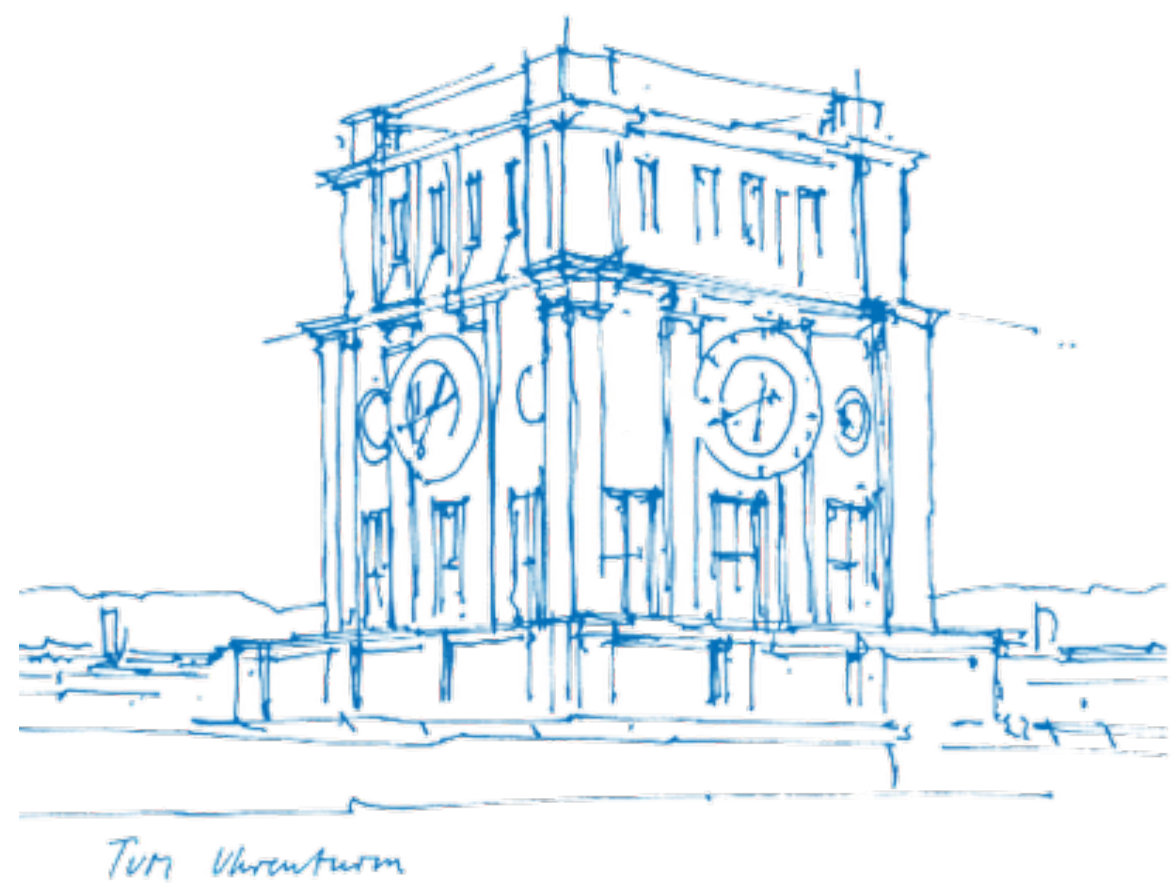
Precision and Accuracy of Packet Generators

