ntt.com

NTT Communications
Transform. Transcend.

# 100G高速PCルーター開発
## ～国産ソフトウェアルーターへの挑戦～

2016年 11月 28日
NTTコミュニケーションズ株式会社
技術開発部　小原 泰弘

Transform your business, transcend expectations with our technologically advanced solutions.

# 概要

- コアルータ１つ１億円弱。
- PCルータはどこまでいけるのか。
- １つ置き換えることができれば、数千万のコストダウン。
- NFVに向けて。

- Poptrie (SIGCOMM'15)
- DPDK, RCU.
- 520K BGP full route で 128B frame で 145Gbps (~122Mpps) を達成。

- Kamuee0 実装、デモ、評価グラフ
- Poptrie 説明

# 100G高速PCルーター開発とは

- NTTコミュニケーションズでソフトウェアルーターを自作してしまおう！という研究プロジェクト

- PCとは：
  - インテルx86アーキテクチャを踏襲したサーバー
- PCルーターとは：
  - PCサーバーでソフトウェアルーターを動作させたもの
- 100G高速とは：
  - 市販の100Gイーサーネットサーバーアダプターをターゲット
  - ショートパケット**でも**ワイヤーレート性能を達成する
  - 経路ルックアップ有り**でも**ワイヤーレート性能を達成する

- 安価な市販部品で高速な専用機械を自作する
- 専用ハードウェアを用いないため安価
- メインはソフトウェア技術

# 背景

- コアバックボーンルーター（専用機）は高価（数千万円）
  - コアバックボーンの経路は最大約65万（2016/10時点）
  - 64B パケットで 100GbE における最大性能は 148.8 Mpps：つまり 1億4880万パケット毎秒
- 経路ルックアップ専用の部品：TCAM
  - 熱とスケール（集積度）の問題
- CPUの高速化、高並列化（3GHz、10コア等）
- ソフトウェアの進化
  - パケット転送：DPDK, NetMap
- NFVの出現
  - 仮想化環境におけるハイパーバイザー（PC）上でのスイッチ、ルーター
- ソフトウェアルーターの重要性

# NTTコミュニケーションズの強み

- TCAMを代替できるアルゴリズム（ソフトウェア）
- 経路ルックアップアルゴリズム Poptrie の開発と提案
  - 東京大学　浅井大史　助教による発明、共同研究
  - ACM SIGCOMM '15 での論文採録
    - ✓ H. Asai, Y. Ohara, "Poptrie: A Compressed Trie with Population Count for Fast and Scalable Software IP Routing Table Lookup", ACM SIGCOMM '15, p.57-70.
- 特許取得
  - 特許第5960863号（特願2015-048657）H.28 7/1 登録
- 実装： Kamuee0
  - DPDKを利用したソフトウェアルーター実装
  - プロプラエタリ（NTTコム固有の技術）
- 40GbE x 4、51万経路 で 128Bytes 145Gbps 程度を達成

# 現状での課題

- 性能問題
  - スループット性能の問題
    - ✓ 128Bytes 未満のスループット性能が低い
  - パケットロス問題
    - ✓ 128Bytes 以上でも（低負荷時でも）パケットロスが多い
  - FPGAサーバーアダプターの開発？
- 機能問題
  - ルーティングプロトコルエンジンの統合
  - 統計カウンター実装
  - NFVサポート（SR-IOV等）
- CAPEX（初期投資）低下、でもOPEXは？

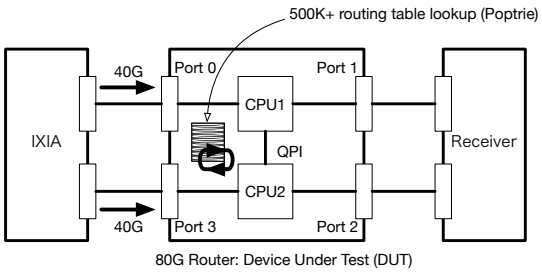
# 顧客ユーザーさまに関連する展望

- NTTコミュニケーションズのバックボーンサービスの品質向上
  - 100GbE x 4 レベルの物理ルーターの低コスト化
  - 後方支援：余裕が出ればリソースを他に投入できる（玉突き効果）
- 顧客さまSI案件での高速で安価な物理ルーターの導入
- 共同の研究開発（FW、CGN、ロードバランサー、DPI、…）

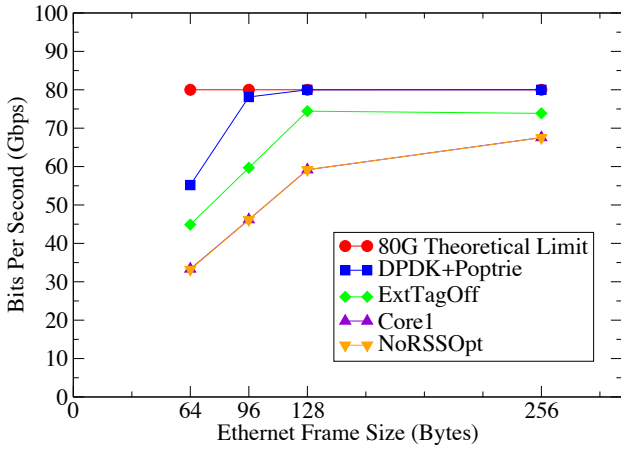
- クラウドやNFVサービスでの、安価で高速な仮想ルーターの選択肢

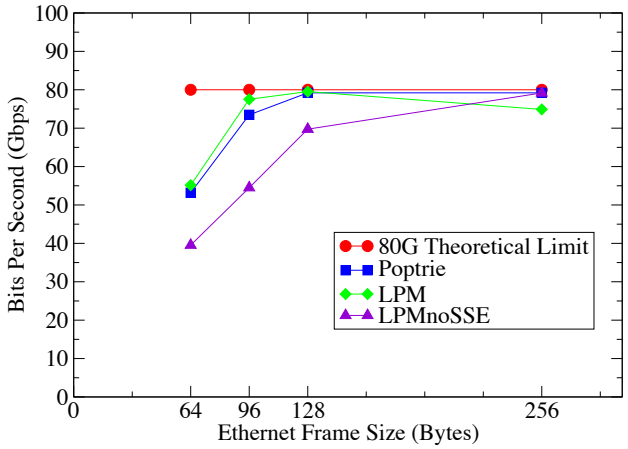
# 80Gbps (unidirectional) achieved with DPDK

- a paper appeared in AINTEC 2015
  - Y. Ohara et. al. "Revealing the Necessary Conditions to Achieve an 80Gbps High-Speed PC Router" AINTEC 2015
- Comparisons of different configurations
- Comparisons of Poptrie v.s. LPM (DPDK's DIR-24-8) with/ without SSE optimizations



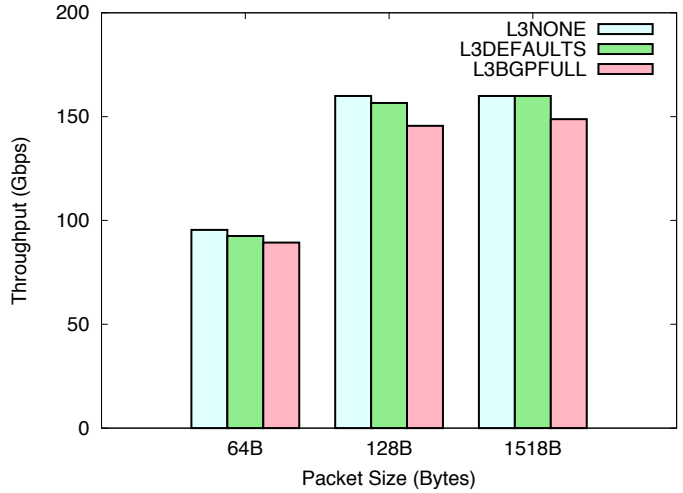
80G Router: Device Under Test (DUT)



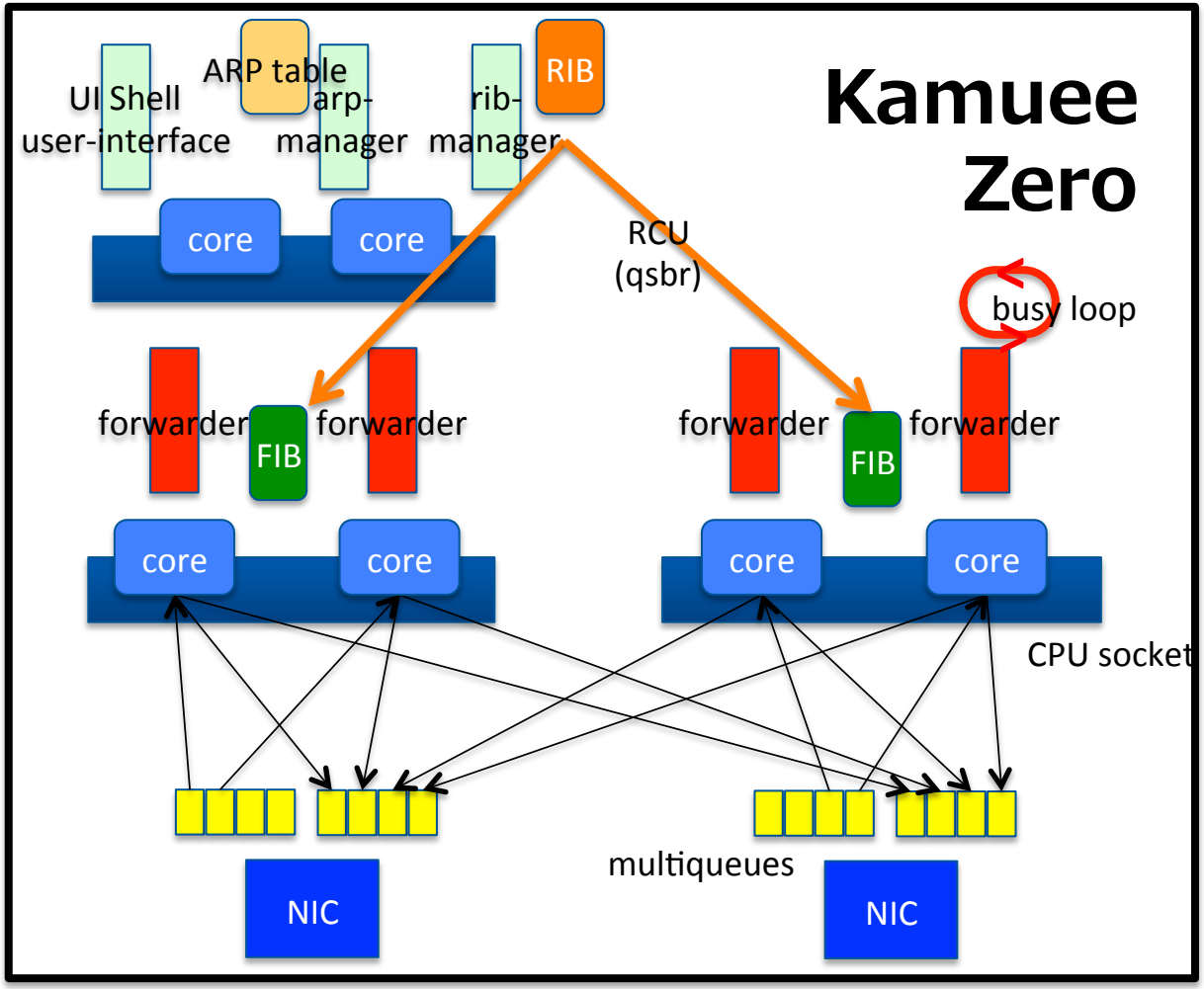



# Performance of Kamuee Zero

- a paper appeared in IC 2016
  - Y. Ohara, Y. Yamagishi, "Kamuee Zero: the Design and Implementation of Route Table for High-Performance Software Router" IC 2016
- Achieved 145Gbps on 40GbE I/F x4, BGP Full Route, 128B



