eBPF for TSN and DetNet

Ferenc Fejes,

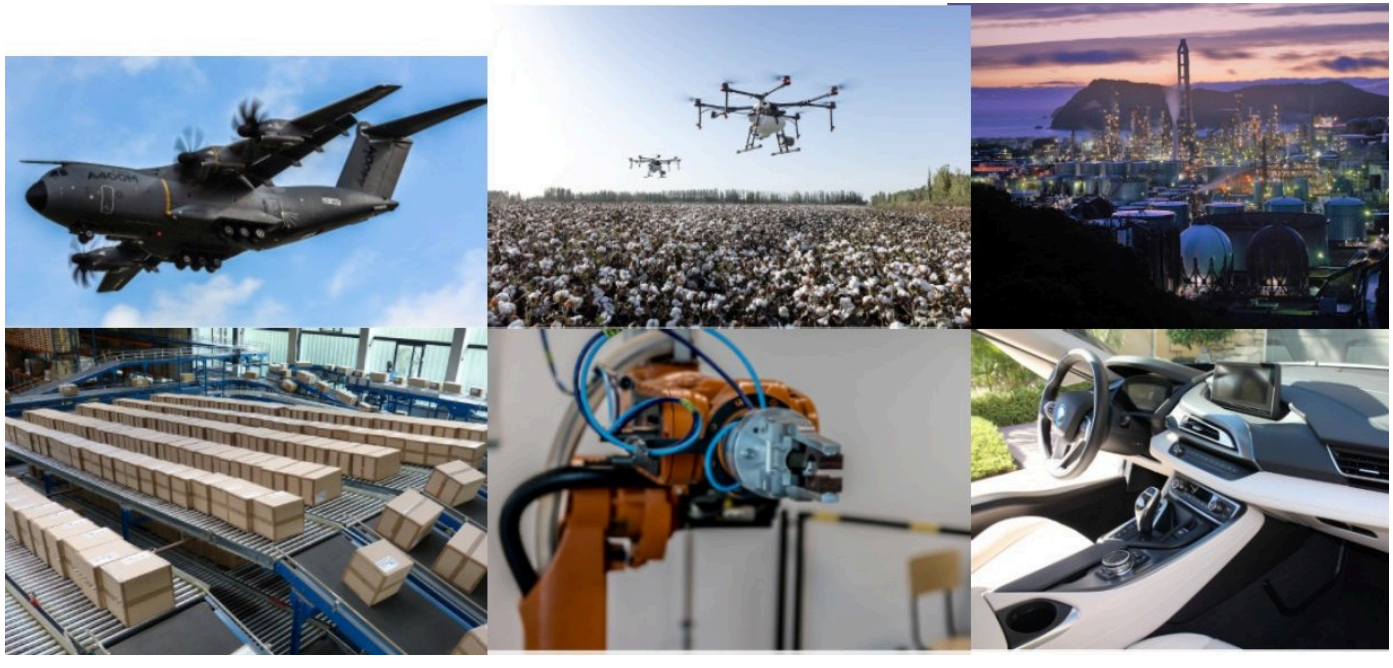
Balázs Varga, János Farkas, Ferenc Orosi

Ericsson Research

2025-07-26 IETF DetNet – IEEE 802.1 TSN Workshop

ERICSSON DETERMINISTIC6G

Mission- and Time-Critical Use Cases



source: pixabay.com

What is BPF?

- BPF (BSD Packet Filter)
- Introduced by Steven McCanne and Van Jacobson in 1992
- Simple instruction set designed for packet filtering
- Interpreter (VM) inside the kernel for it
- Popular users: tcpdump, libpcap, Wireshark

What is eBPF?

- Extends BPF instruction set with new operations
- Implements a verifier to ensure safe memory access, prevent halting problem
- eBPF instruction set & architecture (ISA) is IETF standard: RFC 9669
- Equip the monolithic kernel with runtime programmability
- **Many use cases beyond filtering e.g.:** tracing, packet processing, task scheduling, etc.

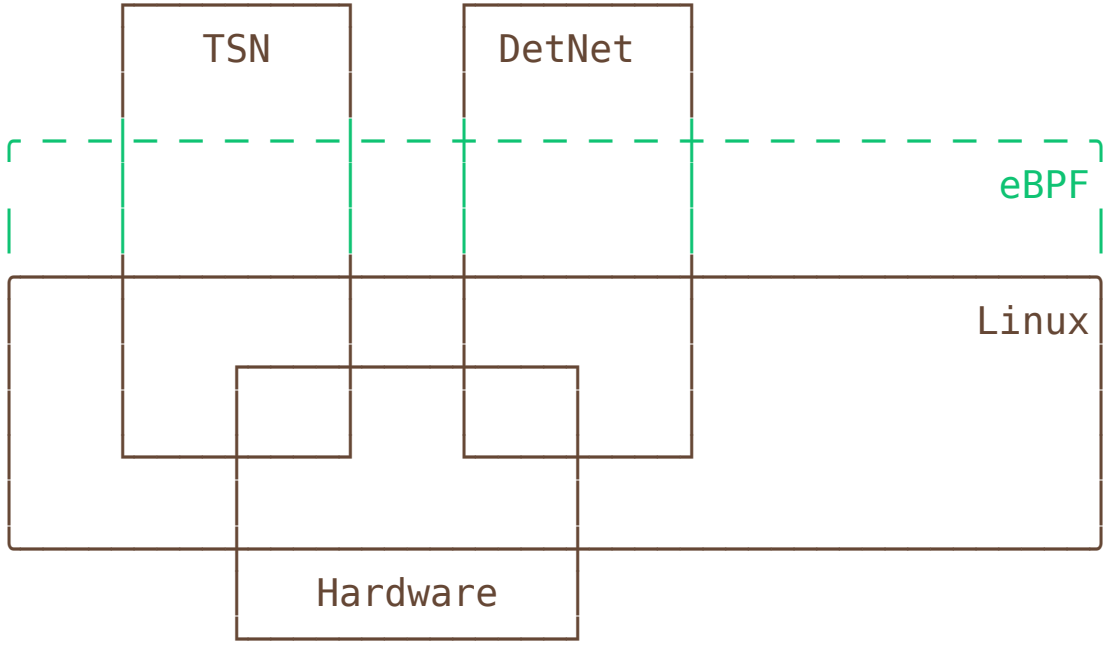
Linux and TSN/DetNet

- Linux with real-time preemption is strong platform for time-sensitive applications (TSA)
- IEEE 802.1 TSN standards (e.g.: Qbv, Qci, Qch, timesync) implemented in the kernel
- eBPF enables rapid iteration, research and experimentation with networking
- TSN/DetNet support can be improved with the help of eBPF

TSN/DetNet Use Cases with eBPF

- XDPFRER: IEEE 802.1CB implementation
- TSN/DetNet proxy: enhanced TSA support for Kubernetes
- Improved observability
  - Layer 2 TSN stream OAM (based on CFM)
  - Routing table lookup tracing

