Traffic measurement in practice
Laboratory guide

Introduction

To devise, develop, and manage infocommunication networks, it is crucial to have an understanding of the traffic running on them. For different netwok applications it is also important to have an idea of their network usage. The amount and characteristics of network traffic depends on the network/application in question and can also change with time drastically. It is clear that internet service providers have to serve very different types and amounts of traffic compared to even a couple of years ago. Consequently, to get a realistic view of traffic on a network, we have to perform traffic measurements and analyze them appropriately. The goal of this measurement is to give an introduction to the tools that can be used to measurement and analysis of network traffic, and give an insight to the traffic patterns of different network applications, and their effect on network usage.

Network traffic measurements and characteristics

Network traffic measurements can be divided to two big groups: active and passive measurements. During active measurements we initiate some kind of traffic through the network and measure its characteristics. When using passive measurements, we passively listen on a link of the network and collect the main properties of traversing messages.

There are many types of active measurements from a simple ping, through measurement of application network data to measurements that include multiple network nodes. These measurements can be done using only endpoints of the network, but can also use internal nodes. Active measurements can be used to analyze the traffic patterns of a network application and to obtain end-to-end traffic characteristics (e.g. delay, jitter, observed QoS, etc.)

With passive measurements we can get a general view on the traffic of a network (link). Often examined charactersitics include link utilization and its change, protocol and port statistics, packet length distribution, flow length distribution, etc.

Statistical basics

To characterize network traffic we can use different statistics. If we think of this traffic as a sequence of packets, every propery of this sequence can be collected into a $\{v_i\}$ vector.

One of the simplest statistics of a $\{v_i\}$ vector of numbers is its *k*th moment, which can be calculated using the

$$m_k = \frac{1}{N}\sum_{i=1}^{N} v_i^k$$

formula, where N is the number of samples in the vector (the length of the vector). In many cases, the first moment (average) and the second moment are considered, using which we can calculate the variance/standard deviation of the data. Higher moments can also be used, but the higher the momentum, the less stable it is, in general.

An often examined property is the standard deviation of the samples, which characterizes the deviation of individual samples from the average. We can calculate this using the

$$\sigma = \sqrt{m_2 - m_1^2}$$

formula. The problem with this measure is that it depends on the order of magnitude of the samples (i.e., the chosen unit, e.g. kbps or Mbps), therefore it is often better to use the coefficient of variation, also known as relative standard deviation, which can be calculated as

$$c_v = \frac{\sigma}{m_1}.$$

So far we used single numerical parameters to characterize a set of samples. To give a more refined description of the distribution of the samples, we often use the empirical cumulative distribution function (eCDF) and empirical probability mass function (ePMF). The CDF of an X random variable is

$$F(x) = Pr(X \leq x),$$

its PMF is

$$f(x) = Pr(X = x).$$

In practice, these functions can be approximated with the eCDF and ePMF functions based our samples. The eCDF corresponding to the $\{v_i\}$ vector is

$$F_N(x) = \frac{1}{N}\sum_{i=1}^{N} I(v_i \leq x),$$

where $I(\alpha)$ is the indicator function, which is 1, if α is true, and 0 otherwise, and N is the sample size. Similarly we can get the ePMF using the

$$f_n(x) = \frac{1}{N}\sum_{i=1}^{N} I(v_i = x)$$

formula. In practice, a lot of sample sets have so many elements with slightly different values, that instead of the ePMF it is better to use the eCDF function or a histogram, which can be thought of as a „lower resolution” version the ePMF.

Wireshark quick guide

The main tool used during the laboratory exercise is Wireshark, one of the most popular packet analysers in the field. It can be used to observe and analyse packets traversing the network. It can analyse a variety of protocols and provides many useful functions including a basic statistical analysis module.

A detailed guide can be found on wireshark.org and many other websites provide guides and examples. The current chapter only presents some of the main functions used during the measurement.

Most functions can be reached from the upper menu bar. Traces can be saved using the Save option in the File menu. Different formats can be chosen, however, we only use the .pcapng format. To save the traffic sample in .csv format use the Export Packet Dissections in the File menu.

Another menu used during the measurement is the Statistics menu. We can examine the addresses resolved by Wireshark in the Resolved Addresses option. In the Protocol Hierarchy option we can see the number of packets classified by protocols. Under the Conversations option we can see the same but on the level of conversations between the endpoints. (E.g. under Ethernet based on the source and destination MAC address, under IP based on the source and destination IP address.) Under the I/O Graph option we can observe the change of traffic in time. Other options give more detailed statistics for different protocols.

The icons under the menubar provide quick access to some of the main functions. We only use the two leftmost ones which start and stop packet capture.

Below the icons there is a display filter which can be used to filter out packets to be shown in the window below. By default the Export Packet Dissections function only exports the filtered packets. (Filters can be set for packets to be captured as well, but this option will not be used.)

The rest of the application window is divided into three smaller windows. In the upmost window (Packet List window) we can see the packets filtered by the display filter and their main parameters (packet sequence number, source and destination IP address, protocol, packet length, other information). In the middle (Packet Details window) we can see details of the highlighted packet on different protocol levels. The third window will not be used during the measurement. This window shows the content of the highlighted packet in hexadecimal and ASCII representations. (In the latter the „.” character denotes non-printable characters.)

During the measurement we use different filters to create the lists of packets that are important for us. A useful function of Wireshark is that we can apply filtering by any parameter using right click and the Apply as Filter option.

Some simple filters:

ip.src == <ip addr>	filter packets with <ip addr> source address

ip.dst == <ip addr>	filter packets with <ip addr> destination address
ipv6.src == <ipv6 addr>, ipv6.dst == <ipv6 addr>	filter packets with <ipv6 addr> source/destination IPv6 address
tcp.port == <port>	filter tcp segments with <port> port number
<protocol>	filter packets that use the <protocol> protocol (e.g. tcp, udp, ip, stb)
!<cond 1>	filter packets that do not satisfy the <cond 1> condition
<cond 1> || <cond 2>	filter packets that satisfy at least one of <cond 1> and <cond 2>
<cond 1> && <cond 2>	filter packets that satisfy both <cond 1> and <cond 2>

With Wireshark, command line tools are also included, which provide functions that can be reached using the graphical interface of Wireshark. These can be useful when many and/or big capture files have to be processed, as they can be used in scripts and have lower resource requirements. During lab tasks it may be recommended to use tshark to export relevant information from .pcap files. The options of tshark can be accessed using the tshark -h command. Some useful options for the lab:

-r <file>	Specifying iving input file
-T fields -e <field 1> [-e <field 2>...]	Specifying output columns

To split capture files, a convenient tool is SplitCap (can be downloaded from https://www.netresec.com/?page=SplitCap). Using SplitCap, capture files can be split based on time, packet number, hosts, flows and sessions.