Responsiveness under Working Conditions

draft-ietf-ippm-responsiveness-01

Christoph Paasch, Randall Meyer, Stuart Cheshire, Omer Shapira, Matt Mathis
Update from 00 to 01

• Closed 11 github issues
• Merged 7 PRs
• Significant changes:
  • Added DNS-Based Service Discovery “_nq._tcp”
  • Server-side example configurations in the appendix
  • Significant rework of the measurement algorithm
  • Wording,…
Update from 00 to 01

- Closed 11 github issues
- Merged 7 PRs
- Significant changes:
  - Added DNS-Based Service Discovery “_nq._tcp”
  - Server-side example configurations in the appendix
  - **Significant rework of the measurement algorithm**
  - Wording,…
-00 algorithm

Create 4 connections
Max goodput reached
Add more connections
Max goodput reached
Add more connections
No change in goodput
Declare Saturation

Probes on separate connections
Probes on load-generating connections

Time
Goodput
-00 algorithm - problems

- Small sample-size
- On extreme bufferbloated links, latency-probes easily time out
- One-shot measurement may be impacted by short-term low buffer occupancy
  - “synchronized” packet-loss issue
The algorithm is as follows:

1. Create 4 connections
2. At each 100ms:
   - Probes on separate connections
   - Probes on load-generating connections
3. Max goodput reached:
   - Add more connections
   - No change in goodput
4. Declare Saturation
-01 algorithm

4 data-sets:

- tcp_foreign = {...}  
  90th Percentile:  
  Weighting: 1/6  
  Normalizing to RPM: \[
  60000 / [s] = \text{[RPM]}
  \]

- tls_foreign = {...}  
  90th Percentile:  
  Weighting: 1/6  
  Normalizing to RPM: \[
  60000 / [s] = \text{[RPM]}
  \]

- http_foreign = {...}  
  90th Percentile:  
  Weighting: 1/6  
  Normalizing to RPM: \[
  60000 / [s] = \text{[RPM]}
  \]

- http_self = {...}  
  90th Percentile:  
  Weighting: 1/2

Responsiveness = \[
\frac{p^{90}(\text{tcp\_foreign})}{6} + \frac{p^{90}(\text{tls\_foreign})}{6} + \frac{p^{90}(\text{http\_foreign})}{6} + \frac{\text{http\_self}}{2}
\]
-01 algorithm

- Very large sample-size (about 150 data-points for a 15-second test)
- Less timeout issues as probing happens right from the start
- Not susceptible to short-term fluctuations in buffer occupancy

- Implemented in macOS Ventura networkQuality tool
- Implemented in open-source goresponsiveness tool
Remaining issues

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

• Issue #17: Use well-known URI for json-config
• Issue #63: Explain the impact of congestion control
• Issue #55: How to evaluate “confidence” of the result
• Issue #66: Allow non-TLS measurements
• Issue #62: Flaw in “Working Conditions” algorithm
Remaining issues

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

- Issue #17: Use well-known URI for json-config
- Issue #63: Explain the impact of congestion control
- Issue #55: How to evaluate “confidence” of the result
- Issue #66: Allow non-TLS measurements
- Issue #62: Flaw in “Working Conditions” algorithm
Issue #62: Flaw in “Working Conditions” algorithm

![Diagram showing the issue with the "Working Conditions" algorithm.]

- Create 4 connections
- Max goodput reached
- Add more connections
- Max goodput reached
- Add more connections
- No change in goodput

Buffer Occupancy:
- 4 Connections: 4 * 4MB = 16MB
- 8 Connections: 8 * 4MB = 16MB
- 12 Connections: 12 * 4MB = 48MB

Measured Responsiveness

Real Responsiveness
Issue #62: solution
Issue #62: solution

Need to saturate not only goodput but also “responsiveness”

Once Goodput **AND** Responsiveness stop changing, we declare “saturation”

In algorithm terms:

Add connections as long as either goodput increases

**OR**

responsiveness decreases
Other news

- Open-source goresponsiveness is evolving rapidly
  - Contribute at https://github.com/network-quality/goresponsiveness
- Ookla Speedtest measures “Loaded Latency”