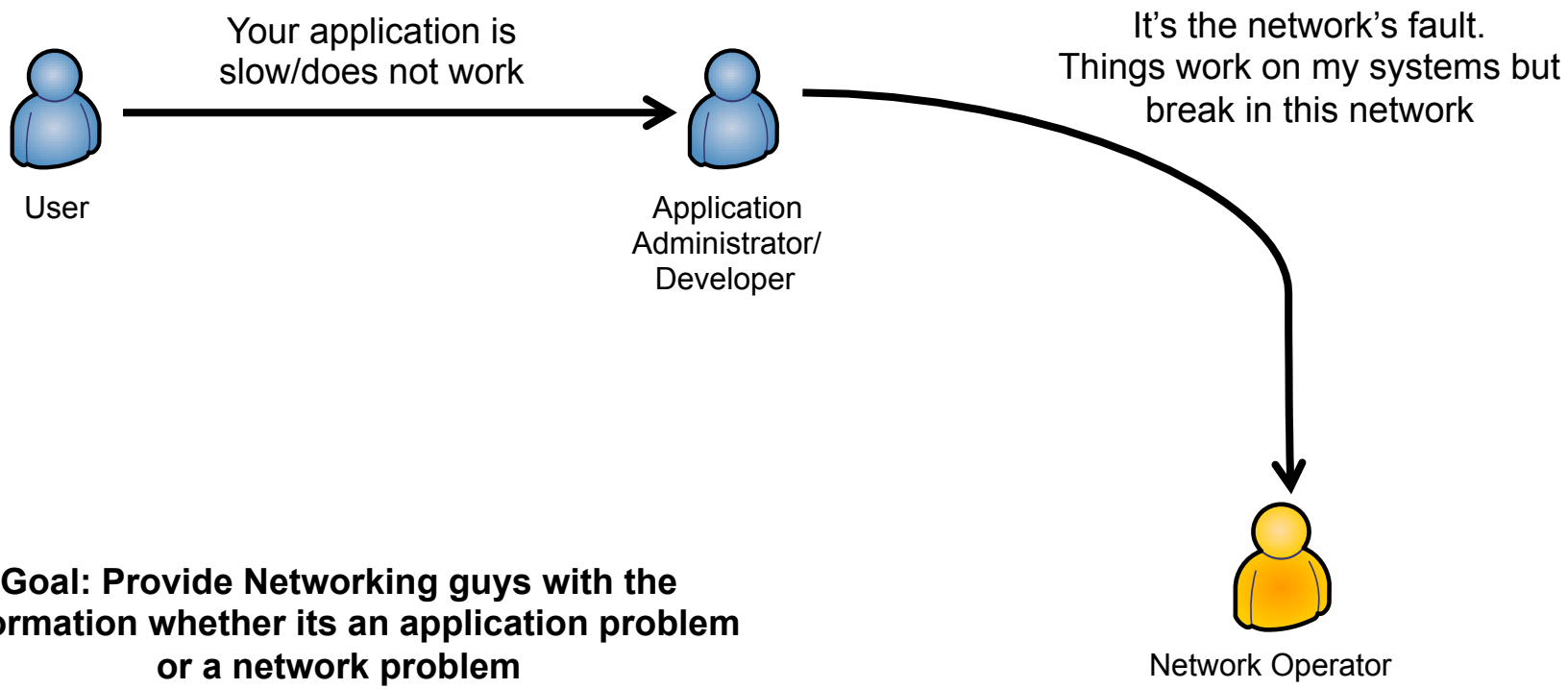
IPFIX QoS Measurement Extension

Problem



Goal: Provide Networking guys with the information whether its an application problem or a network problem

