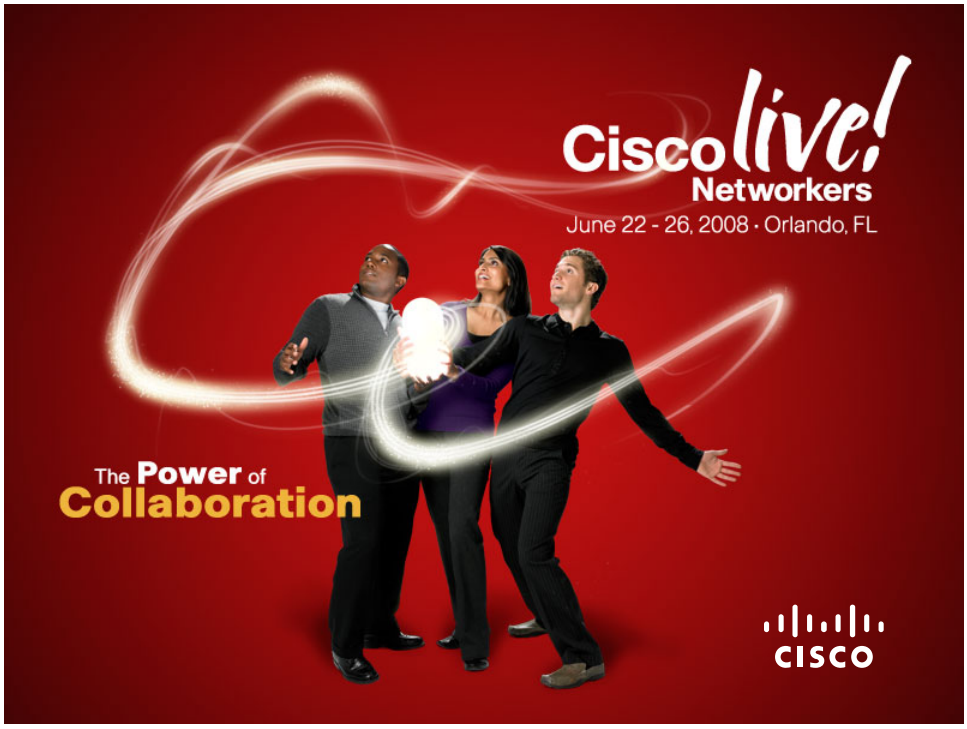
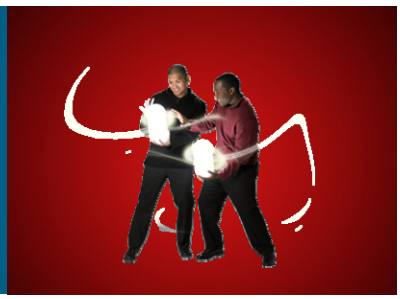


CISCO

Cisco live! Networkers

Routed Fast Convergence and High Availability



BRKRST-3363

Abstract

- As IP networks carry greater varieties of real-time traffic, the convergence speed and availability of the network becomes more critical. Many networks carrying voice and other real-time data must converge in less than three seconds to effectively carry traffic, and convergence times under one second are highly desirable or required in some situations. This session discusses various mechanisms network engineers can use to improve their network's convergence time and availability, including nonstop forwarding, tuning for fast convergence. This session also considers the tradeoffs involved in adding redundancy in terms of network convergence times.

Agenda

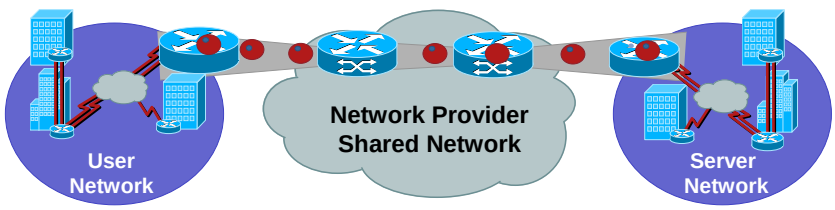
- High-Availability Overview
- IP Event Dampening
- Graceful Restart
- Fast Convergence
- IP Fast Reroute
- Operational Features
- Summary

Overview

Availability Definitions

Availability

- The probability that an item (or network, etc.) is operational, and functional as needed, at any point in time
- Or, the expected or measured fraction of time the defined service, device or area is operational; annual uptime is the amount (in days, hrs., min., etc.) the item is operational in a year



Availability Definitions

Availability

- Availability = (MTBF—MTTR)/MTBF

  Useful definition for theoretical and practical

- MTBF is mean time between failure

  What, when, why, and how does it fail?

- MTTR is mean time to repair

  How long does it take to fix?

What Is High Availability?

Availability	DPM	Downtime Per Year (24 x 365)		
99.000%	10000	3 Days	15 Hours	36 Minutes
99.500%	5000	1 Day	19 Hours	48 Minutes
99.900%	1000		8 Hours	46 Minutes
99.950%	500		4 Hours	23 Minutes
99.990%	100			53 Minutes
99.999%	10			5 Minutes
99.9999%	1			30 Seconds

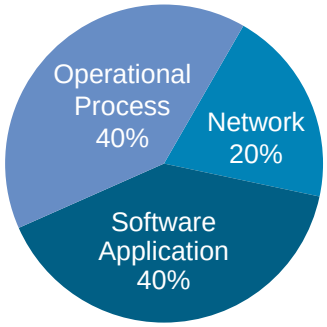
"High Availability"

DPM = Defects per Million (Hours of Running Time)

Downtime

Causes of Unscheduled Downtime

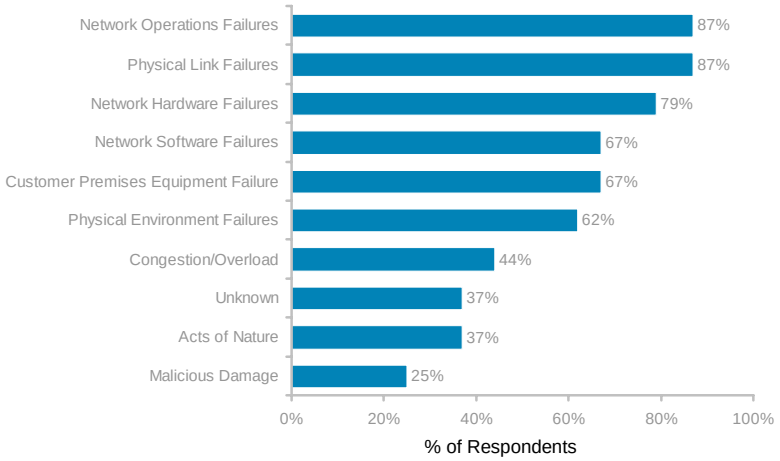
- Change
- Communication
- Process
- Design
- Hardware
- Software
- Link
- Power/env
- Resource utilization



Source: Gartner

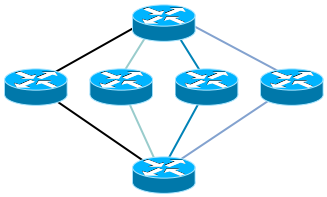
Downtime

Causes of Unscheduled Downtime



Source: Sage Research, IP Service Provider Downtime Study: Analysis of Downtime Causes, Costs and Containment Strategies, August 17, 2001, Prepared for Cisco SPLOB

Hardware Redundancy Options



	Highly Available Hardware		Network Replication of Hardware
–	Failover Redundant Modules Only	+	All Modules Are Redundant
	Operating System Determines Failover		Protocols Determine Failover
+	Typically Cost Effective	–	Increased Cost and Complexity
+	Often the Only Option at the Edge	+	Load Balancing

Highly Available Networks Tend to Have Both

The Culture of Availability

- Identify gaps
- Root cause failure analysis
- Availability modeling
- Availability metrics
- Priority and ROI analysis
- Quality improvement

The Culture of Availability

What's Your Availability Level?

- Analyze the gaps: reactive ~99%
- Few if any identified processes (except maybe to fix problems as reported by users)
- Significant number of SPFs
- Low tool utilization
- Low level of consistency (HW, SW, config, design)
- No quality improvement processes

The Culture of Availability

What's Your Availability Level?

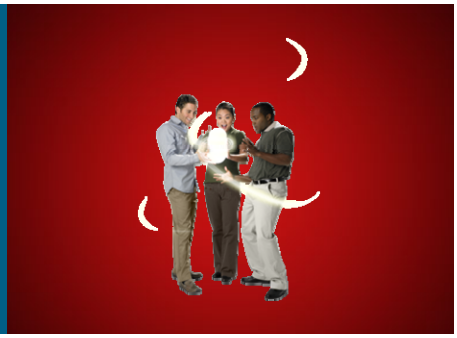
- Analyze the gaps: proactive ~99.9%
- Good change management processes including what-if analysis and change validation
- Low number of SPFs
- Fault and configuration management tools
- Improved consistency (HW, SW, config, design)
- Typically no quality improvement process

The Culture of Availability

What's Your Availability Level?

- Analyze the gaps: predictive ~99.99+%
- Consistent processes for fault, configuration, performance, and security
- No SPFs except at edge of network
- Fault, configuration, performance and workflow process tools
- Excellent consistency (HW, SW, config, design)
- HA culture of quality improvement

IP Event Dampening



IP Event Dampening

- Prevents routing protocol churn caused by constant interface state changes
- Supports all IP-routing protocols
  - Static routing, RIP, EIGRP, OSPF, IS-IS, BGP
  - In addition, it supports HSRP and CLNS routing
  - Applies on physical interfaces and can't be applied on subinterfaces individually
- Available in 12.0(22)S, 12.2(13)T

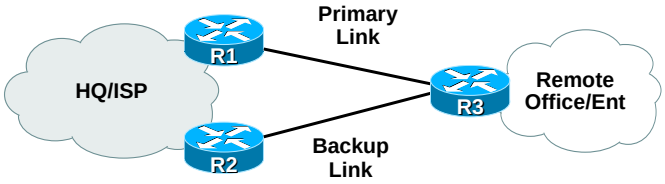
IP Event Dampening

Concept

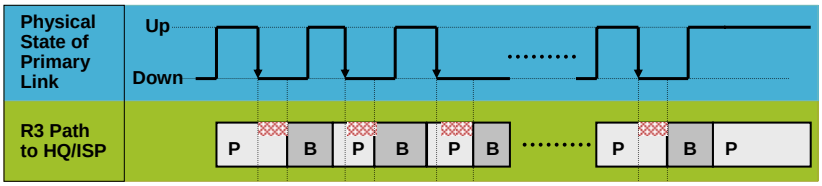
- Takes the concept of BGP route-flap dampening and applies it at the interface level, so all IP routing protocols can benefit
- Tracks interface flapping, applying a "penalty" to a flapping interface
- Puts the interface in "down" state from routing protocol perspective if the penalty is over a threshold tolerance
- Uses exponential decay algorithm to decrease the penalty over time and brings the interface back to "up" state

