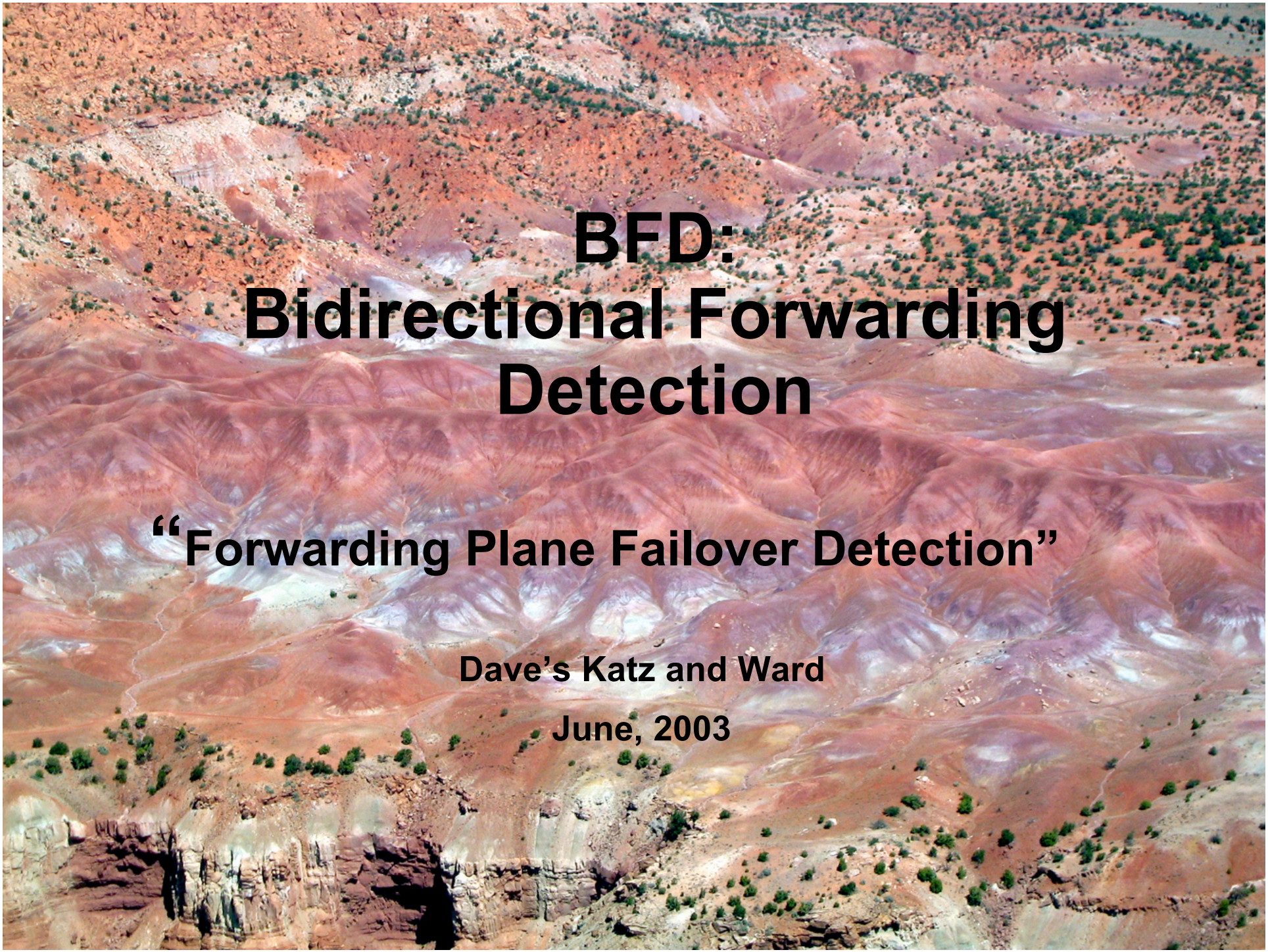


BFD: Bidirectional Forwarding Detection

"Forwarding Plane Failover Detection"

Dave's Katz and Ward

June, 2003

What is the problem?

