

XDPによる GTPv1-U/SRv6の Stateless変換の 実装について

Takeru Hayasaka (@takemio10)

Janog47 LT



- 名前:はやさか たける@takemioIO
- 生息地とか:宮城@東北学院大学
🌀 BBSakuraNetworks のしたっぱ
- 好き:食,寝,酒 🍷,XDP,P4,SRv6
- 趣味:
LKML等のtech系メーリス購読
- janog初参加! :)
- 明日は卒論発表会です👤

- 今日話すこと:

与太話

GTPv1+IPv4のSRv6

ステートレス変換の説明

Go+XDPのシステムデザイン

- 今日話さないこと:

SRv6,GTPv1,XDPの大まかな説明

Go+XDPな開発を始めるときのTips

->フォローアップ書いた<https://takeio.hatenablog.com/entry/2021/01/26/180129>

開発においてのワークアラウンドな話



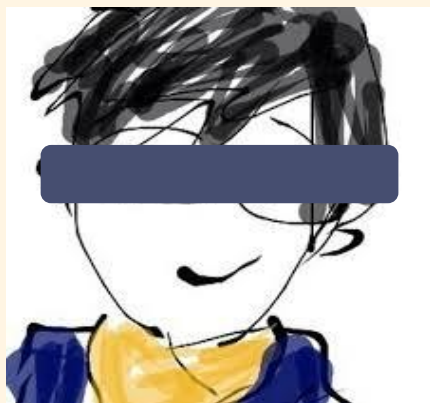
嶽三十六景 神奈川沖
波

時は遡ること一年前



「janog45でXDPネタで
話すんですよー」

「お——いいですね！
なんか手伝いますよ！」



「SRv6とGTPv1の変換なんですけ
ど....雑にやりませんか？w」

「やります！！！！
(何も考えてない顔)」



引き受けたはいいが
期限は残り一週間！

なんか手伝いますよ！」



「SRv6とGTPv1の変換なんですけど....雑にやりませんか？w」

「やります！！！！
(何も考えてない顔)」





引き受けたはいいが
期限は残り一週間！

なんか手伝いますよ！」



何なら大学のテスト期間！！

（何も考えてない顔）」



引き受けた方がいいが

Me

Task

（何も考えてない顔）」

大変申し訳ありませんが、動くコードを実装
できませんでした🙇🙇🙇

※開発はアルバイトで来ている東北学院大学3年生の早坂彪流さんにご協力いただきました。

ありがとうございました🙇🙇🙇

パケット処理の独自実装や高速化手法の比較と実践 | JANOG45@札幌 | 2020/01/24

「何も考えない顔」

cf. https://www.janog.gr.jp/meeting/janog45/application/files/1615/7984/1029/008_pktfwd-xdp_kusakabe_00.pdf



Endは動くのでEnd部分の解説をします

ソース

ス:https://github.com/takehaya/srv6-gtp/blob/feature/t_m_GTP4_D/src/srv6.c

パケット処理の独自実装や高速化手法の比較と実践 | JANOG45@札幌 | 2020/01/24

「何も考えない顔」

cf. https://www.janog.gr.jp/meeting/janog45/application/files/1615/7984/1029/008_pktfwd-xdp_kusakabe_00.pdf





