ELEC-E7130 Internet Traffic Measurements and Analysis

Markus Peuhkuri

2018-09-11 (139d1d5af0)

Abstract

This material is **work-in-process** for a part of Internet Traffic Measurements and Analysis course at Aalto University. Feedback and comments are appreciated via email or via course forum. The bibliography is also incomplete.

Material assumes the reader to have basic knowledge on IP networks and Internet.

Contents

1 Introduction to traffic measurements

This material is **work-in-process** for a part of Internet Traffic Measurements and Analysis course at Aalto University. Feedback and comments are appreciated via email or via course forum. The bibliography is also incomplete.

Material assumes the reader to have **basic knowledge on IP networks and Internet**.

1.1 Why anyone wants to measure network

Networking would be much easier if there would infinite capacity with zero latency. Until faster-than-light communication is invented, we must live with present physical reality limitations for latency. Available bandwidth has been increasing with reasonable pace, but also usage has been increasing at the same or even faster rate. Information theoretical limits of traffic have not been exceeded, just advances in science and technology have made possible to find ways to get more capacity. In any case, increasing network capacity bears an additional cost.

Through of history, communication has been limited latency: the speed a human could move by foot or by some animal over long distances; maximum being around few hundred kilometers a day by Pony Express routes. Optical communication using fires was used in many places to send an alert on attackers. Propagation speed exceeded any other means, but information content was only one bit: we are being attacked. More advanced communication is possible using whistles and drums. These allow distances of 5 or 10 kilometres respectively in good conditions.

On late 18^{th} century first practical optical telegraph lines were constructed in

France. These provided faster communication, but data rate was only one or two words in a minute. There was no way other than physically carry letters to communicate over oceans.

In 1830s first commercial telegraphs were invented and in 1866 first reliably working cable was installed across Atlantic Ocean. Communication was not cheap – sending a minimum-length message (10 words) cost weekly salary of US congressman. Communication speed was limited mostly by operator skills.

Telephone was invented 1876 and rapidly taken in use around the world. As number of end terminals was much larger than in telegraph, telephone exchanges were deployed to reduce needed lines. Originally, each pair of lines carried one telephone call. To increase capacity, additional lines need to be installed. A major question was how many lines there should be installed. If there was too few, users were not happy because calls were blocked very often but installing too many become too expensive.

This is the key question still. Is the current network capable to transfer all needed communication or should it be upgraded. Or, if network is expanded, would there be users demand. Further more, would that be economical?

In 1917 Danish mathematician Erlang published his formula $E = \lambda h$ that created relationship between offered traffic (E), call arrival rate λ and average call-holding time h. When there was knowledge on λ and h, it was possible to estimate how many lines were needed. Quite often, parameters were derived from busy hour figures. As the processes are random, it is still possible that calls block with certain probability. Of course, it is important to find out the parameters. There are basically two ways to do it: educated guess and measuring. Measuring must also happen in similar population. For example, average call duration varies around the globe in different cultures.

Later, carrier wave made it possible to have tens of simultaneous phone calls in one coaxial cable or microwave link. Starting from 1970s, digital transmission in optical lines becomes the standard way for long-distance communication. The first Trans-Atlantic optical cable was taken in use 1988.

This course is about Internet traffic measurements. It means that we have vastly different range of "call" duration and capacity expected compared to telephone networks. The network may also have widely variable transmission capacity.

1.2 Measuring capacity

!: open web browser and open developer toolbar (Mozilla Firefox or Google Chrome: Control-Shift-I) and select *network*. Visit a site by typing address to location bar or select a bookmark.

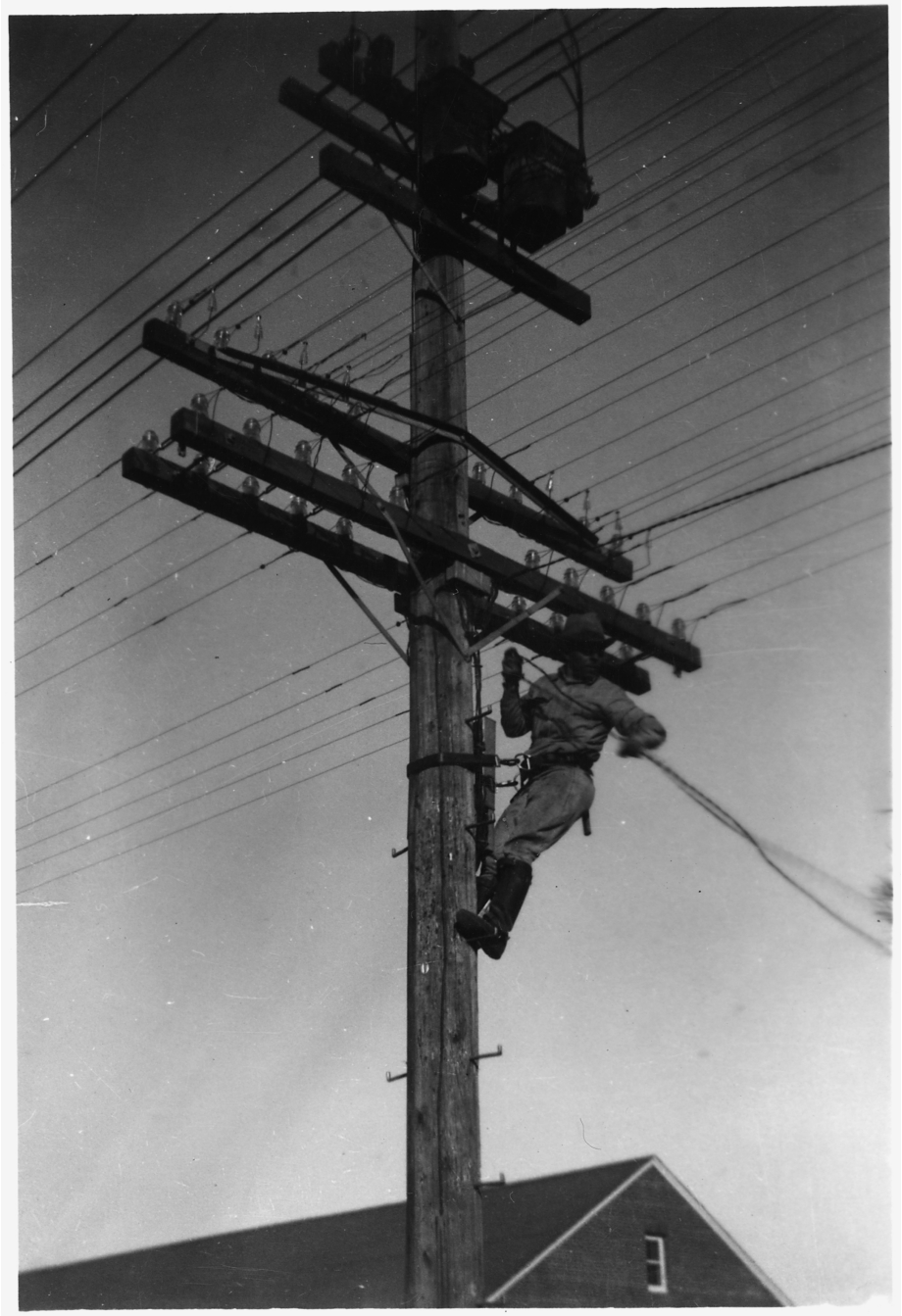

Figure 1: Installing telephone lines (Wikimedia)

1.2.1 How fast web sites load?

Google Chrome supports also emulation of different (mobile) networking technologies. Try set of sites using different parameters.

1.3 Why user should care for measurements

Network user is mostly interested to know if she is receiving what she had paid for. It is common that network operators provide different speeds towards customer with different monthly rates. Most users want to know if they are receiving the speed they have paid for.

One way to check capacity is to use some services to test network speed like http://nettitutka.fi or http://speedtest.net. When run one-off, these provide only snapshot of network performance. It is also possible that capacity towards test server is good but still some service user is interested about is having performance issues. The internet is a network of multiple networks. This means a single end-to-end connection travels over multiple ISP networks and because of **peering** issues between ISPs a site may have bad response performance from one network and good response from another.

> **?:** Have you used any network speed test to resolve or to identify network problems?

While consumers may become irritated about their network being too slow or unavailable, they may have some protection from local consumer protection laws or class-action. Again, it may be complicated to prove that *perceived quality of service* has not been the one advertised. That is even more relevant with mobile network were location and time have more effect on available speed.

> **?:** A key question is how much capacity one needs in real. What is the benefit of having 100 Mbit/s compared to 20 Mbit/s connection if one mostly watches for streaming video from Youtube or Netflix?

1.4 Corporate subscribers

Corporate subscribers, however, do not enjoy similar protection as business-to-business relations are based on contracts. Corporate users also have typically larger incentive for establishing they receive capacity they have paid for. Underperforming network may result loss of productivity, loss of sales or even direct expenses.

The corporate subscribers have thus an incentive to monitor their networks and services. In addition to check if network performance matches the agreement,

it is also important to understand how network is being used. Network traffic monitoring can be divided into following topics:

1. What is current traffic demand? Is traffic increasing or decreasing? Is *traffic matrix* changing?
2. Is there traffic that is prohibited by policy, malicious or otherwise unwanted?
3. Are our customers served with good performance?

Having capacity to match for traffic demand is important as it causes direct costs to a company. It is also important to identify trends in communication as typically traffic demand increases over time. If new capacity is acquired only when shortage is already impacting operations, it is too late. Installing new capacity may not happen overnight or even in month while the demand increase may result from a single software upgrade.

In agreements between corporations there are *service level agreements (SLA)* where some *key performance indicators (KPI)* are defined. These include values such as available throughput, packet loss, downtime and time to repair.

> **?:** Identify what kind of measurement needs there would be for different kind of companies: game server operator, web retail store, video streaming service, call centre, architect studio, hospital, ...

1.5 Network providers

A larger provider network is a significant investment. For a mixed mobile and fixed line operator the investments are tens of euros per year per consumer subscriber especially when technology change is underway. These are capital expenditures (CAPEX). Annual investments can be larger than monthly average revenue per user (ARPU). The resources must be used in efficient way.

In addition to CAPEX, companies need to focus also on operating expenses (OPEX). Operators that have significant consumer subscriber base need to careful to avoid operations that may result large number of customer interactions. Quite often, investments may result lower operational costs. Investments that focus on capacity increases should be based on a forecast backed by measurements.

> **?:** What expenses are part of CAPEX and what OPEX? How can provided move weight into another class?

As networks are more critical for their users and society, reactive problem fixing based on trouble tickets submitted by users is not sufficient. Using measurements and diagnostics from the network it may be possible to identify

problems developing before they impact users. It is more economical to take corrective actions in course of normal operation than as emergency repair.

It is not only upgrades to network that put a demand on measurements; accounting is one of those. Telephone calls have been charged from early on by duration. On manual exchanges the operator marked calls in book and later in automated exchanges a pulse was generated with fixed intervals. Subscriber was charged by number of pulses. On computerized exchanges call lengths are recorded in seconds and charging can be done exactly if wanted. Some providers charge minimum call length, like one minute.

> **?:** How is different if pulse is generated evenly from start of call or at random time points

Internet traffic can also be charged by volume or then there may be a monthly quota. This is common in mobile networks but also many fixed networks include limits. If quota is exceeded then there are three options:

1. Additional extra traffic charge applies. This may become expensive to a customer and result bad experience. There may be user confirmation needed for additional "data package".
2. Service is cut until next period starts.
3. Service is rate-limited and/or is put to lower priority compared other traffic.

In any case, if there are financial implications from measurements, the measurement system must be well designed and well tested. If there is disparity of traffic measured by operator and traffic measured by customer, it may have significant implications.

> The author has first-hand experience about that: one server hosting provider had on layer 2 issue in their network. As a result, the server was accounted nearly thousand time traffic it really communicated. Without my own monitoring I could have not convinced hosting provider that volume was in error.

In addition to customers, also regulatory authorities may have something to say about network performance. It may be, for example that there is mandated minimum capacity that must be available for every customer. In Finland every permanent residence must be able to get minimum 1 Mbit/s broadband. Or network disruptions may not exceed certain percentage of time.

As in Corporate subscribers was discussed, service level agreements and other contracts with corporate customers may require provider to measure its network. A monthly report may be required for performance.

Competition is also one motivation for measurements. It is not uncommon that

providers hire independent companies to conduct measurements and compare quality provided by provider and its competitors. If results are good for the provider, results can be used in marketing. It is important to remember, by selecting criteria and cherry-picking results the same data can show any provider as "the best". Of course, also provider can do comparative analysis on other networks also by its own – these results are not usable in marketing as one made by independent company, however.

> **?:** What kind of cherry-picking you have seen network providers to use.

A nice example how to select data points to support wanted conclusions can be found from tamino.wordpress.com.

1.6 Types of measurements

While network users can in most cases handle provider network only as a black box, a provider or company running its own network can have more insight on network components.

The first and most basic metric is measuring network **performance**. It is important to monitor any errors or anomalies because these can be a sign of soon failing equipment. Network performance can be analyzed by sending test traffic and measuring packet delay and loss among other for that traffic.

A part of network performance is of course individual network devices and end systems. When deploying new hardware, it is very important to verify they match performance in *intended* configuration and usage scenario. It is not uncommon that by enabling both feature A *and* B would drop throughput significantly compared to the case where only one of them is active. If services are provided, these must also provide performance required.

Utilization indicates if there is a need for upgrades or other changes if some link is too congested. A related metric is **traffic demand**. If the network is not able to carry all traffic, the demand can be larger than demand. Also, if network quality is bad, users may not be able to utilize it as much as they want.

It is not use to have network that provides high throughput but is unavailable long periods of time. **Availability** indicates which part a network is usable for the user for indented purpose. Typically it is indicated as percent like 99.8% – the network is unavailable nearly 18 hours a year. A related property is **stability** that can tell if for example delay or throughput varies over time.

> **?:** How long unavailability is allowed if service promise is *five nines* (99.999 %)?

Last, but not least is the need to understand **user behavior**. What is more important is to understand what kind it will be in the future. Like recent increase in mobile video usage: both creating video and consuming it, has greatly increased traffic. Traditionally networks have been planned similarly to broadcast media: user is a sink of information receiving lot and sending little. Now users are creating high-definition video content and uploading should not take too long time. Or even one should be able to stream video real-time.

Same type of measurements can apply also for corporate networks.

> **?:** What are minimum performance characters you would want to have your home network? Are they different compared when you are at university, café, at cottage, or travelling in train or plain?

1.7 Network provider requirements for measurements

As stated earlier, a network is a long-time investment and operations must have continuity. There is a need to have common standards to collect measurement data. For example, it is not sufficient just to have common protocol to transfer measurements but also data collection must be uniform: any inconsistencies in statistical definitions, protocol levels, or in data collection should be avoided. For example, are layer 2 headers and framing included in byte counts or just IP packet content?

Measurement system must scale as network grows and transmission rates increases. Data must be aggregated as much as possible to have measurement data increase be slower than user traffic increase. If test traffic is sent to network, it should not interfere with user data transmission. This may be a challenge if there are very strict requirements for verification or on low-bandwidth links.

> **?:** It is estimated that IoT and digitalisation may increase number of end hosts 1000-fold. 5G networks will have many more base stations compared to currently in 4G. Some data processing may happen on network edge (MEC). What impact this will have on measurement architecture?

Traffic engineering (TE) are ways that make possible to optimize network performance. These are based on traffic measurements. One important tool is traffic matrix. Traffic matrix tells where traffic originates and where it is destined. Using traffic engineering it is possible to improve network utilization by distributing traffic on alternative links.

