Bufferbloat Detection In Network Paths

Danny Groenewegen[1], Harald Kleppe[1]

[1]System and Network Engineering
Universiteit van Amsterdam

Michiel Leenaars[2]
[2]NLnet Foundation

30^{th} of June 2011
RP2 presentations

Bufferbloat

Dark buffers in the Internet

Bufferbloat is the existence of excessively large (bloated) buffers in systems, particularly network communication systems.

- Every network protocol is affected
  - Many of them rely on timely arrival of packets
- The problem was already discussed in RFC 970
- Cheaper memory, increasing buffers
- Increasing traffic

Detecting bufferbloat

Research question

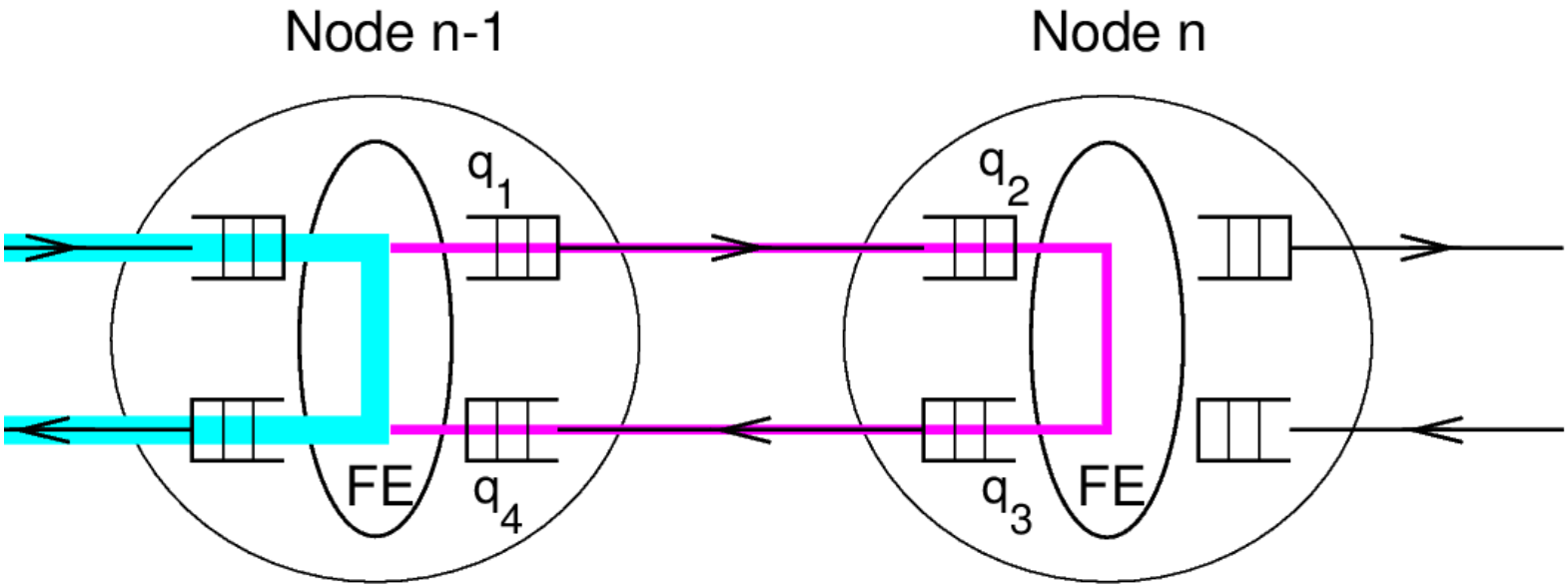
Is it possible to determine which device in a path causes bufferbloat?

- Is it possible to quantify the effects of bufferbloat?
- Can the same methods be applied on the Internet, where they are influenced by other unpredictable traffic?