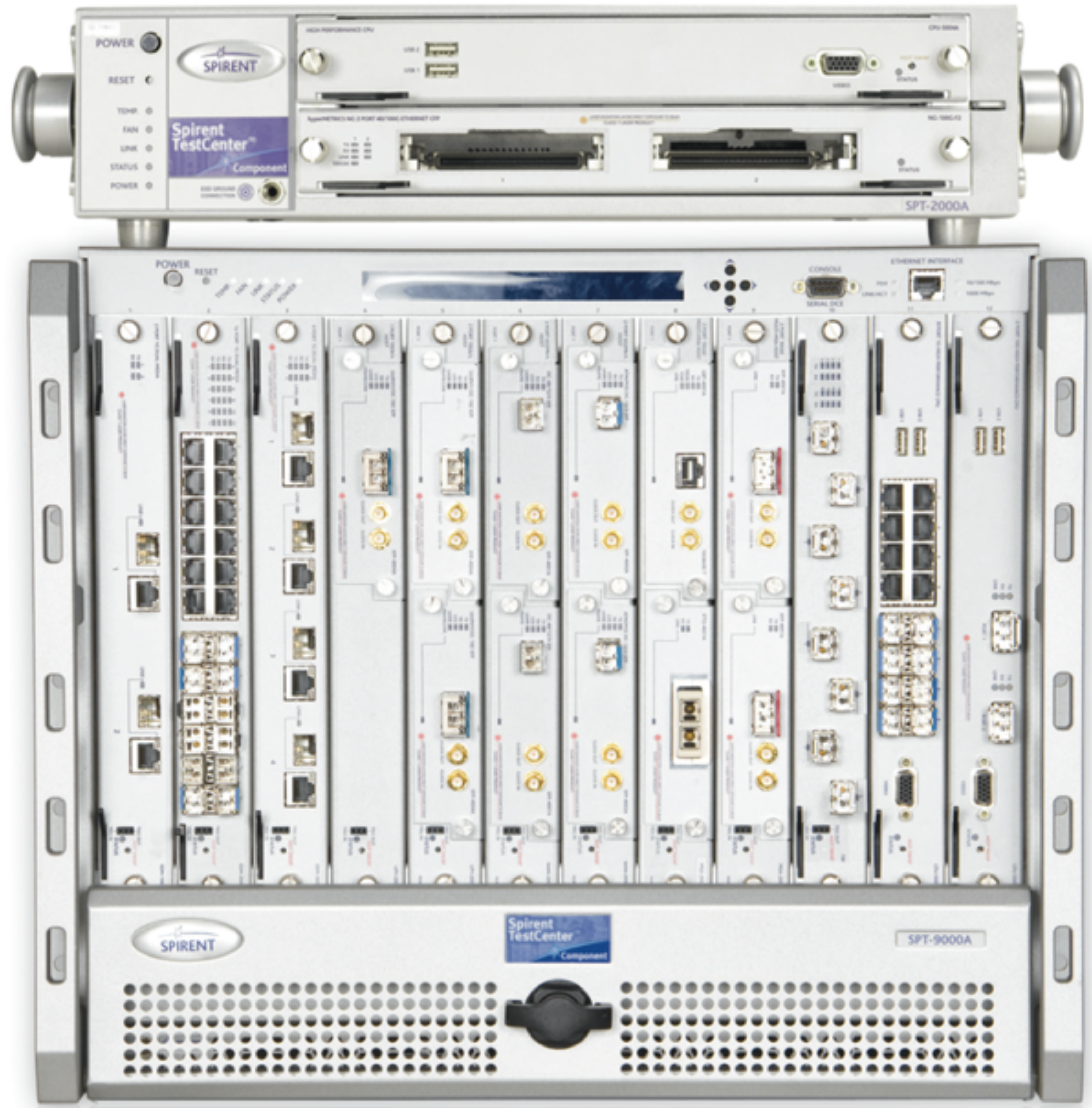
Who tests the testers?

Paul Emmerich, Sebastian Gallenmüller, Alexander Frank, Georg Carle
emmericp@net.in.tum.de
Technical University of Munich
Chair of Network Architectures and Services

IETF-102, 17.07.2018



Expensive packet generators: awesome!



Source: www.spirent.com

Cheap packet generators: awesome?

Source: www.intel.com

Chair of Network Architectures and Services
Department of Informatics
Technical University of Munich

What do you expect of your packet generator?

Let’s start with some questions:

- What should a packet generator be able to do?
- Can a cheap software-based packet generator be reliable?
- How can you validate that your packet generator works as advertised?
- Is your packet generator precise?
- Is it accurate?

What should your packet generator do?

Example: Main packet generator requirements from ETSI GS NFV-TST009
(Draft V0.0.13, 2018-07, Section 7.1 paraphrased)

- Accurately generate constant frame at specified rates
- Accurately generate bursty traffic at specified rates
- Support accurate latency measurements, timestamp applied "as close as possible to actual transmission"

Chair of Network Architectures and Services
Department of Informatics
Technical University of Munich