IP Event Dampening

Deployment

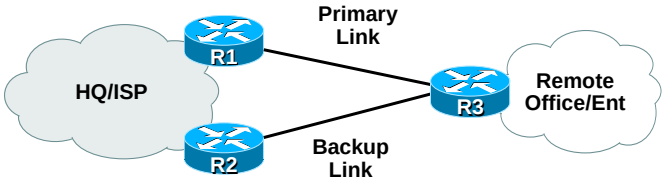


Link Flapping Causes Routing Reconvergence and Packet Loss

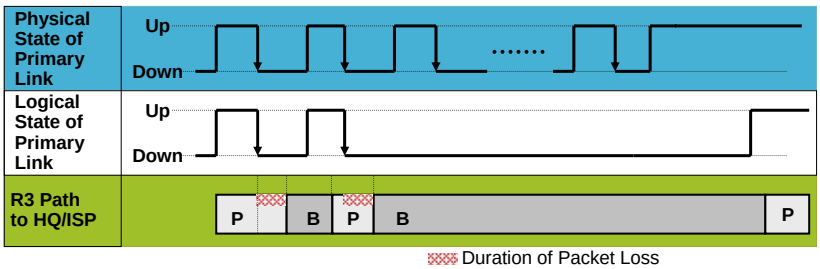


IP Event Dampening

Deployment



IP Event Dampening Absorbs Link Flapping Effects on Routing Protocols



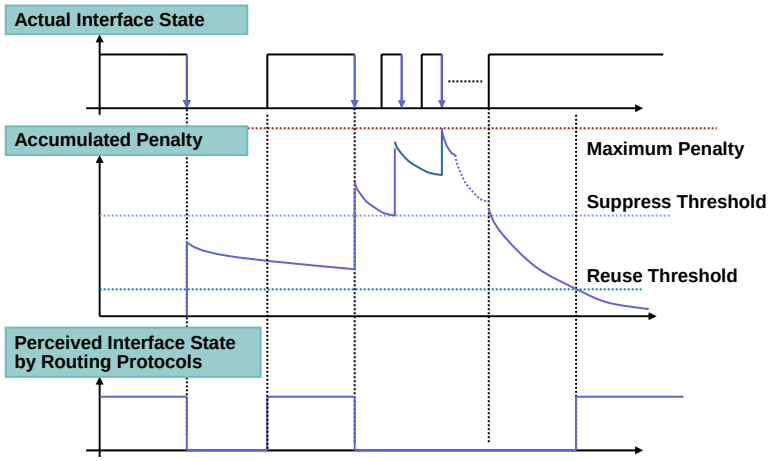
IP Event Dampening

Configuration

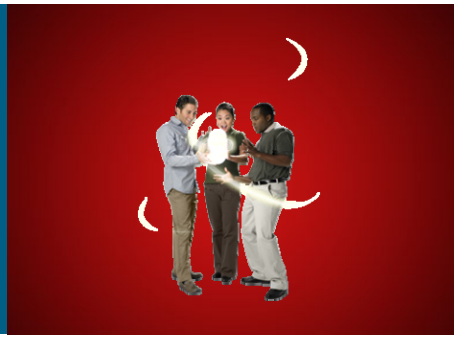
- `interface serial 0`
  `dampening [half-life] [reuse suppress max-suppress] [restart <penalty>]`
- Penalty: a numeric value applied to the interface each time it flaps
- Half-life: amount of time that must elapse without a flap to reduce penalty by half
- Suppress: if penalty exceeds this value, interface is suppressed from routing protocols' perspective
- Reuse: if penalty goes below this numeric limit, interface is reintroduced to routing protocols
- Max-suppress: maximum amount of time an interface can be suppressed
- Restart <penalty>: determines initial penalty (if any) to be applied to interface when system boots

IP Event Dampening

Algorithm

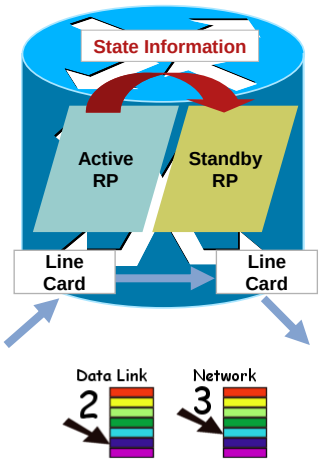


Graceful Restart



NSF/SSO

- Standby Route Processor (RP) takes control of router after a hardware or software fault on the active RP
- SSO allows standby RP to take immediate control and maintain connectivity protocols
- NSF continues to forward packets until route convergence is complete



NSF/SSO

Design Goals

- Provide a scalable solution

  Architecture must scale with workloads and features and meet network requirements

- Minimize state that must be synchronized

  Minimize impact of HA on service

- Detect and react to failures quickly

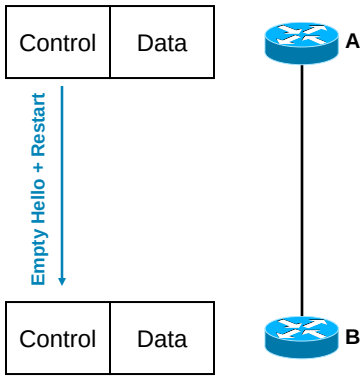
  Continuously monitor active components

  Continuously verify operation of standby components

Graceful Restart

OSPF

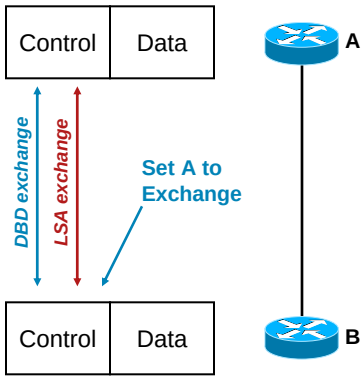
- OSPF uses an extension to the hello packets called link local signaling
- The first hello A sends to B has an empty neighbor list; this tells B that something is wrong with the neighbor relationship
- A sets the restart bit in its hello, which tells B that A is still forwarding traffic, and would like to resynchronize its database



Graceful Restart

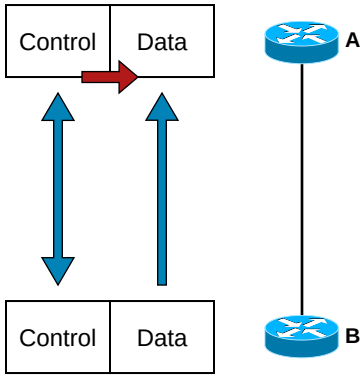
OSPF

- B moves A into the exchange state, and uses out of band signaling (OOB) to resynchronize their databases
- This process is the same as initial database synchronization, but it uses different packet types



OSPF GR/NSF Fundamentals

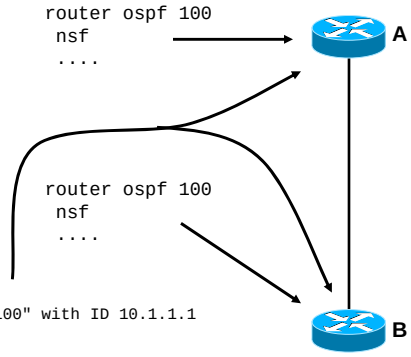
- When A and B have resynchronized their databases, they place each other in full state, and run SPF
- After running SPF, the local routing table is updated, and OSPF notifies CEF
- CEF then updates the forwarding tables, and removes all information marked as stale



Graceful Restart

OSPF

- Use the nsf command under the router ospf configuration mode to enable graceful restart
- Show ip ospf can be used to verify graceful restart is operational




```
router#sh ip ospf
Routing Process "ospf 100" with ID 10.1.1.1
....
Non-Stop Forwarding enabled, last NSF restart 00:02:06
ago (took 44 secs)
router#show ip ospf neighbor detail
 Neighbor 3.3.3.3, interface address 170.10.10.3
 ....
 Options is 0x52
 LLS Options is 0x1 (LR), last OOB-Resync 00:02:22 ago
```


Graceful Restart

OSPF

- Out-of-band resynchronization is described in draft-nguyen-ospf-oob-resync-00.txt

  http://www.ietf.org/internet-drafts/draft-nguyen-ospf-oob-resync-05.txt
- The link local signaling extensions to OSPF's hello packets are described in draft-nguyen-ospf-lls-00.txt

  http://www.ietf.org/internet-drafts/draft-nguyen-ospf-lls-05.txt
- The process of restarting using the above drafts is described in draft-nguyen-ospf-restart-00.txt

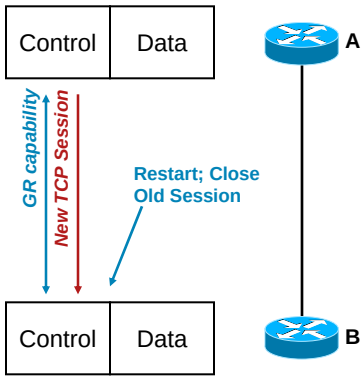
  http://www.ietf.org/internet-drafts/draft-nguyen-ospf-restart-05.txt
- OSPF graceful restart documentation:

  http://www.cisco.com/en/US/partner/products/sw/iosswrel/ps1839/products_feature_guide09186a0080153edd.html

Graceful Restart

BGP

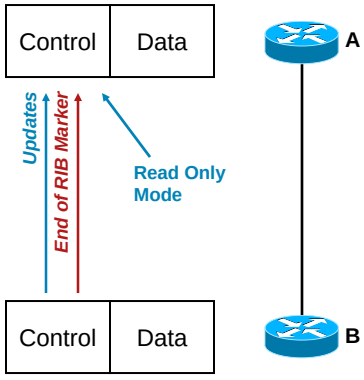
- When the BGP peering session is brought up, the graceful restart capability is negotiated. If both peers state they are capable of GR, it's enabled on the peering session
- When A restarts, it opens a new TCP session to B, using the same router ID
- B interprets this as a restart, and closes the old TCP session



Graceful Restart

BGP

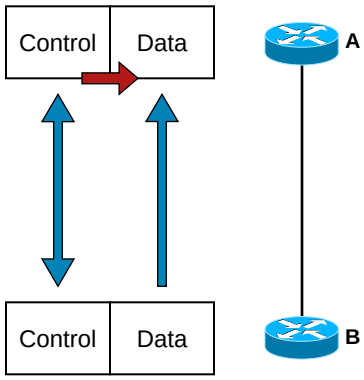
- B transmits updates containing its BGP table (it's local RIB out)
- A goes into read only mode, and does not run the bestpath calculations until its B has finished sending updates
- When B has finished sending updates, it sends an end of RIB marker, which is an update with an empty withdrawn NLRI TLV



Graceful Restart

BGP

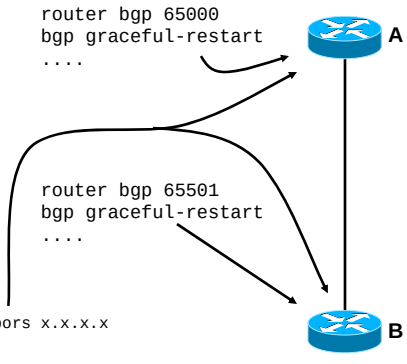
- When A receives the end of RIB marker, it runs bestpath, and installs the best routes in the routing table
- After the local routing table is updated, BGP notifies CEF
- CEF then updates the forwarding tables, and removes all information marked as stale



Graceful Restart

BGP

- Use the bgp graceful-restart command under the router bgp configuration mode to enable graceful restart
- Show ip bgp neighbors can be used to verify graceful restart is operational




```
router#show ip bgp neighbors x.x.x.x
....
Neighbor capabilities:
....
Graceful Restart Capabilty:advertised and received
Remote Restart timer is 120 seconds
Address families preserved by peer:
IPv4 Unicast, IPv4 Multicast
```

Graceful Restart

BGP

- BGP graceful restart is described in draft-ietf-idr-restart-06.txt

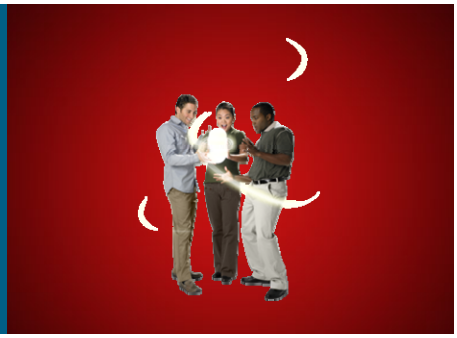
  http://www.ietf.org/internet-drafts/draft-ietf-idr-restart-10.txt

- Cisco's implementation of BGP graceful restart:

  http://www.cisco.com/en/US/partner/products/sw/iosswrel/ps1839/products_feature_guide09186a008015fede.html

  http://www.cisco.com/en/US/partner/tech/tk826/tk364/technologies_white_paper09186a008016317c.shtml

Fast Convergence



Network Convergence

- Network convergence is the time needed for traffic to be rerouted to the alternative or more optimal path after the network event
- Network convergence requires all affected routers to process the event and update the appropriate data structures used for forwarding

Network Convergence

Network Convergence Is the Time Required to:

- Detect event has occurred
- Propagate the event
- Process the event
- Update related forwarding structures

Event Detection

Subsecond Hellos

- At what frequency should hellos be issued?
  - How many interfaces involved?
  - What is the current resource utilization?
  - How fast does a change need to be detected?
- Are subsecond hellos the most effective method?
  - Will layer1/layer2 provide faster notification? (POS/serial)
  - Is MARP available?

Event Detection

OSPF Subsecond Hellos

- Supported: 12.0(23)S, 12.2(18)S, 12.2(15)T
- Operation:
  - DeadInterval—minimum one second
  - Hello multiplier is used to specify how many hellos to send within one second
  - HelloInterval will be advertised as zero second
- Configuration:
  - ip ospf dead-interval minimal hello-multiplier <3-20>

Event Detection

Subsecond Hello Issues

Scaling Is a Major Issue

300 Interfaces x 10 Neighbors/Interface = 3000 Neighbors

Three Hello Packets per Second on Each Interface

Router Has to Generate 900 Hellos per Second

3000 Neighbors Each Send Three Hellos per Second to this Router

Router Has to Accept and Process 9000 Hellos per Second

Router Has to Deal with 9900 Hellos per Second

One Hello Every 10,000th of a Second

Event Propagation

OSPF

- Initial LSA generation delay

  OSPF_LSA_DELAY_INTERVAL—500-ms delay

  Only router and network LSA generation delayed

- Recurring LSA origination delay

  MinLSInterval

  The minimum time between distinct originations of any particular LSA. The value of MinLSInterval is set to five seconds.

