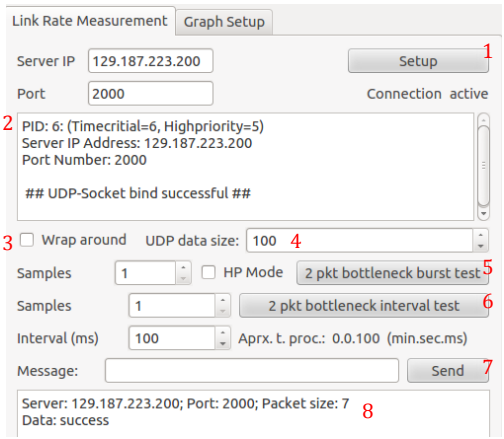
Link Rate Measurement Tool

The Link Rate Measurement Tool was programmed in Qt to make it platform independent and achieve a high data acquisition rate. Due to the fact that the data acquisition rate is crucial for the accuracy of the measurement and the importance of a fast processing system increases linear to the bitrate an implementation in Matlab was neglected. The Link Rate is determined by a high performance sending and receiving routine that blocks all operations within the program to achieve a result as accurate as possible. Therefore the thread of the Link Rate Measurement Tool was set to act as time critical to the operation system. Furthermore the precision of the measurement relies heavily on the performance of the machine and it is strongly recommended not to run any other programs in the background to reduce the number of threads handled by the system scheduler. To increase the precision the pending data flag of the input buffer hardware interface is constantly polled during the high performance transmission and reception routine. In addition independent sending and receiving functions, which are event driven were implemented. These are used by the system for normal not time critical transmissions. While the operating systems based on common Linux Kernels only offer system timers with a resolution of ms the timer provided by the Qt library offers ns resolution, whereby the ns are approximated on basis of instruction executions. If the source code is compiled for a Microsoft Windows operating system the high-resolution counter available in the OS will be used. Additionally the Link Rate Measurement Tool can act as a server. When the wrap around checkbox is set the data will be returned to the sender. Herby the system functions are optimized for large payloads, which leads to less good real time properties.
The software was completely programmed in an object-oriented manner to make it as portable and flexible as possible. While performing a bottleneck link rate measurement the dispersion as well as the link rate can be plotted in real time (except when operating in high performance mode). A real time plotting of the histogram was in advance of the measurement neglected due to resource constraints. The results of the measurement are displayed by two graphs, which appearance can be changed by the user in the Graph Setup tab.



User Interface Tab “Link Rate Measurement”:

1)	Pressing Setup configures a socket with the Server IP and the Port specified in the adjacent text edit windows. When a connection is successfully setup the connection will be active. Troubleshooting: If connection remains inactive after Setup, please close the program, check your network connection and open the program again. If you experience problems modifying the existing socket, please close the program and open it again, due to thread and kernel dependencies.
2)	The console displays the connection properties as well as the system status and determines if a transmission was in principal successful when sending messages with the build in chat client.
3)	When wrap around is set the program returns incoming data back to the sender.
4)	The UDP data has a variable size from 1 to 1500 Byte. In addition to the size of the UDP packet the 20 Byte IP header size and 8 Byte for the UDP header are taken under consideration, when calculating the bottleneck link rate.
5)	When pressing the “2 pkt bottleneck burst test” button the numbers of samples specified in the adjacent spin box are taken. In the burst mode two data packets are sent and after their reception the next sample is taken either directly or after the plots were actualized. To determine if the plots should be updated in between the “HP Mode” check box can be checked. If checked the plots won’t be updated until the last measurement is performed. The HP Mode stresses the network most.
6)	The “2 pkt bottleneck interval test performs the number of measurements defined in the adjacent spin box at the user defined time interval set in the spin box below. The time required for the measurement is approximated to improve the usability for the user. Long time measurements can monitor the network over a long time and provide additional information about temporal bottlenecks, caused by users that create strong load. At long time measurements the time interval should be set relatively coarse to reduce the computational effort for the system. Furthermore the time approximation won’t be reliable if the system is operating at its performance limits.
7)	By pressing the Send button the program will enter a chat client mode and messages can be sent to server and received from the server.
8)	In the result text field the output of the current measurement can be seen. Furthermore a brief summary of the measurement concerning average values is given out at the end of the measurement. In chat mode the memory is used to display the received data.

