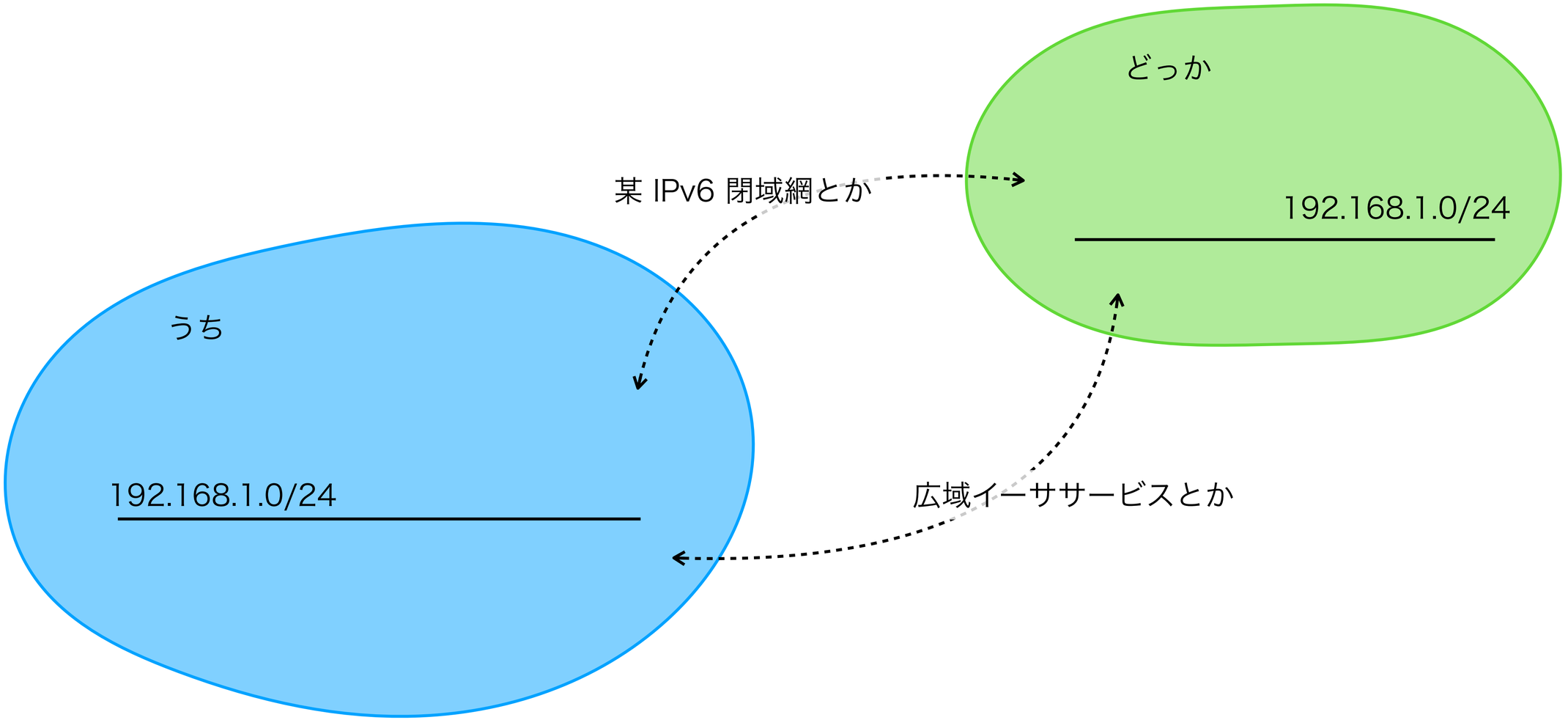
貧乏人のための L2 pseudowire

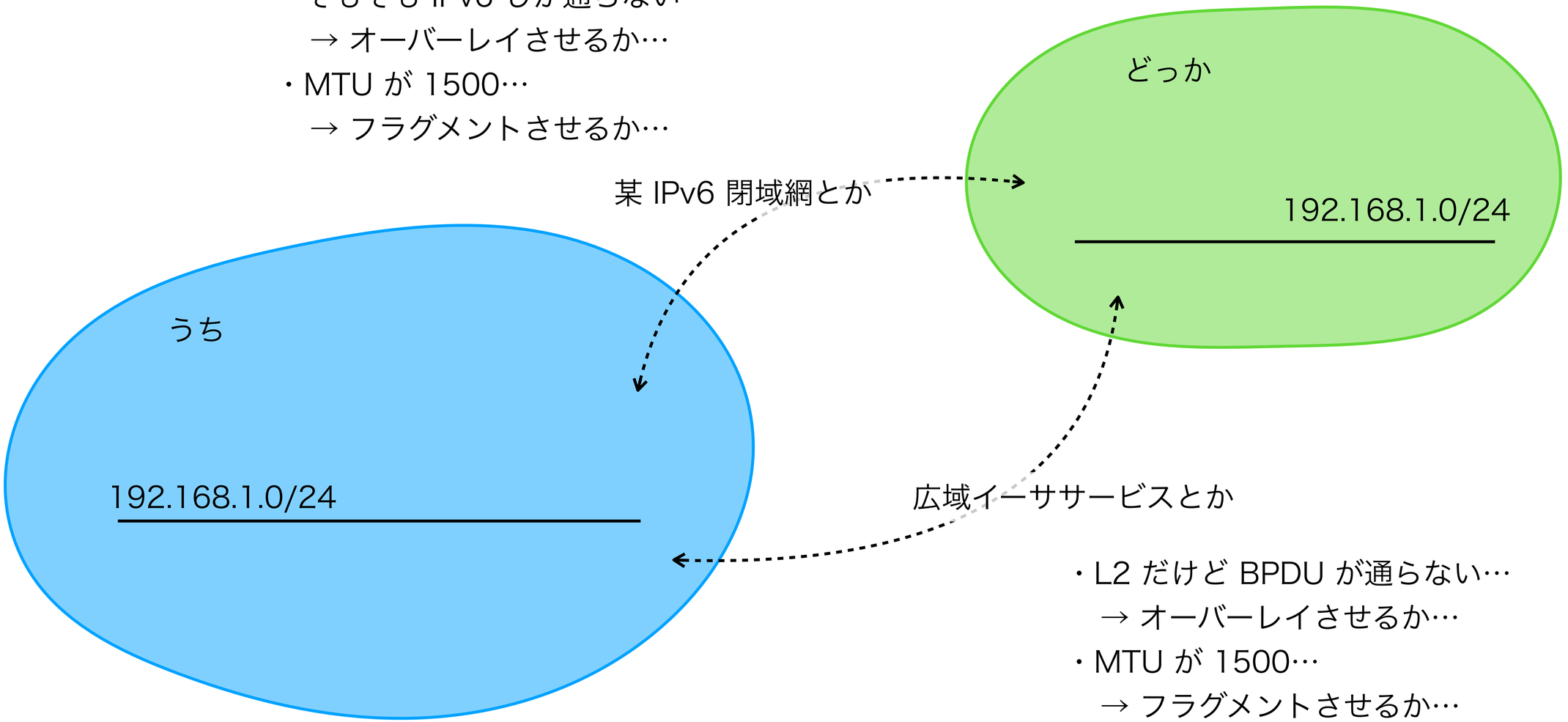
浅間正和@有限会社銀座堂

L2 を(MST で)のばしたい…



L2 を(MST で)のばしたい…

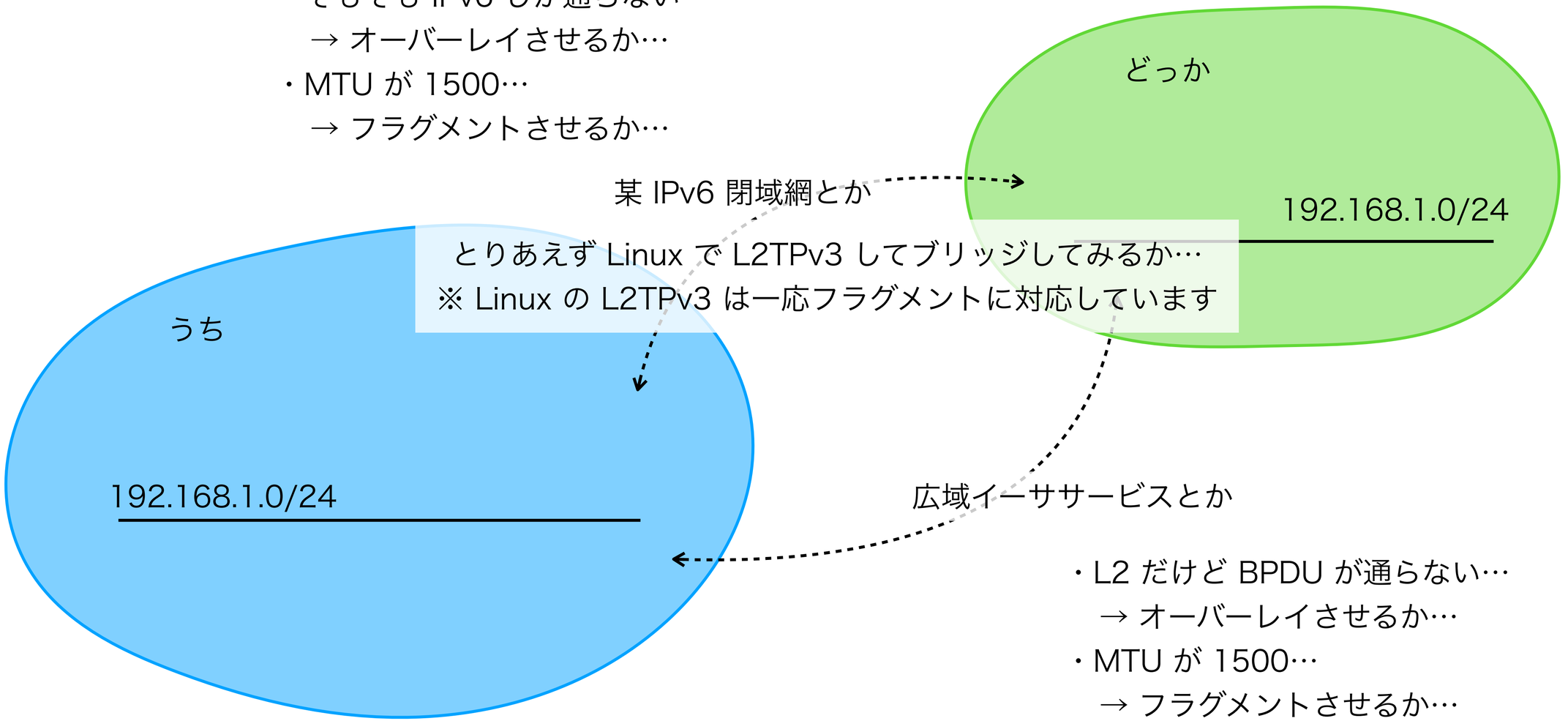