- LSA arrival throttling

  MinLSArrival

  “For any particular LSA, the minimum time that must elapse between reception of new LSA instances during flooding. LSA instances received at higher frequencies are discarded. The value of MinLSArrival is set to one second.”

Event Propagation

OSPF Exponential Backoff

- Fast LSA generation after initial event
- Repeated events increase regeneration delay
- Supported: 12.0(25)S, 12.2(18)S, 12.3(2)T
- Configuration:

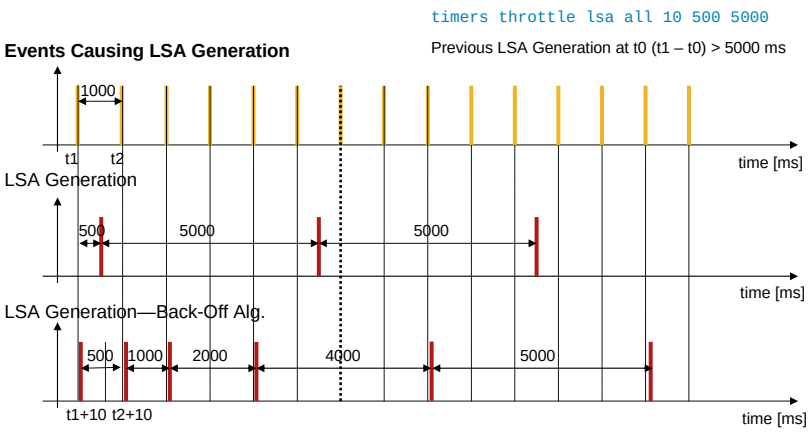
```
timers throttle lsa all  <lsa-start> <lsa-hold>
<lsa-max>
timers lsa arrival <timer>
```

All values are in ms

Note: MinLSArrival must be <= lsa-hold

Event Propagation

OSPF Exponential Backoff



Event Propagation

OSPF

- LSA has to be processed on each node
  - Detect if the LSA/LSP is newer/older
  - If the LSA/LSP is newer, detect if it carries any change
    - Number of links in the LSA/LSP (link comparison)
    - Size of the database (search)
  - If change detected schedule SPF (full/partial)
  - Install in the database
- Above processing time is rather small
  - Router-LSA with 10 links: 0.5 ms
  - Router-LSA with 100 links: 1 ms

Event Propagation

OSPF

- If LSA is declared newer, it's flooded over a certain set of interfaces
  - Excluding the neighbor from which the LSA/LSP has been received
- LSA is not flooded immediately
  - Link-state update packets are paced
  - Pacing timer is 33 ms by default (jittered by 10%)

Event Propagation

OSPF

- With default values and no retransmission each node can add 33-ms delay to the event propagation
- Supported: 12.2(4)T, 12.2(18)S, 12.0(25)S
- Configuration:

  Default values are 33 msec/66 msec

  ```
  timers pacing flood <timer>
  timers pacing retransmission <timer>
  ```


Event Processing

OSPF Exponential Backoff

- SPF-DELAY and SPF-HOLDTIME protect the router as the cost of convergence time
- Supported: 12.0(25)S, 12.2(18)S, 12.3(2)T
- Configuration:

  timers throttle spf <spf-start> <spf-hold> <spf-max>

  All values are in ms

Event Processing

Partial SPF

- Full SPF
  - Triggered by the change in router or network LSA
  - SPT tree is recomputed
  - All LSA types (type-1/2/3/4/5/7) are processed
- Partial SPF
  - Triggered by the change in type-3/4/5/7 LSA
  - If triggered by type-3/ all type-3 LSAs that contribute to the certain destination are processed
  - If triggered by type-5/7 all type-5/7 LSAs that contribute to the certain destination are processed
  - If triggered by type-4 all type-4 LSAs that announce a certain ASBR and all type-5/7 LSAs are processed

Event Processing

Partial SPF

- SPF calculation time
  - Full spf :
    - Depends on:
      - Number of nodes/links in the area
      - Number of Type-3/4/5/7 LSAs
    - Some experimental numbers (GSR/7500)
      - 50 nodes fully-connected topology ~ 10 ms
      - 100 node fully-connected topology ~ 25 ms
      - 500 nodes ~ 50 ms
      - 1000 nodes ~ 100 ms
  - Partial SPF:
    - Fast—less then 0.5 ms

Event Processing

SPF Triggers

- Router/network LSA triggers full SPF

  Some changes does not represent the topology change:

  Stub network UP/DOWN

  IP address change on link

  During the full SPF the whole SPT is rebuilt

  Change in the topology may not require the whole SPT rebuild

  Major part of the tree may stay the same in many cases

ISIS

- Prefix prioritization

  Four priorities: critical, high, medium, low

  /32 IPv4 and /128 IPv6 prefixes are classified by default in medium priority

  Rest is classified by default in low priority

- Customization

  `spf prefix-priority`

  This command supports prefix list for the first three priorities. The unmatched prefixes will be updated with low priority.

  As soon as the “prefix priority” command is used, then the /32 heuristic is no longer applied. If you then want to keep the /32s in medium, you need to configure the medium ACL so.

ISIS

- Prefix prioritization is **the** key behavior
  - **Critical**: IPTV SSM sources
  - **High**: most Important PEs
  - **Medium**: all other PEs
  - **Low**: all other prefixes
- Prefix prioritization customization is required for **critical** and **high**

ISIS: Prefix Priority Customization

```
ipv4 prefix-list isis-critical-acl
10 permit 0.0.0.0/0 eq 32
ipv4 prefix-list isis-high-acl
10 permit 0.0.0.0/0 eq 30
ipv4 prefix-list isis-med-acl
10 permit 0.0.0.0/0 eq 29
router isis 1
 address-family ipv4 unicast
   spf prefix-priority critical isis-critical-acl
   spf prefix-priority high isis-high-acl
   spf prefix-priority medium isis-med-acl
```

OSPF

Prefix Prioritization

- Four priorities: critical, high, medium, low
- /32 IPv4 and /128 IPv6 prefixes are classified by default in medium priority
- Rest is classified by default in low priority

Event Processing

Summary

- Set the SPF and PRC initial wait time to 1 ms
- SPF and PRC increment:

  Build a baseline of the time normally required to run SPF in the network; this will generally be around 50 ms

  Set the increment to this plus some padding, 5 to 10 ms
- SPF and PRC maximum wait time:

  If the normal SPF time is under 100 ms, set the maximum wait to one second

  If it's higher than 100 ms, set it to:
  (1000/S) x P = milliseconds

S = Normal SPF Time
P = Maximum Percentage of Processor Utilization for SPF

Event Processing

Summary

- Set the link-state generation initial wait time to 5 ms
  - The dampens some of the faster link flaps in the network.
  - Consider using IP event dampening to quell link flaps, as well
- Set the increment and the maximum wait times to the same values as you've set the SPF and PRC timers
  - No point in generating LSPs faster than the routers will actually process them!
- Tune carrier delay down to 0, IP event dampening will handled any instability from a flapping link
- Remember: exponential backoff is **not** dampening

Agenda

- BGP Fast Convergence
  - BGP Scanner
  - NHT—Next-Hop Tracking
  - FSD—Fast Session Deactivation
  - Event Driven Route Origination
  - MRAI—Min Route Advertisement Interval
  - TCP PMTU—Path MTU Discovery
  - Software Improvements

BGP Convergence

- BGP and IGP convergence tuning have a different focus
  - IGP convergence—rebuild the topology quickly following an event
  - BGP convergence—transfer large amounts of prefix information very quickly
- The magnitude of time involved is different
  - IGP—subsecond
  - BGP—seconds to minutes
- Fast IGP convergence plays a role in maintaining availability for BGP prefixes
  - Often topological changes can result in no BGP changes, the IGP updates the next-hop information for BGP prefixes

Faster Convergence

- Increased focus on faster BGP convergence
  - Critical for voice
  - VPN customers want IGP-like convergence
- Several factors influence BGP convergence
  - Detection of change
  - Propagation of information
  - Network topology and complexity
  - Network stability

Faster Convergence

- Typically two scenarios where we need faster convergence
- Single route convergence
  - A bestpath change occurs for one prefix
  - How quickly can BGP propagate the change throughout the network?
  - How quickly can the entire BGP network converge?
  - Key for VPNs and voice networks
- Router startup or “clear ip bgp *” convergence
  - Most stressful scenario for BGP
  - CPU may be busy for several minutes
  - Limiting factor in terms of scalability
  - Key for any router with a full Internet table and many peers

Convergence Basics—BGP Scanner

- BGP scanner plays a key role in convergence
- Full BGP table scan happens every 60 seconds
  - `bgp scan-time X`
  - Lowering this value is not recommended
- Full scan performs multiple housekeeping tasks
  - Validate nexthop reachability
  - Validate bestpath selection
  - Route redistribution and network statements
  - Conditional advertisement
  - Route dampening
  - BGP database cleanup
- Import scanner runs once every 15 seconds
  - Imports VPNv4 routes into vrfs
  - `bgp scan-time import X`

Convergence Basics—BGP Nexthops

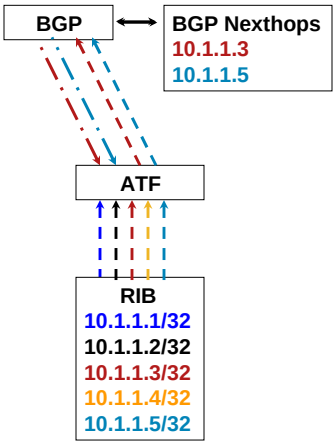
- Every 60 seconds the BGP scanner recalculates bestpath for all prefixes
- Changes to the IGP cost of a BGP nexthop will go unnoticed until scanner's next run
  - IGP may converge in less than a second
  - BGP may not react for as long as 60 seconds ☹
- Need to change from a polling model to an event driven model to improve convergence
  - Polling model—check each BGP nexthop's IGP cost every 60 seconds
  - Event driven model—BGP is informed by a third party when the IGP cost to a BGP nexthop changes

ATF—Address Tracking Filter

- ATF is a middle man between the RIB and RIB clients
  - BGP, OSPF, EIGRP, etc. are all clients of the RIB
- A client tells ATF what prefixes it is interested in
- ATF tracks each prefix
  - Notify the client when the route to a registered prefix changes
  - Client is responsible for taking action based on ATF notification
  - Provides a scalable event driven model for dealing with RIB changes

ATF—Address Tracking Filter

- BGP tells ATF to let us know about any changes to 10.1.1.3 and 10.1.1.5
- ATF filters out any changes for 10.1.1.1/32, 10.1.1.2/32, and 10.1.1.4/32
- Changes to 10.1.1.3/32 and 10.1.1.5/32 are passed along to BGP



NHT—Next Hop Tracking

- BGP next hop tracking
  - Enabled by default
  - 12.0(29)S, 12.3(14)T
  - `[no] bgp nexthop trigger enable`
- BGP registers all next hops with ATF
  - Hidden command will let you see a list of next hops
  - `show ip bgp attr nexthop`
- ATF will let BGP know when a route change occurs for a next hop
- ATF notification will trigger a lightweight “BGP scanner” run
  - Bestpaths will be calculated
  - None of the other “full scan” work will happen

NHT—Next Hop Tracking

- Once an ATF notification is received BGP waits five seconds before triggering NHT scan

  `bgp nexthop trigger delay <0-100>`

  May lower default value as we gain experience

- Event driven model allows BGP to react quickly to IGP changes

  No longer need to wait as long as 60 seconds for BGP to scan the table and recalculate bestpaths

  Tuning your IGP for fast convergence is recommended

NHT—Next Hop Tracking

- Dampening is used to reduce frequency of triggered scans
- `show ip bgp internal`

  Displays data on when the last NHT scan occurred

  Time until the next NHT may occur (dampening information)