At the Graph Setup tab, several changes concerning the appearance and the diagram style can be applied to the plotted curves. Furthermore the update/refresh rate of a plot during a measurement can be changed to reduce the system load.
The resolution of the histogram can be changed by applying changes to the "Hist. res." spin box of the according graph. By default the resolution of the Bitrates is set to 1 Mbit/s steps. If the value is increased the equivalent number of digits after the decimal point increases the precision. Attention must be paid on large data sets when shifting the point, because the graph needs a significant amount of resources to dynamically plot the curve.
At the graph the range of the x and y range can be varied independently. Selecting the axis that shall be changed and using the scroll wheel of the mouse do this. When no axis is selected and the scroll wheel is used the graph will apply zoom operations not changing the ratio between the axes. The user can navigate at the graph performing a right click at the graph and moving the mouse into the desired direction.

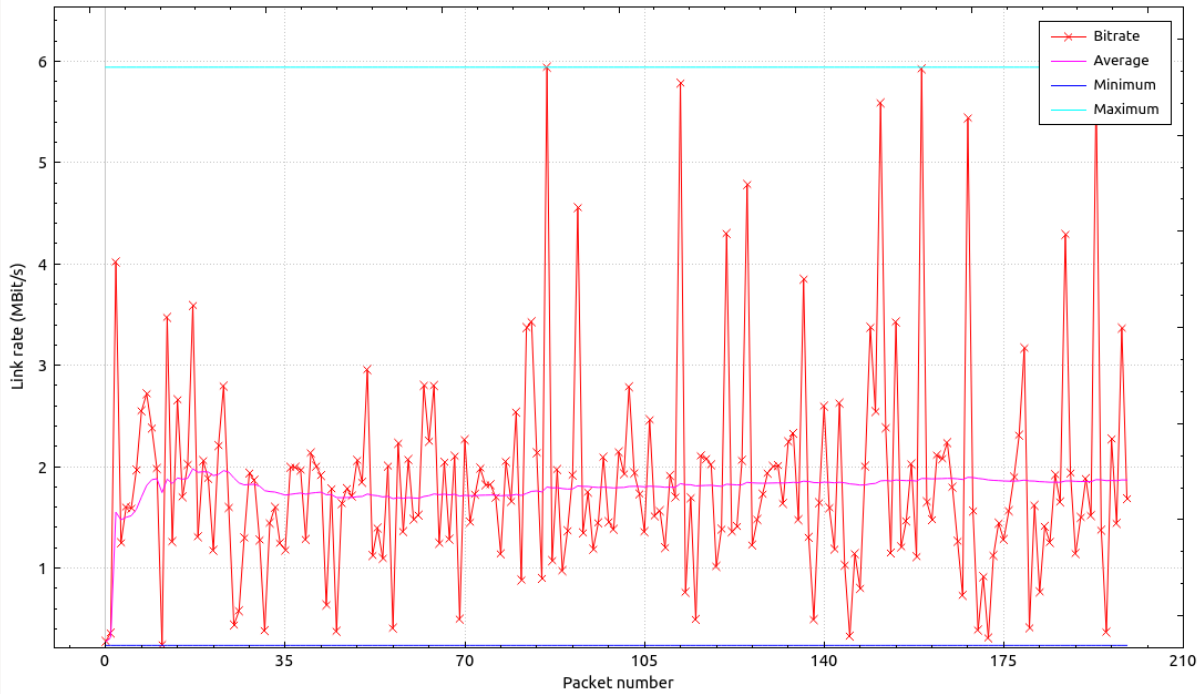
Remarks: It may happen that some measurements determine a Link Rate that physically can't be achieved by the network. This problem is related to the network hardware. High peaks can occur when for example one switch is busy and the first incoming packet is delayed for a certain amount of time because it is stored inside the shared buffer, most common to nowadays switch architectures, before it is forwarded. The second packet will therefore sometimes profit at the switch runtime, because the connection from the switch to the computer is already being served when it arrives. Therefore the time dispersion will vanish and the packets will arrive in sequence with no dispersion time difference. When this happens the minimum dispersion time is the sending time of one packet, which can cause very high peaks during the measurement. High peaks are especially measured when operating in burst mode. The probability increases with the number of packets sent, because this operational mode causes most stress to the network. The high peaks are not discarded, because the measurement was handled by the OS in a valid manner.
To prevent high peaks to occur during the measurement caused by the OS the network buffer is checked for a second packet when the first packet was read. If a second packet is already available in buffer, the program assumes, that the system was not able to meet the real time constraints required by the software and discards the current measurement.
Further problems are caused by the structure of modern computer networks. So different packets can take different path, what makes it more difficult to determine an exact bottleneck link rate.

