IPFIX QoS Measurement Extension
Problem

User: Your application is slow/does not work

Application Administrator/Developer: It’s the network’s fault. Things work on my systems but break in this network

Network Operator: What application? What does it do? Why should the network be the problem?

Goal: Provide Networking guys with the information whether it's an application problem or a network problem
Network Properties that Can Cause Problems

- **Delay**
  - Latencies on the network
  - Changes in latency (jitter)

- **Packet Loss**
  - Drops along the path

- **Bandwidth Limitations**
  - Slow links on the path
  - Buffers between slow and fast links
    → can result in delay

- **Connectivity Limitations**

- **Get information on these properties**
  → if we can’t see one of these issues, it’s not a network problem
Active Measurements

Application Servers

Data Center

Send probing packets
Example: Determine RTT

Provider Network

Enterprise Remote Site

Open Internet

Application Servers

Example: Determine RTT
- **Example:** Buffer bloat and resulting large delays

- **Discussions about if those properties affect the application**

- **What we want to have:**
  - Tell application about the properties of their connections
  - Passive measurements
Include QoS Measurement Results into IPFIX flows

- **Approach**
  - Calculate network performance metrics
  - Attach results to IPFIX flows

- **Measuring of QoS metrics**
  - application-level statistics
    → example: RTP streams
  - What about other applications?

- **Work has been done**
  - Delay measurements
    • RTT estimation
    • Network delay
    • Application delay
  - Inclusion of various other fields
Example: Determine Network Delays

Client

Server

Probe

First message

Second Message

- SYN
- SYN/ACK
- ACK

Add new fields for delays to IPFIX flows

- Server Network Delay
- Client Network Delay
- First Response Delay
- Application Delay

subtract network delay
Measurement Applicability

- Works fine in these scenarios:

- Does not work with asymmetric paths:
Do not export delay values but timestamp of certain packets

Perform your calculation on the flow collector
Approach for Network Performance Metrics

- Do not calculate metrics on the probe
  - extract information from packets
  - do calculation on the collector
  → provide a number of sampling algorithms that help to extract the data

- Advantage
  - Asymmetric paths
  - No complex algorithms on the probe
  - Instead
    - Sample packets and export timestamps

- Drawback
  - Requires more information attached to the flow data
Employ Timestamps in Probes and Network Devices

What about the other flows?

Information about this flow

What we want: Get this information from the network devices
Discussion

- **Requirement**
  - Time synchronization between devices (at least to certain level)
  - Router/Switches: Good timestamps from devices

- **Asymmetric paths**
  - Is performance metrics for asymmetric paths an issue?
  - Should performance metrics be aware of asymmetric paths?

- **Asymmetric path-aware algorithms would require to include more information into a single flow**
  - Does someone have operational experience with generating flow data with a lot of non-flow keys attached to the individual flows?