



# Network Management & Monitoring

## Network Delay



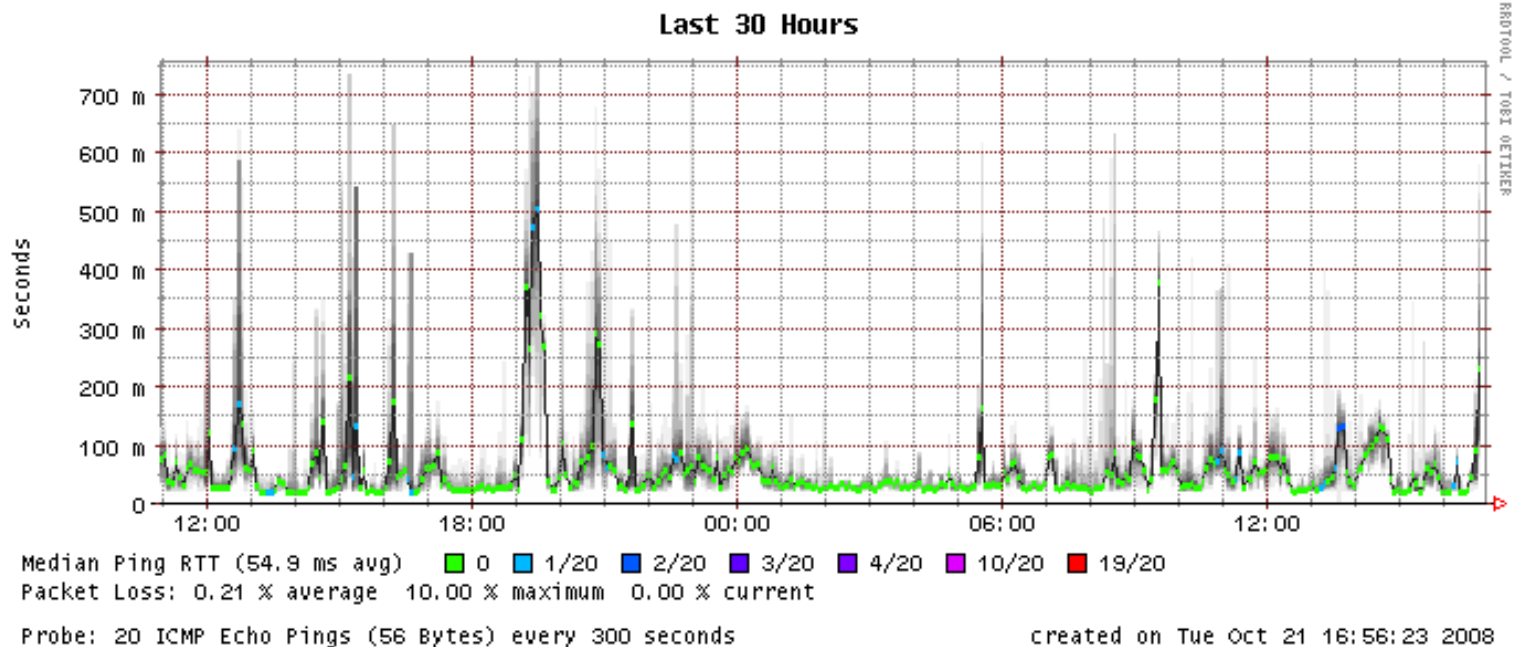
These materials are licensed under the Creative Commons *Attribution-Noncommercial 3.0 Unported* license (<http://creativecommons.org/licenses/by-nc/3.0/>)

# End-to-end Delay

**The time required to transmit a packet along its entire path**

- *Created by an application, handed over to the OS, passed to a network card (NIC), encoded, transmitted over a physical medium (copper, fibre, air), received by an intermediate device (switch, router), analyzed, retransmitted over another medium, etc.*
- The most common measurement uses *ping* for total round-trip-time (RTT).

# Historical Measurement of Delay



# Types of Delay

## Causes of end-to-end delay:

- Processing delays
- Buffer delays
- Transmission delays
- Propagation delays

# Processing Delay

Required time to analyze a packet header and decide where to send the packet (e.g. a routing decision)

*Inside a router this depends on the number of entries in the routing table, the implementation of data structures, hardware in use, etc.*

This can include error verification, such as IPv4, IPv6 header checksum calculations.

# Queuing Delay

- The time a packet is enqueued until it is transmitted
- The number of packets waiting in the queue will depend on traffic intensity and of the type of traffic (bursty or sustained)
- Router queue algorithms try to adapt delays to specific preferences, or impose equal delay on all traffic.