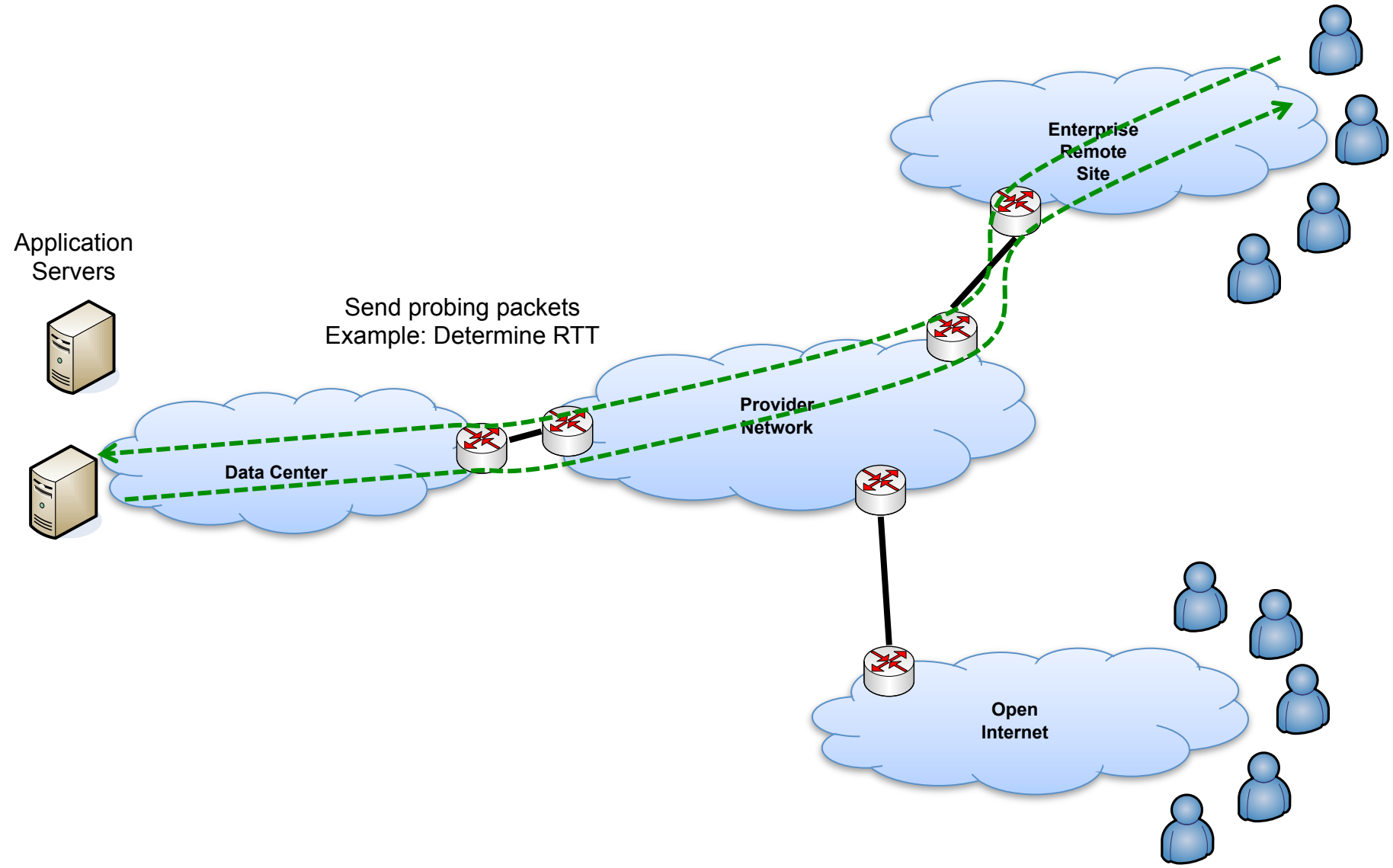
Network Properties that Can Cause Problems

