# Building A Cheaper Peering Router

## (Actually it's more about buying a cheaper router and applying some routing tricks)

Tom Scholl <tscholl@nlayer.net> nLayer Communications, Inc.

# What's this all about?

- Network infrastructure can be expensive.
- One of the most common issues encountered when networks needs peering upgrades are the router ports themselves.
  - If not for your network, than for the network you want to peer with.
  - No ports available, not enough space and/or power for new boxes.
  - Price of ports is an issue, especially when non revenue generating.
- 10GigabitEthernet is the current standard for interconnection.
  - And many smart networks have moved from SONET capable routers (GSR, CRS, T-series, etc) to "Ethernet-centric" boxes for peering.
  - But peering routers are still a significant expense.
  - And many existing peering edge routers are running out of steam in terms of ports, density, etc.

# Vendors have reacted to this need

- Historically you had very expensive core routers
  - Cisco GSR and CRS
  - Juniper T-series
  - etc.
- And very cheap but somewhat feature limited customer boxes.
  - Cisco 6500
  - etc.
- Vendors have also created a “middle tier” in features and price
  - Juniper MX
  - Cisco 7600 with ES cards
  - Cisco ASR 9000
  - Foundry MLX/XMR
  - etc.

# This is nice…but…

- In the last few years there's been an explosion of much cheaper and denser 10 Gigabit Ethernet boxes.
  - Targeted at the datacenter / top-of-rack role.
  - Supporting only datacenter optics (SFP+ only, nothing long reach).
  - Sometimes lacking large packet buffers.
  - Lacking many advanced features an ISP might want.
  - And often using outsourced routing ASICs ("commodity silicon")
- But they've got some pretty neat characteristics too:
  - Small boxes (1U or 2U) with 24 or 48+ port 10GE density.
  - Larger boxes with support for 16-32 slots of Nx10GE cards.
  - And SFP+ optics CAN significantly reduce infrastructure costs.

# What are some of these boxes?

- Cisco ASR9K
  - Based on EZchip chips
- Juniper EX
  - Based on Marvell chips
- Dell
  - Based on Broadcom chips
- Force10
  - Based on Broadcom chips
- Foundry/Brocade
  - Based on Broadcom chips
- Arista
  - Based on Fulcrum chips

# Could these cheap boxes have a viable role in a service provider network?

- Your existing vendors will strongly suggest no, obviously ☺
- There's a few challenges preventing you from doing this:
  - Limited FIB size
    - Internet is ~330,000 routes, these boxes can do maybe ~12,000
  - Lack of QoS features
    - Hierarchical QoS? At best, maybe 4 queues per interface
  - Some of these boxes lack any kind of decent software
    - You like pipe and regex? Ha…
  - Access-Lists / Packet Filters
    - Protecting your network is important
  - Lack of forwarding features
    - MPLS? IPv6? You should be happy with IPv4.
- But maybe not all of these features are hard to implement…

# Getting around FIB constraints

- Separation of the RIB vs. FIB is critical.
  - The RIB holds information from the routing protocols (BGP, IGP).
  - The FIB holds the final table used for forwarding packets.
- We'll need to have a large RIB, since we'll need to exchange lots of BGP routes with neighbors (transits, peers, customers)
- Fortunately RAM is cheap, and even the 1U boxes are shipping with 1GB of DRAM, so this is less of a problem.
- The key is to not install every single route in the FIB, only what you *need* should be there.

# QoS?

- How many people use QoS extensively within their network?

- Most networks focused on transporting bits across the Internet generally arent the major consumers of heavy QoS functionality.
- People selling multi-services (L2VPN, L3VPN, Transit, etc) on a converged network are.
  - (Those people are typically telcos and large carriers)

- The simple QoS features these devices have should be sufficient or not a show stopper.

# Lack of forwarding features?

- As these boxes were destined for the datacenter, they're devoid of features used by most service providers.
- But one application driving datacenter boxes is Cloud…
  - And cloud applications require the ability for servers to talk to each other across networks larger than what L2 would be reasonable for.
- This is driving vendors to support more modern L2 networks.
  - TRILL
  - 802.1aq (Shortest Path Bridging)
  - MPLS VPNs (e.g. VPLS)
- Much of this is still on the roadmap, but there is significant demand and support for implementing it.

