rare.freertr.net BIER implementation

P4 BMv2, TOFINO & DPDK dataplane

Csaba MATE  
GÉANT/KIFU – RARE/freeRtr Lead core developer

Frederic LOUI  
GÉANT/RENATER – RARE/Technical leader

IETF#110 Virtual meeting – BIER-WG

March 9th 2021

Public

www.geant.org
Agenda

• RARE/freeRtr in a nutshell
• BIER RFC’s/draft implementation
• RARE (2021) /freeRtr (2017) BIER implementation experiment
• BIER interworking with Junos
• “Loop unrolling” BIER replication
• Conclusion
RARE project: Group focus

- GEANT project sub-task: RARE
  - Control plane software
  - Multiple data planes
  - Interface them and the result is...

- Fully functional router
  - Running at hardware line rate
  - DIY “hackable/extensible” router
  - Control plane independence

One familiar platform

\[\downarrow\]

Multiple solutions

\[\downarrow\]

Each solution addresses

\[\downarrow\]

R&E

use case
RARE latest news (M27/48)

• RARE p4 targets
  - bmv2 software switch
  - Intel/barefoot Tofino on WEDGE-BF100-32X, APS-BF2556X-T1, others
  - under study

• RARE “p4” emulation targets

• RARE Network Programmable targets
  - Broadcom under study
What we have

• BIER in MPLS - RFC8296
  • All the BitString lengths in software
  • 256bit mode in all the dataplanes

• BIER ISIS – RFC8401

• BIER OSPF – RFC8444

• BIER IDR draft

• BIER PIM draft
Experience

- [www.in.nop.hu/trackMap.tcl](www.in.nop.hu/trackMap.tcl) - a live network running dpdk dataplanes and sometimes a tofino node
- [lg.nop.hu](lg.nop.hu) - an ISP like setup
- [inf.nop.hu/mtrack.tcl](inf.nop.hu/mtrack.tcl) - measured from multiple endpoints talking to each other 0-24
- Regular streaming to loudspeakers with vlc: [demo](www.geant.org)
- All over BIER, initially in sw, nowadays in the dataplane
- We had a successful interop with Juniper! Someone else?
- Forwarding pitfall we’re doing
demo.freertr.net - an online BIER trial with draft-idr for 2+ years

```
dn42#
dn42#
dn42#
dn42# show configuration
dn42# show configuration
dn42# show configuration
dn42# show configuration
router bgp 4
   router bgp 4
   redistribute connected
exit
terminate

interface loopback1
   description vrf forwarding
ipv4 address 1.1.1.1 255.255.255.255
no shutdown
no log-link-change
exit

dn42#
dn42# show ipv4 bier demo
dn42# show ipv4 bier demo
dn42# show ipv4 bier demo
prefix          index  base      oldbase    size
1.1.1.1/32      2       494811    0         3-256
172.23.43.90/32 2       494811    0         3-256
dn42#
```

```
dn42#
dn42# show configuration
dn42# show configuration
dn42# show configuration
dn42# show configuration
router bgp 4
   router bgp 4
   redistribute connected
exit
terminate

interface loopback1
   description vrf forwarding
ipv4 address 1.1.1.2 255.255.255.255
no shutdown
no log-link-change
exit

dn42#
dn42# show ipv4 bier demo
dn42# show ipv4 bier demo
dn42# show ipv4 bier demo
prefix          index  base      oldbase    size
1.1.1.1/32      1       620235    0         3-256
172.23.43.91/32 1       620235    0         3-256
```

