Why Link-State Matters

Andy Gospodarek

Member of Technical Staff — Cumulus Networks

LinuxCON Seattle 2015

Why Link-State Matters — Agenda

cumulus

Why do we need this?

What does the change look like?

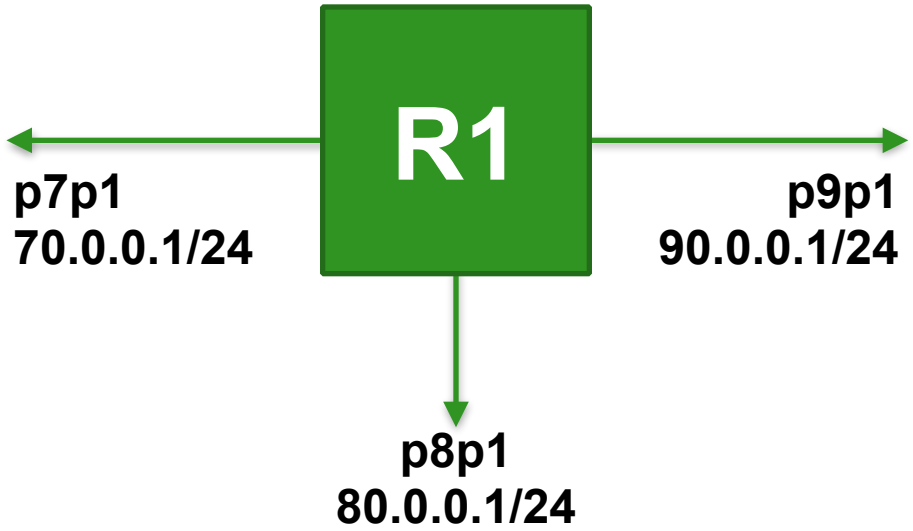
Any future changes planned?

Any lessons worth sharing?

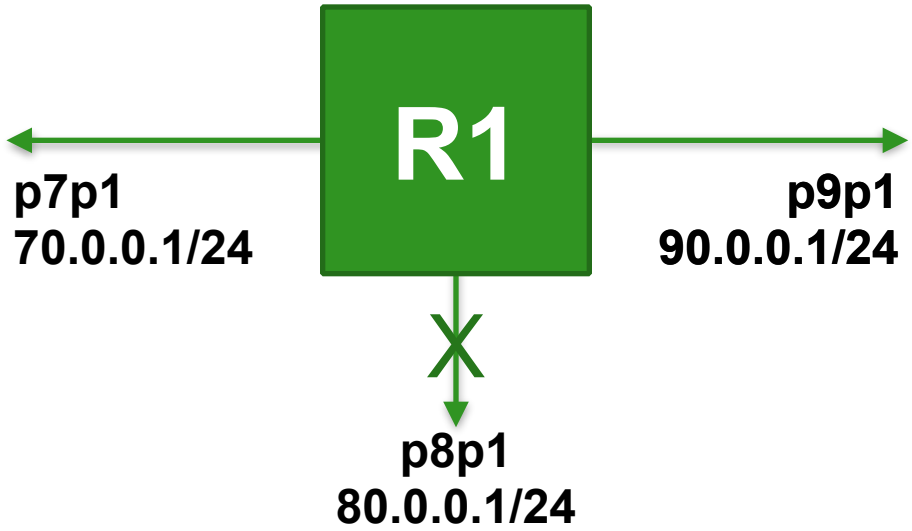
cumulus®

The Problem

Why Link-State Matters — Simple network setup



Why Link-State Matters — p8p1 goes link-down



Why Link-State Matters

cumulus

Result: Traffic black-holed

Why Link-State Matters

cumulus®

Isn’t this problem solved already?

Why Link-State Matters — Userspace solutions

cumulus

netplugd

ifplugd

NetworkManager

[insert name of favorite netlink-based application here]

Why Link-State Matters — Userspace solutions

cumulus®

All behave in a similar manner

Why Link-State Matters — Userspace solutions

cumulus

Listen for netlink events indicating link status change

cumulus®

Call a script

cumulus®

Potential issues with these tools

Why Link-State Matters— Userspace solutions

cumulus

Is the link down or just not configured?