・そもそも IPv6 しか通らない…
　→ オーバーレイさせるか…
・MTU が 1500…
　→ フラグメントさせるか…



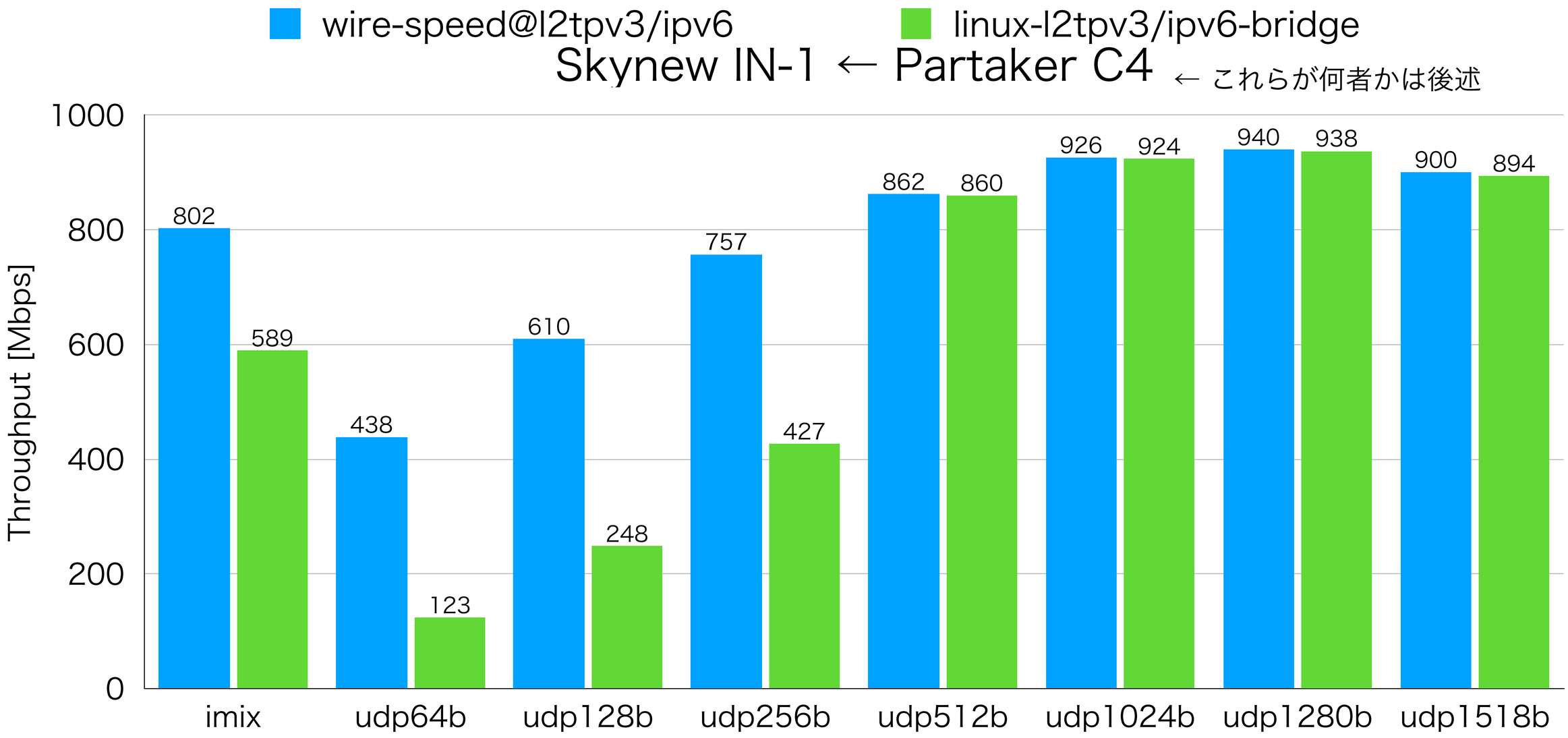
・L2 だけど BPDU が通らない…
　→ オーバーレイさせるか…
・MTU が 1500…
　→ フラグメントさせるか…

