

Troubleshooting

This section goes through some of the most typical special situations that the users can face when performing measurements with Qosium. Hints and solution proposals are given on how to proceed.

Table of Contents

- 1. Dealing with a Failure 3
- 2. Startup Problems 3
 - 2.1. Missing wpcap.dll (Windows) 3
 - 2.2. Qosium Probe Does Not Start 4
 - 2.3. License Activation Issue 4
- 3. Common Special Situations 5
 - 3.1. Negative Delay Values 5
 - 3.2. QoS Results Disappear 6
 - 3.3. No QoS Results 6
 - 3.4. Giving Commands Results in Timeout 7
 - 3.5. Correct Packet Filter but No Packets 8
 - 3.6. Pcap Drops Packets 8
 - 3.7. Excessive Qosium Control Traffic 8
 - 3.8. New Interface Not Appearing 9
 - 3.9. Missing Interface 9
 - 3.10. Mouse Issue with GNSS in Windows 10
- 4. Inconsistent Statistics 10
 - 4.1. The Statistic Sent Info Not Found Grows 10
 - 4.2. The Statistic Duplicate Packet Count Grows 11
 - 4.3. High Packet Losses 12
- 5. Known Limitations 13
 - 5.1. Performance Considerations 13
 - 5.1.1. Qosium Probe's Serving Capabilities 13
 - 5.1.2. Packet Rates 13
 - 5.1.3. Taking Full Captures and Collecting Packet-Level Results 13
 - 5.2. Network Protocol Support 13
 - 5.3. Large Send Offload and Large Receive Offload at NIC 14
 - 5.3.1. Support in General 14
 - 5.3.2. Turning Offloading Off 14
 - 5.3.2.1. Windows 14
 - 5.3.2.2. General Linux 14
 - 5.3.2.3. Making the Setting Persistent in Ubuntu 15
 - 5.3.2.4. Making the Setting Persistent in CentOS/RHEL 7 15
 - 5.3.2.5. Making the Setting Persistent in Most of the Linux Systems 15
 - 5.4. Measuring over Unreliable Network Paths 15
 - 5.5. On the Supported Network Interfaces 16
- 6. Glossary 17

1. Dealing with a Failure

Failing to get sensible measurement results is not a reason to feel ashamed. We do it too, on a rather regular basis. While we have worked hard to make Qosium as easy and intuitive to use, the concept of passive network measurement remains a challenging topic.

In practice, this means that eventually, you'll run across an error notification, or the received measurement results are not as expected. When this happens, the most important step is to take a deep breath and orient yourself into problem-solving mode. The solution might be just around the corner, but hasty actions can lead you further away from it.

To start with, Qosium has some inbuilt intelligence for detecting incorrect parameterization. However, some parameterization errors are still not (yet) detected. For example, selecting wrong interfaces is something that Qosium does not recognize as an error. Thus, it helps a lot if you are well familiar with the use of Qosium, and, in general, the philosophy of passive measurements.

If something peculiar is happening in the measurement, start by considering the following:

- What do I exactly want to measure?
- Does Qosium even support my needs?
- Have I built my measurement setup accordingly?
- Have I parameterized Qosium accordingly?
- Is it possible that the, in my opinion peculiar, results are, in fact, real?

Errors of measurement setup can often be detected by observing the measurement results. In particular, if the statistic [Sent Info Not Found](#) shows continuous non-zero values, it often is a sign of false parameterization. Also, [duplicated packets](#) can cause problems to the QoS calculation of Qosium. That is not necessarily related to false parameterization but to what you want to measure and how.

If it seems likely that Qosium is to blame, try first to disconnect the measurement controller and connect back. That reconstructs the whole measurement context and can help if something went wrong in the measurement process itself. Measurement controller restart can help in some rare cases, but Qosium Probe restart is hardly ever needed, but, of course, can be tried too. Reinstalling Qosium is not recommended since it can help only in some very rare license-related issues.

Try also studying both the log output of Scope and the console output of Probe. They can lead you to the source of the problem.

Next, we go through some common special situations. If you don't find relief to your problem in this section, contact Kaitotek's support.

2. Startup Problems

2.1. Missing wpcap.dll (Windows)



When launching Qosium (Scope or Probe), I get an error saying that wpcap.dll was not found.

Probable cause and solution

This error is given when a packet capturing library was not found. Thus, check that, and if needed, [install a packet capturing library](#).