Accuracy vs. precision in latency measurements

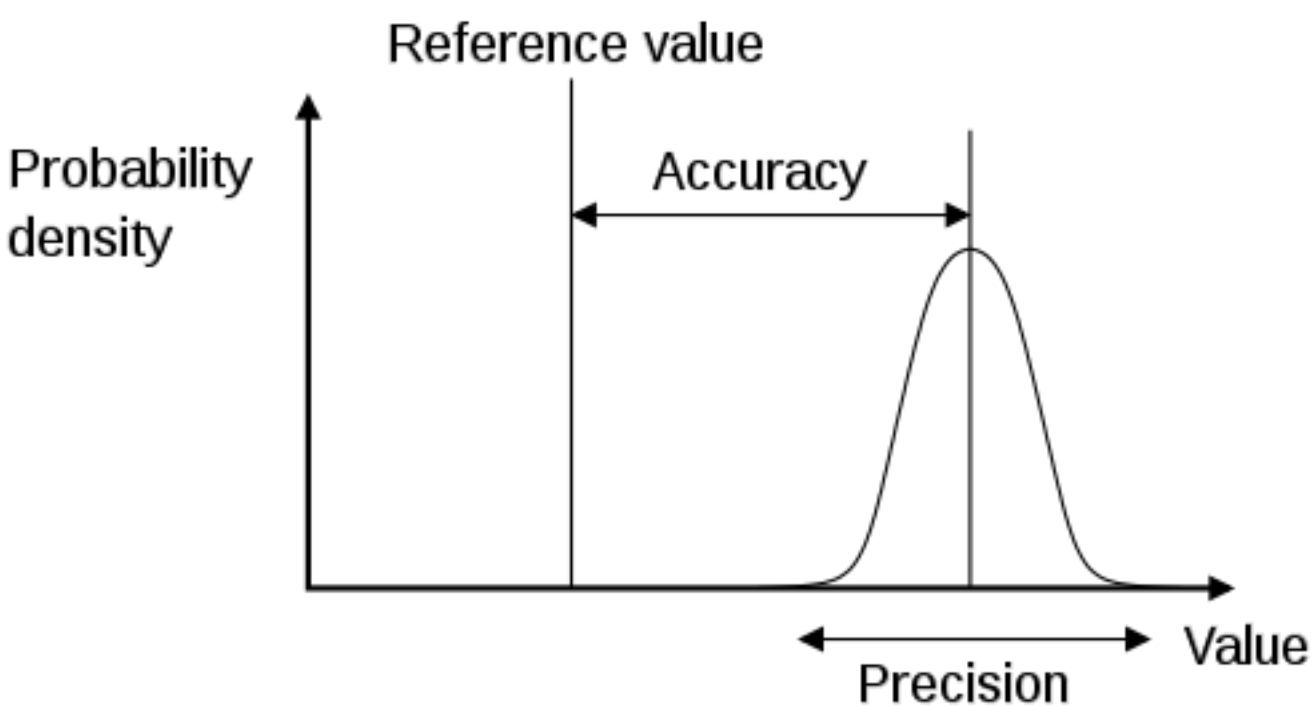
Example setup: a packet generator measuring the latency of a cable.

Precision: The deviation between the measurements is low

- The latency of the cable should not change
- Typical source of measurement error: queuing delays in generator included

Accuracy: The average reported latency is correct

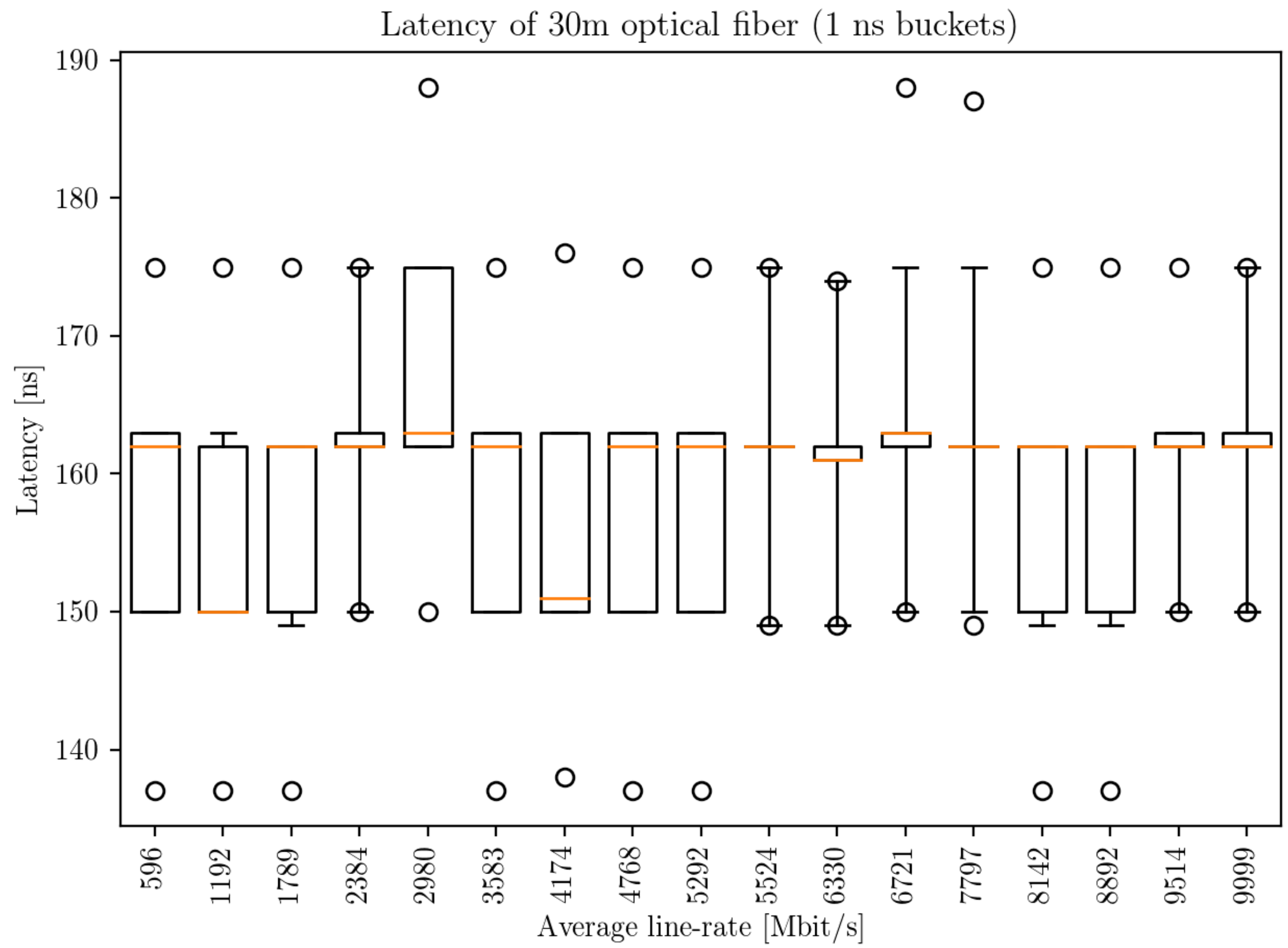
- The latency of a cable can be estimated from its length
- Typical source of measurement error: processing time in generator included