- New commands

  `bgp nexthop trigger enable`

  `bgp nexthop trigger delay <0-100>`

  `show ip bgp attr next-hop ribfilter`

  `debug ip bgp events nexthop`

  `debug ip bgp rib-filter`

- Full BGP scan still happens every 60 seconds

  Full scanner will no longer recalculate bestpaths if NHT is enabled

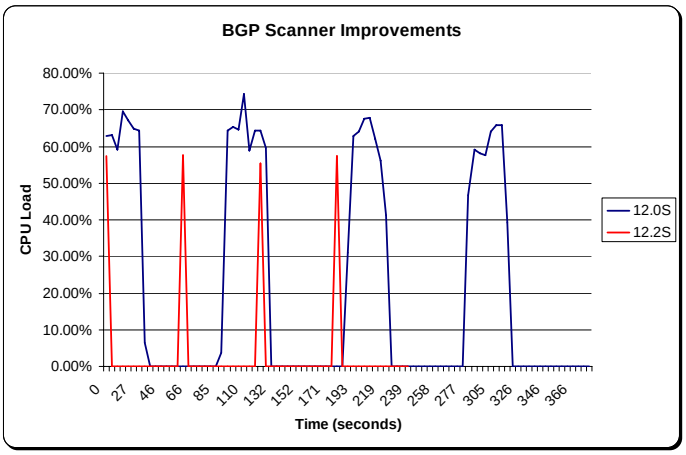
FSD—Fast Session Deactivation

- Register a peer's addresses with ATF
- ATF will let BGP know if there is a change in the route to reach the peer
- If we lose our route to the peer, tear down the session
  - No need to wait for the hold timer to expire!
- Ideal for multihop eBGP peers
- Very dangerous for iBGP peers
  - IGP may not have a route to a peer for a split second
  - FSD would tear down the BGP session
  - Imagine if you lose your IGP route to your RR (Route Reflector) for just 100 ms ☹
- Off by default
  - `neighbor x.x.x.x fall-over`
- Introduced in 12.0(29)S, 12.3(14)T

Event-Driven Route Origination

- Route origination was also based on a scanner dependant polling model
- Scanner traversed the RIB looking for routes that should be originated
- Traversing the RIB consumes a lot of CPU
- Route origination is now event driven
  - Scanner no longer checks the RIB for routes to redistribute
  - Route redistribution is event driven
  - Network statements are event driven
  - CPU impact of scanner is greatly reduced
- On by default, cannot disable
- Introduced in 12.2(28)S, 12.3(13)T via CSCef51906

Event-Driven Route Origination



- 7200 with NPE-G1
- 900k routes in the BGP table
- BGP scanner in 12.2S uses much less CPU

Scalability Update—Overview

- Bootup convergence and “clear ip bgp *” are the biggest challenges
  - Must converge all of our peers from scratch
  - BGP has to build and transmit a ton of data
- Multiple ways to improve bootup convergence and scalability
- Interface input queue drops
  - TCP acks can arrive in waves
  - Dropping a TCP ack is costly
  - If you are getting these drops, increase the size of your interface input queues
- TCP path-mtu-discovery
- Upgrade ☺

BGP Convergence

Peer Groups

- Peer groups are not just to simplify configuration, leading to the requirement for common outbound policy
- Update is formatted once for peer group leader, replicated for additional peers, provided they are in sync
- Update replication is much faster than update formatting
- 12.0(24)S provides support for dynamic update groups, which groups peers dynamically to provide the update replication

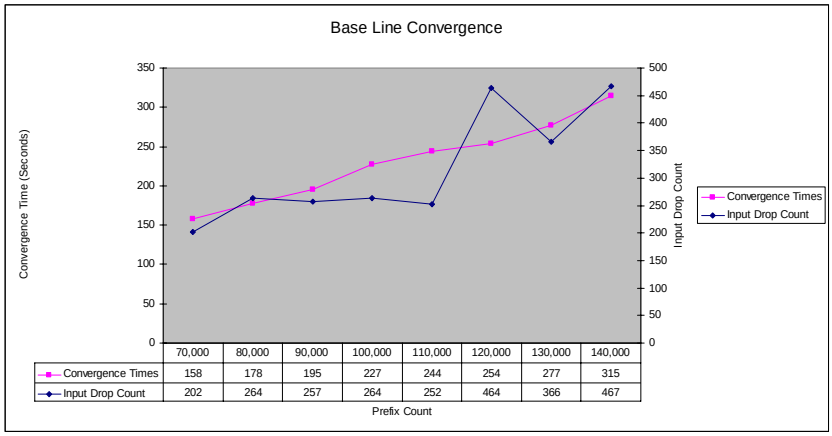
BGP Convergence

Peer Groups: Test

- 7206 VXR w/ NPE-300 and 256MB DRAM
- Cisco IOS 12.0(15)S1 and 12.0(23)S
- Single eBGP peering on which prefixes are received, then advertised over 50 iBGP sessions
- BGP is converged when table version for all peers is equal and the BGP InQ and BGP OutQ are 0
- Connectivity to all peers over same Fast Ethernet interface

BGP Convergence

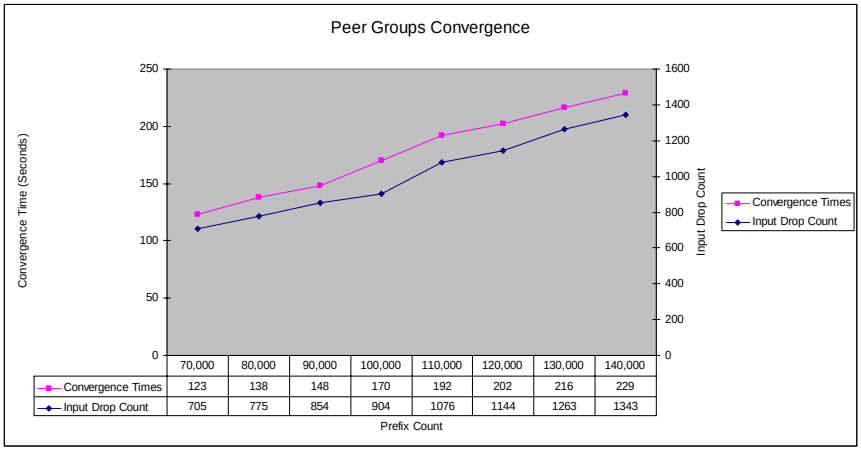
Peer Groups: Test



	70,000	80,000	90,000	100,000	110,000	120,000	130,000	140,000
Convergence Times	158	178	195	227	244	254	277	315
Input Drop Count	202	264	257	264	252	464	366	467

BGP Convergence

Peer Groups: Test

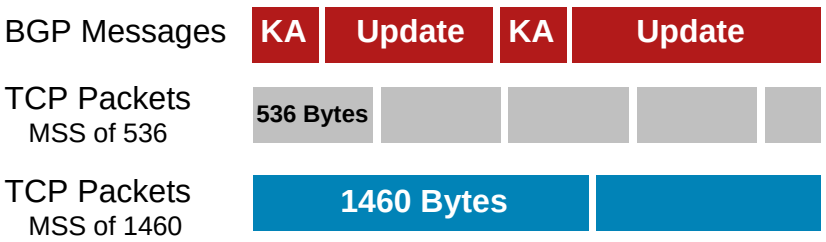


	70,000	80,000	90,000	100,000	110,000	120,000	130,000	140,000
Convergence Times	123	138	148	170	192	202	216	229
Input Drop Count	705	775	854	904	1076	1144	1263	1343

TCP Path MTU Discovery

- MSS (Max Segment Size)—limit on the largest segment that can traverse a TCP session
  - Anything larger must be fragmented and reassembled at the TCP layer
  - MSS is 536 bytes by default
- 536 bytes is inefficient for Ethernet (MTU of 1500) or POS (MTU of 4470) networks
  - TCP is forced to break large segments into 536-byte chunks
  - Adds overheads
  - Slows BGP convergence and reduces scalability
- "`ip tcp path-mtu-discovery`"
  - MSS = lowest MTU between destinations—IP overhead (20 bytes)—TCP overhead (20 bytes)
  - 1460 bytes for Ethernet network
  - 4430 bytes for POS network

TCP Path MTU Discovery

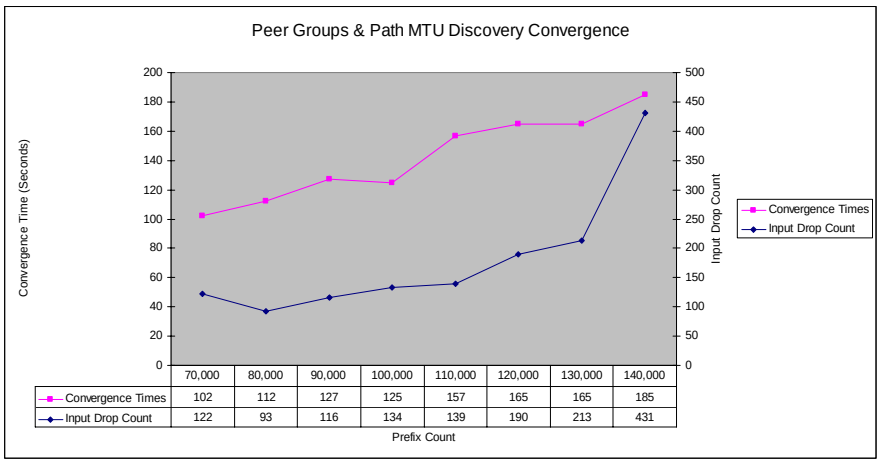


- BGP KAs (KeepAlives) are 19 bytes
- BGP updates vary in size up to 4096 bytes
- The larger the TCP MSS the fewer TCP segments required
- Fewer packets means less overhead and faster convergence
- New knob will allow you to enable/disable per peer

```
[no] neighbor x.x.x.x transport path-mtu-discovery
```

BGP Convergence

Peer Groups and PMTU: Test



	70,000	80,000	90,000	100,000	110,000	120,000	130,000	140,000
Convergence Times	102	112	127	125	157	165	165	185
Input Drop Count	122	93	116	134	139	190	213	431

BGP Convergence

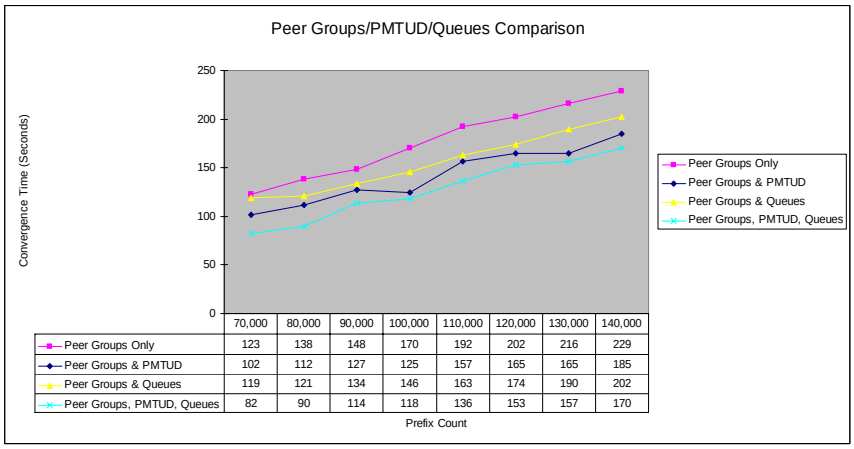
Packet Drops

- The use of peer groups greatly increases the rate at which the router can send BGP UPDATE messages
- The returning TCP ACKs can overflow the input hold queue, resulting in lost ACKs and TCP backoff
- Will result in peers losing sync with peer group Leader

$$\text{Hold Queue Size} = \frac{\text{Window Size}}{2 * \text{MSS}} * \text{Peer Count}$$

BGP Convergence

Peer Groups, PMTU, and Queue Tuning: Test



	70,000	80,000	90,000	100,000	110,000	120,000	130,000	140,000
Peer Groups Only	123	138	148	170	192	202	216	229
Peer Groups & PMTUD	102	112	127	125	157	165	165	185
Peer Groups & Queues	119	121	134	146	163	174	190	202
Peer Groups, PMTUD, Queues	82	90	114	118	136	153	157	170

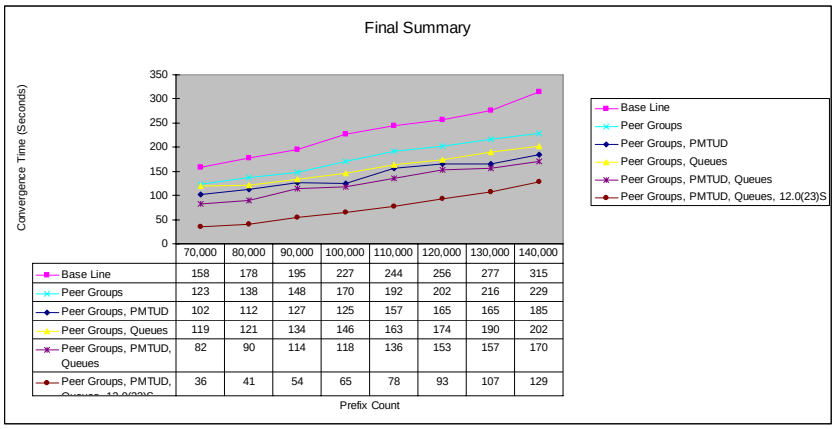
BGP Convergence

Update Packing

- BGP **updates** are based on a set of attributes and a list of prefixes that share that particular set of attributes
- Prior to 12.0(19)S, BGP **update** messages were not packed optimally
- Waiting until all prefixes are received from a peer prior to processing them and sending updates can further increase the ability to pack efficiently
- Fully supported in 12.0(23)S