Sometimes Qosium does not recognize your Npcap installation. This occurs if Npcap was installed without WinPcap compatibility mode. Thus, remove the Npcap installation and [install it properly](#).

2.2. Qosium Probe Does Not Start

Issue

When I try to launch Qosium Probe, it gives server startup failure (280) in the log screen.

Probable cause and solution

The got error means that Qosium Probe cannot set up its QMCP server. This happens if some other program has already booked the TCP port the Probe is trying to use. The most typical situation is that you already have Qosium Probe running in the machine. Thus, check it and check also that is Qosium Probe running already as a system service. If so, there is no need to start another instance of it.

Sometimes, especially in Linux-based systems, a premature termination of Probe execution may leave the socket up for a couple of minutes. This happens when the Probe is shut down by other means than CTRL-C while there are measurement controllers connected. If this was the case, try to start the Probe a bit later again.

If Qosium Probe is and was not running, it can be some other program using the same TCP port. Check from Qosium Probe's log screen which port it is trying to use. Then use, e.g., *netstat* to check if another program has booked the intended Probe port already. You can change Qosium Probe's port in the initialization file (*QosiumProbe.ini*).

2.3. License Activation Issue

Issue

I try to connect Qosium Probe with Qosium Scope, but it complains something about the connection and the license.

Probable cause and solution

It is very likely that Qosium Probe's license activation is tried, but it fails. License activation will be carried out online when the license has ended or when you are using the Probe for the first time, having a license with a limited number of installation rights.

Check first the internet connection in the device where the Probe is installed. The problem can also be that there is a strict firewall in your organization blocking outgoing connections of Qosium Probe. You can contact Kaitotek support to discuss the details of the required license server connection. Please note that the license server access is required only during activation. During normal operation, no Internet connection is required.

You can also check if Qosium Probe's [log](#) gives more thorough information on the issue.

3. Common Special Situations

3.1. Negative Delay Values

Issue

Average results show negative delay values.

Probable cause and solution

It is very likely that the clock synchronization between the measurement points is not accurate. This problem is common when measuring high-speed networks without an accurate enough [synchronization method](#). The effect of inaccurate synchronization causes a mirrored delay behavior, e.g., 10 ms delay to one direction but -10 ms to the other.

To verify this assumption, see in the average results view that the mirrored delay behavior exists. If so, the synchronization problem has been verified. The actual synchronization error can be evaluated by equation

$$\text{synch_error} = (|\text{delay_u1}| + |\text{delay_d1}|)/2 + \text{delay_u1} + \text{delay_d1}$$

As a solution, check first that your synchronization method is working as supposed to. If this does not help, try better synchronization methods (e.g., replace NTP with PTP or even GNSS).

If the average results show that the delay to both directions is negative, the measurement is set up incorrectly (the next solution proposal).

Possible cause and solution

In some rare cases, the measurement setup may be so badly wrong that the packet directions, and, thus, delays, are calculated incorrectly. That can happen if you have a complex measurement setup where the Probes are not at the endpoints or within the measured path, but in external measurement points. In this kind of setup, several difficult parameters, like the senders, need to be configured manually. That can lead, if carried out incorrectly, to false QoS calculation.

As a result, perform a careful check of the measurement setup. Consider how the streams flow in the network and check that you have set your measurement point locations and possibly, sender address parameters accordingly.

3.2. QoS Results Disappear

Issue

During a two-point measurement, the average results first showed some QoS results (delay, jitter, etc.), but not any longer.

Possible cause and solution 1

The traffic to be measured has ended. Check that the traffic source is still running and producing traffic.

Possible cause and solution 2

The QMCP connection between the measurement nodes might be broken or temporarily corrupted.

Qosium measurement controllers will detect a broken QMCP connection as long as Robust QMCP mode is in use. Thus, check your measurement controller's status views and/or logs. Bear in mind that detecting a broken QMCP connection can take some time, e.g., 10 – 30 s. If a broken QMCP connection is verified to be the cause, see [Giving Commands Results in Timeout](#) to fix this.

If it seems that the QMCP connection is ok, check the *packet loss* statistic. If that shows 100 %, no other QoS statistics can be calculated. If you are sure that not all the packets are lost, try to increase the **Loss Timer** parameter value. It will give more time for packet loss detection (to separate it from highly delayed packets). Notice that increasing **Loss timer** will delay the packet loss detection accordingly.

3.3. No QoS Results

Issue

A two-point measurement is set, but no QoS results are shown at all.