Mobile Edge Computing (MEC) and similar distributed data processing technologies can also be used optimise traffic flow in the networks and this must be

also be accounted for in making measurements.

Many of criteria above also apply for corporate networks.

1.7.1 Network provider time scales

Network provider can utilize measurements in different timescales.

1. Network planning and extension takes **months or years** to meet future needs for capacity and reliability. Introducing new technologies takes also long times depending if they are compatible to already installed base or not.
2. In **hours or days** it is possible to reconfigure network to optimize utilization and performance. For example changing routing to balance traffic is one possibility.
3. In case of failure or traffic surge actions can be taken in **near real-time** in sub-second scales automatically. Human operation is slower, but expert insight may improve situation compared to pre-planned automation.

> **?:** Think what kind of operations can be accomplished in various time scales.

Network utilisation is measured with SNMP data collected with one minute intervals typically. This provides view to average network load. However in datacenters, especially with distributed big data workloads, there exists microbursts i.e. periods of high load less than 200 µs. The time is less than RTT and as a result congestion control based on packet loss or marking (ECN) do not have effect. While switch buffers may be able to absorb these, care must be taken when designing applications and protocols. [1]

1.8 Vendor measurements

Compared to other players, networking hardware and software vendors have slightly different needs for measurements. Time-frame is much longer, as it can take years before a planned device is ready to ship to the customer. Software modifications may happen faster. The vendor must be prepared if the reality does not match to the forecast.

First and foremost, the prospective buyer will check the raw figures. How many packets or bytes per second the device is able to relay? How many ports it supports? Does it support any traffic mix at line rate? If it is a security device like IDS[1], can it provide protection in high loads too. To verify these, the device must be benchmarked.

[1] Intrusion detection system

To help with planning, the vendor needs up-to date information about network load and traffic characteristics. If a wrong type of traffic is assumed, it may be a poor match for needs. An example is the increase of video uploads: the network must support more balanced traffic flows and not just one-way streaming.

It is not sufficient for device to just perform it primary task. It also needs helping functions such as diagnostics and measurement facilities. If device is not performing as expected, there must be way to extract performance figures from a device. These include different interface counters, system internal indicators such as CPU load and memory usage. The better an expert can have view on internals of system the better she can pinpoint problems.

> **!:** Resolving problems without sophisticated tools can be very tedious. The system may have option to enable debug log but this may often result too much information so that the root cause of issue is hard to find.

Device testing can be implemented as black-box testing alone to find performance limits but for a better understanding some methods from at least grey-box if not white-box approach give more fruitful information.

The device must also support measurement needs of the user. At least basic traffic statistics such as packet count and bit rate per interface are the minimum. Flow statistics and sampling packet information can be required by the user.

> **?:** Have you performance-tested any network device? Did it affect on acquisition of device?

2 IP networks and how to measure them

Internet connects different networks together and allows communication even to the space. While publicly visible Internet is huge, there are many other IP-based networks sharing the same technology.

One can measure on links, on network devices or on end systems. Here we present basic introduction how different components in the network affect on traffic going through the network.

> **?:** How one can know if performance problem is at end system, link, on router or on some other component in the network?

2.1 Measurement types

Network measurements can be made in two different ways: *test traffic* or *synthetic traffic* is used or *existing traffic* or *user traffic* is observed. These are known as active measurements and passive measurements. These terms will be used throughout this course.

In **active measurements** test traffic is injected into network and it is observed. A typical test is to send a packet like ICMP Echo Request to a destination and wait for a response. If response arrives, one can measure round-trip delay. If no response is received then one of packets was lost in network or destination host is not at network. By sending multiple packets, distribution of delays and proportion of lost packets can be calculated.

As the name indicates, in **passive measurements** no packets are sent. Existing traffic is observed. This can be done by capturing packets, monitoring interface counters or other ways collecting information. The key benefit is not to add additional load into network. On the other hand, if there is no traffic towards destination nothing can be observed.

2.2 Packets in IP Networks

In the packet networks, like in the Internet the smallest interesting unit is a **packet**. The packet arrives to its intended destination or not in which case it is declared as **lost**. It takes some time before packet will arrive to its destination and this is called as **delay**. It is also possible that it is *delayed* more than expected and its usefulness is lower or even negative compared if it arrived in time.

Each packet consumes some part of network **capacity**. While bit-level events are not in the interest of this course, they may result in *corrupting* the packet if some bits get inverted, for example. In most cases link-level error detection will remove these packets before they are forwarded.

> **?:** What are the most common contributing factors to delay or bit errors?

One transmission typically includes many packets. In ideal case, all packets arrive destination, have identical, minimal delay and same order they were sent. However, this might not happen. A group of packets can be a **flow**.

2.3 Flows in IP Networks

While routers in network process individual packets, applications typically create flows. A flow is a series of packets travelling from one part of network to another part of network (packet train). A flow can be either **unidirectional** or **bidirectional**. In first case a packet going A→B is in different flow than one

A←B. Definition depends both type of analysis to be done and measurement point. It is not guaranteed that one can observe both directions at the same link or device.

Internet routing is not usually symmetric: so called *hot potato routing* results typically asymmetric paths [2]. If a company has redundant routes to the Internet, response packets may enter via another gateway. For this reason, many analyses in Internet transit networks do assume unidirectional flows. Measurements on network edge or in access network can assume bidirectional observations.

The second property for a flow is its **granularity**: what criteria are used to determine which packets belong into one flow and what into another. If a flow is considered mostly identical to a TCP connection, then IP source and destination addresses, IP protocol, TCP source and destination ports would form a selector. This is the most detailed criteria for a flow that is commonly used.

For example, if we would just try to define all traffic from one user to IMAP-email server to be part of a single flow, we would use following selection:

- source IP address defines user host
- destination IP address defines server
- protocol would be TCP
- destination TCP port defines application.

That would be for unidirectional flow. For return traffic, we just need to reverse any source to destination and vice versa. Other commonly used criteria are source- and destination IP addresses or source- and destination networks. One may be interested traffic between two autonomous systems or domains. In principle anything that can be used packet-by-packet basis can be used, including links and interfaces.

> **?:** A web browser typically opens multiple TCP connections even to a single server. Part of page may arrive from different server. How you would define flow here?

The third property is to define **when a flow starts and when it ends**. We can use protocol messages like with TCP observing SYN/FIN/RST flags. Unfortunately, not all protocols have similar and not all connections end gracefully. Also it may happen that one or both parties lose their network connectivity or reboot and so connection is not terminated properly. An universal method is to use timeouts. If there are no packets that are part of flow in 60 seconds[2], for example, we consider the flow to end. Flow timeout selection will have an effect on the size and number of flows, of course.

An alternative for a fixed timeout is to have a dynamic timeout. One initially defines maximum packet inter-arrival time to be N seconds and timeout M times

[2]Another popular time-out is 64 seconds, it makes easier to divide interarrival times into bins of $2^{\mathbb{Z}}$ seconds.

inter-arrival time. If packet inter-arrival time t is larger than N but less than $N \times M$ seconds, inter-arrival time N is increased to value t. This will adapt on different applications by identifying segments of flow.

2.4 Defining network properties

When discussing, it is important that all agree on common terms. Because this seldom happens universally, we define some terms used in this course.

- **Throughput** is the number of binary digits that a system accepts and delivers per unit of time. Unit is bits per second or bytes per second.
- **Delay** is the time elapsing between the emission of the first bit of data block by transmitting system and its reception by the receiving system. This is time, typically measured in milliseconds.
- **Error rate** is count of errors per transmitted data or per time unit. Expressed typically as ten's negative exponent: 10^{-6} indicates that one out of million bits (or packet) has an error.

2.5 Throughput

The throughput observed in some part of the network may be quite different compared the throughput an application observes. If network conditions are bad it may result a large number of retransmissions. A link middle of network does then see quite high bit rate but application gets much lower rate. Throughput an application sees is sometimes referred as *goodput* i.e. amount of useful information compared to raw bit rate.

> **?:** What properties of application or transmission protocols do affect if throughput and goodput are near-identical.

Network links are typically key bottleneck in considering throughput. While fixed links in cables like electrical and optical links typically have constant capacity, there can be lots of throughput variation on wireless where capacity may depend on link length and number of other users.

Many wide area networks can be considerd as *overlay networks*. Those are typically virtual networks i.e. topology seen on IP level does not match for the physical network. For example one IP-level link may actually be multiple layer 2 links (MPLS, Ethernet VLAN, ...). Traffic that is competing from the same capacity is not visible and available capacity may be much variable. Packets can be lost on link without any consideration to possible traffic priorities.

Routers handle packets in network: they receive packets and check where to send packet next. Routers have internal architecture that may result bottlenecks depending how traffic flows. A router may also be limited by *packet rate*: this typically manifests itself if there are lots of small packets.

Of course, the end system may be ones limiting data throughput. There may be processing associated in either sending or receiving data. Or disk, some device internal bus or the network interface card may be a bottleneck. Limits may result also from parameters and settings. For example, if TCP window size is 16 kilobytes, one cannot expect high throughput.

Many applications do not just open connection, send with maximum throughput and close connection. They may throttle transmission or pause temporarily. For example a web page maybe not load completly at once but is updated and components loaded while user scrolls further.

> Traditionally throughput at network level is expressed as bits per second and "kilo" equals 1,000. At application level also bytes or octets per second is used and "kilo" equals *usually* 1024. To avoid vagueness, one should use new binary SI prefixes of *kibi* (Ki), *mebi* (Mi) and *gibi* (Gi) when multiplicative prefixes of 2^N*10^ are used.

2.6 Delay

There are two main sources of delay: distance and processing.

On network links the delay results from distance and signal **propagation** speed. While radio waves travel at speed of light, speed in optical cables is "just" 200,000 km/s - 5 ms per 1,000 km or 1 ns per 20 cm.[3] Another delay component is **transmission** delay: how long it takes to send full packet on the link. This is derived from bit rate of the link. If the link is shared or slotted like many radio networks, there is an **access** delay while system is waiting its turn to send the packet.

> **?:** How delays compares using geostationary satellites, low-earth orbit satellites, drone planes, or aerostats to provide network coverage to rural areas?

Processing delay typically happens in routers. Packet forwarding lookup may result (variable) delay. The main source of delay in routers is **queuing** delay: when there is more traffic offered to a link than it can deliver, a queue will fill. There are different queue management methods to control queue not to become too full. While packet is delivered to a link, it is subject to delays on link.

Most network devices receive first whole packet before starting to process it: *store-and-forward*. This adds delay that depends on **line rate**. For many purposes, including checking error correction, whole packet is needed anyway.

[3] A simple rule of thumb: in 1 nanosecond signal travels length of longer side of A4 sheet in air and the shorther one in fibre.

Some network devices, mainly Ethernet switches for data-center use implement also so called *cut-through forwarding*. Here frame forwarding starts once its destination address is determined. This reduces latency and is used e.g. in storage networks in data centers where even small latency impacts performance.

> **?:** How much shorter the forwarding delay is for 1500 byte Ethernet frame at 10 Gbit/s with cut-through forwarding? How if jumbo frames with 8 KiB IP payload are used?

Average delay inside router depends on traffic load. As memory prices have become lower, adding more buffer memory to routers is not a financial limitation. If packet loss is a problem, by adding more buffers this can fixed for sure. However, large buffers create *bufferbloat* that only increases delays and in many cases causes unnecessary packet retransmissions.

On application viewpoint, data may be delayed because of missing TCP segment, for example. Operating system is not able hand already received data to application if it has requested continuous stream of bytes but there is a hole in sequence. For applications that can allow some data discarded in exchange of timely delivery, datagram-based protocols (UDP, DCCP) are more suitable.

> **?:** How common network programming API (sockets) works if a fragment of data is missing.

2.7 Delay variation

In some cases the impact of the **delay variation** is more significant than absolute delay. Even so, the delay of subsequent packets compared to the first packet(s) affects on user experience.

> **?:** What happens in VoIP call if the first packets have lower delay than subsequent packets?

Another term for delay variation is **jitter**. While it is used as synonym, most often it is used for bit or sample level delay variation and not on packet level variation.

2.8 Error rate

In packet networks, most errors manifest itself as packet **loss**[4]. Other types of errors such as **bit errors**[5] result usually packet to be discarded on next router

[4] Packet Loss Rate / Ratio (PLR); Packet Error Rate (BER)
[5] Bit Error Rate (BER)

incoming interface when link-level error checking code such as CRC fails. It is also possible packet to be delivered for a wrong end device.

> **?:** IPv6 header does not have a checksum field while IPv4 has one. What may be the reason not to include checksum to new version?

Bit errors or **data alteration** often happen on network links. Optical links are not affected electromagnetic noise along cable and thus have very low bit error rate, typically from 10^{-9} to 10^{-12} or better. Electrical links like Ethernet on twisted pair can suffer electromagnetic noise like ones resulting from switching power on and off nearby. Radio links have much higher error rates. On fixed radio links the error rate can be quite good like 10^{-8} but on mobile networks with poor coverage it can be 10^{-4} or worse.

If bit error happens and there is no error-correcting code that can fix it, the packet is most likely discarded by receiving device. However, bit errors can happen inside devices too. There may be some internal bus operating badly or then data corruption happens in memory. In these cases error is not detected while in transit so responsibility lies on end systems to check integrity of data.

The most reason for packet loss caused by routers and end devices is queue overflow. There may just queue becoming full or then there is an *active queue management* taking place. Also in high-load situations packets may be lost because the receiving system is not able process those in time.

Yet another type of error is **data duplication**. One or more identical copies of a packet exist in network at same time or within short period of time. There are multiple possible causes, including equipment malfunctioning, but one common place is a wireless network. Because not all devices in wireless network hear every other device all the time, it may happen that terminal thinks its transmission has failed because of collision and re-transmits packet. The receiver, however, receives both successfully. As Ethernet and IP packets do not include unique frame identifiers, both are forwarded.

> **?:** IPv4 header has *Identification* field. Why this cannot be used to identify duplicated packets?

End systems are prone for many same errors as routers. They may not be able to process that many packets per second, for example. Losing packets may happen even if operating system has received it successfully: an application is not able to process incoming data fast enough. Data may be corrupted while it is stored into memory or while it is being transferred to network card or to disk.

> **?:** Have you witnessed data corruption while it has been transferred over network or copied to portable media?

In addition to routers, packets can be handled by some middle boxes like firewalls. These are mostly like routers, but quite often do processing on CPU: this may result packet rate limitations and more variable delay. A firewall may record state information for each flow. To save memory, old flow information is purged from records every now and then. If a connection has been idle for a while, a packet may be unknown to the firewall and gets rejected.

Also switches handle packets on link layer and may result packet loss in case of congestion.

> **?:** Have you identified cases of bit errors either on network or inside end systems?

3 Active measurements

In **active measurements** test traffic is injected into network and it is observed. Active measurements measure treatment traffic receives when it travels from one part of network to another. Measurement point is typically in an end system or at some demarcation point, like at the router between customer and network provider.

Measurements can be taken one-off in case of resolving an issue or then repeated for continuous quality and performance monitoring. Continuous measurements may identify problems before they impact on the users.

3.1 Co-operation in measurements

Typically network measurements are performed over long distances requiring to use devices in different locations. We can identify three different classes of measurements:

- **SO**: Sender Only measurements assume only standard assumed functionality from the other end. Every IP device should respond for *ICMP Echo Requests*, for example. Or a web server would return requested document.

  In these measurements it must be assumed that the end system functions according to the standard. This may not be always an case and depending on measurements, it may not be possible to identify an erroneous behaviour. This is typical case of measurements of services run by other entities.

- **SRP**: Sender and Receiver Paired measurements. In this case we have control of both ends or the functionality of remote end is well defined.