Overview

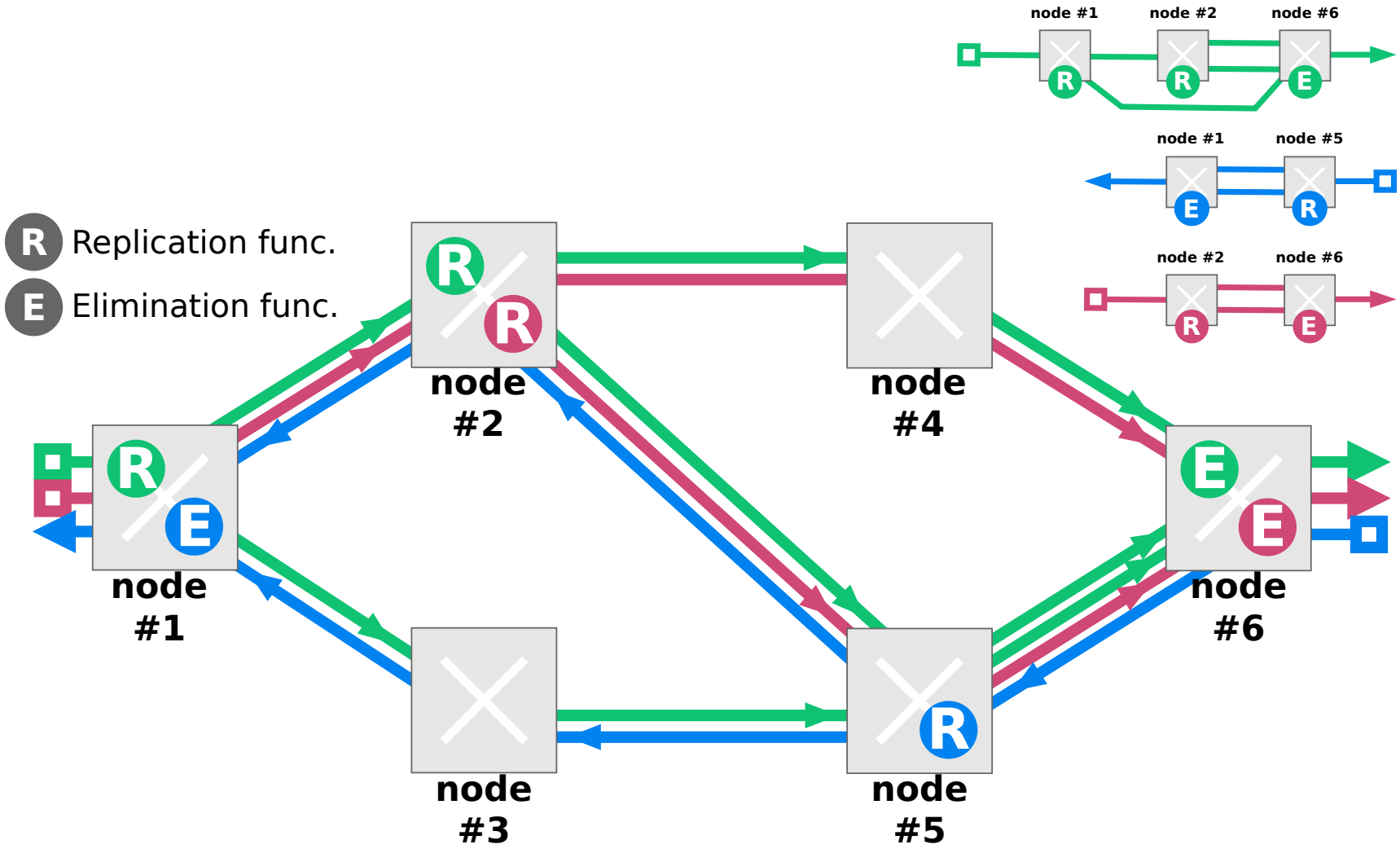


XDPFRER

IEEE 802.1CB

Frame Replication & Elimination for Reliability (FRER) with eBPF

ERICSSON DETERMINISTIC6G


node #1
node #2
node #6
node #1
node #5
node #2
node #6
R Replication func.
E Elimination func.
node #1
node #2
node #3
node #4
node #5
node #6

Upstream Linux FRER Attempts

First attempt from NXP

```
From: Xiaoliang Yang <xiaoliang.yang_1@nxp.com>
Date: Tue, 28 Sep 2021 19:44:51

[RFC, net-next] net: qos: introduce a frer action to implement 802.1CB
```

Second attempt from Cruise LLC

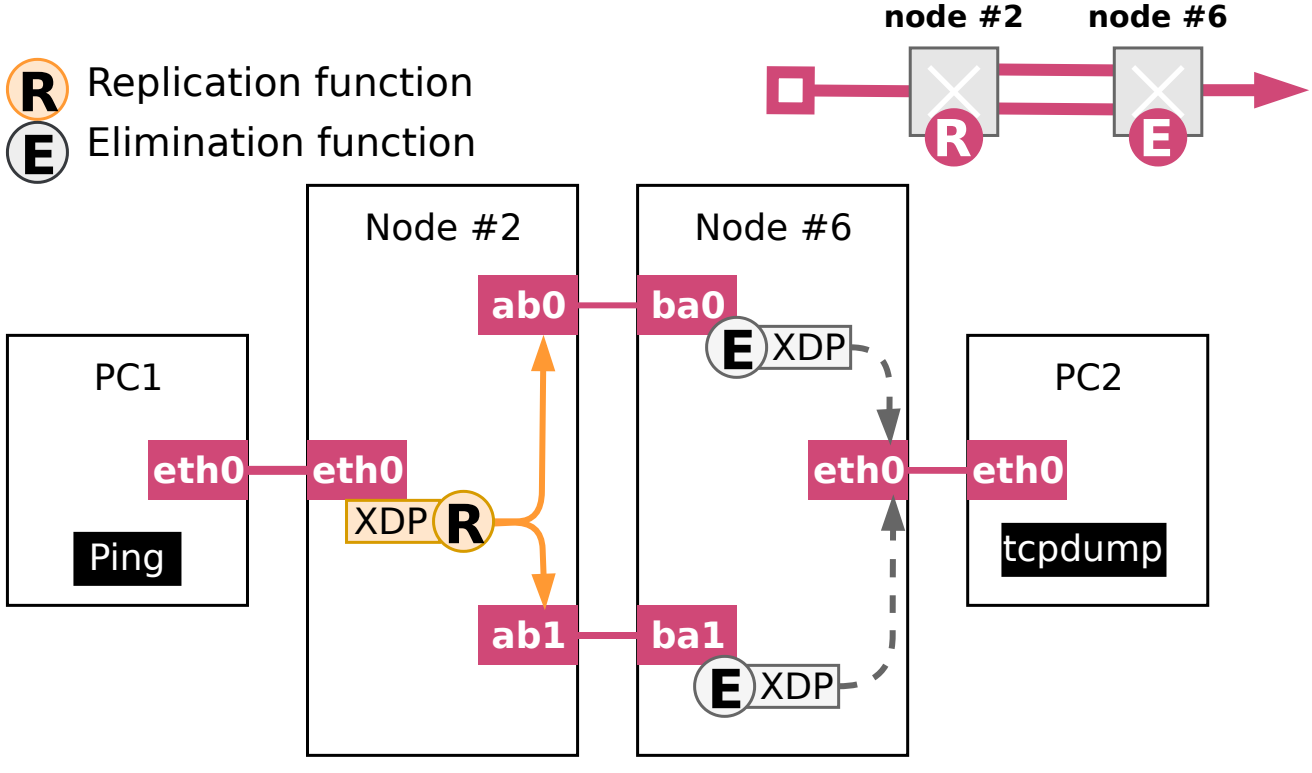
```
From: Steve Williams <steve.williams@getcruise.com>
Date: Fri, 18 Nov 2022 15:26:39

[PATCH net-next] net/hanic: Add the hanic network interface for high
availability links
```