Possible cause and solution 1

The traffic to be measured is not active. Check in the Flows view that the intended application to be measured is present.

Possible cause and solution 2

The measurement setup is incorrect. Capturing interface in one of the ends or both ends could be set incorrectly. Thus, check first the capture interfaces.

The capture filter could have been set incorrectly in such a way that it is not capturing the intended traffic. Thus, re-check the capture filter, and match it with the measured traffic.

Perform two separate single-point measurements in both ends to check that the intended traffic is present with the selected capture interface and filter.

Check also that the measurement point placement parameter is set correctly. In complex setups, check the sender address settings: it is easy to make mistakes there.

Possible cause and solution 3

In bad congestion situations, the QMCP connection can also suffer. That can cause QoS results to disappear, see [here](#).

3.4. Giving Commands Results in Timeout

Issue

Commands given in Qosium Scope or Scope Lite do not work any longer – a timeout error is always received (e.g., when trying to stop a measurement).

Possible cause and solution

Likely the QMCP connection from the controller or between the measurement points is broken. Typically, the measurement controller will inform about such situations, but sometimes it may not be possible (especially, if *Robust QMCP mode* is not activated). Check that the Qosium Probes are still up and running. Next, check that the connection to the Probe platforms is also ok. If so, you can try to disconnect and connect back the measurement controller. If the connection to be measured is generally of bad quality, consider routing the QMCP traffic elsewhere if just possible.



It is recommended to use the Robust QMCP Mode: it handles the QMCP connection reconnection automatically. Once the network problem is over, the connection will get back up unless the Robust QMCP timeout is exceeded.

3.5. Correct Packet Filter but No Packets

Issue

The manual packet filter seems correct, but no packets are being captured. However, by using an empty filter, the packets, including the flow of interest, appear.

Possible cause and solution

This could be caused by Ethernet-layer's *VLAN* tag. You can check this by using, for example Wireshark or other protocol analyzer to see if VLAN tag is present in the packets. If this is the case, type `vlan` and to the beginning of the filter.

3.6. Pcap Drops Packets

Issue

It seems that Pcap is dropping packets, why is this happening?

Probable cause and solution

The Pcap buffer may be too small.

Increase the size of the Pcap buffer. See the instructions and the related things to consider [here](#). Please note that Probe should be restarted after the changes have been made.

Possible cause and solution

If you did increase the Pcap buffer already, but it didn't help, the platform running Qosium Probe might not have enough resources to perform the intended measurement.

The higher is the packet rate of the measured traffic, the more it requires computational effort in the platform. In the Qosium Probe platform, take a look at the processes. That can be carried out in Windows by *Task manager* and in Linux by *top* command. If the Qosium Probe process consumes a lot of processor capacity, likely, it is not able to process the measured traffic fast enough, leading to drops.

Consider if you can filter the measured traffic part smaller to relieve the burden of the Probe. If not, consider moving the Probe to another platform having enough computational resources for the measurement.

3.7. Excessive Qosium Control Traffic

Issue

I started a measurement but it seems that Qosium Scope somehow starts to generate a lot of control traffic.

Probable cause and solution

If you are performing remote full packet capturing, perhaps you are capturing Qosium's control traffic in addition to the intended application traffic.

Modify the capture filter in such way that Qosium's own control traffic is excluded from the capture. For more information, see [How to Take Packet Capture?](#).

Possible cause and solution

Are you also collecting packet-level statistics? If so, the QMCP overhead between the measurement controller and the primary Probe will be multiplied.

Consider your measurement topology, and would it be possible to run the measurement controller in the same platform with the primary Probe. If it is not possible, try to limit the packet-level statistic collection only for special cases.

3.8. New Interface Not Appearing

Issue

I added a new network interface to my measurement computer, but it does not seem to appear in the Qosium Scope's interface list.

Probable cause and solution

Qosium Probe has an interface cache in order to prevent some unstable features of the packet capturing libraries to disturb the measurements. This cache length is typically 60 s.

Wait for a while (1 min should be enough), and try again. If waiting does not help, disconnect Qosium Scope, shut down Qosium Probe, restart the packet capturing library, start Qosium Probe, and try again. If even this does not help, then the interface is probably not supported by the packet capturing library. In this case, contact Kaitotek support.

3.9. Missing Interface

Issue

The interface I want to use is not in the available interface list.

Probable cause and solution

If there are no interfaces at all in the list, Qosium Probe is probably not run with superuser rights. Thus, restart Qosium Probe with superuser rights. Another reason might be that the packet capturing library is not activated. Thus, check that [the packet capturing library is active](#).