Source: https://commons.wikimedia.org/wiki/File:Accuracy_and_precision.svg

Chair of Network Architectures and Services
Department of Informatics
Technical University of Munich

Testing latency measurements: measure a cable



- Cable's latency should not change under increasing packet rate
- Validate with different cable lengths? (Only 30 meter single mode fiber here)
- Precision here: 37 nanoseconds
- Accuracy here: average reported latency is 161 nanoseconds
  - Estimated correct latency: 150 nanoseconds with 0.66c propagation speed
  - What to use as ground truth?

What are traffic patterns?

Traffic pattern: the way packets are spaced on the wire, most common are

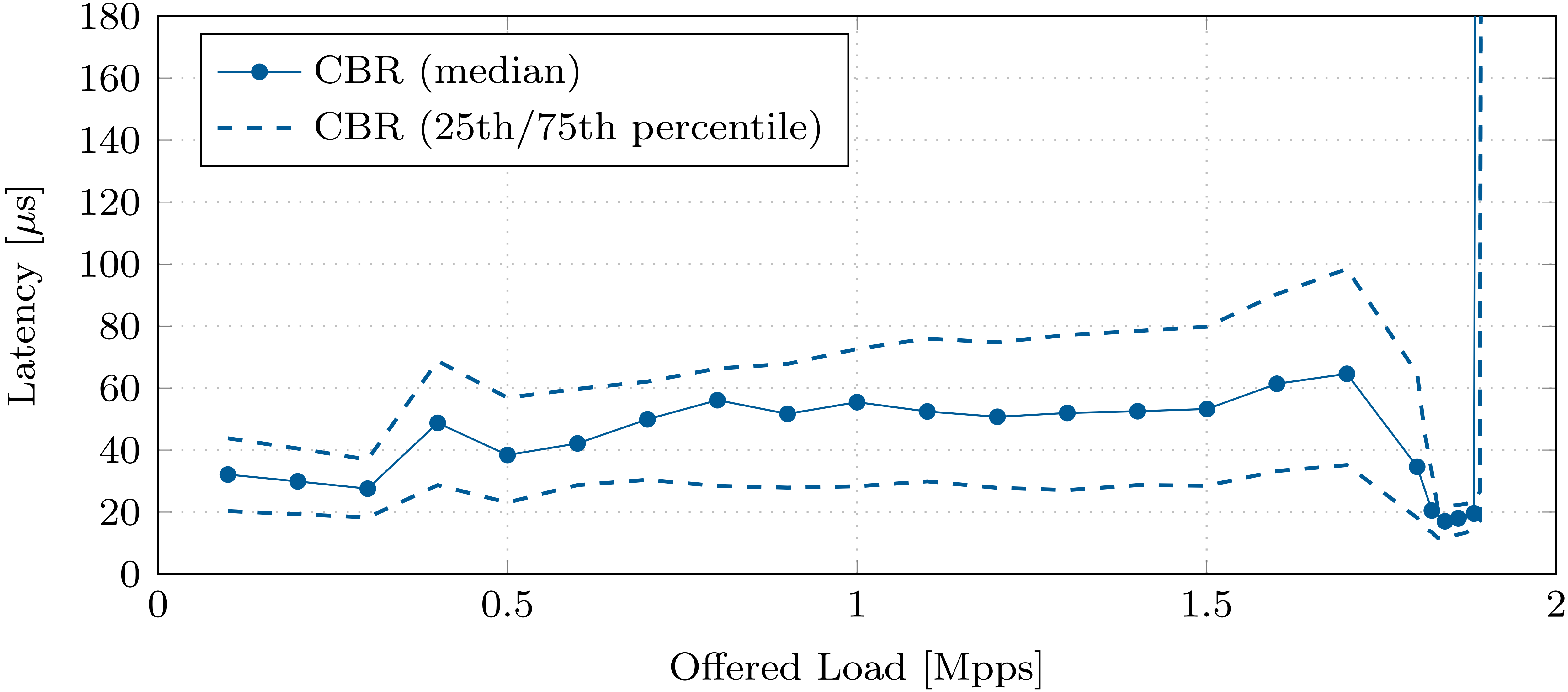
- Constant bit rate (CBR): same space between all packets
- Bursty: back-to-back packets followed by a longer gap
- Poisson: exponential distribution of delays

