Introduction

Recapitulation

• Programming Protocol-independent Packet Processors
• EdgeCore Wedge100BF-32QS
• GÉANT P4 Network
• Network Operating System

On-going project and tests

• Packet and Flow Marking Specification
• First approach: layer 3 routing
• Second approach: layer 2 bridging

Conclusions and Future lines
Recapitulation
Programming Protocol-independent Packet Processors: P4 language

Language for programming the data plane of network devices

- Define how packets are processed
- P4 program structure: header types, parser/deparser, match-action tables, user-defined metadata and intrinsic metadata

Domain-specific language designed to be implementable on a large variety of targets

- Programmable network interface cards, FPGAs, software switches and hardware ASICs.
EdgeCore Wedge100BF-32QS

- 100GbE Data Center Switch
  - Bare-Metal Hardware
  - L2/L3 Switching
  - 32xQSFP28 Ports
- Data-Plane Programmability
  - Intel Tofino Switch Silicon
  - Barefoot Networks
- Quad-Pipe Programmable Packet Processing Pipeline
  - 6.4 Tbps Total Bandwidth
- CPU: Intel x86 Xeon 2.0GHz
  - 8-core/48GB/2TB SSD

Figure: Intel Tofino P4-programmable Ethernet Switch ASIC

Figure: EdgeCore Wedge100BF-32QS
Network Operating System

RARE/FreeRtr

- Controls the data plane by managing entries in routing tables
- Free and open source router operating system
- Export forwarding tables to DPDK or hardware switches
  - via OpenFlow or P4lang
- No global routing table
  - Every routed interface must be in a virtual routing table
On-going project and tests
Packet and Flow Marking Specification

Flow label field of IPv6 header: 20 bits

- 5 entropy bits to match RFC 6436
- 9 bits to define the science domain
- 6 bits to define the application/type of traffic

Astro/HEP Science Domains:
- Reserved - 0
- Default - 65536
- ATLAS - 32768
- CMS - 98304
- LHCb - 16384
- ALICE - 81920
- Bellell - 49152
- SKA - 114688
- LSST - 73728
- DUNE - 8192

Application:
- Reserved - 0
- Default - 4
- perfSONAR - 8
- Cache - 12
- DataChallenge - 16
First approach: layer 3 routing

Network configuration:

- Virtual Routing Forwarding
- Policy-based routing based on flow label field value
  - Flow label 10 → VLAN 40
  - Flow label 20 → VLAN 41
- SRV-01 managed by Cisco TRex Realistic Traffic Generator
  - Python script Scapy library: generate IPv6 packets flow label tagged
  - Cisco TRex Client: Python script → Scapy library
  - Cisco TRex Server: get statistic of the traffic in real-time
- SRV-02 managed by DPDK FreeRtr
First approach: layer 3 routing

Cisco Trex Realistic Traffic Generator
- Open source → Intel DPDK-based processors
- Supports stateful and stateless mode
- Build your own packets by using Scapy library

Data Plane Development Kit (DPDK)
- High speed data packet applications
  - Data plane libraries and network interface controller polling-mode drivers to accelerate packet processing
- Offloads TCP packet processing from kernel
  - Higher computing efficiency
  - Higher packet throughput
First approach: layer 3 routing

Results:

• Successful PBR routing based on the flow label field

<table>
<thead>
<tr>
<th>E513–E–YECWH–1#show interfaces traffic</th>
</tr>
</thead>
<tbody>
<tr>
<td>interface</td>
</tr>
<tr>
<td>sdn1</td>
</tr>
<tr>
<td>sdn1.10</td>
</tr>
<tr>
<td>sdn1.20</td>
</tr>
<tr>
<td>sdn110</td>
</tr>
<tr>
<td>sdn110.40</td>
</tr>
<tr>
<td>sdn110.41</td>
</tr>
<tr>
<td>sdn111</td>
</tr>
<tr>
<td>sdn111.60</td>
</tr>
<tr>
<td>sdn3</td>
</tr>
<tr>
<td>sdn50</td>
</tr>
<tr>
<td>sdn50.50</td>
</tr>
<tr>
<td>sdn51</td>
</tr>
<tr>
<td>sdn51.70</td>
</tr>
</tbody>
</table>

Figure: P4 switch statistics per interface and VLAN

[NOTE: test performed for fl=10 → VLAN 40]

• Cisco TRex + DPDK FreeRtr exploits NIC at its full capacity

<table>
<thead>
<tr>
<th>Per port stats table</th>
</tr>
</thead>
<tbody>
<tr>
<td>ports</td>
</tr>
<tr>
<td>opackets</td>
</tr>
<tr>
<td>obytes</td>
</tr>
<tr>
<td>ipackets</td>
</tr>
<tr>
<td>ibytes</td>
</tr>
<tr>
<td>ierrors</td>
</tr>
<tr>
<td>oerrors</td>
</tr>
<tr>
<td>Tx Bw</td>
</tr>
</tbody>
</table>