Why Link-State Matters — Userspace solutions

```
# ip addr show p7p1
4: p7p1: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 ...
    link/ether 08:00:27:9d:62:9f brd ff:ff:ff:ff:ff:ff
```

Why Link-State Matters — Userspace solutions

cumulus

“I think this interface is supposed to be up, so I’ll go ahead and enable it.”

Why Link-State Matters — Userspace solutions

```
# ifup p7p1
```

Why Link-State Matters — Userspace solutions

cumulus

Result: Traffic black-holed

Why Link-State Matters

cumulus®

How can we make the kernel do this for us?

Why Link-State Matters

cumulus®

2 choices

Why Link-State Matters — Option 1

cumulus

After a carrier state change, NETDEV_CHANGE events are sent mark routes with a down interface for the next hop appropriately

Why Link-State Matters — Option 1

cumulus

Check for that mark each time a frame is forwarded or originates

Why Link-State Matters — Option 2

cumulus

Check interface link status for a next hop each time a frame is forwarded or originates

Why Link-State Matters

cumulus

Either one should be able to enable or disable the forwarding decision based on sysctl/netconf setting

Why Link-State Matters

cumulus

Which one should we use?

Why Link-State Matters

cumulus®

How about both?

cumulus

IPv4 FIB code uses Option 1

```
8a3d03166f19329b46c6f9e900f93a89f446077b
0eeb075fad736fb92620af995c47c204bbb5e829
96ac5cc9636318e7d92e72a3f1032456152a3a2f
974d7af5fcc295dcf8315255142b2fe44fd74b0c
```

cumulus

IPv6 FIB code uses Option 2

```
cea45e208d700e9d633a636384a49f19cda979b7
35103d11173b8fea874183f8aa508ae71234d299
```

Why Link-State Matters

cumulus

Implementation details aside, both produce the exact same user experience

cumulus

IPv4 example

Why Link-State Matters — Routing table with quagga running

```
# ip -4 route show
70.0.0.0/24 dev p7p1  proto kernel  scope link  src 70.0.0.1
80.0.0.0/24 dev p8p1  proto kernel  scope link  src 80.0.0.1 linkdown
80.0.0.0/24 via 90.0.0.2 dev p9p1  proto zebra  metric 20
90.0.0.0/24 dev p9p1  proto kernel  scope link  src 90.0.0.1
100.0.0.0/24
   nexthop via 70.0.0.2  dev p7p1 weight 1
   nexthop via 80.0.0.2  dev p8p1 weight 1 linkdown
```

Why Link-State Matters — No change in default behavior

```
# ip route get 80.0.0.2
80.0.0.2 dev p8p1  src 80.0.0.1
    cache
# ip route get 100.0.0.2
100.0.0.2 via 70.0.0.2 dev p7p1  src 70.0.0.1
    cache
# ip route get 100.0.0.2
100.0.0.2 via 80.0.0.2 dev p8p1  src 80.0.0.1
    cache
```

Why Link-State Matters — Enable new sysctl for p8p1

```
# echo 1 > /proc/sys/net/ipv4/conf/p8p1/ignore_routes_with_linkdown
```

Why Link-State Matters — ‘linkdown’ routes are now also ‘dead’

```
# ip -4 route show
70.0.0.0/24 dev p7p1  proto kernel  scope link  src 70.0.0.1
80.0.0.0/24 dev p8p1  proto kernel  scope link  src 80.0.0.1 dead linkdown
80.0.0.0/24 via 90.0.0.2 dev p9p1  proto zebra  metric 20
90.0.0.0/24 dev p9p1  proto kernel  scope link  src 90.0.0.1
100.0.0.0/24
   nexthop via 70.0.0.2  dev p7p1 weight 1
   nexthop via 80.0.0.2  dev p8p1 weight 1 dead linkdown
```

Why Link-State Matters — connected route no longer used

```
# ip route get 80.0.0.2
80.0.0.2 via 90.0.0.2 dev p9p1  src 90.0.0.1
    cache
# ip route get 100.0.0.2
100.0.0.2 via 70.0.0.2 dev p7p1  src 70.0.0.1
    cache
# ip route get 100.0.0.2
100.0.0.2 via 70.0.0.2 dev p7p1  src 70.0.0.1
    cache
```