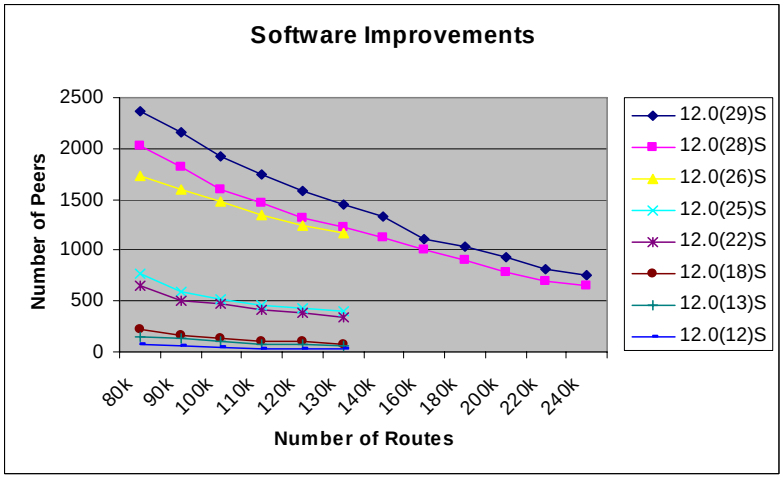
BGP Convergence

Complete Optimization Test



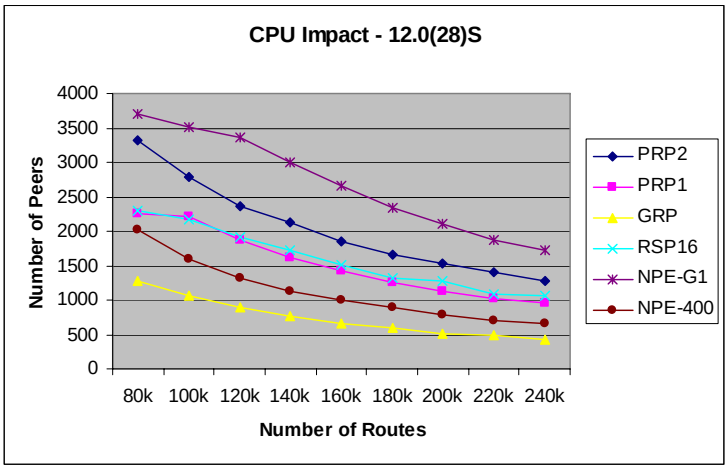
	70,000	80,000	90,000	100,000	110,000	120,000	130,000	140,000
Base Line	158	178	195	227	244	256	277	315
Peer Groups	123	138	148	170	192	202	216	229
Peer Groups, PMTUD	102	112	127	125	157	165	165	185
Peer Groups, Queues	119	121	134	146	163	174	190	202
Peer Groups, PMTUD, Queues	82	90	114	118	136	153	157	170
Peer Groups, PMTUD,	36	41	54	65	78	93	107	129

Scalability Update—Software



- 7200 with NPE-400

Scalability Update—Hardware

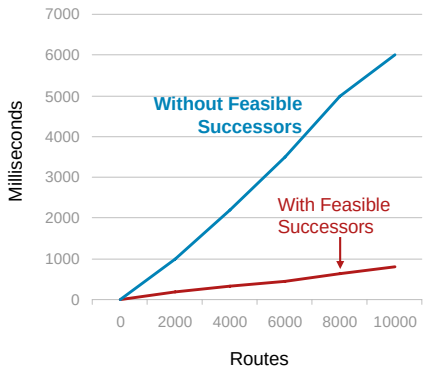


- CPU processing power plays a big role

EIGRP Fast Convergence

Feasible Successors

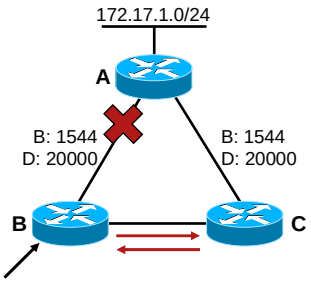
- Whether an alternate path is a feasible successor or not makes a large difference in convergence
- In this test, switching from the best path to a feasible successor takes less than one second; switching to some other neighbor takes about six seconds
- It's important to consider not only the best paths through an EIGRP network, but also the feasible successors



EIGRP Fast Convergence

Feasible Successors

- In this network, B and C have equal cost paths to A; neither one will see the other as a feasible successor because the feasible distance is equal to the reported distance
- If either link fails, at least one query/reply will be required to converge

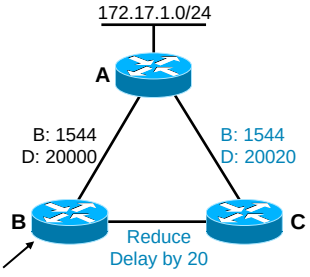



```
router-b#sho ip eigrp topo 172.17.1.0
IP-EIGRP (AS 100): Topology entry for 172.17.1.0/24
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is 2172416
  Routing Descriptor Blocks:
  172.17.2.1 (Serial0/0), from 208.0.8.4, Send flag is 0x0
      Composite metric is (2172416/18944), Route is Internal
      ....
  172.17.1.0 (Serial0/3), from 172.17.3.1, Send flag is 0x0
      Composite metric is (2684416/2172416), Route is Internal
      ....
```


EIGRP Fast Convergence

Feasible Successors

- Increasing the C to A metric, and decreasing the C to B metric by the same amount, will allow B to become C's feasible successor
- But this only works one way; there's no way to make B and C point at each other as feasible successors of each other



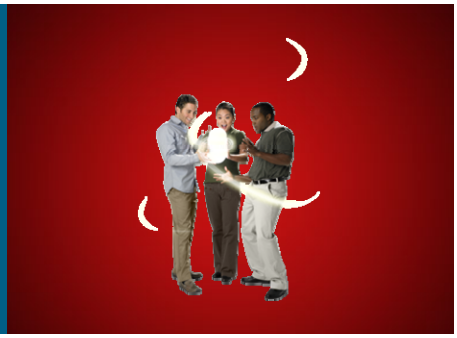

```
router-b#sho ip eigrp topo 172.17.1.0
IP-EIGRP (AS 100): Topology entry for 172.17.1.0/24
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is 2172416
  Routing Descriptor Blocks:
  172.17.2.1 (Serial0/0), from 208.0.8.4, Send flag is 0x0
      Composite metric is (2172416/18944), Route is Internal
      ....
  172.17.1.0 (Serial0/3), from 172.17.3.1, Send flag is 0x0
      Composite metric is (2684416/2167296), Route is Internal
      ....
```

EIGRP Fast Convergence

Feasible Successors

- Whether the next best path is considered loop free by EIGRP (a feasible successor) or not has a large impact on convergence times
- Don’t just consider the best path from every point in your network, but also the next best path
- Determine how best to set up your path metrics to improve convergence performance
- Always use the delay metric to engineer your routing, never the bandwidth metric!

IP Fast Reroute



Objective

- Provide fast reroute in pure IP networks and MPLS/LDP networks without deploying RSVP-TE
- To restore productive forwarding to all reachable addresses within 50 ms
- Control the transition of the network from repair to normal forwarding without further packet loss or microlooping

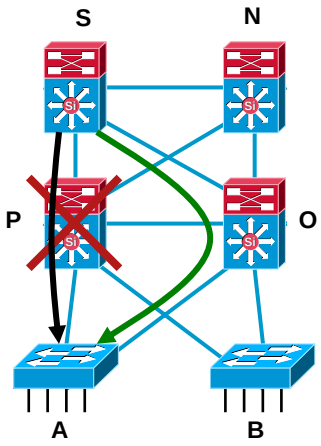
The Four Stages of IPFRR

1. Pre-computation of repair paths
2. Detection of failure (e.g., BFD)
3. Invocation of appropriate repair
4. Controlled reconvergence of network

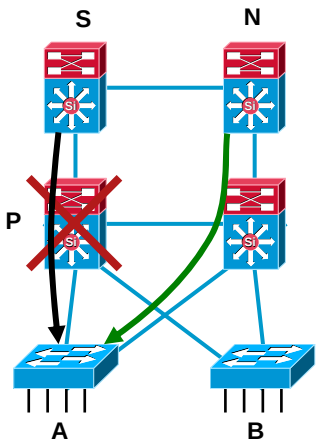
Basic Repair

- Uses ECMP and Loop Free Alternates (LFA) where available
- LFAs easily computed in OSPF and IS-IS
- Analogous to feasible successors in EIGRP

Triangle Topology—ECMP



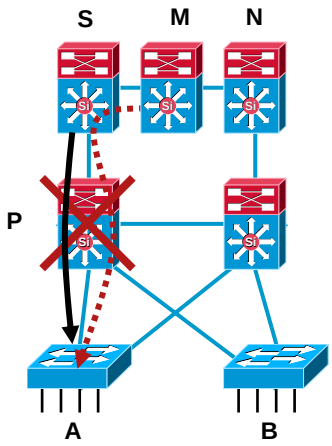
Square Topology—LFA



Basic Repair Properties

- In general topologies around 80% of failures allow **all** destinations to be repaired
- For 20%, only a subset of destinations can be repaired
  - These packets are dropped until convergence is complete
  - Loop free reconvergence mechanisms delay convergence (see later…)
  - Packet loss may be worse than conventional

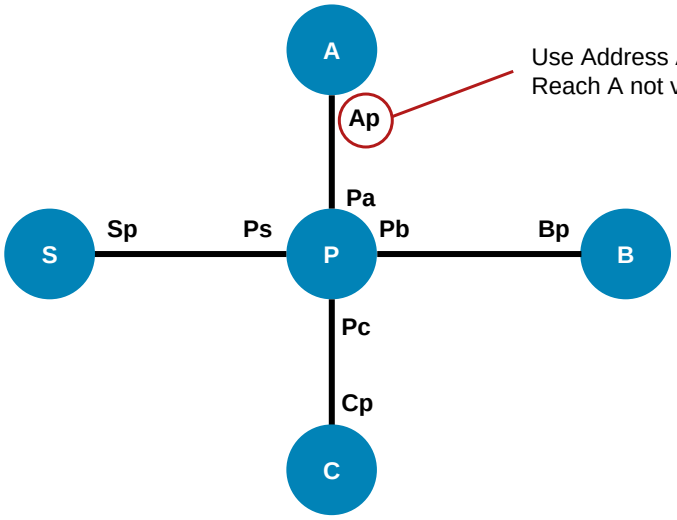
More Complex Topology—No LFA Available



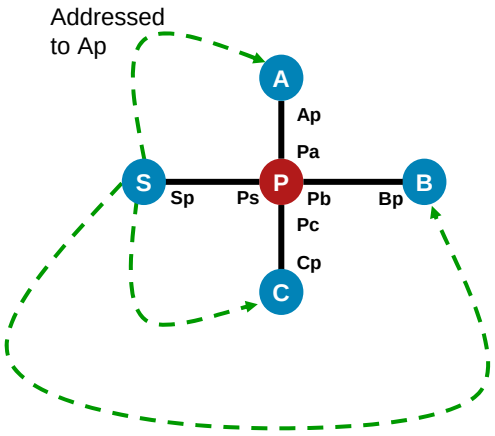
Not-via Overview

- Aim is to fix the remaining 20%
- When a failure occurs, the repairing router needs to get the packet to its destination not via the failure
- For each protected network component—link, node, etc., we calculate the required set of not-via paths avoiding the protected component.
- To repair, we encapsulate the packet to the not-via address on the far side of the failure

Not-via Addresses

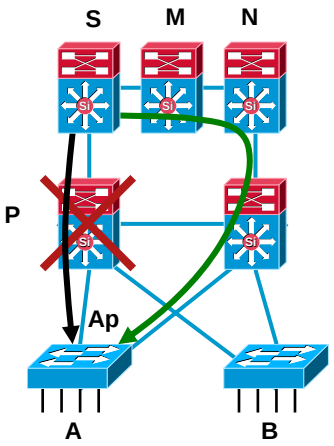


Not-via Repairs



All Repairs Take the Shortest Path from S to P's Neighbor not via P. This Will Be the Shortest Path from S to P's Neighbors After the Network Has Reconverged After Learning that P Has Failed.