ERICSSON DETERMINISTIC6G

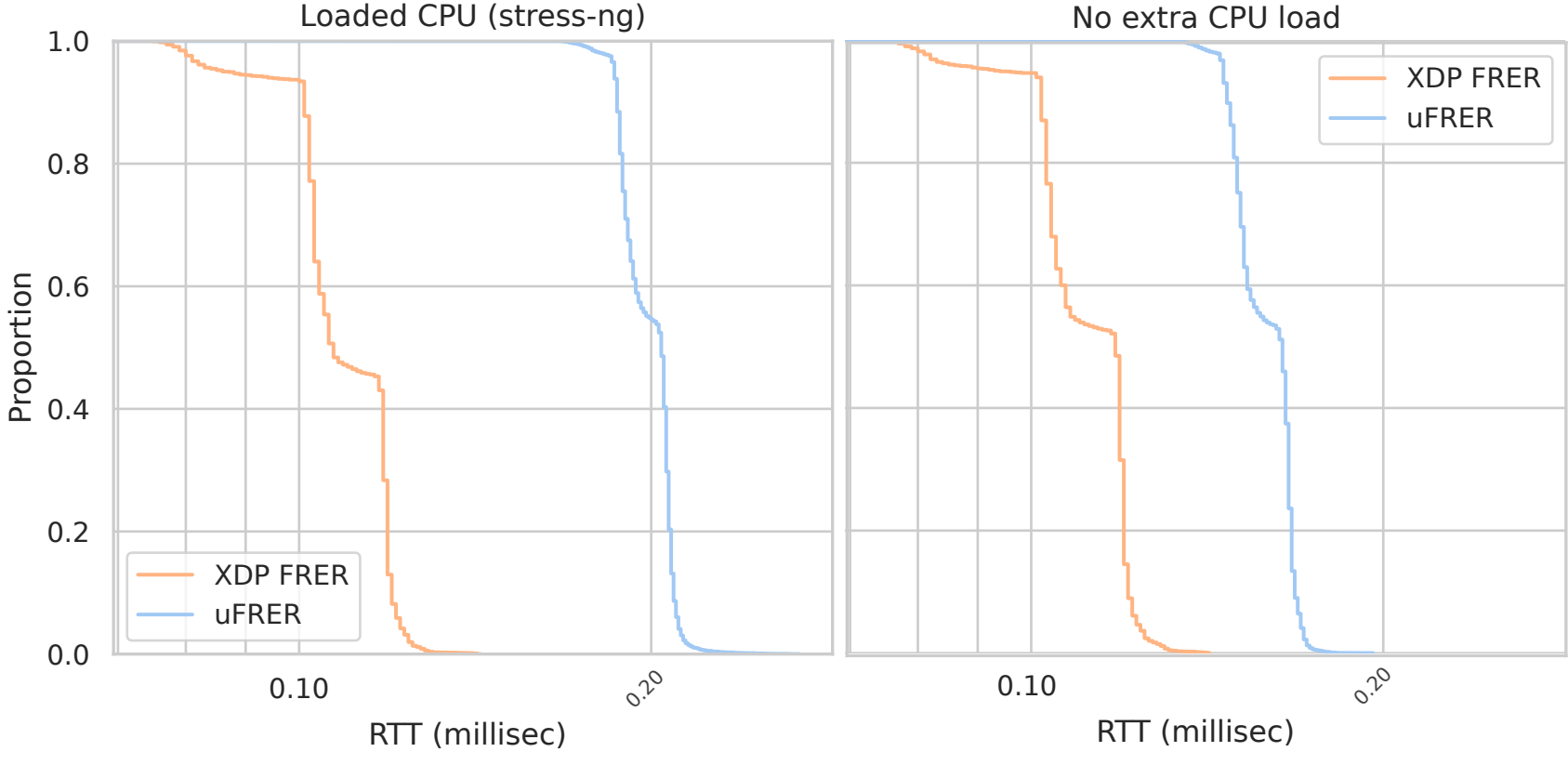
Alternative Approach: eBPF and XDP

- FRER functions can be implemented with the XDP packet-processing APIs
  - R-tag push/pop with head adjustment support
  - Replication with broadcast transmission flag
  - Elimination enabled by BPF locking API: serialized access to internal state (history window, counters)

Example Testbed



XDPFRER vs User-Space FRER with CPU Load



TSN/DetNet metadata proxy

Context

- Microservices: main direction of application deployment since the past decade
- This trend emerged in the field of industrial applications as well
- *Kubernetes* has become the de-facto orchestration platform
- There are challenges however…

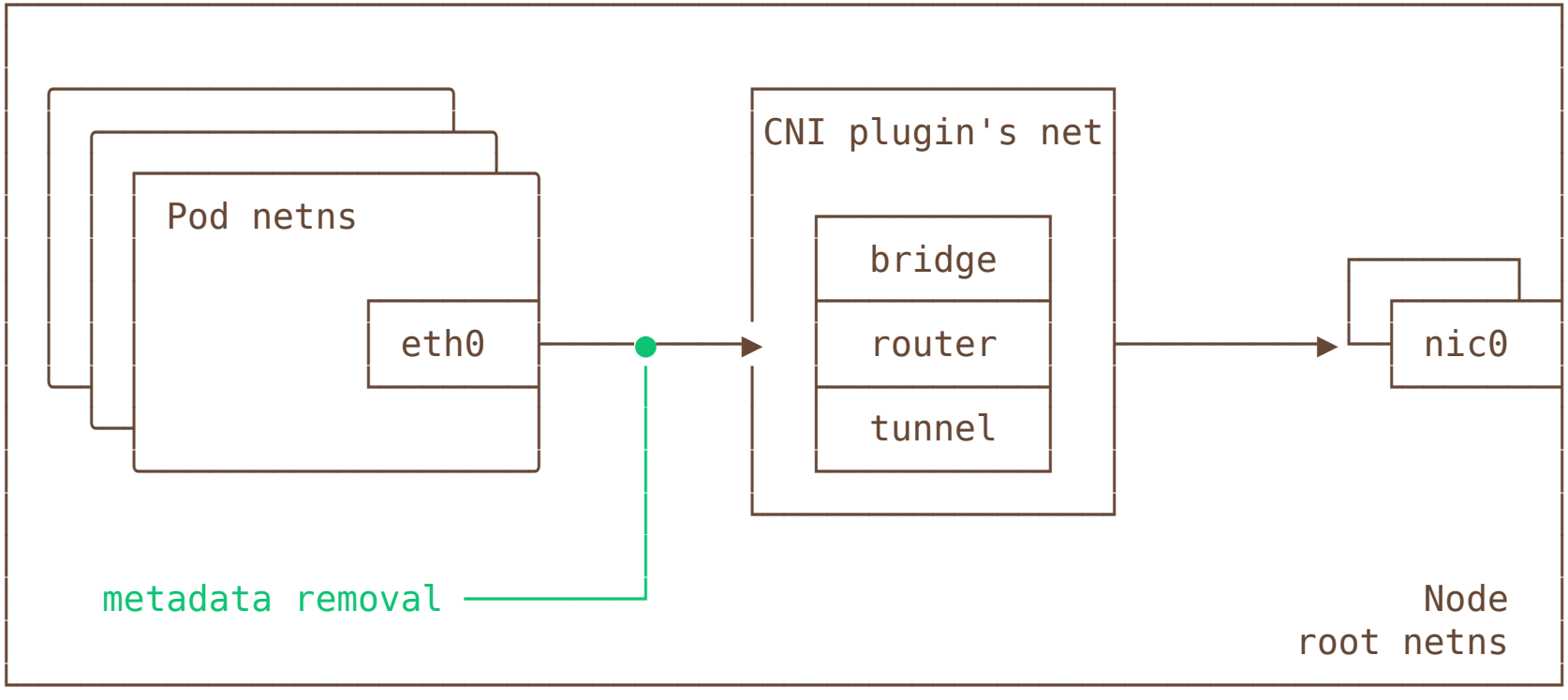
Kubernetes Networking

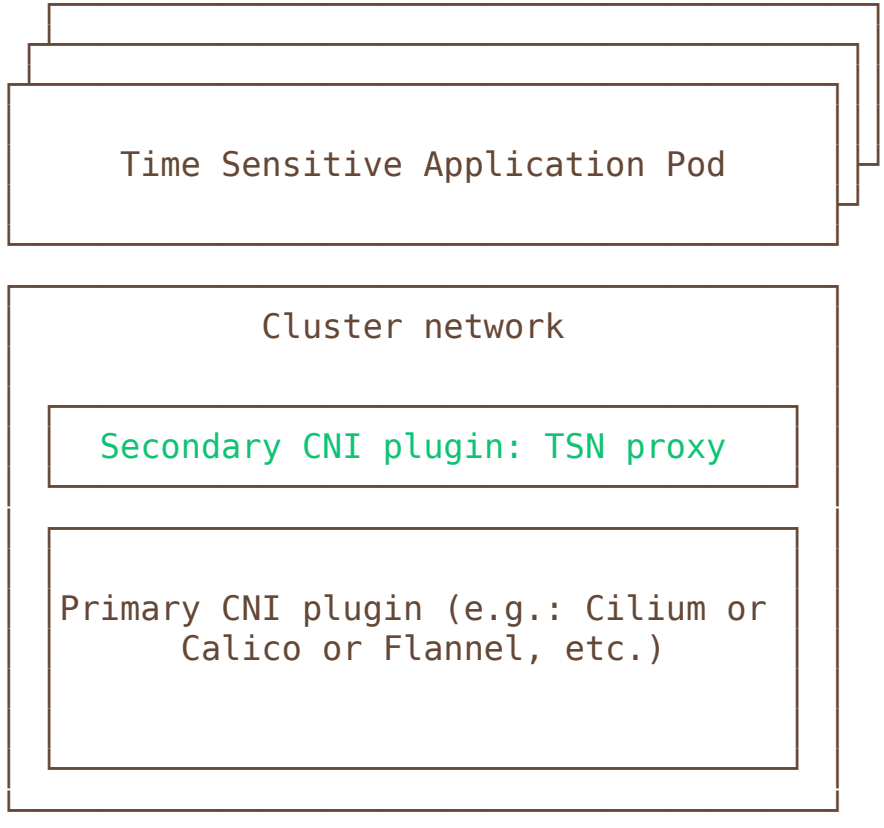
- Networking implemented by *Container Network Interface* (CNI) plugins
  - Popular CNI plugins: Calico, Cilium, Flannel, Weave Net, Kindnet, Canal
- Each implement the cluster network differently
- Most of them rely on GNU/Linux tools and APIs to do it

Kubernetes and Time-Sensitive Applications (TSA)

- TSA need `timestamp` and `priority` metadata to be preserved
- They cleared by Linux network namespace isolation
- CNI plugin diversity: different strategies to implement the network
- Must be transparent to the applications, plugin specific network APIs should be avoided

