IBM MQ V9.1.4 with the fasp.io Gateway

Performance

Paul Harris - IBM Hursley

fasp.io Gateway with IBM MQ

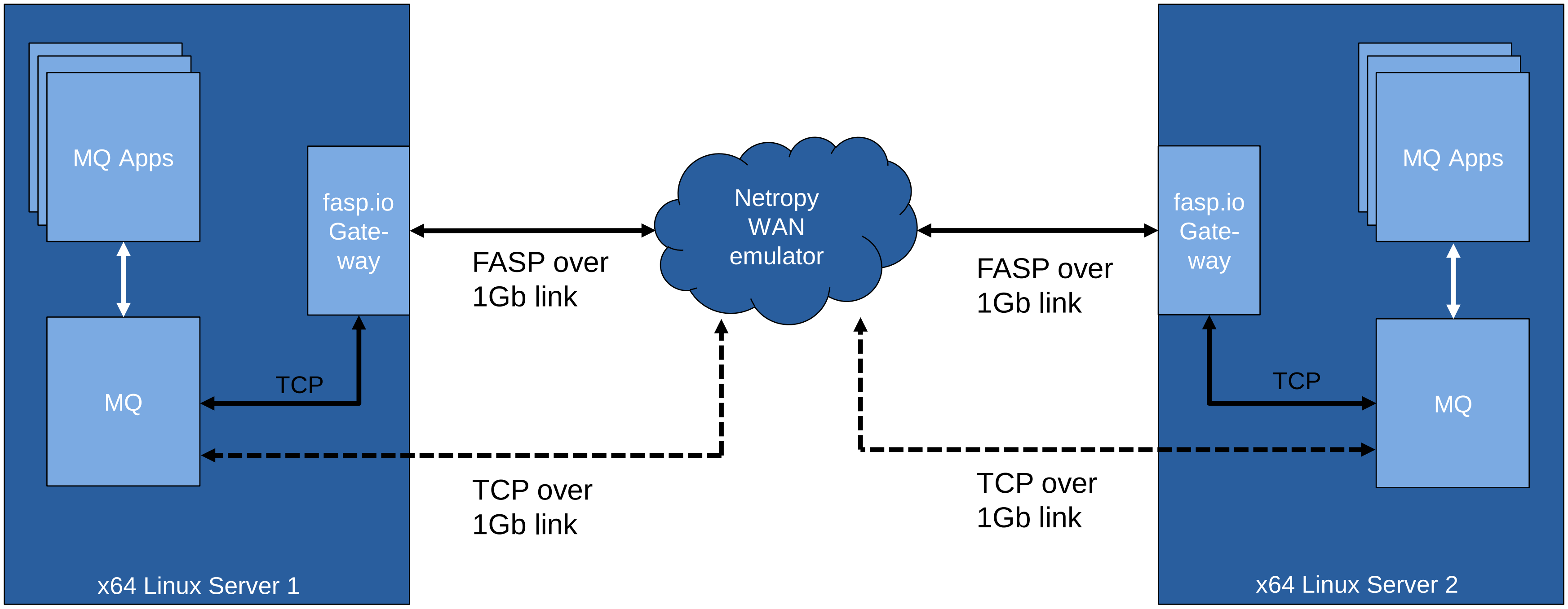
fasp.io Gateway is part of MQ V9.1.4 Advanced edition.

- Available as a download from passport advantage (part numbers: CJ6CBML, CC47WEN, CC47ZEN).
- Can dramatically improve throughput between queue managers, where links are across high latency and/or lossy networks (where traffic intended to be diverted through a single Gateway is <2Gb/s in total).
- Particularly suited to streaming workloads with larger messages.
- Distributed queueing considered in this presentation but other use cases can benefit (e.g. QREP).

Deployment Options

- Can be co-located on the same server as the queue manager or deployed on a separate host.
- Simply point the server channels at the pair of linked fasp.io Gateways instead of the remote queue manager listeners.
- No special setup or tuning of MQ is required.

Distributed Queueing Test Topology



fasp.io Gateway connected server channels ⟷ Direct connected server channels ⇠⇢

Workloads

Two workloads tested

- Point-to-point (uni-directional) send/receive workload utilizing 10 pairs of server channels, with multiple applications (MQ-CPH).
- Requester/responder (bi-directional) workload utilizing 10 pairs of server channels, with multiple applications (MQ-CPH).