# Packet Filters

- It's a pretty critical requirement to have some sort of ACLs on your edge to prevent bad things:
  - Protect your infrastructure
  - Protect your customers
    - Hello packet police? Yes, our IRC server is getting attacked and…
- At best expect simple packet filters up to layer 3 or 4.
- Don't expect logging or complex matches
  - Packet length, policing, IP options are probably out the window
- Focus on filtering packets towards infrastructure and perhaps a combination of Null routing portions of infrastructure space you don't want packets going.

# Decent software?

- Many cheap 1U/2U boxes have really horrible software code.
- This is because chip manufacturers don't know the first thing about writing good software for routers.
  - And why should they? It's not their area of expertise.
  - Can you imagine if Intel had to write your computer's OS?
- Some vendors will just ship the reference software.
  - Dell, Force10 S50, etc.
  - Most try to duplicate Cisco IOS, but at a 1st grade level.
- Other vendors will modify their existing OS to control the 3rd party ASICs.
  - Cisco, Juniper, Foundry/Brocade, etc.

# Decent software? (cont'd)

- The OpenFlow project is particularly interesting here.
  - Allows developers to write third party software to control the router.
  - This removes the dependency that every router must have a decent control plane and software running on it.
  - Instead, you write your control plane and run it off-router, then push the FIB results to the hardware.

# What's unique about this approach?

- To pull off routing without a full table, we're going to rely upon BGP Unicast-Label.
- BGP Unicast-Label is another BGP address-family, similar to IPv4 unicast, IPv6 unicast, etc.
- What is unique about BGP Unicast-Label is that it allows you to allocate a MPLS label for a prefix.
  - This is similar to how LDP and RSVP allocate labels.

# So lets try a little experiment…

- For testing purposes, we’re working on a Juniper EX4500
  - Line rate 48x10GE in 2U, 12K FIB, 1GB of DRAM.
- Step 1: Hang it off something “smart”
  - Our cheap box is a stub that hangs off of a larger core router.
  - Assume that Core router has a full table and our little 1U box will handle links to peers, transit and customers.
- You only really need to do this if you need a route of last resort and your small cheap box can handle encapsulating traffic in some protocol (GRE, MPLS, etc)

# Participate in the IGP

- Step 2: Establish IGP adjacencies

- This is critical for a few reasons:
  - We want BGP next-hops visible when we advertise routes via BGP
    - Passing IGP costs into BGP MED
    - BGP next-hop validation/reachability
  - Link-liveliness detection with the rest of the network

- A well designed IGP has a small number of routes anyways.

# Split the RIB from FIB

- Step 3: Define what routes you want installed in the FIB
  - On Juniper this is done in the “forwarding-table export” policy.

- Some ideas:
  - Directly connected interfaces
    - It’s directly connected, you probably want to know you can forward to it
  - IGP routes
    - ISIS or OSPF
  - Internal / Customer networks
    - Match on BGP community
  - Default-Route pointing internally
    - Questionable – Depends if you trust your peers

# Establishing iBGP connectivity

- Step 4: Bring up BGP internally

- Utilize BGP Add-Paths
  - Allows you to advertise multiple paths for the same prefix, not just the single best path.

- This is a new feature, but a very cool one.
  - Has a significantly higher memory footprint, but you can control what routes you want to advertise duplicates of.

# Next-Hop Tricks

- Step 5: Use BGP Unicast-Label + MPLS to bypass lookups
- Advertise each peer point-to-point as into iBGP with a unicast-label.
  - Redistributing directly connected (/30s for example) only originates implicit-null labels, which is useless.
  - You can however generate a /32 static route of the peers IP address with a next-hop of the peers /32 address:

```
route 69.22.173.26/32 next-hop 69.22.173.26;
route 69.22.173.5/32 next-hop 69.22.173.5;
```

Yes, it looks funny but it works.

