Experimentation: Measuring QoS parameters to determine media transport capabilities

Tatu Kilappa, Jaakko Salo

Keywords: streaming, QoS, network probing

Introduction

It can be desirable to be able to make conclusions on the feasibility of transferring desired media types over an available network link in realtime. However, the simple classification of the network interface rarely, if ever, offers sufficient information to determine its real-world capabilities.

Thus the objective of this experimentation is to determine whether or not a link is feasible for the transmission of certain distinct types of media. The results are rough classification of media transport capability instead of numeric representations of QoS values. All requirements are assumed realtime for use in e.g. medical or surveillance applications. The following media types are used in the experiment:

- Video stream, H264, HD quality, ~25Mbps
- Raw numerical measurement data, ~1Mbps
- Numerical measurement data packed using lossy compression, ~256kBps

Hypothesis

Using a few simple measurements on the network, it is possible to determine what kind of media can be transferred over the current link.

Description of methods

The expriment uses three distinct network probes to gather data. All of the probes require a network tester program running on both connection endpoints.

1. *Data collection.* The network tester software intiates several connections to the data recipient to gather data about the end-to-end link. They are described below in sections 1.1, to 1.3.

1.1. Collection of data transmission records using unreliable connection, active replies. The tester opens an UDP connection to the end host and sends a stream of maximum MTU size packets containing an index. The data contained in the packets is random – this will prevent any effective lossless compression from taking place in lower network layers. The entropy in real media formats is typically also very high, so this reflects a real situation well. The end host replies with an acknowledgement packet for every received packet. The initiator stores the timestamp and index of each sent packet and each acknowledgement. The test is repeated for several commonly used transmission speeds (64kbps, 256kbps, 1Mbps, 10Mbps, 100Mbps) and ran for 10 seconds.

1.2. Collection of data transmission records using unreliable connection, no replies. As in 1.1, but instead of replying, the server stores the timestamp of each received packet and when they stop arriving, opens a reliable connection to the initiating host and sends a full record.

Methods (cont.)

1.3. Collection of data transmission records using reliable connection. The tester opens a TCP connection to the end host and sends a stream of data. The end host records timestamps of every new read that can be done, along with the total amount of data currently received.

2. *Data processing.* The transmission records are Processed in Matlab to produce throughput (average bytes of data transferred in regard to time), packet loss (number of packets lost as a % of total transmitted packets) and jitter (variance of the time to next packet or next read segment).

3. *Data analysis.* Metrics produced by phase 4 are compared against the predefined requirements set by the media types presented earlier.

Focus

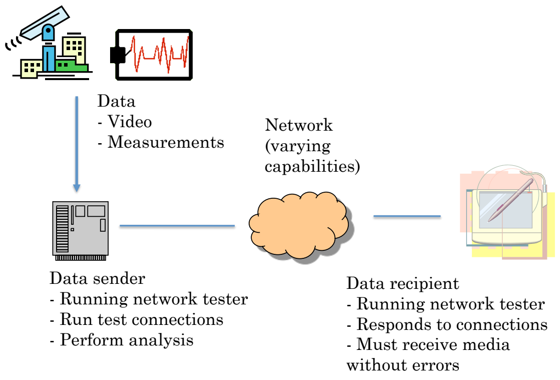
The distinct network types are not taken into account – only the end-to-end –capabilities.

Phase	Methods	Result
Data collection	1. Test connections to data recipient	Traffic records of packet index vs. timestamp or sent data vs. timestamp.
Data processing	2. Extracting data from the traffic records.	Throughput, packet loss rate and jitter for every connection
Data analysis	3. Matching of the processing results against the predefined media requirements	Information on fitness to transfer certain media types.

Table 1: Summary of methods

Experimentation setting

An essential part of the experimentation setting is the network in which the experimentation is conducted, ie. through which the traffic of the test software is passed. Four different networks were considered: a public ADSL ISP, public 3G ISP, laboratory Fast Ethernet switched network and laboratory 802.11g WLAN network. A small network testing software needs to be implemented for conducting the experiment. The network testing software and host configuration need to be carefully crafted so that they do not produce unintended delays affecting the test results, for example by bad program design so that the network stack is not powerfully utilized. A part of the software runs in two separate hosts in the selected network. For each network selected, the test is run individually, thus the results produced reflect the capabilities of that individual network.



Data
- Video
- Measurements

Network (varying capabilities)

Data sender
- Running network tester
- Run test connections
- Perform analysis

Data recipient
- Running network tester
- Responds to connections
- Must receive media without errors

Figure 1: Experimentation setting

T-106.6200 Special course in Software Techniques: Research and product development methods
Arranged by PM&RG research group

Email: pmrg@tkk.fi
WWW pages: http://www.cs.hut.fi/~pmrg
Telephone: +358 9 451 4807

Head of the research group: Mervi Ranta
Coordinator: Henrik J. Asplund