- **RFC 2544 wants CBR by default, also allows for further tests with other patterns**
- **ETSI GS NFV-TST009 wants CBR and bursty traffic**

- Software packet generators prefer bursty traffic (sometimes even if configured otherwise!)
- Bursty traffic is easiest to generate (NIC drivers work that way)
- CBR is hardest to generate (multi-core scaling is challenging without hardware support)
- Poisson is easy to scale (adding Poisson distributions yields a new poisson distribution)

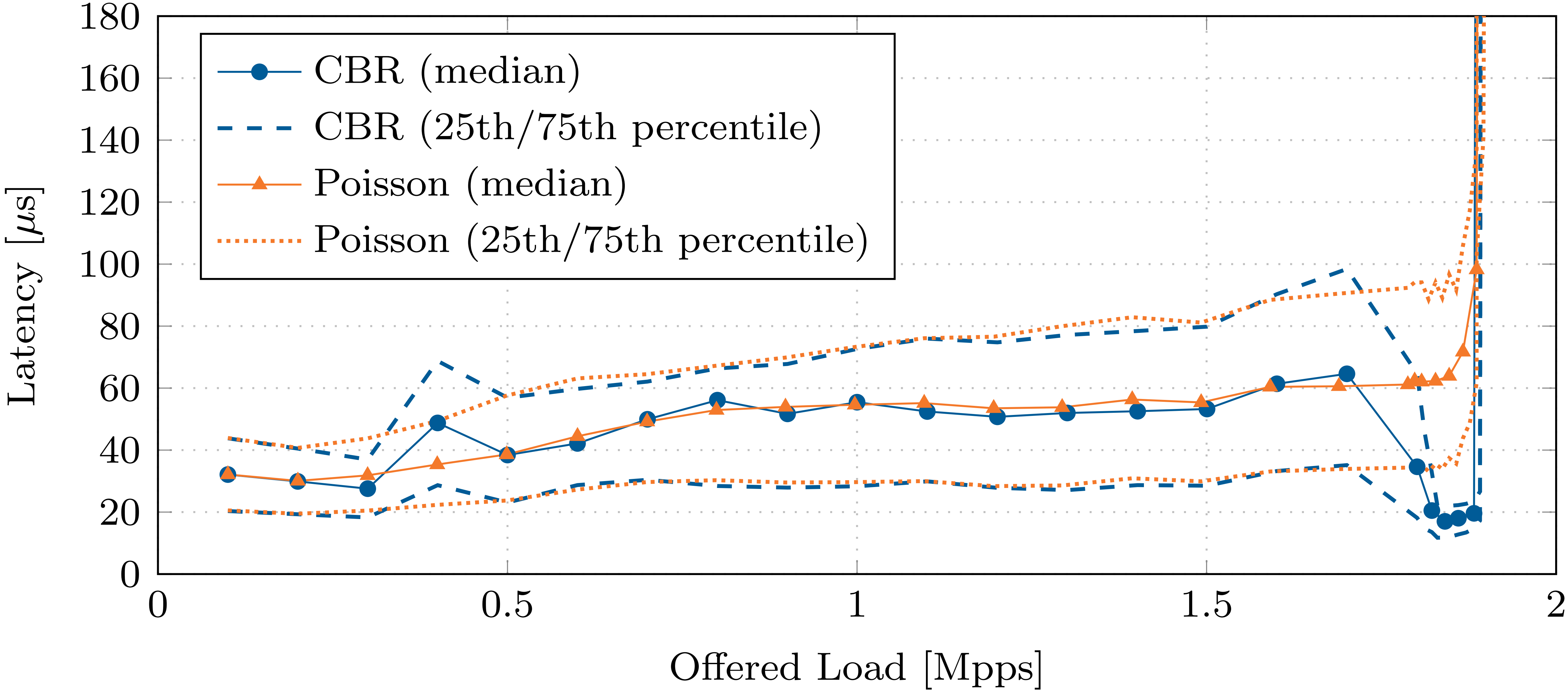
- Poisson is arguably most realistic, **CBR least realistic**

CBR can lead to weird effects



- Forwarding latency of Open vSwitch (kernel), increasing load
- Dynamic interrupt throttling (ixgbe driver) and poll-mode (NAPI) don't play well with CBR traffic

Real-world traffic isn't CBR



- Only change: time between packets
- Completely different response from the device under test

Summary: what to measure and how to benchmark?

Latency measurements (ideas)

- Measure the latency of cables of different length
- Report the minimum, maximum and average reported latency
- Repeat measurements with varying packet rates: does it get worse at high rates?
  - Queues filling up?
- What is the ground truth for the latency of a cable?

Traffic pattern measurements

- Measure packet arrival at device under test with high-precision timestamping
- Hard to measure with commodity hardware
- We have done some measurements using a NetFPGA

Require Poisson traffic? Or is bursty traffic close enough?