There are properties that allow more accurate measurements by using software and messages tailored for the measurement in hand.

Typical improvements over standard application protocols include using of accurate time stamps for both receiving, processing, and sending. Also additional information including likes the CPU load of end systems could be included into reports to identify if the challenge is at the end system.

- **RO**: Recipient Only measurements provide most limited functionality. These provide most limited functionality as the one who measures has no control on sending messages and is even more dependent on the other party to perform according to specification. Many measurements here relate for Passive measurements.

> **?:** Consider what kind of custom software or hardware deployment needs there are in each class? What is the scale of deployment?

3.2 Application measurements

Application performance is the key aspect that matter to the network user. While it at first sight looks easy to measure, it is not always clear how measurement should be conducted. An application may have very complex interaction with network, uses multiple protocols and accesses different servers. In most cases for example DNS is used. At worst case some of critical steps are not measured at all.

An obvious way is to record a user performing tasks with application and then have software to emulate user actions repeatedly around the clock. If an application is using a proprietary protocol this may be the only way to estimate it. If task to be tested includes for example entering records to a database, there may become a problem with data integrity and separating test records from real one. If tests are directed to test data database, then the production database performance is *not* evaluated. User authentication and authorisation, transaction audits and similar also be a bottleneck that is different in test environment and production environment.

If application uses standard protocols, then it may be easier to enumerate network operations the application makes and have these separately tested by using different software step by step. However, modern single-page web applications are more complex to benchmark compared to traditional HTML+images pages. For example, many use websockets to communicate between the server and the client. Replicating this with traditional HTTP tools is difficult if not impossible.

> **!:** Google Chrome browser has headless mode that allows programmatic browser operation. There are also other programs

like cutycapt that interpret CSS ja javascript. The result image can be saved as in PDF, JPEG, or PNG format.

3.3 Application protocol measurements

Instead of testing full application, in many cases it is sufficient to test individual application protocol entities. For example, a simple HTTP query to `http://example.com/page.html` includes following operations simplified:

1. Resolve `example.com` host name to IP address. This may include many *DNS* servers and multiple requests. Let's assume answer is `192.0.2.3`.
2. Open *TCP* connection to `192.0.2.3` to port `80`.
3. Issue *HTTP* request for `/page.html` that server interprets and retrieves content from storage like hard drive.
4. Response is transferred over *TCP* connection.
5. Connection is closed.

The list above is incomplete even in wired Ethernet. If you are connected via WLAN, what additional steps may take place?

From above, we can identify several phases that contribute overall performance and can be individually recorded.

1. *DNS* query time. Because of caching, this can be very variable over time.
2. Time to establish *TCP* connection with 3-way handshake. This depends on network delay and server load.
3. *HTTP* request processing time including possible look-ups and database access both in client and server.
4. Actual data transfer over network i.e. throughput.

If that would be a real web page, one must retrieve all page components like images, style sheets and scripts too. To be able to identify all components, for example possible JavaScript must be evaluated emulating real browser all display properties and such.

Present-day web pages can be very complex to evaluate - part of page may be cached in browser cache and not requested with every page load. Also much of page content can arrive from a *content distribution network* (CDN) that has servers closer to the users than the origin server. A part of page content may be loaded only based on user actions, like howering mouse over elements or by scrolling page further down.

An application may have also multiple different protocols associated with it. A VoIP call, for example, gets started with *SIP* signaling, media is transported over *RTP* and call is again ended with *SIP*. A measurement would include both *SIP* and *RTP* analysis.

In any case, it is important to include all components of real transaction and be able to identify the problems. It is not worthwhile to try to optimise web server response times if there are problems in WLAN authentication, *ARP* resolving or local *DNS* server.

For practical testing, it may be worthwhile to divide application into components and test them separately: tests for DNS requests and set of HTTP tests for various page components; maybe just using sample of requests for each server.

> **?:** Visit few different types of web sites. Use developer tools in browser to identify
>
> 1. how many requests are performed,
> 2. how many servers those are distributed over,
> 3. how many domains those servers are distributed over?

3.4 Transmission protocol measurements

On application measurements the end system and server software plays very significant role. If one wants to know how the network affects on data transmission, then tests can be done without any application dependent processing.

A simple way to benchmark a network is to just open a *TCP* connection and then other end sends dummy test data as fast as possible and other receives as fast as possible. Typically, either fixed amount of data is transferred or transmission runs fixed time. Average bit rate is simply number of bits transmitted divided by time.

> **!:** Run an *iperf* test to `iperf.funet.fi`.

This removes any server software or subsystem bottlenecks from affecting measurements. It does, however, depend on *TCP* version and parameters used in addition to hardware.

For example, Linux kernel supports multiple TCP congestion control mechanisms. By default the kernel in e.g. Ubuntu 16.04 allows cubic and reno for normal users. One can check allowed ones from `/proc/sys/net/ipv4/tcp_allowed_congestion_control`

Linux source code includes more information about each in comments at start of each `tcp_*.c` file.[6] There you can find bibliography or a link describing that congestion control.

All modules that are compiled with kernel can be found from `/lib/modules/`*kernelversion*`/kernel/net/ipv4/tc` Admin user can take them into use by loading the module. To al-

[6]Note that some `tcp_*.c` files are not congestion control but some other supporting files.

low any user to select mechanism, allowed names should be written `tcp_allowed_congestion_control` file above.

> **?:** Check what TCP congestion control mechanisms are supported by current operating system version. If you have admin rights, activate few and compare e.g. `iperf` results.

3.5 Packet level measurements

In pure Internet, the network only inspects one packet at time and has no memory of any earlier packets. While this is not always a case – like with traffic inspecting firewalls or address translation devices – it is still a good approximation.

> **!:** IP networks provide no guarantees that subsequent packets between two hosts travel over same path.

Each and every IP device must be able to process and respond for *ICMP* messages [3]. Among various error reporting functions, there is a way to ask another host to send response packet: *ICMP Echo Request*. On receiving this packet, the host will respond with *ICMP Echo Response* packet destined to sender of the request.

Each ICMP request includes both *identifier* and *sequence number* that allows request sender on receiving response to know for which packets it received a response for. By sending multiple packets, it is possible to learn few properties from the network:

- How many packets we received response for?
- Round-trip delay for each packet and subsequently we are able to calculate statistical values like mean and variance out of those.

If response was not received, there are three options we cannot separate from each other:

1. Request packet was lost in transit to destination.
2. Other system failed to respond. Reasons include overload, policy, rate limit, and host being down.
3. Response packet was lost.

Similarly we cannot know how long the request packet was delayed in networks, how long it took remote system to respond and how long response packet was delayed. We just know time it took from sending the packet on receiving to the last packet. While information is lacking, this is sufficient for most purposes to know about network conditions. Most protocols do have some kind of request-response structure and total round-trip time is what matters.

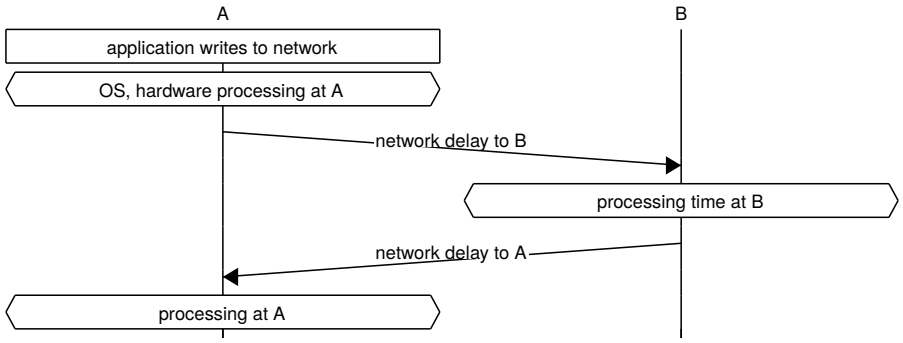

Figure 2: Delays in RTT mesurement

> **?:** Have a look on few different protocols like DNS, TCP, RTP, FTP, NFS. Do they have some fields that may allow mapping request-response packet pairs?

If we want to improve our measurements, however, there are some simple steps to implement. At first, remote end keeps track of sequence number of received packets and reports these back to the sender. For example, we could add sequence numbers or bit mask[7] of recently received packets. Now if one reply packet gets lost, the sender will still learn fate of packets if it receives some of the responses.

The second improvement is recording time stamp when a packet was received and reply was sent. For high accuracy, these are separate time values but for a simple case both can be same. If we assume accurate clocks, one can calculate delays in both directions separately. However, providing accurate time is not trivial, as we study later in Time synchronisation chapter.

For example IETF OWAMP [4] and TWAMP [5] protocols implement these features. Application protocols such as *RTP* include control fields that can provide diagnostic tools for applications.

> **?:** Look up ICMP, RTP, OWAMP, and TWAMP packet descriptions. What kind of measurements latter two allow that ICMP or RTP do not?

[7] Bit mask is a compact way to represent information. Each byte contains 8 bits so with 4 bytes on can report statuses (received/not received) of $\text{pkt}_{\text{n-1}}$ to $\text{pkt}_{\text{n-33}}$, for example.

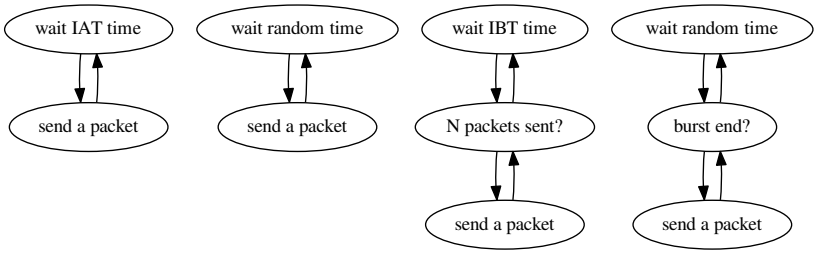

Figure 3: Some packet send processes

3.6 Uniform criteria for performance

Determining the packet in the network sounds quite simple. One just sends a bunch of packets and observers how many replies one gets back. Someone else can make the same measurement at the same time on the same network, but get quite different results.

For example, were packet sent with constant intervals, random intervals or in bursts of uniform or random sizes and interwals? When is a packet declared lost, does one wait for three seconds, one minute or until measurement concludes? How possible duplicates are handled: can one replace a lost one or can it be identified? For this reason standard metrics are needed. There are ones to measure network devices like RFC-2544 [6] and for network ones specified in IP Performance Metrics working group.

The goal has been to define set of standard metrics so that one can design a measurement system that provides similar results at the same situation like different, independently developed system. One can measure *quality*, *performance* and *reliability*. Measurement can be done by network operators, end users or third parties and they all should give comparable results. How numerical values are explained, are those good or bad, are left for each individual to decide. For instance, 0.1 % packet loss is a dreadful value in datacenter storage network but very tolerable for a mobile network in wilderness.

> **?:** Can you identify applications that are sensitive for one metric but not as demanding for the other?

Metrics defined in IPPM include connectivity [7], delay and loss for one-way [8], [9] and round-trip [10], delay variation [11], loss patterns [12], packet reordering [13] or duplication [14], bulk transport capacity [15], [16], and link bandwidth capacity [17]

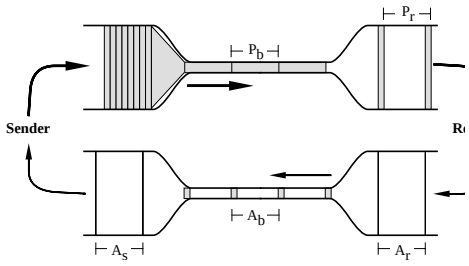

Figure 4: Self clock in packet networks

3.7 Network throughput measurements

Considering network and its throughput, two properties are most interesting. The first is the *maximum* throughput i.e. **bottleneck capacity**. If there is no other traffic in the network, this is the throughput a connection would archive over current path. The second is **available capacity** that is often more interesting from the user perspective. It is important to understand that *current traffic + available capacity* does **not** equal *bottleneck capacity*.

?: Why *current traffic + available capacity* $\neq$ *bottleneck capacity*?

One possible way to identify bottleneck capacity is to send test traffic in many instances over period of time. With some probability, the bottleneck link has low utilization at some point of time and also other links do not limit throughput. The measurement with highest throughput would be the bottleneck throughput.

This approach has its limitations, however. It may require many measurements to hit "empty" network. Furthermore, it does not identify on which link that bottleneck is. An alternative is to use *packet pairs*.

If two packets are sent back-to-back as fast as possible the second packet is delayed by time that is dependent on link speed and packet size. If we assume the first link is a 1 Gbit/s link, a 1500-byte packet is delayed by 12 µs.[8] If the same packet travels later over link that has bit rate of 10 Mbit/s, the second packet is delayed by 1.2 milliseconds. If all other links are have higher bit rate and there is no other traffic, the packets maintain their time separation when they arrive to destination.

The destination host then replies to these packets with smaller packets. Again, if there is not much additional traffic, these packets have maintained their separation when they arrive to original sender. By measuring the time difference, the bottleneck speed can be identified. [18]

The result still does not tell us which link was the bottleneck. One can identify the

[8] 10^{-9} seconds (1 nanosecond) per bit, $8 \times 1{,}500 = 12{,}000$ bits.

bottleneck by sending packets with different time-to-live (TTL) values similarly to `traceroute`. Each router in path then replies to packets by its turn and each link with its capacity can be identified.

> **!:** Test a network with `pchar` or `bing`. Compare results with `iperf`.

While single pair of packets could give mostly correct results, in practise many more packet pairs or packet trains are needed to have any confidence on results. [20] Most methods assume traditional fixed line network that has no variable delay sources other than queuing. These methods work quite badly when these expectations are not met like in mobile broadband networks.

3.8 Network delay measurements

Delay measurements are quite easy if one is interested about round-trip time. One sends a packet and waits for the response. A response may be a single packet like in *ICMP Echo* or more complex like *TCP 3-way handshake*. Both can be used as a way to estimate for RTT. One needs to take possible end system delays into account. A TCP server, for example may perform some lookups to find out if it wants to allow us to connect to.

Absolute one-way delays are more difficult to measure. Accurate time synchronization is not a trivial task. Typically for example few milliseconds accuracy can be reached with NTP (Network Time Protocol). Short-time changes can be monitored with better accuracy but also there one must be prepared for a clock skew.

As with any test traffic, a possible traffic classification must be taken into account. The test traffic may receive better or worse treatment compared for normal traffic. Radio technologies such as UMTS have different channels with different characteristics. *Random access channel* (RACH) is used if there are only few packets to be sent. If mobile staion sends many packets, it may be allocated an *dedicated channel* (DCH). A packet expires lower delay when sent over DCH and the delay is dependent on packet interval and threshold of RACH-DCH switch. [22]

Network delay measurements are the workhorse of network monitoring. With frequent delay measurements, one will learn:

1. Connectivity problems.
2. Packet loss.
3. Delay variation.
4. Network overload situations.

> **?:** What additional value does one receive from measuring directions separately compared to just recording two-way roundtrip message delays and loss?

3.9 When to measure

A straight forward method for measurements is measure continuously or with even intervals throughout the day. For example, we can monitor delay by sending one packet each second. This is easy to implement and we get 86,400 measurement results each day. For each sent packet, the packet is lost or we have a round-trip-time recorded.

Listing all measurements is a bit too much, so we like to have just one figure to report. Taking mean value is simple and we only take account any packets that arrive within 2 seconds. Otherwise we report them as lost.

Our daily average would be then 75 milliseconds that is a reasonable value for an international VoIP traffic. However, we receive complaints about bad quality related for increase of delay at day time. When data is looked more in detail, it is found that during day time (9-15) the mean value is 200 ms that is not a good value for VoIP. Mean value outside of that time is only 33 ms.