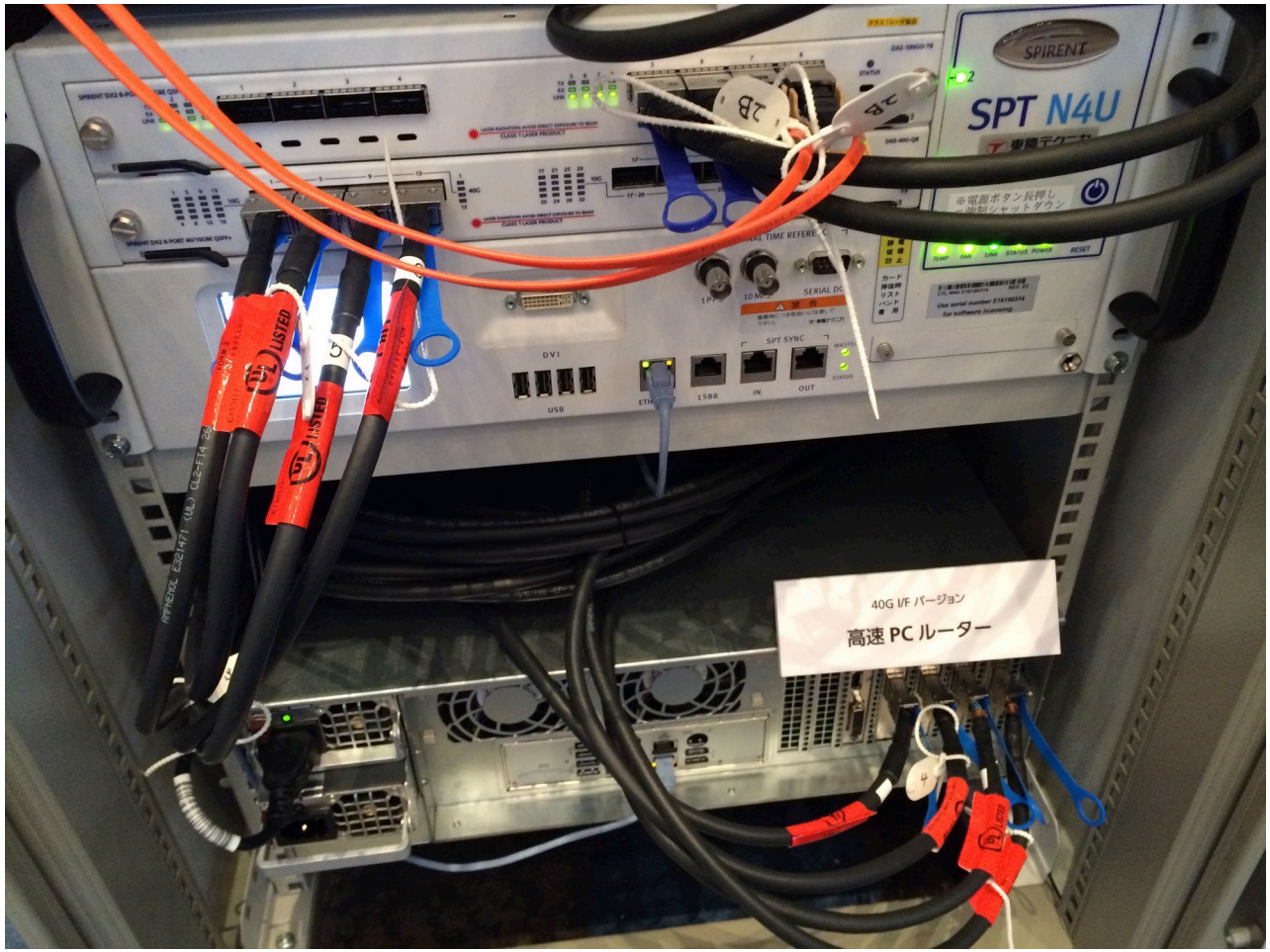
# Kamuee0's internal structure



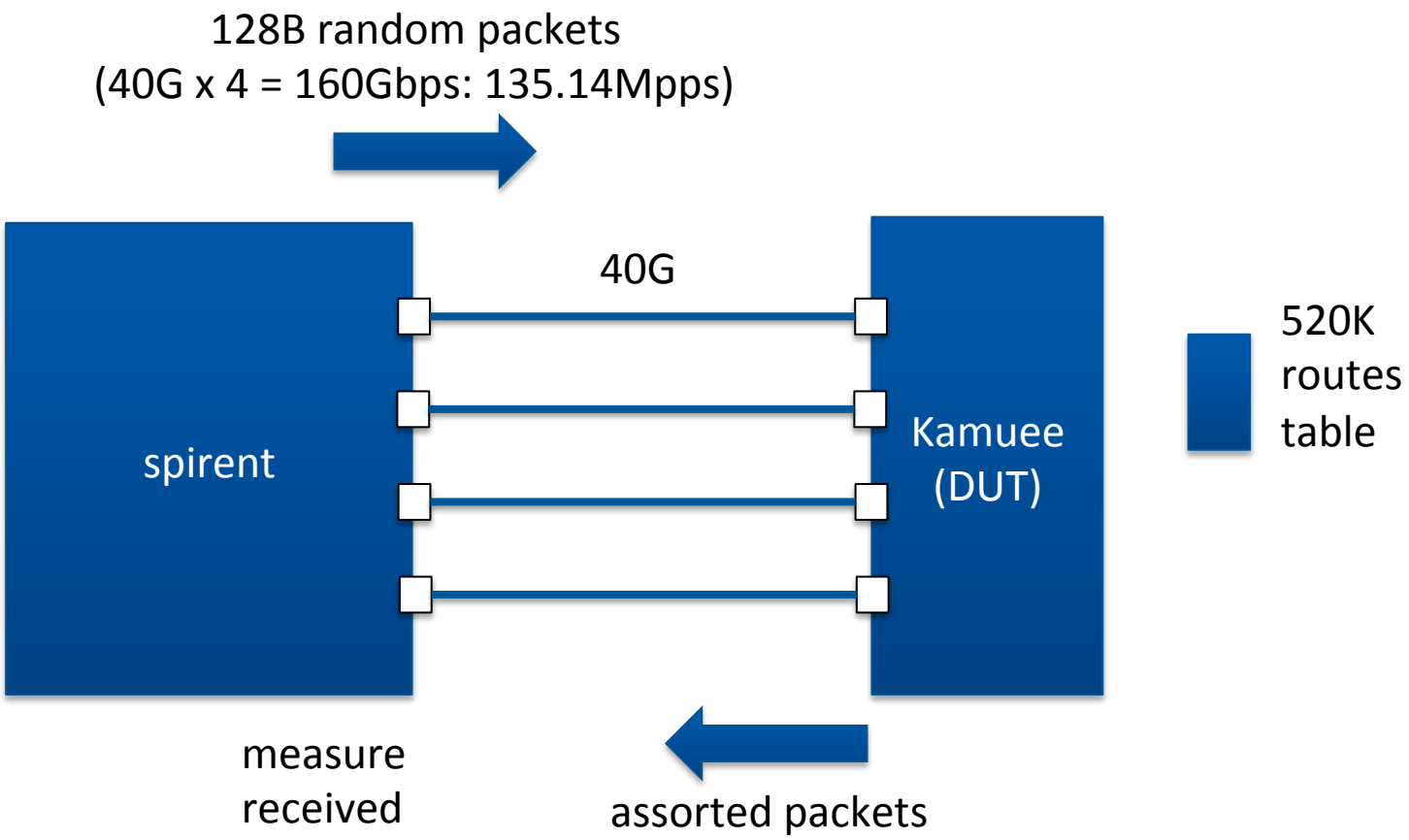
# At ComForum 2016 (10/6,7, Tokyo)