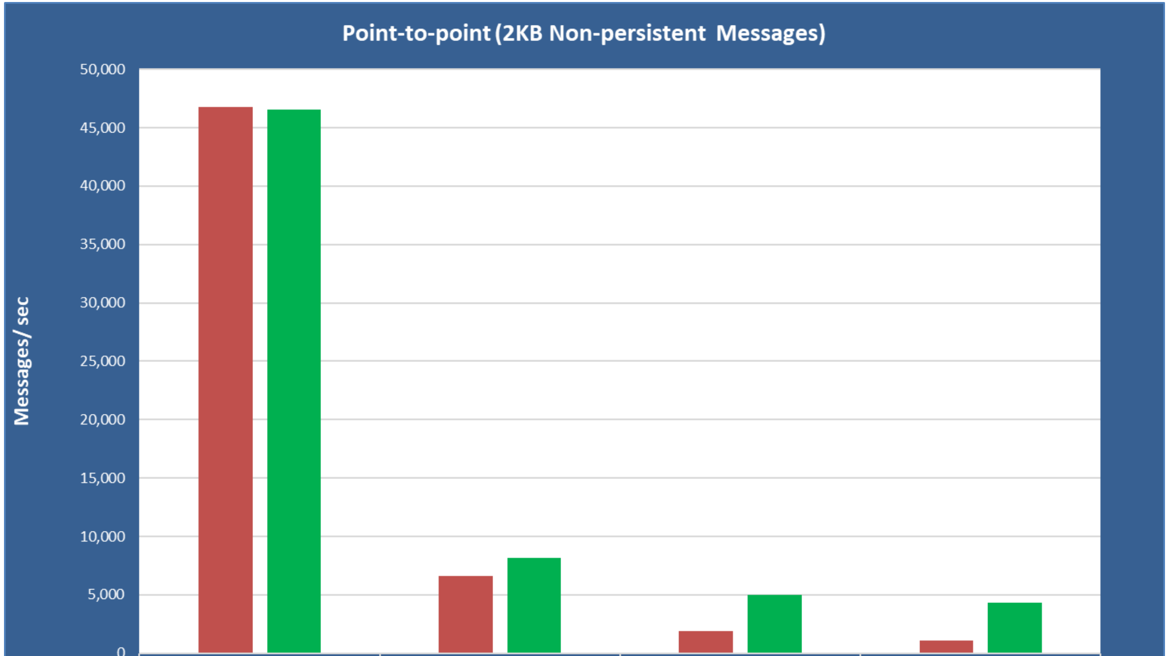
Workloads tested with different latencies and losses

- **N1:** 25ms network latency* (no packet loss)
- **N2:** 40ms network latency* (0.1% packet loss)
- **N3:** 50ms network latency* (0.5% packet loss)

*Applied to both directions, so round trip case will be 2x latency

Results

2KB Non-persistent Point-to-point Workload



	Bypass	N1	N2	N3
TCP 2KB	46,750	6,600	1,870	1,100
FASP 2KB	46,530	8,140	4,950	4,290