It is quite clear that monitoring performance of empty network has very a little value - other than checking the network is working. The values derived from measurements must be weighted by its impact on users.

> **?:** Consider you are responsible for providing network quality measurement results for a company with 10 sites. You need to provide for CIO daily report. What kind of report you would provide?

Considering available bandwidth, an similar observation can be made. If you are out of home at daytime and sleeping after nightfall the available bandwidth be that 20, 50 or 150 Mbit/s to your home does not matter at those times. However, if you want to watch your favourite series from streaming service after dinner, you would be happy to receive constant 5-10 Mbit/s for video.

4 Passive measurements

While active measurements provide strong insight towards how network behaves under additional load, it does not answer about how network and services are being used. Passive measurements provide insight on *existing traffic* in the network. It can answer on many same questions active measurements can answer too, just a slightly different view. As with active measurements, passive

measurements have a huge range of scales between very detailed and highly aggregate information.

Because passive measurements observe user traffic, there is a risk for privacy violations that may lead into legal problems. User data can include sensitive personal information and passwords, credit card number and other confidential information. End point identifiers (IP addresses, telephone numbers) can identify a single person or household. Information about with whom one communicates with is sensitive information. This can reveal human relations. List of applications that are used within a network segment can be also sensitive information. These questions are discussed in detail in Privacy chapter.

4.1 Counters

Most network devices, including end systems, collect very basic statistics on traffic they generate, forward or receive. Most operating systems have command `netstat`[9]. When run with `-s` option it will report large number of different counters, including count of IP packets received, forwarded and sent.

```
Ip:
    60885114 total packets received
    9895 with invalid headers
    192368 forwarded
    0 incoming packets discarded
    60681949 incoming packets delivered
    32992714 requests sent out
    2800 outgoing packets dropped
    6203 dropped because of missing rout
...
```

These are overall statistics and these will tell how busy the system is overall. In networks, we are often interested on link utilisation levels. These can be seen from interface-level information like one below:

```
> show interfaces xe-5/3/3 statistics detail
Physical interface: xe-5/3/3

  Logical interface xe-5/3/3.3200 (Index 1240) (SNMP ifIndex 591) (Generation 1049)
    Description: Uplink to funet
    Flags: Up SNMP-Traps 0x4000 VLAN-Tag [ 0x8100.3200 ]  Encapsulation: ENET2
    Traffic statistics:
     Input  bytes  :        3969175664638
     Output bytes  :        6888285686661
     Input  packets:           5490812889
     Output packets:           7396842466
```

[9]Modern Linux systems do have `netstat` replacement as `ss` command that provides more TCP and state information but does not provide statistics in same format.

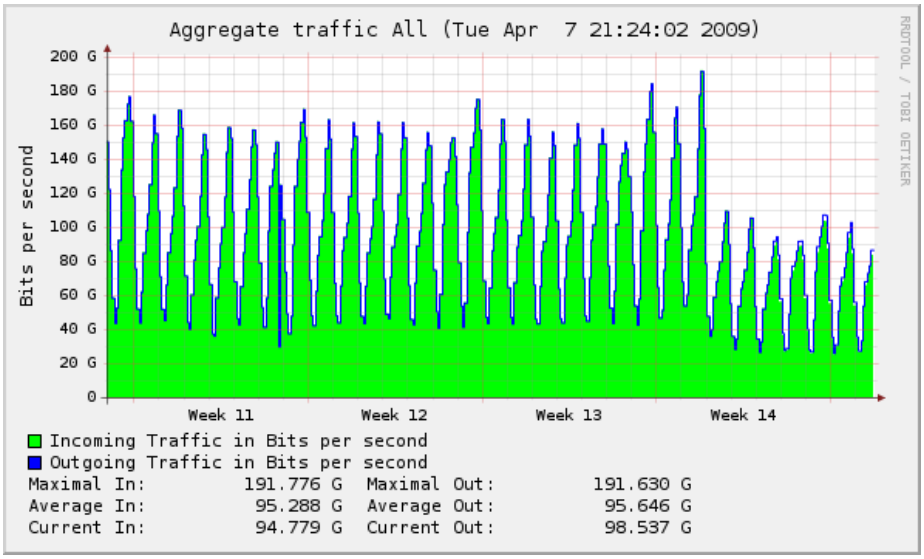

Figure 5: Traffic statistics from Netnod IX

```
    IPv6 transit statistics:
     Input  bytes  :        132395964214
     Output bytes  :         46040678048
     Input  packets:           187576308
     Output packets:           149566082
...
```

Collecting information manually is of course quite tedious task, but the SNMP[10] is designed to collect this information. Quite often interface counters are queried every minute and information stored into database for reporting. When this information is collected network-wide, network operator will have a good understanding on network load.

End systems provide interface statistics too. For example, in a Linux system one can use command `ip -s link` to get information on each interface how many packets and bytes it has received and sent.[11]

> **!:** Check statistics on your own laptop and on some Aalto ITS computer like `kosh.aalto.fi`.

[10] Simple Network Managament Protocol

[11] The older `ifconfig` used to provide that information but it is now obsoleted in modern Linux systems. Information is also available via `/proc/net/dev` file.

?: Is there risk of privacy violation if counters are observed?

4.2 Application: log files

A typical example of network traffic related log is a web server access log. The web server records information about each request. Depending of configuration and software, it can include following items:

- IP address of the host making request
- operation requested (GET, POST, ...)
- resource requested
- wall time the request was made
- response code (was resource found or permitted)
- bytes transferred
- time elapsed in serving.

Different stakeholders will look on different items. One responsible for the content on server is interested on how many time each resource is requested. Bytes transferred and response time (time elapsed in serving) are interest of who takes care of server and network performance.

Like web servers, almost all servers include some kind of logging function. These are useful also in estimating traffic volume and service times.

```
Sep 28 21:25:36 mailsrv postfix/cleanup[14351]: 4E677C116E:
  message-id=<008101d0fa1b$09297fc0$1b7c7f40$@example.net>
Sep 28 21:25:36 mailsrv postfix/qmgr[19685]: 4E677C116E:
  from=<emailuser@example.net>, size=132052, nrcpt=14 (queue active)
Sep 28 21:25:40 mailsrv postfix/smtp[14352]: 4E677C116E: to=<vast.ott6@example.com>,
  relay=smtp.example.com[2001:db8:1234::25]:25, delay=5.2,
  delays=0.96/0.01/1.1/3.1,
  dsn=2.0.0, status=sent (250 2.0.0 OK 1443464742 z1948878378fd.172 - smtp)
```

As log files are collected in any case for service monitoring and fault identification, it is useful to use them also for service demand and traffic estimation. One needs to take suitable precautions related to privacy, of course.

In addition to log files, there are services that provide lots of information about service users, among other performance information. A well-known example is Google Analytics for web pages. Many applications can be instrumented to provide feedback, for example WebRTC using monitoring.

When looking for performance estimates, one must consider all components of service process. For example, a typical email receiving process can have following subtasks:

1. TCP connection is established.

2. Waiting for any pre-greet traffic. This is a surprisingly efficient way to reduce spam emails. Well-behaving email server won't communicate before server responds while many junk mailers just push "dialog" through without waiting responses from the server.
3. Real-time block list lookups. Email servers check multiple DNS-based blocking lists to find out if the connecting server is known spammer or a host that should not be sending email.
4. Reception of the email.
5. Anti-spam and anti-virus processing. These can also include DNS or other requests over network.
6. Delivery of the email to a mailbox or forwarding it to a next server.

In addition, some servers use graylisting: if a previously unknown server connects, the first email delivery attempt is denied with a temporary error code. A proper sender will resend email after some time while drive-by spammer won't.

Similar behind-the-scenes processes are in many service processes. A modern web page is not just block of data ready to be sent to network as fast as possible but there may be multiple database lookups and communication with other servers.

It is important to understand **what** you are measuring. Is it network performance or database performance? Log file analysis can supplement active measurements by providing a different viewpoint.

> **?:** Consider web server. What kind of caveats there is in making conclusion on network quality?

4.3 Packet capture

A network provider or other entity does not have access to all end devices in the network.[12] Monitoring each end device is then out of question. We are then left to monitor traffic in the network. A network traffic can be observed via following methods:

1. Capturing outgoing and incoming traffic in end devices. Traffic is limited to that traffic where a host is participant and then some broadcast and multicast traffic.
2. Capturing traffic in a network device. This can be a purpose-built device that receives traffic, stores a copy and forwards packet. Also some routers, firewalls and network switches can have this kind of functionality but buffer space is limited making them unsuitable for sustained capture. They can, however provide flow statistics.

[12] Company network or a network where only provider provisioned devices are used can be exceptions.

3. Mirroring traffic in a network device. Routers and switches can support *port mirroring* or *monitoring port* functionality. A network interface can be configured to forward a copy of incoming and outgoing packets to a monitoring port. A device to capture traffic would be connected there. Mirroring includes many limitations that are discussed in detail in Port mirroring.
4. Capturing traffic on link. An device that is transparent to routers is installed on link. On optical links signal spitter can be used to direct part of light to a monitoring device. On electrical lines there is need for an amplifier that allows multiple devices to be on link. On radio links just antenna is sufficient. These methods will be discussed in Wiretapping chapter.

When traffic is mirrored or wiretapped, the traffic must received by some device for real-time analysis or storage. At low bit rates, a normal computer with a matching network interface card is just fine for the purpose. With higher bit rates and especially on high packet rates performance becomes an issue.

> **?:** A residential network provider has an option to put monitoring device on each consumer link or alternatively on links aggregating these to core routers. Consider benefits of either option.

When a packet is captured, there are lots of information associated with it.

Firstly, it includes metadata that is derived by the device that captured packets. The most important one is typically the *arrival time* of the packet, i.e. the time when the packet was captured. Another information relates for the *interface* it was captured from if system can do multi-interface capture. Also size of packet on wire and if it had a correct checksum are interesting related for analysis.

The packet itself includes also multiple components. There are multiple protocol layers that can be identified. At first there are link layer headers, like Ethernet frame headers. Following are network layer headers, most likely IP or IPv6. On top of that are transport layer headers like TCP and UDP followed by application data.

Up to transport layer, each packet is self-contained. It can be identified to be part of TCP connection using ports 80 and 47821 between hosts `192.0.2.10` and `192.0.2.12` with link layer headers identifying layer 2 sending and receiving host. On application level, a single packet does not always contain sufficient information to identify application or to make more detailed analysis.

There are application protocols, like domain name system (DNS) where each packet is self-contained in default UDP-mode. Also real-time transfer protocol (RTP) can be considered as a packet based protocols. However, to able to decode RTP packets, one may need to capture control protocol packets like

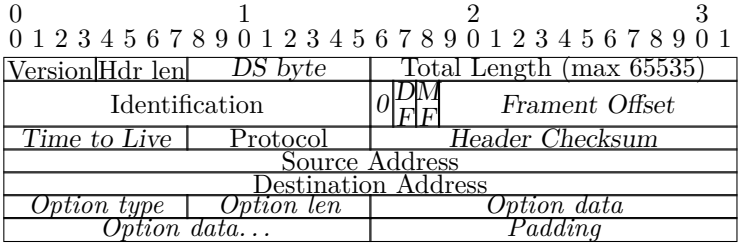

Figure 6: IP version 4 header

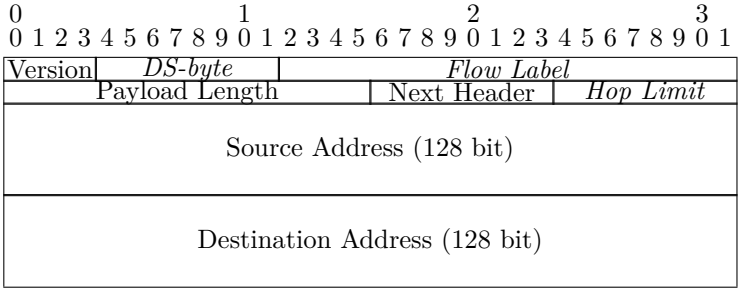

Figure 7: IP version 6 header

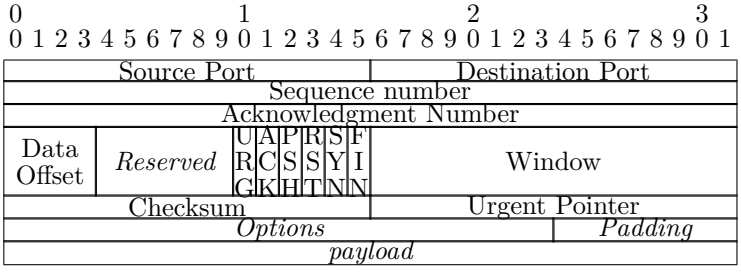

Figure 8: TCP header

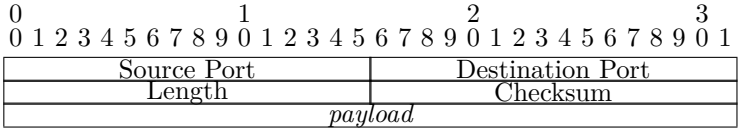

Figure 9: UDP header

session initiation protocol[13] (SIP) and extract session description to find out voice encoding, for example. Unless it is one with permanent payload types.

Most protocols running on top of TCP do not have any consideration on packet boundaries. Even relatively small data element may span multiple segments. To be able to decode these traffic, full packet capture is needed and one must maintain state and contents of surrounding segments.

Take a hypertext transfer protocol as an example: after a TCP connection has been established, the first packet toward server includes HTTP request. The following packet from the server includes HTTP header but following packets do not carry any indication what protocol is used. They only contain data stream of objects transferred. Furthermore, the request and response may not fit into one packet but may be split into two or more.

> **!:** Monitor web traffic. Observe sizes of HTTP request and response headers. Can you identify HTTP/1.0, HTTP/1.1, and HTTP/2 requests and responses?

Packet data can be stored to disk for post-processing and analysis or it can be analysed real-time. The first alternative requires large disk capacity with good write performance, while the latter requires good processing capacity in real-time.

4.4 Packet data analysis

Analysing packet traces can be divided into three principal groups. The most basic one is to include only those analysis where state information is limited and small. Even quite low-performance network device could collect this information. For example, following IP protocol statistics are easy to collect with small tables of counters:

- **IP length distribution**. Either using individual values or histogram bins like 0-32, 33-64, 65-128, . . . , 1024-2048.
- **IP protocols** observed. Eight-bit value, so 256-element table is sufficient.
- **TCP, UDP and STCP port numbers**. If source and destination port numbers are tabulated separately then two 65536 element tables are needed per protocol. If source-destination port pairs are recorded, then over 4 billion (2^{32}) elements are needed per protocol.

The second group would be those analysis that have much larger and sparse value space but still there is no need to keep information about individual flows. Examples of this kind of analysis include:

[13] That use different UDP port numbers and SIP messages may even have different source and destination IP addresses than the media stream.

- Time between two subsequent packets i.e. **packet inter-arrival** time (IAT).[14]
- **IP addresses** of communicating hosts. While a large server can store data structure for each possible IPv4 address (2^{32}) but not for every possible IPv6 address (2^{128}), in practise number of hosts communicating towards one network is much smaller. In core network many addresses are seen over time.

The next step is keep track on individual flows. At this stage there is most likely a need to employ some time based garbage collection to reclaim memory space from information that is not active anymore.

- **Flow or TCP connection logging**. Each connection (5-tuple as discussed in Flows in IP Networks) or flow will consume amount of memory to keep track. One will use timeouts to clear old and stale entries.
- TCP **connection analysis** for congestion windows.