- Convergence doesn't begin until detection of failure
- Can't use Media layer in all cases
- Can't detect FORWARDING PLANE failures
- Fundamental assumption of Graceful Restart mechanisms never tested
  - Assumption must be correct or GR fails
- Routing or signaling protocols have hello mechanisms
  - Too slow
  - Overloaded semantics of hello packets
  - Too costly to implement
  - Protocols with tweaked 'fast-hellos' not always in use

Other Proposed Solutions?

RSVP, ISIS, OSPF Fast-hellos

- High cost in terms of impact to router
  - Overload RP CPU, Large packets, extra information
- High level of false alarms
- Not able to handle x0ms detection times
- Each protocol has own mechanism
- Not all can handle unidirectional links

Other Proposed Solutions.2?

- PLP - Protocol Liveness Protocol
  - Offboard protocol hello mechanism
  - TLV structure - expensive to parse
  - Contained extra information about internal state of router
  - Contained information about init state of protocol

Other Proposed Solutions.3?

- MARP: Multiaccess Redundancy Protocol
  - TLV based - very flexible
  - Centralized and distributed components
  - Solves a different problem in a different environment
  - Still under Specification
  - For routers and Switches
    - Can be router to router

What is BFD?

Low-overhead, short-duration detection of failures in the path between adjacent forwarding engines, including the interfaces, data link(s), and forwarding planes

Implemented in forwarding engine of the system, where forwarding and control engines are separated

- Decouples the protocol from the fate of the RP and thus, very useful for GR verification

What is BFD.2?

Runs on top of any data protocol being forwarded between two systems

Unicast, point-to-point: Multicast doesn't help

Supports adaptive detection times to balance fast detection and network stability

- Three-way handshake to ensure both systems are aware of the change

Overview of BFD

Detect failures in communicating with a data plane

- NextHop

Fixed Field

Pair of systems send the message over each path between two systems

- Passed over whatever encapsulation is appropriate

Overview of BFD.2

Separate session for each communication and data path between two systems

Able to handle unidirectional links

Each system estimates how quickly they can send and receive packets in order to come to agreement about how rapidly a failure detection will take place

Also allows for fast systems to be on a shared medium w/ a slow system allowing for each system to detect failures to the best of their ability

What does it look like?

In the case of IPv4 or IPv6, BFD packets are transmitted with source and destination UDP port <TBD> in an IPv4 or IPv6 packet.

Packet:

- Version
- I Hear You
- Diagnostic code - why last session was reset
- Detection Multiplier
- My Discriminator
- Your Discriminator
- Suggested Min TX interval - Tell other end how fast you want to send packets
- Required Min RX interval - Tell other end how fast you are willing to receive

How is it lighter weight?

Once the parameters are negotiated, the values for subsequent generation of all packets to the remote peer is known

Once the parameters are negotiated, the values for subsequent reception of all packets from the remote peer is known

Thus, we can achieve x0ms detection time

Only have to pay attention when something changes

What are issues?

What is in SW and what is in HW?