リベンジ

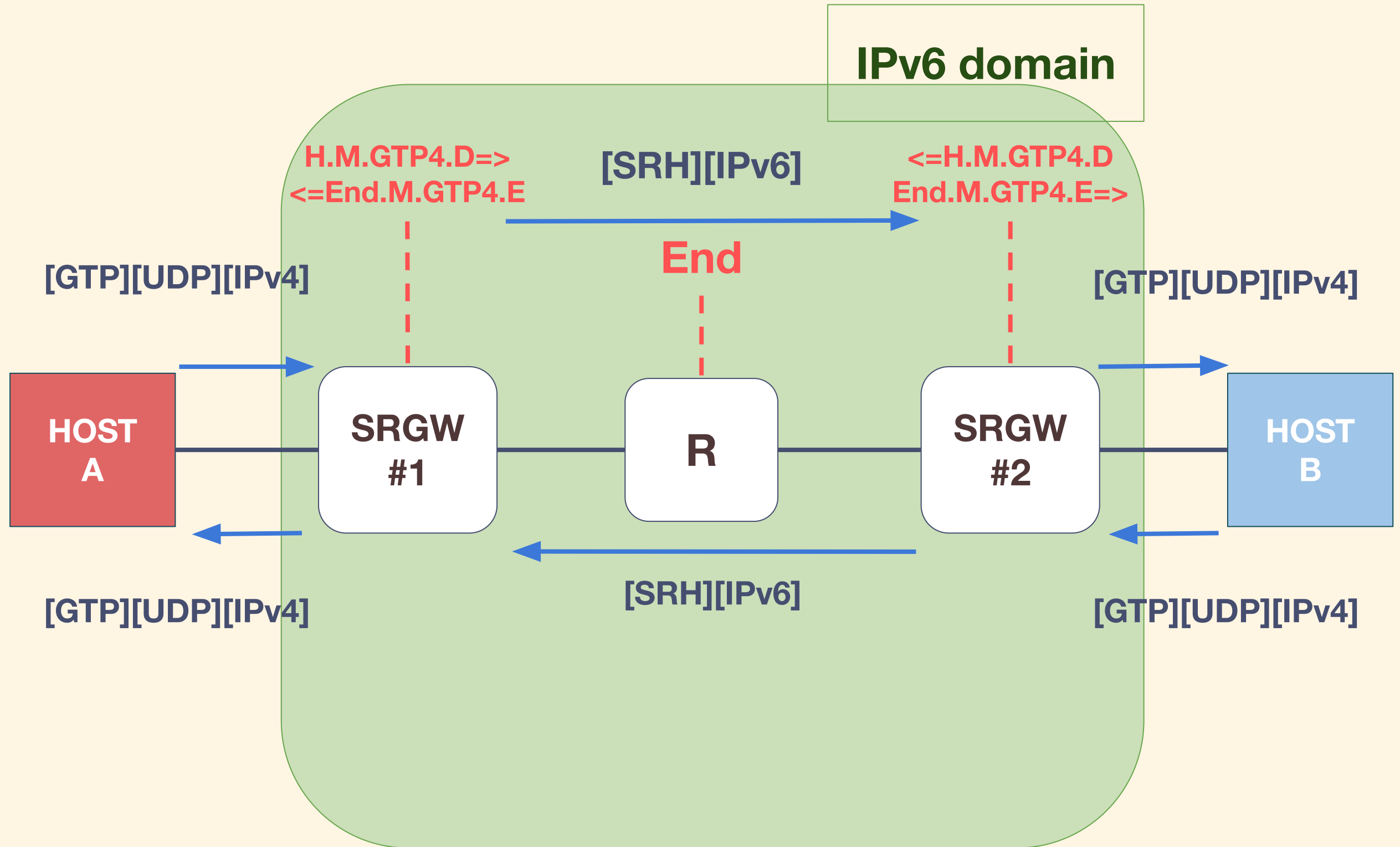




GTPv1+IPv4/SRv6

Stateless Conversion

GTP-U<=>SRv6 Conversion Scenario



H.M.GTP4.D: GTP-U/IPv4=>SRv6

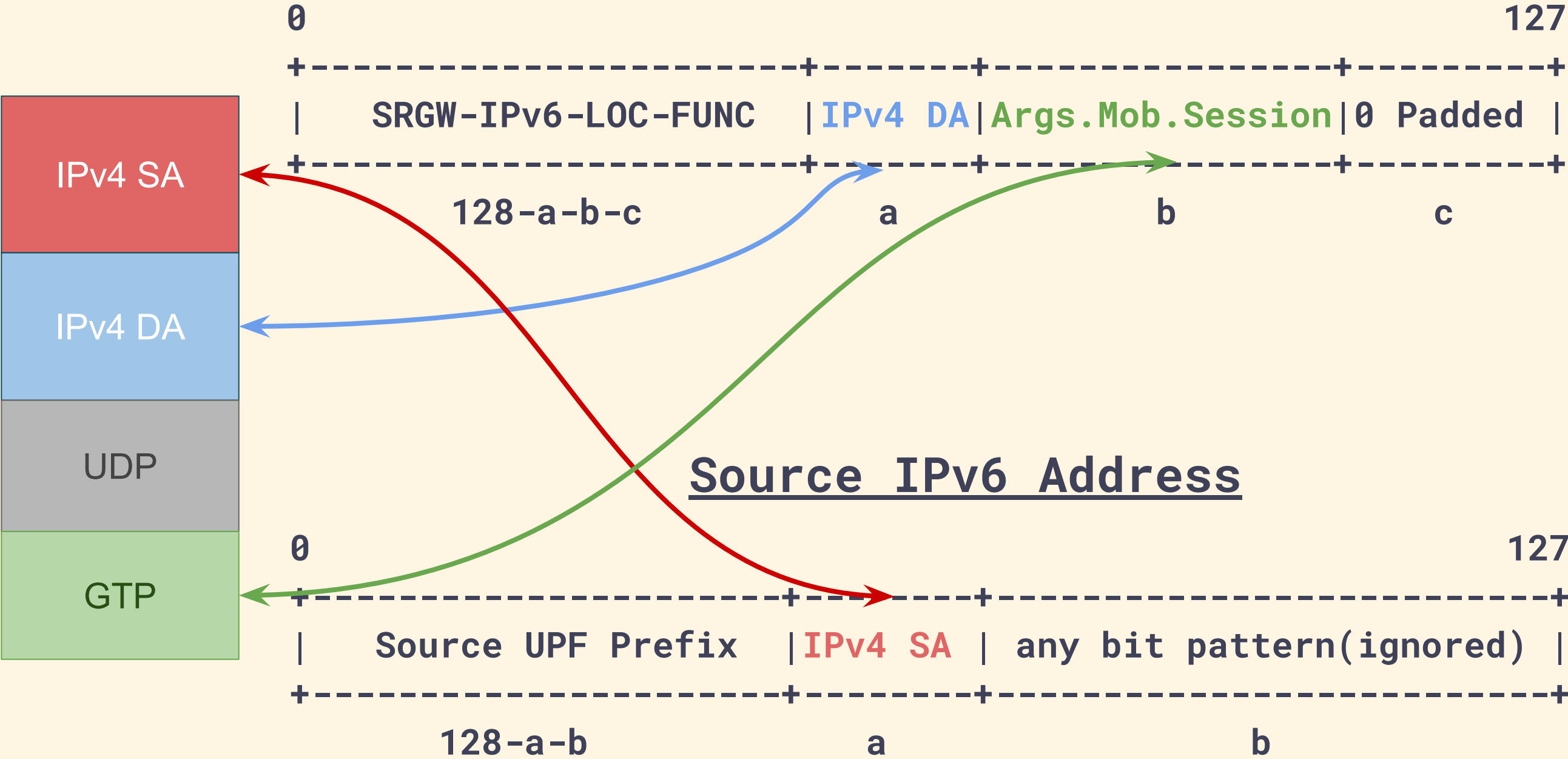
End.M.GTP4.E: SRv6=>GTP-U/IPv4

GTP IPv4 Addr ⇔ SRv6 SegmentID & Src Addr

H.M.GTP4.D=>

<=End.M.GTP4.E

Destination Segment ID
(Last Destination IPv6 Addr)



Example of a real packet

```
▶ Frame 1: 134 bytes on wire (1072 bits), 134 bytes captured (1072 bits) on interface XDPredict,
▶ Ethernet II, Src: SAKURAIIn_30:d3:28 (9c:a3:ba:30:d3:28), Dst: SAKURAIIn_30:76:a5 (9c:a3:ba:30:76:a5)
▶ Internet Protocol Version 4, Src: 172.0.2.1, Dst: 172.0.1.1
▶ User Datagram Protocol, Src Port: 2152, Dst Port: 2152
▼ GPRS Tunneling Protocol
  ▶ Flags: 0x30
    Message Type: T-PDU (0xff)
    Length: 84
    TEID: 0x000000c8 (200)
▶ Internet Protocol Version 4, Src: 10.0.1.2, Dst: 10.0.1.1
▶ Internet Control Message Protocol
```