The fourth analysis group requires storing all or part of payload data of transport protocols. Protocol data units must be defragmented to correctly parse values. Once part of header or payload data is fully analysed, data can be discarded. This type of analysis include:

- Analysis on **application-layer protocols** like HTTP, SMTP
- Analysis on **user data**, i.e. deep packet inspection (DPI)

With encrypted connections becoming more prevalent, possibilities of application protocol analysis and DPI have less value.

> **?:** How one could estimate needed memory capacity in each group of analysis? Are those dependent on traffic volume, number of network (stub) users, or both?

4.5 Flow measurements

Collecting and storing packet data quite often requires additional hardware to implement. Many routers implement flow information export, IPFIX protocol [23], or similar vendor defined protocol like NetFlow or J-Flow. This allows collecting flow data without introducing additional elements on network data path. Only data collector needs to be installed.

The flow information reporter sends data to collector. A flow record includes typically information common to all packets in flow: IP source and destination addresses, DSCP values, transport layer protocol, and transport layer ports. Information related to routing such as input and output interfaces and AS routing

[14] Strictly speaking, packet inter-arrival can be anything from minimal packet length divided by line rate to infinity.

information can also be included. In addition, there are flow meta information such as flow start and end time, number of bytes and packets in a flow.

> **?:** What would be approximate difference in data volume generated by packet capture and by flow information collection?

It is important to note that statistics collection is a secondary function for a router. In case router is overloaded, it may drop flow reporting to keep forwarding packets. Also it is possible that it reports only part of flows, not all.

4.6 Measurement sampling

Data volumes in both packet and flow-based measurement may be too large to reasonable process them. An obvious question is, can we reduce data by maybe taking only one out of ten packets or flows and still get reasonable good results for our analysis.

How packets or flows are then selected will have an effect on for which analysis we can use data for. Some common selection processes are following:[24]

- Count-based sampling: every N packet one packet is selected or every N packet M packets are selected.
- Time-based sampling: in every interval of length T all packets that arrive withing time t are captured.
- Random sampling selecting n out of N packets. For example, a vector of length N will have n random positions marked. For a sequence of N packets, those packets are selected whose sequence matches in vector.
- Random sampling with uniform probability p. For each packet a random selection is made resulting desired probability.
- Random sampling with weighted probability. Some packets are selected always (i.e. important packets) or with higher probability than rest of packets that are selected based on uniform probability p.
- Filter selection of packets. This is a deterministic selection. Packets may be selected based on protocol fields in packet like source or destination addresses, or protocol numbers. Selection may be based also on events: for example if the packet is rejected by router access list, fails reverse path check or destination AS number matches for rule.
- Hash-based selection allows selecting quite random but still same packets at different observation points. This is mainly used for trajectory sampling.

Effect of sampling strategies to analysis is discussed more detail in analysis part of the course.

> **?:** Does some sampling strategy work badly / exceptionally for some kind of application or protocol?

4.6.1 Trajectory sampling

If we want to observe for example packet delay or packet loss probability between two points in the network we can set up two measurement points and capture all packets at both locations. Then we compare traces and identify packets that were seen or should have seen in both locations. This will result lots of data and if they are different locations, packet data must be transferred between sites.

Some sampling could clearly be used. However, selecting every N^{th} packet or a packet randomly with probability p will not provide results we want. In random selection, if p=0.01, only every 100th *sampled* packet will match for the packet at remote end and we can use only 1/10,000 of packets. Situation is even worse if we have more measurement locations.

Clearly we must use more deterministic selection process. If it is some known test traffic we want to monitor, we could just put the right 5-tuple filter in place and receive only this test traffic. If we want to select our traffic more randomly, we can use hash-based filtering. We calculate hash value from non-volatile packet headers and possibly from payload and use it as an selection criteria.

An IP packet is not unmodified while it travels over network. In version 4, at least *time-to-live* (TTL) value is decreased by each router resulting also new IP header checksum to be calculated. Version 6 does not have header checksum but *hop limit* is reduced. In addition, differentiated services code points (DSCP) and early congestion notification (ECN) may be changed over life-time of packet. While IPv6 non-zero flow label should not be modified, in certain situations it can. However, withing one security domain it should stay intact [25]. Transport protocol headers or payload is not modified.[15] Our hash function must exclude those volatile fields. Quite often these are set to defined value, like zero.

If we use only fields that do not change between different packets of the same connection, then we are able to capture all packets that are part of same flow. Depending on analysis, this may be desirable. From security perspective, the hash function should be such that malicious players cannot game it to either overload analysis or evading sampling. At minimum the hash seed value and selection value may not be known by outsiders.

> **?:** Could trajectory sampling work if only payload would be included into hash calculation.

A hash function is a method to generate fixed length value of arbitrary long data with low probability two messages producing the same value. Cryptographic message digest functions like SHA [26] make it highly difficult to generate two messages producing same digest value but those algorithms are computationally expensive. For many purposes cyclic redundancy check (CRC) family functions provide sufficient separation of different messages if no active attacks are expected.

[15] In principle. Some service providers modify e.g. HTTP traffic by injecting advertisements.

> **!:** There exists command line tools like `crc32`, `md5sum`, `sha1sum`, `sha224sum`, and others. Run those on some large file and compare time spent. Command `time` may be helpful.

If algorithm is implemented on hardware (ASIC or FPGA), CRC functions are very simple to implement (shift register and XOR gates) compared to cryptographically secure hash algorithms.

4.7 Analysis based on passive measurements

The location of the observation point – where a packet traffic is captured – is located, does have an effect on how the results should be interpreted. If the point is at the same host as one of end systems, we can assume to have almost the same view as this end. At high load situations some packets may be lost either by the capture process or by the application. The delays may also be slightly different.

If we want to observe a larger network, one can find points in a network where we can observe all traffic towards one end of communication. Many server networks do have redundant paths up to close to end systems: in these cases two measurement points are needed. Wired local area network most likely has low delay and loss compared to Internet connections or wireless networks. Of course, it is not unheard for a datacenter LAN to have serious problems with packet loss or capacity.

In any case, we need to take into account any possible events for a packet may encounter between our observation point and the destination. One cannot assume that a incoming packet seen at datacenter border router will be delivered to a destination server or packet seen at home cable modem will be delivered over WLAN to a user device.

> **!:** Send ICMP Echo requests (`ping`) to WiFi access points (default gateway) when you are connected to network. Let it run for several hours. Are there any lost packets? How much delay variation there exists?

4.7.1 Throughput measurements

Finding lower bound for maximum speed a host can send traffic is quite easy from packet traces. By aggregating traffic from a source it is possible to calculate total throughput. In the smallest timescales some buffering may result too high rates, but on timescales exceeding seconds this should be reasonably good.

Of course, the key question here is if the host is sending with maximum rate at any point of time or is the traffic limited by the application or host itself

or by some part of the network. Using packet traces it is possible to estimate congestion window at sender. If the congestion window estimate is larger than actual amount of data in flight, then transmission is limited by sender.

We can use that analysis to extract select only those flows that are not limited by hosts or applications. Another possibility is to identify periods of bulk transfer. For example a HTTP connection can be left idle after transfer has taken place to wait for additional requests from the same host. Taking average over whole connection life-time will result lower measured throughput.

If we have only processed flow data from IPFIX collector and not packets, the analysis is a bit more complicated. We cannot identify from the flow data if there has been any lulls or idle periods where host has not been sending data actively: it does not have data to send or the recipient is not willing to receive data.

Different applications utilise network capacity differently. For example video streaming over HTTP often has high bitrate at start to fill playout buffer but later reduces to rate close to actual video bit rate. In this way client buffering is reasonable and there is enough data in buffer for most of network slowdowns. The bittorrent application will control its connected peers by throttling those connected peers whose chunks are not preferred ones. TCP connection stays open but not traffic is exchanged for a period of time.

> **!:** Start packet capture on your computer. Visit several video streaming sites (YouTube, Vimeo, YLE Areena, Netflix...) and watch some videos, preferable over 15 minutes. Make note of start and end times for each. Stop packet capture and identify video streams: most likely these are the largest TCP streams. Observe if there are any differences how network capacity is used over time of video.

4.7.2 Delay measurements

Any protocol that has request-response type with identifiable packets can be suitable for delay measurements. A prime example is TCP 3-way handshake. Using time difference between SYN and SYN+ACK segments we know round-trip time to the server from the observation poit. Time between SYN+ACK and first ACK-only segment is round-trip time to the client.[16]

Other useful messages include those used in active measurements. Unless they are encrypted, one can correlate ICMP Echo Requests and Responses, OWAMP and TWAMP messages, and DNS queries among other. Again, one must consider possible end system delays. Like with DNS queries: can the server respond without any additional delay.

[16] We use client and server in typical settings: the server makes passive open i.e. listens for incoming connections.

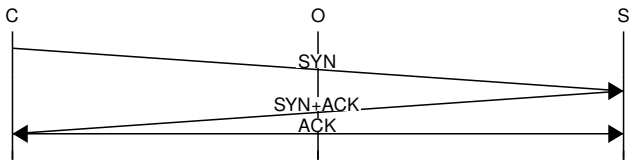

Figure 10: Delay difference at Observer compared to Client or Server

?: What other protocols would be good for delay estimates?

We cannot observe individual packets from flow data. However, in case of unidirectional flow data, we can map both halves of connections and compare start times. This could give in case of TCP connections the round-trip time towards server. One must however consider the timestamp accuracy of device collecting flow data.

4.7.3 Loss measurements

Similarly to delay measurements, we can also use request-response messages for loss measurements. If we see a request without corresponding response, there are four different cases:

1. Request was lost in transit to destination.
2. Destination did not reply, or was not reachable at that point.
3. Response was lost before we saw it.
4. Response was sent via another path.

Furthermore, it is also possible that response will get lost between the observation point and the destination.

?: Would the network type (i.e. fixed, mobile, wireless) the client is connecte have an impact on conclusions? How one could take into account.

If the protocol uses sequence numbers, like *ICMP Echo requests*, RTP or IPSec, we can see if any packets are missing between two packets. Also if there is any reordering, it can be identified.

TCP connections can also be used to estimate packet loss ratio without keeping full TCP state data. There are few cases that can be used.

- If we observe sequence numbers in a TCP connection to increase by approximately maximum segment size without any holes, but then see some earlier segment number. This indicate some segment was lost after we observed it.

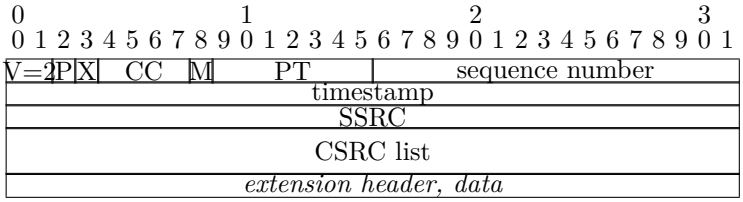

Figure 11: RTP header

- If there are some sequence number missing but then we observer some of earlier sequence number. The segment was lost before we saw it and we observe retransmission. An alternative for maintaining full TCP state is to just monitor sequence number and if we see segment with earlier sequence number, then at least one segment is lost.

> **?:** What kind of factors at end systems would affect on above estimate?

Flow data is little use for loss analysis.

4.7.4 Real-time transport protocol measurements

RTP is a protocol run top of UDP[17] to transport time-critical information such as voice or video between sender and receiver. It also can be used with multicast having multiple receivers and senders. It provides sequence numbering and time synchronisation that is very useful in analysing network quality.

One cannot identify RTP packet from just one packet because it does not use well-defined port numbers but ones that are randomly allocated by an application. The best way is to monitor signalling protocols like SIP, H.323, or Jingle and extract IP addresses and port numbers there. With SIP, session description protocol (SDP) contains port and media information. Also it is possible to guess RTP packets by monitoring several packets to check if they would form an RTP session: sequence number and timestamp would be increasing while other fields stay same.

> **?:** Look on RTP header. Can you define a packet to be RTP one based on single packet? If not, how many packets are needed?

Sequence numbers can be used to find out if there are any lost or duplicate packets. Delay and delay variation can be measured too if timestamp rate is

[17] Other transport protocols like STCP, DCCP and TCP are also possible, but rarely used.

known i.e. by SDP or guessing. Many voice codec use timestamp rate of 8000/s.

RTP is accompanied with real-time transport control protocol (RTCP) that helps with synchronising different senders and provides quality reports. The *sender reports* include NTP-RTP timestamp mapping and information how many packets and bytes the sender has sent.

Receiver reports include percentage[18] of packet lost, cumulative count of lost packets, highest sequence number received, time when last sender report was received and jitter value calculated with formula below:

$$J(i) = J(i-1) + (|D(i-1,i)| - J(i-1))/16 \tag{1}$$

$$D(i,j) = (R_j - R_i) - (S_j - S_i) = (R_j - S_j) - (R_i - S_i) \tag{2}$$

RTP performance analysis can be used both estimating [voice quality][Voice quality analysis] and overall network quality.

> **?:** What kind of heuristics one could use to identify RTP streams in the network?

4.7.5 Encrypted connections

IPSec encrypted data can be monitored using IPSec sequence numbers. We can see if there are some lost packets.

> **?:** Look on headers of IPSec header. What fields can be used for analysing connection quality?

It is also possible to identify applications and situations used inside VPN connection by observing packet lengths, packet interarrival times and changes in bandwidth consumed. In high security environments *Traffic Flow Confidentiality* (TFC) mechanisms are often used.

4.7.6 Anomaly detection

An important area related for security is to identify any abnormal changes in network traffic. Most Network Intruision Detection Systems (NIDS) are signature based, i.e. they match for known malicious traffic. Anomaly detection may be efficient for unknown attacks. However, the challenge is know the baseline and normal variations.

Furthermore For example, [27] includes survey of various facets of network anomaly detection.

[18] Actually, how many 1/256th are lost.

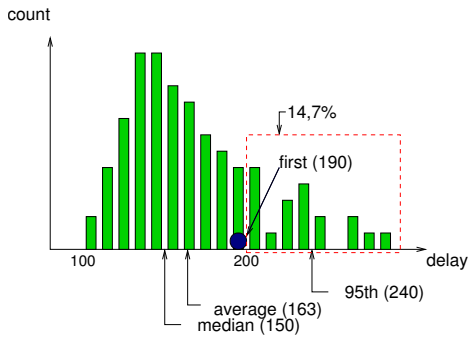

Figure 12: Delay distributions by Y.1541

5 Concluding results

So far we have made measurements using active, passive, or both methods. We have some numerical results using analysis on the measurements. Now the question: is the network any good for our applications?

5.1 Characterising delay distribution

In typical network that is not heavily congested, delay distribution is left-sided. There is a minimum delay that has no queuing delay but just processing, serialisation, and propagation delays. A majority of packets are close to this minimum value. There are often some packets that are delayed very much.

If a mean delay is selected as an indicator, the long tail of delays will increase average delay much higher than it really is perceived by users. The most relevant definition of delay depends on application and transport protocol used.

For example ITU-T Y.1541 [28] defines following ways to define delay variation or delay distribution:

1. For each packet, variation is difference between then delay of the *first* packet and current packet delay.
2. For each packet, variation is difference between *mean* delay of population and current packet delay.
3. Select acceptable delay interval in advance and count the *proportion* of packets that fall outside this interval.
4. Distance between two *quantiles* like 0.95 and 0.5.

For example from the figure above following values can be identified:

1. First packet delay is 190 ms, so variation for each packet is $|d_i - 190|$ ms.
2. Mean delay is 163 ms, variation is $|d_i - 163|$
3. If acceptable delay interval is 100-200 ms[19] then 14.7 % fall off this region.

[19]The shortest delay is more than 100 ms, so lower bound could be zero as well.