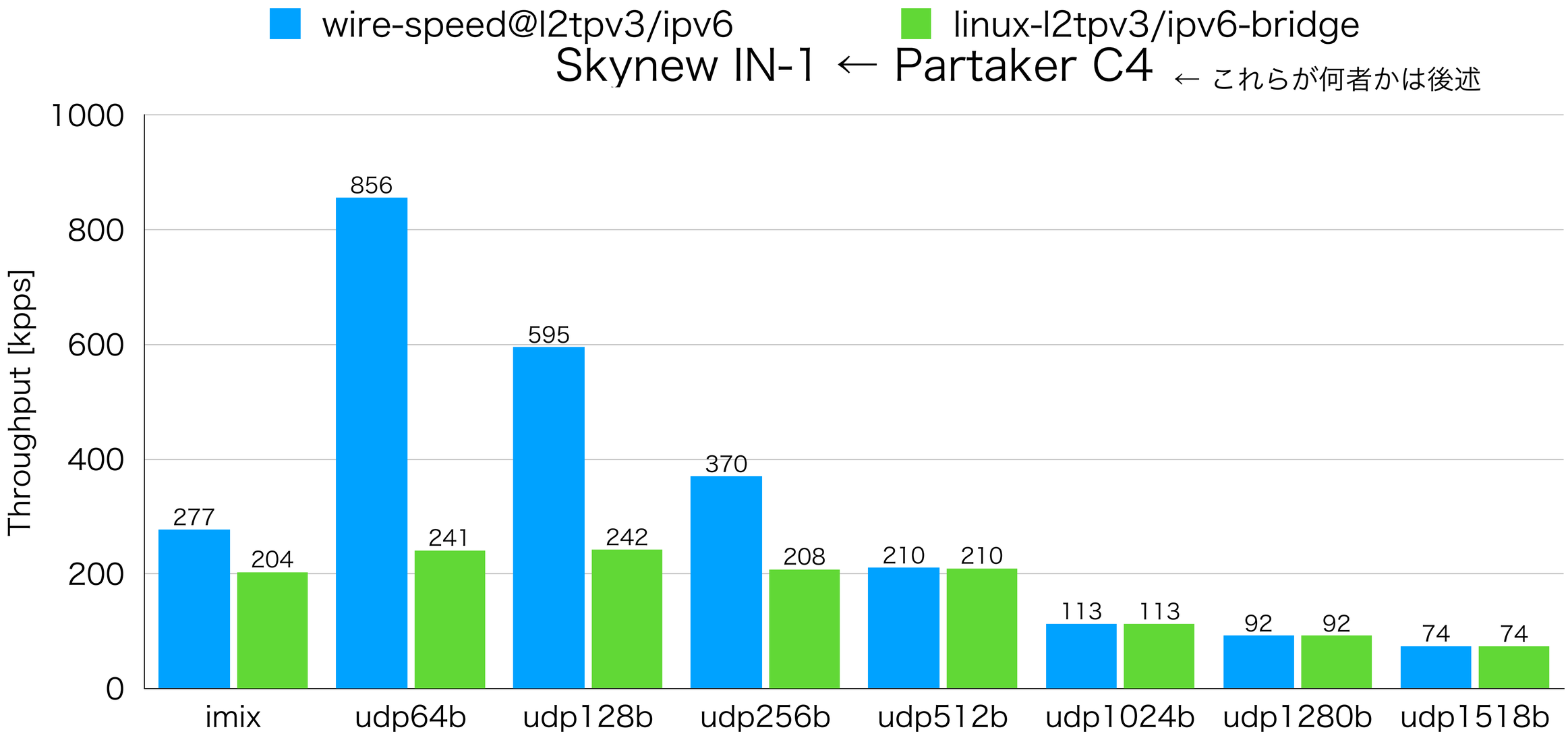
L2 を(MST で)のばしたい…

・そもそも IPv6 しか通らない…
　→ オーバーレイさせるか…
・MTU が 1500…
　→ フラグメントさせるか…



・L2 だけど BPDU が通らない…
　→ オーバーレイさせるか…
・MTU が 1500…
　→ フラグメントさせるか…


wire-speed@l2tpv3/ipv6
linux-l2tpv3/ipv6-bridge
Skynew IN-1 ← Partaker C4
← これらが何者かは後述
Throughput [Mbps]
1000
800
600
400
200
0
802
589
438
123
610
248
757
427
862
860
926
924
940
938
900
894
imix
udp64b
udp128b
udp256b
udp512b
udp1024b
udp1280b
udp1518b


wire-speed@l2tpv3/ipv6
linux-l2tpv3/ipv6-bridge
Skynew IN-1 ← Partaker C4 ← これらが何者かは後述
Throughput [kpps]
1000
800
600
400
200
0
277
204
856
241
595
242
370
208
210
210
113
113
92
92
74
74
imix
udp64b
udp128b
udp256b
udp512b
udp1024b
udp1280b
udp1518b

ということで DPDK で書いてみた

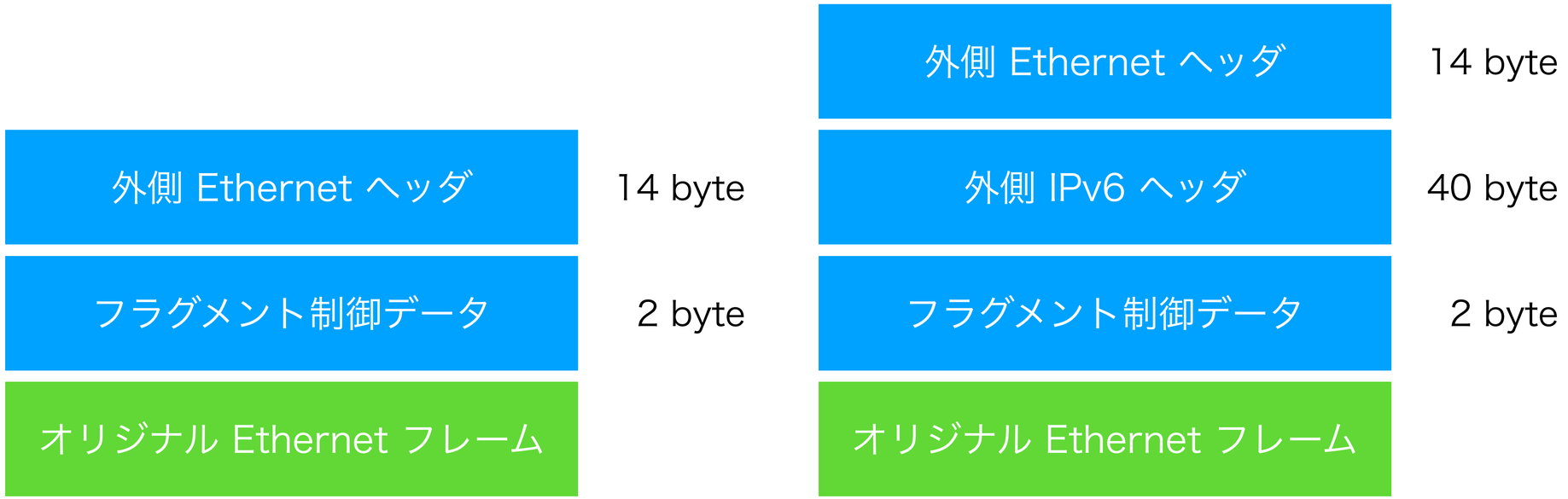
- Ethernet over Ethernet と Ethernet over IPv6 の両方に対応させる
  - ちょっとでもオーバーヘッドを抑えたい…
- over Ethernet でもフラグメントに対応させるために独自ヘッダとして 2 バイト使う(over IPv6 でもフラグメントは同じ仕組みにする)
  - L2TPv3 の 8 バイトよりは小さいのでそれよりは効率的なはず…
- リアセンブルはちゃんとするのが面倒なのでリオーダはせずインオーダで届くことを前提してしまいしかも 2 分割までしか対応しない
  - 完全同一な Ethernet/IPv6 ヘッダのパケットで入れ替わることなんてないですよね…
- over IPv6 は RS 出すのもめんどいので RA をひたすら待つ