Measuring link characteristics with pathchar

Probing each hop in a path

- Vary TTL to probe each node in a path
- Difference between $n-1$ and n gives link characteristics

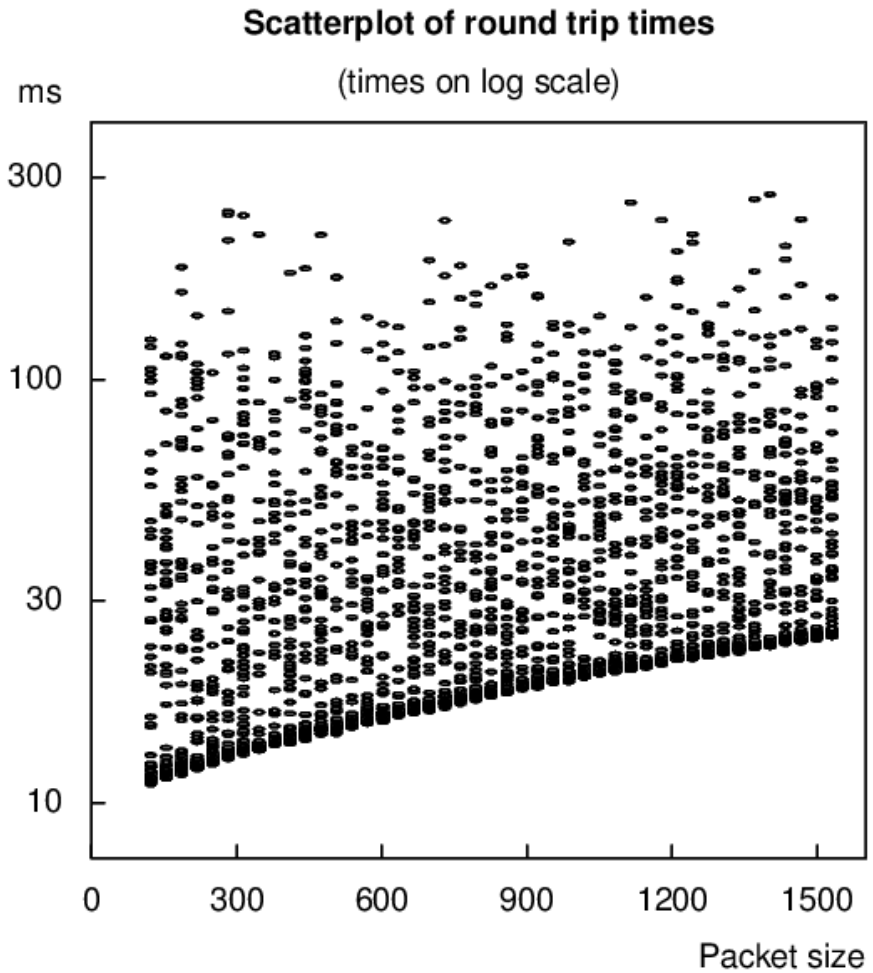


1

[1] Allen Downey. Using pathchar to estimate Internet link characteristics; ACM SIGCOMM, pages 241-250, 1999

Measuring link characteristics with pathchar

Estimate bandwidth and latency



- Varying packet sizes gives bandwidth and latency
- Multiple repetitions give queuing and loss information

Estimating queue delays

From pathchar to measuring bufferbloat

- Improve the accuracy of queue delay estimates
  - Probing each hop at (almost) the same time
  - Only probing with one packet size
- Observing the minimum RTT in a congested path
- Bufferbloat effects are not always visible
  - Measure with five minute intervals

Quantifying bufferbloat

Bufferbloat scoring model

```
grade = 0

for each measurement interval:
    median = median of queue delays
    grade += (median/0.010) / number of intervals

if grade > 10:
    grade = "10+"
```