# Demo

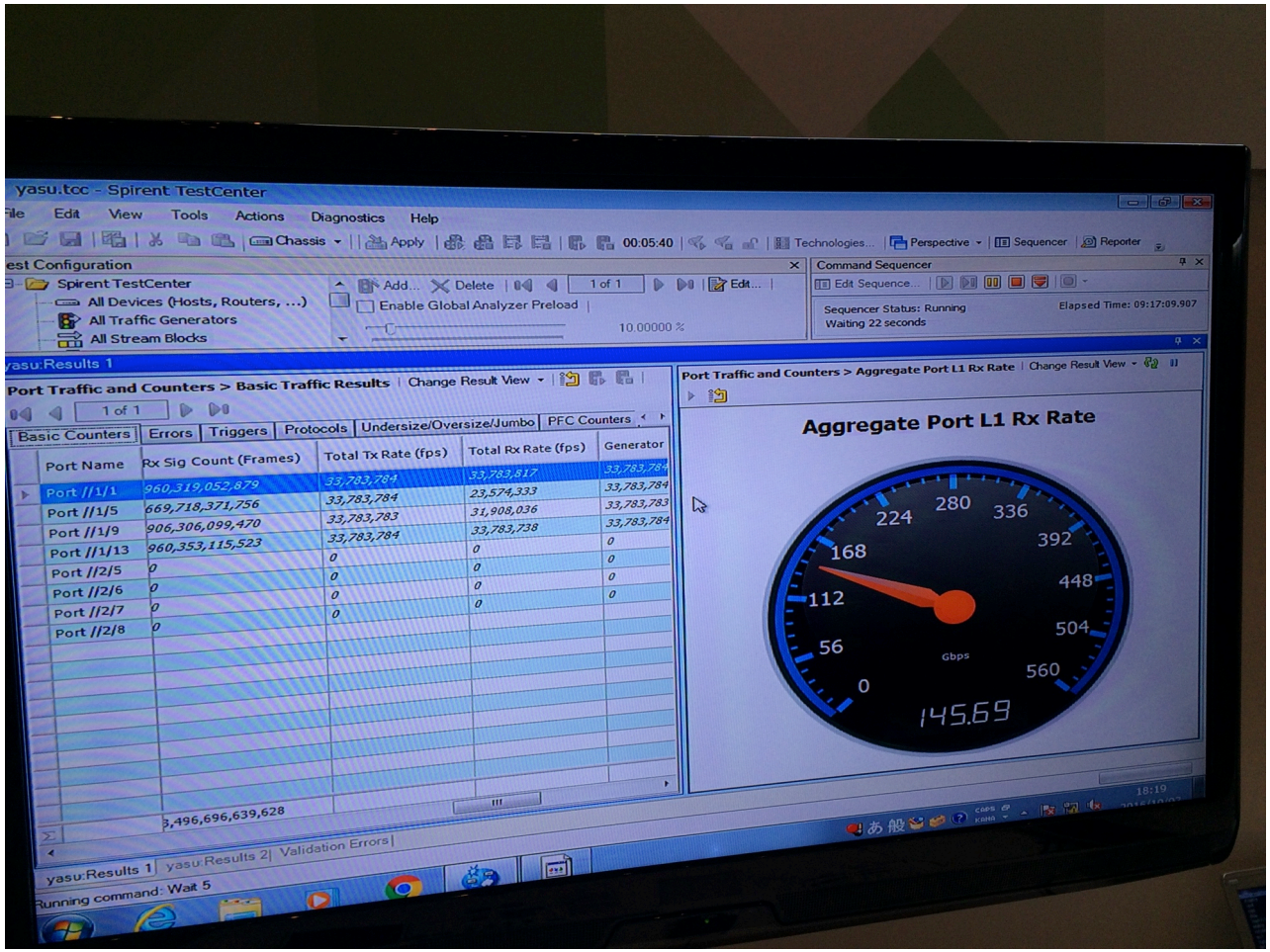
# 40G x4 version



# 接続設定

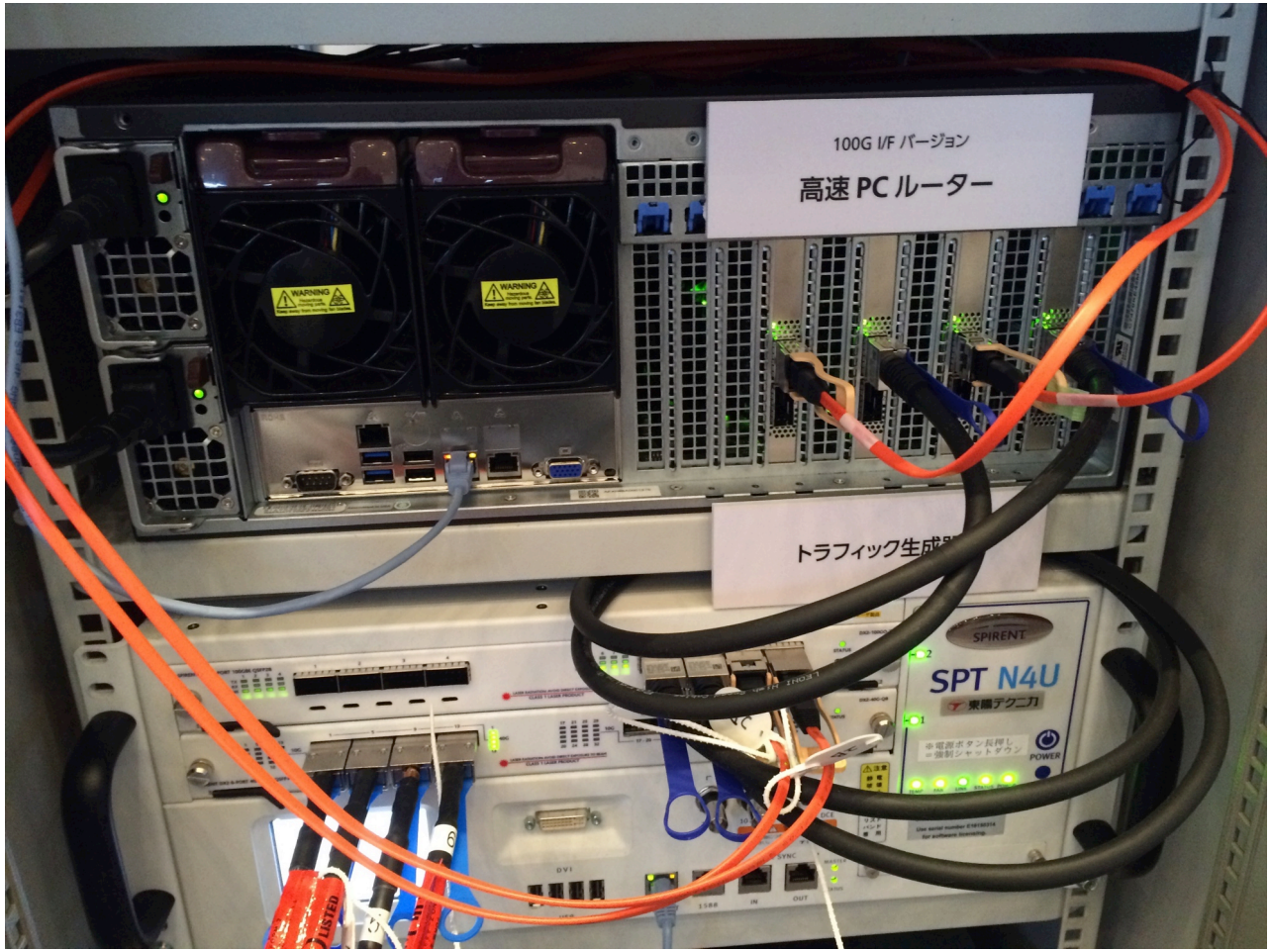


# Benchmark (145Gbps on 128B 40Gx4 510K routes.)

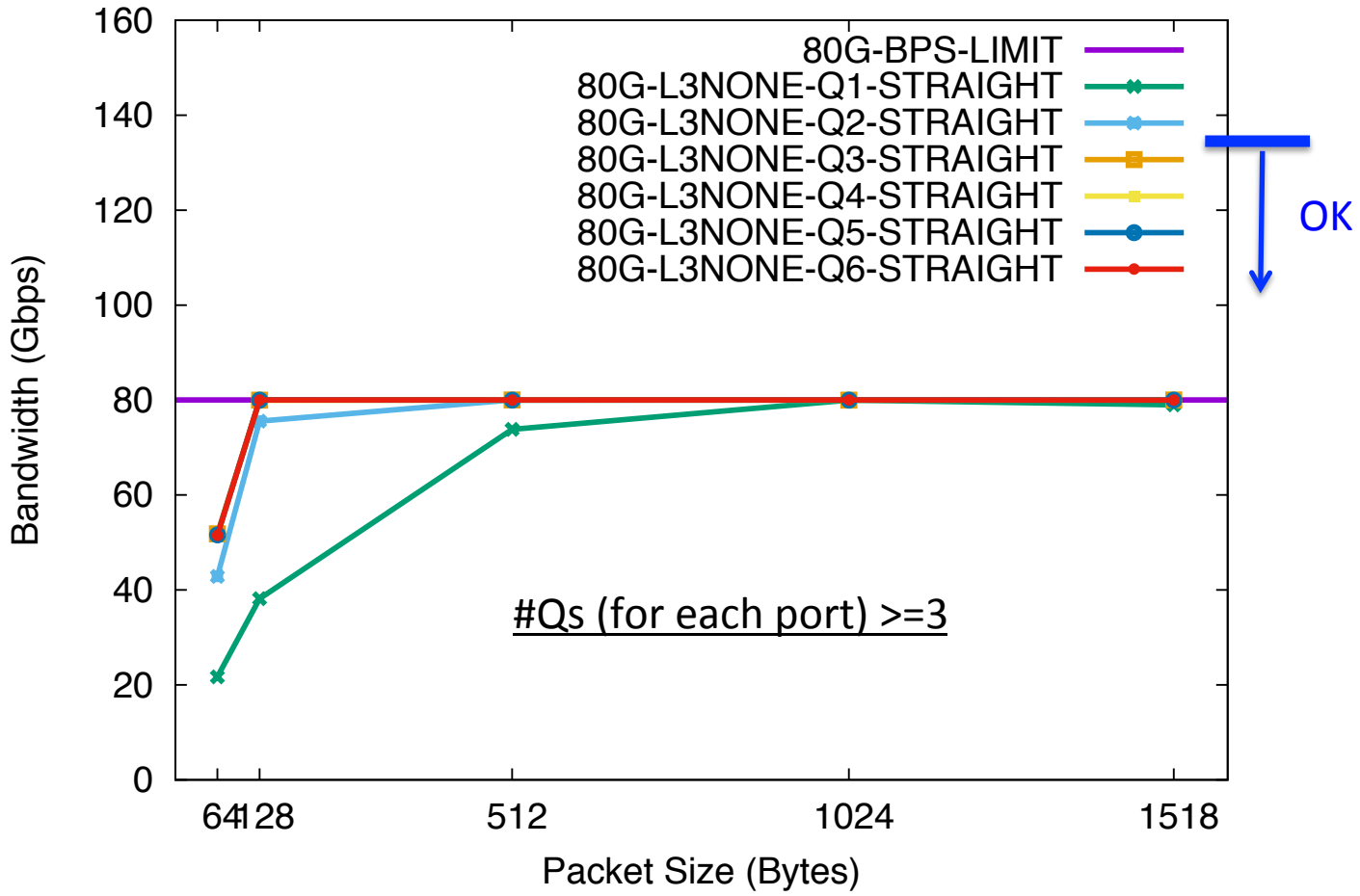


# CLI

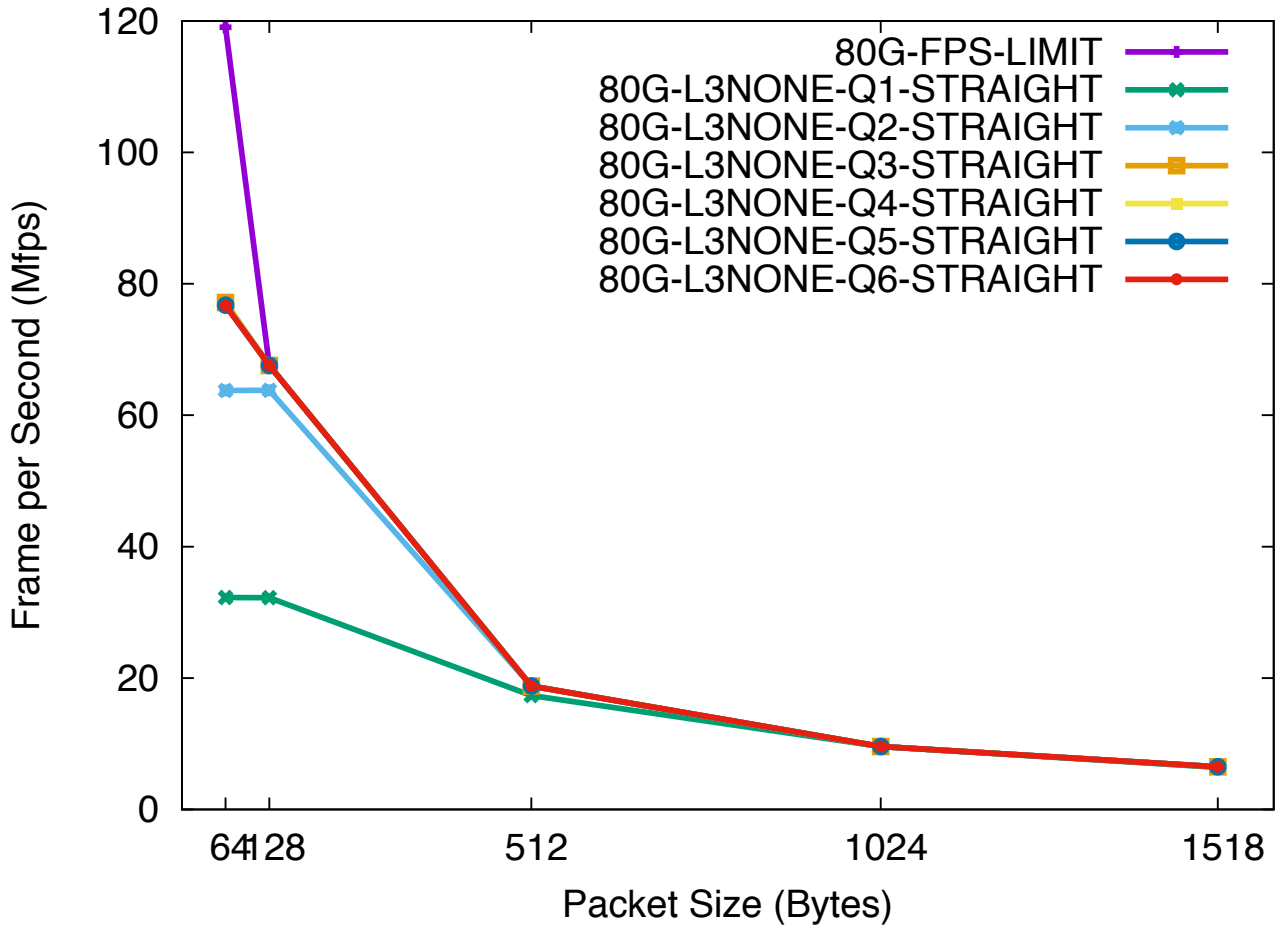
# 100G x4 version



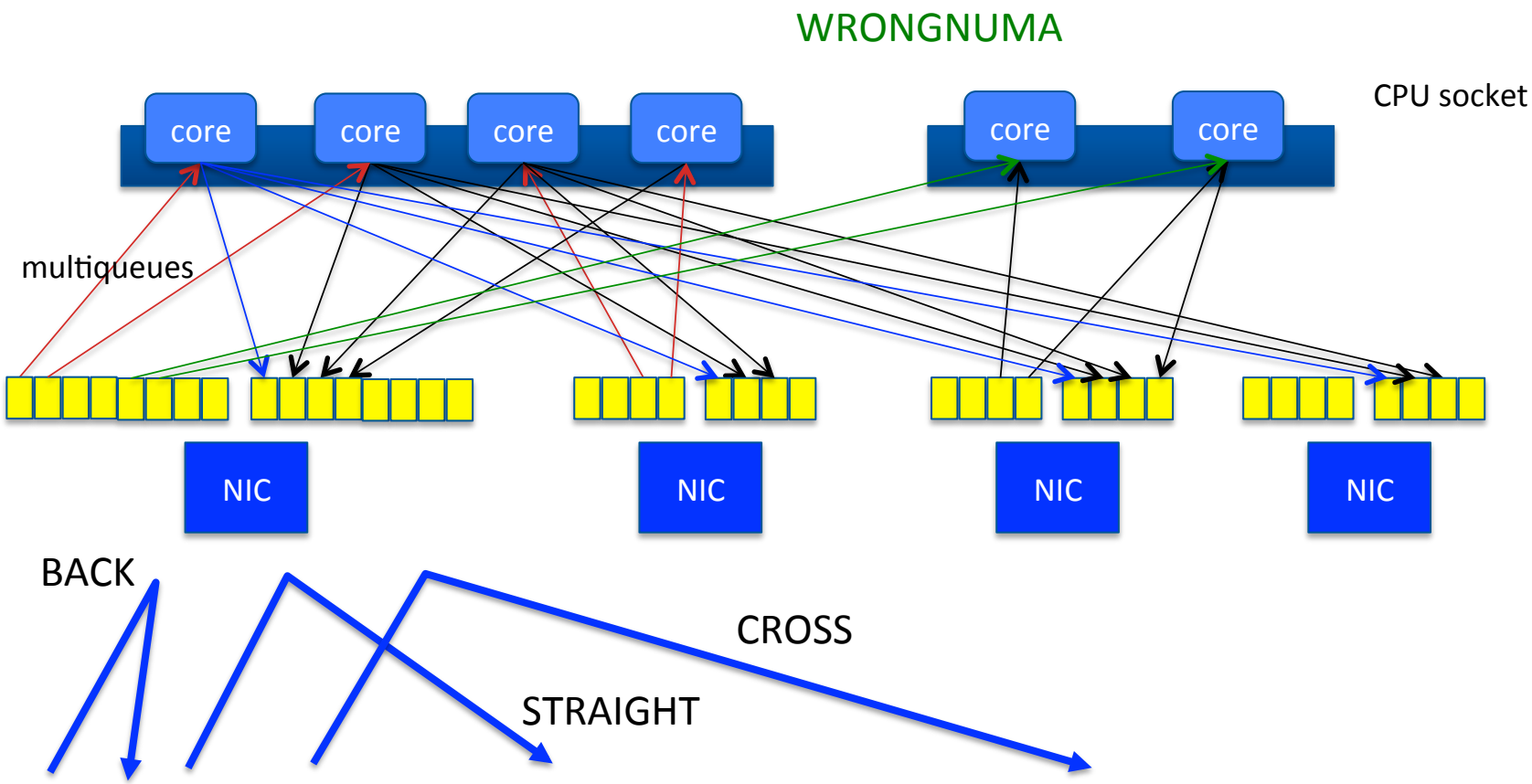
# Q1 – Q6 (bps)



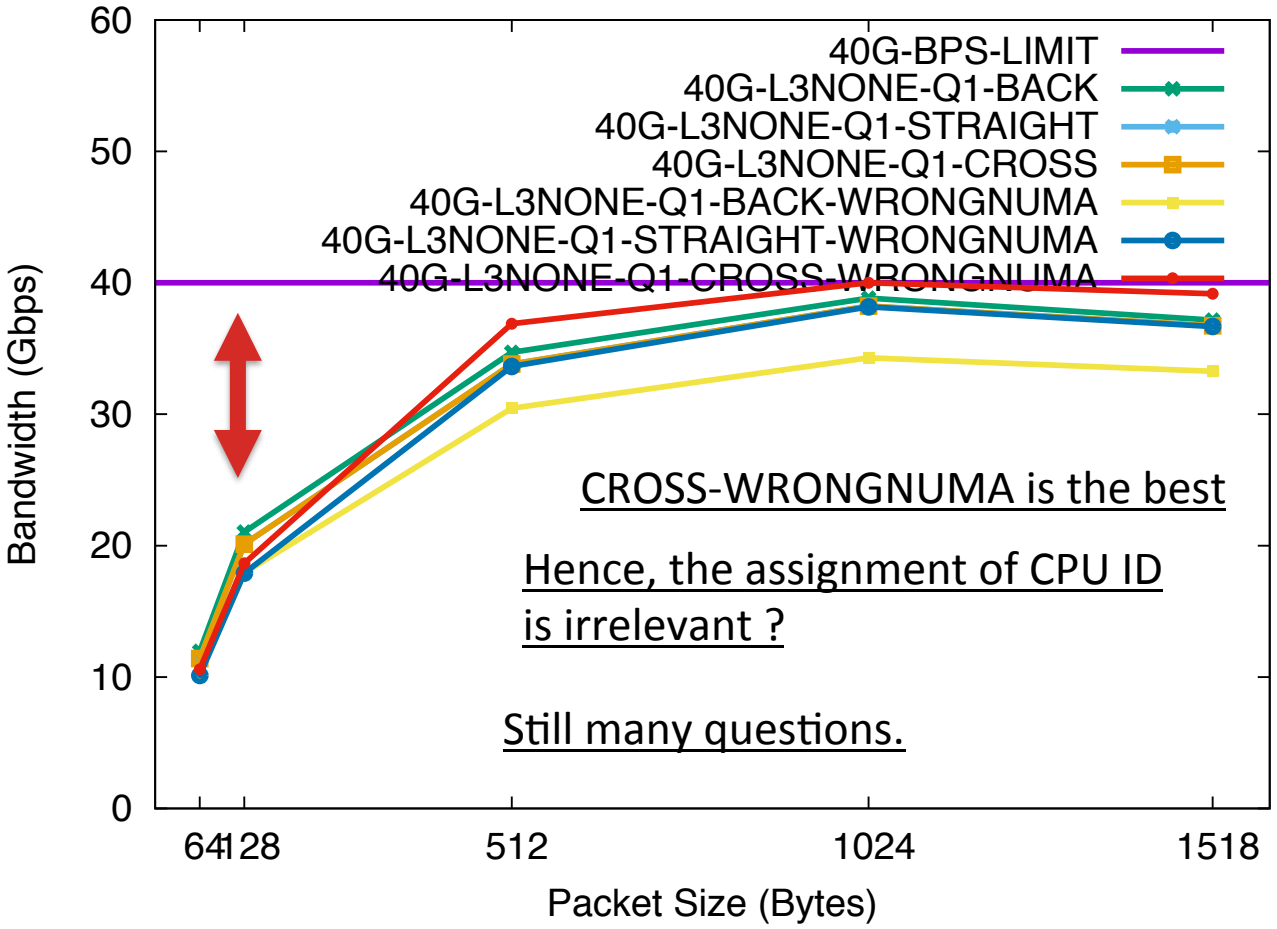
# Q1 – Q6 (fps) (80g -> 2 ports)



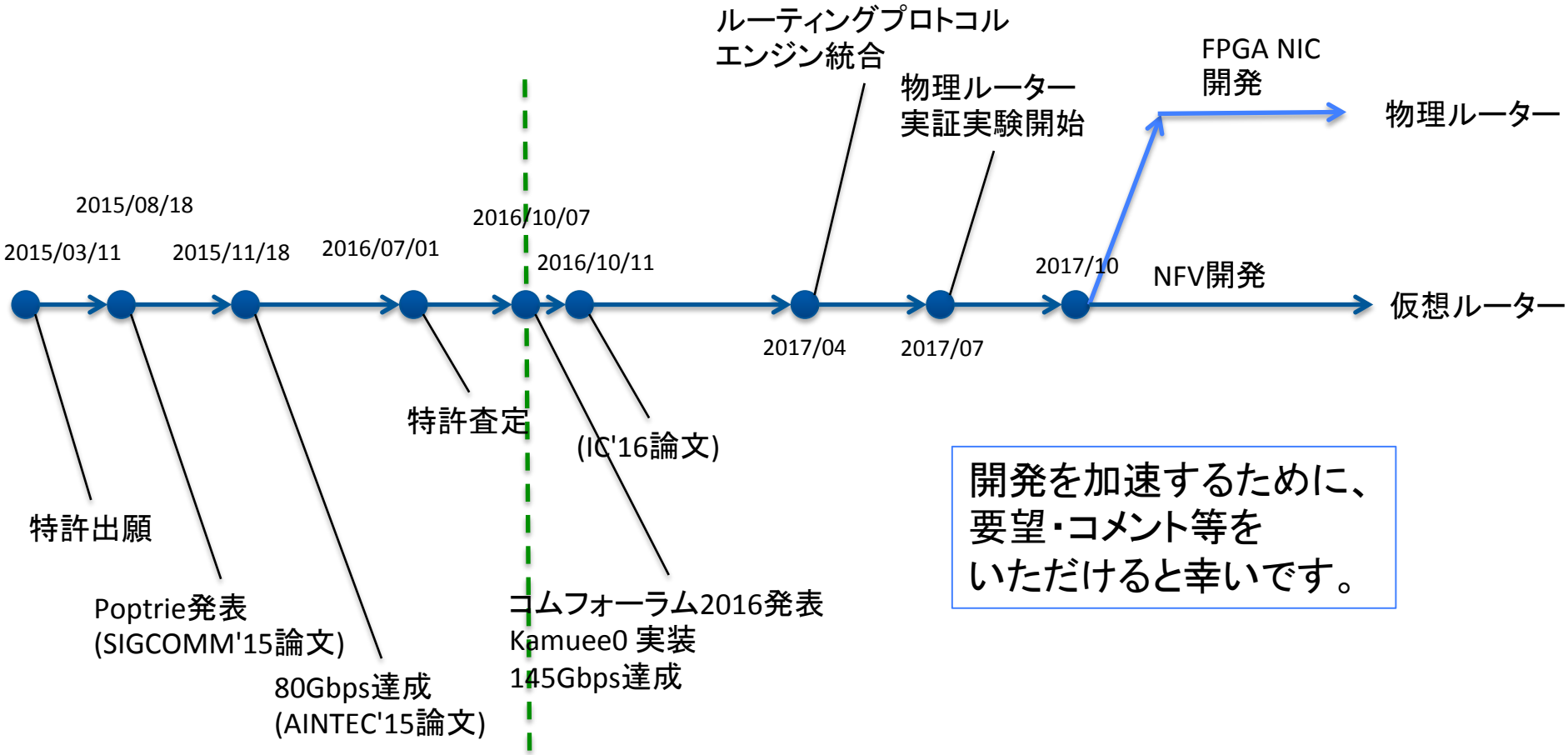
# Back / Straight / Cross, and WrongNuma



# Single Queue, traffic paths (bps)



# 開発の歩みと期待




開発を加速するために、
要望・コメント等を
いただけると幸いです。

# これから

- Tap デバイスを使って Quagga と連結する
- Netlink に対応
- 実験網で試用
- 商用網で運用

- NFV(VRF, SR-IOV, SPP等)に対応
- NTTグループ内SDN案件に貢献（したい）

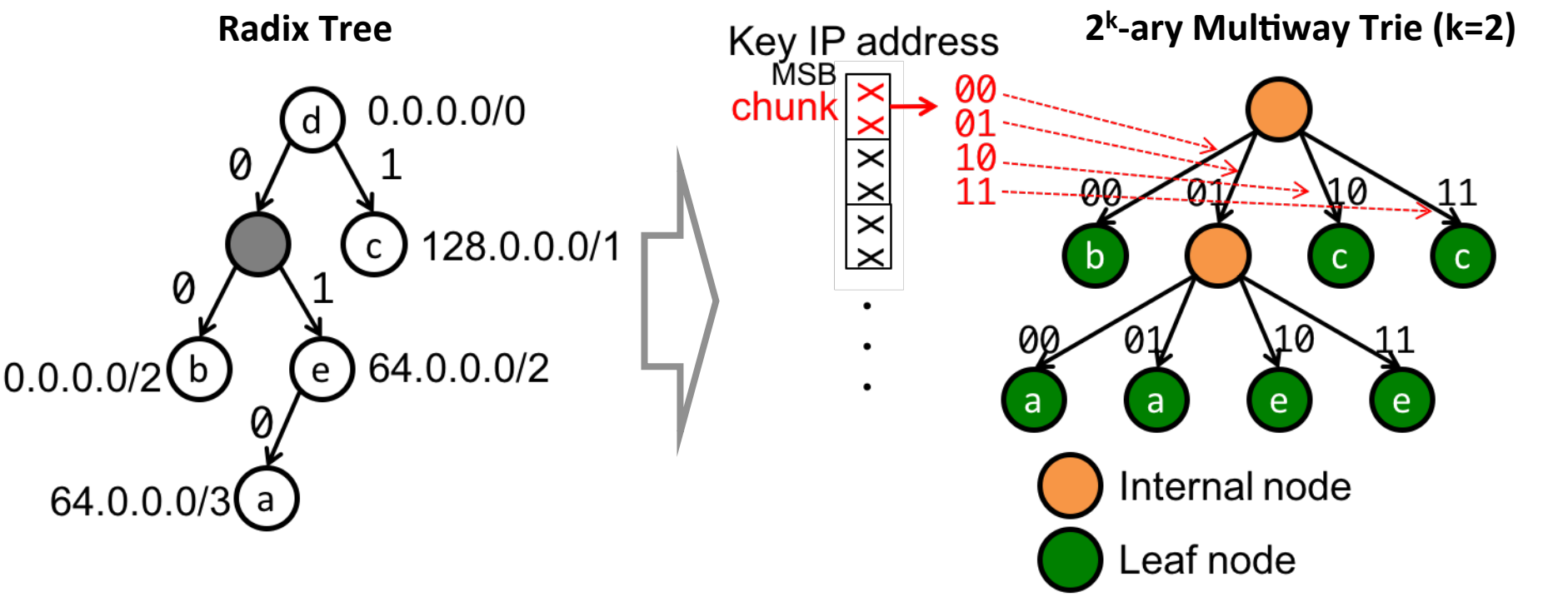
# Poptrie 概要 (1)

