Transport Layer Statistics for Network Quality

This paper describes some techniques for using transport layer statistics to measure network quality and for improving performance of cloud services and edge devices. The techniques discussed do not require the cloud service and the edge devices to be in the same administrative or platform domain. The paper also describes some challenges and possible research areas.

Cloud Services

Cloud services serve user requests typically over an application layer protocol like HTTP and make heavy use of transport layers like TCP or QUIC. While the application layer can measure end to end service latency and requests/sec, it historically did not have visibility into transport layer metrics. Exposing transport layer statistics (like RTT, loss signals, reordering, etc.), and HTTP layer statistics (TLS handshake timings) per flow allows the application layer to correlate each request with underlying network conditions as measured by the transport.

Metrics

- Application layer metrics like request latency at P50, P90, P99, and P99.9, time to first byte
  Correlated with:
- Transport layer metrics like handshake time, smoothed RTT, RTT variance, reordering degree, congestion signals, measured inbound and outbound bandwidth, retransmissions, app versus sender versus receiver limited time

Such correlation allows for detailed analysis of request latency and workload performance. For cloud services it is very straightforward to measure workload performance but it is hard to measure last mile network characteristics acting alone. Additional data from network providers is sometimes necessary to attribute increased request latency to say a route flap event or congestion in the network.

Edge Devices

Edge devices typically make requests to cloud services over an application layer protocol like HTTP. With edge devices the metrics span billions of data points spread across a variety of network conditions. Any measured network quality data needs to be statistically significant.

Metrics

- Application layer metrics like response time, bytes transferred over unit time, audio jitter, video rebuffer rate
  Correlated with:
- Transport layer metrics like handshake time, smoothed RTT, RTT variance, reordering degree, congestion signals, measured inbound and outbound bandwidth, retransmissions, app versus sender versus receiver limited time

For Edge devices there is no clear single definition of network quality. Every application has different goals like lower latency or higher throughput but today's APIs do not allow expression of intent. And then there is user intent as well where certain traffic should be treated with a higher priority – and again for user friendliness this needs to be inferred rather than require intervention. With proliferation of Wi-fi and LTE connectivity, L2 network quality signals need to be correlated with transport layer and application layer metrics.

Challenges

- Receive side of the transport connection usually has very little information compared to the sender side. For example, receiver side of the TCP connection may have very few RTT samples (for a connection that primarily receives data) and no visibility into the congestion state which managed by the sender side.
- Network conditions can change rapidly and correlating these using transport layer data in real time with application layer metrics is difficult.
- At cloud scale and with billions of edge devices, the scale of data that needs to ingested and processed is immense. Figuring out how to throttle the amount of data without losing information requires very careful design of metrics.

Research areas

- Designing protocol extensions to exchange metrics with the peer. For example, allowing HTTP peers to exchange transport layer statistics. This would allow a cloud service to correlate transport metrics with that observed by the edge device.
- Privacy preserving metrics where entities not under the same administrative or device platform domains could exchange and correlate quality metrics.
- Defining standard metrics for measuring efficacy of congestion control algorithms. What is the minimum bar for incremental deployment without causing harm to cross-traffic?
- How to measure efficacy of less than best effort network protocols? How can we define metrics for network network citizenship i.e. behaving better on the wire compared to other traffic?