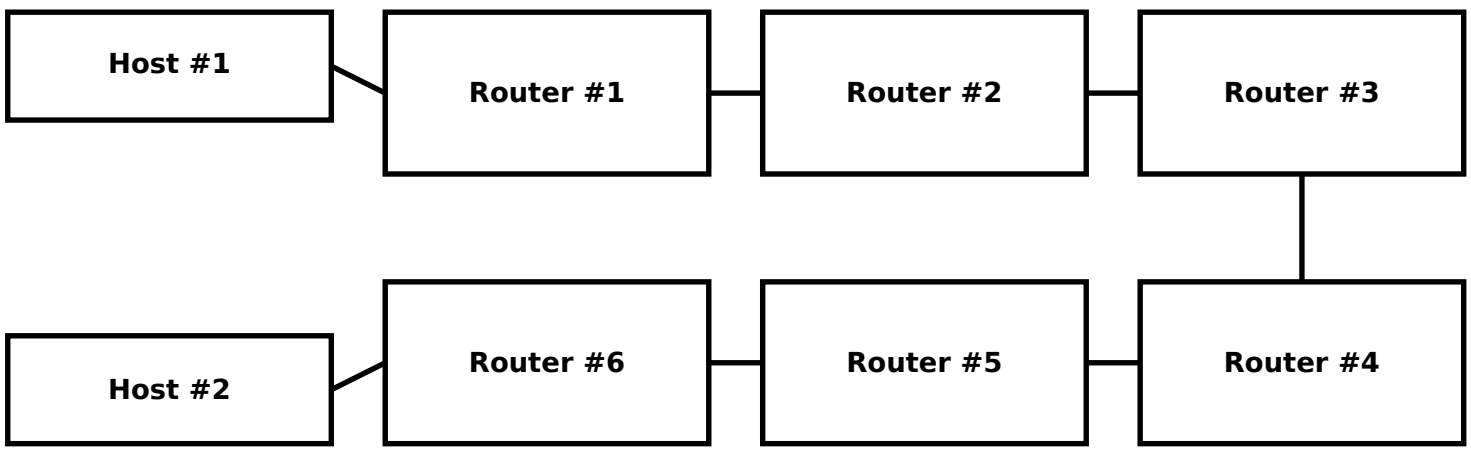
Tests and results

- Simulated experiments in a lab setup
- Experiments on the Internet
- Validate the tool
  - Detect queue distribution in a path
  - Compare bufferbloat scoring

Results

Lab test #1

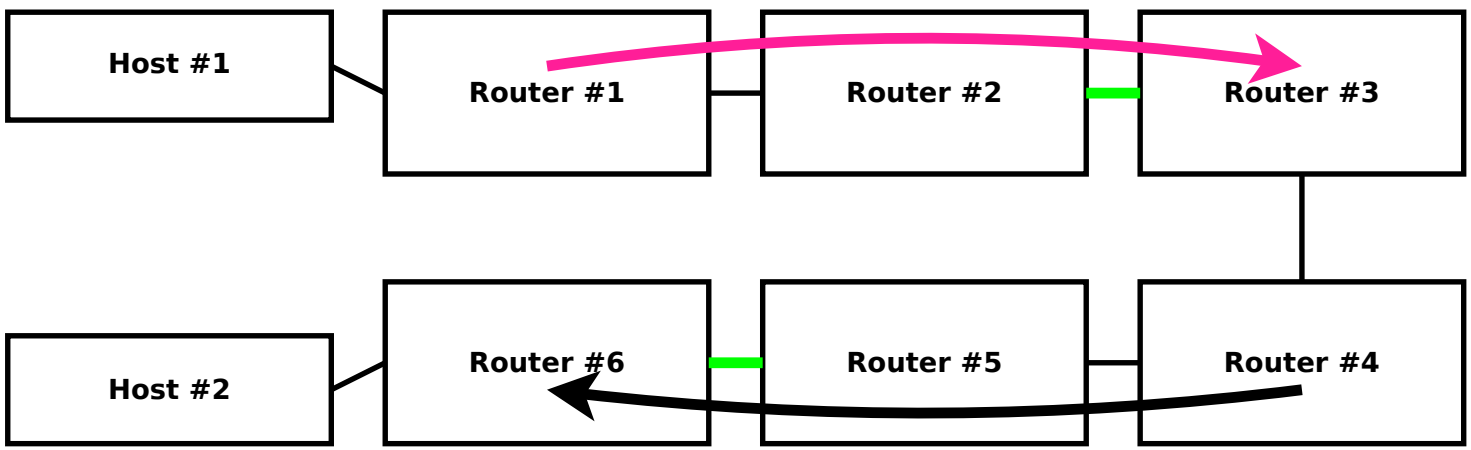
- 10 and 100 Mbit bottlenecks
- Saturate bottlenecks
- Identify second bottleneck behind an intermediate bottleneck