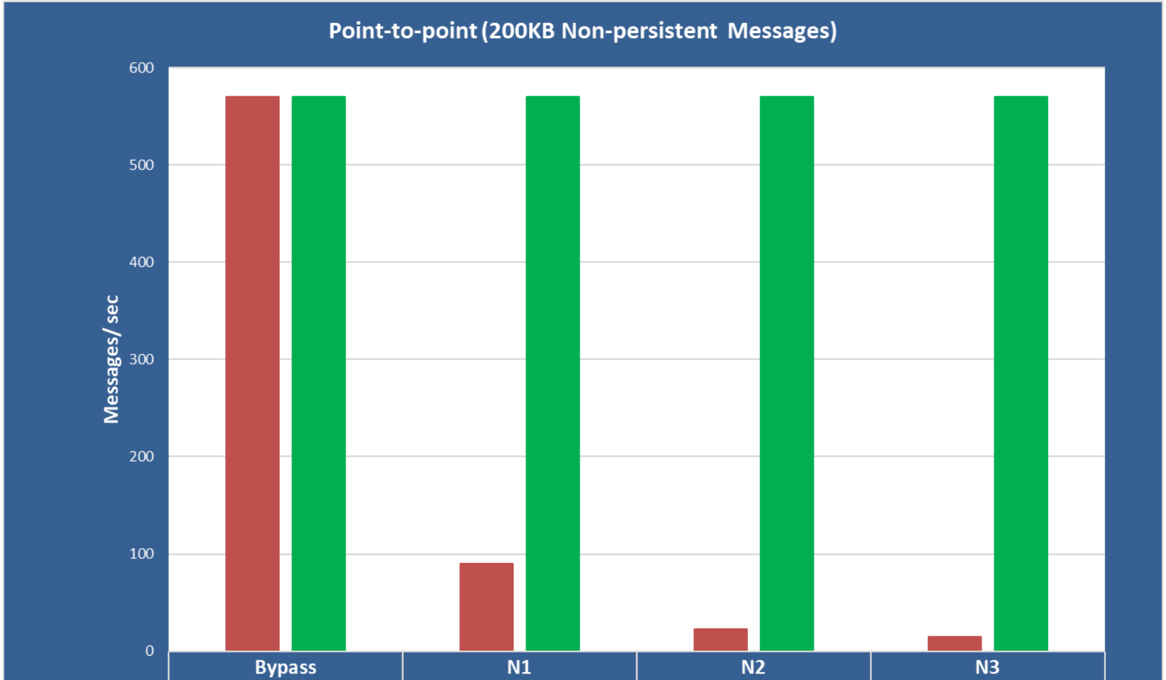
Summary

- For unconstrained network fasp.io Gateway showed near parity with TCP.
- As network constraints increased (N1 -> N3), the fasp.io Gateway showed increased benefit, reaching 3.9x TCP case for N3 network.

Reading The Graph

- Y-axis : Maximum messages/sec (before xmit queues start to back up).
- X-axis: Network constraint type (bypass = unconstrained).

200KB Non-persistent Point-to-point Workload

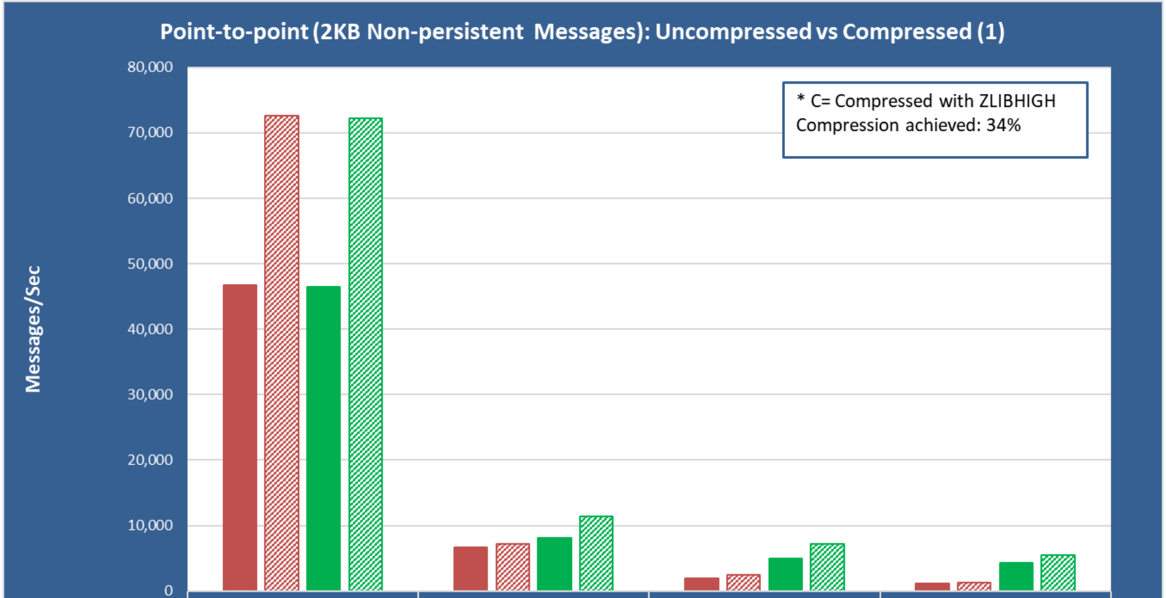


	Bypass	N1	N2	N3
TCP 200KB	570	90	23	15
FASP 200KB	570	570	570	570

Summary

- With larger message size, the fasp.io Gateway shows dramatic improvements over constrained TCP links.
- For unconstrained network the fasp.io Gateway showed near parity with TCP.
- As network constraints increased (N1 -> N3), the fasp.io Gateway showed increased benefit, reaching 30x TCP case for N3.