0000	9c a3 ba 30 76 a5 9c a3 ba 30 d3 28 08 00 45 00	...0v... 0.(.E.
0010	00 78 fc 00 40 00 3f 11 e4 71 ac 00 02 01 ac 00	..x..@.?..q.....
0020	01 01 08 68 08 68 00 64 00 00 30 ff 00 54 00 00	...h..h..d..0..T..
0030	00 c8 45 00 00 54 bc c9 40 00 40 01 67 dd 0a 00	..E..T.. @.@.g...
0040	01 02 0a 00 01 01 08 00 5c 64 00 06 24 b9 ed f2	 \d..\$...
0050	f4 5f 00 00 00 00 cf b6 06 00 00 00 00 00 10 11	 _.....
0060	12 13 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21	 !
0070	22 23 24 25 26 27 28 29 2a 2b 2c 2d 2e 2f 30 31	"#\$%&'()*+,-./01
0080	32 33 34 35 36 37	234567

GTPv1 packet

```
▶ Frame 2: 178 bytes on wire (1424 bits), 178 bytes captured (1424 bits) on interface XDPredict
▶ Ethernet II, Src: SAKURAIIn_30:3a:39 (9c:a3:ba:30:3a:39), Dst: SAKURAIIn_30:bc:81 (9c:a3:ba:30:bc:81)
▼ Internet Protocol Version 6, Src: fc00:1::ac00:101:0:0, Dst: fc00:2::1
  0110 .... = Version: 6
  ▶ .... 0000 0000 .... = Traffic Class: 0x00 (DSCP: CS0, ECN: Not-ECT)
  .... 0000 0000 0000 0000 0000 0000 = Flow Label: 0x000000
  Payload Length: 124
  Next Header: Routing Header for IPv6 (43)
  Hop Limit: 63
  Source Address: fc00:1::ac00:101:0:0
  Destination Address: fc00:2::1
  ▼ Routing Header for IPv6 (Segment Routing)
    Next Header: IPIP (4)
    Length: 4
    [Length: 40 bytes]
    Type: Segment Routing (4)
    Segments Left: 1
    Last Entry: 1
    Flags: 0x00
    Tag: 0000
    Address[0]: fc00:3:0:ac00:201::6400
    Address[1]: fc00:2::1
  ▶ Internet Protocol Version 4, Src: 10.0.1.1, Dst: 10.0.1.2
  ▶ Internet Control Message Protocol
```

0000	9c a3 ba 30 bc 81 9c a3 ba 30 3a 39 86 dd 60 00	...0... 0:9..`
0010	00 00 00 7c 2b 3f fc 00 00 01 00 00 00 00 ac 00	... +?... ..
0020	01 01 00 00 00 00 fc 00 00 02 00 00 00 00 00 00	
0030	00 00 00 00 00 01 04 04 04 01 01 00 00 00 fc 00	
0040	00 03 00 00 ac 00 02 01 00 00 00 00 64 00 fc 00	d..
0050	00 02 00 00 00 00 00 00 00 00 00 00 00 01 45 00	E.
0060	00 54 5e 9f 00 00 40 01 06 08 0a 00 01 01 0a 00	..T^...@.
0070	01 02 00 00 64 64 00 06 24 b9 ed f2 f4 5f 00 00	...dd.. \$... _..
0080	00 00 cf b6 06 00 00 00 00 00 10 11 12 13 14 15	
0090	16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25	 !"#\$\$%
00a0	26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35	&'()*+,-./012345
00b0	36 37	67

SRv6 packet
(gtpv1 packet decap)

For more information (e.g. Args.Mob.Session..etc)

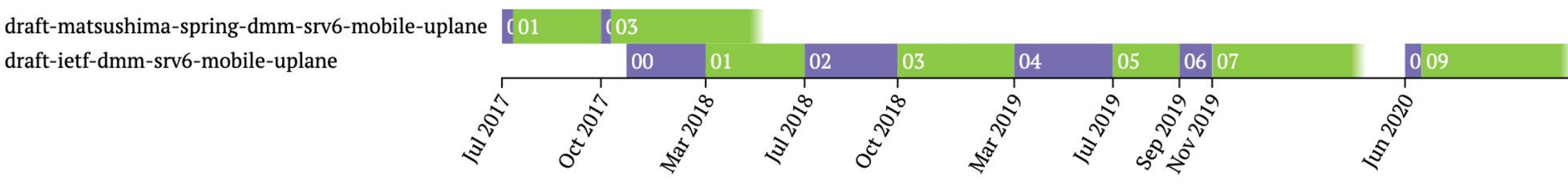
 Datatracker Groups Documents Meetings Other User

Segment Routing IPv6 for Mobile User Plane

draft-ietf-dmm-srv6-mobile-uplane-09

Status IESG evaluation record IESG writeups Email expansions History

Versions 00 01 02 03 04 05 06 07 08 09



draft-matsushima-spring-dmm-srv6-mobile-uplane (01 (03

draft-ietf-dmm-srv6-mobile-uplane 00 01 02 03 04 05 06 07 09

Jul 2017 Oct 2017 Mar 2018 Jul 2018 Oct 2018 Mar 2019 Jul 2019 Sep 2019 Nov 2019 Jun 2020

Document **Type** Expired Internet-Draft ([dmm WG](#))

Authors [Satoru Matsushima](#) ✉, [Clarence Filsfils](#) ✉, [Miya Kohno](#) ✉, [Pablo Camarillo](#) ✉, [Daniel Voyer](#) ✉, [Charles Perkins](#) ✉

Last updated 2021-01-14 (latest revision 2020-07-13)

Replaces [draft-matsushima-spring-dmm-srv6-mobile-uplane](#)

Stream IETF

Intended RFC status (None)

Formats [pdf](#) [htmlized \(tools\)](#) [htmlized](#) [bibtex](#)

Expired & archived

<https://datatracker.ietf.org/doc/draft-ietf-dmm-srv6-mobile-uplane/>



Go+XDP

System Design

What is Vinbero(ビベロ)