4. Median delay is 150 ms and 95^{th} percentile is 240 ms so variation is 90 ms.

The third criteria would be suitable for real-time communications if upper bound is set for playout delay. First two are suitable to measure instantaneous delay variation while the fourth could be a good one for generic use.

> **?:** Consider different types of applications. What delay definition would be the most suitable?

5.2 How to report quality measures to the users

Providing 86,400 measurement values for each day and each connection is not very useful or easy for the user to understand. Situation is harder if one needs to compare different providers. On the other hand, a single number may be too little compared for needs. Consider, for example, a journalist working for TV station. There are two needs for her: high capacity link to upload video segments and low-delay, low-jitter for real-time report in news broadcast. A single network may not provide both features.

A good approach is to select few key indicators and provide a single number for each of those:

1. **Median delay** for each packets. If packet is lost then its delay is set to inf.
2. **Loss ratio** with wait time of 2 seconds.
3. **Delay spread** as difference between 25^{th} and 75^{th} percentiles.
4. **Duplicate packets** within 2 seconds.
5. **Reordering** as fraction of packets received later than following packets.

When coupled with nominal or measured throughput figures, these provide quality for the network. Smaller values are better.

> **?:** What key metric is missing? Why it is not part of above list?

> **?:** You are looking for a network connection to your home and there are several providers (with different technologies) and they provide estimation of quality metrics above. How you would make decision if there are none above all others in all metrics?

5.3 User sessions

One can tell difference between humans and machines by one criteria: machines are more deterministic. In network traffic this is implied by two-level process: most user actions are randomly distributed and can often modelled as Poisson

process or similar.

For example, the time when a user decides to make a call is an example of random process. After the call has established, the process at packet level follows deterministic process when packets are sent with fixed intervals, typically every 20 milliseconds. If there is a silence suppression, this will result periods of no packets whose interval and length are randomly distributed. The call length is also a random.

> **?:** Consider the difference in participating to a conference call and a normal two-party call. How the traffic pattern would be different?

Similarly, a decision to visit a web page by clicking on a link has random timing. Following HTTP requests are deterministic but may depend on the past browsing history as some objects are cached on user device. Those TCP connections that carry HTTP requests produce packet flow that is quite similar in many different network conditions. The next user action resulting network traffic also happens at random point depending when user has completed reading the page or finds the next link to click to.

One must, however, consider situations where user actions are triggered by some external event. In these cases user arrivals are not randomly distributed but there is a high burst of incoming users. These events are called as *flash mobs* or *slashdot effects* where large number of users are trying to reach service at the same time. Examples include tele-voting in TV programs and ticket sales for popular events.

> **!:** One of challenges in network security is to differentiate between flood of genuine users and distributed denial of service (DDoS) attack.

> **?:** Consider user actions when trying to reach a (suddenly) popular web site overhelming its capacity. How you could identify a situation where many requests fail from a) network traces or b) application logs.

5.4 What are the end points and use case?

Quite often, a network user can trade between throughput, delay, and packet loss. However, the use case will have an effect on how much there is room for trade.

Considering traffic sources, we can identify three different cases:

- *Interacting* situation where both (or multiple) endpoints produce traffic and changing turns. For example in voice communication one-way delay of 150 ms is considered as the limit for a good quality. If the one-way delay exceeds 0.5 seconds it becomes hard to maintain conversation.

  When multiple senses are combined, then delay offsets or inter-media synchronisation must be quite strict. For example in virtual reality setting even difference of order 10 ms between visual image and motion can be disturbing causing motion sickness.

- *Live* broadcast has more relaxed delay requirements and allows for more playout buffer. Playout buffers are used to compensate for possible packet loss or temporary changes in bandwidth. The delay live-to-receiver is a sum of encoding delay, network delay and playout buffer delay. The playout buffer should be large enough to handle all delay variations.

  In principle, the playout buffer can be as large as wanted. It is just not nice if a video broadcast you are watching shows a goal only after Twitter feed. Or worse, shouts of *"Goooooal"* are heard from neighbour.

- *Stored* media, for example a movie, does not have any specific real-time requirements. It is more like file transmission. However, there are few requirements that depend on architecture. First of all, the movie should start as soon as possible after user has acted on. At the beginning a higher throughput is needed to fill in the playout buffer. After that if user performs any actions like pausing or seeking these should happen within a reasonable time. With case of IPTV, channel change time is one critical criteria.

?: What would be a tolerable delay for various type of live event streaming:

1. Football match
2. Tennis game
3. Golf tournament
4. Live concert
5. Lecture having both on-site and remote participants. Discussion via Twitter wall.
6. Prime minister press event announcing establishing top research unit.
7. President declaring war against a rogue nation.
8. Publicly traded company announcing acquisition of their major rival.

It also depends on the receiver what kind of errors are an issue. A human watching or listening is mostly sensitive on delay changes. Transitory errors are not problem if they are not too frequent and there is sufficient error-free time

between them. In voice communication distortion free interval should be at least 1.3 seconds [29].

If the receiver is a recorder, then delay or retransmissions are not a problem. An error-free transmission is preferred. If the sending party has capacity to locally store all of transmission, non-realtime transfer with TCP may be the best option.

> **?:** Consider a case of TV series. User has an option to either watch it streaming, download each episode manually and watch when download completed, or have a subscription were new episode is automatically downloaded when published. How this has impact on network requirements?

5.5 Interactive use

Non-media applications need also reasonable response times from the usability perspective. Usability studies has shown requirement for **0.1/1/10** seconds rule: [30]

- Immediate response withing **0.1** seconds. User feels that the action she made (mouse click, key press) was registered by the the computer and some visual or audible feedback is provided if even if the task was not fully completed.
- The task initiated should complete within **1** second. This keeps the user workflow seamless even if the user recognises the delay compared to instantaneous action. The user feels she is still controlling the device and not waiting for it.
- If the task could not be completed within 1 second, some feedback should be provided to the user about how long wait is estimated before it is completed. If the task is not completed within **10** seconds, the user easily starts thinking or doing other tasks and it is harder to come back on the original task.

There are some discussion if increased use of networked services has changed these limits. Even as low values as 4 seconds are provided as critical abandon time for web pages. One such study was sponsored by a company providing front end web caches, so that may be taken by grain of salt.

For good usability from network side, low latency and sufficient available bandwidth are needed.

> **?:** Compare HTTP load time for 10 kibibytes document on two different networks:

1. 128 kbit/s, 10 ms round-trip-time
2. 1 Mbit/s, 150 ms RTT

5.6 Quality needs by application

What network capacity is then sufficient for an application? For media (voice, video) streams it is quite simple to determine as each quality and compression level and technology requires certain minimum rate.

5.6.1 Media applications

Voice communication can be served in different quality levels: narrowband codecs that provide *telephone quality* with 300-3400 Hz frequency range. Wideband codecs provide better speech fidelity with 8-16 kHz frequency range. Fullband codecs cover full human audible spectrum and are suitable also for music.

Frequency range for telephone circuits was determined at time of analog circuits to be sufficient for voice communication. When transition to digital was made, 8 bits sampled 8000 times a second was considered sufficient using companding algorithm (A or μ-law) to carry 3.1 kHz frequency range. This resulted G.711 codec that has bit rate 64 kbit/s. As it is uncompressed algorithm it can tolerate quite high bit error rates (10^{-2}).

Voice codecs to provide approximately same quality are commonly used. These have bit rates on 8-16 kbit/s and typically can tolerate 1-4 % packet loss rate. There are even lower bandwidth codecs providing rates of 2 kbit/s and below, typically for military use.

As IP communication like VoIP is not limited by traditional PCM hierarchy, it can provide better quality voice and audio. Codecs optimised voice communication have typically lower encoding latency compared to ones intended for file distribution. While "CD quality" audio takes about 1.4 Mbit/s uncompressed, 128-384 kbit/s is typically sufficient for the same quality as compressed.

!: A short review of voice codec quality with different codecs can be found from http://opus-codec.org/comparison/

Video codecs provide even higher reductions between uncompressed and compressed video stream. A high definition video (resolution 1920x1080 at 60 frames a second) has uncompressed rate 2 Gbit/s but can be delivered less than 10 Mbit/s stream. Normal standard definition TV broadcast is 166 Mbit/s uncompressed and 1.2 Mbit/s as compressed. Video compression algorithms continue to evolve thanks to increased processing power and allow higher resolutions without significant increase in bit rates.

A trivial implementation of video transmission may lead into very bursty traffic. Key frames can be more than 10 times larger than intermediate frames. If this is not considered, it may lead into situation where packet loss affects on key frames and results much worse subjective quality than packet loss suggests.

> **?:** Most audio and video codecs can operate on either as constant bit rate (CBR) or variable bit rate (VBR) modes. For which use-cases the CBR is more suitable? What are the benefits of VBR?

5.6.2 Non-media applications

A web browser is the common platform for accessing network services. Its usage and requirements should follow the 0.1/1/10 seconds rule. One significant change has been the increase of web page sizes in recent years. While in 1995 the average web page was less than 15 KiB, in 2012 the average exceeded 1 MiB. In year 2015 average was more than 2 MiB.

Network throughput must be approximately 20 Mbit/s to load 2 MiB page in less than one second, depending on latency. On the other hand, it is quite common the sites to be application-like: not all of page is loaded every time user makes an action but background actions are taking place.

> **?:** How long it takes to load 15 KiB page with 33.4 kbit/s modem? How about 2 MiB page with 2 Mbit/s mobile connection.

Of course, it is not only the network that is responsible from the user experience and delay between user action and response. One must account also for both server processing (e.g. database searches and program logic) and the client processing (parsing and decoding received data and rendering it to the user).

Then there are also applications that can perform operations on background. For example sending email could be performed on background. Also if there is a large file to download, user is prepared to wait for it and have it perform on background while doing other tasks. This applies both client-server and peer-to-peer applications. This kind of applications are mainly dependent on throughput. The lower limit of throughput depends on time user wants to wait for.

> **?:** How much feedback user must be given that there is outstanding operation? In what kind of situations this can be important?

Also operations between server processes often have timeouts. These are in part defined to protect resources at end hosts. If the other party malfunctions,

the other party can clear session after timeout. For example, SMTP has per-command timeout 2-10 minutes and session timeout of one hour. DNS uses 5-second timeout with exponential back-off to one minute. Without timeouts, processes may get stuck indefinitely because of some error state.

> **?:** With timeouts a drawback maybe the premature abortion of long-running task that just takes longer time than anticipated in system design and configuration. What kind of timeout and watchdog mechanisms can be implemented?

6 Practical aspects of measurements

Measurement by itself is quite simple: just send some test traffic or capture packets from the network. However, there are many small details to be taken into account while performing measurements. Some of these originate from the very nature of networks that is distributed and highly parallel, others are implementation artefacts.

6.1 Time synchronisation

To make useful measurements systems should have reasonably accurate clocks. It is much easier to combine results from different hosts if they agree on time reasonably well. For many purposes wrist clock accuracy (i.e. seconds) is just fine considering time zones are taken into account.

> **?:** Systems can keep the time with UTC and represent a local time to user calculating it based on timezone information. Another alternative is to use the local time as system time. What benefits either option has?

However, many measurements are easier to implement if each host has a very accurate clock. However, this is not typically a case and errors can found be the same magnitude as the measurement results.

We can define following properties for time keeping:

- **Accuracy** is information how close to the current universal time the clock is
- **Resolution** is the smallest measurement unit the clock gives reading. It is possible that a system gives reading with microsecond resolution but in reality the clock is updated maybe 100 or 1000 times per second. In the latter case the resolution is only one millisecond.
- **Skew** is an error in clock rate. Either compared to universal time or against some other clock in the system.

- **Offset** is difference of two clocks or time references. Because of skew, this will change over time.

> **?:** Consider different measurements. In what kind of measurements offset does not matter if skew is not significant? What kind of problems it causes if skew is alternating compared to relatively stable one?

Most ICT equipment have their time-keeping based on crystal oscillators (XO). Typically there is a low-resolution battery-backed real-time clock to keep time when powered off. When system is running, time is kept based on crystal oscillator providing also other clock signals. While XOs are quite stable, they are dependent on operating temperature:

$$f = f_0 \left[1 - 0.04ppm(T - T_0)^2\right] \tag{3}$$

Typically the error is around 10^{-7}/Kelvin. One way to stabilise XO is to keep it in controlled temperature (typically around +75°C) and in that way performance around 10^{-9}/day can be reached with oven-controlled crystal oscillator (OCXO).

> **?:** Why computer do not commonly have OCXO to provide time information?

6.1.1 Network time synchronisation

The common method to synchronise clocks in Internet is to use Network Time Protocol (NTP) [31]. It builds a multi-tree-like structure having high stratum clocks synchronised to time references at each root. In unloaded network one can reach accuracy of few milliseconds.

The key uncertainty in NTP are the queuing delays. For example, time a packet was sent at the host is not exactly know as are not known the delays in routers. To overcome these uncertainties, more detailed control is needed.

> **?:** If you have Linux computer with NTP installed, you can use `ntpq -p` command to print out the peers you computer is synchronising to and various statistics. What Stratum values you find?

Precision Time Protocol (PTP, IEEE 1588) is designed to Ethernet networks. It has master-slave structure. After a synchronisation message is sent, it is followed by another message including the information of real time the message was sent. For a good accuracy, every devices in path must support PTP. Typically,

microsecond is a typical accuracy that can be reached while in some environments even 100 ns is reachable.

> **?:** How network technology affects on synchronisation protocols? Can one use PTP over WLAN? How about other radio technologies?

With PTP one uncertainty is still difference in forward and return path. A multi-fiber cable each core has typically slightly different length and when cables are joined together additional differences are created. A variation of PTPv2 developed in CERN, WhiteRabbit, runs over single fiber optical link using different wavelengths for each direction. This guarantees they are the same length and thus the delay is the same in both directions. A correctly set up system can reach accuracy of few nanoseconds. Light travels 30 cm in one nanosecond in air, about 20 cm in fiber.

6.1.2 Out-of-band synchronisation

Global Navigation Satellite Systems (GNSS) are based on triangulation using accurate time measurement for satellites. This information can be used for time synchronisation too. A GPS disciplined clock can provide accuracy of order tens of nanoseconds. These have short-term stability based on crystal oscillator that is tuned with GPS or CDMA clock source.

Most NTP and PTP servers use GNSS time receiver as reference clock and have OCXO for short-term stability and backup. Modern mobile networks depend on good synchronisation among base stations. This information can be used for time synchronisation too.

The remaining challenge is to interface this time information (typically 1 pulse-per-second PPS output and 10 MHz reference) to ICT device. Some of those have direct inputs but for most devices some add-on card is needed. Not all GPS devices provide accurate time synchronisation output and are not suitable for time keeping. They do keep long-term time, but can have error upto one second.

Traditionally serial port signals were used to signal epoach of second, but this is not very accurate. Furthermore, in modern computers there is no “real” serial ports. An interface card with appropriate trigger and counter logic is needed. In these cases, also latency resulting internal system bus access must be taken into account.

> **?:** What challenges there are using satellite-based time syncronisation compared to using network-based?

6.2 Packet capture

Packets travelling over network must be copied to monitoring system to be available for analysis. There are several steps, starting with creating a copy of packet.

6.2.1 Port mirroring

Most modern wired networks use only point-to-point links and thus a computer connected to wired network can only see its own, multicast, and broadcast traffic.[20] To capture traffic from point-to-point links one can use monitoring ("mirror", "span") ports or VLANs on network switches and routers. Network interface card of monitoring device is then connected to monitoring port.