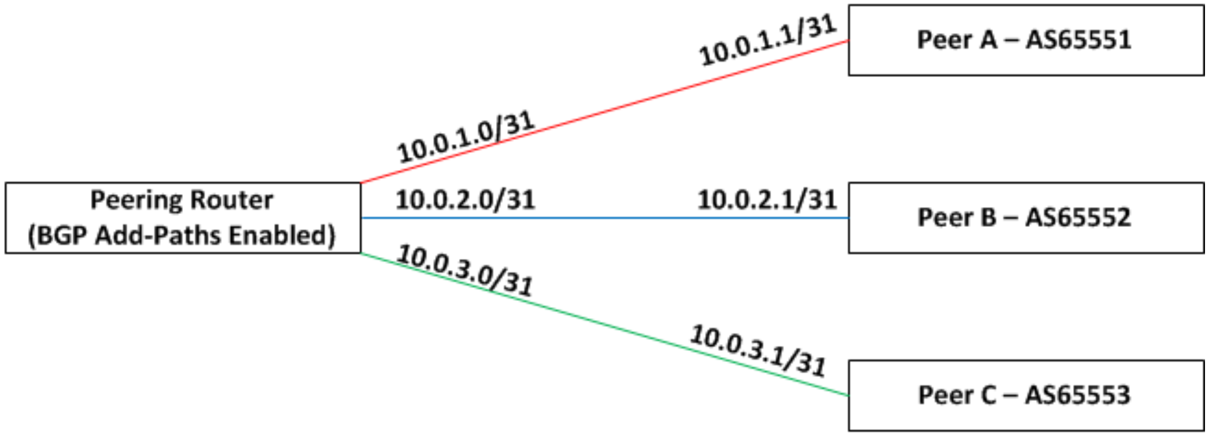
# Next-Hop Tricks

- When you learn routes from a peer/customer/transit, do not rewrite BGP next-hop-self.
  - Advertise the true next-hop (remote end of a /30)
- IP lookup is bypassed as you are performing a MPLS label operation
  - POP and forwarding traffic out the egress interface

