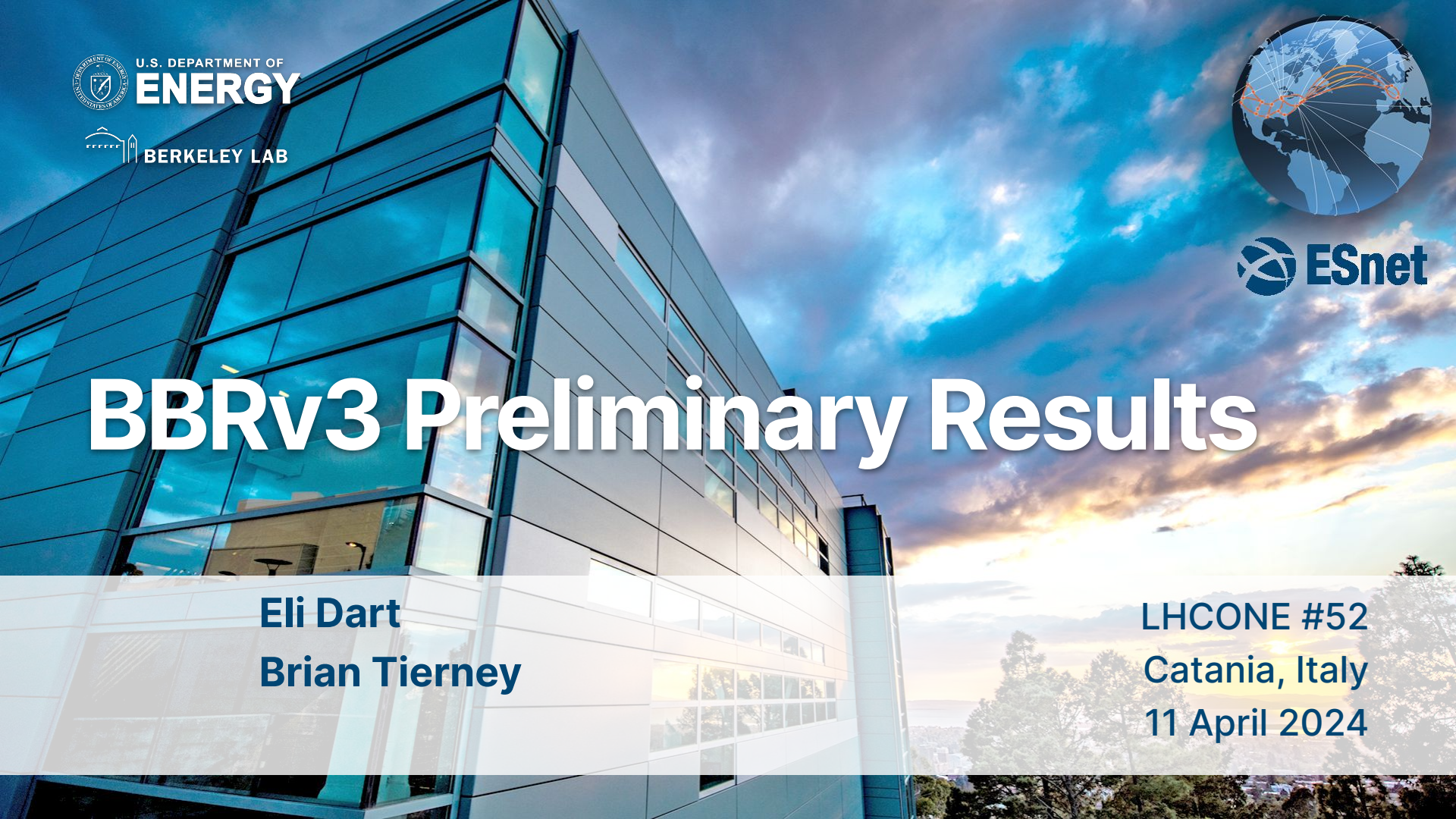


BBRv3 Preliminary Results

Eli Dart
Brian Tierney

LHCONE #52
Catania, Italy
11 April 2024

Motivation

- Understand how BBR behaves for R&E workloads
  - Specifically DTN / data transfer workloads
  - Relevant to other workloads too
- Understand implications for future deployments
  - Are there implications for routers, switches, etc?
  - What host configurations are useful?
- Spoiler: still more questions than answers

BBRv3: where are we?

