Meet

QoS best practices

Do not use QoS

You should not use QoS in your network, because Meet automatically adapts to network conditions. Use QoS only if you have compelling reasons, such as a congested network, and are able to deploy and maintain an end-to-end QoS model in your network.

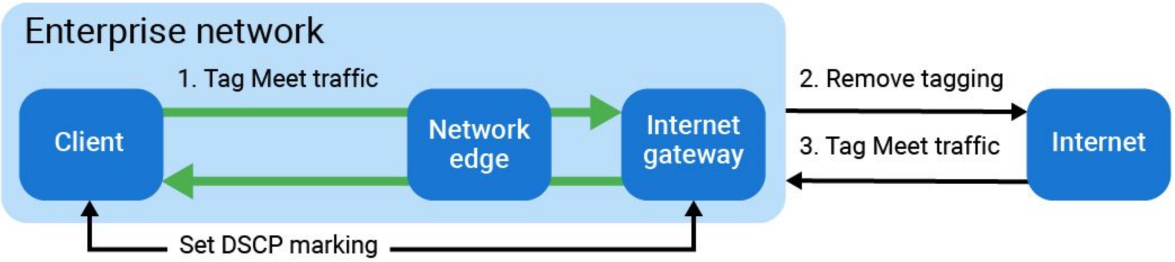
If QoS must be used

If your network is congested and you must secure a certain quality of service (QoS) for Meet, take one of the following actions:

- Add QoS on Meet clients.
- Add QoS at the network edge.

Add QoS on Meet clients

Meet traffic is tagged on the client machines for QoS within the enterprise network. The QoS tags are removed when the traffic is sent to the internet. Incoming Meet traffic is tagged when it enters the enterprise network.

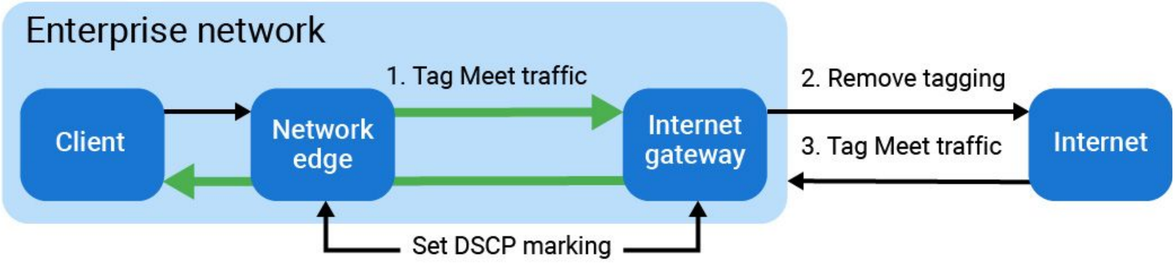


Tag Meet traffic at the client

1. Set a DSCP marking through a Windows QoS Policy in the Group Policy Management Console. (Identify Meet traffic using Meet's port range.)
2. Remove the DSCP tagging traffic that leaves your internal gateway to the internet.
3. Tag Meet traffic received from the internet. This is the real-time transport (control) protocol (RTP/RTCP) traffic that uses the Meet port ranges.

Add QoS at the network edge

Client Meet traffic is tagged at the network edge for QoS within the enterprise network. The QoS tags are removed when the traffic is sent to the internet. Incoming Meet traffic is tagged when it enters the enterprise network.



Tag Meet traffic at the network edge

1. On all network edges, add a rule to mark Meet traffic. You should assign the EF (Expedited Forward) class for Meet traffic to ensure low delay and low jitter. This is the RTP/RTCP traffic that uses the Meet port ranges.
2. Remove DSCP tagging for the traffic that leaves your internal gateway to the internet.
3. Tag Meet traffic received from the Internet. Ensure that the EF class is used. This is the RTP/RTCP traffic that uses the Meet port ranges.
4. Within the company, network EF traffic should be prioritized to achieve low delay, jitter, and loss values. This is usually achieved by placing such traffic into low latency or strict priority queues. Implement additional precautions, such as rate limiting above pre-defined bandwidth values, to make sure that this EF traffic doesn't starve other traffic classes on the network.