- ロンゲストマッチを実行するIP経路ルックアップのためのデータ構造・アルゴリズム
  - ビットマップと popcnt 命令を使った多進木(64分木)トライ (trie)
- メモリ使用量がとても少ない
  - ティア1 ISPのバックボーンルータの経路（BGPフルルート+IGP経路, 53万経路）が2.4MBに圧縮できる
  - CPUキャッシュに載る
- 超高速なルックアップ速度
  - CPU1コアで200Mlps （ルックアップ/秒）（１００GbE 最小パケットで 148.8 Mlps必要）
  - コア数でほぼリニアにスケール： 4コアで 900Mlps 達成
- 経路アップデート（追加・削除）は複雑だが高速
  - 全経路表の再構築で 40ms 程度
  - 差分の追加・削除は１経路あたり平均で 2～5 μs

# Poptrie 概要 (2)

- ACM SIGCOMM 2015 に採録
  - http://dl.acm.org/citation.cfm?id=2787474
  - 約40種類の経路表でテスト済み
  - SAIL, DXR, TreeBitMap と比較
- 強力、強固なアルゴリズム
  - ほぼすべてのケースで他のアルゴリズムを凌ぐ
  - 悪い状況のときほど優位（プリフィックス数、長いプリフィックスの混在）
- 発明者
  - 東京大学　浅井 大史　助教
- 特許出願済み
  - 特願2015-048657

# Poptrie (basic): $2^k$-ary Multiway Trie

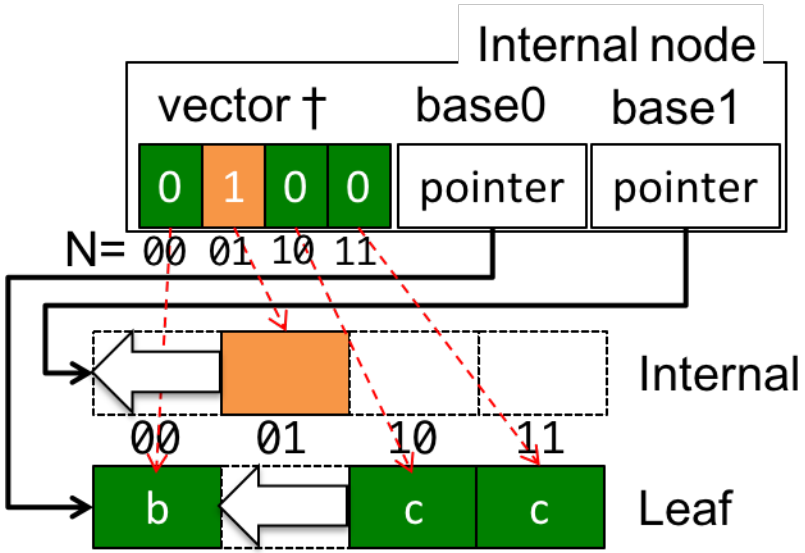



Push the next hop information in the internal nodes to leaf nodes

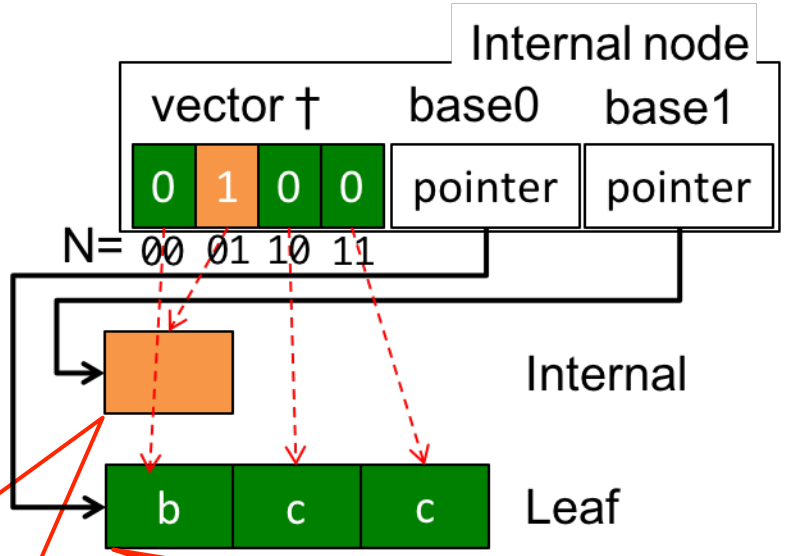
# Poptrie (basic): Pointer Compression with Population Count

† in little endian

**Pointer compression with bit-vector and array**




**Array compression with population count**




Index: # of 1's bits in the least significant N+1 bits of vector (-1)

Index: # of 0's bits in the least significant N+1 bits of vector (-1)

N: Value of the chunk

Counting # of 1s: Use x86's *popcnt* instruction

# Comparison with Other Algorithms (for random traffic pattern)

| Algorithm | Memory [MiB] | Rate [Mlps] |
|---|---|---|
| Radix | 30.48 | 8.82 |
| Tree BitMap | 2.62 | 56.24 |
| Tree BitMap (64-ary) | 3.10 | 61.61 |
| SAIL | 44.24 | 158.22 |
| D16R | 1.16 | 116.63 |
| D18R | 1.91 | 179.92 |
| $\text{Poptrie}_{16}$ | 2.75 | 198.28 |
| $\text{Poptrie}_{18}$ | 2.40 | 240.52 |

Routing table: Backbone core router of tier-1 ISP (Jan. 9, 2015), Traffic pattern: Random

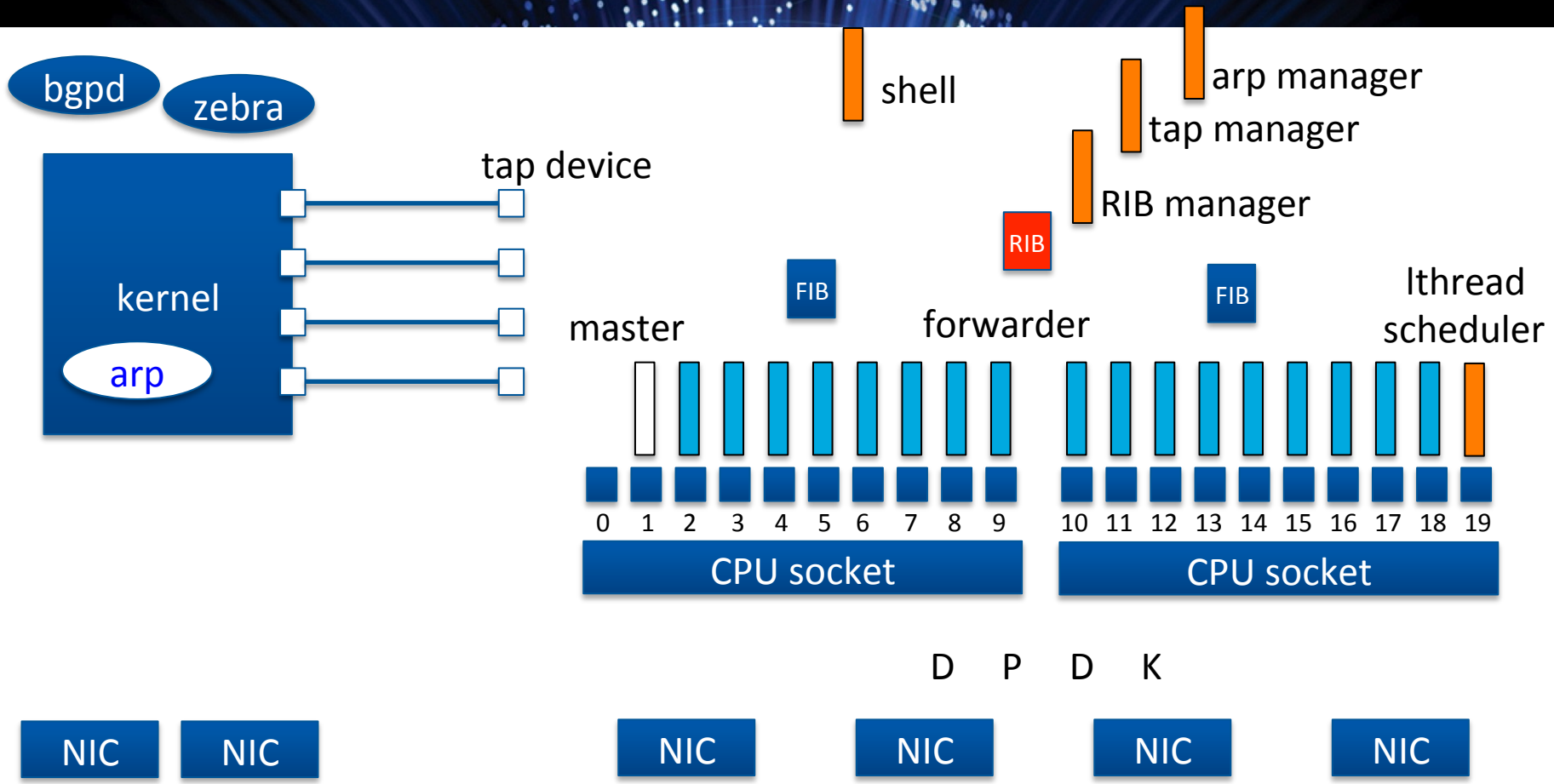
$\text{Poptrie}_{18}$ runs 1.34–27.3x faster than the other algorithms

# Papers


- H. Asai, Y. Ohara, "Poptrie: A Compressed Trie with Population Count for Fast and Scalable Software IP Routing Table Lookup", ACM SIGCOMM '15, p. 57-70.
- Y. Ohara, Y. Yamagishi, S. Sakai, A. DattaBanik, S. Miyakawa, "Revealing the Necessary Conditions to Achieve an 80Gbps High-Speed PC Router", AINTEC '15, p. 25-31.
- Y. Ohara, Y. Yamagishi, "Kamuee Zero: the Design and Implementation of Route Table for High-Performance Software Router", IC 2016.

# パケロス

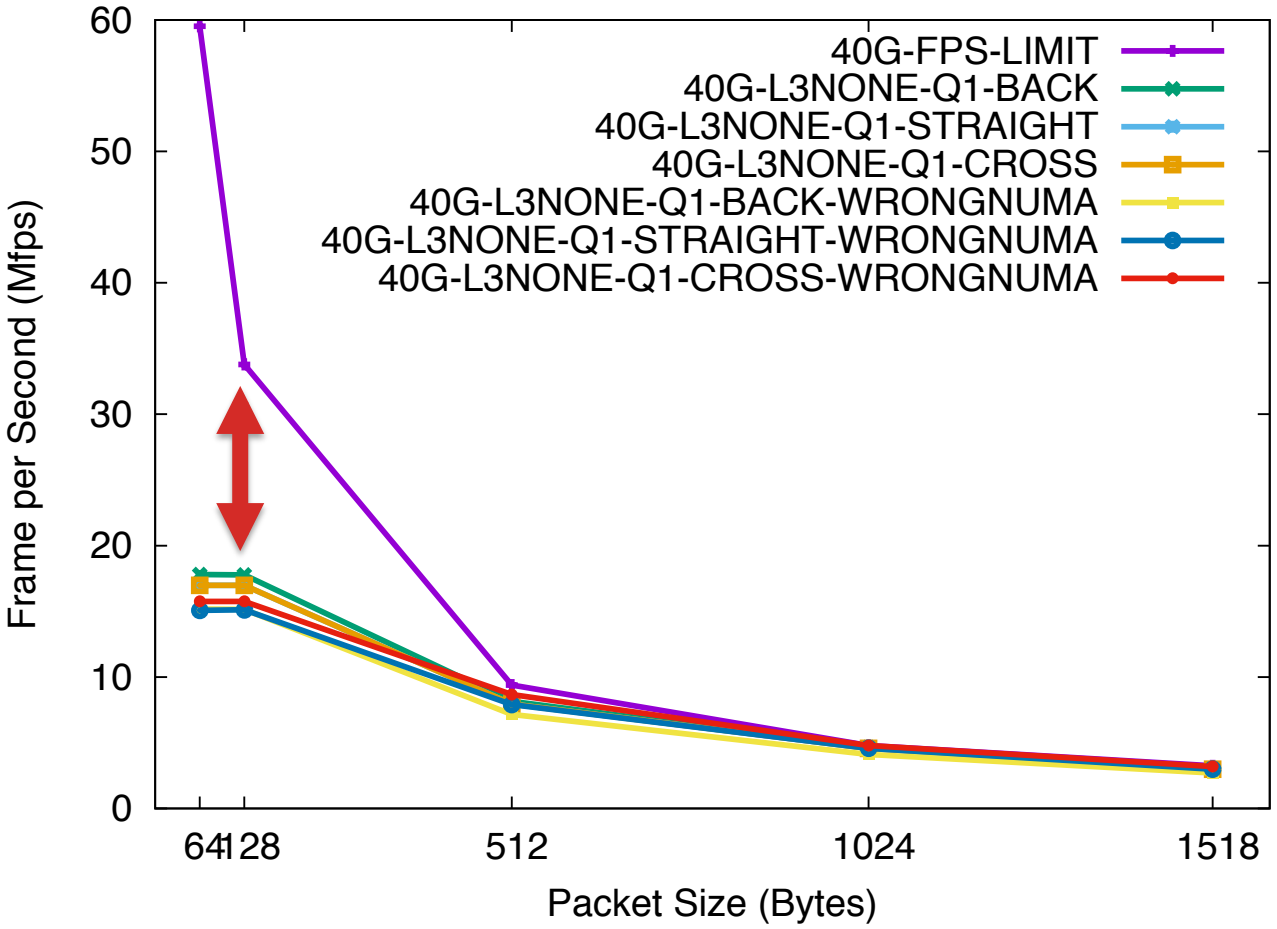
- 共通：20G/40G(50%) 518,231経路 BGP停止 600sec(10分)
- 1 I/F入力
  - port1: 101.53億パケット中 31 ロス（3.05E-09/0.0000000031）
  - port2: 101.54億パケット中 21 ロス（2.07E-09/0.0000000021）
  - port3: 101.47億パケット中 300,879（2.97E-05/0.0000296530）
  - port4: 101.48億パケット中 13 ロス（1.28E-09/0.0000000013）
- 2 I/F入力
  - port1&2: 203.10億P中 1225 ロス (6.03E-08/0.0000000603)
  - port1&3: 203.04億P中 299,312 (1.47E-05/0.0000147419)
  - port1&4: 203.03億P中 22 ロス (1.08E-09/0.0000000011)
  - port2&4: 203.06億P中 28 ロス (1.38E-09/0.0000000014)