- **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**

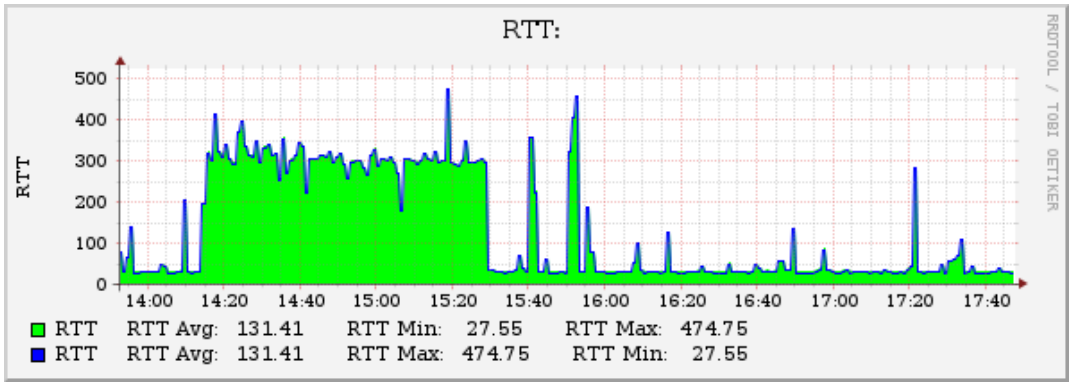
  → if we can't see one of these issues, it's not a network problem

Active Measurements



Network Properties and Possible Problems

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



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

- **What we what to have:**
  - Tell application about the properties of their connections

  → Passive measurements

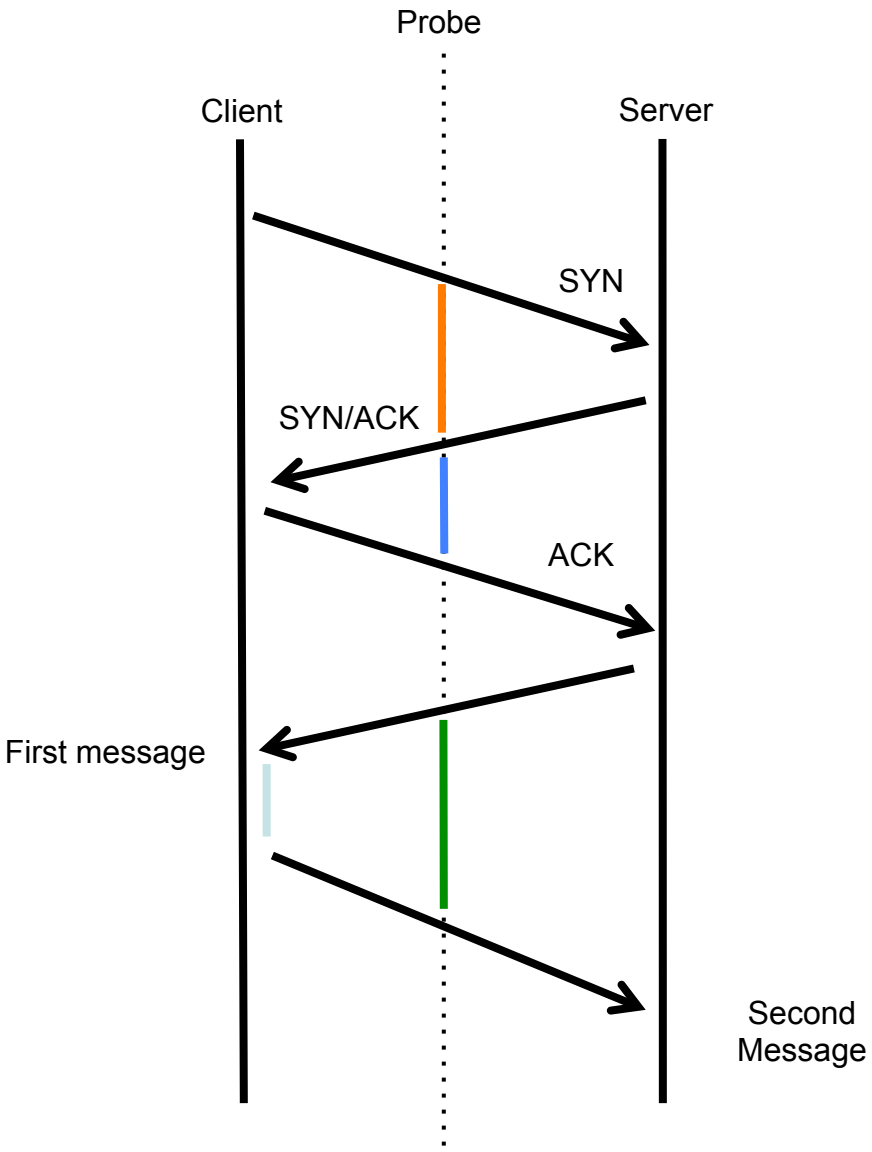
Include QoS Measurement Results into IPFIX flows

