Responsiveness under Working Conditions
draft-ietf-ippm-responsiveness-01

Christoph Paasch, Randall Meyer, Stuart Cheshire, Omer Shapira, Matt Mathis
Upcoming update

Major changes

• Issue #62 (Flaw in “Working Conditions” algorithm)
  • Algorithm should terminate when delay stops increasing, not when throughput stops increasing

• Issue #17 (Use well-known URIs)
  • Default well-known URI that all web servers can supply if they wish
Issue #62: Problem statement

1 Gb/s
256 ms RTT
Bandwidth-Delay Product (BDP) = 32 MB
**Issue #62: Problem statement**

- **Bandwidth-Delay Product (BDP):** 32 MB
- **1 Gb/s**
- **256 ms RTT**

### Table: Time and Goodput

<table>
<thead>
<tr>
<th>Time</th>
<th>Buffer Occupancy</th>
<th>Goodput</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create 4 connections</td>
<td>64MB</td>
<td></td>
</tr>
<tr>
<td>4 Connections: 4 * 4MB = 16MB</td>
<td>64MB</td>
<td>1 Gb/s</td>
</tr>
</tbody>
</table>

### Diagram:

- **Maximum Goodput:**
- **Maximum Buffer occupancy:**
**Issue #62: Problem statement**

1 Gb/s
256 ms RTT
Bandwidth-Delay Product (BDP) = 32 MB

- Create 4 connections
  - 4 Connections: \(4 \times 4\text{MB} = 16\text{MB}\)
- Add more connections
  - 8 Connections: \(8 \times 4\text{MB} = 32\text{MB}\)

- Maximum Buffer occupancy
  - 64MB

- Maximum Goodput
## Issue #62: Problem statement

**Bandwidth-Delay Product (BDP)** = 32 MB

### Measured Bufferbloat

<table>
<thead>
<tr>
<th>Time</th>
<th>Buffer Occupancy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create 4 connections</td>
<td>64MB</td>
</tr>
<tr>
<td>4 Connections: 4 * 4MB = 16MB</td>
<td></td>
</tr>
<tr>
<td>Add more connections</td>
<td></td>
</tr>
<tr>
<td>8 Connections: 8 * 4MB = 32MB</td>
<td></td>
</tr>
<tr>
<td>Add more connections</td>
<td></td>
</tr>
<tr>
<td>12 Connections: 12 * 4MB = 48MB</td>
<td></td>
</tr>
</tbody>
</table>

### Real Bufferbloat

- Maximum Buffer occupancy
- Maximum Goodput

### Graphical Representation

- 1 Gb/s
- 256 ms RTT
- Issue #62: Problem statement

---

IETF 115 - IPPM - draft-ietf-ippm-responsiveness-01 - Responsiveness under Working Conditions
Issue #62: Solution

- Continuously add 1 connection to the pool every second
- Continuously probe responsiveness every 100ms
- Monitor goodput and responsiveness evolution over time
- Once both stabilize, declare saturation and use final responsiveness result
Issue #17: Use well-known URIs

- Json-configuration allows to bootstrap a responsiveness test
- Servers may implement a default responsiveness measurement endpoint
- Well-known URI to host the json-configuration
  .well-known/responsiveness.json
Remaining issues

https://github.com/network-quality/draft-ietf-ippm-responsiveness/issues

• Issue #55: Specify confidence in the results
  • If stability is not reached within a certain timeframe, responsiveness can produce an “early” result with a lower confidence.

• Issue #66: Allow non-TLS measurements
  • Low-end devices don’t have sufficient CPU to load the link with TLS traffic