ERICSSON DETERMINISTIC6G

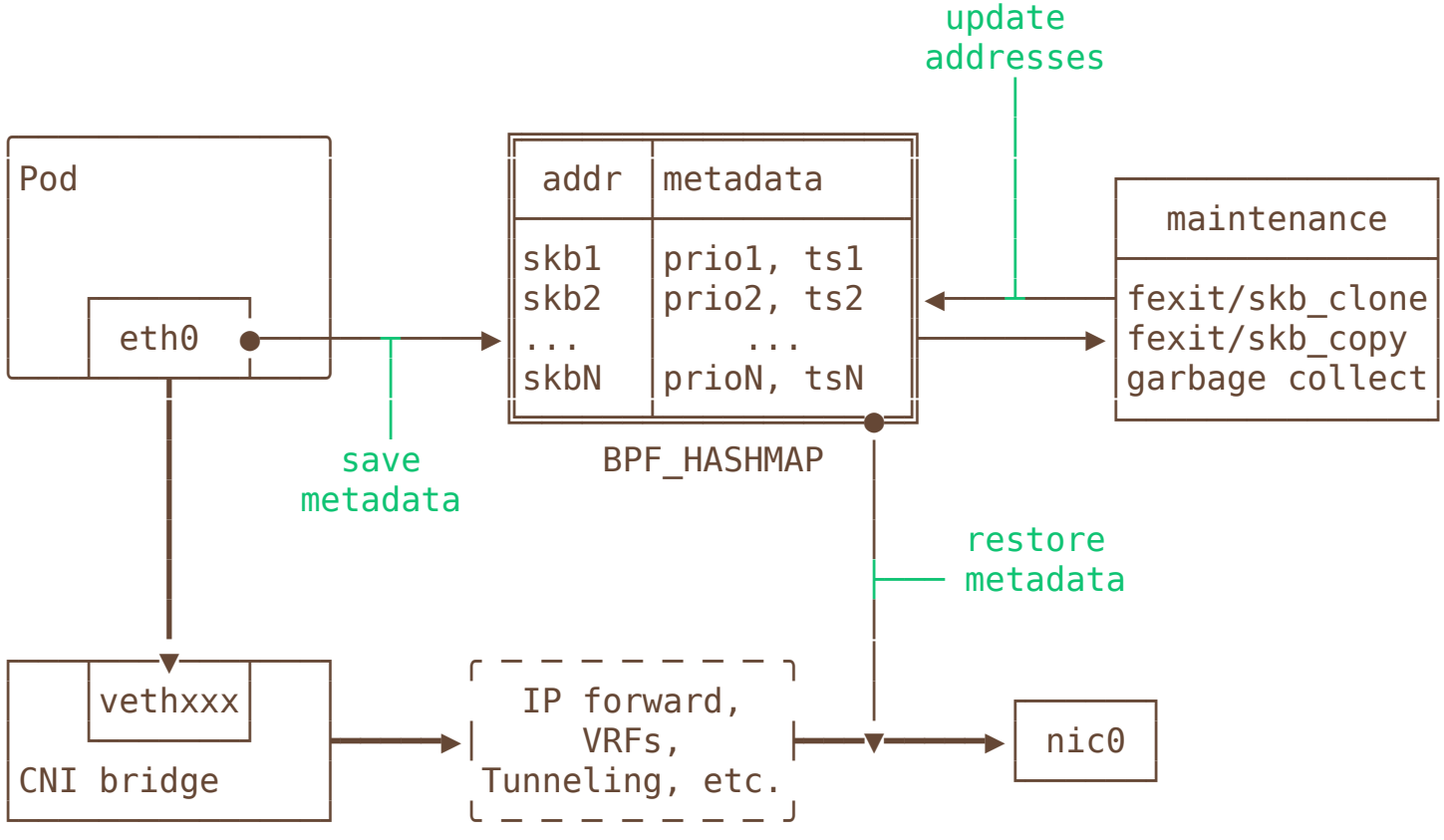

Pod netns
eth0
CNI plugin's net
bridge
router
tunnel
nic0
metadata removal
Node
root netns


Time Sensitive Application Pod
Cluster network
Secondary CNI plugin: TSN proxy
Primary CNI plugin (e.g.: Cilium or Calico or Flannel, etc.)

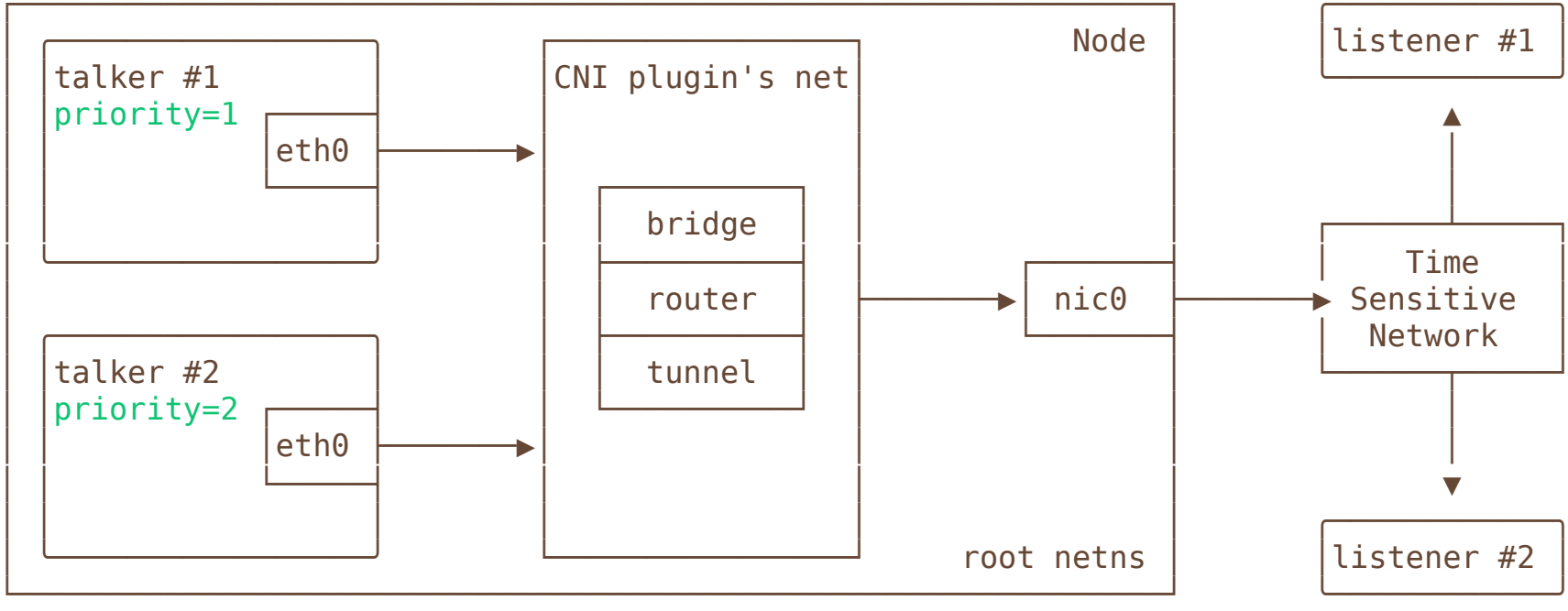
TSN Proxy CNI Plugin

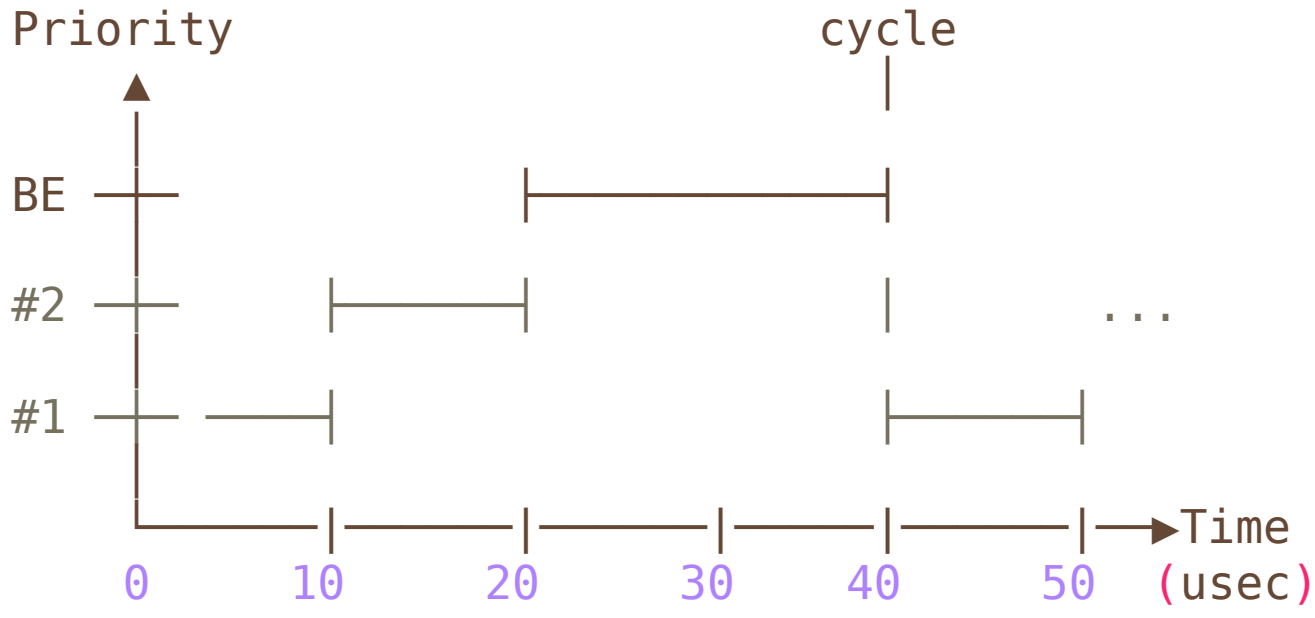
- Equip primary CNI plugins with TSN capability
- Does not require TSA microservice modification, customized container images
- No Linux kernel bypass required
- Does not require custom kernel modules, eBPF provide every tool the TSA need