- **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



Server Network Delay

Client Network Delay

First Response Delay

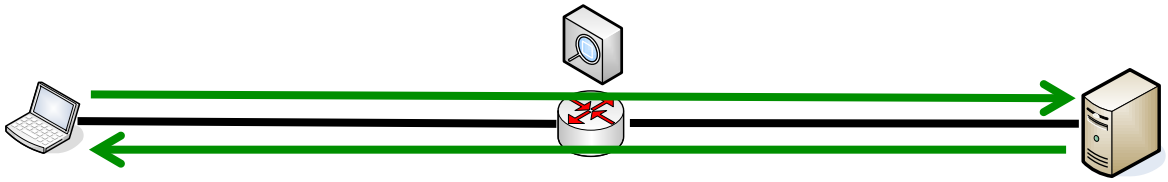
subtract network delay

Application Delay

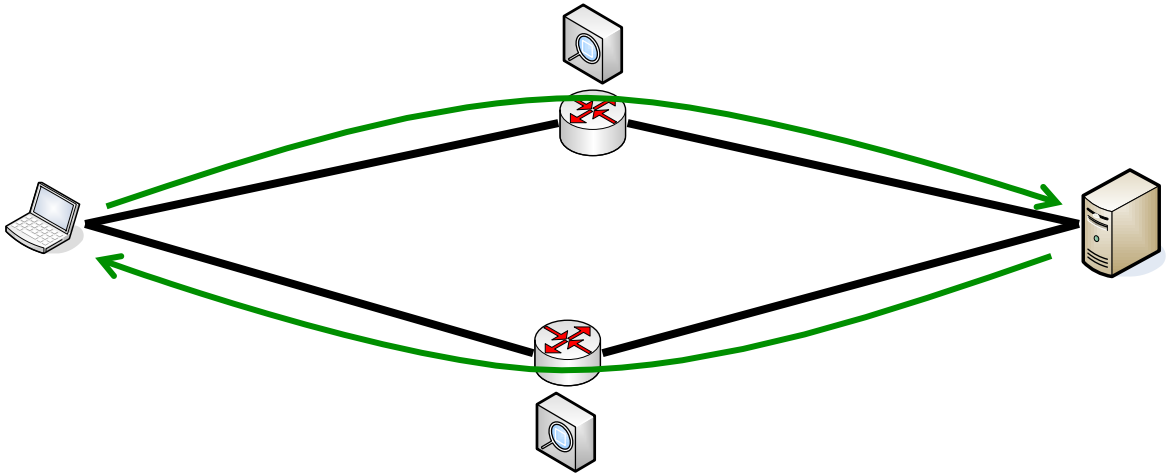
Add new fields for delays to IPFIX flows

Measurement Applicability

- Works fine in these scenarios:

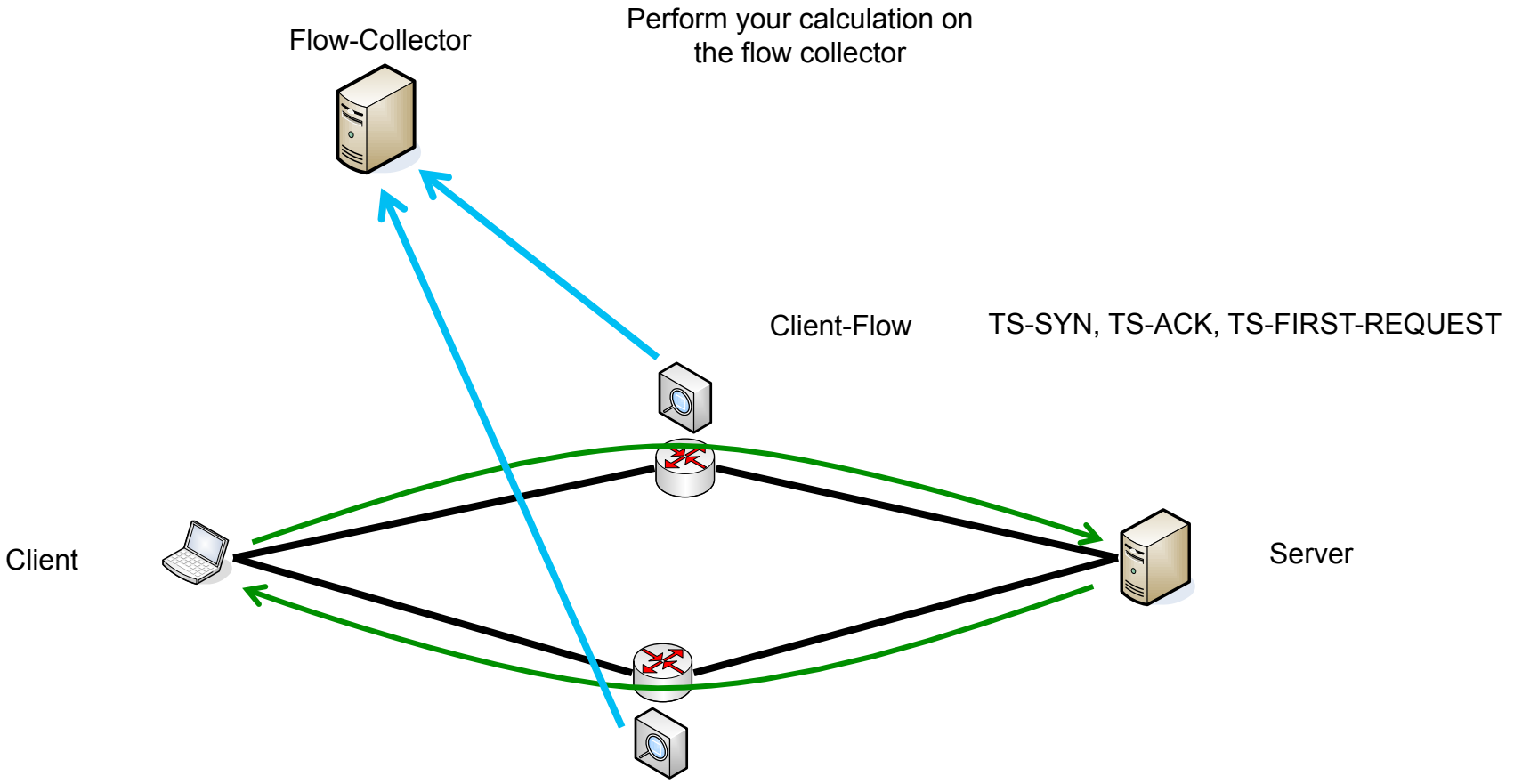


- Does not work with asymmetric paths:



Delay Estimation for Asymmetric Paths

- **Do not export delay values but timestamp of certain packets**



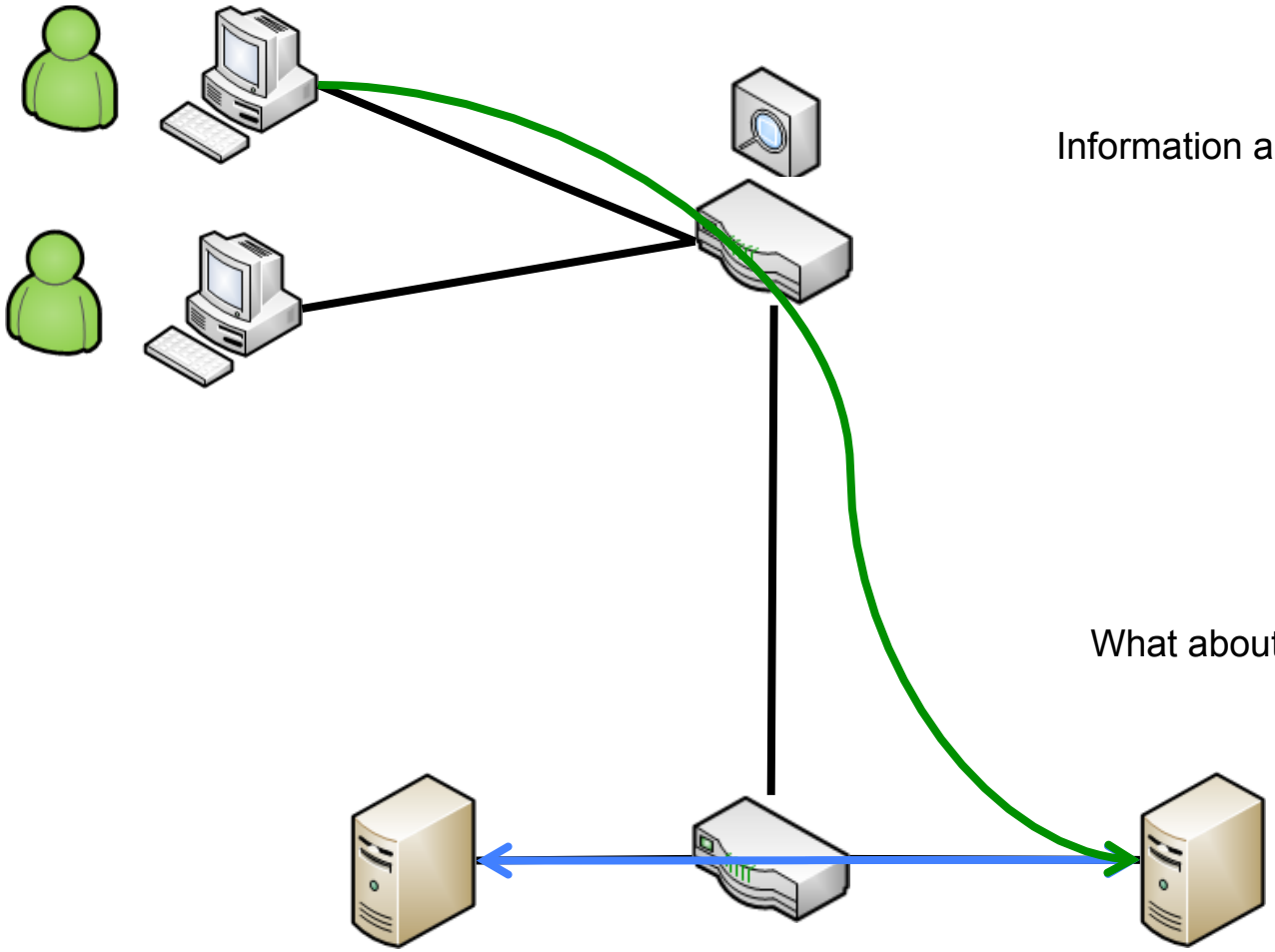
Approach for Network Performance Metrics

- **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 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?