Results

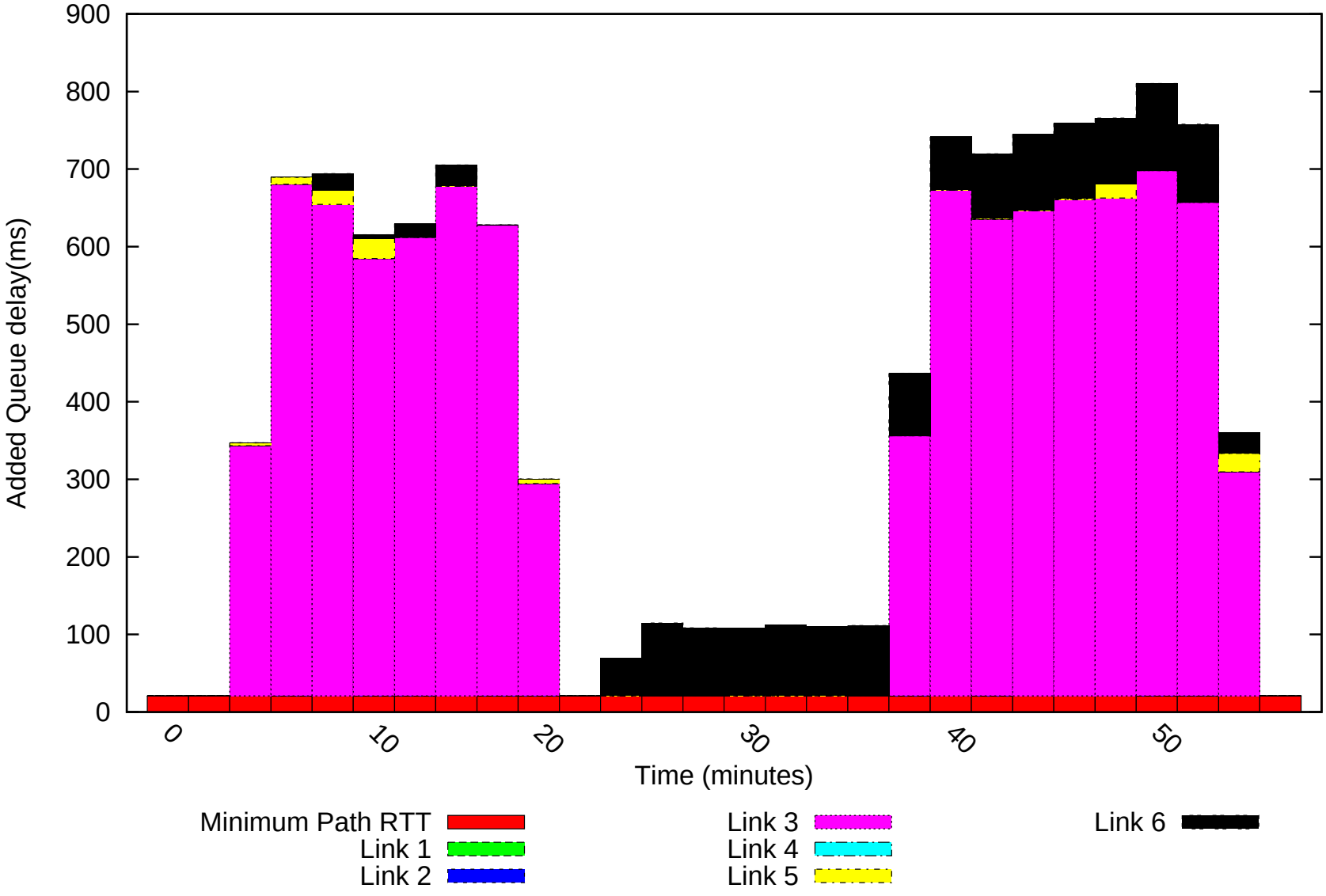
Lab test #1

- 10 and 100 Mbit bottlenecks
- Saturate bottlenecks
- Identify second bottleneck behind an intermediate bottleneck