TSN Proxy Operation

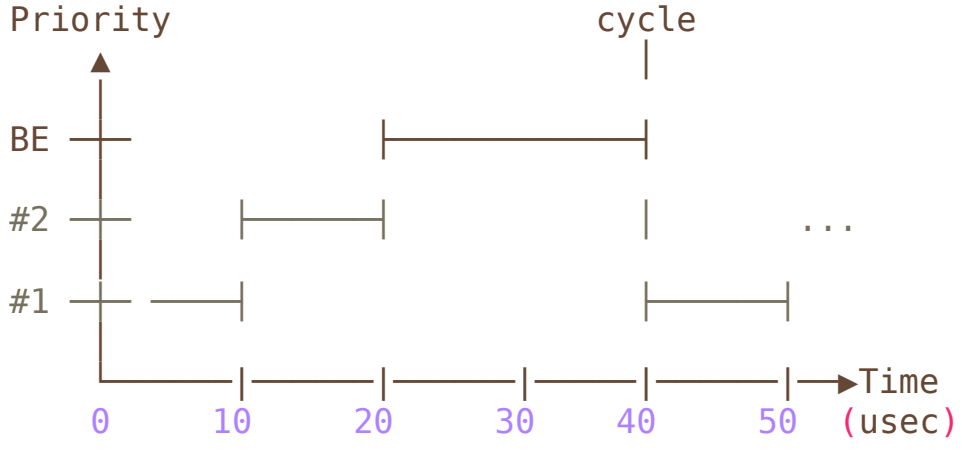


Example Scenario



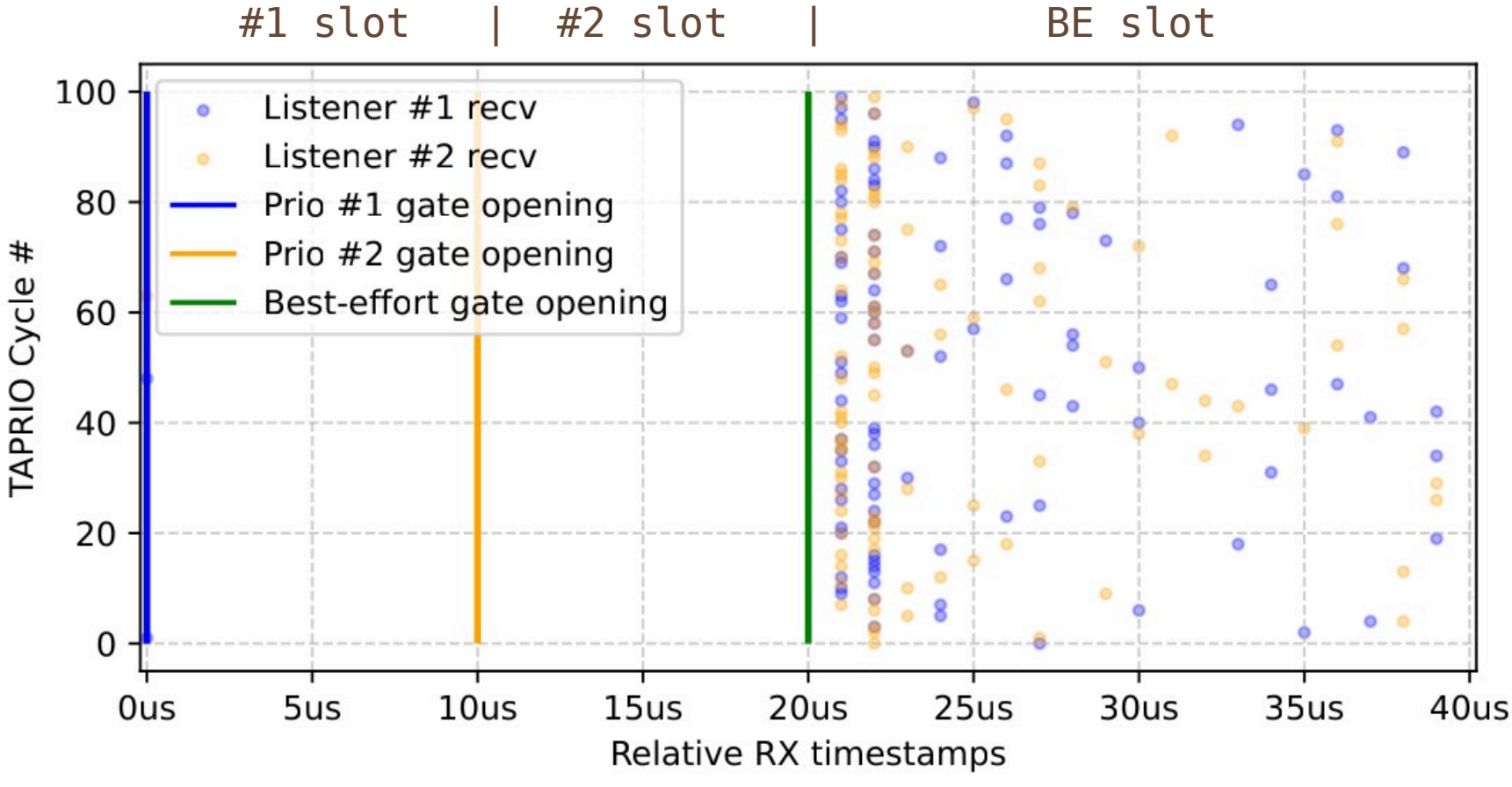


- `nic0` configured with `taprio` Qdisc (802.1Qbv)
  - 1st slot: priority=1 packets, length is 10 usec
  - 2nd slot: priority=2 packets, length is 10 usec
  - 3rd slot: best effort, length is 20 usec




```
tc qdisc add dev tsn0 parent root handle 100 taprio \
  num_tc 4 \
  map 0 1 2 3 \
  queues 1@0 1@1 1@2 1@3 \
  base-time 1693996801300000000 \
  sched-entry S 03 10000 \
  sched-entry S 05 10000 \
  sched-entry S 01 20000 \
  clockid CLOCK_TAI
```