おおまかな処理内容

- エンキャップ処理専用スレッドとデキャップ処理専用スレッドとそれ以外の処理スレッド(メインスレッド)の合計 3 つのスレッドをぶん回す
- エンキャップ処理専用スレッド:
  - DL(ダウンリンク)ポートから受信したパケットを受け取りエンキャップしたものを UL(アップリンク)ポートから送信するをひたすら繰り返す
  - UL ポートの MTU に収まらない時はフラグメントし送信する(後述)
- デキャップ処理専用スレッド:
  - UL ポートから受信したパケットを受け取りデキャップしたものを DL ポートから送信するをひたすら繰り返す
  - フラグメントされたパケットの場合はリアセンブルし送信する(後述)

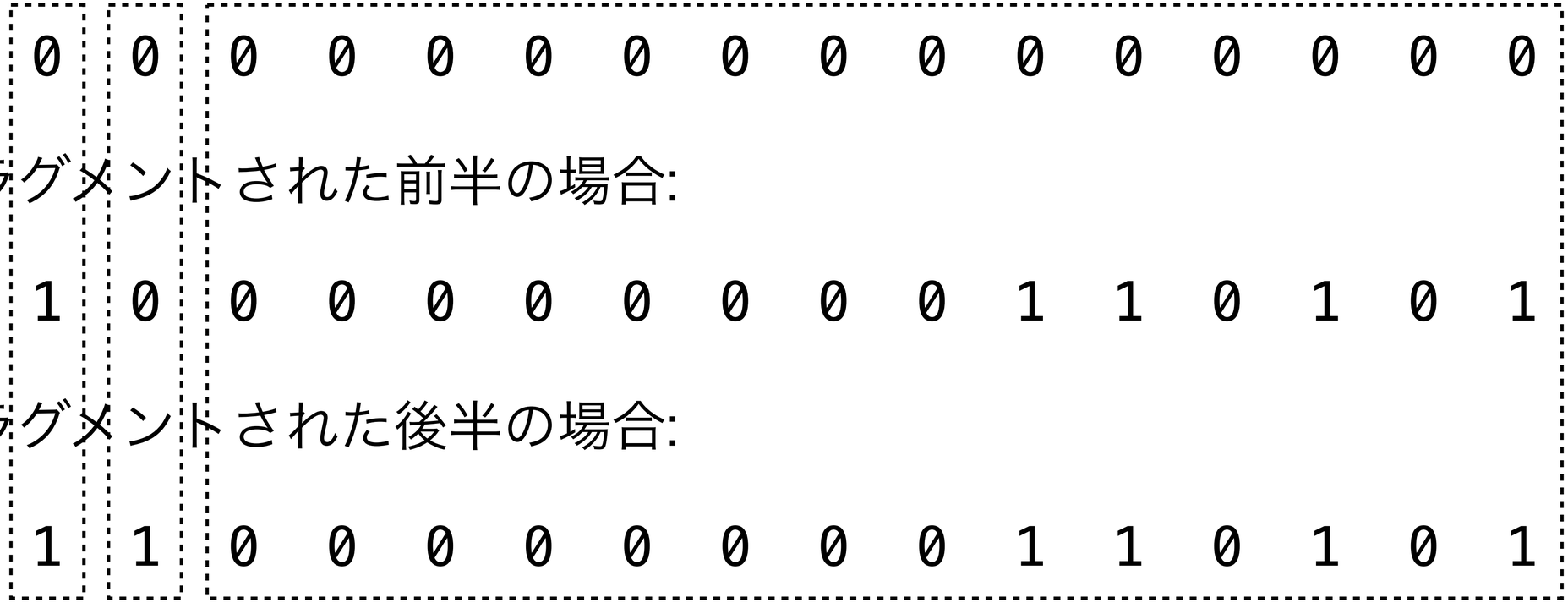
エンキャップとデキャップ

- エンキャップ時のヘッダはこんな感じ



フラグメント制御データ

- フラグメントされていない場合:



- フラグメントされた前半の場合:

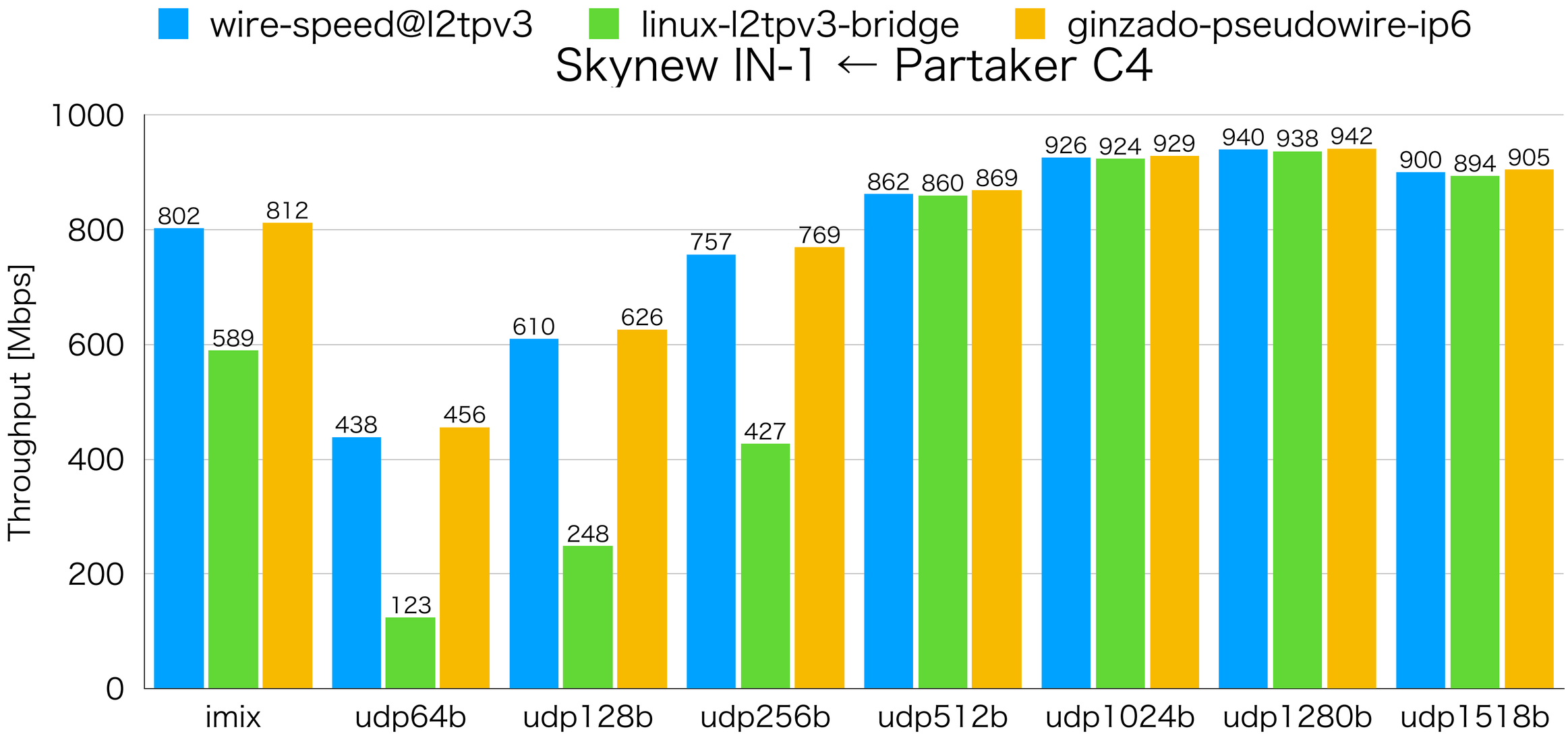
- フラグメントされた後半の場合:

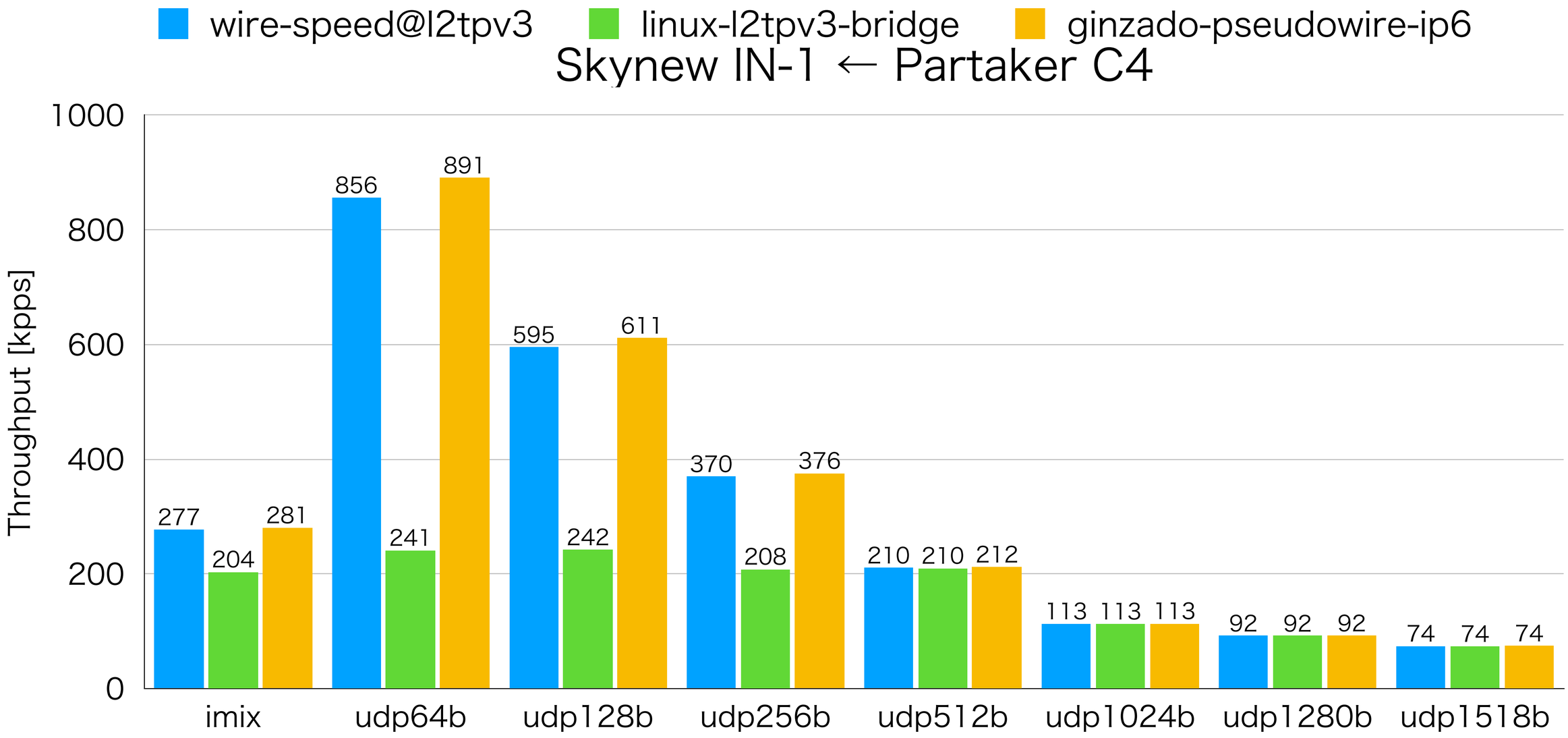
ID(リアセンブル時の判断用; IP の ID と同じようなもの？)(フラグメントする毎にインクリメント)

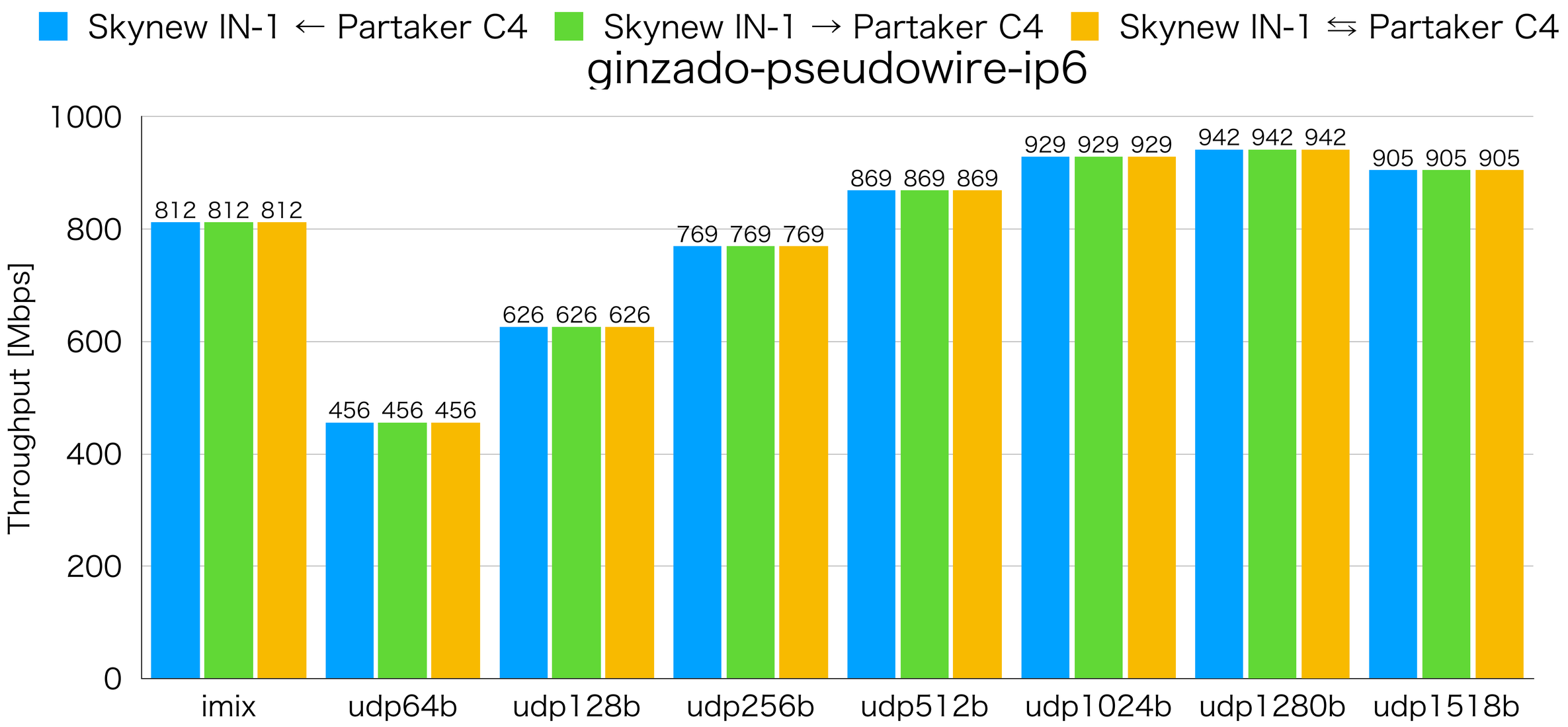
後半フラグ(1: フラグメントされた後半; 0: フラグメントされた前半)(フラグメントフラグ=1の時のみ)