Results

Test setup; Link 3: 10Mb/s; Link 6: 100Mb/s; Bufferbloat score: 10+



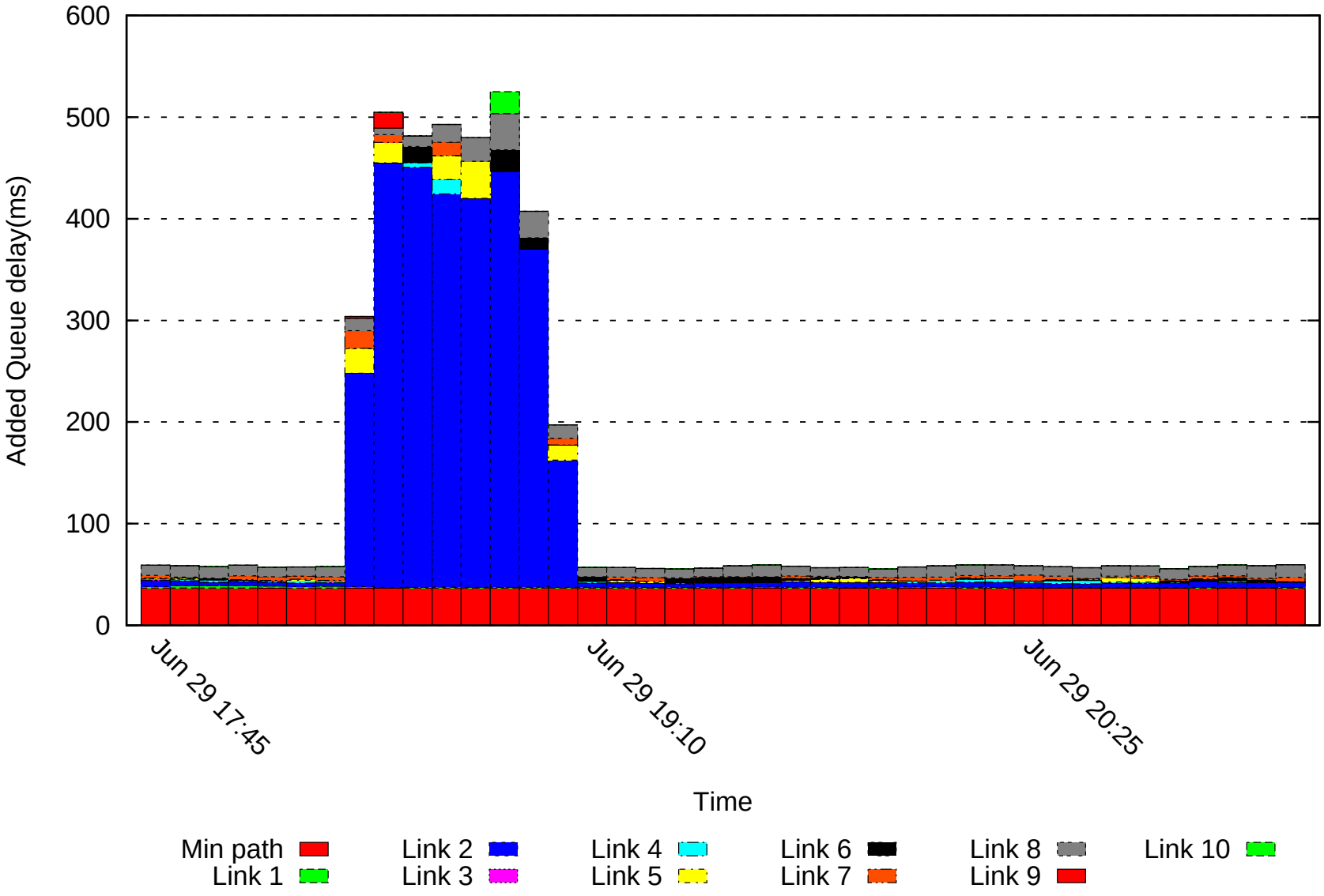
Results

Consumer ISP connection; Bufferbloat score: 5.0

- Probing from Diemen to Norway
- Saturating traffic from OS3 lab
- 25 / 1.5 Mbit consumer grade connection