Juniper’s vMX parsed the BIER info from OSPF

Prefix Length (2), length 1:
32
AF (3), length 1:
0
Flags (4), length 1:
0x00
Prefix (5), length 32:
2.2.2.111
BIER (9), length 16:
Sub-domain ID (1), length 1:
0
MT ID (2), length 1:
0
BFR-id (3), length 2:
111
MPLS (10), length 12:
Range size (1), length 1:
4
Label Range Base (2), length 3:
0x31646
BitString Length, length 4 bits:
3

cmc36@vmx> show lldp neighbors
Local Interface Parent Interface Chassis Id Port info System Name
ge-0/0/2 - 00:34:64:47:48:68 pwether2 sid
ge-0/0/1 - 00:6e:4e:5e:7a:2c pwether1 sid

cmc36@vmx>
the vMX populated the forwarding tables correctly
some more forwarding info

```
> to 1.1.2.11 via ge-0/0/2.0, Push 385064
mc36@vmx> show route table :bier-0-0.bier.0
:bier-0-0.bier.0: 3 destinations, 3 routes (3 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
111/16
  *[OSPF/10] 00:04:40, metric 2
  > to 1.1.1.11 via ge-0/0/1.0, Push 202310
123/16
  *[BIER/70] 00:09:03
    Local
222/16
  *[OSPF/10] 00:04:35, metric 2
  > to 1.1.2.11 via ge-0/0/2.0, Push 385064
mc36@vmx> show route table :bier-0.inet.9 detail | match "BCN|via"
  Next hop: 1.1.1.11 via ge-0/0/1.0
  Next hop: 1.1.2.11 via ge-0/0/2.0
mc36@vmx> show route table :bier-0-0.bier.0 detail | match "BCN|via"
  Next hop: 1.1.1.11 via ge-0/0/1.0
  Next hop: 1.1.2.11 via ge-0/0/2.0
mc36@vmx>
```
BFid set on the loopback on rare/freerrtr

```plaintext
router ospf 4
vrf left
router-id 1.1.1.111
traffeng-id 1.1.1.111
bier 256 1024
area 0 enable
area 0 traffeng
area 0 bier
exit
router ospf 4 3
vrf right
router-id 1.1.1.222
traffeng-id 1.1.1.222
bier 256 1024
area 0 enable
area 0 traffeng
area 0 bier
exit
interface loopback2
no description
vrf forwarding left
ipv4 address 2.2.2.111 255.255.255.255
router ospf4 2 enable
router ospf4 2 area 0
router ospf4 2 traffeng bandwidth 1000000000
router ospf4 2 bier index 111
no shutdown
no log-link-change
exit
interface loopback3
```
the static BIER encap tunnels with the setdel filter :)

```
delete interface pwether2 log-link-change
set interface pwether2 exit
set interface tunnel2
delete interface tunnel2 description
set interface tunnel2 tunnel key 111
set interface tunnel2 tunnel vrf left
set interface tunnel2 tunnel source loopback2
set interface tunnel2 tunnel destination 9.9.9.9
set interface tunnel2 tunnel domain-name 2.2.2.222
set interface tunnel2 tunnel mode bier
set interface tunnel2 vrf forwarding left
set interface tunnel2 ipv4 address 3.3.3.1 255.255.255.252
delete interface tunnel2 shutdown
delete interface tunnel2 log-link-change
set interface tunnel2 exit
set interface tunnel3
delete interface tunnel3 description
set interface tunnel3 tunnel key 222
set interface tunnel3 tunnel vrf right
set interface tunnel3 tunnel source loopback3
set interface tunnel3 tunnel destination 9.9.9.9
set interface tunnel3 tunnel domain-name 2.2.2.111
set interface tunnel3 tunnel mode bier
set interface tunnel3 vrf forwarding right
set interface tunnel3 ipv4 address 3.3.3.2 255.255.255.252
delete interface tunnel3 shutdown
delete interface tunnel3 log-link-change
set interface tunnel3 exit

sid#show config-differences | setdel
```
### BIER info from the vMX's left and right sides

#### show ipv4 bier left
```
2021-02-20 10:04:27
prefix   index   base     oldbase     size
2.2.2.123/32   123   800000   800000     3-256
2.2.2.222/32   222   800000   385064     3-256
```

#### show ipv4 bier right
```
2021-02-20 10:04:28
prefix   index   base     oldbase     size
2.2.2.111/32   111   800000   202310     3-256
2.2.2.123/32   123   800000   800000     3-256
```

