBytes   606 Mbits/sec    0    106 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Retr
[  4]   0.00-10.00  sec   711 MBytes   597 Mbits/sec    0      sender
[  4]   0.00-10.00  sec   711 MBytes   597 Mbits/sec           receiver
```

Fig. 5. An example run of Iperf

the path stationary during the measurement. Dynamic routing changes or traffic fluctuation may cause problems on the measurements or bias the measurement results [2].

Iperf and **NetPerf** are simple tools to measure *bulk transfer capacity* between two hosts [7]. Both tools uses similar active measurement methods. A client and a server are set up on the two hosts. Access to both hosts is required, so methods are considered cooperative. However, superuser privileges are not required. The measurement is then performed by establishing multiple parallel TCP connections between the client and the server, and random data is transferred between the hosts as fast as possible. Several samples of maximum achieved throughputs are measured. TCP implementation of the underlying operating system is used [6]. **Cap** is a similar tool to Iperf and NetPerf, with the difference that it actually uses UDP packets to emulate TCP. Both data and acknowledgement packets are emulated by sending similar kinds of UDP packets between the two hosts.

Figure 5 illustrates an example run of Iperf. In the figure, the client has been connected to remote server on a local network. TCP throughput has been measured in one second interval for total of ten seconds. Achieved throughput is listed for each interval separately and as average for the whole measurement.

With Iperf, NetPerf and Cap, all available bandwidth is consumed during the measurement. In this sense, they are more intrusive than VPS and PPTD for example. However, as all these three tools either use TCP or emulate it, *TCP congestion control* ensures that they react to congestion and therefore other possible network applications are not *locked out*. The tools are considered more intrusive in terms of bandwidth usage, but more congestion friendly than VPS and PPTD [7].

NIMI (The National Internet Measurement Infrastructure) [14] is a large-scale multipoint measurement [5] infrastructure. It consists of set of servers running on different hosts [15]. These hosts are called platforms or probes. Scalability is the key design of NIMI, as potentially thousands of probe hosts can reside within a single measurement infrastructure. The probes within a single infrastructure can be administered by different parties. There often exists multiple contact points where probes report the measurements. Also the contact points can be administered by different parties.

NIMI can perform diverse types of active measurements [15]. Basically any measurement tool, in a form of a binary or a script, can be integrated to NIMI as a measurement module. NIMI deploys Traceroute, a File Transfer Protocol wrapper, Cap and several other common measurement tools out of the box [15].

Cisco NetFlow is a *traffic* measurement software [10], [11], [9], [2] supported by Cisco Routers. Netflow uses passive measurement method to collect, sample and analyze network traffic. Netflow is so widely used among network operators and access providers that it became the industry standard on traffic measurement in the beginning of the century [2], [9]. Netflow is a still hot topic and the basis of the new universal standard **IPFIX** (IP Flow Information Export) [10]. IPFIX is a protocol for exporting flow data from the routers. IPFIX is defined by the IETF (Internet Engineering Task Force) and an Internet Standard since 2013.

Netflow uses systematic sampling to sample network traffic [2]. Traffic is then aggregated to traffic flows. Flows are then analyzed. Analyzes can be used for example to detect security problems, to profile traffic or to construct origin-destination traffic matrix for *network tomography* [2], [11].

Netflow has its drawbacks. The systematic sampling method that Netflow uses is prone to both sampling bias and performance problems [10], [9]. Bias can be introduced to the data if the traffic is periodical. Periodically transferred packets might systematically be left out for selection. This is why even random sampling is generally preferred over systematic sampling [10].

In Netflow, the systematic sampling rate is configured manually [9]. This might lead to performance problems. Finding a good value is difficult. A rate too low saves load on the measurement system, but increases the measurement error. Too high rate value increases the measurement accuracy but might overload the measurement system. In addition, the load on the system is varying in proportion to the traffic load in the network. As an result an unexpectedly high network traffic peak might overload the system so severely, that the normal routing operations of the routers might be affected or jeopardized [9].

Netflow is also suffering from capacity issues concerning non-TCP flows [2]. Netflow is under constant development [10], so these issues might be addressed in the future versions of Netflow.

VII. Future work

As Internet is on the verge of a new era, Internet of Things, the expansion of Internet is expected to continue even faster than before. The huge IPv6 address space allows the number of network devices to grow in massive multitudes but it is unclear how networks can take the increased traffic demands. In addition to the amount of traffic, new type of traffic with different demands will be introduced to the networks.

Network measurement will keep its status as crucial part of network development. Future studies on new ways to sample the increasing amount of traffic and new ways to decrease the intrusiveness of active measurement methods are essential. Improved support for network measurement in future Internet protocols is demanded.