If there are interfaces but not the one you desire, check [the earlier issue](#).

3.10. Mouse Issue with GNSS in Windows

Issue

In Windows, when I plug my GNSS device to the computer, the mouse cursor starts jumping recklessly in the screen.

Probable cause and solution

When using serial based GNSS receiver with Windows, for some reason, the enumerator of it can mix with Microsoft serial ball pointing device. This causes very annoying random mouse cursor jumping in the screen. In order to avoid this, change the activation of the Microsoft serial mouse driver (assuming that a serial mouse is not in use):

- Open *Registry Editor*
- Go to `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\SerMouse`
- Locate value `start` and change it to 4 (typically 3 by default)
- Restart computer

4. Inconsistent Statistics

4.1. The Statistic *Sent Info Not Found* Grows

Issue

QoS statistic **Sent Info Not Found** continuously shows non-zero values. What is that and where does it come from?

Probable cause and solution: Misparameterization

The statistic *Sent info not found* is in a way *negative packet loss*. It means that, in a two-point measurement, one end has registered a received packet that was never sent from the other end.

Typically this suggests either an incorrect parameterization, erroneous topology, or loose filtering. Double-check the packet filter, and that Probe topology configuration matches with the network topology. Check with two single-point measurements that at, both ends, you see only the traffic you intend to measure.

Possible cause and solution: Altered traffic

Sometimes the network path is such that it alters the content of the packets. For example, a video stream can go through a device that transcodes the video, e.g., to a higher compression level to save bandwidth. This will cause a mismatch of the packet context between the measurement endpoints, and the QoS measurement will fail. If this is the case, the only way to proceed with the measurement is to disable the transcoding (or similar packet content alteration) or to perform the measurement in two separate parts.

Then, it could be that there is a device in the network path that has a smaller *MTU* than the traffic endpoints. That can cause a single large packet to be segmented as multiple smaller ones. However, depending on the *NIC*'s capabilities, the segmentation can be masked from the measurement perspective, causing no problems. If not, however, the QoS measurement will fail at least partially. In such a case, the solution is to lower the MTU in the communicating endpoints in such a way that packet segmentation does not occur. Unfortunately, typically, the only way to do this is just to test which MTU value is low enough. Please note that MTU, generally, should not be set too low since it can cause inefficiencies in network resource usage.

Possible cause and solution: Bad conditions

If the network connection between the measurement points is bad, QMCP can also suffer, interfering with the QoS calculation process. In this case, there is typically high delay, excessive packet loss, including also *Sent info not found* packets. You can try to increase the parameter **Packet loss timer** value. It will delay the packet loss detection, giving more time for QoS calculation. If it does not help, consider routing the QMCP traffic elsewhere if just possible.

4.2. The Statistic *Duplicate Packet Count* Grows

Issue

I see duplicate packet statistics increasing all the time. Also, QoS values seem strange or absent from time to time. What is happening?

Probable cause and solution

Most likely, there is nothing wrong with such, but it is just that the traffic you measure has interesting properties. By default, Qosium identifies packets based on payload. Thus, if the payload from one packet to another is exactly the same, Qosium counts them as duplicates. That is normal – some applications just generate similar content to packets in some situations. Even normal TCP connections can cause duplicate packets. If information is lost, TCP resends the lost data, appearing as a duplicated packet for Qosium, if the packet context matches suitably.

Since Qosium Probe version 1.4.15.0, QoS will not be calculated for duplicated packets any longer. That is because there is a danger that QoS will be calculated incorrectly especially if packet losses are occurring over the measured network path. Thus, if you are using newer Qosium versions, this explains the absence of QoS results, and with an older one, the mentioned strange QoS results are likely caused by this. Remember, however, that this happens only for those packets that are duplicates. Other packets are calculated normally. In addition, if the duplicate packet interarrival time » Qosium's **Packet loss timer**, they are calculated normally as individual packets.

If you wish to measure QoS also for the duplicated packets, try different Packet identification methods. For example, if the traffic is of IPv4, using the IPv4 ID often relieves this problem. Be aware, however, that if you measure high traffic loads, the IPv4 ID field rolls over within Qosium's averaging interval, and, again, there will be duplicated packets. Moreover, some (fortunately rare) applications do not respect the IPv4 ID field's purpose. Instead, they always put the same value to this field. As understood, in this case, the packet identification will fail.