2KB Non-persistent Point-to-point Workload with Compression

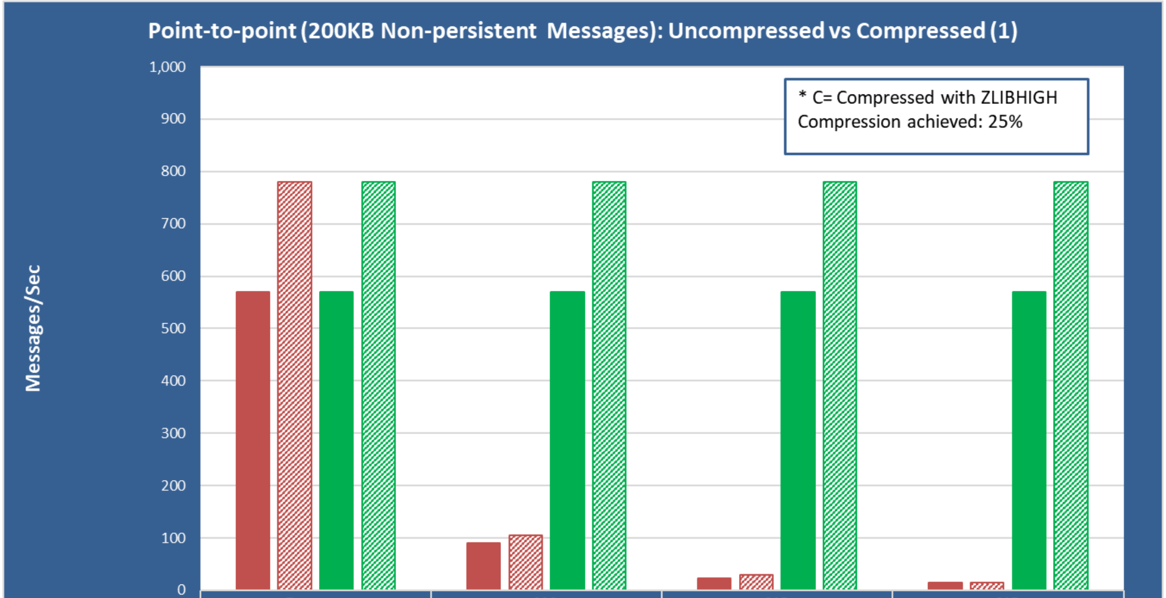


	Bypass	N1	N2	N3
TCP 2KB	46,750	6,600	1,870	1,100
TCP 2KB (C)*	72,600	7,150	2,420	1,320
FASP 2KB	46,530	8,140	4,950	4,290
FASP 2KB (C)*	72,270	11,440	7,150	5,500

Summary

- Messages compressed by 34%
- Compression and fasp.io Gateway improvements are cumulative
- .As network constraints increased (N1 -> N3), the fasp.io Gateway showed increased benefit, reaching 5x TCP (compressed) case for N3 network.
- CPU per message roughly doubles (but no application logic to ameliorate this). CPU still low on high latency links.