Test QoS

When looking at network statistics and measuring QoS performance and success, start with a small test bed to discover how a single device performs.

Follow the packets' path through the individual network devices. Use this to validate that the network path respects client markings (see below) and to understand individual queue drops and throughput on devices.

Different hardware vendors have different QoS implementations, so things may differ slightly. Fine-tuning is needed to ensure end-to-end QoS.

Some non-intelligent network devices, such as a hub or low-end switch, may not support the full QoS feature. Make sure that the DSCP value marked at the upstream device is not modified, so that the downstream intelligent devices can apply the correct QoS strategy, based on the correct marking.

Ensuring the network path respects client markings

There are different options to validate whether the network path respects client DSCP markings.

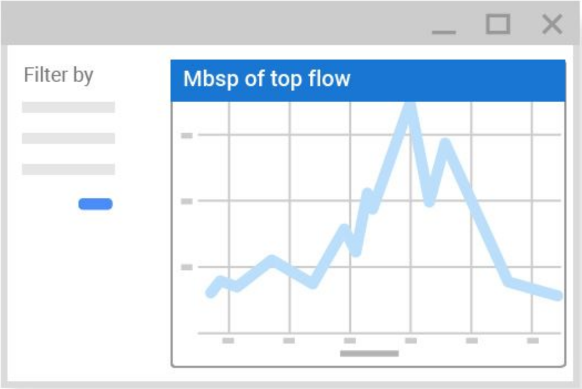
Packet sniffing, with Wireshark for example, can verify the correct DSCP markings on both the network device (AP, router, switch) and end device (laptop). Use a port mirroring or switch port analyzer (SPAN) to send captured data to a selected destination port for local port mirroring. A remote port protocol such as remote switch port analyzer (RSPAN) can send captured data to a remote server for analysis.

Netflow is another powerful tool to verify DSCP marking on the network device. The DSCP value is exported by default to the collector. Filter the 5-tuple (IP, protocols, and ports) from the captured data to verify the DSCP value for each specific application.

Monitoring Meet QoS performance on the network level

An SNMP-based monitoring tool can display the trending view of different queue utilization, as well as queue drop. Hangouts Meet is marked at the application level as AF4. By looking at the AF4 utilization and drop rate of this class, we know the Meet performance for an interface on the network.

By aggregating the application data, Netflow can display a stacking view of a site-specific or global view.



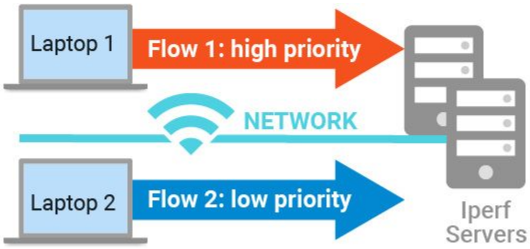
Netflow-based Aggregated Meet utilization

Simulating congestion and QoS validation

To simulate congestion, generate multiple flows of traffic that exceed the maximum bandwidth of the media. For example, generate 2G traffic over a 1G path. Then compare the throughput at the receiving endpoint to verify if high-priority traffic gets adequate treatment.

For wireless, emulate congestion by generating multiple flows to the same access point. For example, for 802.11n, send 2 x 150 mbps for each flow, because 802.11n can support a maximum throughput of about 180 mbps. Then verify the throughput at the receiving endpoint.

For example, to prove high-priority traffic gets better service, send a different class of traffic over Wi-Fi (at the same access point, in the same collision domain). High-priority traffic should get all the throughput without dropping, while low-priority traffic should drop dramatically.



Wi-Fi QoS proof of concept illustration

Use the following command line to test that QoS is performing as expected:

Best Effort:

```
iperf3 -c <IP ADDRESS> -u -b 150m -t 50 -l 1000B -i 10 -S 0x0
```

AF4:

```
iperf3 -c <IP ADDRESS> -u -b 150m -t 50 -l 1000B -i 10 -S 0x80
```