#### show mpls forwarding | include bier|target
```
2021-02-20 10:04:41
label   vrf   iface   hop   label   targets   bytes
202310  left:4  null   null   unlabelled   bier   0
202311  left:4  null   null   unlabelled   bier   0
202312  left:4  null   null   unlabelled   bier   0
202313  left:4  null   null   unlabelled   bier   0
385064  right:4 null   null   unlabelled   bier   0
385065  right:4 null   null   unlabelled   bier   0
385066  right:4 null   null   unlabelled   bier   0
385067  right:4 null   null   unlabelled   bier   0
656330  v1:4    null   null   unlabelled   bier   0
656331  v1:4    null   null   unlabelled   bier   0
982822  v1:6    null   null   unlabelled   bier   0
982823  v1:6    null   null   unlabelled   bier   0
```
rare/freertr's forwarding info from the vMX's left side
first packets to the tunnel, the counters seems ok, so the vMX forwards perfectly!

<table>
<thead>
<tr>
<th>interface</th>
<th>state</th>
<th>tx</th>
<th>rx</th>
<th>drop</th>
</tr>
</thead>
<tbody>
<tr>
<td>loopback0</td>
<td>up</td>
<td>648</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>loopback2</td>
<td>up</td>
<td>66</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>loopback3</td>
<td>up</td>
<td>66</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>loopback42</td>
<td>up</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>loopback65535</td>
<td>up</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>template1</td>
<td>admin</td>
<td>0</td>
<td>0</td>
<td>368</td>
</tr>
<tr>
<td>bundle9</td>
<td>up</td>
<td>50532</td>
<td>53922</td>
<td>0</td>
</tr>
<tr>
<td>bundle9.11</td>
<td>up</td>
<td>2526</td>
<td>836</td>
<td>0</td>
</tr>
<tr>
<td>bundle9.12</td>
<td>up</td>
<td>46810</td>
<td>51858</td>
<td>0</td>
</tr>
<tr>
<td>bvi1</td>
<td>up</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>bvi2</td>
<td>up</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>bvi3</td>
<td>up</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>bvi4</td>
<td>up</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ethernet1</td>
<td>up</td>
<td>48512</td>
<td>4341</td>
<td>0</td>
</tr>
<tr>
<td>ethernet2</td>
<td>up</td>
<td>2020</td>
<td>49441</td>
<td>0</td>
</tr>
<tr>
<td>ethernet8</td>
<td>up</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ethernet9</td>
<td>up</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>pwether1</td>
<td>up</td>
<td>17497</td>
<td>17427</td>
<td>0</td>
</tr>
<tr>
<td>pwether2</td>
<td>up</td>
<td>17497</td>
<td>17427</td>
<td>0</td>
</tr>
<tr>
<td>tunnel2</td>
<td>up</td>
<td>12543</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>tunnel3</td>
<td>up</td>
<td>12543</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Forwarding pitfall without inter-replica knowledge: loop unrolling

- r4 and r5 got the IGMP report from the connected VLCs
- both looked up the group’s source in mrib, both decided to send PIM in BIER to r1
- both looked up r1 loopback’s bfdid from the rib and sent the PIM in BIER join
- first I tried the plain old PIM behavior: r1 sent the BIER encapped mcast on the same interface where it got the PIM in BIER join from, but r4 and r5 was able to hash to different incoming interfaces
- then I tried to do a rib lookup on r1 for r4 and r5’s loopbacks, but r1 was able to hash to different outgoing interfaces
- so for now, I use only the first path on r1 from the rib lookup and for now, duplication happens on the last possible hop
- RFC 6754 does not apply as r2 and r3 are unaware of the s,g. better idea?
Key take-away – We are ready to roll into production

• Automated testing: www.freertr.net/tests.html
• 3rd party testing via Spirent usage
  • (thanks PSNC@WB team)
• P4 profile calibration
• DPDK is in operation
• Production instance

• Someone else? :)
Special thanks …

And others …
Who makes this possible!
Thank you

Any questions?

www.geant.org