200KB Non-persistent Point-to-point Workload with Compression

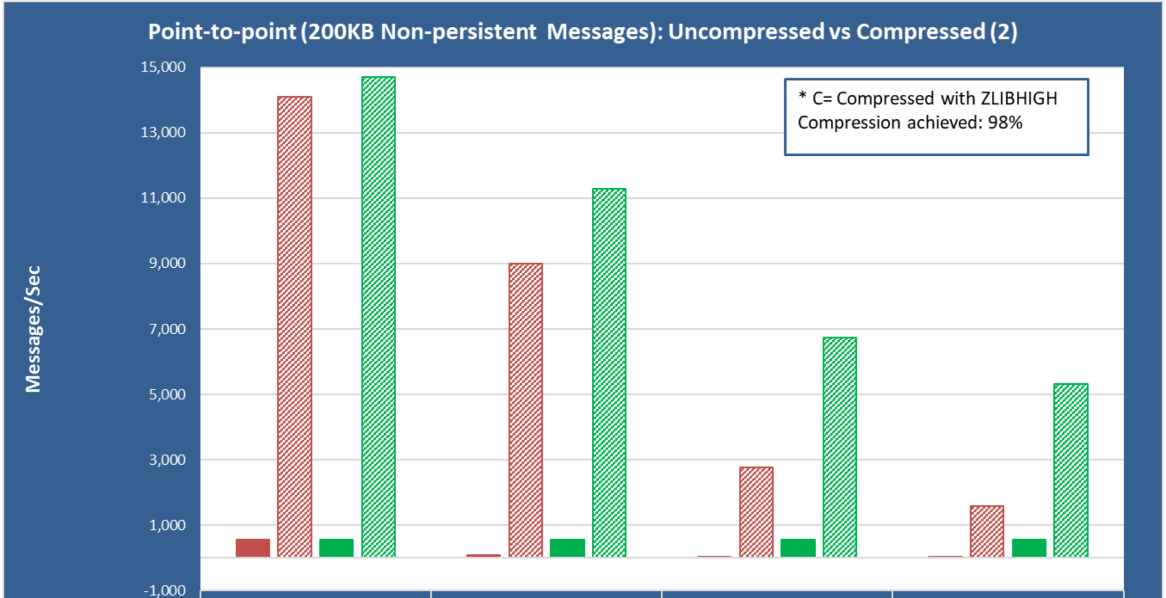


	Bypass	N1	N2	N3
TCP 200KB	570	90	22.5	15
TCP 200KB (C)*	780	105	30	15
FASP 200KB	570	570	570	570
FASP 200KB (C)*	780	780	780	780

Summary

- Messages compressed by 25%
- Compression and fasp.io Gateway improvements are cumulative
- .As network constraints increased (N1 -> N3), the fasp.io Gateway showed increased benefit, reaching 52x TCP (compressed) case for N3 network.
- CPU per message roughly doubles (but no application logic to ameliorate this). CPU still low on high latency links.

Compression/Gateway effect on Highly Compressible Messages

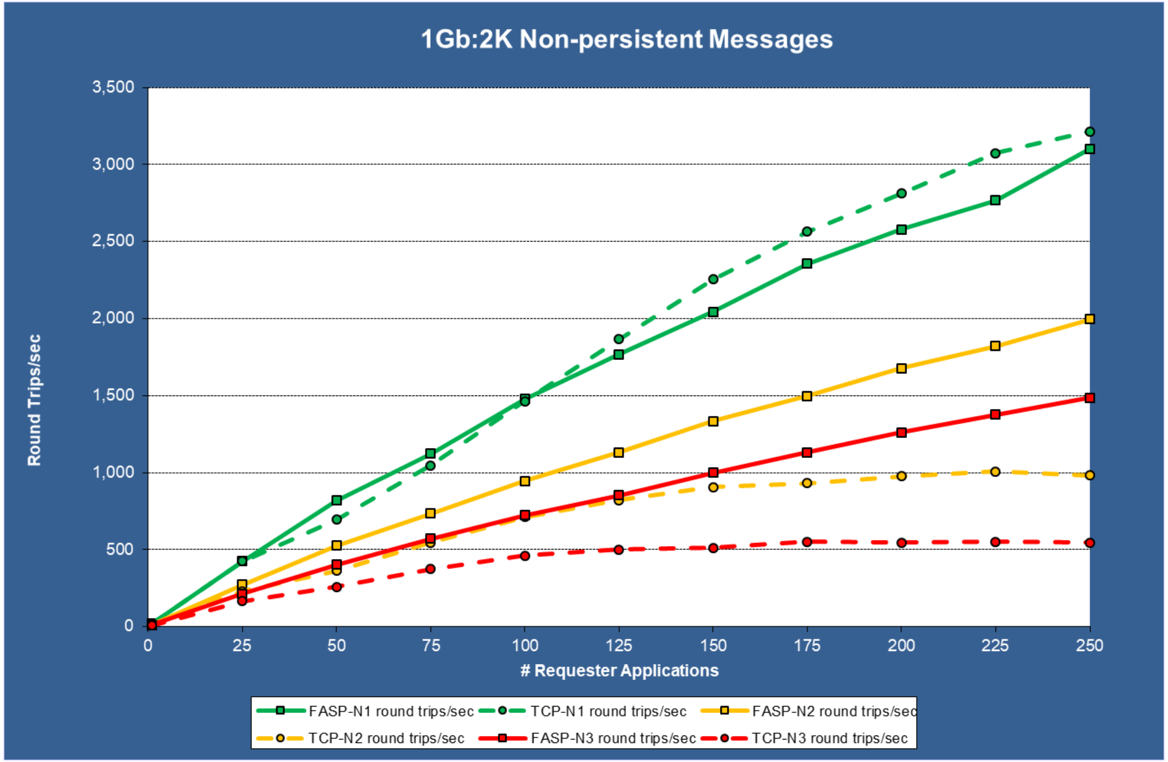


	Bypass	N1	N2	N3
TCP 200KB	570	90	22.5	15
TCP 200KB (C)*	14100	9000	2775	1575
FASP 200KB	570	570	570	570
FASP 200KB (C)*	14700	11295	6750	5325

Summary

- When messages are *very* compressible (98% in this case), then compression can give the greatest improvement.
- 200KB example here shows big impact of compression, but as network degrades, the fasp.io Gateway helps more.
- Smaller messages will benefit less from the Gateway (as with uncompressed scenarios), so testing on production-like data is highly desirable to simulate size and compressibility.