Installation / Run Problems
If the program won't launch on the computer please make sure that the program has permission to apply changes to the computer libraries and links. You can grant the program permission to apply changes by executing the command: $chmod 777 Link_Rate_Measurement
At the first start-up it will be necessary to allow the program to apply changes to your system, when Qt libraries are not installed on your system yet.

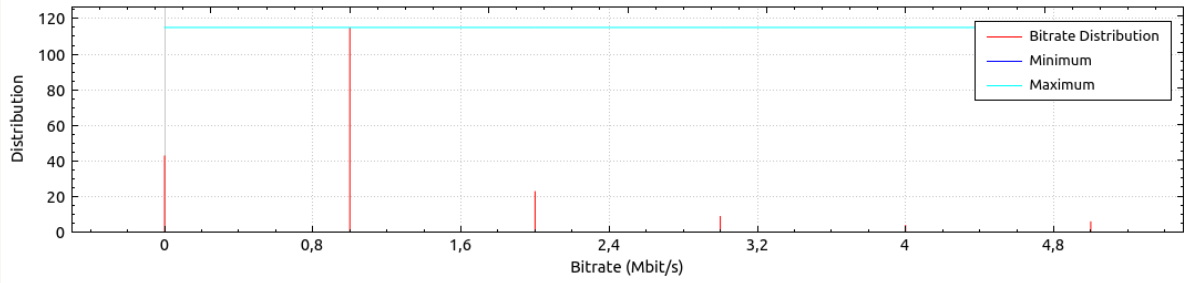
Link Rate Measurements

Mode	HP mode	Packet size	Samples
Burst	enabled	100	200

Graph 1

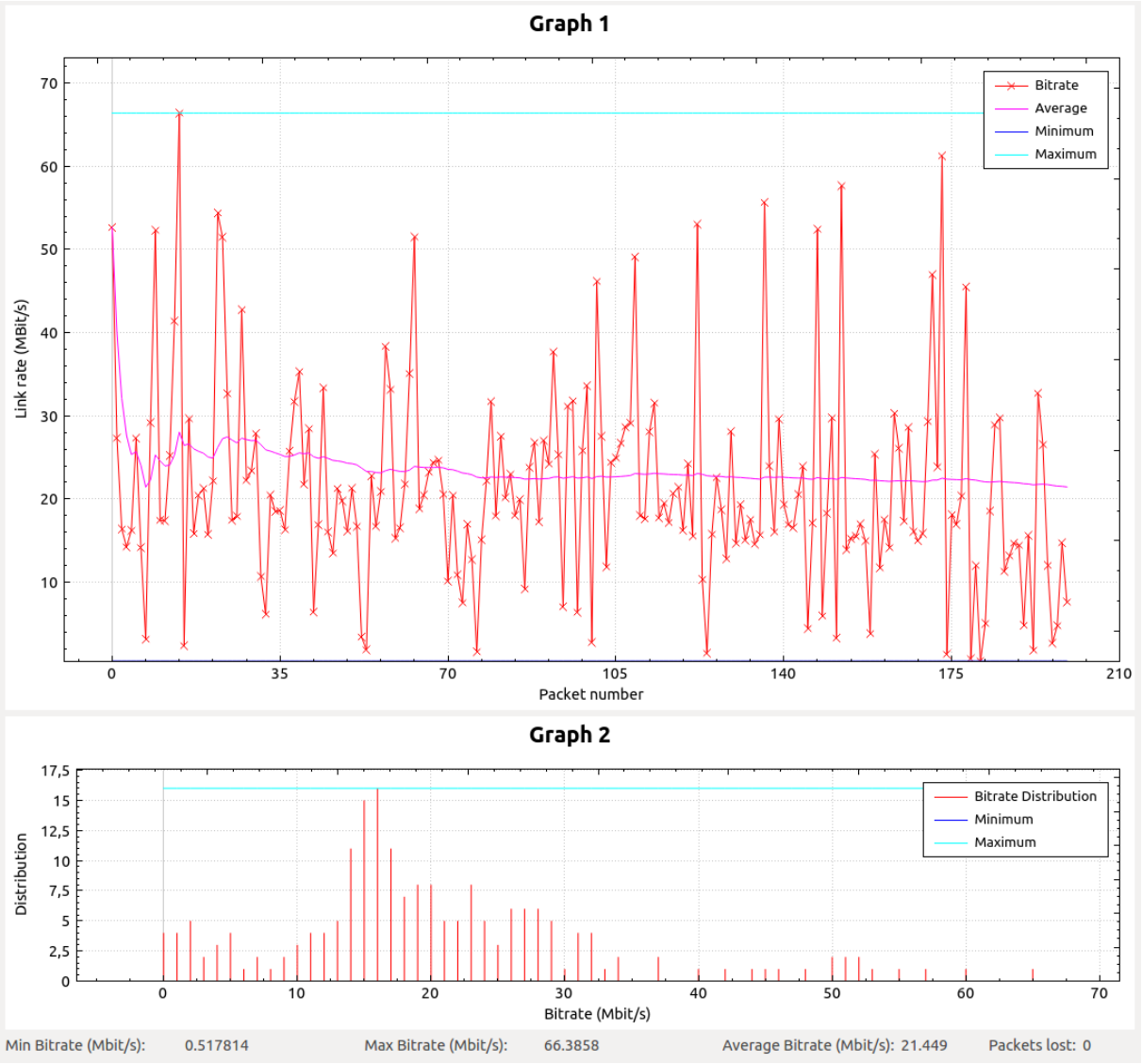


Graph 2



Min Bitrate (Mbit/s): 0.236543 Max Bitrate (Mbit/s): 5.94081 Average Bitrate (Mbit/s): 1.86824 Packets lost: 0

Mode	HP mode	Packet size	Samples
Burst	enabled	1500	200



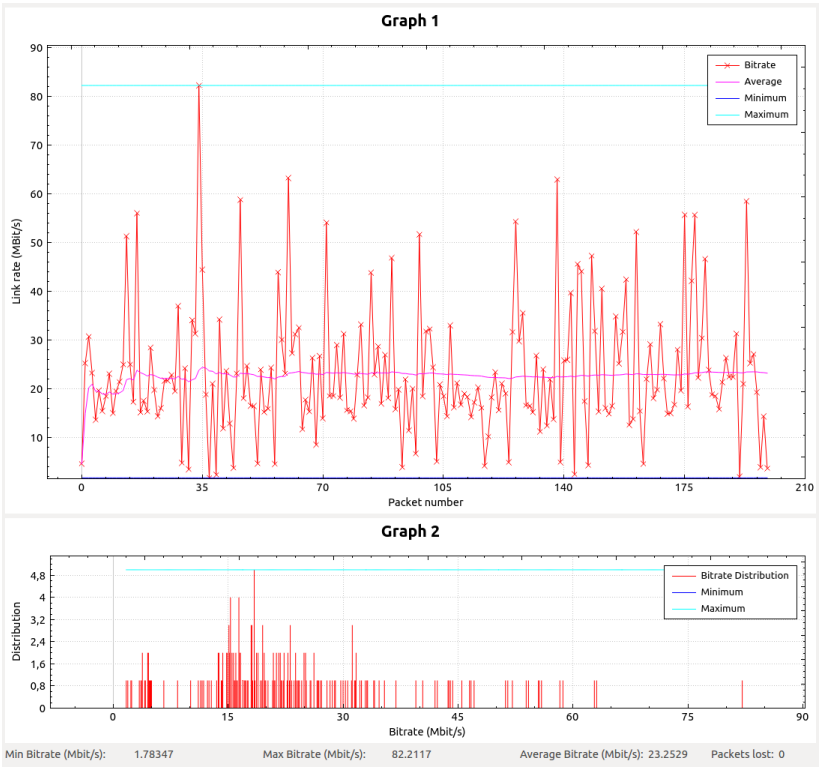
Small packets suffer a greater dispersion in relation to their total size than big packets.
The bit rate is not constant and fluctuates due to cross traffic and different packet routes. For the histogram we can see one dominant peak and a decreasing distribution of link rates moving from the dominant peak.

- The final bottleneck link rate for a packet with the data size of 100 bytes is 1.87 Mbit/s in average.
- The final bottleneck link rate for a packet with the data size of 1500 bytes is 21.45 Mbit/s in average.
The bottleneck link rate depends on the network and the network load. At long time measurements phases can be seen, where the bit rate decreases dramatically and remains on a very low constant level for some time. This is cause by extensive usage of the network by other users. Very high peaks at the measurement can be caused like previously described by the delay of two following packets by e.g. one switch. Furthermore the bit rate fluctuates stronger when using a wireless connection than when using a cable, due to distortions and the shared medium among several users.

In order to get the result as accurate as possible, what is the suitable packet size in your experiment?
A UDP packet that still fits into one IP packet gives the most accurate results. The program is limited to a UDP data size of 1500 Bytes.

Measurement with Graph resolution set to 1 position after decimal point.

Mode	HP mode	Packet size	Samples
Burst	enabled	1500	200



Dispersion seen in Graph 1 and related link rate shown in Graph 2.