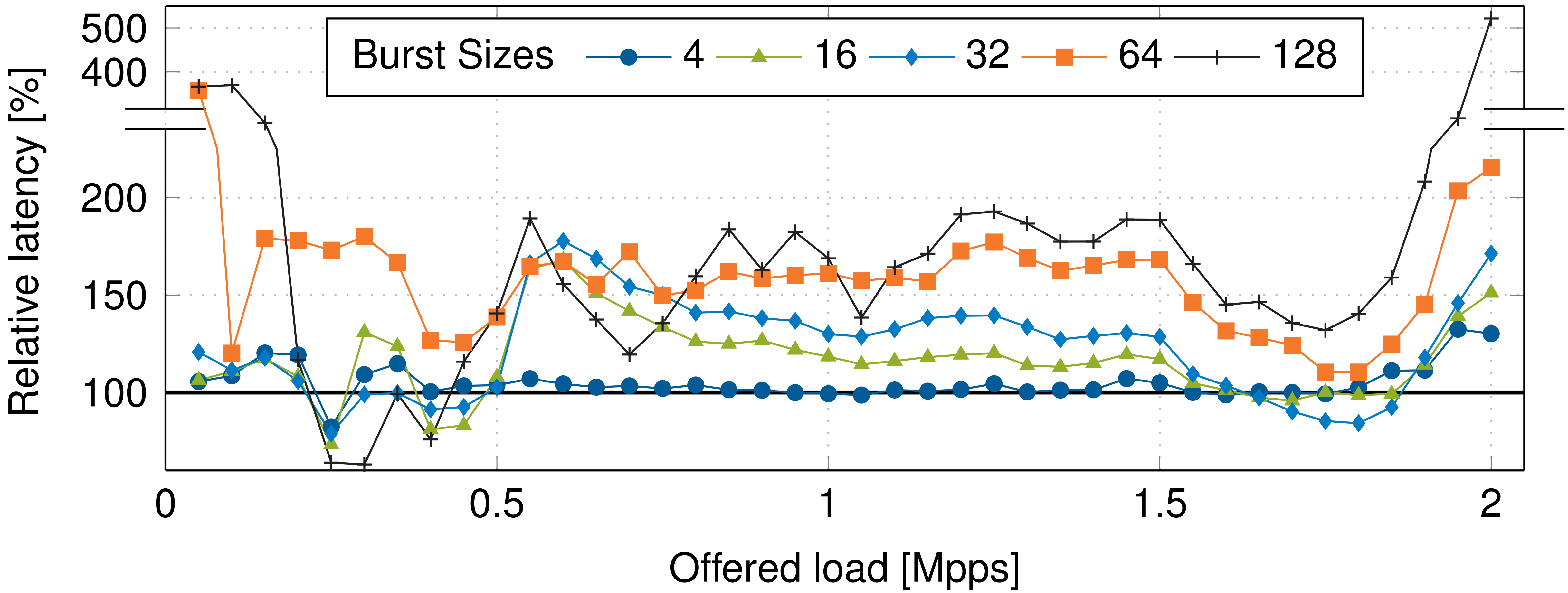
- CBR is not realistic

Lots of open questions, so …

Discussion?