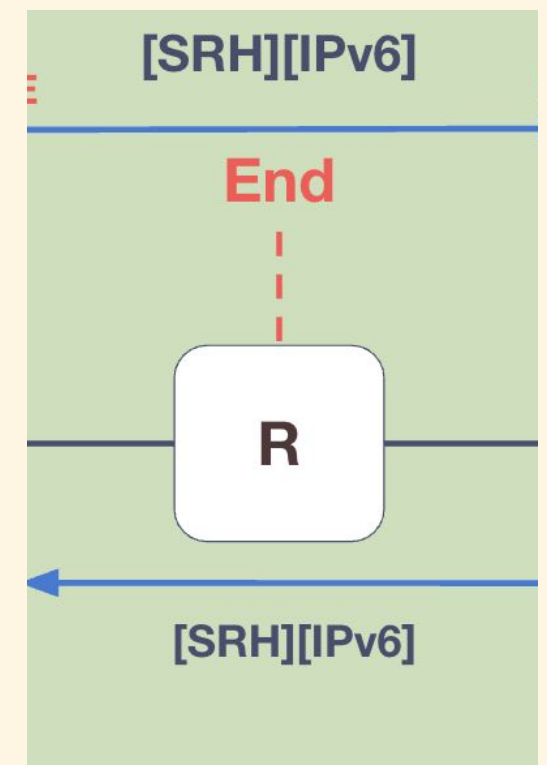
<https://github.com/takehaya/Vinbero>

```
internal:
  devices:
    - eth1
    - eth2
  settings:
    functions:
      - action: SEG6_LOCAL_ACTION_END
        triggerAddr: fc00:2::1/128
        actionSrcAddr: fc00:1::1
        flavor: NONE
      - action: SEG6_LOCAL_ACTION_END
        triggerAddr: fc00:2::2/128
        actionSrcAddr: fc00:1::1
        flavor: NONE
```

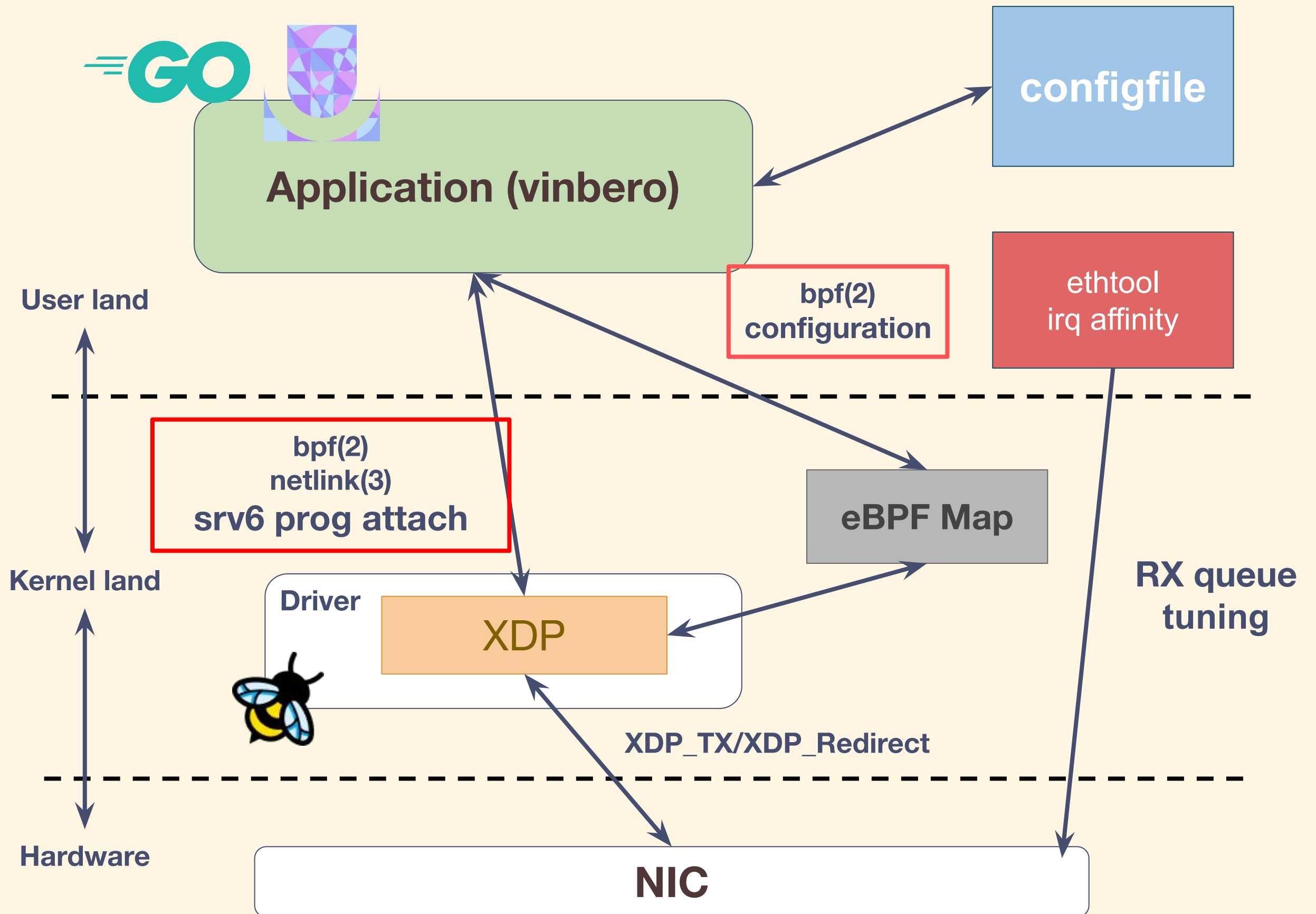
- Subset of SRv6 Dplane
- It's written in Go+XDP
- 今回の成果を動かします:)
- <= this example End function case
- Vinbero is esperanto for grape 🍇
- graph,graph,graph....,grape.... is joke:)
- and The grape looks like a mesh.



