Responsiveness under working conditions

Christoph Paasch <cpaasch@apple.com>
Randall Meyer <rrm@apple.com>
Stuart Cheshire <cheshire@apple.com>
What is a good Internet?
What is a good Internet?

*It’s all about the throughput!*

- Countless tools to measure capacity
- Internet offerings are mostly focusing on capacity
- Internet evolution from 56kbps to Mbps to Gbps
What is a good Internet?

It's all about the throughput!
What’s wrong with traditional Latency?

• Latency measurements are unrealistic
  - Artificial protocols (ICMP ping)
  - Unreal conditions (Idle or ideal conditions)

Need for a more realistic Latency / Responsiveness measurement
What is a good Internet?

- Protocol Conformance
- Latency/Jitter
- Throughput
- Responsiveness under working conditions
Responsiveness under working conditions
Measurement Principles

• Using today’s most common protocols (HTTP/2 and HTTP/3)
  - Thus, representative user-experience measurement

• Measure during “working conditions”
  - During active use of the network
  - Measures network’s ability to multi-task:
    • Provide low latency while also providing high throughput
  - Measures network’s agility:
    • Quickly react to changes in user-input
Measurement Methodology: Working conditions

• Goals:
  - Push network to its capacity limit, like any normal file transfer should
  - Maintain realistic traffic scenarios

• Methodology:
  - Multiple high-throughput bulk data transfers, acting as load-generating connections
  - HTTP/2 or HTTP/3
Measurement Methodology: Responsiveness probes

- **Goals:**
  - Measure latency in different user-representative scenarios
  - During working conditions
- **Methodology:**
  - HTTP GET requests for small (1-byte) objects
  - Measure on separate connections as well as load-generating connections
  - Collect HTTP request/response times
Responsiveness under working conditions

- networkQuality tool in macOS
- Open-source goreponsiveness (github.com/network-quality/goreponsiveness)
- iperf2 --bounceback
- Ookla Speedtest
Responsiveness under working conditions

PING ms

- Idle: 6
- Download: 3317
- Upload: 802
- Jitter: 4, 102, 90
What does Responsiveness measure?
What does Responsiveness measure?

“traditional” Latency
What does Responsiveness measure?
What does Responsiveness measure?

Responsiveness under working conditions
What causes low responsiveness

Data Transfers (email, web browsing,...)

Working Conditions

Increased Buffer Occupancy

High Latency / Low Responsiveness
Network-side Responsiveness
Network-side Responsiveness

- Transport protocols need to probe for capacity
- Probing fills buffers until they overflow
- Network buffers are too big
Network-side Responsiveness

- Goodput
- Buffer occupancy (added delay)
- Desired operating point
- Amount of data in flight

Today’s typical operating point
Optimal operating point

Need input from network to run at optimal operating point
L4S

- Low Latency, Low Loss, Scalable Throughput
- IETF Standard
- Mechanism:
  - Early notification on queue build-up
  - End-hosts react to keep queues

- Significant improvements in Responsiveness
L4S Interop at IETF 114, July 2022

• 15 Companies
• 7 bottleneck implementations (DOCSIS, Wi-Fi AP, 5G Emulator)
• Multiple end-host implementations

• Successful demo of ultra-low latency
What you can do
## What you can do

<table>
<thead>
<tr>
<th>Operators</th>
<th>Vendors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Measure Responsiveness in your network/equipment</td>
<td>Implement L4S in your equipment</td>
</tr>
<tr>
<td>ECN friendly configuration</td>
<td></td>
</tr>
<tr>
<td>Ask Vendors about L4S</td>
<td>Implement L4S in your equipment</td>
</tr>
<tr>
<td>Enable L4S</td>
<td>Participate in interop events</td>
</tr>
</tbody>
</table>

**References**

L4S: Ultra-Low Queueing - https://l4s.net
Server-side Responsiveness
Server-side Responsiveness
Server-side Responsiveness
Server-side Responsiveness
Server-side Responsiveness

Frontend

Backend

HTTP
TLS
Transport
Congestion Control
Server-side Responsiveness

- Significant server-side contribution to responsiveness issues
- Many opportunities for over-buffering
  - Transport layer (TCP)
  - Security layer (TLS)
  - Session layer (HTTP/2)
  - Multiple termination layers (TCP-termination, TLS-termination, HTTP-termination)
User-impact of server-side responsiveness issues

- Video streaming
  - Skipping forward in movie streams
  - Adaptive Bitrate Algorithms (e.g., MPEG-DASH and HLS)

- Interleaved HTTP streams
  - Larger HTTP stream combined with shorter latency-sensitive stream

- Faster web browsing, stock quotes, weather forecasts, driving directions, ...
Improving Server-side responsiveness

• Reduce buffering at the transport layer
  - Minimize backlog of unsent data using TCP_NOTSENT_LOWAT

• Reduce buffering at the application layer
  - Above socket layer
  - Within TLS
  - Within HTTP/2

• Buffers can be everywhere!
What you can do
What you can do

• Server-side responsiveness has concrete impact on user-experience

• Recommendation:
  - Configure your server with specific URLs to support responsiveness tests
  - Run responsiveness test
    - networkQuality (macOS)
    - go responsiveness (open-source)
  - Tune server buffer allocations to improve responsiveness
  - Retest
Resources

Responsiveness measurement
Open-source responsiveness measurement - https://github.com/network-quality/goreponsiveness
Server-side configurations - https://github.com/network-quality/server

L4S
L4S: Ultra-Low Queueing - https://l4s.net
Closing Remarks

• Better responsiveness gives a better user experience
• New tools for measuring responsiveness
• New techniques to improve responsiveness
  - L4S
  - Server buffer management