Why Link-State Matters — ‘linkdown’ routes are now also ‘dead’

```
# ip -4 route show
70.0.0.0/24 dev p7p1  proto kernel  scope link  src 70.0.0.1
80.0.0.0/24 dev p8p1  proto kernel  scope link  src 80.0.0.1 dead linkdown
80.0.0.0/24 via 90.0.0.2 dev p9p1  proto zebra  metric 20
90.0.0.0/24 dev p9p1  proto kernel  scope link  src 90.0.0.1
100.0.0.0/24
   nexthop via 70.0.0.2  dev p7p1 weight 1
   nexthop via 80.0.0.2  dev p8p1 weight 1 dead link-down
```

Why Link-State Matters

cumulus

Stop quagga

Why Link-State Matters — route from quagga disappears

```
# ip -4 route show
70.0.0.0/24 dev p7p1  proto kernel  scope link  src 70.0.0.1
80.0.0.0/24 dev p8p1  proto kernel  scope link  src 80.0.0.1 dead linkdown
80.0.0.0/24 via 90.0.0.2 dev p9p1  proto zebra  metric 20
90.0.0.0/24 dev p9p1  proto kernel  scope link  src 90.0.0.1
100.0.0.0/24
   nexthop via 70.0.0.2  dev p7p1 weight 1
   nexthop via 80.0.0.2  dev p8p1 weight 1 dead linkdown
```

Why Link-State Matters — 80.0.0.0/24 is unreachable

```
# ip route get 80.0.0.2
RTNETLINK answers: Network is unreachable
# ip route get 100.0.0.2
100.0.0.2 via 70.0.0.2 dev p7p1  src 70.0.0.1
    cache
# ip route get 100.0.0.2
100.0.0.2 via 70.0.0.2 dev p7p1  src 70.0.0.1
   cache
```

Why Link-State Matters

cumulus

Static configuration and status

Why Link-State Matters — Configuration stored in netconf

```
# ip -4 netconf
ipv4 dev lo forwarding on ... ignore_routes off
ipv4 dev ip6_vti0 forwarding on ... ignore_routes on
ipv4 dev sit0 forwarding on ... ignore_routes on
ipv4 dev ip6tnl0 forwarding on ... ignore_routes on
ipv4 dev p2p1 forwarding on ... ignore_routes off
ipv4 dev p7p1 forwarding on ... ignore_routes off
ipv4 dev p8p1 forwarding on ... ignore_routes on
ipv4 dev p9p1 forwarding on ... ignore_routes on
ipv4 all forwarding on ... ignore_routes on
ipv4 default forwarding on ... ignore_routes on
```

Why Link-State Matters — Configuration via sysctl

```
# cat /etc/sysctl.conf
# System default settings live in /usr/lib/sysctl.d/00-
system.conf.
# To override those settings, enter new settings here, or ...
#
# For more information, see sysctl.conf(5) and sysctl.d(5).
net.ipv4.conf.all.forwarding = 1
net.ipv6.conf.all.forwarding = 1

net.ipv4.conf.all.ignore_routes_with_linkdown = 1
net.ipv4.conf.default.ignore_routes_with_linkdown = 1
net.ipv4.conf.p2p1.ignore_routes_with_linkdown = 0
net.ipv4.conf.p7p1.ignore_routes_with_linkdown = 0
net.ipv4.conf.p8p1.ignore_routes_with_linkdown = 1
```

cumulus®

Future development plans?

Why Link-State Matters

cumulus

Consider not responding to IP address on down interface

Status: Not implemented

Why Link-State Matters

cumulus

Send netlink notifications for any route that gets linkdown or dead flag set

Status: Testing

Why Link-State Matters

cumulus

Test and ensure this is integrated with switchdev/forwarding offload layer

Status: Testing IPv4, but no IPv6 switchdev offload support today

cumulus®

Lessons Learned

Why Link-State Matters

cumulus

Below the netlink layer, IPv4 and IPv6 stacks are quite different

Why Link-State Matters

cumulus

Networking infrastructure sometimes takes more careful consideration than servers

cumulus®

Final Thoughts

Why Link-State Matters

cumulus

Suggestions for new features can come from a variety of places — keep your ears and eyes open

Thank You!