- How fast can it go? Msec timers

Packet loss/delay

- Simple detection multiplier

Security

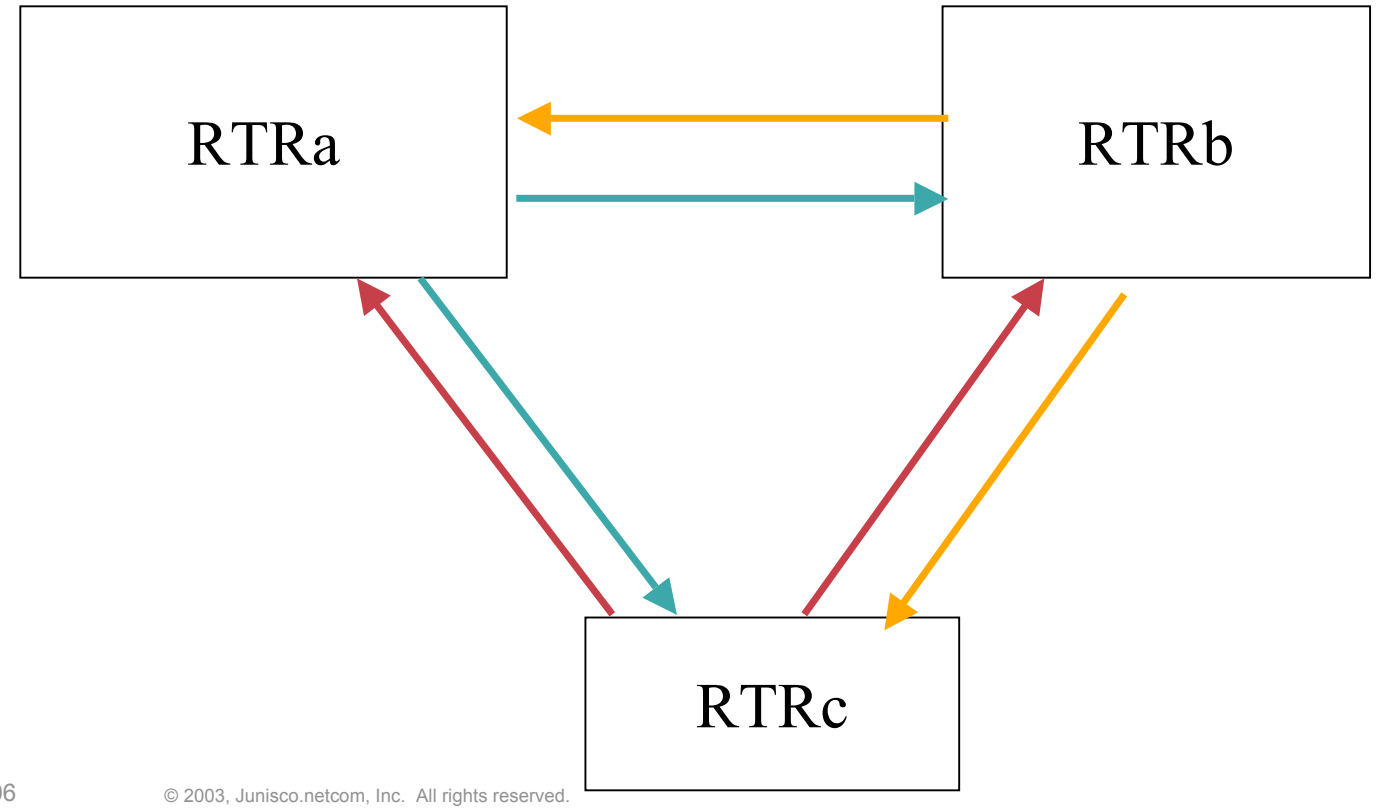
- Use TTL = well known value for directly adjacent routers
- If not directly adjacent, use addresses not routable outside the domain

What do we mean by forwarding plane?

- May appear to be a “Link Deadness Detector”
- Triggers Diagnostic and can signal forwarding plane down or reset
  - This can be due to disconnect w/ RP
  - Inability to update FIB
  - FIB hosed
  - Protocols crash and don’t reinit
  - We can have it triggered by any number of failures
- “I Hear You” field, if sent zero tell the remote end that you stopped receiving BFD packets from them.