# Transmission Delay

The time required to push all the bits in a packet on the transmission medium in use

For  $N$ =Number of bits,  $S$ =Size of packet,  $d$ =delay

$$d = S/N$$

For example, to transmit 1024 bits using Fast Ethernet (100Mbps):

$$d = 1024/1 \times 10^8 = 10.24 \text{ micro seconds}$$

# Propagation Delay

- Once a bit is 'pushed' on to the transmission medium, the time required for the bit to propagate to the end of its physical trajectory
- The velocity of propagation of the circuit depends mainly on the actual distance of the physical circuit
- In the majority of cases this is close to the speed of light.

For  $d$  = distance,  $s$  = propagation velocity

$$\mathbf{PD = d/s}$$



# Transmission vs. Propagation

Can be confusing at first

Consider this example:

## **Two 100 Mbps circuits**

- 1 km of optic fiber
- Via satellite with a distance of 35,000 km between the base and the satellite

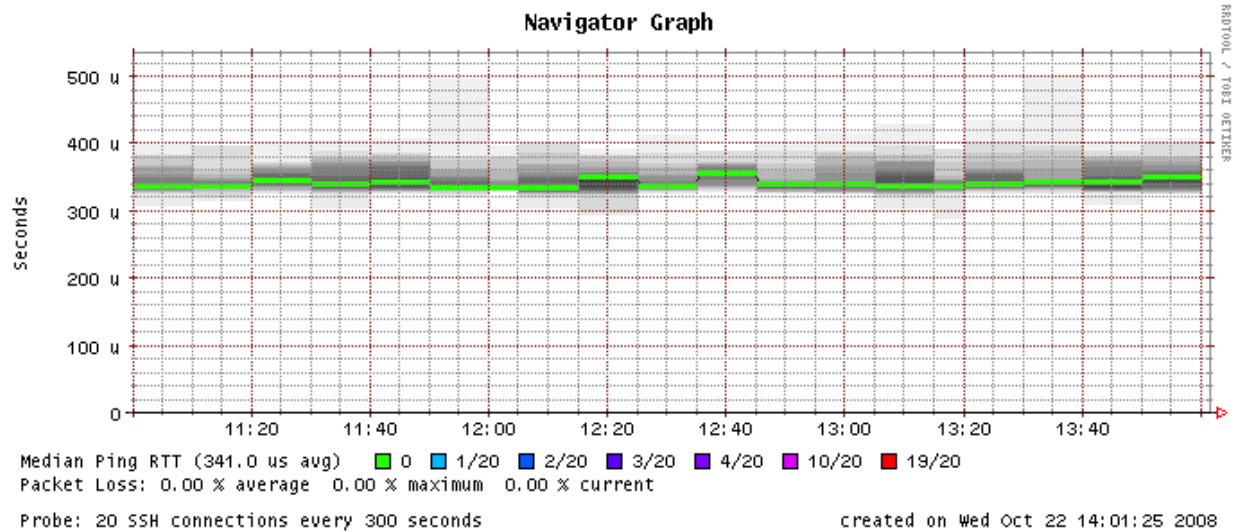
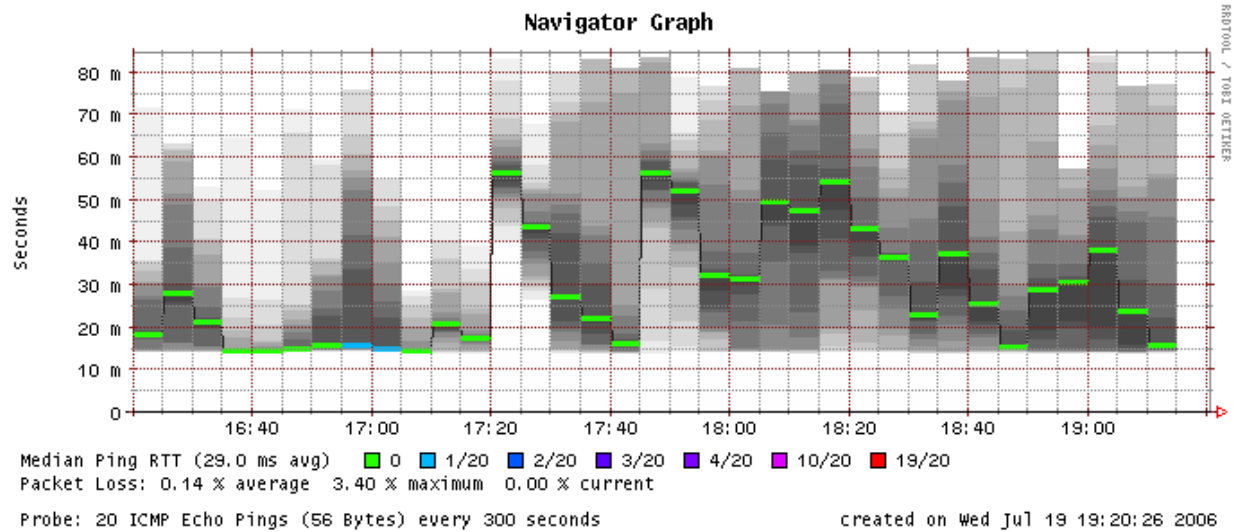
For two packets of the same size which will have the larger transmission delay?  
Propagation delay?