2KB Non-persistent Requester-responder Workload



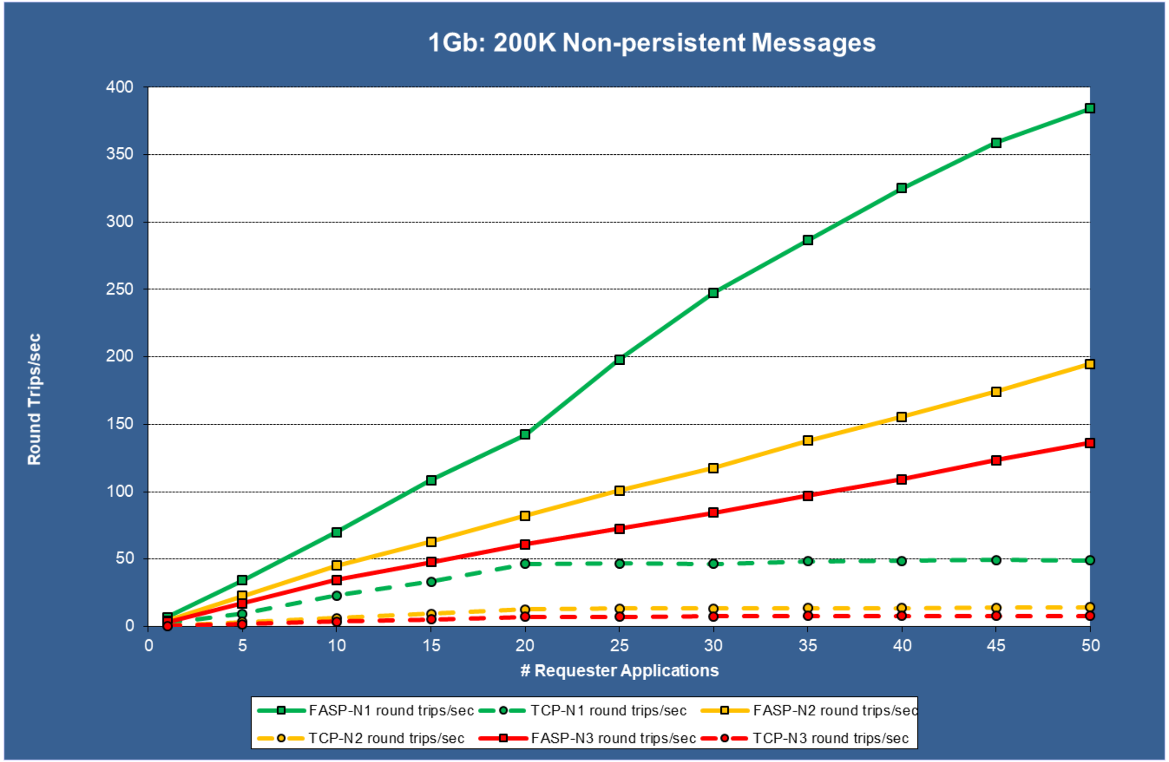
Summary

- For requester/responder workload (not optimal for fasp.io Gateway), higher latency networks (N2 & N3) showed benefits from using the fasp.io Gateway.
- N1 link (50ms round-trip delay) does not benefit.
- Not recommended for fast, unrestricted network.

Reading The Graph

- X-axis : Increasing numbers of requester applications
- Y-Axis Round trips/sec
- Solid lines are tests via fasp.io Gateway, dashed lines of same colour are direct TCP links for same test.

200KB Non-persistent Requester-Responder Workload



Summary

- Larger message size showed good benefits of using the fasp.io Gateway for all three restricted network cases.
- Not recommended for fast, unrestricted network with this type of workload, where fasp.io Gateway was typically slower (not shown here).

Resources

- MQ knowledge centre page
  https://www.ibm.com/support/knowledgecenter/SSFKSJ_9.1.0/com.ibm.mq.con.doc/aspera_def_unixlinux.htm

- MQ-CPH Workload:
  https://github.com/ibm-messaging/mq-cph

Questions?