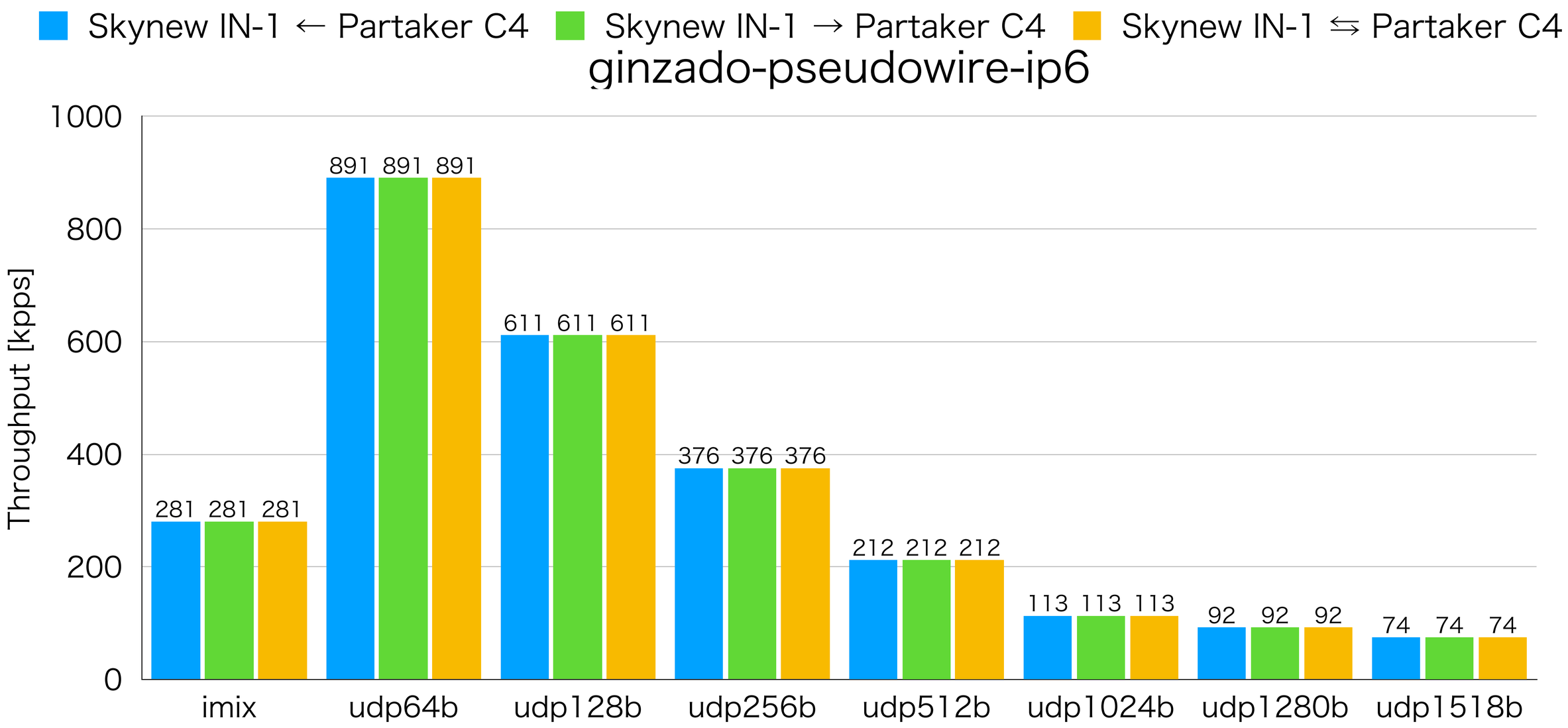
フラグメントフラグ(1: フラグメントされている; 0: フラグメントされていない)

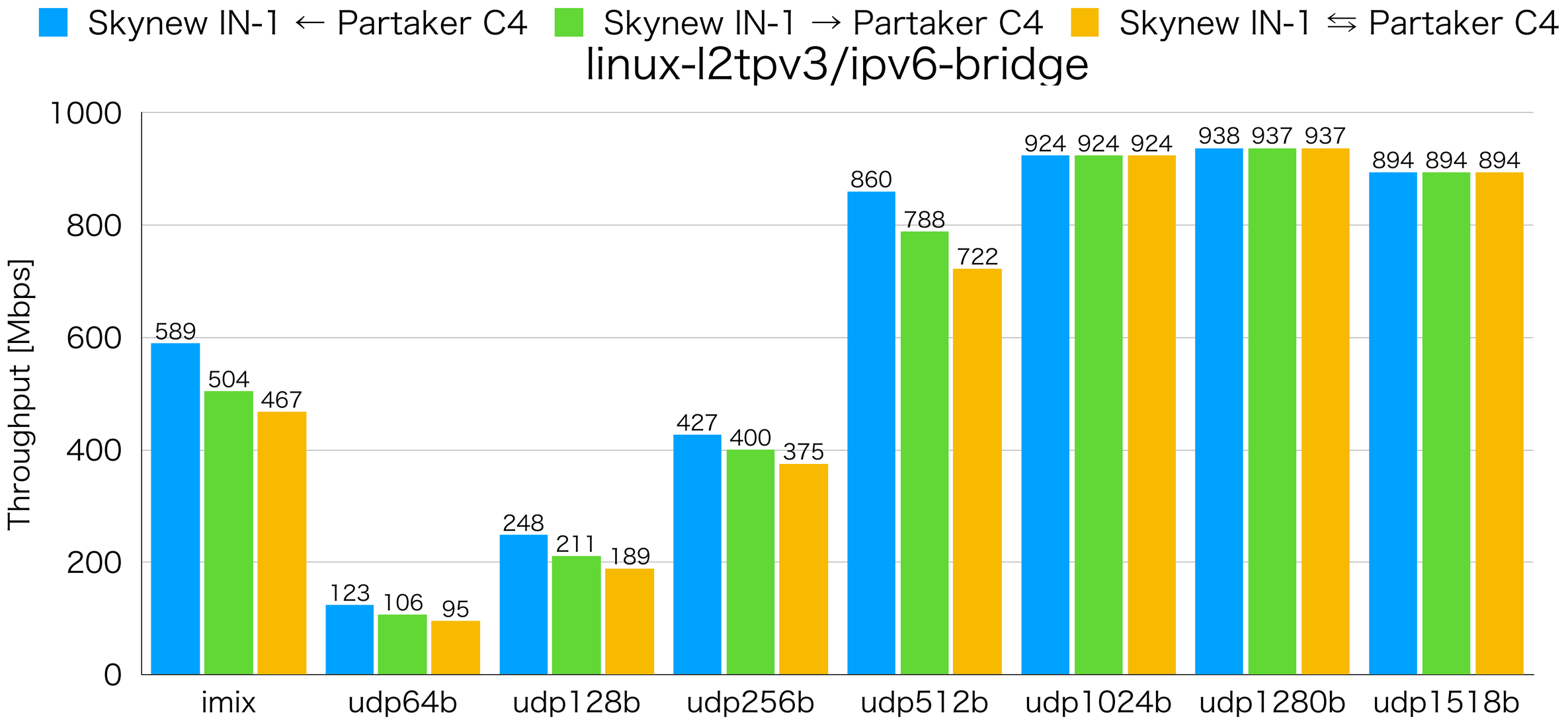
	Skynew IN-1	Partaker C4
外観	 	 
価格	26,990円(税込) (Skynew)	38,421円(税込) (Amazon)
CPU	Intel Celeron N2940 @ 1.83GHz 最大 2.25GHz 4C/4T	Intel Celeron J3160 @ 1.60GHz 最大 2.24GHz 4C/4T
メモリ	DDR3 1333MHz 4GiB	DDR3 1600MHz 4GiB
NIC	Intel I211 1000BASE-T x 4 ポート	Intel I211 1000BASE-T x 4 ポート
備考		Protectli Vault の OEM 品？