Async mode:

Independent streams between devices
Control packets used for liveness detection

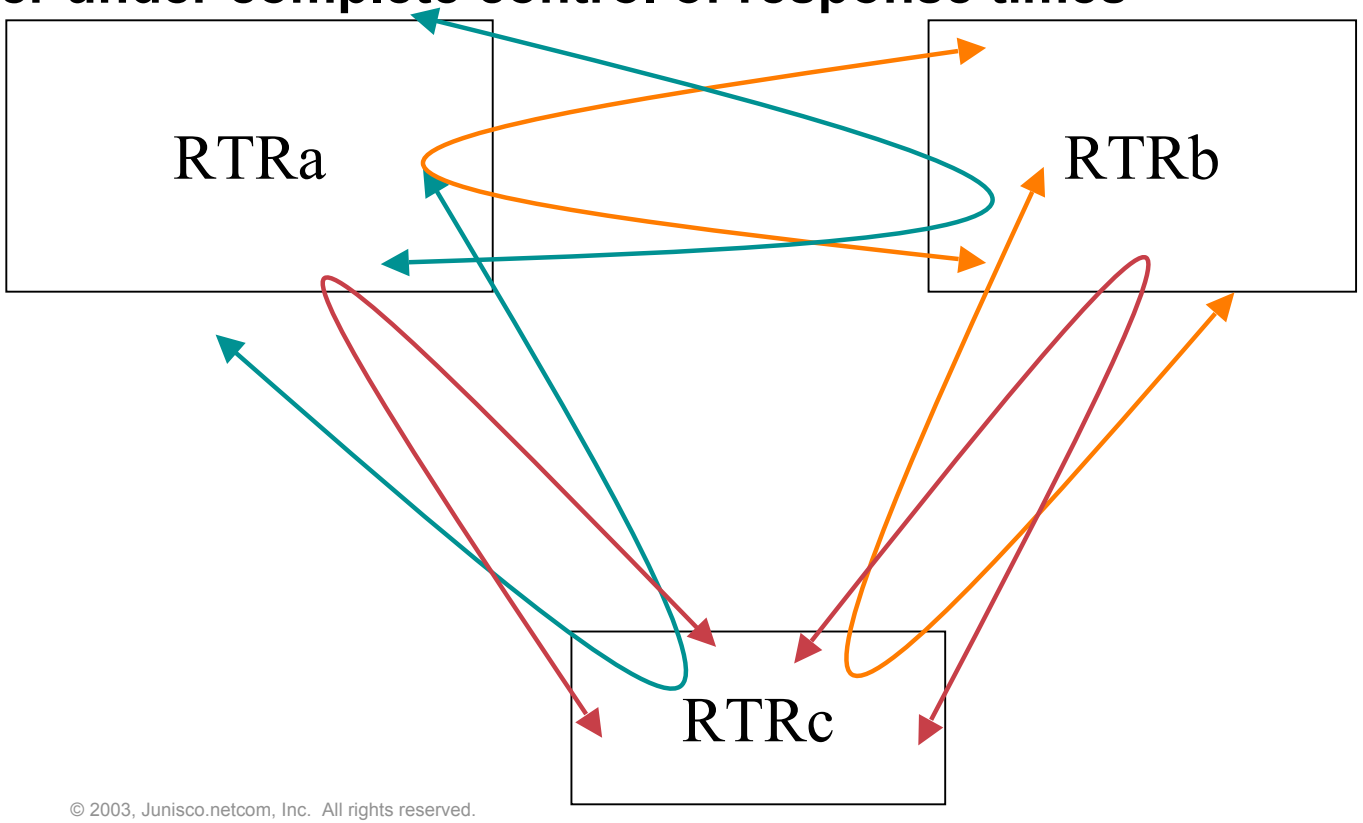


Echo Mode

Control packets only for parameter negotiation

Allows for very aggressive detection for future:

- Far end simply turns around echo packets in forwarding plane as ‘regular’ packet
- Sender under complete control of response times