Complex Topology—not-via



Tunnel to Ap
i.e., a not-via P

Not-via and Basic

- Not-via is a complete solution in its own right
- However amount of tunnel traffic can be dramatically reduced by using “basic” where LFA and ECMP paths exist
- Typically < 20% of traffic will require tunnels

Not-via Summary

- A intuitive approach that has 100% coverage of nonpartitioning faults
- Can repair abstract predefined fault groups
- Uses existing MPLS FRR hardware, or single level IP encapsulation

Microloops

- Fast-reroute prevents all packet loss once a failure has been detected
- **But** packets can still be lost due to microlooping when the network reconverges

Micro-Loops Are Bad

- Whenever a network reconverges microloops may form
- Microloops result in collateral damage to traffic not affected by the change, as well as causing the affected traffic to be lost
- Microloop damage has always been accepted as a necessary evil of the routing convergence process

FIB Update Order

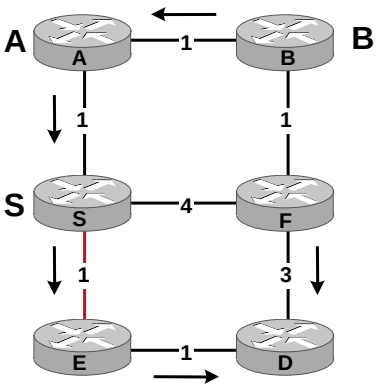
- Micro-loops are caused when router FIBs are updated in the “wrong” order
- The “natural” order for failure events is the wrong order
- Implementation specific factors will affect the exact order, but most failure events will cause loops

“Good News” Events

- The natural order for good news events is such that loops should not occur, but implementation factors can still result in loops
- These loops during “benign” events are particularly annoying!
- E.g., bring up a new link and packets get dropped

What Are Micro-Loops?

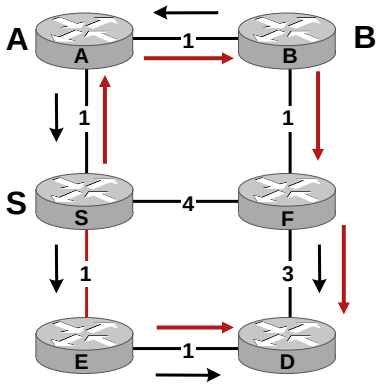
Paths **Before** Failure of S-E



What Are Micro-Loops?

Paths to D **After** Failure of S-E

Paths to D **Before** Failure of S-E



If A Changes Before B, We Have A Loop Across A-B Until B Changes

Note that We Have a Potential Loop Anywhere that the Red and Black Arrows Are in Different Directions

What Causes Micro-Loops?

- Micro-Loops can form during **any** topology change
  - Link/router down
  - Link/router up
  - Link cost change
- Not just failures
- Planned operational “management” changes can cause Micro-Loops and hence packet loss

Ordered FIB changes

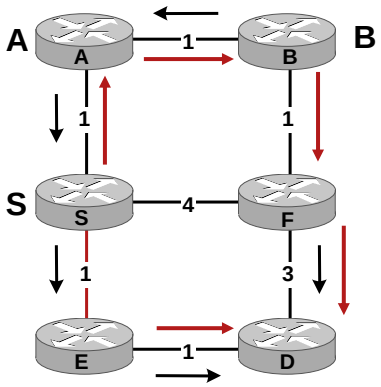
- For any isolated link/node change
- Determine “safe” ordering for FIB installation

  Bad news: update from edge to failure,

  Good news: update from change to edge
- Each router computes its “rank” with respect to the change
- Delays for a number of worst-case FIB compute/install times proportional to its rank

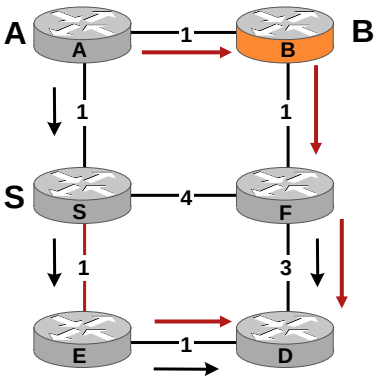
Ordered Change Example

- Ensure the changes are in the order B,A,S



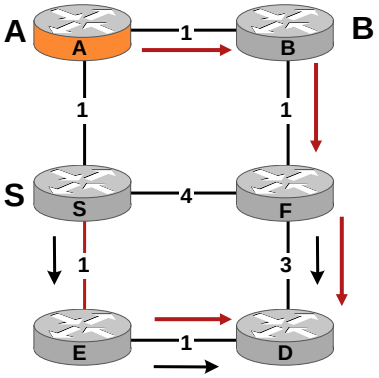
Ordered Change Example

- Ensure the changes are in the order B,A,S



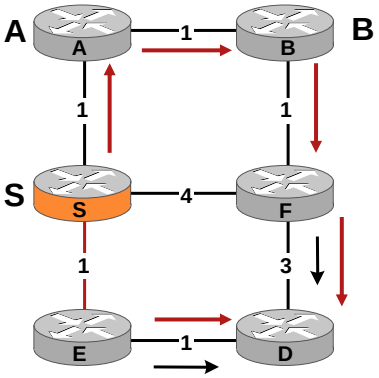
Ordered Change Example

- Ensure the changes are in the order B,A,S



Ordered Change Example

- Ensure the changes are in the order B,A,S



Ordered SPF Properties

- No forwarding changes required
- Complete prevention of loops for isolated node or link changes
- Requires cooperation from all routers
- Delay is proportional to network diameter
  - May delay reconvergence for tens of seconds (unless optional signalling used)

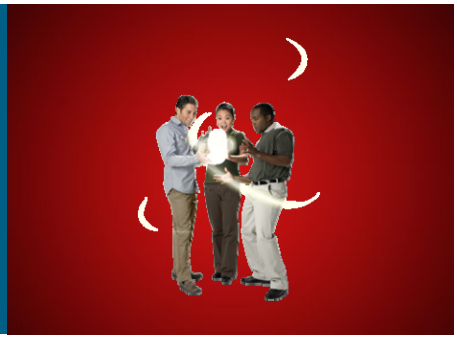
Signalling Optimization to Reduce delay

- Without signalling per router delay **must** be worst case
  - In many cases, actual delay is 0 because no change needed
- Signal to parents in rSPF when
  - Nothing to do, or
  - Completed FIB changes
- Can change FIB when received signal from all children (or when delay expires)
- Reduces controlled convergence time to subsecond
- Only an optimization
  - Loss of signals falls back to delay based

Applicability

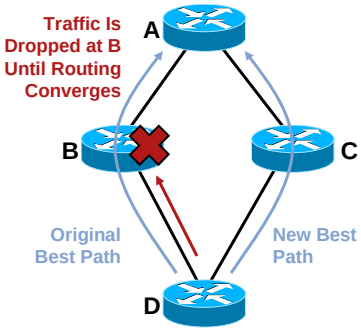
- Technique has wider applicability than just IPFRR transition
  - MPLS TE one hop tunnel
  - Network management
    - Link add, delete, modify cost
    - Router reload, reconfig, hardware swap
    - Virtual topology changes
    - Etc.

Operational Features



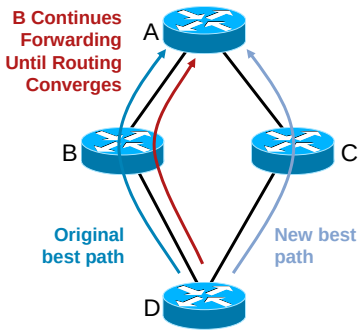
Graceful Shutdown

- You want to bring B down for maintenance; the routing protocol will reroute around B
- The packets “in flight” will be lost when B is taken off line, though—and this could be a lot of packets, if these are high-speed links
- It’s better to get A and D to route around B while B can still forward traffic, so B can be taken off line gracefully



Graceful Shutdown

- Graceful shutdown will allow the routing protocols to tear their adjacencies down without impacting the forwarding tables for some short period of time
- Once the protocol has torn its adjacencies down, it will then clean up the forwarding tables



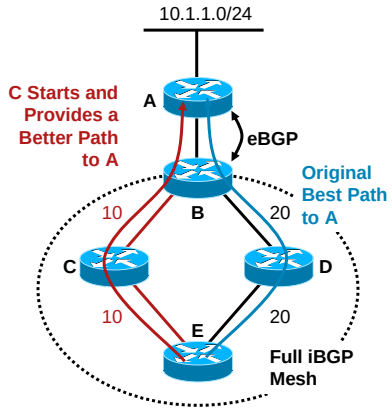
Graceful Shutdown

	Protocol Specific Signaling
BGP	CEASE NOTIFICATION
EIGRP	UPDATE with INIT-Bit Set
ISIS	LAN Interfaces: IIH with an Empty “IS Neighbors” TLV Point-to-Point Links: “p2p Adjacency State” TLV Set to “Init”
OSPF	Hello with an Empty Neighbor List

Graceful Shutdown Is Currently Under Development!
IS-IS Supported: 12.0(27)S

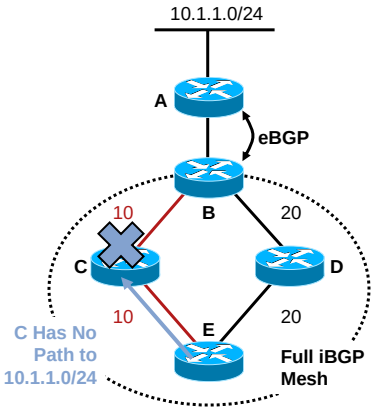
Wait for BGP

- E is learning 10.1.1.0/24 through iBGP from D with a next hop of A
- E examines the path to A, and finds an IGP route through D to A. It installs this route in the routing table.
- C is now inserted into the circuit; after a few seconds, the IGP has converged, and E now chooses C as the best path to A



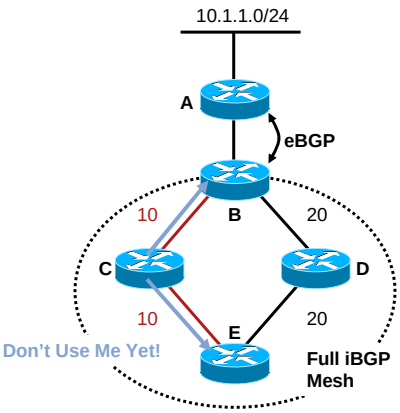
Wait for BGP

- However, BGP takes much longer to converge if C is accepting full routes (about 150,000 routes) from A; at least five minutes
- When E forwards packets to C for 10.1.1.1, C hasn't finished building its BGP tables, so it doesn't know how to reach this destination
- C drops the packets



Wait for BGP

- Instead, once the IGP has converged, C signals its IGP neighbors that they should not route this direction
- The IGP remains in this state until BGP notifies the IGP it has converged
- E will continue using D as its best path to A, even though a better one is available, until BGP converges on C



Wait for BGP

- OSPF uses max-metric router-lsa on-startup wait-for-bgp to configure this feature

  Available in 12.2T

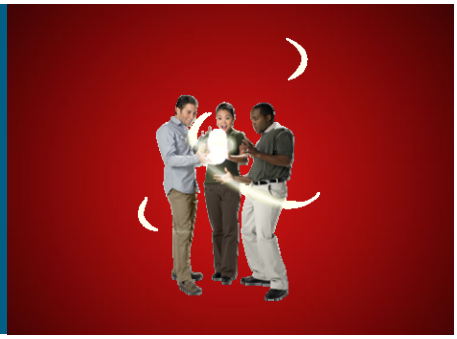
  http://www.cisco.com/en/US/partner/products/sw/iosswrel/ps1839/products_feature_guide09186a0080087c09.html

- IS-IS uses set-overload-bit on-startup wait-for-bgp to configure this feature

  Available in 11.3

  http://www.cisco.com/en/US/partner/tech/tk472/tk474/technologies_tech_note09186a00800a4bb1.shtml

Summary



Getting to 4 Nines

Roadblocks

- Single point of failure (edge card, edge router, single trunk)
- Outage required for hardware and software upgrades
- Long recovery time for reboot or switchover
- No tested hardware spares available on site
- Long repair times due to a lack of troubleshooting guides and process
- Inappropriate environmental conditions