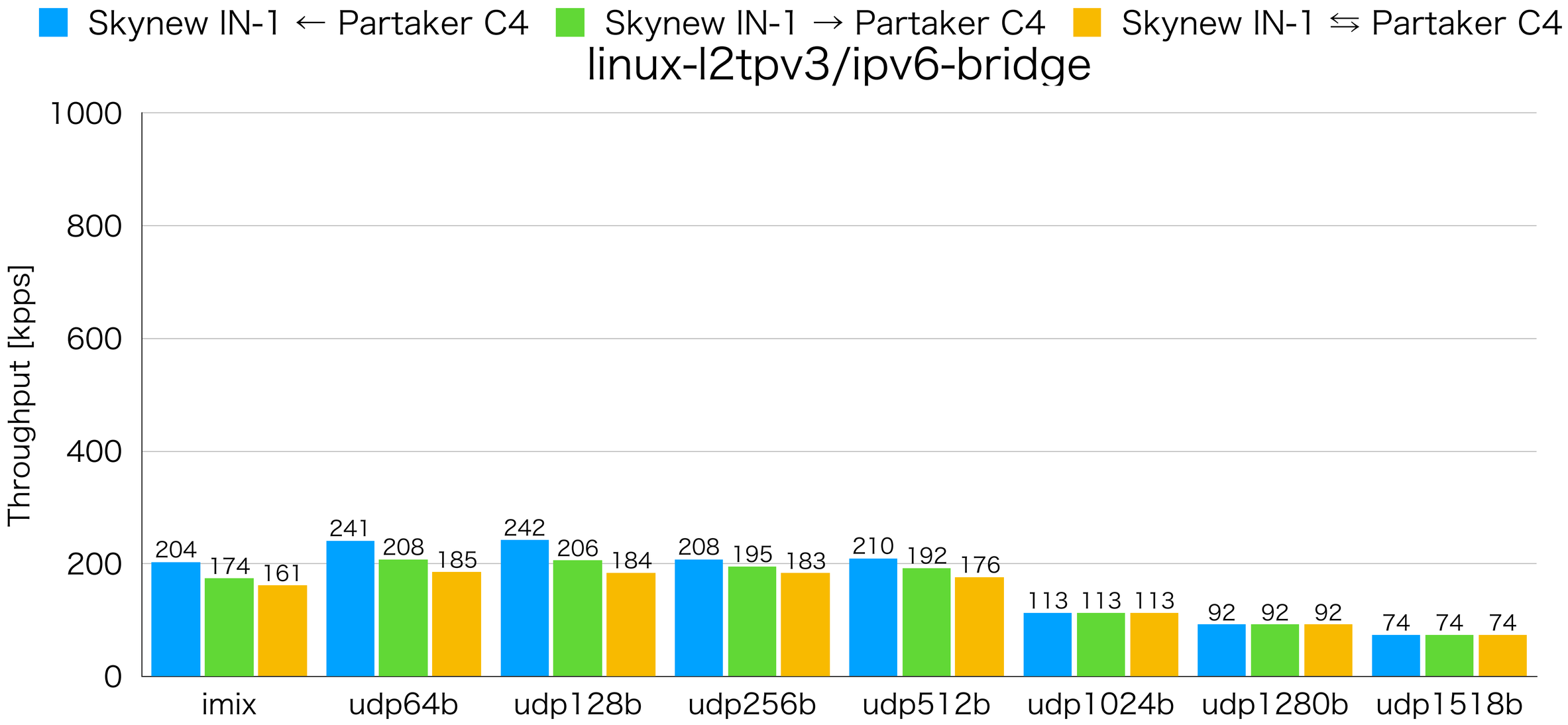

wire-speed@l2tpv3
linux-l2tpv3-bridge
ginzado-pseudowire-ip6
Skynew IN-1 ← Partaker C4
Throughput [Mbps]
1000
800
600
400
200
0
802
589
812
438
123
456
610
248
626
757
427
769
862
860
869
926
924
929
940
938
942
900
894
905
imix
udp64b
udp128b
udp256b
udp512b
udp1024b
udp1280b
udp1518b

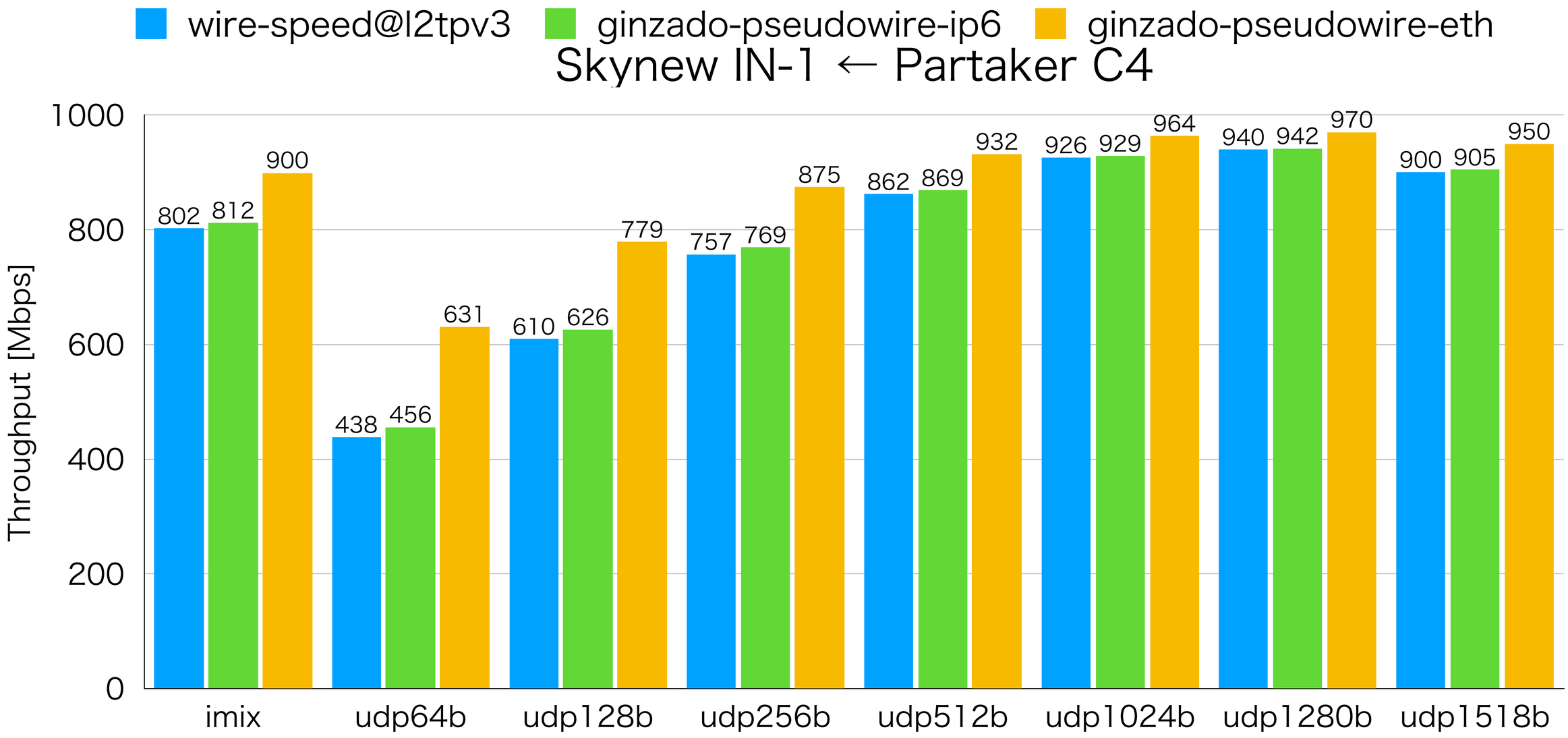

wire-speed@l2tpv3
linux-l2tpv3-bridge
ginzado-pseudowire-ip6
Skynew IN-1 ← Partaker C4
Throughput [kpps]
1000
800
600
400
200
0
277
204
281
856
241
891
595
242
611
370
208
376
210
210
212
113
113
113
92
92
92
74
74
74
imix
udp64b
udp128b
udp256b
udp512b
udp1024b
udp1280b
udp1518b