There are many implementation dependent issues to consider on port mirroring:

- If two-way traffic in a port is mirrored to one port, traffic is lost if combined (in+out) port utilisation exceeds one-way line rate.
- Switch internal capacity may be limiting number of monitoring sessions.
- Not all packets may be mirrored. For example, some devices do not mirror packets generated by themselves like ICMP Time Exceeded messages or routing protocol messages.
- VLAN information may not be correctly recorded.

Some switches also support remote mirroring. A port or VLAN can forward traffic to a port on another switch over VLAN. This allows some switch-internal traffic to be collected to central location.

> **?:** Find from Internet the documentation of some network switches. Study how port mirroring can be configured. What kind of limitations there exists related e.g. for number of sessions, direction of traffic, VLANs, packets generated by switch and capacity? What differences you can find?

6.2.2 Wiretapping

Optical links are easy to monitor using optical splitter that diverts some part of light to a second output. This causes some signal attenuation on path. For example 70/30 splitter allows 70 % of light to travel on original path with 2.5 dB attenuation. At monitoring port attenuation is 6.3 dB. If monitoring happens on remote end of long link there may not enough signal strength for monitoring even if the actual link still works. There exists multiple split rates, the most common being 90/10, 80/20, 70/30, and 50/50. These work typically on limited wavelength range; device for 1310 nm does not work at 1550 nm. As passive optical devices they are very reliable and do not cause issues after installation.

[20] There are ways to get around it, but they are more focus of security course.

Complexity in copper wire networks monitoring depends on if there is echo cancelling used or not. Ethernet links at 100 Mbit/s and below do not use echo cancelling and using just suitable amplifier to provide strong signal for the monitoring device without attenuation for the link. On gigabit Ethernet as well in ADSL the device installed must efficiently establish two links (one up, one down) that makes the device more complicated and dependent on power supply.

> **?:** In wired networks monitoring device is typically installed between the device and cross-connect or between two devices with normal connectors. Is it possible to wiretap copper cable without causing disconnect? How about optical cable?

While radio networks are easy to listen in principle, MIMO environments cause similar issues as echo cancelling in wired links.

Many devices that are installed on-line like Ethernet demarcation devices may also provide monitoring function. Multiport monitoring devices may include either optical switch to select the link to monitor or then may have some aggregation function: this will have similar challenges as switch monitor port.

6.2.3 High-speed capture

Traffic analysis has three time-critical tasks:

1. Packet capture
2. Extracting interesting information from data flow
3. Writing this information to non-volatile storage

The packet capture is very time-critical, link rate of 1 Gbit/s corresponds to 1 MB in 10 ms that is on region on disk access latency. By default buffer size in Linux is 2 MB if the application does not specify other value.

> **?:** What is the impact for analysis if some packets are lost? Would packet loss be correlated to some events in the network?

One possible approach is to utilise purpose-build network capture cards from companies like Emulex or Napatech. These cards have on-board processor and memory to take care of time-critical steps of packet capture. These provide an API to access packet data in efficient manner.

If a normal network interface card is used for capture, then the processor easily becomes a bottleneck. It is not able to process packets fast enough as only one core of CPU can process packets at each time. Fortunately high-end server interface cards (10 Gbit/s) provide many queues. A packet is forwarded to a queue for example based on source and destination IP addresses using selection similar to hash-based sampling. Each queue is served by one processor core.

Linux kernel has support for multi-threaded packet capture using `PACKET_FANOUT` from version 3.1 (Oct 2011). Unfortunately support on user-level libraries (e.g. libpcap) and tools is so far thin.

Recent libpcap versions (1.5.3 or later from end of 2013) support Linux `TPACKET_V3` version that provide better throughput by more flexible buffer implementation. Again, to use this application program must request it through library.

Also Data Plane Development Kit (DPDK) and other user space networking frameworks are an interesting approach. When packet processing is done in large scale blocks, for example timing accuracy of individual packet may suffer. Depending on type of analysis wanted, this may be a problem or not.

> **?:** What kind of accuracy one can expect if a captured packet is timestamped in
>
> 1. network interface card,
> 2. operating system kernel (NIC driver), or
> 3. application program.

> **?:** What kind of impact errors in packet time stamps have on analysis? Are there difference how tolerant different analysis are?

7 How to use measurements to test systems and networks

7.1 Testing devices

When one is starting to procurement for network devices, one need to define how to select correct one. If there is just one figure, like "throughput 100 Gbit/s" it is not known how this information is measured. For example, was it with maximum packet size or with some reduced feature set?

There is clearly need for standard set of parameters that can be used in evaluating different systems. The RFC 2544 [6] is a test for this purpose.

The device under test must be configured for *normal use*, i.e. following normal best practices concerning routing and other settings. There should not be any *performance test optimisation* or *test detection*.

> **?:** What kind of optimisation could be performed for a router for example?

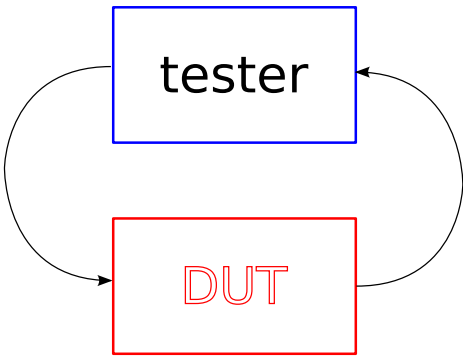

Figure 13: Device under test (DUT) configuration

Figure 14: Network under test

Test is run with variable frame sizes (64, 128, 256, 512, 1024, 1280, 1518 bytes for Ethernet) and burst sizes (16, 64, 256, 1024 frames). If the receiver receives any non-test frames these are not counted for total volume. The report includes:

- dropped frames,
- number of subsequent frame losses,
- out-of-order frames, and
- duplicate frames.

In many cases RFC 2544 is run with zero dropped frames. This is a problem for software-based routers as they may drop some random packets even on quite low levels of load. In these cases some low percentage of loss is allowed to get practical throughput figure.

Tests can be run also as bidirectional and multiport tests. These are intended to test internal queuing performance by sending same destination sequence from each port.

> **?:** Why sending same destination sequence (A, B, C, ...) from each port would test queue performance?

The RFC 2544 principle can also be used to test network. The packet structure is device-dependent but in most cases the packets carry sufficient information to measure delays and packet losses and identify any bit errors.

7.2 Testing networks

While the RFC 2544 can be used for service activation tests (SAT) by network providers, it is not optimal for multi-class service provisioning. First of all, it uses only one traffic class and no different treatment for different traffic classes are studied. Even if tests are run for each traffic class, those are tested separately and no information on interactions between classes are learnt. It has the focus on throughput and no delay information is reported. Full test cycle on RFC 2544 can take hours to complete.

The ITU Y.1564 is developed to overcome these limitations. [32] There are three traffic levels:

- **Committed Information Rate** (CIR) where the traffic must meet those performance objectives defined in [SLA].
- **Excess Information Rate** (EIR) where traffic exceeding CIR is provided best-effort treatment
- **Overshoot rate** that is rejected by the service.

At CIR level the **Key Performance Indicators** (KPI) defined include the frame delay, delay variation and loss ratio. Quite often there is a defined loss ratio for the service acceptance criteria. If the loss ratio is higher that that, test is considered as failure.

> **?:** If service activation test would be "birth certificate" for a connection, what kind of periodic checkups are needed? How they should be tested or monitored?

7.3 Testing services

It is not uncommon that a new service when introduced with fanfare still fails under load when it is opened for the public and many interested people want to use it. For that reason it is important to test service capacity both when introduced and also regularly when any modifications are made.

> **?:** Some examples of failed deployments of new services? Was the reason for those reported afterwards?

For a web service, it is not sufficient just to load the main page but the interaction on the site should correspond to how real users would navigate and use the site. This includes having accounts or other state information per emulated user. In that way the full system is tested. If tests are done with one user only, the user database won't receive any real load, for example.

One challenge is providing this real data to the service. While just copying existing production database to test environment would provide exact data

model for testing, this can have serious issues related for user privacy. For this reason, test database must be artificially generated in many cases.

> **?:** The company you work for is acquiring new web service to replace existing site. There has been peak 10,000 daily visitors on the old site while there are 150,000 customers in database. For how many visitors you would stress-test the site.

The service should degrade gracefully, possibly to switch low-fidelity mode if possible. In some cases security of service is compromised because of malfunctioning components allowing unauthenticated users to access data or user is presented with other people's data.

There are many commercial tools and many free tools available for testing. Free tools include jmeter, ab, and siege.

With dynamic page updates just loading main page URLs is not enough. One needs to account background communication too. Browser developer tools are helpful in providing view of resources the browser requests over network.

> **?:** Open developer tools within browser. Open network tab on dev tools. Visit a web page and wait it to load. Right click on list and select "Copy All as HAR" and paste to editor or "Save All as HAR". The HAR file is JSON format data that includes all requests made by broser. How many different servers contribute to that web page?

7.3.1 DDoS testing

Denial of Service (DoS) is *the prevention of authorised access to a system resource or the delaying of system operations and functions* [33]. If there are a large number of hosts attacking (or intermediate) then it is Distributed Denial of Service (DDoS).

The denial of service can be accomplished with multiple ways:

1. Sending large number of traffic so that network links or network devices are overwhelmed and no useful amount of legitimate traffic can pass.
2. Protocol properties may result that target system must keep lot of state information and the resources are exhausted. A classic example is a TCP SYN flood that may be effective even today.
3. Implementations may have vulnerabilities. A malformed data may result either application to crash or consume large amount of processing. This is similar to the previous one but only affects some implementations.

4. Some operations may be expensive to perform. The attacker may select input data that it exploits worst case performance.
5. Attack may target also to infrastructure or supporting functions: power supply, cooling. Also attacks on routing can be considered infrastructure attacks in may cases.

Protocol and implementation vulnerabilities can be searched using various tools, including fuzzing tools. Typically traffic load in these cases is not high - unless a bug related for high-load situation is expected. Those include various race conditions.

One can try to protect from DoS with multiple ways which efficiency depends on type of attack. If attack comes from single or from few sources, it is easy to filter out. In distributed case it is difficult to filter only malicious traffic without impacting useful traffic. Quite often traffic profile in attack traffic is different from normal traffic and this could be utilised in clearing traffic.

A test setup depends on the attack type the system is designed to be protecting from. In typical case there would be a normal load traffic testing analysing system performance under attack and another system generating attack traffic. In an ideal case normal traffic would not deteriorate at all while the system is under attack.

7.4 Service Level Agreements

The **Service Level Agreements** (SLA) are between customer and service provider. The service provider can be an external organisation, but also another department within organisation. With agreements, the parties have common understanding about services, priorities, responsibilities, guarantees and warranties.

Service levels are defined using **key performance indicators** (KPI). These are defined as target values, i.e. long-time averages and minimum values over a measurement period. If targets are not met, there are sanctions towards provider.

Metrics are monitored using SLA assurance tools that monitor KPIs and network metrics related to those. For example, a probe could send a packet every 0.1 second and monitor how many packets are lost, what is average delay and delay variation. Email delivery, DNS lookup, and web page requests can be metrics to be monitored.

> **?:** Who will perform the SLA measurements: customer, provider, or someone else?

Most of SLA measurements are implemented using active probes in network devices, separate measurement devices or as a software on client and server

computers. It is also possible to use passive analysis methods to collect statistics and build analysis on top of that.

7.4.1 Service (*plural services*)

Service is:

- An event in which an entity takes the responsibility that something desirable happens on the behalf of another entity.
- (economics) Action or work that is produced, then traded, bought or sold, then finally consumed.
- (computing) A function that is provided by one program or machine for another.
- The military.
- A set of dishes or utensils.
- (sports) The act of initially starting, or serving, the ball in play in tennis, volleyball, and other games.
- A religious rite or ritual.
- (law) The serving, or delivery, of a summons or writ.
- (public service) that which is provided by the government or its agents
- (religion) Doing something for someone else without thought of reward or payment.

8 Privacy and protection of personal data

After Snowden disclosed NSA monitoring, many started to care for their privacy more. Finland has also had case of telephone operator CEO using phone records to find out who was leaking information discussed in board meetings to the press in year 2000.

Different countries have different laws and e.g. relation between employer and employee or between business and customer vary widely. Something that may be mandatory in one country may be illegal in another. This text is written mostly from Finnish perspective but for many parts it should apply within EU.

The right of privacy and protection of personal data is not an absolute right. Like the free speech, those rights must be considered in relation to other fundamental rights should these be in conflict. An example would be aspects of national or Union security.

8.1 Identification information

The corner stone in privacy is personal data or identification data. This is data that can be linked to an individual or household with reasonable effort. If there are multiple personal data records that are easy to process, it is a personal data

file.[21]

Personal data must be collected only extent needed and used only for purpose that is *compatible with original purpose* it was collected for. Use of data for historical, scientific, or statistical purposes is permitted.

> **!:** Remember: if you did not collect data at the first place, you **cannot lose or disclose it later!**

Processing identification data is of course need for many purposes. For example, this is needed to realise services: router must look on IP addresses to forward packets. Information security services like firewalls and virus scanners need access to data. Resolving technical problems, fraud, or misuse[22] needs identification data. Charging is not possible without revealing the paying party. The other party (B-subscriber) is not needed.

Identification information can be used to improve technical implementation or services. This includes automated statistical analysis as long the final output does not contain any identifiable entities. This includes also scientific research.

Identification data handling has more strict rules if it happens on working place. This is to protect privacy of employee because of the imbalance in relationship between employer and employee.

To summarise how to handle identification data within network operator:

- Only when needed.
- Only as much as needed.
- Only those whose duties it belongs to.
- Handing information over only to those that have right.
- Must have an audit trail for two years for any data access to personal data.
- Professional discretion must be maintained.

This can be considered as *data minimisation*: reducing data to minimum sufficient.

> **?:** How about user or payload data? Is there a difference if encrypted connections are used or not?

8.2 Terminology related for data protection

Extract from GDPR

[21] Example: a row of mail boxes with names is not a personal data file because that information is not easy to retrieve. A phone book on the other hand is.

[22] Misuse must kind of that causes some direct costs or endangers information.

- **Personal data** means any information relating to an identified or identifiable natural person ('data subject'); an identifiable natural person is one who can be identified, directly or indirectly, in particular by reference to an identifier such as a name, an identification number, location data, an online identifier or to one or more factors specific to the physical, physiological, genetic, mental, economic, cultural or social identity of that natural person.
- **Processing** means any operation or set of operations which is performed on personal data or on sets of personal data, whether or not by automated means, such as collection, recording, organisation, structuring, storage, adaptation or alteration, retrieval, consultation, use, disclosure by transmission, dissemination or otherwise making available, alignment or combination, restriction, erasure or destruction.
- **Restriction of processing** means the marking of stored personal data with the aim of limiting their processing in the future.
- **Profiling** means any form of automated processing of personal data consisting of the use of personal data to evaluate certain personal aspects relating to a natural person, in particular to analyse or predict aspects concerning that natural person's performance at work, economic situation, health, personal preferences, interests, reliability, behaviour, location or movements.
- **Pseudonymisation** means the processing of personal data in such a manner that the personal data can no longer be attributed to a specific data subject without the use of additional information, provided that such additional information is kept separately and is subject to technical and organisational measures to ensure that the personal data are not attributed to an identified or identifiable natural person.
- **Filing system** means any structured set of personal data which are accessible according to specific criteria, whether centralised, decentralised or dispersed on a functional or geographical basis.
- **Controller** means the natural or legal person, public authority, agency or other body which, alone or jointly with others, determines the purposes and means of the processing of personal data.
- **Processor** means a natural or legal person, public authority, agency or other body which processes personal data on behalf of the controller.
- **Recipient** means a natural or legal person, public authority, agency or another body, to which the personal data are disclosed, whether a third party or not.
- **Third party** means a natural or legal person, public authority, agency or body other than the data subject, controller, processor and persons who, under the direct authority of the controller or processor, are authorised to process personal data.