- Improvements over BBRv2
  - BBRv2 was a ***significant*** improvement over BBRv1
  - Recommend against deploying BBRv1 in production
- BBR team is trying to get BBRv3 merged into mainline Linux kernel
  - They had hoped to have this done by now
- Unknown if BBRv3 will be the “final” version that makes it into production kernels
  - Appears likely, but not done yet

Cubic vs BBRv3

lbl-dev-dtn1.es.net to cern773-ps-tp.es.net (100G) RTT = 150ms

Streams	CC Alg	Pacing (Gbps)	Tput (Gbps)	Stddev (nvals)	RXMTs
1	bbr	0.0	**22.90**	3.58 (10)	568340
1	cubic	0.0	**19.50**	4.04 (10)	68051
8	bbr	0.0	**47.96**	2.43 (10)	300157
8	cubic	0.0	**30.64**	7.35 (10)	49778
8	bbr	12.0	**50.58**	2.21 (10)	302178
8	cubic	12.0	**44.08**	3.58 (10)	61393

Things to note:

- Throughput is slightly better with BBR on this path
- Results are more consistent for BBR (stddev lower) for 8 stream tests
- Pacing helps with parallel CUBIC flows
- BBR typically has about 10x more retransmits than CUBIC

Cubic vs BBRv3
lbl-dev-dtn1.es.net to pygrid-sonar2.lancs.ac.uk
(4×10G, RTT = 147ms)

Streams	CC Alg	Pacing (Gbps)	Tput (Gbps)	Stddev (nvals)	RXMTs
1	bbr	0.0	**5.61**	0.84 (6)	19721
1	cubic	0.0	**2.60**	1.18 (6)	4966
8	bbr	0.0	**20.26**	0.90 (5)	525351
8	cubic	0.0	**8.34**	0.73 (5)	112758
8	bbr	11.0	**18.19**	1.32 (5)	685559
8	cubic	11.0	**8.46**	2.82 (6)	89142

Things to note:

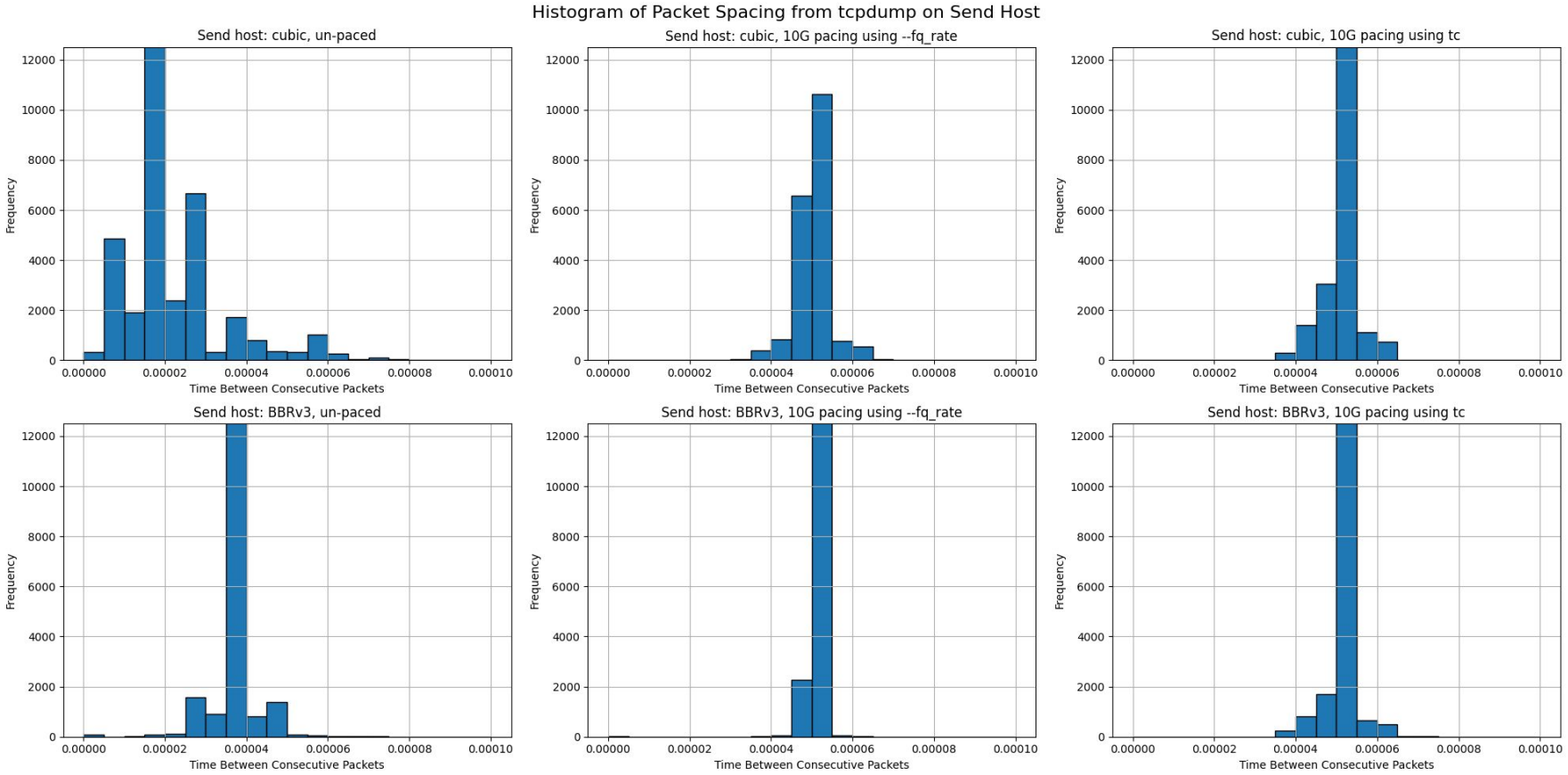
- Throughput is considerably better with BBR
- Low throughput due to receive host TCP window limited

Packet Spacing Histograms

BBR vs Cubic, with and without pacing

Q: Which CC Algorithm will do better with smaller router/switch buffers?

Packet gap histogram on send host: Cubic vs BBRv3, Unpaced vs Paced



Notes on Previous Slide

- pacing for these tests 10G
- `tcpdump -j adapter_unsynced --time-stamp-precision nano`
- BBR unpaced is similar to CUBIC paced
- BBR packet gaps are larger than CUBIC on the send host
- fq-rate and tc seem to be equivalent
- Pacing everywhere might be a good solution until BBR is everywhere?
- More testing is needed

Complications for Pacing Testing

- TSO and LRO are routinely used in modern systems
  - Significantly muddies the water when doing testing
  - Difficult to eliminate bursts completely
    - Pacing will always be imperfect
    - Does this matter?
- BBR takes TSO and LRO behavior into account
  - Perhaps it's OK to always have some burstiness?
- HighTouch has been very valuable
- There is more testing to be done

Conclusions

- ***These results are preliminary:*** more testing needed
- Unpaced BBR seems to be similar to CUBIC paced
  - BBR has pacing built in
- BBR helps on some paths
  - Only minor improvements over CUBIC on clean paths with large buffered devices
  - Need to find paths will small buffered devices to really see improvements
    - If you have a perfSONAR host behind a smaller buffer switch let us know
- Detailed analysis of pacing behavior is difficult: more work needed


U.S. DEPARTMENT OF
ENERGY
BERKELEY LAB
ESnet
Thanks!
Eli Dart
dart@es.net
https://my.es.net/
https://www.es.net/
https://fasterdata.es.net/