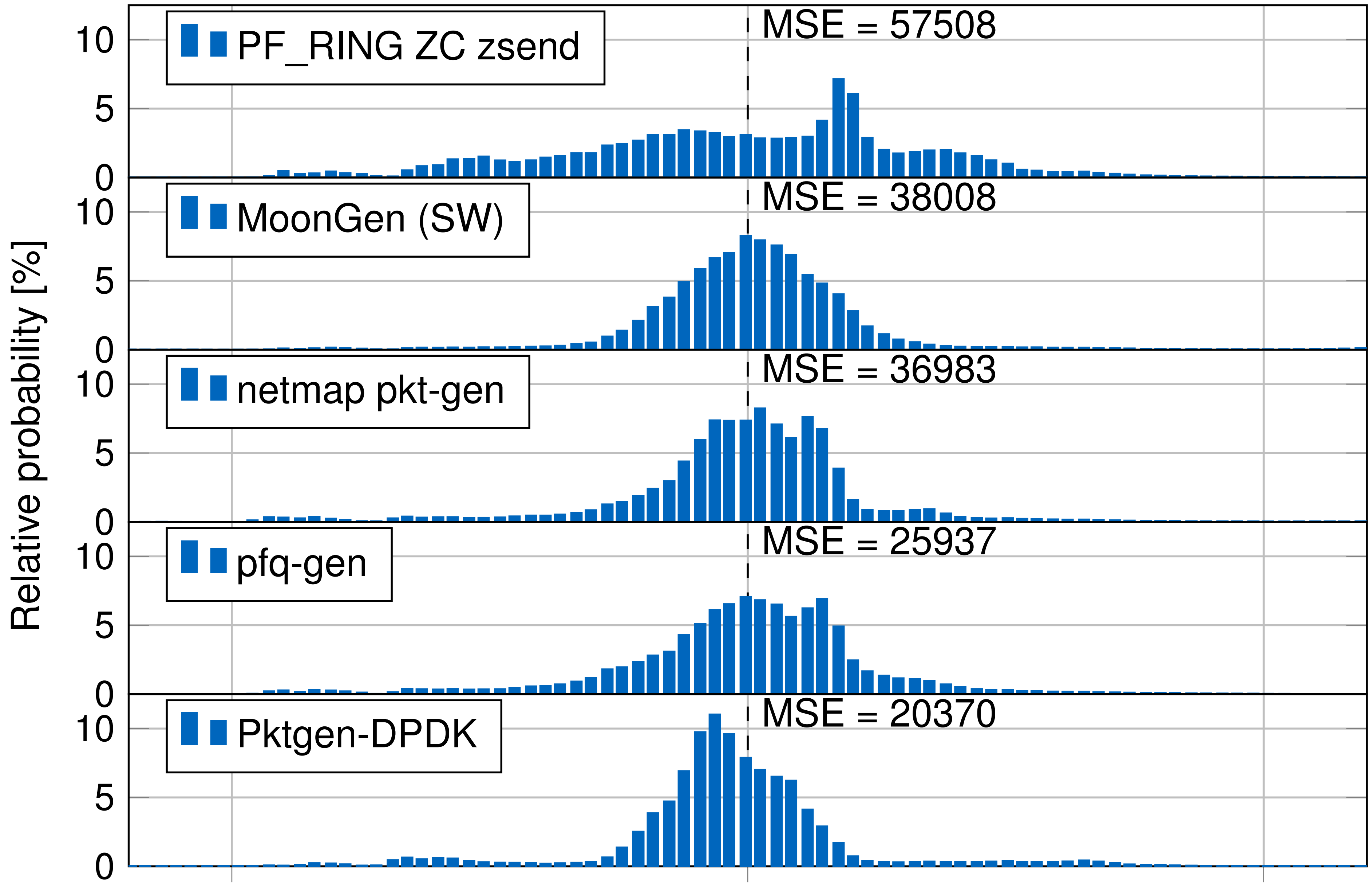
Backup: CBR/Burst Comparison

- Forwarding latency of Open vSwitch (kernel), increasing load
- Baseline latency: CBR traffic, varying burst sizes



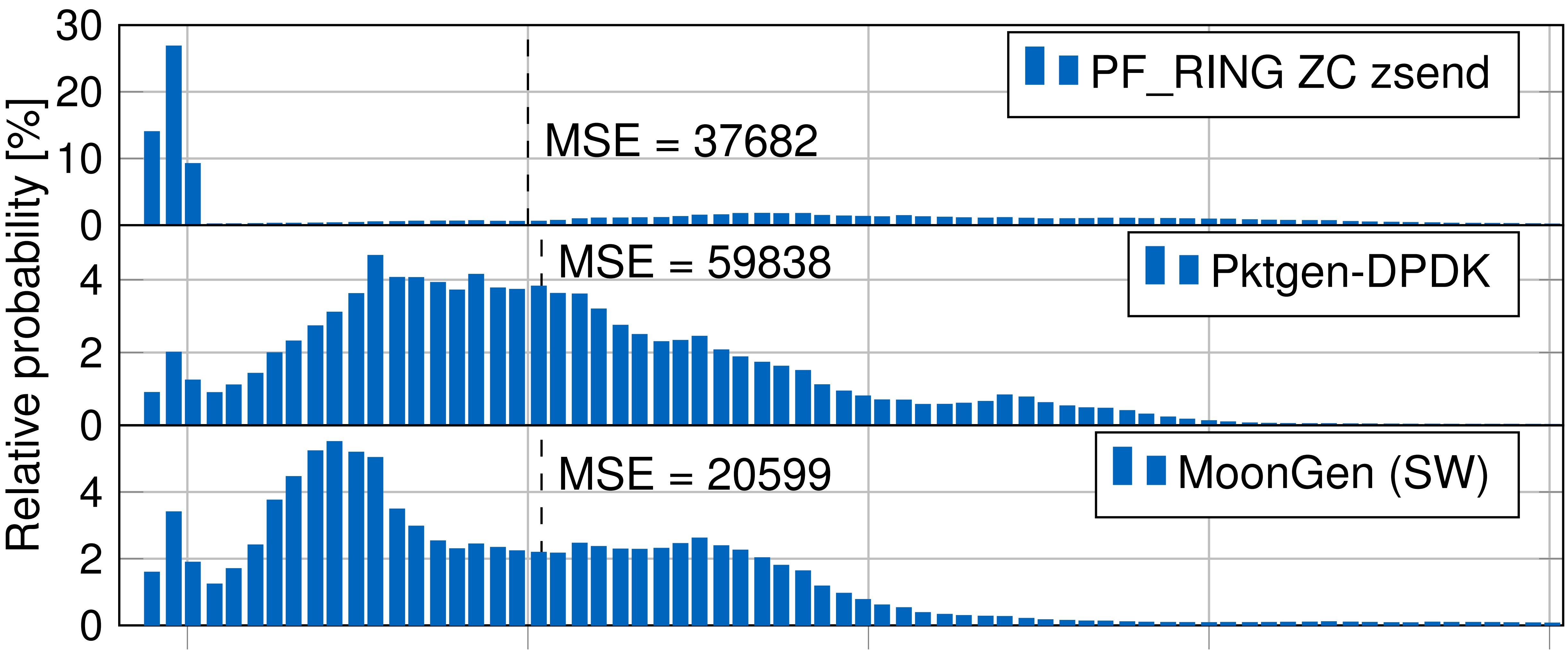
- Bursts are important for performance
- Typical default burst sizes: 16 to 256
- Packet generators often fail to generate CBR reliably