Results

Consumer ISP connection



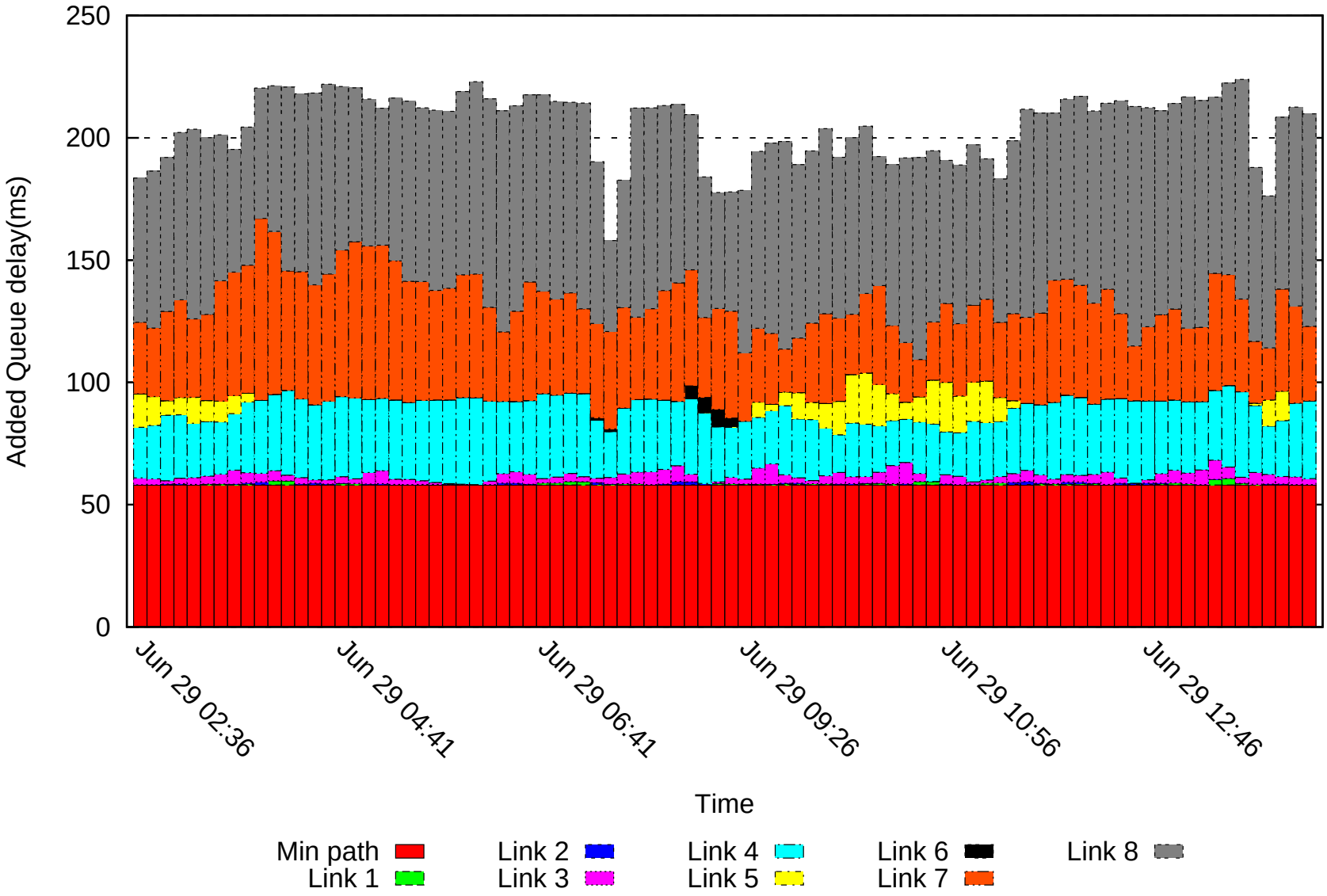
Results

Chicago - Tokyo

- From a VPS in *Chicago*
- Towards a RIPE TTM host in *Tokyo*
- Eight hop path, trans-pacific

Results

Chicago - Tokyo: Bufferbloat score: 8.6



Results Summary

- Queue delay can be determined per link
- Possible to observe near zero queue delay
- Generally slightly less queue delay on IPv6 links than IPv4

Conclusion

- Possible to detect bufferbloated link
  - Determining which queue
  - Layer 2 components
- Methods seem to work on the Internet
  - Scoring model needs fine tuning

Questions?

Thanks for listening!

Tool: `https://github.com/hkleppe/Buffchar`

More info: `http://www.bufferbloat.net`