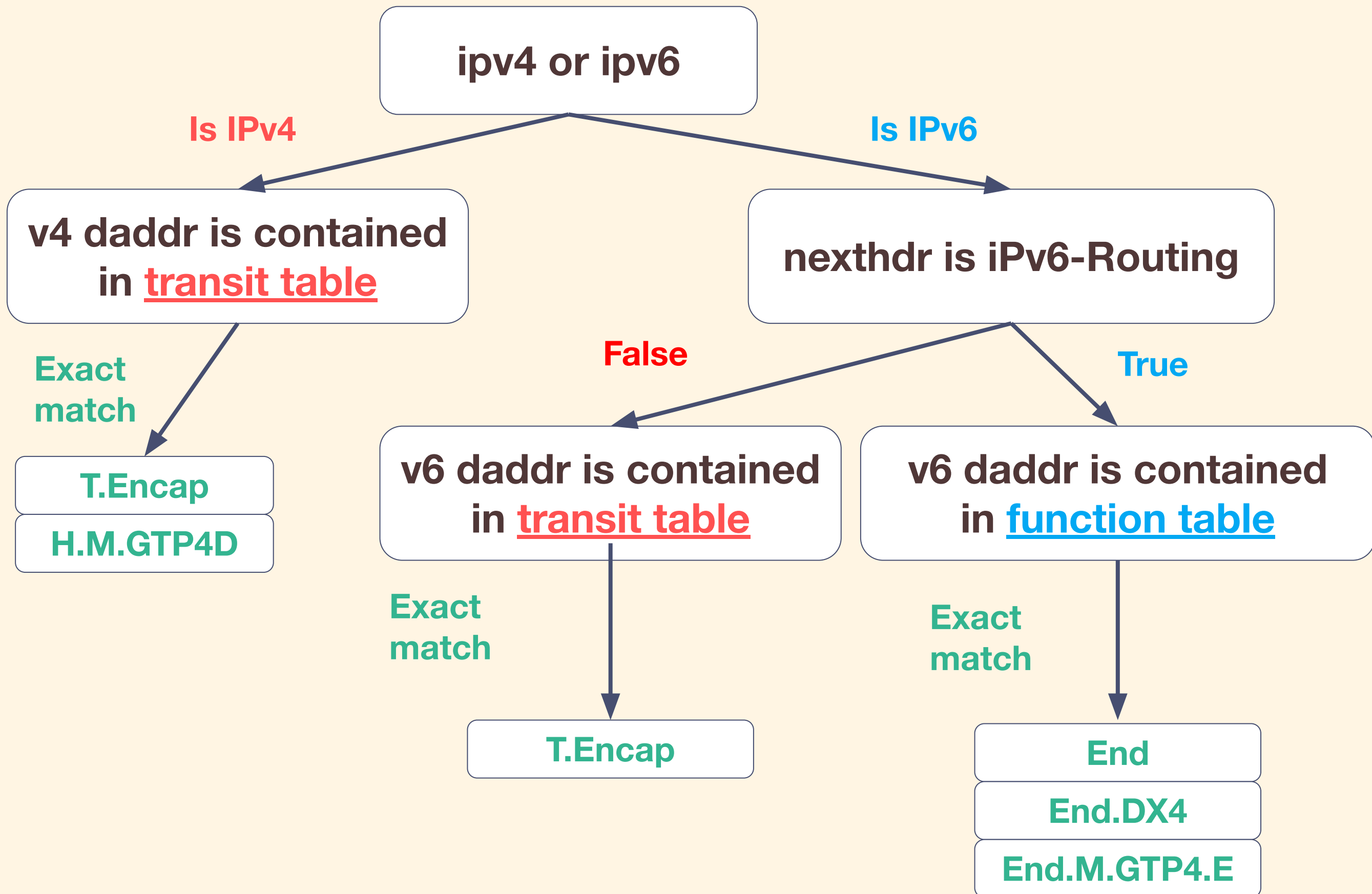
<= 🍷 image logo.



High-level Architecture



Parsing Graph



Measurement configuration

Traffic gen:
Corei7-5960X@3.0GH

z

8 core 16th

SRv6 DUT

(Device Under Test):

