Bidirectional Forwarding Detection (BFD) implementation and support in OpenBSD

Peter Hessler
phessler@openbsd.org

OpenBSD

10 June, 2016

before bfd

- normally, you monitor the link state
- ...not always reliable
- sometimes there are active devices between you and your neighbor
- ...ixp switches
- ...long reach connect

what is bfd?

- bidirectional forwarding detection (RFC 5880)
  - detecting faults between two forwarding devices
  - kinda like gre-keepalives
  - protocol independent
  - ...commonly used with BGP
- bfd for ipv4 and ipv6 (single hop) (RFC 5881)
  - encapsulates bfd in a normal udp packet

what is bfd?

- bgp timers are generally 90 seconds
- how much traffic is that when you are sending 10Gbps?
- .... 100Gbps?

what is bfd?

- bgp timers are generally 90 seconds
- how much traffic is that when you are sending 10Gbps?
- .... 100Gbps?
- fastest possible is 3 seconds

what is bfd?

- found on big iron routers
- specs use microseconds!
- (μs not ms)
- ...implementation detail, we won't support more often than 50ms

bfd modes

- 'async/active' send keep alives
- ...bog standard
- 'demand' monitor traffic counters over the actual interface
- ...intimate knowledge of the dataplane counters
- ...if there isn't traffic within that timeframe, send a keepalive

specs can be stupid

`RFC 5881 - BFD for IPv4 and IPv6 (Single Hop)`

4. Encapsulation

BFD Control packets MUST be transmitted in UDP packets with destination port 3784, within an IPv4 or IPv6 packet. The **source port MUST be in the range 49152 through 65535**. The same UDP source port number MUST be used for all BFD Control packets associated with a particular session. **The source port number SHOULD be unique among all BFD sessions on the system**. If more than 16384 BFD sessions are simultaneously active, UDP source port numbers MAY be reused on multiple sessions, but **the number of distinct uses of the same UDP source port number SHOULD be minimized**. An implementation MAY use the UDP port source number to aid in demultiplexing incoming BFD Control packets, but **ultimately the mechanisms in [BFD] MUST be used to demultiplex incoming packets to the proper session**.

specs can be stupid

RFC 5880 - Bidirectional Forwarding Detection (BFD)

4.4. Keyed SHA1 and Meticulous Keyed SHA1 Authentication Section Format

Sequence Number

The sequence number for this packet. For Keyed SHA1 Authentication, this value is incremented occasionally. **For Meticulous Keyed SHA1 Authentication, this value is incremented for each successive packet transmitted for a session.** This provides protection against replay attacks.

current status

- minimal implementation (all of the MUSTs)
- can successfully negotiate against a Juniper MX-80 router
- basic configuration options are available
- basic logging
- route messages
- pf rules

current status

- part of the interface
- ...that was an initial idea, but turned out to be kinda dumb
- ...hard to adjust the interface state and still packets over it
- ...not to mention, more than one BFD peer on an interface
- ...almost the definition of the wrong place

current status

- partially moved to route
- ...we monitor nexthop, this makes sense
- difficult to adjust route UP/DOWN state for directly connected hosts
- ...punt for now
- special bfd flags (F/f)
- special route messages

current status

- panics in soreceive() after 8 hours!?!
- weirdness happens if I re-configure BFD
- only one peer at a time

Simple setup

```
$ ifconfig em1
em1: flags=408843<UP,BROADCAST,...> mtu 1500
        lladdr 00:25:90:0a:ea:cc
        index 2 priority 0
        bfd bfdsrc 203.0.113.1 bfddst 203.0.113.9
        bfd mode active min 1000ms rx 1000ms mult 3
        media: Ethernet autoselect (1000baseT full-duplex)
        status: active
        inet 203.0.113.1/24 ... broadcast 203.0.113.255
```

Simple setup

```
$ route -n show -inet
Routing tables

Internet:
Destination        Gateway            Flags     Prio Iface
default            172.16.255.1       UGS          8 em2
224/4              127.0.0.1          URS          8 lo0
127/8              127.0.0.1          UGRS         8 lo0
127.0.0.1          127.0.0.1          UHl          1 lo0
172.16.255/24      172.16.255.39      UC           4 em2
172.16.255.1       00:25:90:7f:7c:ac  UHLc         4 em2
172.16.255.39      00:25:90:0a:ea:cd  UHLl         1 em2
172.16.255.255     172.16.255.39      UHb          1 em2
192.0.2.1          192.0.2.1          UHl          1 lo1
203.0.113/24       203.0.113.1        UC           4 em1
203.0.113.1        00:25:90:0a:ea:cc  UHLl         1 em1
203.0.113.9        3c:8a:b0:8c:81:48  UHLcF        4 em1
```

Simple setup

```
cli> show bfd session extensive
                                 Detect   Transmit
Address      State  Interface    Time     Interval  Mult
203.0.113.1  Up     xe-0/0/0.0   3.000    1.000     3
 Client Static, TX interval 1.000, RX interval 1.000
 Session up time 01:24:50, previous down time 00:00:19
 Local diagnostic CtlExpire, remote diagnostic None
 Remote state Up, version 1
 Min async interval 1.000, min slow interval 1.000
 Adaptive async TX interval 1.000, RX interval 1.000
 Local min TX interval 1.000, min RX 1.000, mult 3
 Remote min TX interval 1.000, min RX 1.000, mult 3
 Local discriminator 16, remote discriminator 2669020539
 Echo mode disabled/inactive Session ID: 0x1

1 sessions, 1 clients
Cumulative transmit rate 1.0 pps, cumulative recv rate 1.0 pps
```

future plans

- fix ze bugs
- migrate from interface subsystem to route subsystem
- much better UI/UX
- multiple peers over the same interface
- ”encryption” support
- actual manipulation of route UP/DOWN state

future plans

- integrated knowledge in bgpd, ospfd, eigrpd, etc
- switchd, vxlan, etc
- draft-ymbk-idr-rs-bfd

Questions?