- **Consent** of the data subject means any freely given, specific, informed and unambiguous indication of the data subject's wishes by which he or she, by a statement or by a clear affirmative action, signifies agreement to the processing of personal data relating to him or her.
- **Personal data breach** means a breach of security leading to the accidental or unlawful destruction, loss, alteration, unauthorised disclosure of, or access to, personal data transmitted, stored or otherwise processed.
- **Biometric data** means personal data resulting from specific technical processing relating to the physical, physiological or behavioural characteristics of a natural person, which allow or confirm the unique identification of that natural person, such as facial images or dactyloscopic data. Also consider **genetic data**.
- **Data concerning health** means personal data related to the physical or mental health of a natural person, including the provision of health care services, which reveal information about his or her health status.

> **?:** Network service provider monitors network and stores IPFIX information into database. Network misuse identification is performed by a contractor. Considering list above, what subjects exists in this case and in which role?

> **?:** Not all data that is sensitive is personal data. What other kind of sensitive data there exists?

8.3 Network measurements and identification information

IP addresses can be considered as a personal information. While in many cases those is randomly allocated and may change for a single device multiple times a day, there are many cases were it can identify a user or household.

> **?:** How often your devices change IP address? You can use `ip addr` command on Linux systems (including Android), `ipconfig` on Windows and `ifconfig` on MacOS. If you run *Nettitutka* authenticated, you can look on measurements afterwards. Each measurement includes both local and public IP address.

For many types of analysis actual identities are not needed. For some long-time analysis it may be useful to know that this user now was the same user yesterday. Having 1:1 mapping between users and identities is useful but IP addresses may not provide it. This on the other hand carries a risk of disclosure and the data must be handled as personal data.

The key is then to get rid of any identification information as soon as possible. How anonymisation should be performed depends on kind of analysis wanted. In many cases full anonymisation makes data as useless for analysis as fully removing IP addresses. In these cases, pseudonymisation can be used.

When data does not identify a person, it is not anymore identification data. However, it is important to remember that even if data considered to be anonymised, it may still be sensitive. In framework of GDPR also pseudonymous data must be considered also as personal data. Disclosure of pseudonymous data must be handled as if actual data was disclosed.

There are basically three different strategies to create pseudonymous identities from real IP addresses:

1. Random replacement. Each time a new IP address is found, a random IP address is allocated. In this method no topology information is saved: address `192.0.2.1` would map to `10.4.4.5` and `192.0.2.2` to `192.168.254.7`. This provides best possible protection of pseudonymous identifiers.
2. Prefix-preserving mapping by keeping the upper part of addresses (like 24 bits) unmodified and lower bits are masked. For example `192.0.2.1`→`192.0.2.75` and `192.0.2.2`→`192.0.2.43`. If there are only few users in network, this may result disclosure.
3. Prefix-preserving mapping by anonymising the upper bits. This make traces anonymous on organisational scale. If organisation (network, i.e. upper bits) is identified then it is possible individual users to be identified. For example `192.0.2.1`→`91.54.1.1` and `192.0.2.2`→`91.54.1.2`.

> **?:** Consider few different kind of analysis from network data. For each to these, is there a difference what of above strategies is used?

If pseudonymisation is performed once for each batch of data, there is no need to store data further. This will result that the address `192.0.2.1` would first map to `172.22.3.4` and on following run to `10.98.76.243` and so forth. If we want to keep same mapping between data processing runs, the lookup table or rules to generate values must be stored. This table is of course sensitive because it would allow reversing identifiers.

8.3.1 Attacks on pseudonymisation

There are many ways to attack on pseudonymisation. The key problem is that there are too few IPv4 addresses, too few networks and too few users. Exhaustive search is feasible.

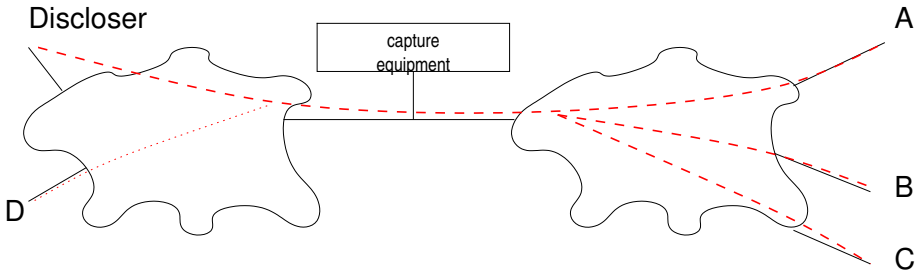

Figure 15: Attack on pseudonymisation

> **?:** How IPv6 would change situation if any?

Taken traffic of random IP address, it is usually not possible to determine whose traffic it is. However, it is often possible to answer opposite: is this IP address person X by knowing something *a priori* about traffic by X. Like what is her network usage pattern.

If very little is known about traffic but the IP address[23] of the user is known, one can create traffic that will help identifying user.

By sending packets by using a defined temporal process or other mechanism making it possible to later identify packets from anonymised trace, a discloser can learn it's own IP address and IP addresses of A, B, and C. In security terms this is a *covert channel* used to disclosure sensitive information.

Using suitable external information, pseudonymisation schemes can be broken. This information may be almost anything. For this reason, GDPR considers in many cases also pseudonymous data as identification data. An example how well one could identy a mobile phone based on the applications instaled in is presented in [34]. If you are interested about attacks on pseudoymisation, number of studies are cited by the publication above.

> **?:** What kind of information can be used to connect identifiers in the data collected from the network (like IP addresses, modified or original) to real persons or households?

8.4 Statistical and scientific use

With exception of troubleshooting and misuse investigation, one is not interested about particular single user. For most purposes measurement are performed for

[23] For example, if the user has visited discussion forum administered by the attacker or IP address is revealed with tracker image in email

two reasons:

1. identify any performance issues within network
2. study how users are using the network (including malicious usage).

For the first point the subject of study is not the user but the network. For the latter, users are the subject but as a larger group. When data is processed and groups are formed by several criteria, one must take care that groups are not too small. In publication of statistical data and there is a group with less than 20 persons, this information is not disclosed but combined with another group.

> **?:** Combining multiple criteria can be very efficient way to single out a single person. Consider information about you in different databases (area where you live, car owned, education, employment, family status, ...). How many group criteria (every criteria has more than 20 matches) are needed to identify you unanimously?

It is not always known in advance what scientific use the data may be good for when it is collected. The data subjects should be allowed to express their permission for the area of study. A person may have positive view on some area of science and negative for another.

8.5 Safeguarding data

Consequences of data breach can be significant. The GDPR allows upto 20 million euros[24] administrative fine assigned to a company that fails to protect personal information. This gives a good incentive for organisations to keep data secured. Furthermore, the company must notify *supervisory authority* latest 72 hours after incident has been identified. Also data subjects need to be notified in many cases.

Breaches cannot be avoided. Even major companies that audit other companies have failed as victims of data breaches, often resulting from poor oversight or cutting corners. However, efforts the organisation has put into protecting information and reducing damage caused by data leak would count as advantage for organisation.

Of course, there exists yet not legal cases against any company (as Directive entered into force in May 2018) were fine would be assigned. A decision regarding Whois data was made by a German court but there was no fine assigned. It can be assumed that penalty fee is limited in case organisation has followed and exceeded the industry standard safe-guards. Data protection certificates, seals and other marks can be used to demonstrate compliance to data subjects but also to supervisory authority.

[24] Or 4% of annual turnover if larger.

> **?:** Look for data protection seals and certificates. What kind of audits and tests they perform?

An organisation that has significant personal data processing, must appoint *data protection officer*. Her role is to provide guidance to *controller* or *processor* regarding to their obligations, monitor compliance, provide advice on data protection impact assessment and to co-operate with supervisory authority. She can be employed by the *controller* or by *processor* but position must be independent and able to report to high management without fear of retaliation. She cannot be fired for performing her duties.

Regulations include rules how personal data can be transferred outside of EU to a third nation. How easy it is depends very much of level of privacy protection in that nation.

> **?:** What kind of aspects would be related to subcontracting and administrating systems from non-EU nation?

9 References

[1] Q. Zhang, V. Liu, H. Zeng, and A. Krishnamurthy, "High-resolution measurement of data center microbursts," in *Proceedings of the 2017 Internet Measurement Conference*, 2017, pp. 78–85.

[2] R. Teixeira, A. Shaikh, T. Griffin, and J. Rexford, "Dynamics of hot-potato routing in IP networks," in *ACM SIGMETRICS Performance Evaluation Review*, 2004, vol. 32, pp. 307–319.

[3] J. Postel, "Internet Control Message Protocol." RFC Editor; RFC Editor; RFC 792 (Internet Standard); RFC Editor, Fremont, CA, USA, pp. 1–21, Sep-1981.

[4] S. Shalunov, B. Teitelbaum, A. Karp, J. Boote, and M. Zekauskas, "A One-way Active Measurement Protocol (OWAMP)." RFC Editor; RFC Editor; RFC 4656 (Proposed Standard); RFC Editor, Fremont, CA, USA, pp. 1–56, Sep-2006.

[5] K. Hedayat, R. Krzanowski, A. Morton, K. Yum, and J. Babiarz, "A Two-Way Active Measurement Protocol (TWAMP)." RFC Editor; RFC Editor; RFC 5357 (Proposed Standard); RFC Editor, Fremont, CA, USA, pp. 1–26, Oct-2008.

[6] S. Bradner and J. McQuaid, "Benchmarking Methodology for Network Interconnect Devices." RFC Editor; RFC Editor; RFC 2544 (Informational); RFC Editor, Fremont, CA, USA, pp. 1–31, Mar-1999.

[7] J. Mahdavi and V. Paxson, "IPPM Metrics for Measuring Connectivity." RFC

Editor; RFC Editor; RFC 2678 (Proposed Standard); RFC Editor, Fremont, CA, USA, pp. 1–10, Sep-1999.

[8] G. Almes, S. Kalidindi, and M. Zekauskas, "A One-way Delay Metric for IPPM." RFC Editor; RFC Editor; RFC 2679 (Proposed Standard); RFC Editor, Fremont, CA, USA, pp. 1–20, Sep-1999.

[9] G. Almes, S. Kalidindi, and M. Zekauskas, "A One-way Packet Loss Metric for IPPM." RFC Editor; RFC Editor; RFC 2680 (Proposed Standard); RFC Editor, Fremont, CA, USA, pp. 1–15, Sep-1999.

[10] G. Almes, S. Kalidindi, and M. Zekauskas, "A Round-trip Delay Metric for IPPM." RFC Editor; RFC Editor; RFC 2681 (Proposed Standard); RFC Editor, Fremont, CA, USA, pp. 1–20, Sep-1999.

[11] C. Demichelis and P. Chimento, "IP Packet Delay Variation Metric for IP Performance Metrics (IPPM)." RFC Editor; RFC Editor; RFC 3393 (Proposed Standard); RFC Editor, Fremont, CA, USA, pp. 1–21, Nov-2002.

[12] R. Koodli and R. Ravikanth, "One-way Loss Pattern Sample Metrics." RFC Editor; RFC Editor; RFC 3357 (Informational); RFC Editor, Fremont, CA, USA, pp. 1–15, Aug-2002.

[13] A. Morton, L. Ciavattone, G. Ramachandran, S. Shalunov, and J. Perser, "Packet Reordering Metrics." RFC Editor; RFC Editor; RFC 4737 (Proposed Standard); RFC Editor, Fremont, CA, USA, pp. 1–45, Nov-2006.

[14] H. Uijterwaal, "A One-Way Packet Duplication Metric." RFC Editor; RFC Editor; RFC 5560 (Proposed Standard); RFC Editor, Fremont, CA, USA, pp. 1–14, May-2009.

[15] M. Mathis and M. Allman, "A Framework for Defining Empirical Bulk Transfer Capacity Metrics." RFC Editor; RFC Editor; RFC 3148 (Informational); RFC Editor, Fremont, CA, USA, pp. 1–16, Jul-2001.

[16] V. Raisanen, G. Grotefeld, and A. Morton, "Network performance measurement with periodic streams." RFC Editor; RFC Editor; RFC 3432 (Proposed Standard); RFC Editor, Fremont, CA, USA, pp. 1–23, Nov-2002.

[17] P. Chimento and J. Ishac, "Defining Network Capacity." RFC Editor; RFC Editor; RFC 5136 (Informational); RFC Editor, Fremont, CA, USA, pp. 1–14, Feb-2008.

[18] R. L. Carter and M. E. Crovella, "Measuring bottleneck link speed in packet-switched networks," *Performance evaluation*, vol. 27, pp. 297–318, 1996.

[19] V. Jacobson, "Pathchar: How to Infer the Characteristics of Internet Paths." Lecture at Mathematical Sciences Research Institute, Apr-1997.

[20] 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, 2003.

[21] N. Hu and P. Steenkiste, "Evaluation and characterization of available bandwidth probing techniques," *IEEE journal on Selected Areas in Communications*, vol. 21, no. 6, pp. 879–894, 2003.

[22] E. Dahlman, B. Gudmundson, M. Nilsson, and A. Skold, "UMTS/IMT-2000 based on wideband CDMA," *IEEE Communications magazine*, vol. 36, no. 9, pp. 70–80, 1998.

[23] B. C. (Ed.), B. T. (Ed.), and P. Aitken, "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information." RFC Editor; RFC Editor; RFC 7011 (Internet Standard); RFC Editor, Fremont, CA, USA, pp. 1–76, Sep-2013.

[24] T. Zseby, M. Molina, N. Duffield, S. Niccolini, and F. Raspall, "Sampling and Filtering Techniques for IP Packet Selection." RFC Editor; RFC Editor; RFC 5475 (Proposed Standard); RFC Editor, Fremont, CA, USA, pp. 1–46, Mar-2009.

[25] S. Amante, B. Carpenter, S. Jiang, and J. Rajahalme, "IPv6 Flow Label Specification." RFC Editor; RFC Editor; RFC 6437 (Proposed Standard); RFC Editor, Fremont, CA, USA, pp. 1–15, Nov-2011.

[26] D. Eastlake 3rd and T. Hansen, "US Secure Hash Algorithms (SHA and SHA-based HMAC and HKDF)." RFC Editor; RFC Editor; RFC 6234 (Informational); RFC Editor, Fremont, CA, USA, pp. 1–127, May-2011.

[27] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "Network Anomaly Detection: Methods, Systems and Tools," *IEEE Communications Surveys Tutorials*, vol. 16, no. 1, pp. 303–336, 2014.

[28] "Network performance objectives for IP-Based services," International Telecommunication Union, ITU-T Recommendation Y.1541, 2002.

[29] P. Bardy, "A Statistical analysis of on-off patterns in 16 conversations," *The Bell System Technical Journal*, vol. 47, pp. 73–91, Jan. 1968.

[30] J. Nielsen, *Usability Engineering*. Boston: AP Professional, 1993.

[31] D. Mills, "Network Time Protocol (Version 3) Specification, Implementation and Analysis." RFC Editor; RFC Editor; RFC 1305 (Draft Standard); RFC Editor, Fremont, CA, USA, pp. 1–109, Mar-1992.

[32] "Ethernet service activation test methodology," International Telecommunication Union, ITU-T Recommendation Y-1564, 2011.

[33] R. Shirey, "Internet Security Glossary." RFC Editor; RFC Editor; RFC 2828 (Informational); RFC Editor, Fremont, CA, USA, pp. 1–212, May-2000.

[34] V. Sekara, E. Mones, and H. Jonsson, "Temporal Limits of Privacy in Human Behavior." 2018.