Backup: Rate Control Comparison



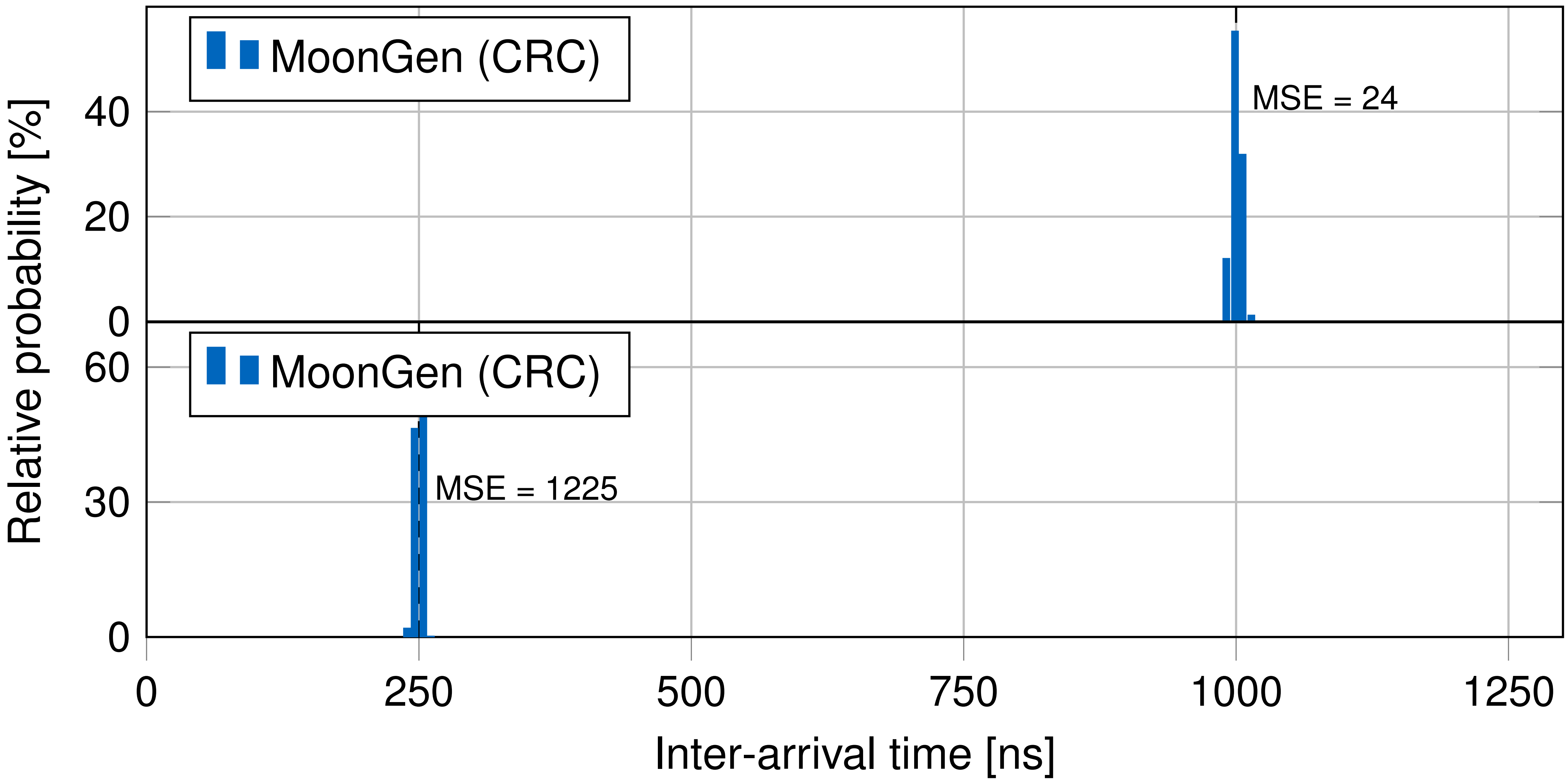
Target rate = 1 Mpps/1µs inter-arrival time

Backup: Rate Control Comparison



Target rate = 4 Mpps/0.25μs inter-arrival time
Most packet generators fail to generate this when configured without bursts

Backup: MoonGen Rate Control



1 and 4 Mpps, MoonGen SW rate control with corrupted packets filling the gaps

Backup: Latency measurements