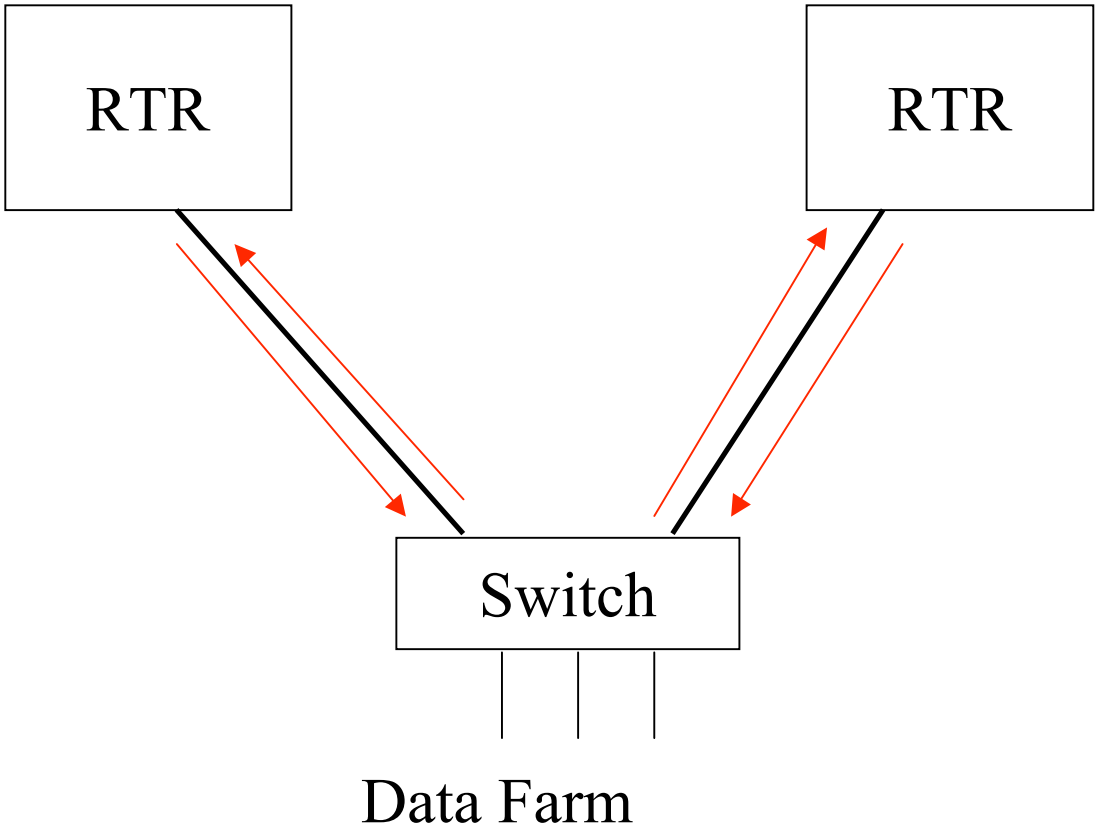
How does it handle undirectional links and node failure?

Packets demuxed independently of the interface from which they were rx'ed

Discriminator fields allow multiple sessions to be rx'ed on same interface

Problem detail for Ethernet Switches: one-way failure:

- Following figure illustrates a generic topology:



Problem Detail:

- If one of the uplinks, for one reason or another fails and goes unidirectional, the link does not always go to a “up/down” state on the router in a timely fashion
- The router may continue to advertise the link via IGPs, possibly “black-holing” traffic inbound to the affected network.
- Really, traditional “subnet reachability” problem for routers
- Note: cannot use echo mode with switches

What scenarios can it handle by running independent streams?

- **Per Address Family Detection**

  Runs over IPv4 or IPv6

- **Per VRF Detection**

  Can send to/thru packets in different VRFs

BFD Conclusions

Very Simple

Very fast

Easy implementation

Decouple from routing/signaling protocols

Bidirectional and unidirectional forwarding plane failure detection

Adaptive changes to protocol values to not reset any state machine

Works without upgrading software in Switches if between routers

Will Replace PLP - no changes to existing routing protocols

BFD Conclusions.2

Detects when the forwarding plane is belly-up