Getting to 5 Nines

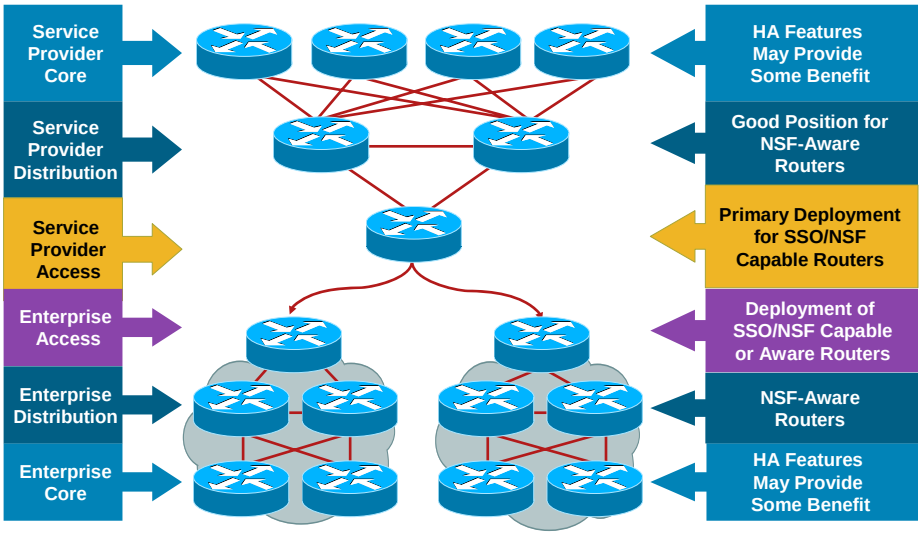
Roadblocks

- High probability of redundancy failure (failure not detected—redundancy not implemented)
- High probability of double failures
- Long convergence time for rerouting traffic around a failed trunk or router in the core
- Rely on manual operations

Network Design Conclusion

- Reduce complexity, increase modularity and consistency
- Consider solution manageability
- Minimize failure domain size (reduce single points of failure)
- Consider control plane resource requirements and the affect of busy CPU/memory
- Consider protocol attributes
- Consider budget, requirements, areas of network contributing the most downtime or at the greatest risk
- Test, test, test before deployment

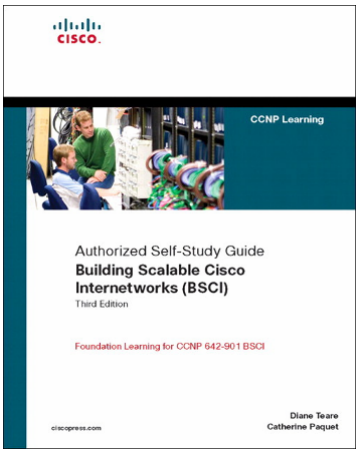
NSF/SSO: Deployment Strategies



Q and A

Recommended Reading

- Continue your Cisco Live learning experience with further reading from Cisco Press
- Check the Recommended Reading flyer for suggested books



Available Onsite at the Cisco Company Store

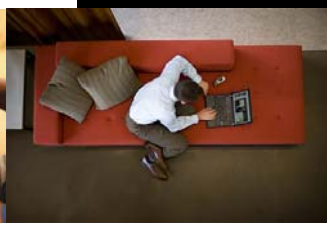
Complete Your Online Session Evaluation

- Give us your feedback and you could win fabulous prizes. Winners announced daily.
- Receive 20 Passport points for each session evaluation you complete.
- Complete your session evaluation online now (open a browser through our wireless network to access our portal) or visit one of the Internet stations throughout the Convention Center.

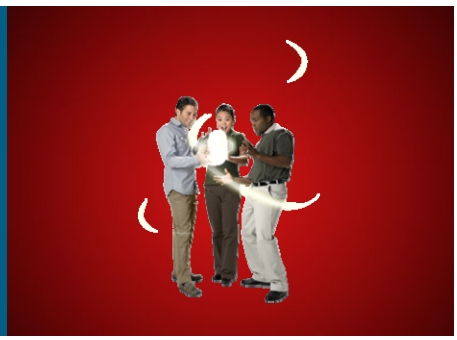
Don't forget to activate your Cisco Live virtual account for access to all session material on-demand and return for our live virtual event in October 2008.

Go to the Collaboration Zone in World of Solutions or visit www.cisco-live.com.





Backup Slides/ Extra Material



NSF/SSO

Enabling SSO

- Perform this step on Cisco 7500 series devices only

```
router(config)#hw-module slot slot-number image file-spec
```

slot-number—specifies the active RSP slot where the flash memory card is located

file-spec—Indicates the flash device and the name of the image on the active RSP

Repeat command for standby RSP

- Enter redundancy configuration mode and set the redundancy configuration mode to SSO on both the active and standby RP

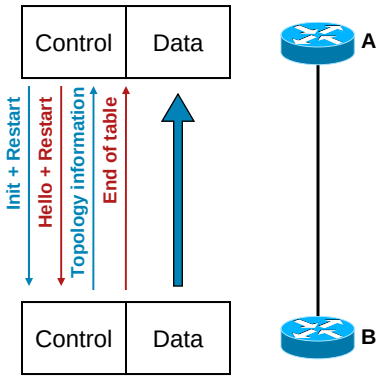
```
Router(config)#redundancy
Router(config-red)#mode sso
```

Note: standby will reset after this command

Graceful Restart

EIGRP

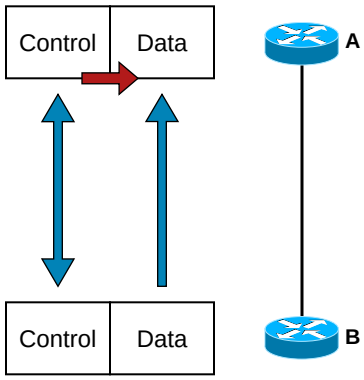
- The signal in EIGRP is an update with the initialization and restart (RS) bits set
- A sends its hellos with the restart bit set until GR is complete
- B transmits the routing information it knows to A
- When B is finished sending information, it sends a special end of table signal so A knows the table is complete



Graceful Restart

EIGRP

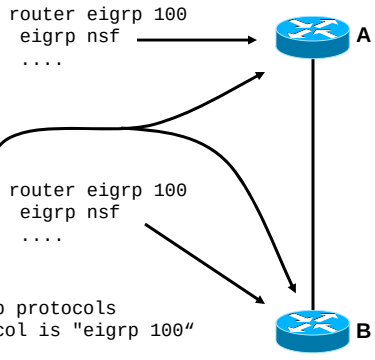
- When A receives this end of table marker, it recalculates its topology table, and updates the local routing table
- When the local routing table is completely updated, EIGRP notifies CEF
- CEF then updates the forwarding tables, and removes all information marked as stale



Graceful Restart

EIGRP

- Use the eigrp nsf command under the router eigrp configuration mode to enable graceful restart
- Show ip protocols can be used to verify graceful restart is operational
- http://www.cisco.com/en/US/products/sw/iosswrel/ps1839/products_feature_guide09186a008080160010.html




```
router#show ip protocols
Routing Protocol is "eigrp 100“
....
Redistributing: eigrp 100
EIGRP NSF-aware route hold timer is 240s
Automatic network summarization is in effect
Maximum path: 4
....
```

Event Processing

Incremental SPF Overview

- Incremental SPF

  Modified Dijkstra algorithm

  Keep the unchanged part of the tree

  Rebuild only the affected parts of the tree

  Reattach the affected parts of the tree to the unchanged part of the tree

Event Processing

Incremental SPF Overview

- Analyse the changes in the newly received LSA

  all new or changed LSAs received during the spf-wait interval are put in a NEW_LSA_LIST

- LSA can carry:

  Good news—a better path to the node becomes available

  Bad news—current best path to the node becomes worse (or is lost)

  No news—no topological related change

- Based on the type of the news algorithm decides what to do

Event Processing

Incremental SPF Overview

- Once the incremental SPF analyzes all new LSAs received during the wait-interval a standard Dijkstra will be executed
  - iSPF populates the CANDIDATE_LIST (TENTATIVE_LIST)
  - If the CANDIDATE_LIST is empty – no work to do
- If no change in topology (stub link changed), CANDIDATE_LIST would be empty
  - Stub(s) would be recalculated and some route(s) may need to be updated

Event Processing

Incremental SPF Overview

- Gain of iSPF depends on how far (topologically) the change happens from the calculating node
- If the change affects only a small part of Shortest Path Tree (SPT), gain is significant
  - We were able to run SPF and update the RT for the 1000 node network in less then 10 ms
- If the change is close to the calculating node it is likely are larger portion of the SPT will be affected, reducing the impact of iSPF

Event Processing

Incremental SPF Overview

- There are always nodes closer to the topology change and nodes that are more remote
- Flooding takes some time—nodes that are most remote from the change are usually notified last
- If full SPF runs on all nodes regardless of the change, then routers notified last about it will converge last (giving that it takes same amount of time to run SPF on each node)
- With iSPF, the more remote the node is from the change, less work it needs to do during iSPF, resulting in faster network wide convergence

Event Processing

Incremental SPF Convergence Times

Stub Link Down Event (IP Prefix Lost):

Full SPF:

```
Sep 25 14:07:37.795:    OSPF: Begin SPF at 187751.852ms, process time 149100ms
Sep 25 14:07:37.795:    spf_time 2d04h, wait_interval 10s
Sep 25 14:07:37.839:    OSPF: End SPF at 187751.896ms,Total elapsed time 44ms
Sep 25 14:07:37.839:    Intra: 44ms, Inter: 0ms, External: 0ms
Sep 25 14:07:37.839:    R: 506, N: 786, Stubs: 620
Sep 25 14:07:37.839:    SN: 0, SA: 0, X5: 0, X7: 0
Sep 25 14:07:37.839:    SPF suspends: 0 intra, 0 total
```

Incremental SPF:

```
Sep 25 14:06:27.715:    OSPF: Begin SPF at 187681.772ms, process time 149016ms
Sep 25 14:06:27.715:    spf_time 2d04h, wait_interval 10s
Sep 25 14:06:27.719:    OSPF: End SPF at 187681.776ms,Total elapsed time 4ms
Sep 25 14:06:27.719:    Incremental-SPF: 0ms
Sep 25 14:06:27.719:    Intra: 0ms, Inter: 0ms, External: 0ms
Sep 25 14:06:27.719:    R: 0, N: 0, Stubs: 1
Sep 25 14:06:27.719:    SN: 0, SA: 0, X5: 0, X7: 0
Sep 25 14:06:27.723:    SPF suspends: 0 intra, 0 total
```

Event Processing

Incremental SPF Convergence Times

Link Up Event—Part of the Network Becomes Reachable:

Full SPF:

```
Sep 25 14:27:13.463:    OSPF: Begin SPF at 188927.520ms, process time 149760ms
Sep 25 14:27:13.463:    spf_time 2d04h, wait_interval 5s
Sep 25 14:27:13.515:    OSPF: End SPF at 188927.572ms,Total elapsed time 52ms
Sep 25 14:27:13.515:    Intra: 48ms, Inter: 0ms, External: 0ms
Sep 25 14:27:13.515:    R: 488, N: 758, Stubs: 598
Sep 25 14:27:13.515:    SN: 0, SA: 0, X5: 0, X7: 0
Sep 25 14:27:13.515:    SPF suspends: 0 intra, 0 total
```

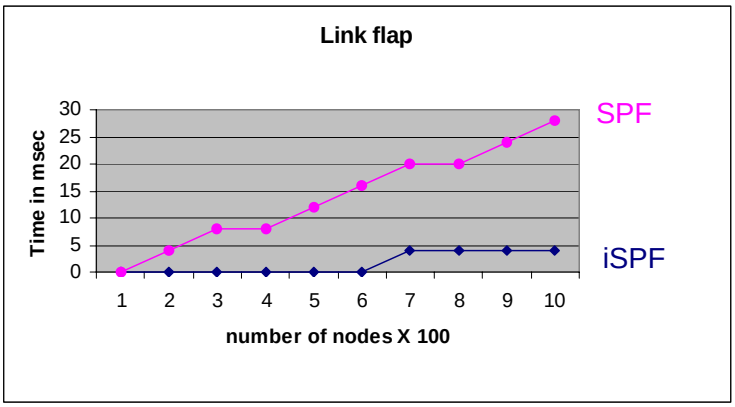
Incremental SPF:

```
Sep 25 14:23:13.467:    OSPF: Begin SPF at 188687.524ms, process time 149612ms
Sep 25 14:23:13.467:    spf_time 2d04h, wait_interval 5s
Sep 25 14:23:13.479:    OSPF: End SPF at 188687.536ms,Total elapsed time 12ms
Sep 25 14:23:13.479:    Incremental-SPF: 0ms
Sep 25 14:23:13.479:    Intra: 8ms, Inter: 0msExternal: 0ms
Sep 25 14:23:13.479:    R: 18, N: 29, Stubs: 22
Sep 25 14:23:13.479:    SN: 0, SA: 0, X5: 0, X7: 0
Sep 25 14:23:13.479:    SPF suspends: 0 intra, 0 total
```