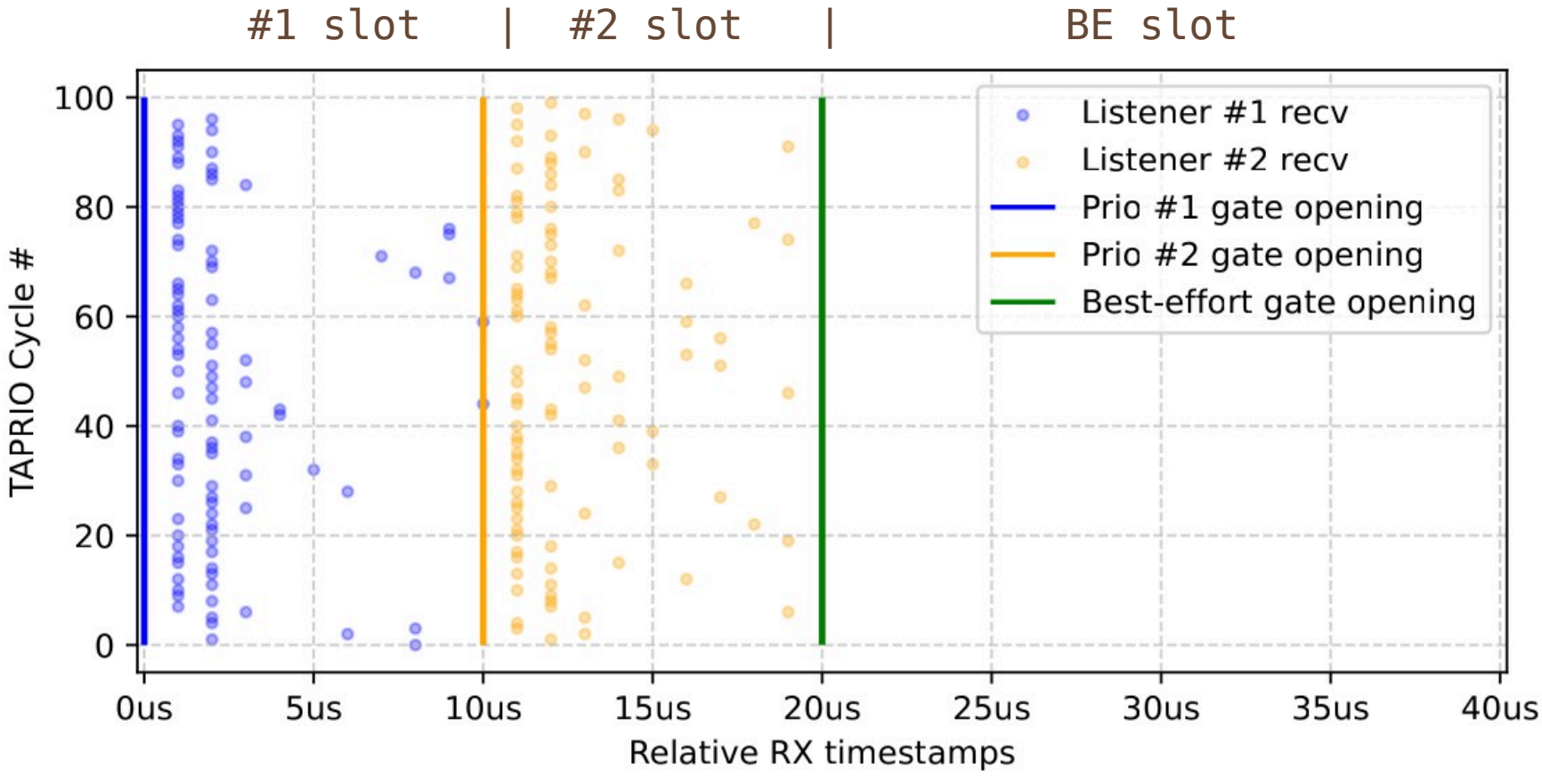
Without TSN Proxy



Packets missing their timeslots: skb->priority == 0

ERICSSON DETERMINISTIC6G

With TSN Proxy



Packets scheduled according to their priority

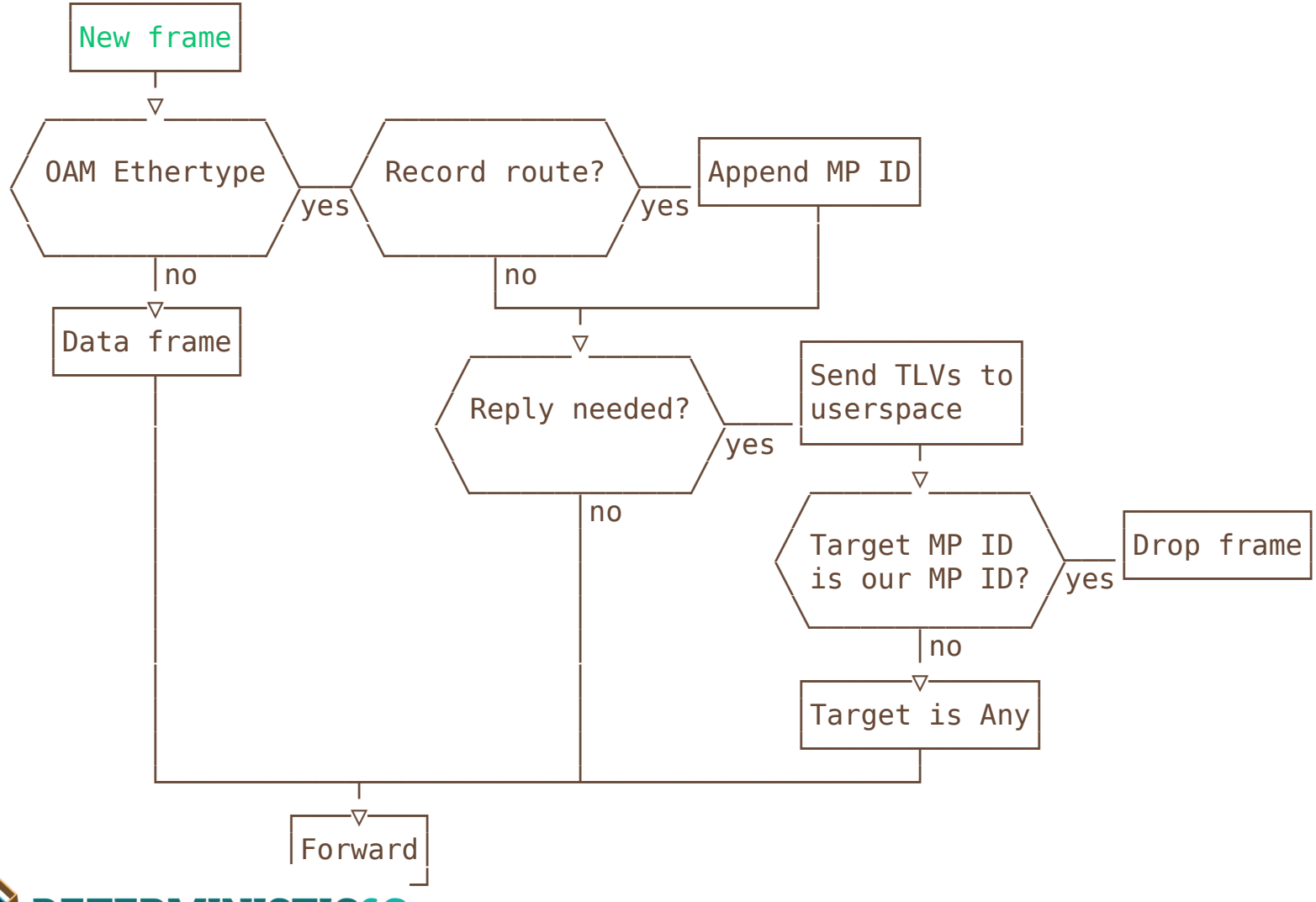
ERICSSON DETERMINISTIC6G

Observability: CFM Extensions for TSN

CFM Extensions for TSN

- Based on IEEE 802.1ag Connectivity Fault Management: Continuity Check (CC), Loopback (LB message and reply), Linktrace (LT)
  - Protocol messages between Maintenance Points (MP)
- Our focus is in FRER
  - CC between MPs
  - Our contribution: Record Route (RR) and FRER recovery graph discovery with it

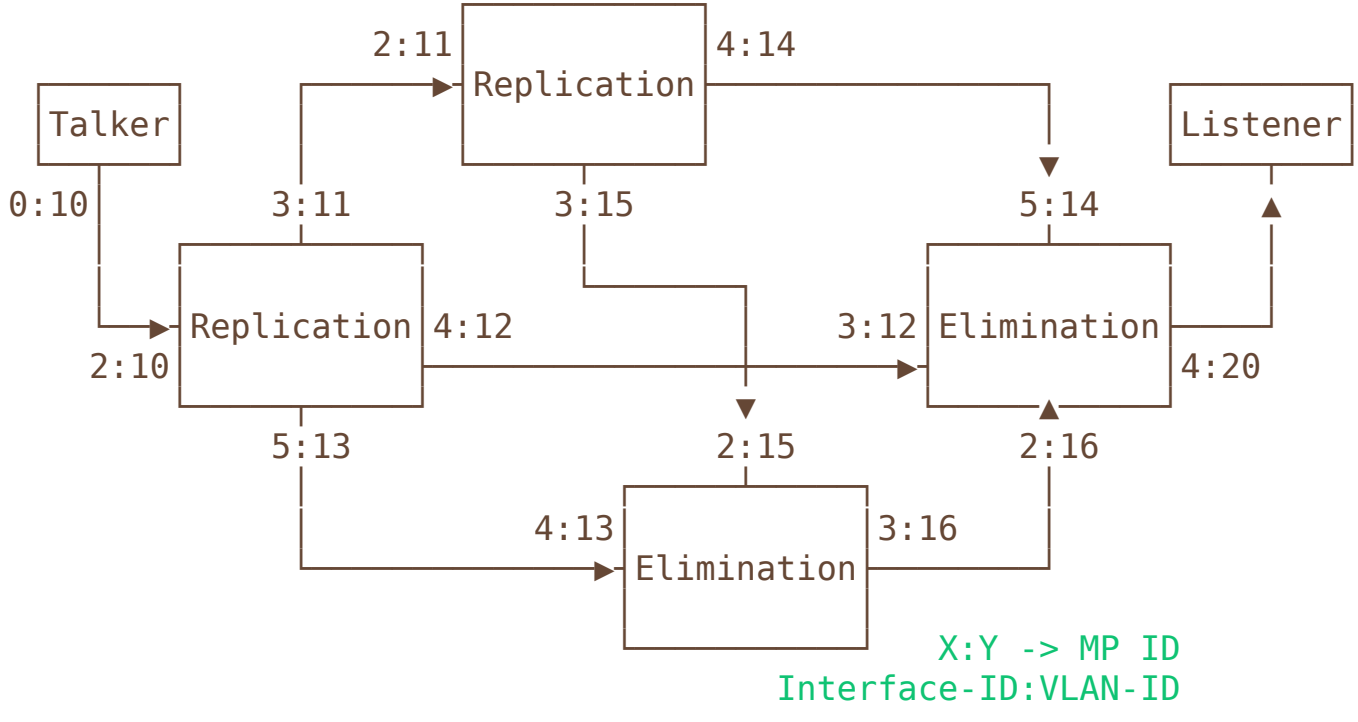
OAM Frame Processing (Simplified)