# Packet Loss

Occurs due to the fact that buffers are not infinite in size

- When a packet arrives to a buffer that is full the packet is discarded.
- Packet loss, if it must be corrected, is resolved at higher levels in the network stack (transport or application layers)
- Loss correction using retransmission of packets can cause yet more congestion if some type of (flow) control is not used (to inform the source that it's pointless to keep sending more packets at the present time)

# Jitter



**Questions?**

**?**

# Flow Control and Congestion

Limits the transmission amount (rate):

- because the receiver cannot process packets at the same rate that packets are arriving.
- because of loss or delays in the circuit.

# Controls in TCP

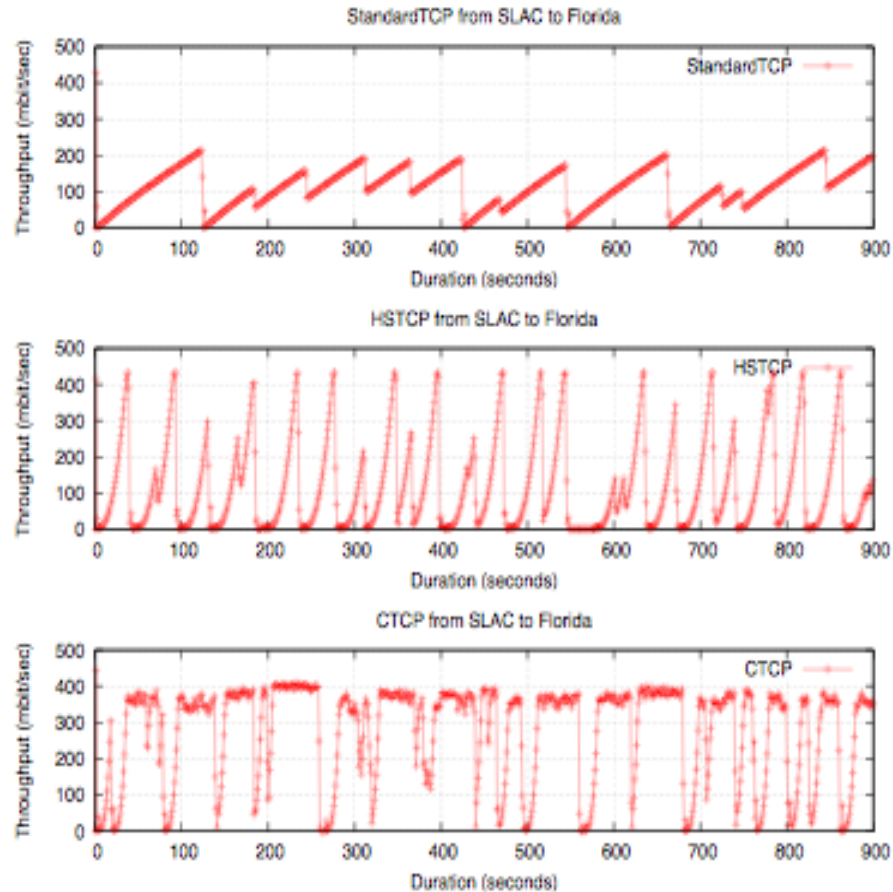
IP (Internet Protocol) implements service that is not connection oriented.

- There is no mechanism in IP to deal with packet loss.

TCP (Transmission *Control* Protocol) implements flow and congestion control.

- Only on the ends as the intermediate nodes at the network level do not talk TCP

# Different TCP Congestion Control Algorithms



# Congestion vs. Flow in TCP

**Flow**: controlled by window size (RcvWindow), which is sent by the receiving end.

**Congestion**: controlled by the value of the congestion window (Congwin)

- Maintained independently by the sender
- This varies based on the detection of packets lost
  - Timeout or receiving three ACKs repeated
- **Behaviors:**
  - Additive Increments / Multiplicative Decrements (AIMD)
  - Slow Start
  - React to *timeout* events