Event Processing

Incremental SPF Convergence Times

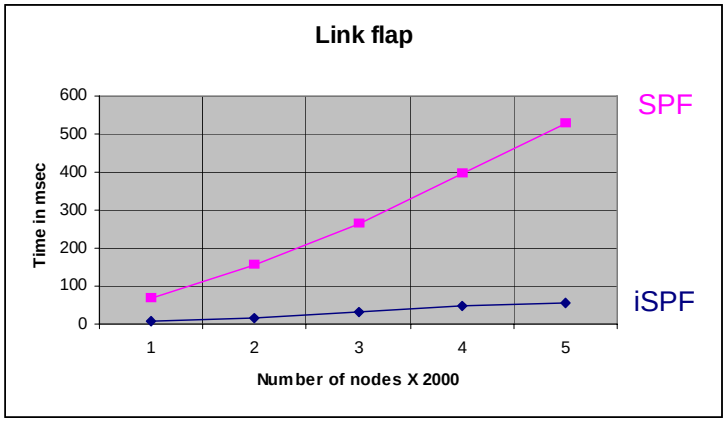
Time It Takes to Run the SPF with the Transit Link Flap



Event Processing

Incremental SPF Convergence Times

Time It Takes to Run the SPF with the Transit Link Flap



Event Processing

Incremental SPF Convergence Times

- Supported: 12.0(24)S, 12.2(18)S, 12.3(2)T
- Configuration:

```
router ospf <process number>
ispf
```

- 'sh ip ospf'

```
Routing Process "ospf 1" with ID 170.99.99.99 and Domain ID 0.0.0.1
Supports only single TOS(TOS0) routes
Supports opaque LSA
It is an area border and autonomous system boundary router
Redistributing External Routes from,
SPF schedule delay 5 secs, Hold time between two SPFs 10 secs
Incremental-SPF enabled
Minimum LSA interval 5 secs. Minimum LSA arrival 1 secs
```

"…determines the minimum amount of time that must elapse between an advertisement and/or withdrawal of routes to a particular destination by a BGP speaker to a peer. This rate limiting procedure applies on a per-destination basis, although the value of MinRouteAdvertisementIntervalTimer is set on a per BGP peer basis."

RFC 4271

Section 9.2.1.1

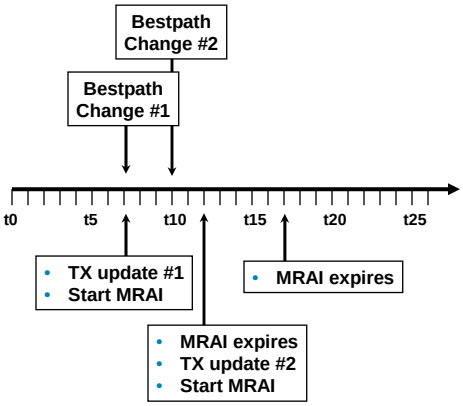
MRAI—Basics

- MRAI timers are maintained per peer
  - iBGP—five seconds by default
  - eBGP—30 seconds by default
  - `neighbor x.x.x.x advertisement-interval <0-600>`
- Popular misconception that withdraws are not affected
- Pros
  - Promotes stability by batching route changes
  - Improves update packing in some situations
- Cons
  - May drastically slow convergence
  - Current defaults are too conservative
  - One flapping prefix can slow convergence for other prefixes

MRAI—Implementation

- How is the timer enforced for peer X?
  - Timer starts when all routes have been advertised to X
  - For the next MRAI (seconds) we will not propagate any bestpath changes to peer X
  - Once X's MRAI timer expires, send him updates and withdraws
  - Restart the timer and the process repeats…
- User may see a wave of updates and withdraws to peer X every MRAI
- User will **not** see a delay of MRAI between each individual update and/or withdraw
  - BGP would probably never converge if this was the case

MRAI—Implementation

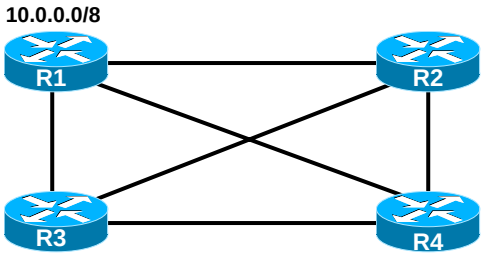


- MRAI timeline for iBGP peer
- Bestpath change #1 at t7 is TXed immediately
- MRAI timer starts at t7, will expire at t12
- Bestpath change #2 at t10 must wait until t12 for MRAI to expire
- Bestpath change #2 is TXed at t12
- MRAI timer starts at t12, will expire at t17
- MRAI expires at t17… no updates are pending

MRAI—Slows Convergence

- BGP is not a link-state protocol, but instead is path vector based
- May take several “rounds/cycles” of exchanging updates and withdraws for the network to converge
- MRAI must expire between each round!
- The more fully meshed the network and the more tiers of autonomous systems, the more rounds required for convergence
- Think about
  - The many tiers of autonomous systems that are in the Internet
  - The degree to which peering can be fully meshed

MRAI—Convergence Example



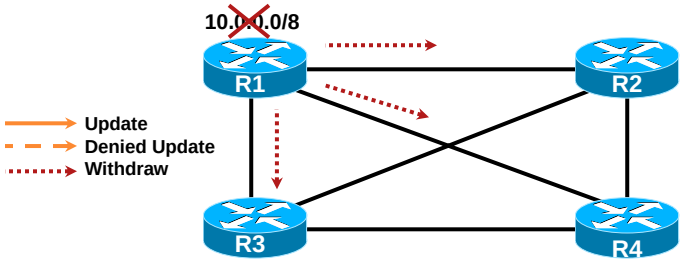
R2—10.0.0.0/8
- **Via R1**
- **Via R3, R1**
- **Via R4, R1**

R3—10.0.0.0/8
- **Via R1**
- **Via R2, R1**
- **Via R4, R1**

R4—10.0.0.0/8
- ***Via R1***
- **Via R2, R1**
- **Via R3, R1**

- Full mesh is the worst case MRAI convergence scenario
- R1 will send a withdraw to all peers for 10.0.0.0/8
- Count the number of rounds of **updates** and withdraws until the network has converged
- Note how MRAI slows convergence
- **Orange** path is the bestpath

MRAI—Convergence Example



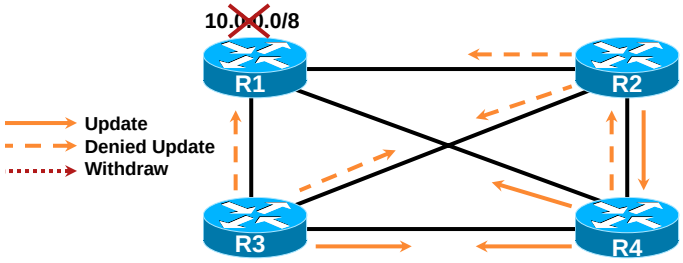
- R1 withdraws 10.0.0.0/8 to all peers
- R1 starts a MRAI timer for each peer

R2—10.0.0.0/8
- ~~Via R1~~
- Via R3, R1
- Via R4, R1

R3—10.0.0.0/8
- ~~Via R1~~
- Via R2, R1
- Via R4, R1

R4—10.0.0.0/8
- ~~Via R1~~
- Via R2, R1
- Via R3, R1

MRAI—Convergence Example



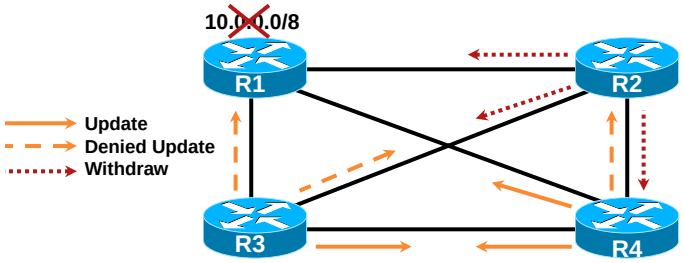
- R2, R3, and R4 recalculate their bestpaths
- R2, R3, and R4 send updates based on new bestpaths
- R2, R3, and R4 start a MRAI timer for each peer
- End of round 1

R2—10.0.0.0/8
- ~~Via R1~~
- Via R3, R1
- Via R4, R1

R3—10.0.0.0/8
- ~~Via R1~~
- Via R2, R1
- Via R4, R1

R4—10.0.0.0/8
- ~~Via R1~~
- Via R2, R1
- Via R3, R1

MRAI—Convergence Example



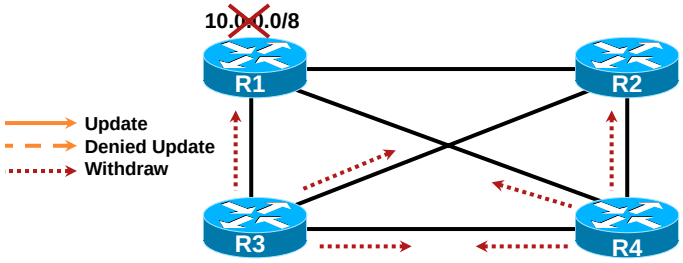
R2—10.0.0.0/8
- ~~Via R1~~
- ~~Via R3, R1~~
- ~~Via R4, R1~~

R3—10.0.0.0/8
- ~~Via R1~~
- ~~Via R2, R1~~
- Via R4, R2, R1

R4—10.0.0.0/8
- ~~Via R1~~
- ~~Via R2, R1~~
- Via R3, R2, R1
- Via R2, R3, R1

- R2, R3, and R4 recalculate their bestpaths
- R2, R3 and R4 must wait for their MRAI timers to expire!
- R2, R3, and R4 send updates and withdraws based on their new bestpaths
- R2, R3, and R4 restart the MRAI timer for each peer
- End of round 2

MRAI—Convergence Example



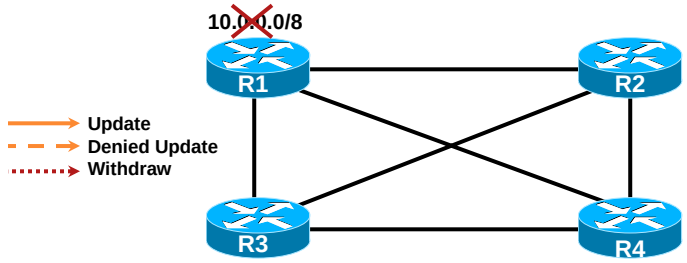
R2—10.0.0.0/8
- ~~Via R1~~
- ~~Via R3, R1~~
- ~~Via R4, R1~~

R3—10.0.0.0/8
- ~~Via R1~~
- ~~Via R2, R1~~
- ~~Via R4, R2, R1~~

R4—10.0.0.0/8
- ~~Via R1~~
- ~~Via R2, R1~~
- ~~Via R3, R2, R1~~
- ~~Via R2, R3, R1~~

- R3 and R4 recalculate their bestpaths
- R3 and R4 must wait for their MRAI timers to expire!
- R3 and R4 send updates and withdraws based on their new bestpaths
- R3 and R4 restart the MRAI timer for each peer
- End of round 3

MRAI—Convergence Example



R2—10.0.0.0/8
- ~~Via R1~~
- ~~Via R3, R1~~
- ~~Via R4, R1~~

R3—10.0.0.0/8
- ~~Via R1~~
- ~~Via R2, R1~~
- ~~Via R4, R2, R1~~

R4—10.0.0.0/8
- ~~Via R1~~
- ~~Via R2, R1~~
- ~~Via R3, R2, R1~~
- ~~Via R2, R3, R1~~

- R2, R3, and R4 took three rounds of messages to converge
- MRAI timers had to expire between 1st/2nd round and between 2nd/3rd round
- Total MRAI convergence delay for this example
  - iBGP mesh—10 seconds
  - eBGP mesh—60 seconds

MRAI—Tuning

- Internet churn means we are constantly setting and waiting on MRAI timers
  - One flapping prefix slows convergence for all prefixes
  - Internet table sees roughly 1-2 bestpath changes per second
  - Based on Geoff Huston's research:
    - http://www.potaroo.net/presentations/2006-11-03-caida-wide.pdf
- For iBGP and PE→CE eBGP peers
  - `neighbor x.x.x.x advertisement-interval 0`
  - Will be the default in 12.0(32)S
- For regular eBGP peers
  - Lowering to 0 may get you dampened
  - OK to lower for eBGP peers if they are not using dampening

MRAI—Tuning

- Will a MRAI of 0 eliminate batching?
  - Somewhat but not much happens anyway
  - TCP, the operating system, and BGP code provide some batching
    - Process all message from peer InQs
    - Calculate bestpaths based on received messages
    - Format **updates** to advertise new bestpaths
- What about CPU load from 0 second MRAI?
  - Internet table has ~1–2 bestpath changes per second
  - This number may differ for you, your mileage may vary
  - Easy for a router under normal conditions to handle, five seconds of delay is not needed