Figure: Cisco TRex statistics on the transmitter and receiver side

[NOTE: test performed with Intel 82599ES 10 GbE Controller: IXGBE driver]
Second approach: layer 2 bridging

Network configuration:

- Emulates a Tier 1/0 link
- Tier1/0 routers
  - IPv4/IPv6 BGP peerings
- Tier0 router
  - LHCOPN production border router
- Pure layer 2 bridges
  - VLAN 1000: IPv4 traffic
  - VLAN 1001: IPv6 traffic
- Tier0 servers
  - OpenStack product servers
Second approach: layer 2 bridging

P4 switch network configuration: pure layer 2 bridges

```plaintext
access-list acl_all_ipv6_flowlabels
    # Match <Experiment> and <ANY Application>
    sequence 10 permit all any all any all flow 131076 & 261884
    sequence 11 permit all any all any all flow 65540 & 261884
    sequence 12 permit all any all any all flow 196612 & 261884
    sequence 13 permit all any all any all flow 32772 & 261884
    # Match <Experiment> and <perfSONAR Application>
    sequence 20 permit all any all any all flow 131072 & 261832
    sequence 21 permit all any all any all flow 65536 & 261832
    sequence 22 permit all any all any all flow 196608 & 261832
    sequence 23 permit all any all any all flow 32768 & 261632
    # Permit the rest of the traffic
    sequence 30 permit all any all any all
exit

interface sdn1.1000
    description [VLAN ID=1000]
    bridge-group 1
    no shutdown
    no log-link-change
    exit

interface sdn1.1001
    description [VLAN ID=1001]
    bridge-group 2
    bridge-filter ipv6in acl_all_ipv6_flowlabels
    no shutdown
    no log-link-change
    exit
```

ATLAS <any>
CMS   <any>
LHCb  <any>
ALICE <any>

ATLAS <perfSONAR>
CMS   <perfSONAR>
LHCb  <perfSONAR>
ALICE <perfSONAR>

VLAN 1000 belongs to bridge 1
VLAN 1001 belongs to bridge 2

Filter IPv6 traffic at the input based on the access-list sentences
Second approach: layer 2 bridging

IPv6 packets flow label tagged were generated by using:

- iperf3
- ipv6_flow_label library developed by Marian Babik

```
E513-E-YECWH-1#show access-list acl_all_ipv6_flowlabels
seq   txb  txp  rxb            rxp         last      timeout    cfg
10   0+0  0+0  0+12374638771  0+8743031  00:03:02  00:00:00  permit all any any all flow 131076&261884
11   0+0  0+0  0+37019728635  0+24984028  00:02:30  00:00:00  permit all any any all flow 65540&261884
12   0+0  0+0  0+23940164205  0+15797973  00:02:00  00:00:00  permit all any any all flow 196612&261884
13   0+0  0+0  0+18150017192  0+12017039  00:02:00  00:00:00  permit all any any all flow 32772&261884
20   0+0  0+0  0+8346726207   0+20005622  00:01:29  00:00:00  permit all any any all flow 131072&261632
21   0+0  0+0  0+25281078379  0+16663278  00:01:29  00:00:00  permit all any any all flow 65536&261632
22   0+0  0+0  0+28556351375  0+19008806  00:00:58  00:00:00  permit all any any all flow 196608&261632
23   0+0  0+0  0+37078713993  0+25770785  00:00:26  00:00:00  permit all any any all flow 32768&261632
30   0+0  0+0  0+2715536713   0+1802921   00:00:26  00:00:00  permit all any any all

Figure: Counters of the access-list on the P4 switch
```

---

Figure: iperf3 performance during the tests

---

30th March 2022

Programming Switches for Flow Label Accounting and Routing
Second approach: layer 2 bridging

The counters of the switch are exported to Prometheus DB and Grafana:
Conclusions and Future lines

• IPv6 flow label accounting and routing can be implemented at layer 3 and layer 2.

• Use marked traffic generated by WLCG applications for testing the flow label counters

• Evaluate the use of the flow label routing approach to implement MultiONE

• Evaluate the use of P4 applications to use it as SDN controller for NOTED
Acknowledgements

Frederic Loui
RARE Technical Leader
https://www.rare.freertr.net
https://docs.freertr.net
https://blog.freertr.net

Csaba Mate
RARE/freeRtr Lead Core Developer
https://freertr.net

Alexander Gall
https://github.com/alexandergall/bf-sde-nixpkgs
https://github.com/alexandergall/packet-broker
Thanks for your attention!
Programming Switches for Flow Label Accounting and Routing

CERN Data Center
IT-CS-CE Department

Carmen Misa Moreira
Edoardo Martelli