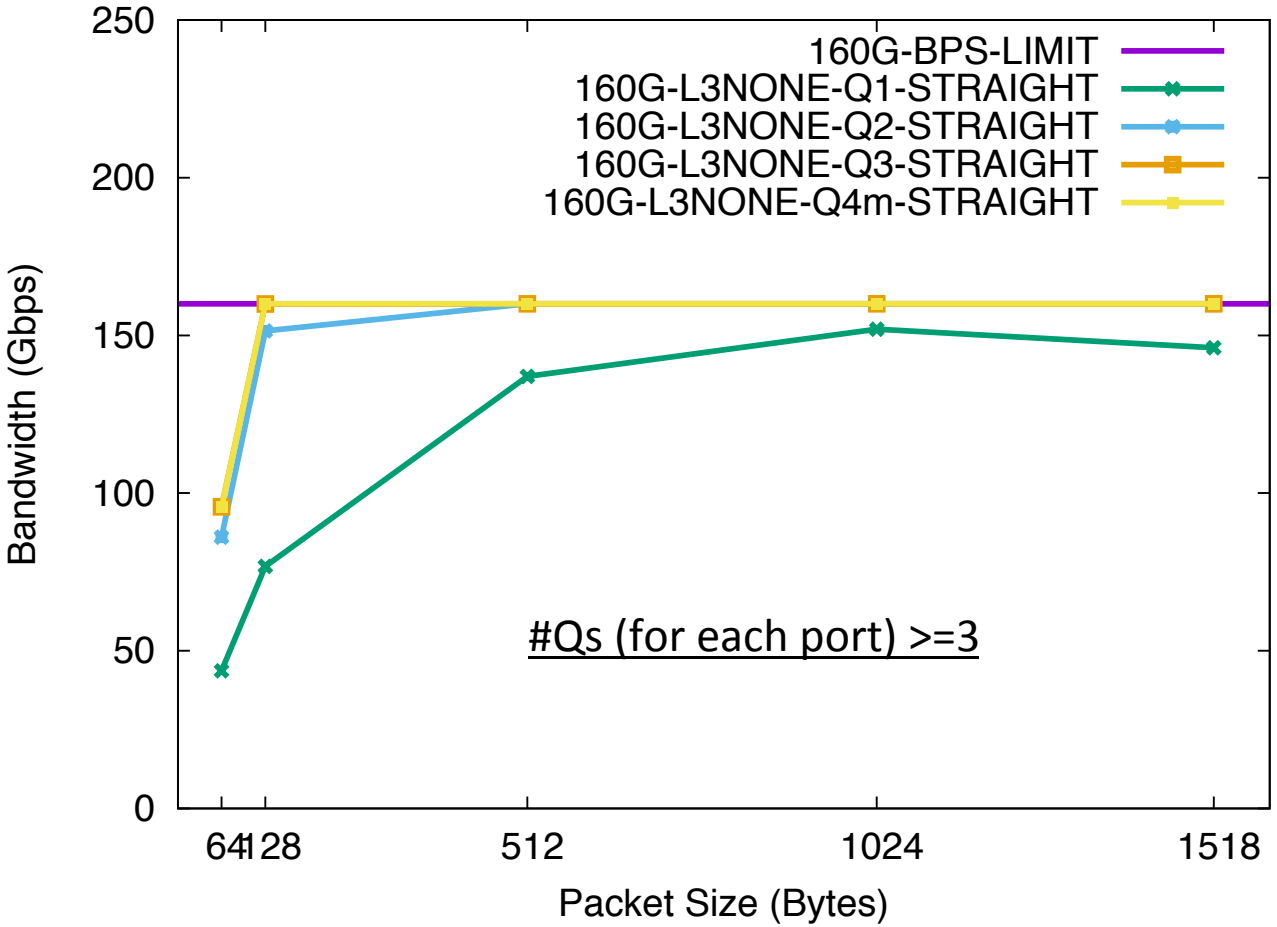
# デモ

終わり

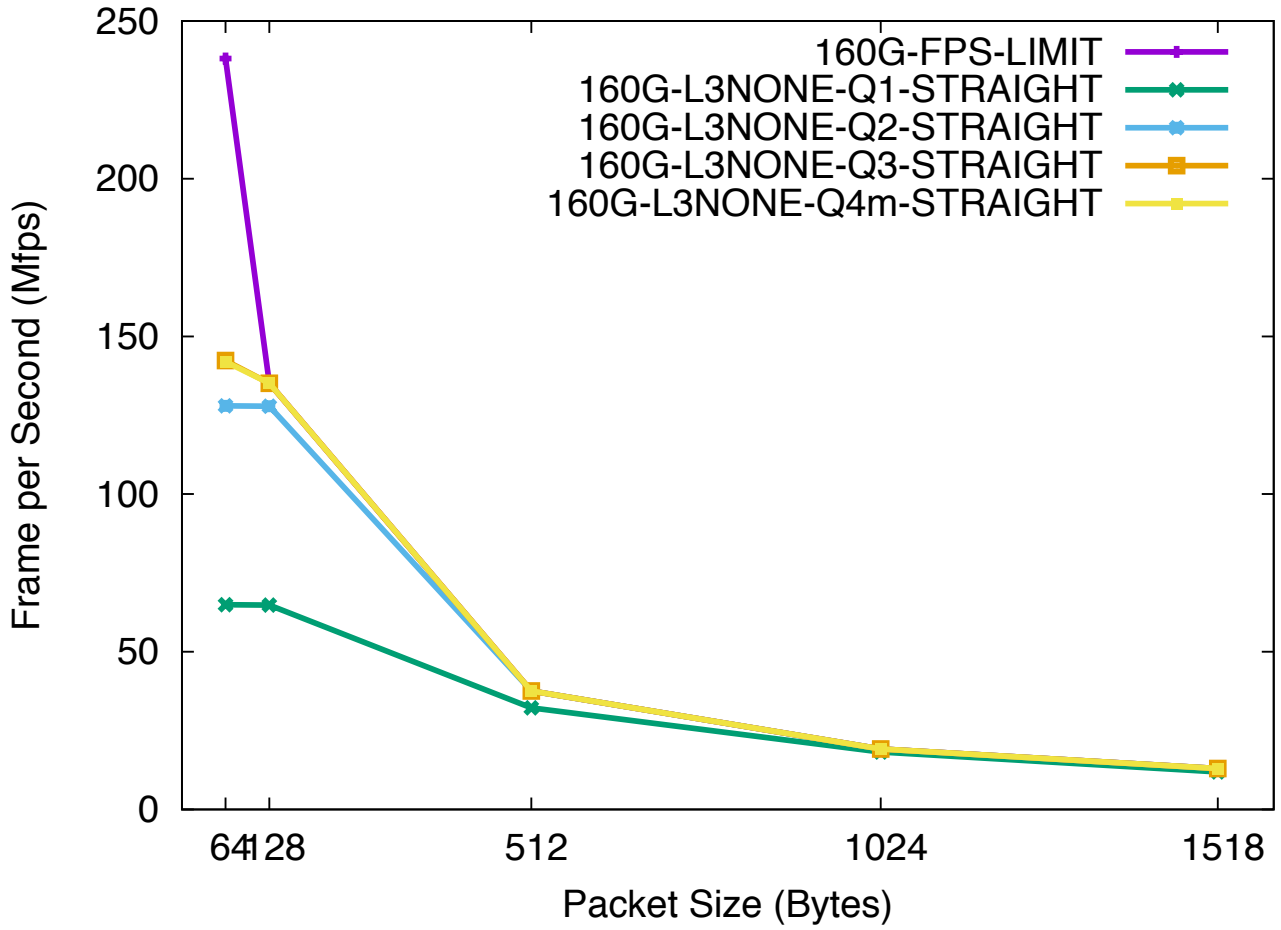
# Single Queue, traffic paths (fps)



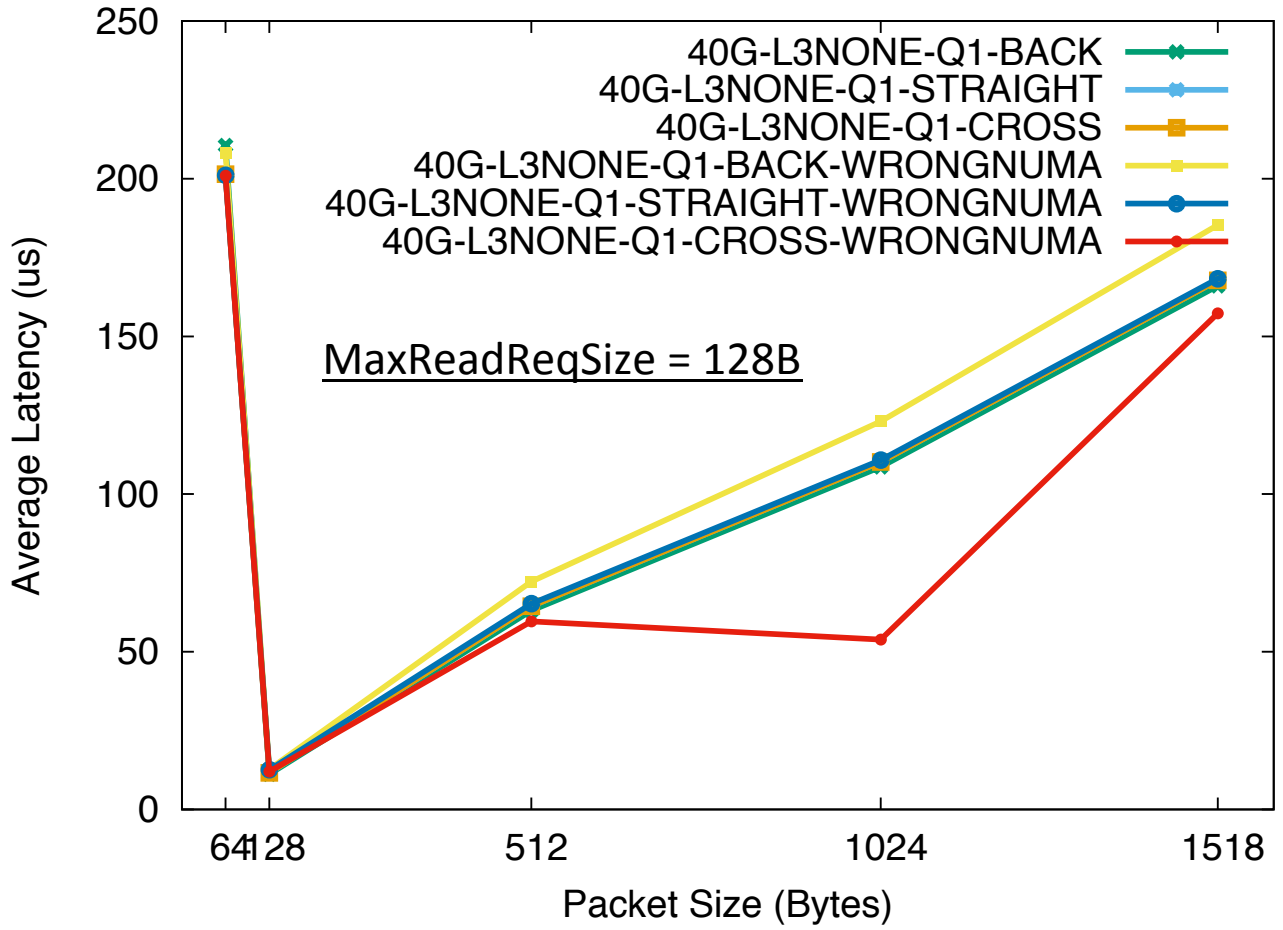
# Peak (160G -> 4 ports) (bps)



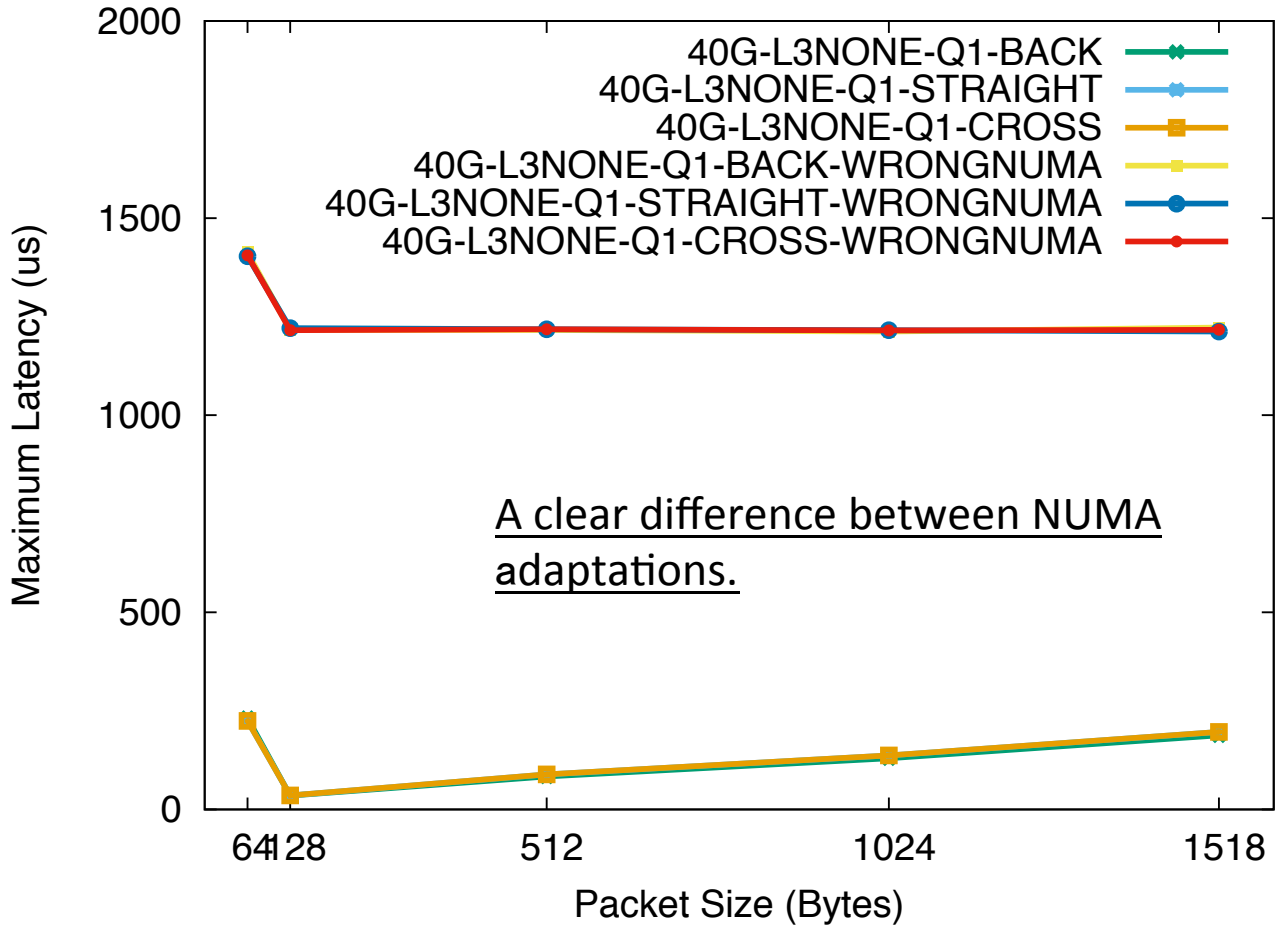
# Peak (160G -> 4 ports) (fps)