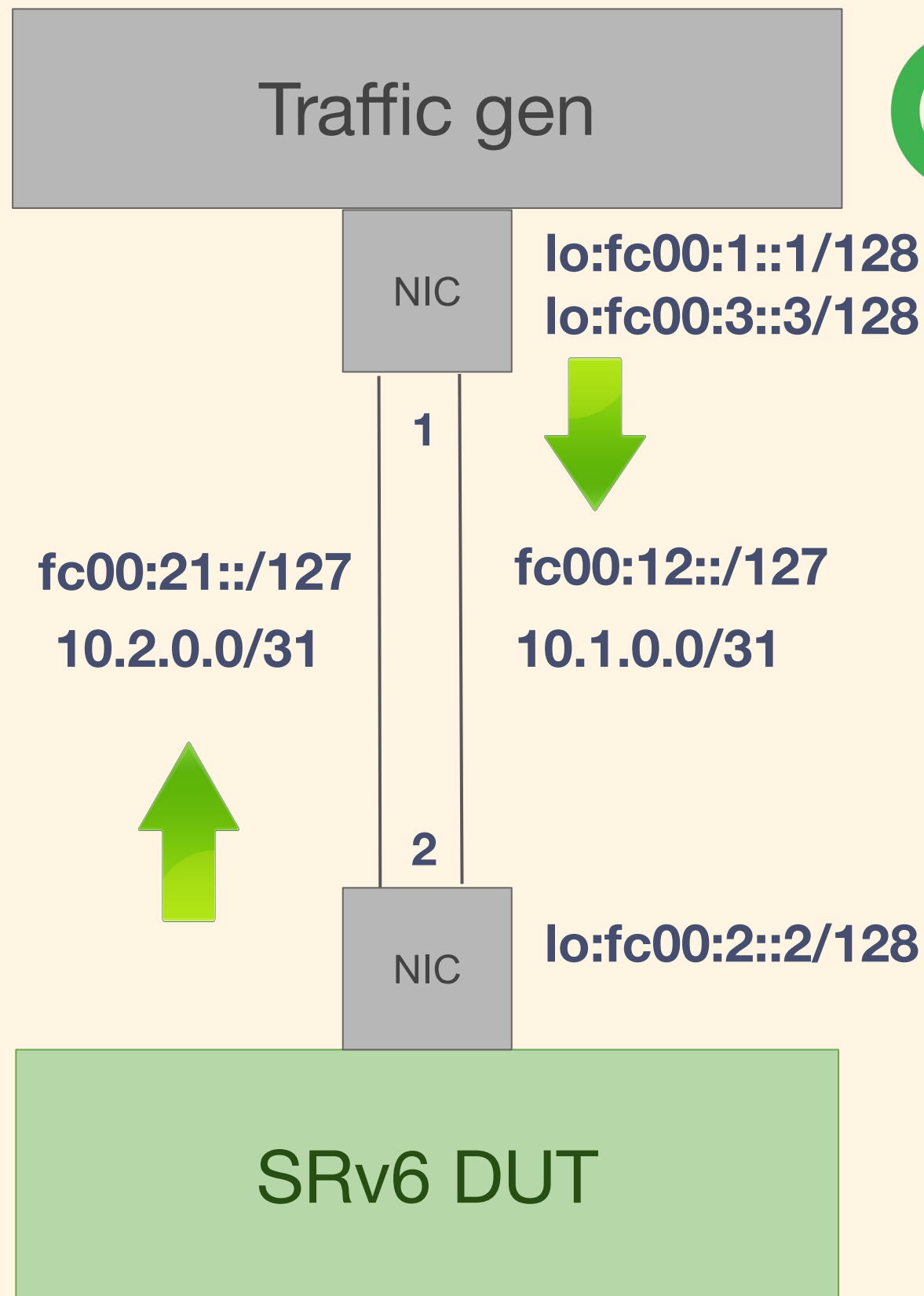
Corei9-10940X@

3.30GHz

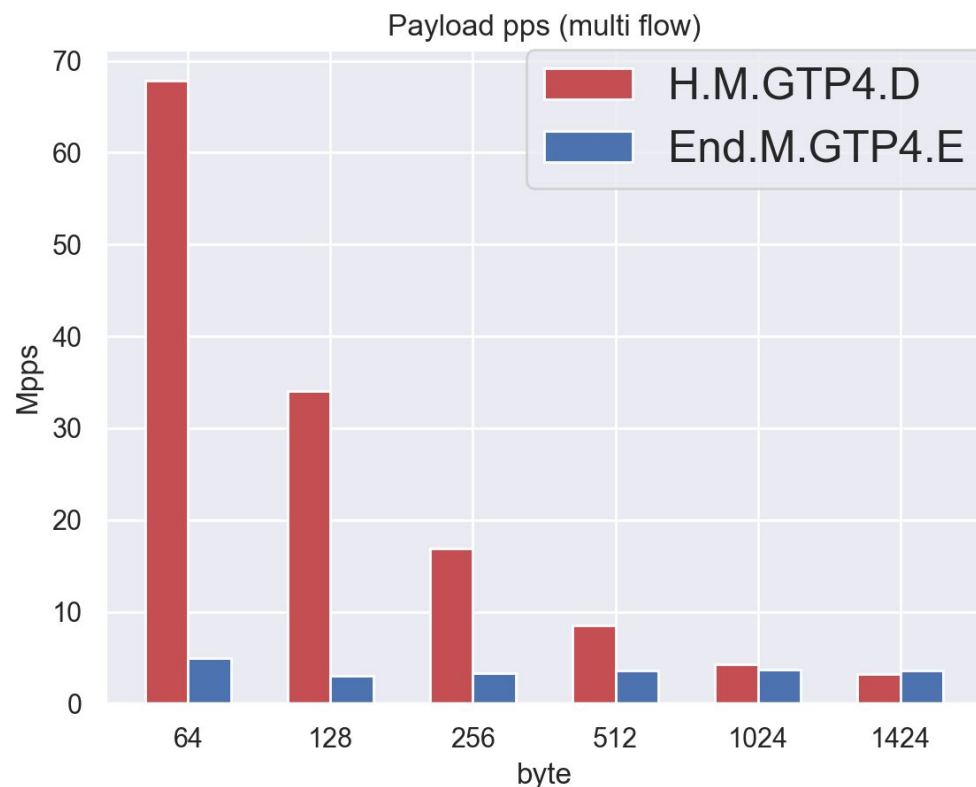
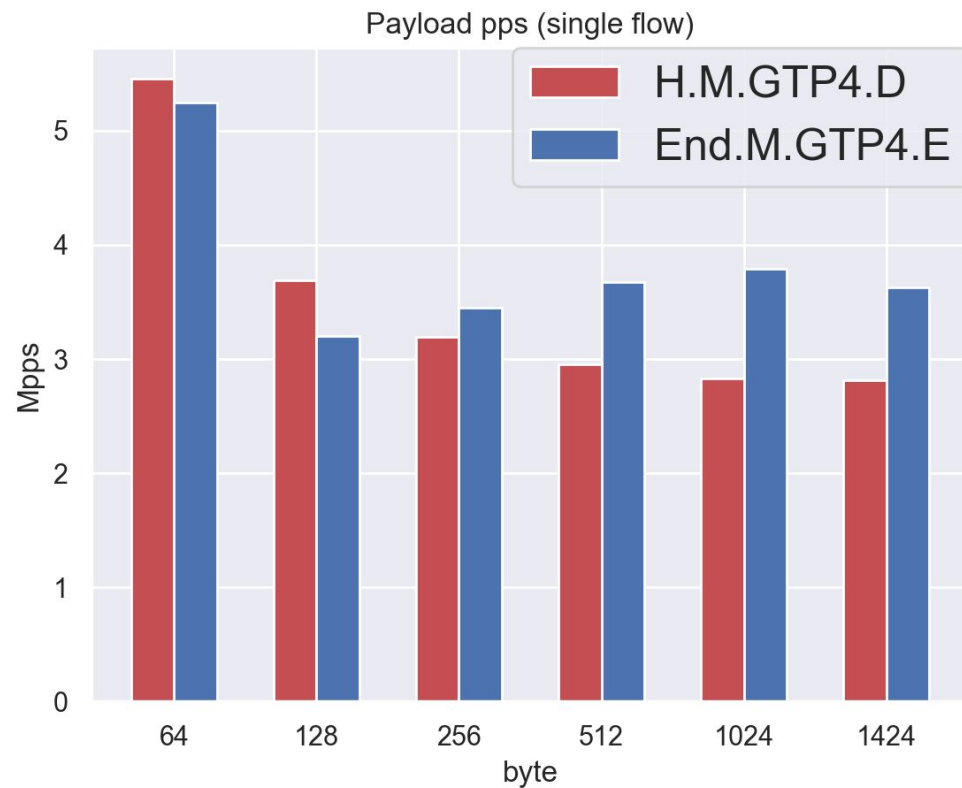
12 core 24th