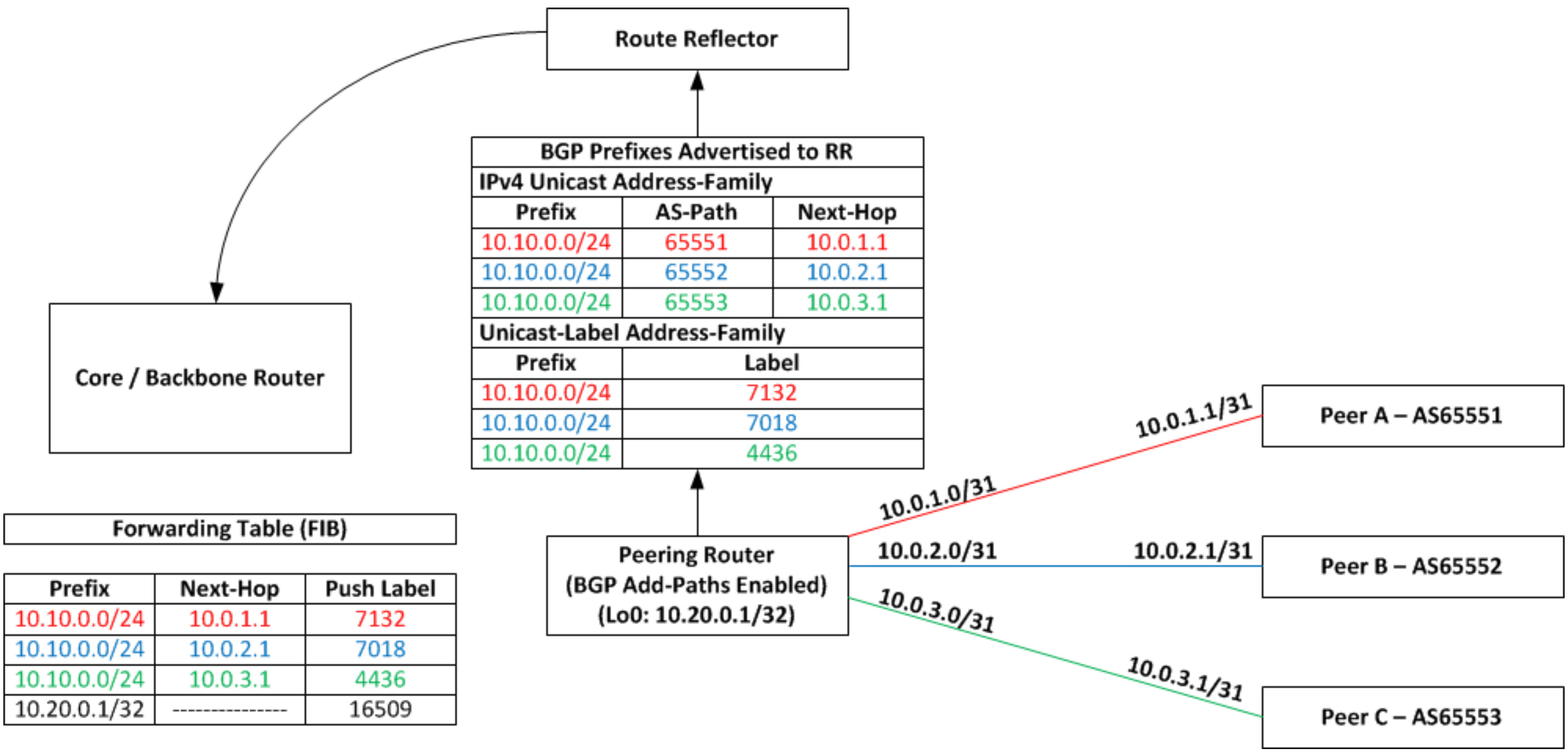
# Example: RIB & FIB

| Routing Table (RIB) | | |
|---|---|---|
| Prefix | AS-Path | Next-Hop |
| 10.10.0.0/24 | 65551 | 10.0.1.1 |
| 10.10.0.0/24 | 65552 | 10.0.2.1 |
| 10.10.0.0/24 | 65553 | 10.0.3.1 |

| Forwarding Table (FIB) | |
|---|---|
| Prefix / Label | Egress Interface |
| 7132 | xe-0/0/1 |
| 7018 | xe-0/0/2 |
| 4436 | xe-0/0/3 |
| 10.0.1.1/31 | xe-0/0/1 |
| 10.0.2.1/31 | xe-0/0/2 |
| 10.0.3.1/31 | xe-0/0/3 |