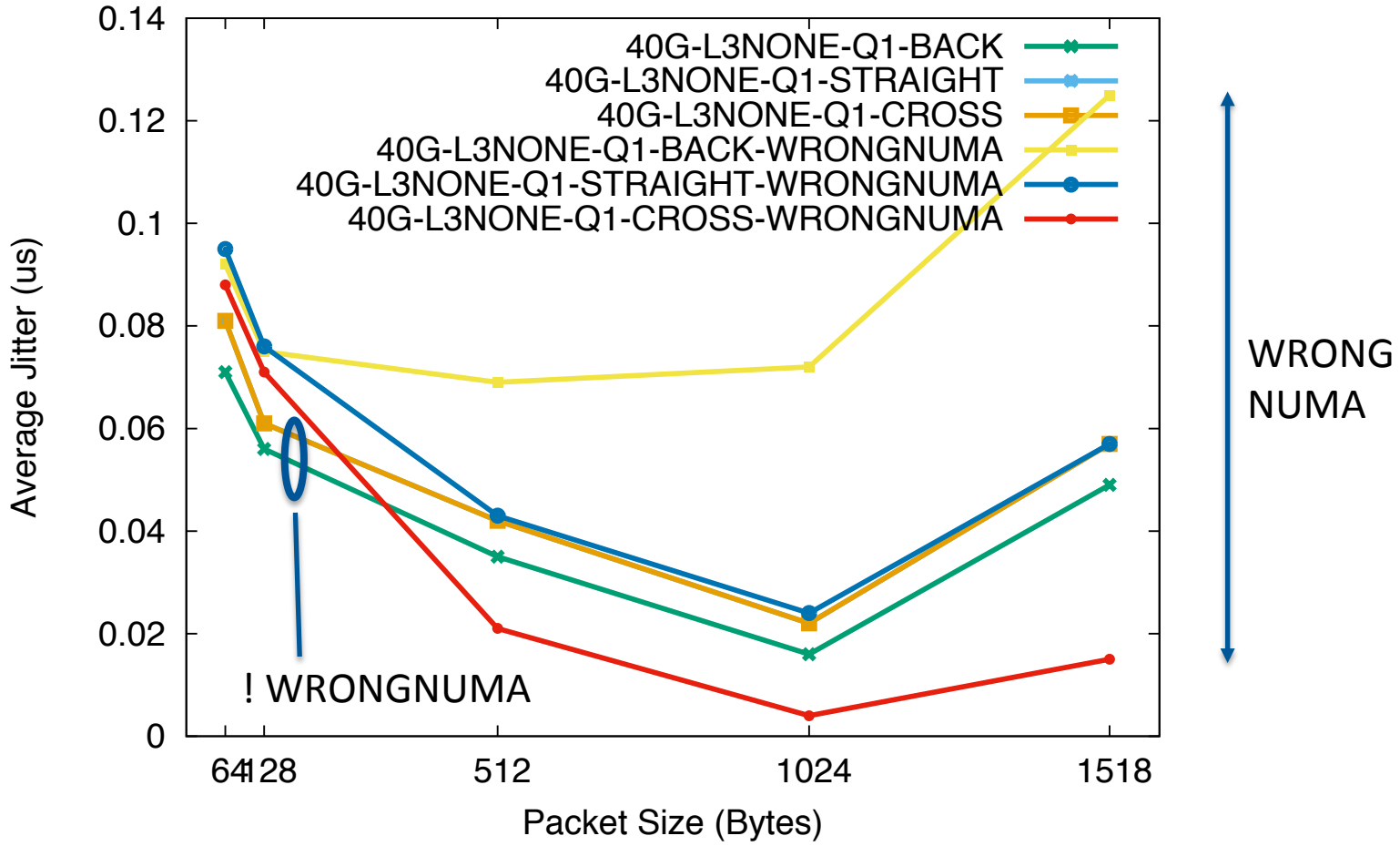
# Average Latency



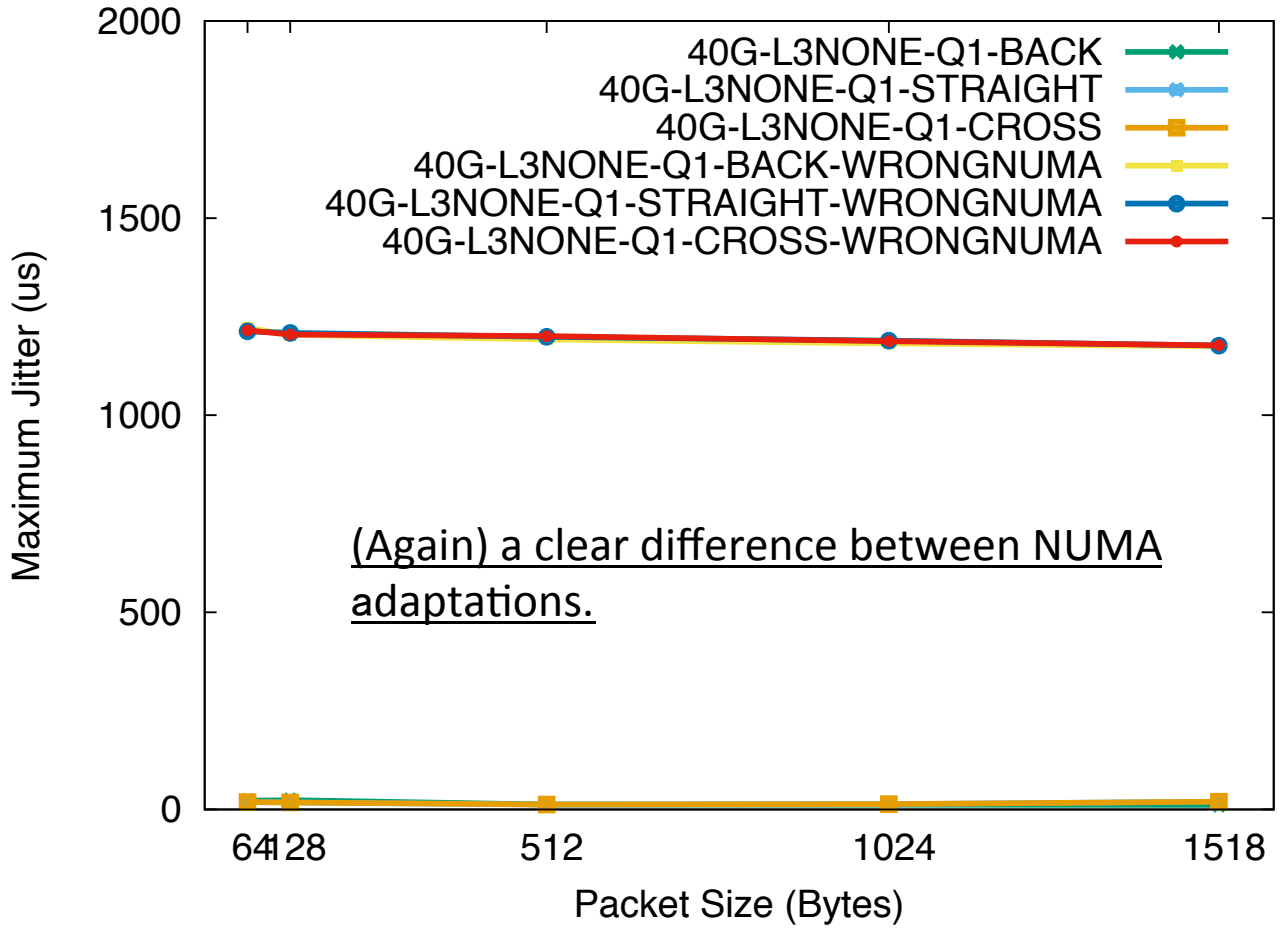
# Maximum Latency



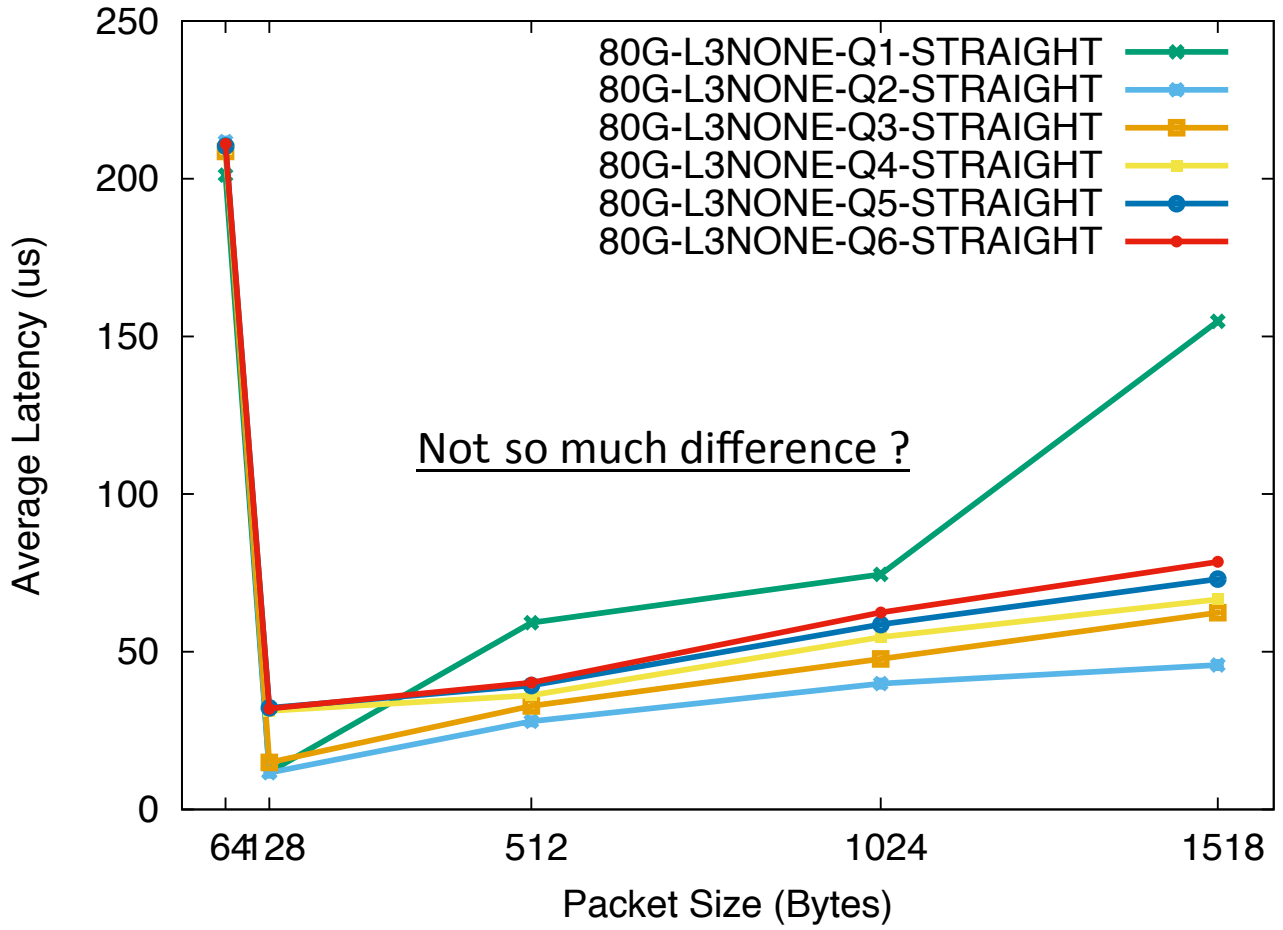
# Average Jitter



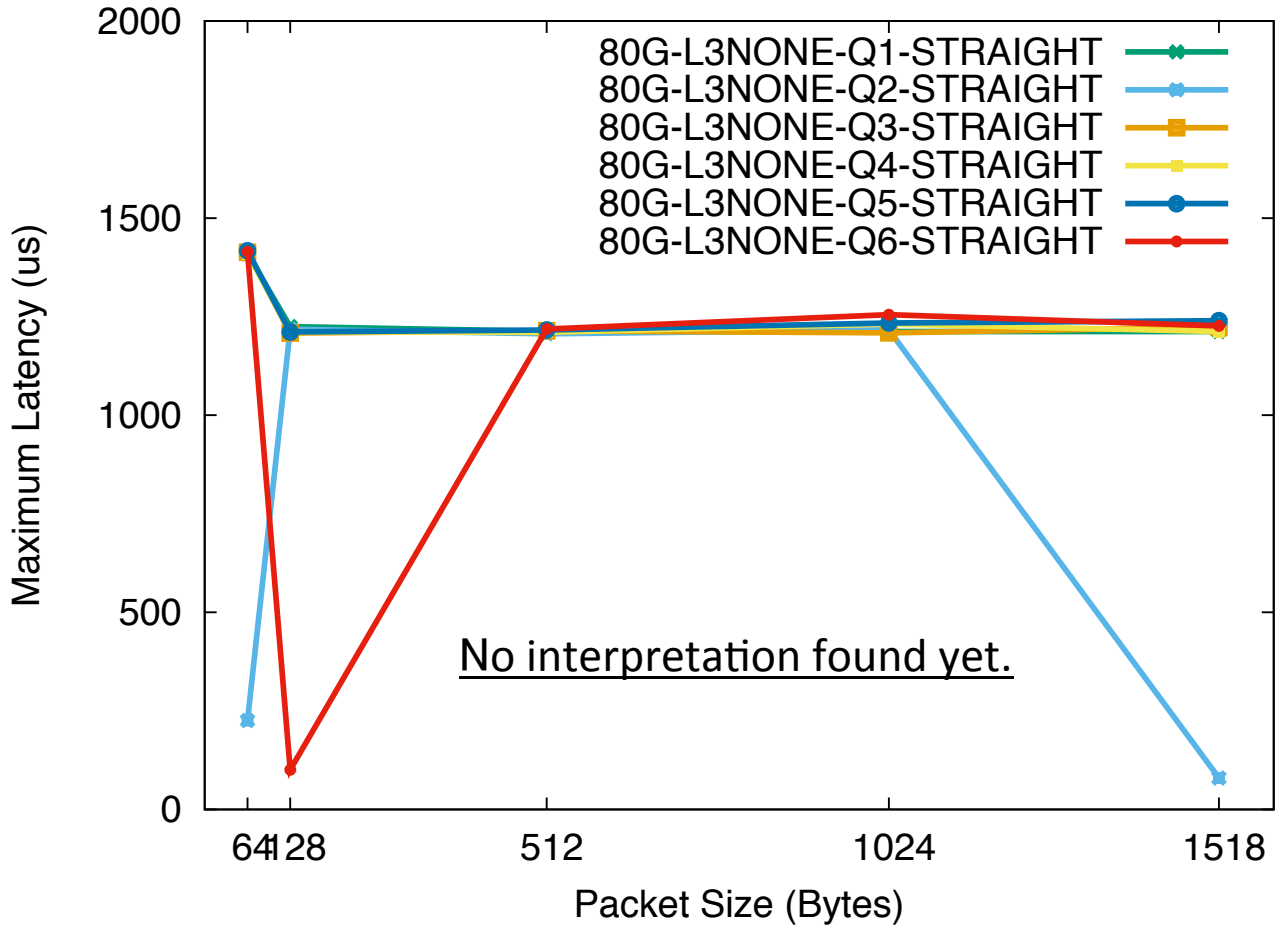
# Max Jitter



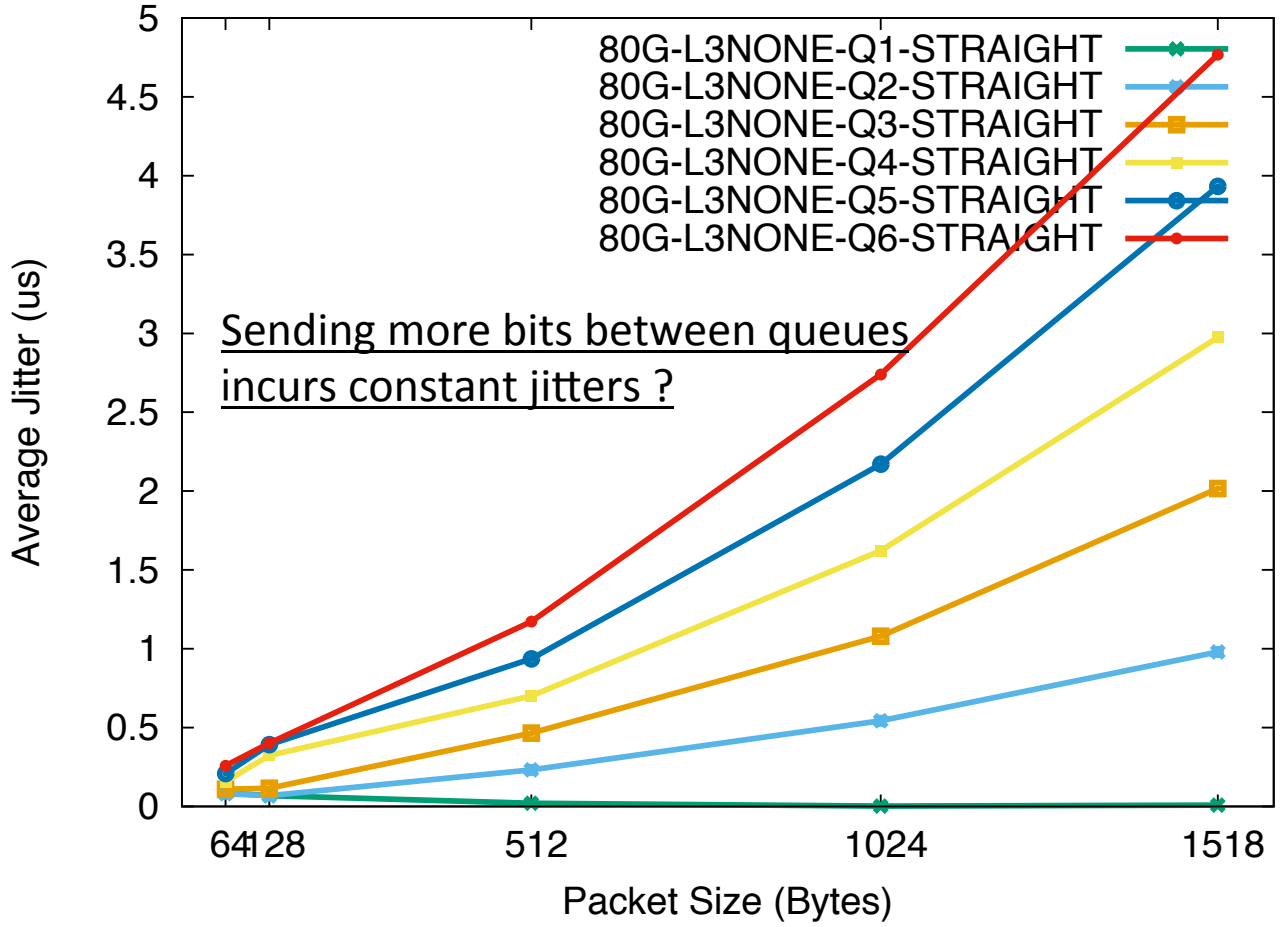
# 80G QX Average Latency



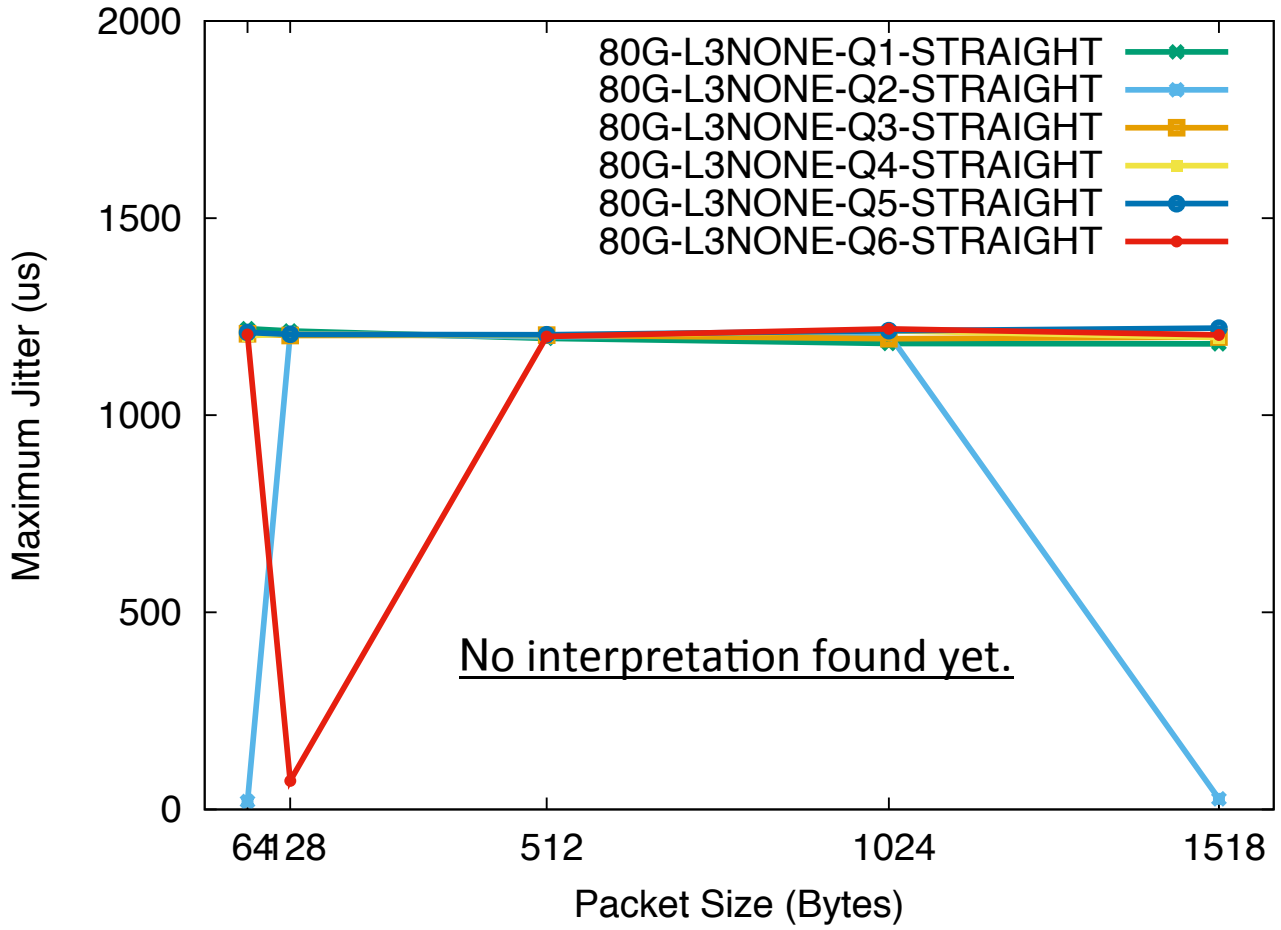