NIC: ConnectX-5

40GbE uplink



Measurement Result



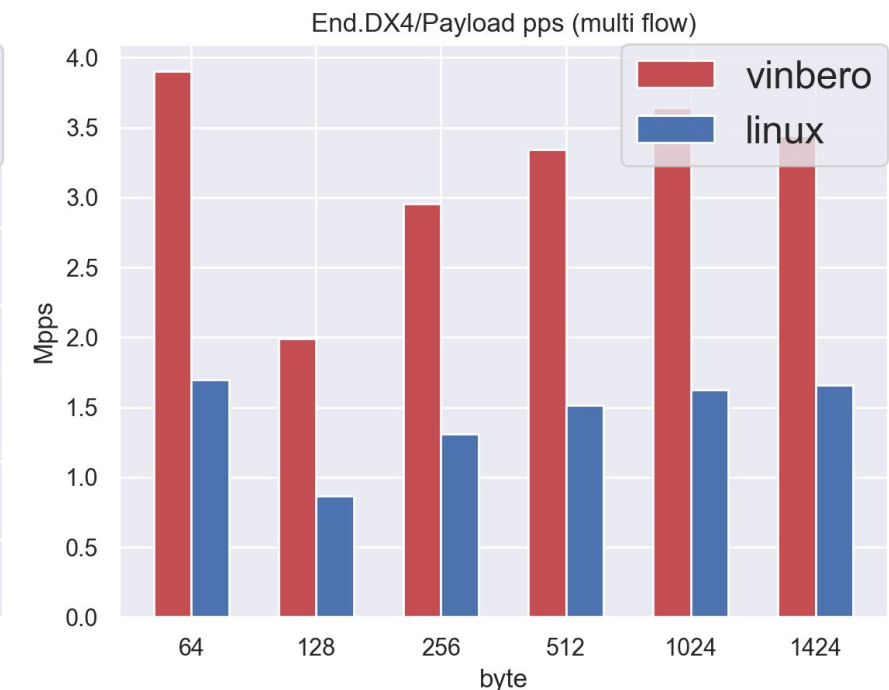
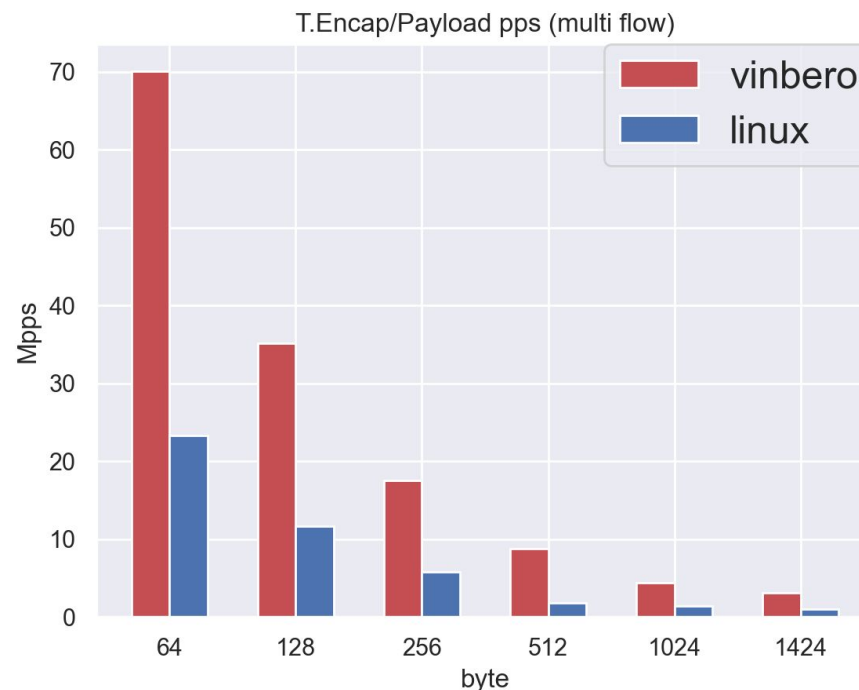
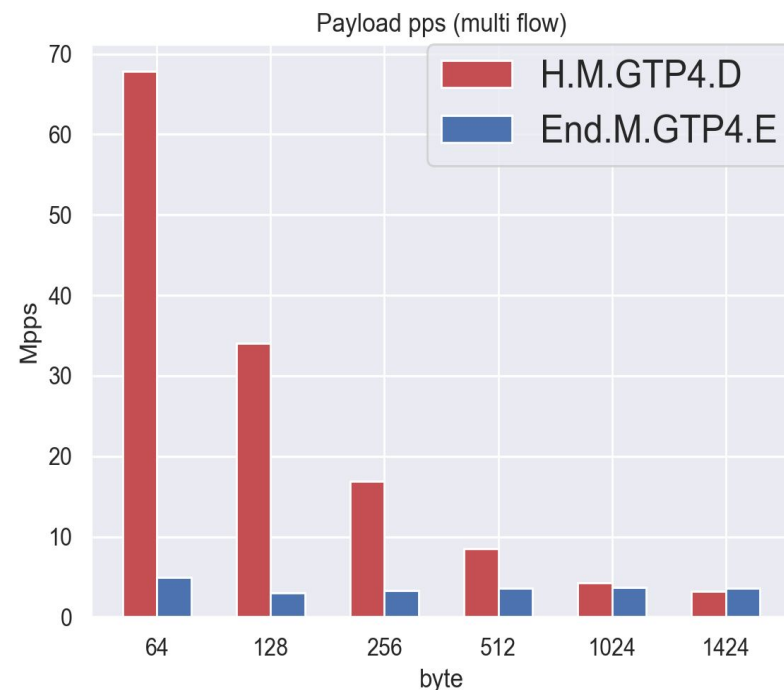
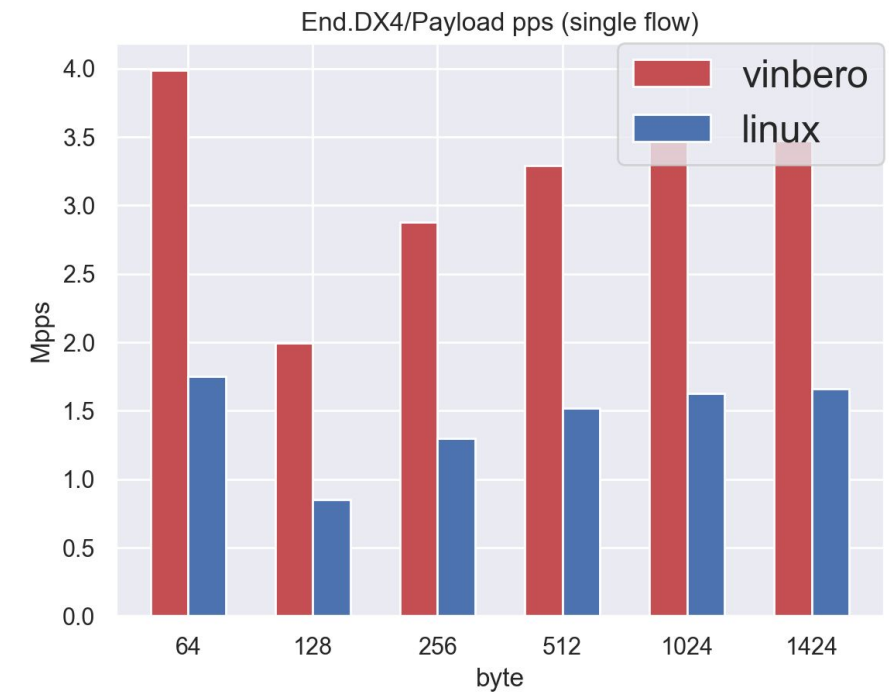
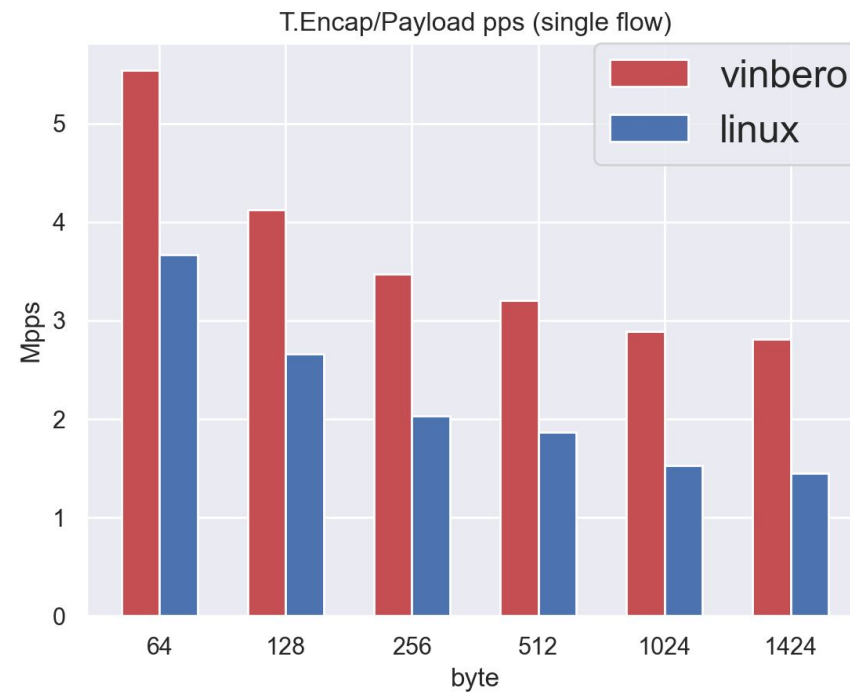
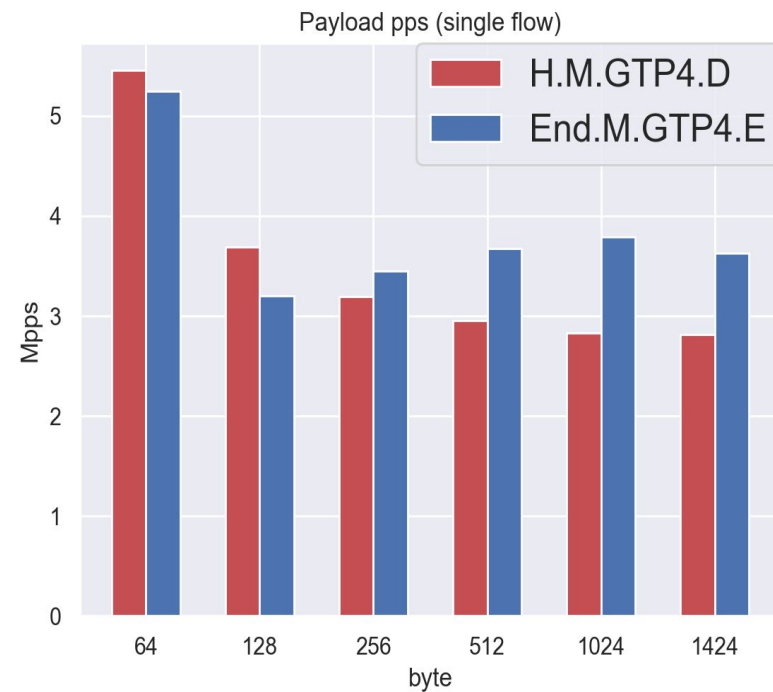
- **single flow**は大体同じ。
 - 大体**3~5Mpps**ぐらい出る
 - shortはgtp decapの性能が高いが、longになっていくとgtp encapの方が性能が良いのが興味深い
- **multi flow** はひどい 😊
 - H.M.GTP4.Dはshortで**67Mpps**
 - End.M.GTP4.Eはshortで**4.9Mpps**
 - CPUリソースを見ると分散されてなかったなのでSRv6に対してRSSが効いてないように見える。
恐らくRSSが効けば解決すると思う
- RSS効いてる側のRXを見ると飽和してたので100GbE NICとかならまだ伸びそう。

Reference Value

mobile v4 funcs

T.Encap

End.DX4



Vinberoにある近い実装の関数をlinuxと比較するとlinuxより2~3倍ほど性能が良いのでkernelに実装するよりも高速であると予想される

まとめと感想

- 皆さんはmobileで必要なSRv6を完全に理解した
- SRv6に対してのRSSが効いてないのでencap以外はマルチコア性能が低い。この辺詳しい人がいたら教えて欲しい
- 今回の成果をもとにvinberoというXDP製SRv6を公開した
<https://github.com/takehaya/Vinbero>
StarとかIssueとかPRとかお待ちしております！！！！！！
- Go+XDPに関する話は別途フォローアップしたのでこっちを参照
<https://takeio.hatenablog.com/entry/2021/01/26/180129>
- ~~説明に時間が足りないので端折りまくった。こういう実装系の説明は流石に20-30分ないと厳しいので本会議目指したい...~~