# Advertising routes back to the network

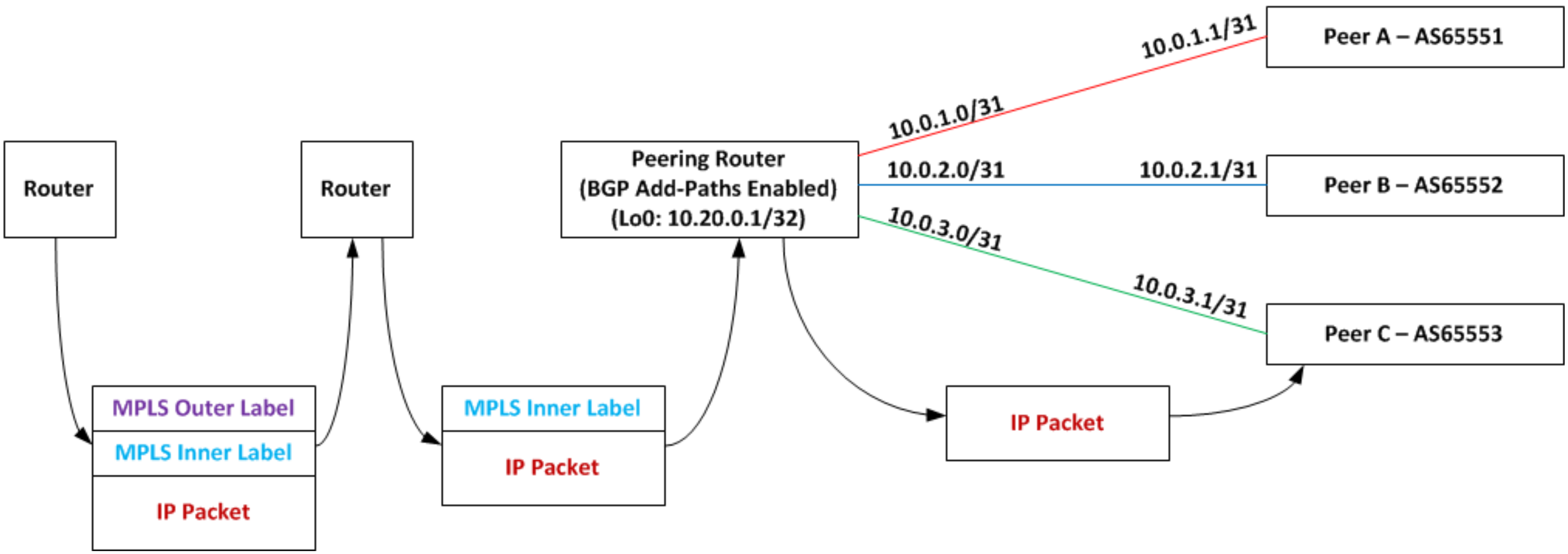



| BGP Prefixes Advertised to RR | | |
|---|---|---|
| **IPv4 Unicast Address-Family** | | |
| **Prefix** | **AS-Path** | **Next-Hop** |
| 10.10.0.0/24 | 65551 | 10.0.1.1 |
| 10.10.0.0/24 | 65552 | 10.0.2.1 |
| 10.10.0.0/24 | 65553 | 10.0.3.1 |
| **Unicast-Label Address-Family** | | |
| **Prefix** | **Label** | |
| 10.10.0.0/24 | 7132 | |
| 10.10.0.0/24 | 7018 | |
| 10.10.0.0/24 | 4436 | |

**Forwarding Table (FIB)**

| Prefix | Next-Hop | Push Label |
|---|---|---|
| 10.10.0.0/24 | 10.0.1.1 | 7132 |
| 10.10.0.0/24 | 10.0.2.1 | 7018 |
| 10.10.0.0/24 | 10.0.3.1 | 4436 |
| 10.20.0.1/32 | --------------- | 16509 |

# Packet Flow



Router
Router
Peering Router
(BGP Add-Paths Enabled)
(Lo0: 10.20.0.1/32)
10.0.1.0/31
10.0.1.1/31
10.0.2.0/31
10.0.2.1/31
10.0.3.0/31
10.0.3.1/31
Peer A – AS65551
Peer B – AS65552
Peer C – AS65553
MPLS Outer Label
MPLS Inner Label
IP Packet
MPLS Inner Label
IP Packet
IP Packet

# Examples

```
show route table mpls.0:

300064              *[VPN/170] 1d 12:18:12
                     > to 69.22.173.26 via ae1.80, Pop
300080              *[VPN/170] 1d 11:30:51
                     > to 69.22.173.5 via ae1.71, Pop
```

# Examples

```
Next-Hop: 69.22.173.26 (Label 300064)
 4  xe-3-0-0.cr1.ord1.us.nlayer.net (69.22.142.74)  124.127 ms  93.544 ms  93.526 ms
     MPLS Label=455769 CoS=0 TTL=1 S=0 ← Outer Label
     MPLS Label=300064 CoS=0 TTL=4 S=1 ← Inner Label
 5  xe-2-0-0-91.mx240-1.lab1.nlayer.net (69.22.173.34)  94.030 ms  101.947 ms  108.593 ms
     MPLS Label=300064 CoS=0 TTL=1 S=1 ← POP
 6  10.251.1.2 (10.251.1.2)  95.773 ms  96.924 ms  95.972 ms
```

```
Next-Hop: 69.22.173.5 (Label 30080)
 4  xe-3-0-0.cr1.ord1.us.nlayer.net (69.22.142.74)  99.206 ms  93.627 ms  97.893 ms
     MPLS Label=455769 CoS=0 TTL=1 S=0 ← Outer Label
     MPLS Label=300080 CoS=0 TTL=4 S=1 ← Inner Label
 5  xe-2-0-0-91.mx240-1.lab1.nlayer.net (69.22.173.34)  98.022 ms  104.927 ms  101.347 ms
     MPLS Label=300080 CoS=0 TTL=1 S=1 ← POP
 6  10.251.1.2 (10.251.1.2)  103.776 ms  95.709 ms  95.911 ms
```

