Measuring IP Performance

Geoff Huston

Telstra

What are you trying to measure?

- User experience
  - Responsiveness
  - Sustained Throughput
  - Application performance quality
  - Consistency
  - Availability
- Network Behaviour
  - Routing Stability
  - Path characteristics
- Element Behaviour
  - Subnet characteristics
  - Switch element behaviour
  - Switch resource consumption
- Network availability
  - Element availability
  - Transmission path availability
  - Transmission element BER
  - Network path availability
- Path characteristics
  - Latency
  - Jitter characteristics
  - Loss characteristics

Observation

- Using a combination of active and passive measurement techniques there is a massive set of possible aspects of network behaviour that can be measured
- Few measurements have any real bearing on the performance characteristics of applications that include some form of network interaction
  - i.e. there's a difference between measuring any old thing and measuring something relevant and useful

- If you are going to measure something…
  - Know why you are measuring it
  - Understand the limitations of the measurement technique
  - Understand the limitations of any interpretation of the measurement
  - Understand who is the consumer of the measurement

IP Performance

- The end-to-end architectural principle of IP:
  - The network should not duplicate or mimic functionality that can or should be provided through end-to-end transport-level signalling
  - IP networks can be seen as queue-controlled passive switching devices connected through fixed delay channels

- Network performance is the interaction of concurrent end-to-end applications performing a role of mutually enforced resource sharing
  - The network is not a mediator or controller of an application's resource requirements
  - Each network transport application behaves in a fair greedy fashion, consuming as much of the network's resources as other concurrent network transport applications will permit

Network Measurement Approaches

- PING and related probe techniques
  - Send an ICMP echo request to a target device and measure the time to respond
  - Often used to interpret some indication of delay, loss and jitter
  - BUT has little relationship to application performance, as the probe measurement is heavily impacted by the behaviour of the probe and the echo point
    - i.e. beyond being a remote device availability beacon, its of little practical use
- SNMP
  - Per-element probe to poll various aspects of an element's current status
  - Of little practical value in determining end-to-end network performance, as there is a distinct gap between end-to-end path performance and periodic polling of network element state
- Active Test Traffic
  - Perform a particular network transaction in a periodic fashion and correlate application performance across invocations
  - Often measures the performance limitations of the test gear and the target rather than the network
  - Tests only a small number of network transit paths
  - Provides only a weak correlation between measurement results and actual end-user experiences of application performance

Maybe asking how to measure network performance is the wrong question

- How well your car operates is an **interaction** between the functions and characteristics of the car and the characteristics of the road – trip performance is not just the quality or otherwise of the road

- How well an application operates across a network is also an interaction between the application and the local host and the interaction by its remote counterparts and their hosts as well as the interaction between the application's transport drivers and other concurrent applications that occur within the network

Analysing a typical network transaction

1. Query the DNS to translate a name to an IP address
   - This may involve repeated interactions between DNS forwarders and the hierarchy of servers
   - The elapsed time is a function of the DNS deployment, the domain name in question, the characteristics of the zone file and those of its parents and the state of DNS caches along the forwarding resolution path
2. Start the TCP session
   - 1 ½ RTT interval to complete the 3 way TCP handshake
3. Send the query
   - 1 RTT (data and ACK)
4. Receive data
   - TCP uses a congestion avoidance algorithm that starts slowly (1 packet per RTT) and then increases the number of packets in flight at each RTT interval until the server protocol window is exhausted or the network drops a packet. In most off-the-shelf host TCP implementations its restricted protocol memory buffers in the host that limit steady state transaction speed, not the network
   - TCP is designed to adapt its behaviour to share the network's resources across multiple active sessions – there is no fixed 'TCP Performance' metric
   - Deliberately driving the network path into packet loss is TCP's way of establishing the current point of maximum path capacity

Observations

- Most current network transactions take 15 RTT intervals
  - The most common transaction is a web page pull
  - i.e. the constraint is the latency between client and server, not necessarily bandwidth
  - One you have 'enough' bandwidth, more won't help
  - Altering the speed of light and/or reducing the radius of the planet and/or speeding up tectonic plate drift are about the most effective ways to universally improve network performance ☺
  - Many performance issues are the result of poor clients (insufficient memory, poor TCP stack, poor application design) and poor servers (insufficient memory, poor TCP stack, poor application design)
- A well tuned client and server should drive a network to the point of periodic packet loss
  - The resultant overall packet loss rate is a function of the average RTT and the average size of network transactions

How to improve "Performance"

- Tune your host and tune your server
  - Use a decent TCP stack with accurate timers
  - Increase protocol buffer size
  - Provide sufficient memory and CPU
  - Turn on windows scaling
  - Turn on SACK
  - Increase CWIN size
  - Turn on ECN
  - Turn off fragmentation
  - Use MSS discovery
  - Use a massive local MSS upper bound (9K)
- Tune the DNS
  - Use extended TTLs on zones where possible
  - Use caching forwarders
  - Use up to date and high quality DNS implementations
- Tune the Network
  - Reduce Latency
  - Use TCP-friendly queue management (RED) with tuned RED parameters
  - Turn on ECN
  - Tune queue sizes to correlate with delay-bandwidth product per link
  - Increase the MSS
  - Manage bandwidth resources

One possible approach to measurement…

- IETF IPPM Working Group has developed a number of useful standards that describe performance metrics and implementation approaches

- My preferred approach is a combination of the IPPM one way metric and the IPPM bulk transfer metric
  - Statistical sampling of selected paths using one way active probing with clocked data flows
  - Use TCP headers and TCP congestion control management to mimic end user flow behaviour
  - Correlate flow behaviour metrics with network events

Observations

- IP performance measurement is not a well understood activity with mature tools and a coherent understanding of how to interpret various metrics that may be pulled out from hosts and networks
- The complex interaction of applications, host systems, protocols, network switches and transmission systems is at best only weakly understood
  - But there's a lot of slideware out there claiming to provide The Answer!