VIII. Conclusions

This paper surveyed the field of network measurement. The goal was to give all-around picture on the subject. The paper can be summarized in seven steps: first, the need for network measurement was justified. Second, the measurement metrics were introduced. Third, basic concepts of network measurement were surveyed. Fourth, different ways to classify network measurement techniques were reviewed. Fifth, additional definitions were introduced to complete the understanding of the field. Sixth, measurement tools, platforms and concrete techniques were introduced to cover each measurement metric. Seventh, future insights were presented.

References

[1] C. Chaudet, E. Fleury, I. G. Lassous, H. Rivano, and M.-E. Voge, "Optimal positioning of active and passive monitoring devices," in *Proceedings of the 2005 ACM Conference on Emerging Network Experiment and Technology*, ser. CoNEXT '05. New York, NY, USA: ACM, 2005, pp. 71–82. [Online]. Available: http://doi.acm.org.libproxy.helsinki.fi/10.1145/1095921.1095932

[2] A. Ziviani, "An overview of Internet measurements: fundamentals, techniques, and trends," *African Journal of Information and Communication Technology*, vol. 2, no. 1, 2006.

[3] R. Caceres, N. Duffield, A. Feldmann, J. D. Friedmann, A. Greenberg, R. Greer, T. Johnson, C. R. Kalmanek, B. Krishnamurthy, D. Lavelle, P. P. Mishra, J. Rexford, K. K. Ramakrishnan, F. D. True, and J. E. van der Memle, "Measurement and analysis of IP network usage and behavior," *IEEE Communications Magazine*, vol. 38, no. 5, pp. 144–151, May 2000.

[4] S. Savage, "Sting: a TCP-based network measurement tool," in *Proceedings of the 2Nd Conference on USENIX Symposium on Internet Technologies and Systems - Volume 2*, ser. USITS'99. Berkeley, CA, USA: USENIX Association, 1999, pp. 7–7. [Online]. Available: http://dl.acm.org/citation.cfm?id=1251480.1251487

[5] H. Wang, L. Song, H. Chen, and L. Yan, "Study on network measurement technologies and performance evaluation methods," in *Cross Strait Quad-Regional Radio Science and Wireless Technology Conference (CSQRWC), 2013*, July 2013, pp. 377–380.

[6] V. Mohan, Y. J. Reddy, and K. Kalpana, "Active and passive network measurements: A survey," *International Journal of Computer Science and Information Technologies*, vol. 2, no. 4, pp. 1372–1385, 2011.

[7] R. Prasad, C. Dovrolis, M. Murray, and K. Claffy, "Bandwidth estimation: metrics, measurement techniques, and tools," *IEEE Network*, vol. 17, no. 6, pp. 27–35, Nov 2003.

[8] L. Haili, Z. Ke, and H. Wanwei, "Key technologies of network traffic measurement," in *Software Engineering and Service Science (ICSESS), 2014 5th IEEE International Conference on*, June 2014, pp. 1093–1098.

[9] C. Lv, B. Ma, X. Du, B. Li, and T. Liang, "Analysis and research of network measurement technologies," in *Intelligent Computing and Internet of Things (ICIT), 2014 International Conference on*, Jan 2015, pp. 117–121.

[10] R. Hofstede, P. Čeleda, B. Trammell, I. Drago, R. Sadre, A. Sperotto, and A. Pras, "Flow monitoring explained: From packet capture to data analysis with netflow and ipfix," *IEEE Communications Surveys Tutorials*, vol. 16, no. 4, pp. 2037–2064, Fourthquarter 2014.

[11] A. Coates, A. O. H. III, R. Nowak, and B. Yu, "Internet tomography," *IEEE Signal Processing Magazine*, vol. 19, no. 3, pp. 47–65, May 2002.

[12] M. Jain and C. Dovrolis, "End-to-end available bandwidth: Measurement methodology, dynamics, and relation with tcp throughput," *IEEE/ACM Trans. Netw.*, vol. 11, no. 4, pp. 537–549, Aug. 2003. [Online]. Available: http://dx.doi.org.libproxy.helsinki.fi/10.1109/TNET.2003.815304

[13] B. Melander, M. Bjorkman, and P. Gunningberg, "A new end-to-end probing and analysis method for estimating bandwidth bottlenecks," in *Global Telecommunications Conference, 2000. GLOBECOM '00. IEEE*, vol. 1, 2000, pp. 415–420 vol.1.

[14] V. Paxson, J. Mahdavi, A. Adams, and M. Mathis, "An architecture for large scale Internet measurement," *IEEE Communications Magazine*, vol. 36, no. 8, pp. 48–54, Aug 1998.

[15] V. Paxson, A. K. Adams, and M. Mathis, "Experiences with NIMI," in *Applications and the Internet (SAINT) Workshops, 2002. Proceedings. 2002 Symposium on*, 2002, pp. 108–118.