# What about inbound traffic?

- That depends on the network.
- If your “internal routes” are larger than the FIB of your cheap router, you’re going to have to cheat.
  - By cheat, I mean point a default route to the next upstream router.
  - You can also point this to another router elsewhere in the network that has a full table (anycast “helper routers” as it were).
- If your internal routes are few, you can simply install those into the FIB.

- The problem with the default is you are at risk for someone pointing static routes at you.

# Wow, does this really work?

- It looks like it does…

```
> show route table mpls.0

    mpls.0: 7 destinations, 7 routes (7 active, 0 holddown, 0 hidden)
    + = Active Route, - = Last Active, * = Both

    0                  *[MPLS/0] 6w2d 04:10:23, metric 1
                        Receive
    1                  *[MPLS/0] 6w2d 04:10:23, metric 1
                        Receive
    2                  *[MPLS/0] 6w2d 04:10:23, metric 1
                        Receive
    1000001            *[MPLS/6] 03:37:35, metric 1
                      > to 69.22.173.17 via xe-0/0/9.0, Pop
    1000001(S=0)       *[MPLS/6] 02:08:27, metric 1
                      > to 69.22.173.17 via xe-0/0/9.0, Pop
    1000002            *[MPLS/6] 03:35:25, metric 1
                      > to 69.22.173.21 via xe-0/0/11.0, Pop
    1000002(S=0)       *[MPLS/6] 03:35:25, metric 1
                      > to 69.22.173.21 via xe-0/0/11.0, Pop
```

# Failboat

- …but doesn't:

```
[271451] mrvl_rt_entry_create: MRVL_RT-vrf:0 rt:1000001
    [271466] mrvl_rt_entry_install: MRVL_RT-vrf:0 rt:1000001, action:0
    [271467] mrvl_rt_entry_construct_ltt_entry: MRVL_RT-1000001
    [271470] mrvl_rt_entry_populate_ltt_entry: MRVL_RT-rt_nh tbl entry idx:25 entry count:1 type:unicast
    [271471] mrvl_rt_mpls_ltt_install: MRVL_RT-mpls ltt install device 0, entry 3277
    [271472] mrvl_rt_mpls_ltt_install: MRVL_RT-mpls ltt install device 1, entry 3277
    [271474] mrvl_rt_regular_mpls_entry_install: mpls rt tti_set failed: 5
    [271481] mrvl_rt_entry_create: MRVL_RT-vrf:0 rt:1000001(S=0)
    [271487] mrvl_rt_mpls_entry_create: MRVL_RT-rt:1000001(S=0) nh: 0
```

- Output from pfem daemon crashing after executing some choice show commands.

# Almost there…

- Most of the pieces are there to do this:
  - BGP Add-Paths to give you multiple BGP path visibility
  - RIB/FIB separation so you can operate BGP with peers, route full tables, full policy controls
- It is feasible to do this on other Juniper gear that actually support MPLS.
  - Working examples given on a MX240
- It works on some Cisco boxes, too.

# But there is hope

- MPLS support should be coming in future Juniper EX models
- Other merchant silicon boxes should have some MPLS support hopefully within the next year
  - Don’t expect great MPLS software implementations of RSVP, LDP.
  - May have to rely upon static/nailed up LSPs
  - Routing protocol functionality (BGP, OSPF, ISIS) may not be all there.
  - Juniper EX used as an example as it does the routing part fairly well.

- An open item is how well will these cheap boxes handle lookups and MPLS actions (push/pop/swap).

## Send questions, comments, complaints to:

Tom Scholl tscholl@nlayer.net