ERICSSON

DETERMINISTIC6G

Sample Topology

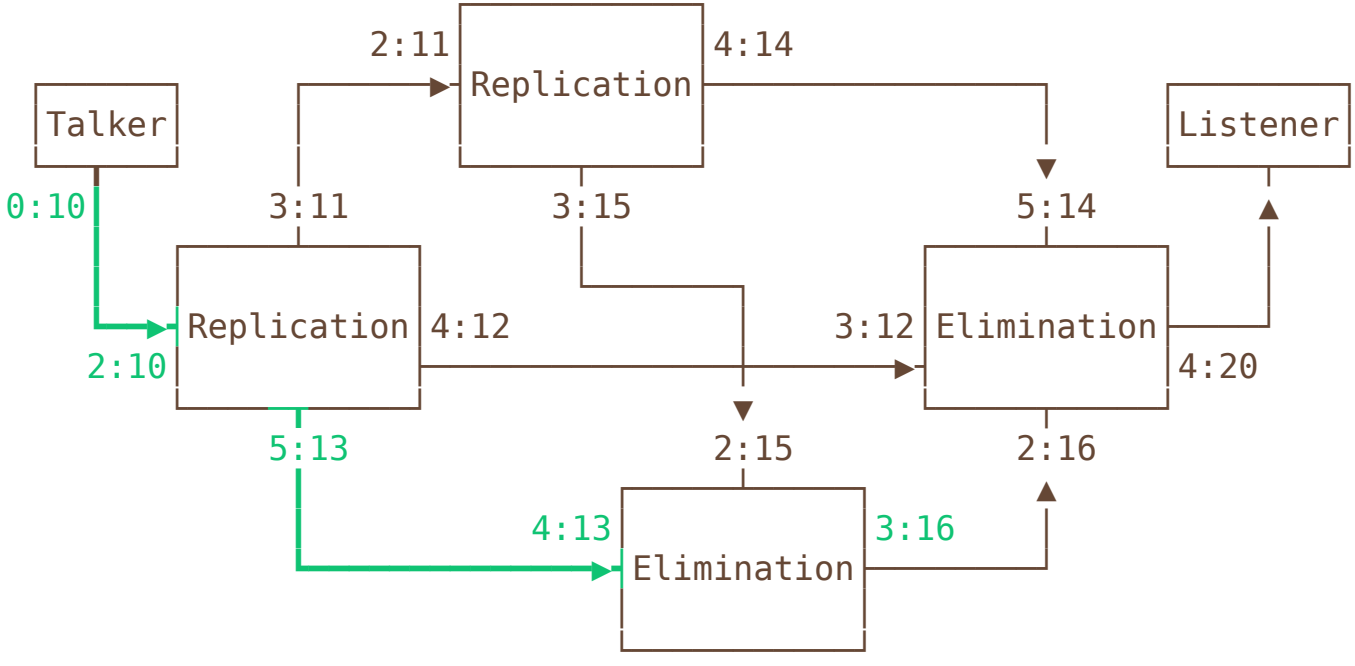


ERICSSON DETERMINISTIC6G

Continuity Check (CC)

```
oamgen -v 10 -c 1 -I tx-eth0 -m CC --mpid 3:16
50 bytes from 8e:23:a1:28:8a:11, rtag_seq=0, oam_type=OAM_CC_REPL

--- MP-ID: 3:16 statistics ---
1 packets transmitted, 1 received, 0% packet loss
```

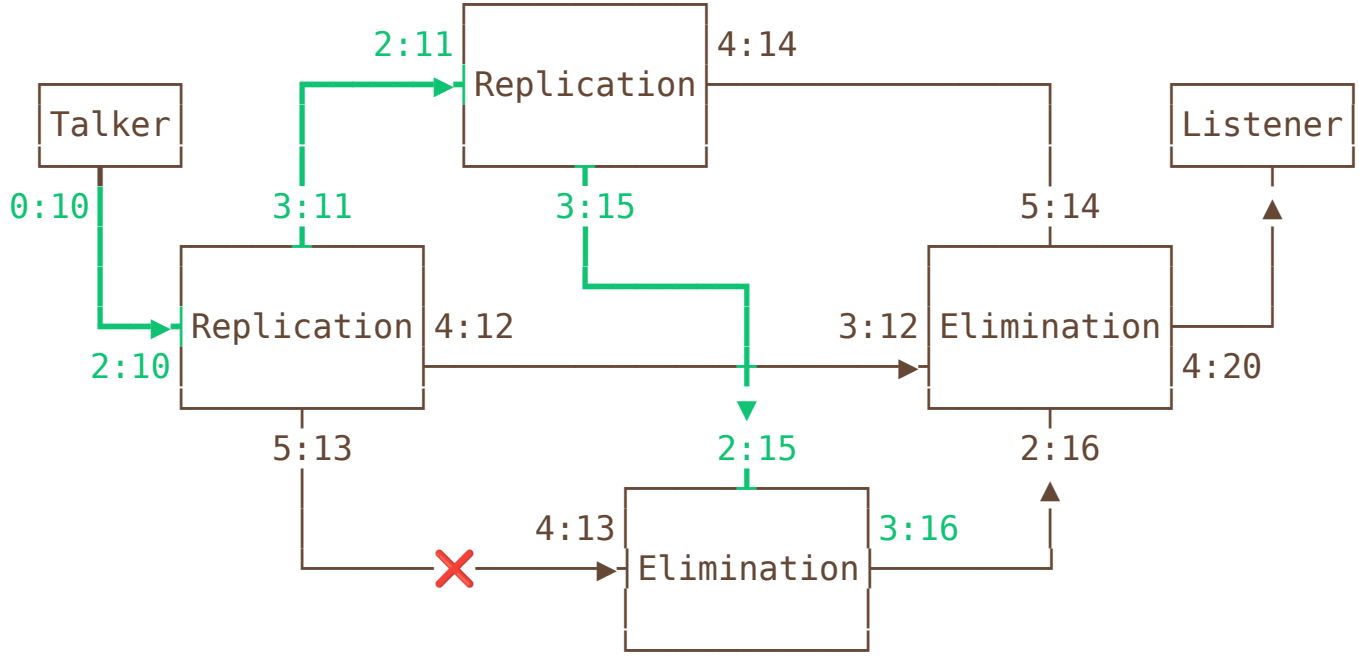
Record Route (RR)




```
oamgen -v 10 -c 1 -I tx-eth0 -m RR --mpid 3:16
98 bytes from 8e:23:a1:28:8a:11, rtag_seq=0, oam_type=OAM_RR_REPL, RR=[2:10 5:13 4:13 3:16]

--- MP-ID: 3:16 statistics ---
1 packets transmitted, 1 received, 0% packet loss
```

ERICSSON DETERMINISTIC6G

Path Failure




```
oamgen -v 10 -c 1 -I tx-eth0 -m RR --mpid 3:16
122 bytes from 8e:23:a1:28:8a:11, rtag_seq=0,
    oam_type=OAM_RR_REPL, RR=[2:10 3:11 2:11 3:15 2:15 3:16]

--- MP-ID: 3:16 statistics ---
1 packets transmitted, 1 received, 0% packet loss
```