It is also possible that a network device is operating incorrectly, causing duplicates. In this case, changing the Packet identification method makes no sense. Instead, Qosium provides you valuable information about a misbehaving network device. Thus, try to locate the device and see what is wrong.

4.3. High Packet Losses

Issue

I am measuring a specific application and its traffic is shown in the Flows view. Qosium shows high packet loss even though there are no problems in the experienced application performance.

Probable cause and solution

It is likely that your measurement setup or parameterization does not match with the measurement scenario.

Check first that does the statistic *Sent info not found* grow. If so, act [accordingly](#).

Possible cause and solution

Some network devices may induce duplicate packets to be seen only in one of the measurement points. In older Qosium versions, those duplicates will be calculated as lost packets.

If this is the case, [see what is causing duplicates and how to deal with them](#).

Possible cause and solution

Link might be slow. To test the case of a slow link, set the parameter **Packet loss timer** to higher value.

5. Known Limitations

This section discusses some of the known limitations of Qosium and how to deal with them.

5.1. Performance Considerations

5.1.1. Qosium Probe's Serving Capabilities

A single Qosium Probe can support theoretically up to about 32000 simultaneous measurements. However, in practice, the number of simultaneous measurements is much lower. Typically the limiting factor is the memory. In a normal PC, you can perform typically hundreds of measurements, but not thousands. You can affect this by [packet capturing parameterization at Probe](#).

5.1.2. Packet Rates

The higher the packet rate of the measured traffic, the more computational effort measuring it requires. The meaning of packet sizes is negligible. In a typical PC, you can easily perform a single-point measurement of streams of 100000 pkts/s, meaning that a 1 Gbit/s link is filled already with traffic. On more powerful desktop computers and servers, measuring streams of 500000 pkts/s or higher should work fine. Then again, in low-powered embedded devices, the limits can be much lower, e.g., 5000 pkts/s. Hence, the measurement limits are heavily dependent on the device where you run Qosium Probe.

Certain measurement features also affect the reachable performance. A two-point measurement consumes roughly 2.5 times more computational resources than a single-point one. Then, payload-based packet identification is about two-times more laborious than IPv4 or RTP identification. Still, however, the measurable packet rate does not always decrease accordingly. The reason for this is that Qosium Probe is able to exploit multiple processor cores.

5.1.3. Taking Full Captures and Collecting Packet-Level Results

Full packet captures from a remote measurement point generate a QMCP load about equal to the measured traffic stream. Thus, it is recommended to take full captures only when needed.

Packet-level measurement results also increase the QMCP load between the measurement controller and the primary measurement point. The increase is clearly less when compared to the full capture but still significant. The overhead increases as a function of the measured traffic packet rate. Besides, collecting packet-level statistics is more laborious for the measurement controller than sticking with the average results. Thus, case by case, consider that do you really need the packet-level statistics or would the average results be enough.

5.2. Network Protocol Support

Qosium can measure the QoS of any application or service using Ethernet, IPv4, or IPv6. Currently, IPv6 support is developed only for the measurement, and the control channel for the measurement happens over IPv4 only at the moment. Bringing full IPv6 support is on the roadmap, but its implementation pace depends on the need. Thus, if you desire full IPv6 support, ask Kaitotek support!

5.3. Large Send Offload and Large Receive Offload at NIC

5.3.1. Support in General

Most of the modern NICs are capable of performing some operations normally done by the IP stack of the operating system. Handing calculation from the IP stack to the NIC is called *offloading*. Example methods are Large Send Offload (LRO) and Large Receive Offload (LSO). Qosium can handle most of the LRO and LSO cases without problems. A known limitation still is that the MTU sizes between the measurement points should match. If not, Qosium's offloading support likely fails.

Some things are worth of noting related to the offloading support of Qosium. First, Qosium's offloading support is available only for the following Packet Identification Methods:

- Pure Payload Based ID (the safest)
- IPv4 ID (works in most cases)

Then, in Linux, with the interface *any*, through which all traffic from all the interfaces is seen, the offloading support of Qosium does not work. The reason is simple: in order for the handling to work, Qosium needs to operate with the real network interface directly, i.e., the one which performs the possible segmentation or reassembly of the packets. With the interface *any*, this is not the case.

Finally, in the tests it has been noticed that some setups can mess up the offloading operations in such way that it affects the measurement. This kind of operation has been noticed to happen at least in Ubuntu 18.04 when the machine is used as a bridge. Also, Windows 10 machines with some USB Ethernet adapters seem also cause similar effects. However, the found cases affect only when the *IPv4 ID* method is selected as the **Packet Identification Method**. Operating with *Pure Payload Based ID* should work fine.