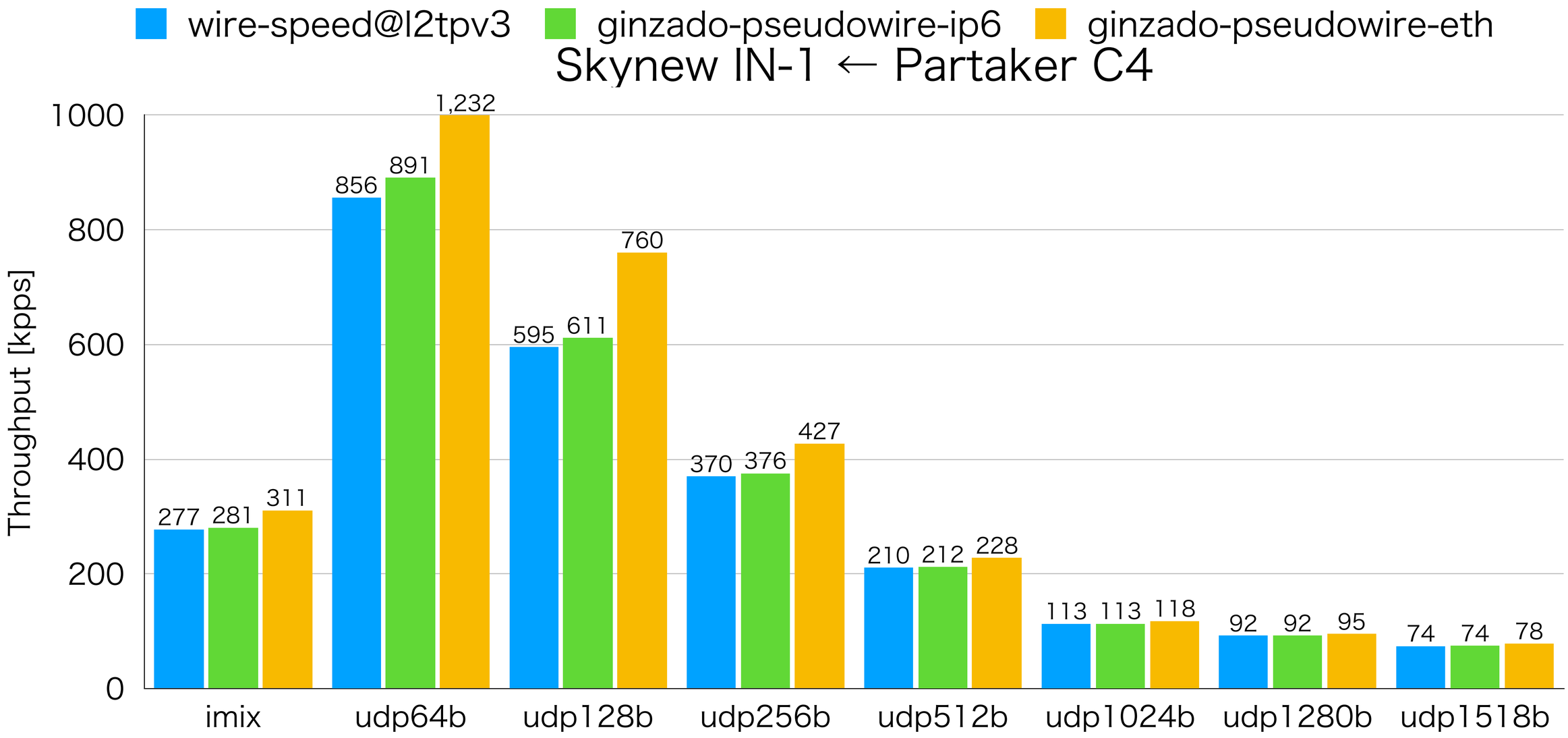

Skynew IN-1 ← Partaker C4
Skynew IN-1 → Partaker C4
Skynew IN-1 ⇆ Partaker C4
ginzado-pseudowire-ip6
Throughput [Mbps]
1000
800
600
400
200
0
812 812 812
456 456 456
626 626 626
769 769 769
869 869 869
929 929 929
942 942 942
905 905 905
imix
udp64b
udp128b
udp256b
udp512b
udp1024b
udp1280b
udp1518b


Skynew IN-1 ← Partaker C4
Skynew IN-1 → Partaker C4
Skynew IN-1 ⇆ Partaker C4
ginzado-pseudowire-ip6
Throughput [kpps]
1000
800
600
400
200
0
281 281 281
891 891 891
611 611 611
376 376 376
212 212 212
113 113 113
92 92 92
74 74 74
imix
udp64b
udp128b
udp256b
udp512b
udp1024b
udp1280b
udp1518b


Skynew IN-1 ← Partaker C4
Skynew IN-1 → Partaker C4
Skynew IN-1 ⇆ Partaker C4
linux-l2tpv3/ipv6-bridge
Throughput [Mbps]
1000
800
600
400
200
0
589
504
467
123
106
95
248
211
189
427
400
375
860
788
722
924
924
924
938
937
937
894
894
894
imix
udp64b
udp128b
udp256b
udp512b
udp1024b
udp1280b
udp1518b


Skynew IN-1 ← Partaker C4
Skynew IN-1 → Partaker C4
Skynew IN-1 ⇆ Partaker C4
linux-l2tpv3/ipv6-bridge
Throughput [kpps]
1000
800
600
400
200
0
204
174
161
241
208
185
242
206
184
208
195
183
210
192
176
113
113
113
92
92
92
74
74
74
imix
udp64b
udp128b
udp256b
udp512b
udp1024b
udp1280b
udp1518b


wire-speed@l2tpv3
ginzado-pseudowire-ip6
ginzado-pseudowire-eth
Skynew IN-1 ← Partaker C4
Throughput [Mbps]
1000
800
600
400
200
0
802
812
900
438
456
631
610
626
779
757
769
875
862
869
932
926
929
964
940
942
970
900
905
950
imix
udp64b
udp128b
udp256b
udp512b
udp1024b
udp1280b
udp1518b


wire-speed@l2tpv3
ginzado-pseudowire-ip6
ginzado-pseudowire-eth
Skynew IN-1 ← Partaker C4
Throughput [kpps]
1000
800
600
400
200
0
277 281 311
856 891 1,232
595 611 760
370 376 427
210 212 228
113 113 118
92 92 95
74 74 78
imix
udp64b
udp128b
udp256b
udp512b
udp1024b
udp1280b
udp1518b

まとめ

- 初期費用両端 60,000円(税込)くらい(某 IPv6 閉域網だけなら)維持費用も月額 12,000円(税込)くらいで L2 伸ばせます
- ただし電気代はもしかしたら結構くうかも？
  - 我が家ではこれを動かしている時に(因果関係は不明ですが) 2 回ブレーカーが落ちました

- ソースコード公開しました
  - https://github.com/ginzado/dpdk

おまけ: BPDU カウンタ

- MST でバックアップが生きてるのか死んでるのか監視したい
  - メインが落ちた際に「実はバックアップも(メインよりも前に)死んでたんだけど気づかず切り替え失敗」みたいなことが起こるのは避けたい
  - 統計カウンタに UL ポートで受け取った(エンキャップされた) BPDU フレームをカウントする機能を実装

```
$ ./gpwstats
gpwstats.ul_rx_packets                    2
gpwstats.ul_rx_bytes                    120
gpwstats.ul_rx_bpdus                      0
gpwstats.ul_tx_packets                    2
gpwstats.ul_tx_bytes                    152
... 省略 ...
```

  - これを使って BPDU フレームが流れてこなくなったらアラートを上げる Nagios プラグインも作成(GitHub のリポジトリにも入ってます)