FRER Graph Discovery by Record Route

```
oamgen -v 10 -c 1 -I tx-eth0 -m RR --mpid any -x 14
86 bytes from 8e:23:a1:28:8a:11, rtag_seq=0, oam_type=OAM_RR_REPL, RR=[2:10 5:13 4:13]
62 bytes from 9e:e5:7a:36:45:a1, rtag_seq=0, oam_type=OAM_RR_REPL, RR=[2:10]
98 bytes from 8e:23:a1:28:8a:11, rtag_seq=0, oam_type=OAM_RR_REPL, RR=[2:10 5:13 4:13 3:16]
110 bytes from 8e:23:a1:28:8a:11, rtag_seq=0, oam_type=OAM_RR_REPL, RR=[2:10 3:11 2:11 3:15 2:15]
74 bytes from 9e:e5:7a:36:45:a1, rtag_seq=0, oam_type=OAM_RR_REPL, RR=[2:10 5:13]
86 bytes from 3a:e5:27:34:50:04, rtag_seq=0, oam_type=OAM_RR_REPL, RR=[2:10 4:12 3:12]
74 bytes from 9e:e5:7a:36:45:a1, rtag_seq=0, oam_type=OAM_RR_REPL, RR=[2:10 4:12]
74 bytes from 9e:e5:7a:36:45:a1, rtag_seq=0, oam_type=OAM_RR_REPL, RR=[2:10 3:11]
98 bytes from 3a:e5:27:34:50:04, rtag_seq=0, oam_type=OAM_RR_REPL, RR=[2:10 4:12 3:12 4:20]
110 bytes from 3a:e5:27:34:50:04, rtag_seq=0, oam_type=OAM_RR_REPL, RR=[2:10 5:13 4:13 3:16 2:16]
110 bytes from 3a:e5:27:34:50:04, rtag_seq=0, oam_type=OAM_RR_REPL, RR=[2:10 3:11 2:11 4:14 5:14]
86 bytes from 86:2c:e3:1c:77:91, rtag_seq=0, oam_type=OAM_RR_REPL, RR=[2:10 3:11 2:11]
98 bytes from 86:2c:e3:1c:77:91, rtag_seq=0, oam_type=OAM_RR_REPL, RR=[2:10 3:11 2:11 4:14]
98 bytes from 86:2c:e3:1c:77:91, rtag_seq=0, oam_type=OAM_RR_REPL, RR=[2:10 3:11 2:11 3:15]

--- MP-ID: any statistics ---
1 packets transmitted, 14 received, 0% packet loss
```

ERICSSON DETERMINISTIC6G

Observability: Table Lookup Tracing

Motivation

- Our focus so far: what happens between the nodes?
- Limitations of this approach
  - Actual forwarding = Linux config + static config
  - TSN/DetNet forwarding (e.g.: TE) configured
  - Does the forwarding config execute what is expected?

Limitations

- Wireshark does not record routing lookups
- Policy based lookups are complicated, not to mention recursive lookups and loops
  - VRFs, prioritized policy tables, SRv6 end behaviors and encaps
- SRv6 DetNet dataplane (*draft-varga-detnet-srv6-data-plane*)
  - DetNet flowID and sequence number encoded in the DetNet-specific SID (PREOF-SID)

Tablesnoop

- Open-source tool for tracing table lookups
  - URL: github.com/EricssonResearch/tablesnoop
- IPv4/IPv6 policy and routing lookup tracing
- Network namespace aware (for Mininet/Docker/Kubernetes)
- Unsuccessful lookups are traced too
- Verify if TSN/DetNet and Linux config matching properly

Tablesnoop with Verbose Output

```
netns: 4026533619 iif: lo oif: - table id: 255 dscp: 0 flowlabel: 8 v6: src: 2001:abba:c:a3::a3  dst: 2001:be1a:c:d2::d2
netns: 4026533619 iif: lo oif: - table id: 254 dscp: 0 flowlabel: 8 v6: src: 2001:abba:c:a3::a3  dst: 2001:be1a:c:d2::d2 --> gw: 2001:abba:c:8a32::2 egress: to-r2-c
netns: 4026533619 rule6: pref: 32766 table: 254
netns: 4026533796 iif: vrf99 oif: - table id: 255 dscp: 0 flowlabel: 8 v6: src: 2001:abba:c:a3::a3  dst: 2001:be1a:c:d2::d2
netns: 4026533796 iif: vrf99 oif: - table id: 99 dscp: 0 flowlabel: 8 v6: src: 2001:abba:c:a3::a3  dst: 2001:be1a:c:d2::d2 --> gw: fe80::1888:8ff:fed4:798d egress: to2-r4
netns: 4026533796 rule6: pref: 1000 table: 0 l3mdev: 1
netns: 4026533796 iif: vrf99 oif: - table id: 255 dscp: 0 flowlabel: 8 v6: src: 2001:fade:0:2::  dst: 5f00:c51d:4:e099::
netns: 4026533796 iif: vrf99 oif: - table id: 99 dscp: 0 flowlabel: 8 v6: src: 2001:fade:0:2::  dst: 5f00:c51d:4:e099::
netns: 4026533796 iif: vrf99 oif: - table id: 254 dscp: 0 flowlabel: 8 v6: src: 2001:fade:0:2::  dst: 5f00:c51d:4:e099:: --> gw: 2001:a1fa:0:8022::22 egress: to-te2
netns: 4026533796 rule6: pref: 32766 table: 254
netns: 4026533855 iif: to-r2 oif: - table id: 255 dscp: 0 flowlabel: 8 v6: src: 2001:fade:0:2::  dst: 5f00:c51d:4:e099::
netns: 4026533855 iif: to-r2 oif: - table id: 88 dscp: 0 flowlabel: 8 v6: src: 2001:fade:0:2::  dst: 5f00:c51d:4:e099:: --> gw: :: egress: to-r2
netns: 4026533855 rule6: pref: 32765 table: 88
netns: 4026533855 iif: to-r2 oif: - table id: 255 dscp: 0 flowlabel: 5a93 v6: src: 2001:fade:0:22::  dst: 5f00:c51d:4:e2f8::
netns: 4026533855 iif: to-r2 oif: - table id: 254 dscp: 0 flowlabel: 5a93 v6: src: 2001:fade:0:22::  dst: 5f00:c51d:4:e2f8:: --> gw: 2001:a1fa:0:8022::2 egress: to-r2
netns: 4026533855 rule6: pref: 32766 table: 254
netns: 4026533796 iif: to-te2 oif: - table id: 255 dscp: 0 flowlabel: 5a93 v6: src: 2001:fade:0:22::  dst: 5f00:c51d:4:e2f8::
netns: 4026533796 iif: to-te2 oif: - table id: 254 dscp: 0 flowlabel: 5a93 v6: src: 2001:fade:0:22::  dst: 5f00:c51d:4:e2f8:: --> gw: fe80::1888:8ff:fed4:798d egress: to2-r4
netns: 4026533796 rule6: pref: 32766 table: 254
netns: 4026534032 iif: to-r2 oif: - table id: 255 dscp: 0 flowlabel: 5a93 v6: src: 2001:fade:0:22::  dst: 5f00:c51d:4:e2f8::
netns: 4026534032 iif: to-r2 oif: - table id: 254 dscp: 0 flowlabel: 5a93 v6: src: 2001:fade:0:22::  dst: 5f00:c51d:4:e2f8:: --> gw: :: egress: sr0-csid
netns: 4026534032 rule6: pref: 32766 table: 254
netns: 4026534032 iif: to-r2 oif: - table id: 255 dscp: 0 flowlabel: 5a93 v6: src: 2001:fade:0:22::  dst: 5f00:c51d:e2f8::
netns: 4026534032 iif: to-r2 oif: - table id: 254 dscp: 0 flowlabel: 5a93 v6: src: 2001:fade:0:22::  dst: 5f00:c51d:e2f8:: --> gw: :: egress: sr0-csid
netns: 4026534032 rule6: pref: 32766 table: 254
netns: 4026534032 iif: to-r2 oif: - table id: 254 dscp: 0 flowlabel: 8 v6: src: 2001:fade:0:2::  dst: 5f00:c51d:4:e099:: --> gw: :: egress: vrf99
netns: 4026534032 iif: to-r2 oif: - table id: 99 dscp: 0 flowlabel: 8 v6: src: 2001:abba:c:a3::a3  dst: 2001:be1a:c:d2::d2 --> gw: 2001:be1a:c:8d24::d2 egress: to-du2-c
netns: 4026534327 iif: to-r4-c oif: - table id: 255 dscp: 0 flowlabel: 8 v6: src: 2001:abba:c:a3::a3  dst: 2001:be1a:c:d2::d2 --> gw: :: egress: lo
netns: 4026534327 rule6: pref: 0 table: 255
```

ERICSSON

DETERMINISTIC6G

Thank you!

XDPFRER: github.com/EricssonResearch/xdpfrer

TSN proxy: github.com/EricssonResearch/tsn-proxy

Tablesnoop:
github.com/EricssonResearch/tablesnoop

E-mail: ferenc.fejes@ericsson.com

DETERMINISTIC6G Grant Agreement No. 101096504

The DETERMINISTIC6G project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No. 101096504.

If you need further information, please contact the coordinator: János Harmatos, ERICSSON

E-Mail: coordinator@deterministic6g.eu or visit: www.deterministic6g.eu