# 80G QX Maximum Latency



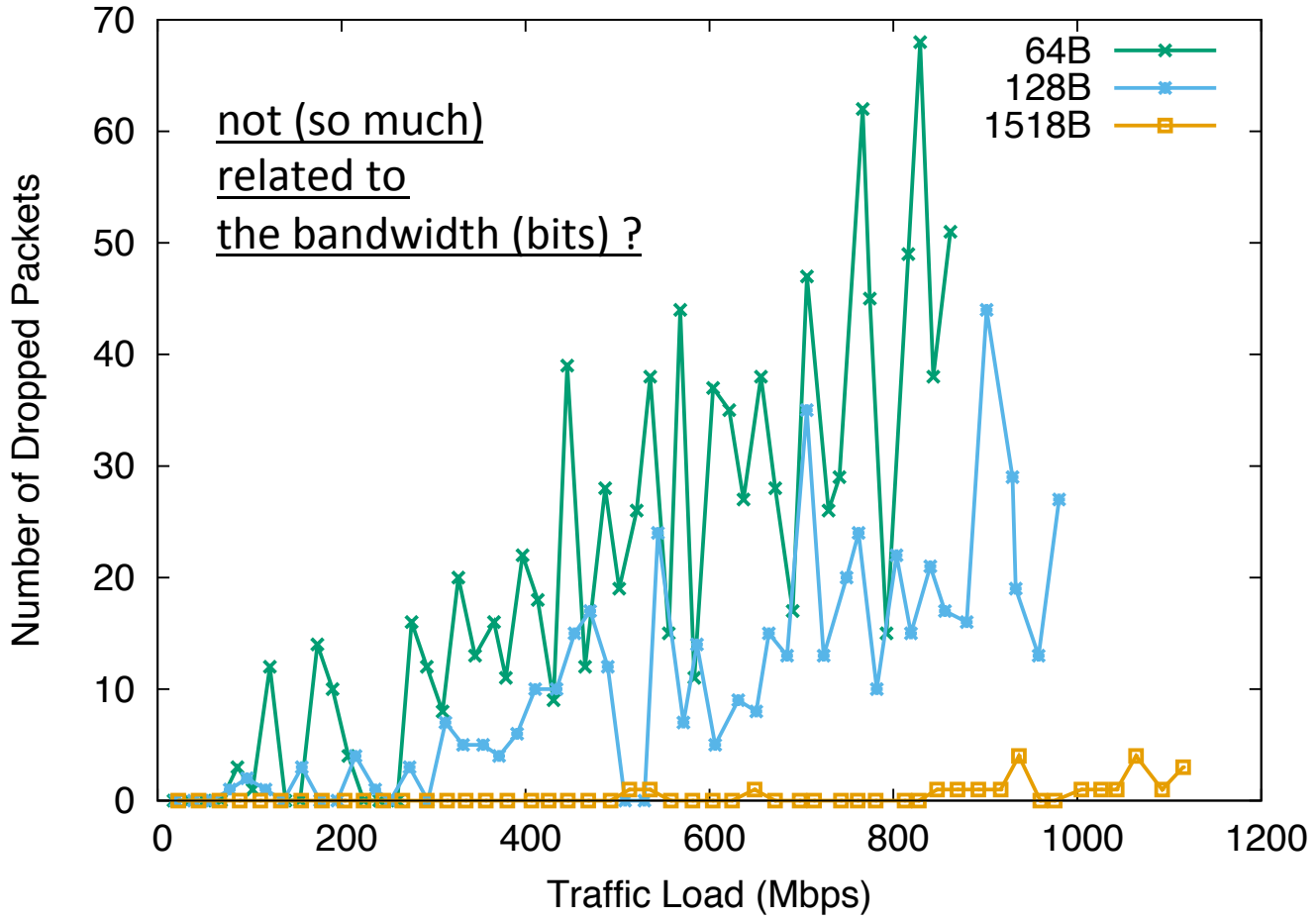
# 80G QX Average Jitter



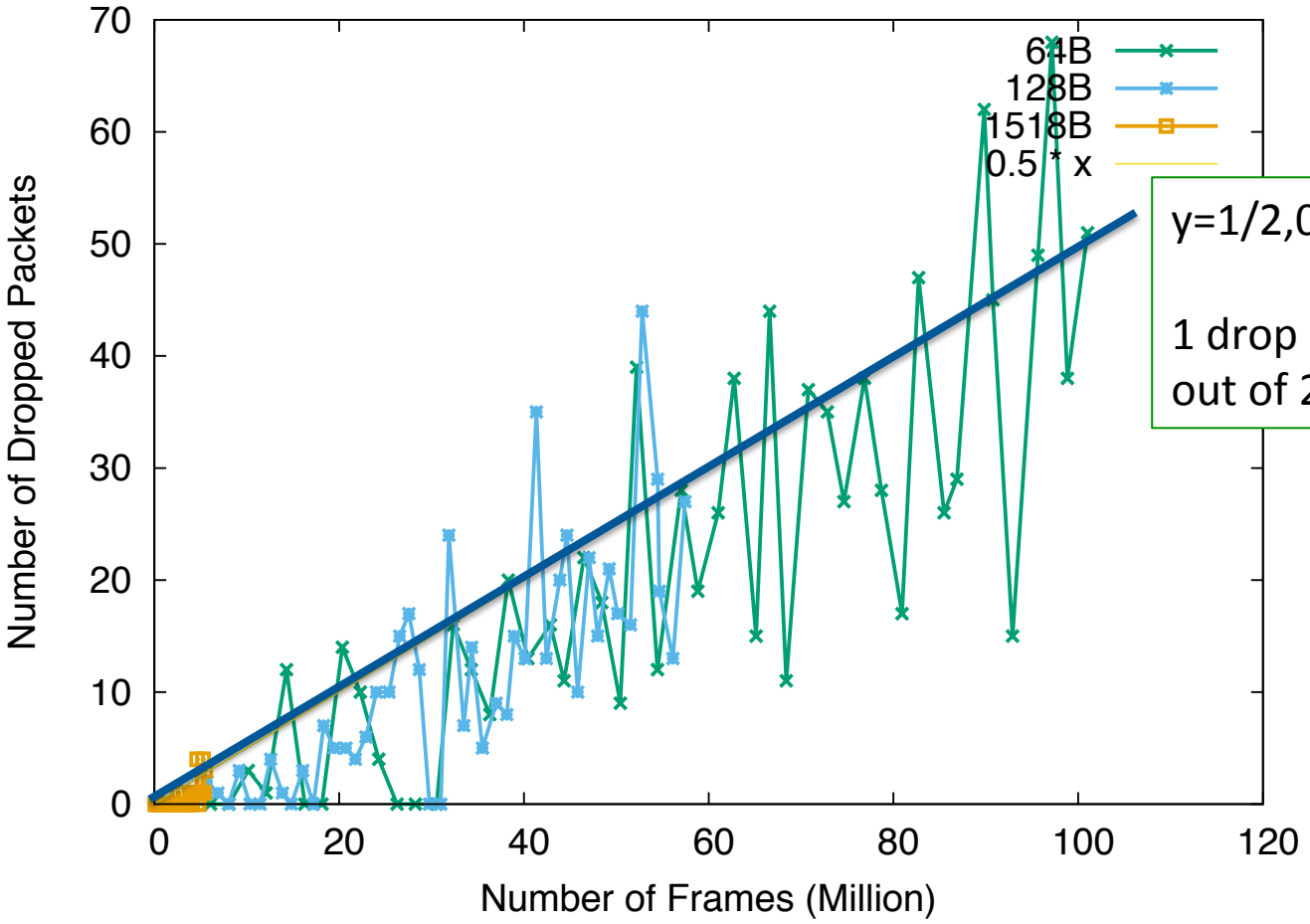
# 80G QX Maximum Jitter



# 20M – 1Gbps Dropped (against Load)



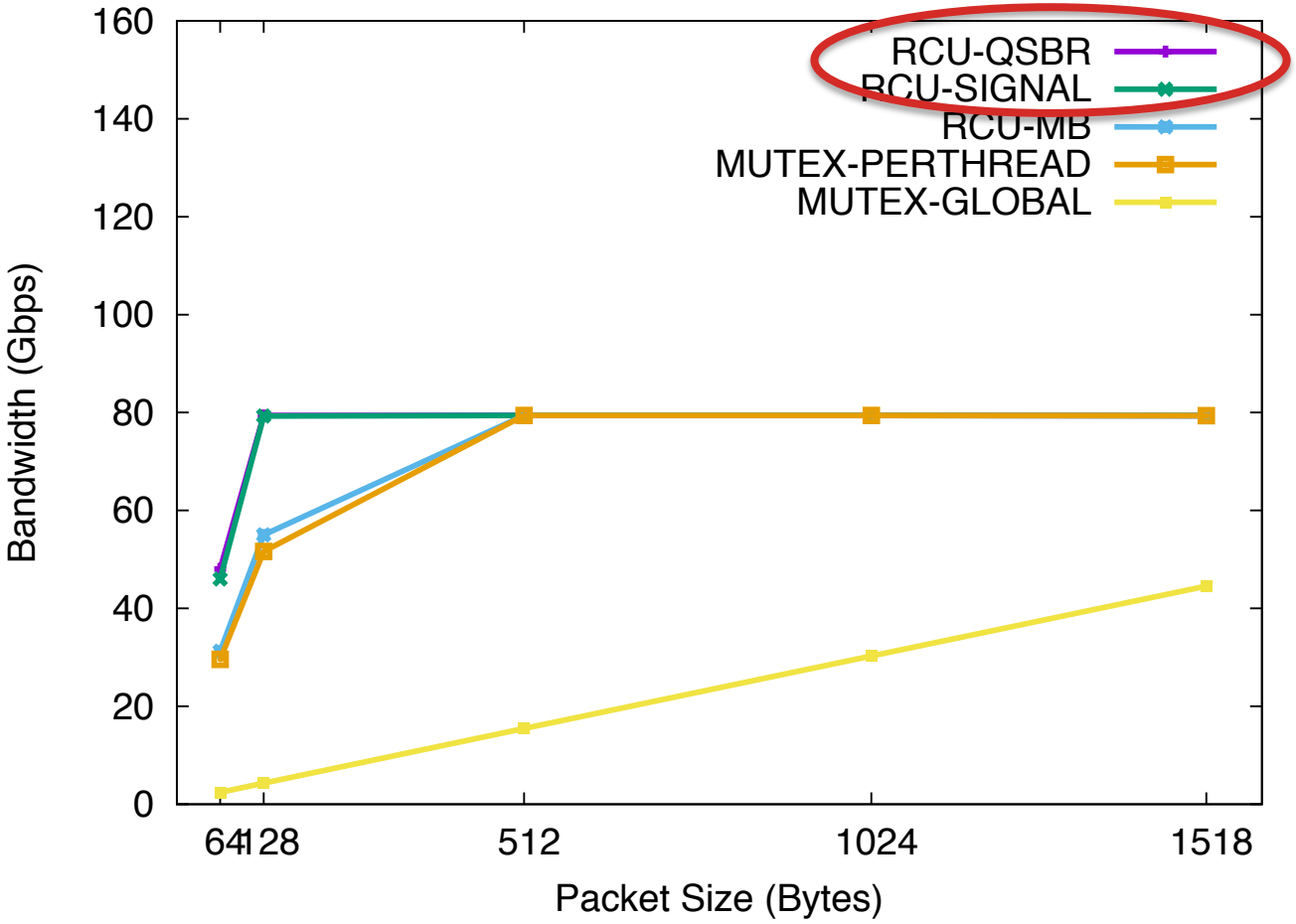
# Million Frames Dropped (against #packets)



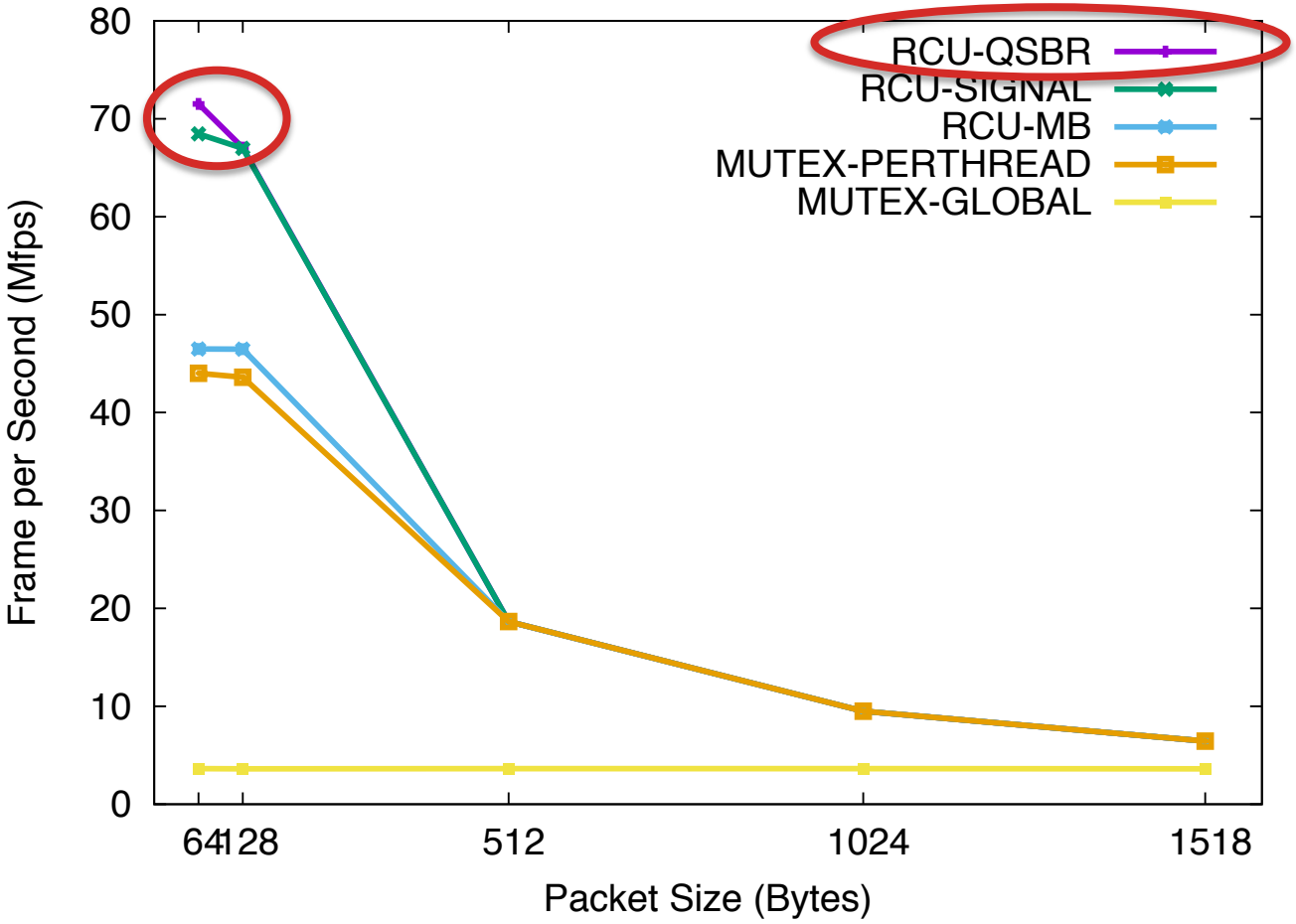

y=1/2,000,000 * x

1 drop
out of 2M frames.