5.3.2. Turning Offloading Off

In some special cases there might be a need to turn the offloading functions off. This section tells how to do that in different platforms.

5.3.2.1. Windows

In Windows, you can change the adapter setting through network adapter list (e.g., in Windows 10: *Control panel => Network and Internet => Network Connections*). Open the network adapter properties, and there click *Configure*, which opens up the accurate parameters. There, in leaf *Advanced*, you can change the offloading settings. The changed settings should be permanent.

5.3.2.2. General Linux

In many Linux based systems, the command line tool Ethtool can be used to manipulate the interface options. First, offloading settings can be checked by a command

```
ethtool -k <interface>
```

Then, you can manipulate the offloading options with an option *-K*. Remember that if, e.g., segmentation and reassembly offloading is enabled, one can lose some accuracy in the link level delay measurement. Regarding the measurement, harmful offloading features are: Generic Receive Offload (GRO), Generic Segmentation Offload (GSO), and TCP Segmentation Offload (TSO). Disabling these can be done with the following Ethtool command:

```
sudo ethtool -K <interface> gro off gso off tso off
```

5.3.2.3. Making the Setting Persistent in Ubuntu

To make reassembly setting persistent in Ubuntu, modify the `/etc/network/interfaces` file and add a post-up line to the network interface to be measured. Replace `<interface>` with the real network interface name in the following example:

```
auto <interface>
iface <interface> inet <static/dhcp>
[...]
post-up /sbin/ethtool -K <interface> gro off tso off
```

5.3.2.4. Making the Setting Persistent in CentOS/RHEL 7

To make reassembly setting persistent in CentOS and RHEL 7 Modify the `/etc/sysconfig/network-scripts/ifcfg-<interface>` file and add `ETHTOOL_OPTS` as follows:

```
ETHTOOL_OPTS="-K ${DEVICE} gro off tso off
```

If Ethtool is not available, install it with a command `yum install ethtool`.

5.3.2.5. Making the Setting Persistent in Most of the Linux Systems

Add the Ethtool command to be always run at the start-up of operating system. There are multiple options to do this. Replace `<interface>` with the real network interface name.

Crontab: Type `sudo crontab -e` and add the following line to the end of the file:

```
@reboot /sbin/ethtool -K <interface> gro off tso off
```

`/etc/rc.local`: open the file with the superuser rights and add

```
/sbin/ethtool -K <interface> gro off tso off
```

to the end of the file before `exit 0`.

5.4. Measuring over Unreliable Network Paths

QMCP stream, as it is currently carried over TCP, even though it is light, is also as vulnerable to network conditions similarly as the measured other streams. Hence, e.g., if the network is congested, QMCP packets might be occasionally lost, and the real-time measurement could become unreliable. Besides occasional delays (no results), the standard TCP connections over unreliable links may break. Thus, if this happens, the two-point measurement can seize. Under very bad network conditions, it might be that the measurement cannot be made. There are at least three ways to cope with this challenge:

The first thing to do is to make sure that the **Robust QMCP Mode** is active (by default, it is). When used, Qosium detects a broken link quickly and is able to recover from it. Qosium will tolerate connection breaks of length defined by parameter **Robust QMCP Timeout**. However, notice that during a measurement, the breaks may occasionally be long, meaning that no results are got during that period of time. Nevertheless, the **Robust QMCP Mode** allows measurements to recover when measuring long periods of time over

unreliable links.

If the connection between the Qosium Probes is very bad, the QMCP traffic can also be routed through a reliable connection if possible in the measurement scenario. This is the optimal case, since then the QMCP traffic is not present at all in the measured connection.

If the connection is not good but not very bad either, try to increase the value of the **Packet loss timer** parameter. This might be enough to cope with the unreliable link, but the down side is that the packet loss result calculation is delayed accordingly.

5.5. On the Supported Network Interfaces

Qosium captures packets directly from the network interface using Pcap. However, some network interfaces might not work properly with Pcap. Thus, if there are problems with capture interfaces, please ask Kaitotek support for assistance.

6. Glossary

Virtual LAN

A LAN segment which has no physical hardware, but instead is carried over network.

Maximum Transmission Unit

This determines the largest packet size that can be communicated in a single network-layer transaction.

Network Interface Card

A piece of hardware which offers a device a networking interface.