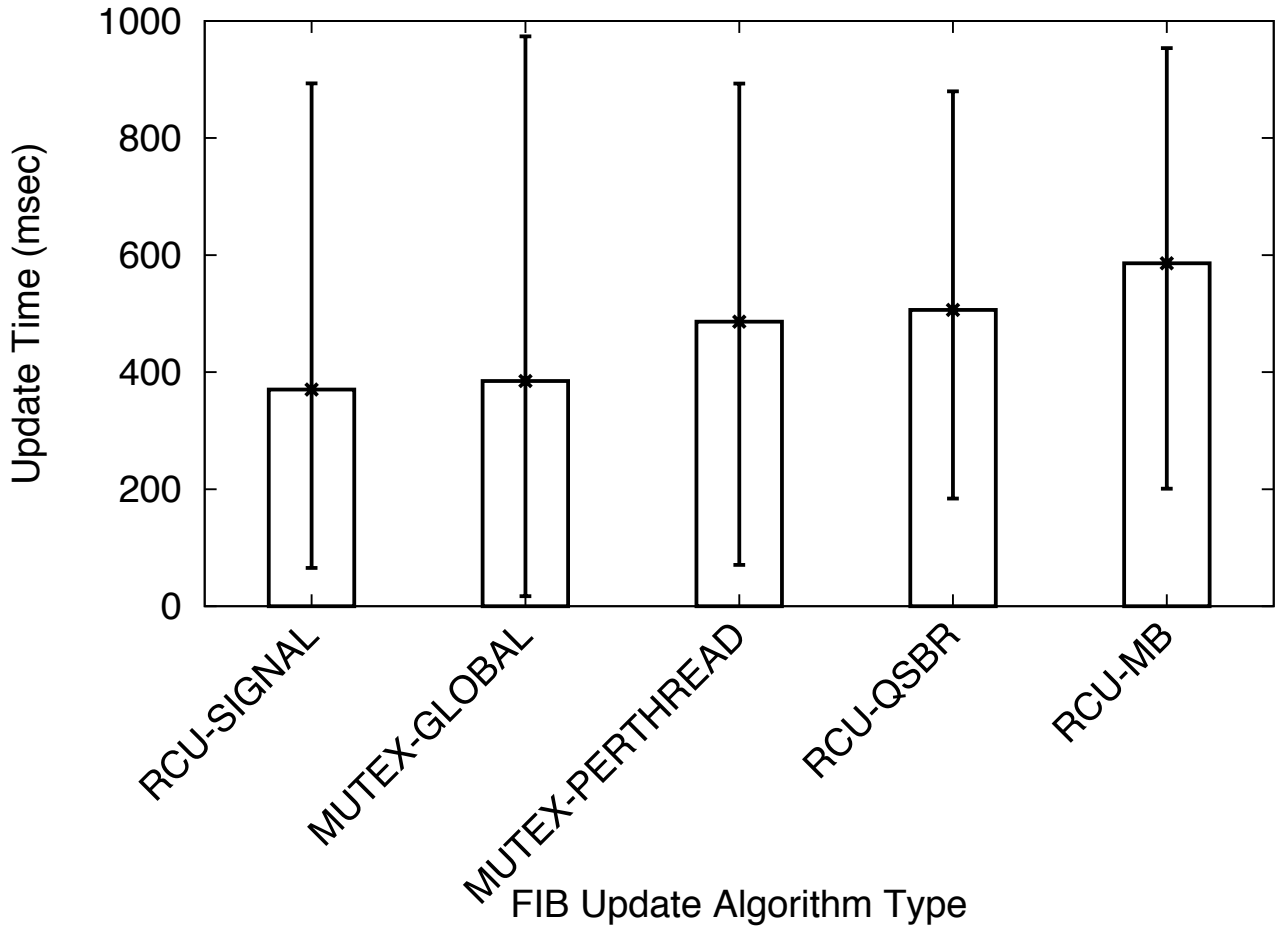
# 80G RCU BPS



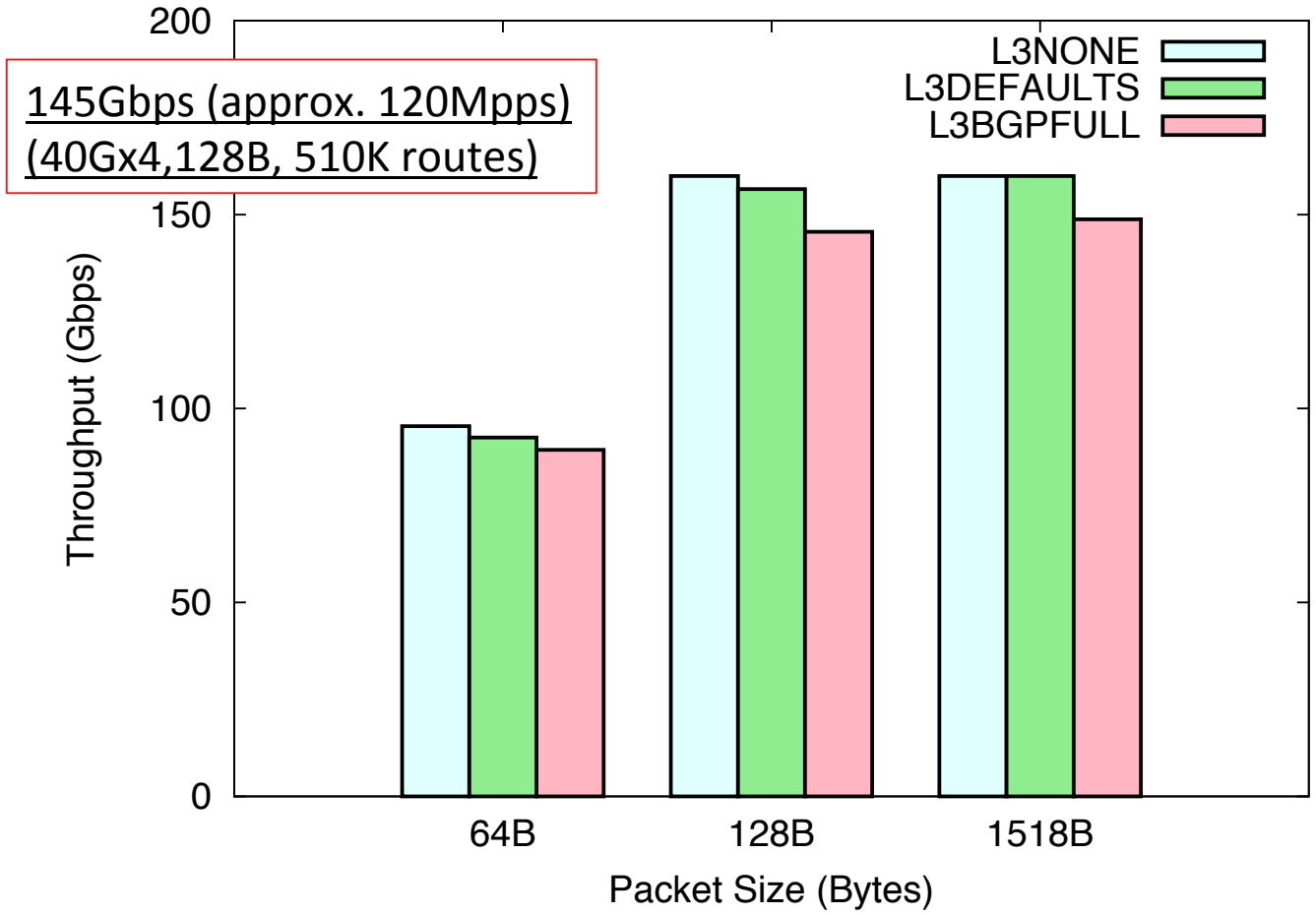
# 80G RCU FPS



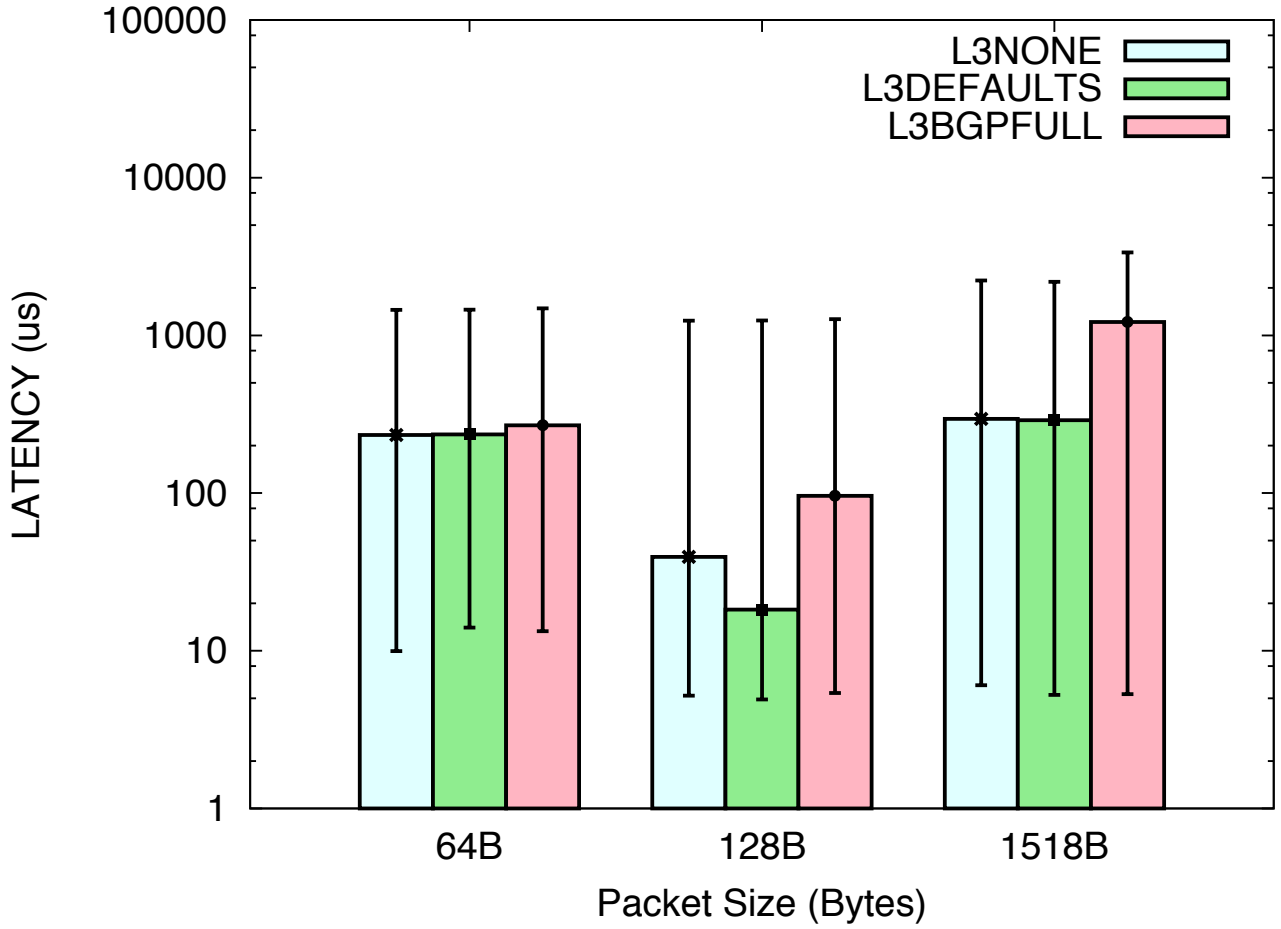
# FIB Update Time



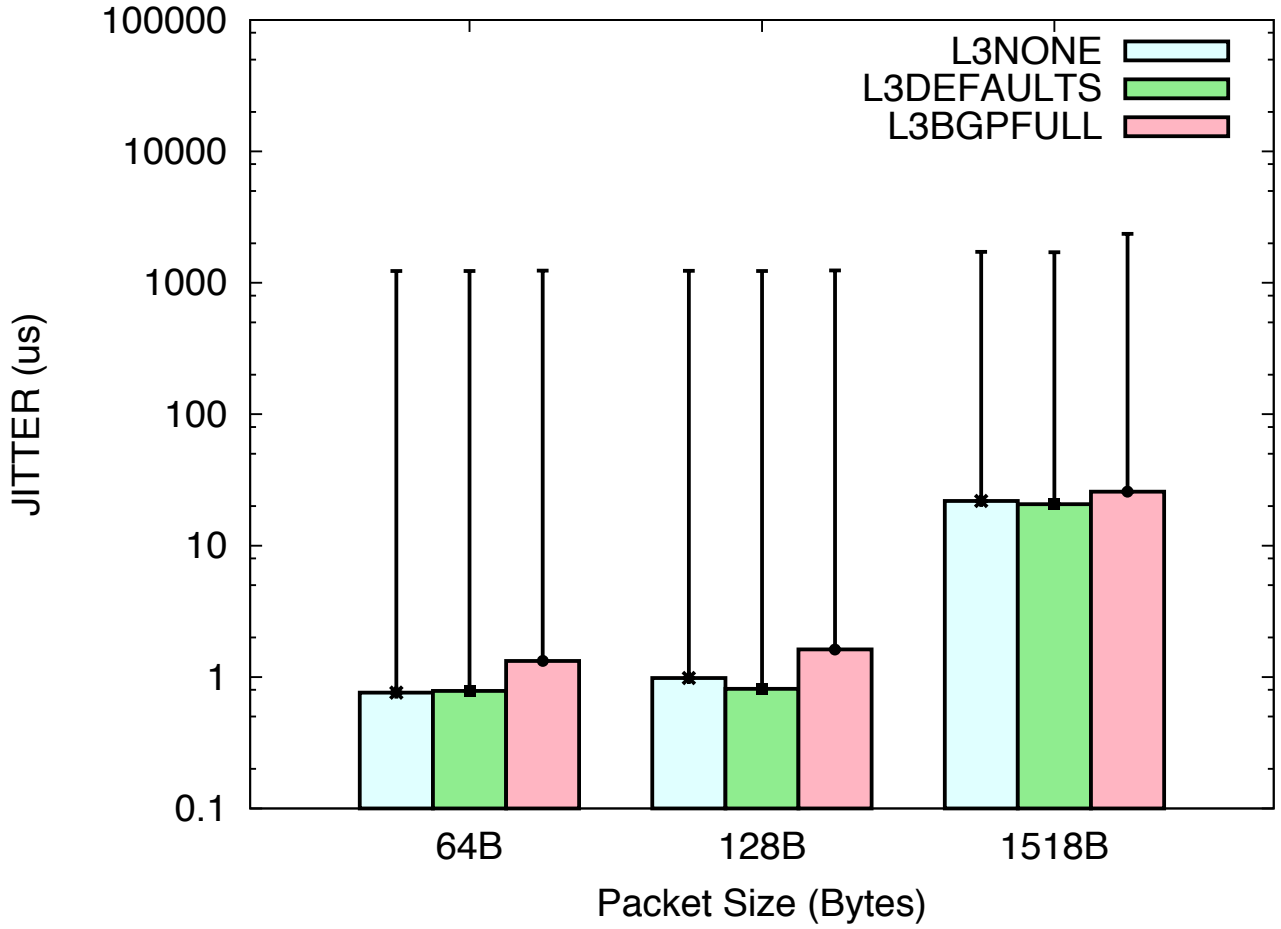
# Overall Throughput



# Overall Latency



# Overall Jitter



# Conclusion

- Kamuee0 Implementation.
  - Integration of Poptrie, RCU, CLI into DPDK l3fwd.
- Achieved 145Gbps in 128B 510K routes (approx. >120Mpps)